

BlackBerry Administration API For Microsoft .NET Developers

Version: 5.0 | Service Pack: 1

Getting Started Guide

Contents

1	Setting up a development BlackBerry Enterprise Server.....	2
	Obtaining a license and installing a development BlackBerry Enterprise Server.....	2
	Creating administrator accounts for your application.....	3
	Permissions for the BlackBerry Administration API.....	3
	Create a role.....	7
	Create a role based on an existing role.....	7
	Create an administrator account.....	8
	Add a user account to the development BlackBerry Enterprise Server.....	8
2	Configuring your development computer.....	10
	System requirements: BlackBerry Administration API development computers.....	10
	Add the development BlackBerry Enterprise Server as a trusted authority.....	10
	Generate the client proxy.....	11
3	Creating your first application.....	12
	Create a project.....	12
	Code sample: Creating your first application.....	12
4	Glossary.....	18
5	Provide feedback.....	19
6	Legal notice.....	20

Setting up a development BlackBerry Enterprise Server

1

You can install one or more BlackBerry® Enterprise Server instances that you can use to develop, debug, and test your application. A development BlackBerry Enterprise Server can help you avoid accidental changes to important settings on a production BlackBerry Enterprise Server, for example, the deletion of user accounts. To create an application that uses the BlackBerry® Administration API web services, the computer that you use to develop your application must have network access to a BlackBerry Enterprise Server on which the BlackBerry Administration Service and the BlackBerry Administration API are installed.

The development BlackBerry Enterprise Server should match, as closely as possible, the environment in which you want to use your application. The APIs that are available in the BlackBerry Administration API might vary depending on the components that are installed on the BlackBerry Enterprise Server. If a component that is installed on the development BlackBerry Enterprise Server is not available on the production BlackBerry Enterprise Server, the functionality that is provided by that component is available on the production BlackBerry Enterprise Server. For example, if the development BlackBerry Enterprise Server uses a messaging platform that is different from your production BlackBerry Enterprise Server, such as Microsoft® Exchange instead of IBM® Lotus® Domino®, the methods that are available in the web service might stop working when you use your application with your production BlackBerry Enterprise Server.

When you set up your BlackBerry Enterprise Server instances for development, you can add the default certificate to the Trusted Root Certificate Authority key store on each development computer instead of replacing the default SSL certificate with one signed by a certificate authority.

If you use BlackBerry Enterprise Server 5.0 SP1 or newer, the BlackBerry Administration API is installed when you install the BlackBerry Administration Service. If you use the BlackBerry Enterprise Server version 5.0, you must manually install the BlackBerry Administration API. If you upgrade the BlackBerry Enterprise Server from version 5.0 to version 5.0 SP1 or later, you should remove the BlackBerry Administration API before you upgrade. The BlackBerry Administration API is not supported in the BlackBerry Enterprise Server earlier than version 5.0.

Obtaining a license and installing a development BlackBerry Enterprise Server

To install a development BlackBerry® Enterprise Server, you must download the BlackBerry Enterprise Server software and obtain a BlackBerry® Client Access License key and the SRP credentials from Research In Motion. If you require a development BlackBerry Enterprise Server for a short period of time, you can download a 60-day trial version. For longer or ongoing development projects, you must purchase a BlackBerry Enterprise Server license.

To download a trial version or purchase a license, visit www.blackberry.com. For more information about installing your BlackBerry Enterprise Server for development, see the *BlackBerry Enterprise Server Installation Guide* for your messaging platform.

Creating administrator accounts for your application

When your application invokes an API, your application must provide an administrator account name and password for the BlackBerry® Administration Service to authenticate your application and authorize your application to use the API. Applications can use the same administrator account name and password that administrators use to log in to the BlackBerry Administration Service. You can provide the developers with the credentials of an existing administrator account or you can create a new account specifically for applications that use the BlackBerry® Administration API.

An application that uses the BlackBerry Administration API requires an administrator account that has the appropriate permissions. Permissions are associated with each task that an administrator, or an application, can perform in the BlackBerry Administration Service. If the administrator account that an application uses does not have the necessary permissions to complete an API request, the request is not completed. Make sure that your developers use administrator accounts that have the necessary permissions to perform the required tasks. The permissions for tasks that an administrator or application can perform are defined by the roles that are assigned to the administrator account.

For more information on the permissions that developers require to use a specific API, see the API reference for the BlackBerry Administration API.

Permissions for the BlackBerry Administration API

The functionality that is available in the BlackBerry Administration API might vary depending on the components that are installed on your BlackBerry® Enterprise Server. You might require additional permissions to those listed below. Some permissions and their associated BlackBerry Administration API methods might not be available on your BlackBerry Enterprise Server, which means that they do not appear in your BlackBerry Administration Service or WSDL files.

The following list describes the permissions that you require to access most of the methods that you might use in the BlackBerry Administration API.

Tab	Permission	Associated BlackBerry Administration API methods
User and device	Create a group	<code>createGroup()</code>
User and device	Delete a group	<code>deleteGroup()</code>
User and device	View a group	<ul style="list-style-type: none"> • <code>getUser()</code> • <code>findGroups()</code> • <code>findITPoliciesForGroup()</code> • <code>findUsers()</code> • <code>getGroup()</code> • <code>setGroup()</code> • <code>setUser()</code>

Tab	Permission	Associated BlackBerry Administration API methods
User and device	Edit a group	<ul style="list-style-type: none"> • <code>findSoftwareConfigurationsForGroup()</code> • <code>createUser()</code> • <code>createUsers()</code> • <code>addGroupToGroup()</code> • <code>addUserToGroup()</code> • <code>removeUserFromGroup()</code> • <code>removeGroupFromGroup()</code> • <code>setGroup()</code>
User and device	Create a user	<ul style="list-style-type: none"> • <code>createUser()</code> • <code>createUsers()</code>
User and device	Delete a user	<ul style="list-style-type: none"> • <code>deleteUsers()</code> • <code>deleteUser()</code>
User and device	View a user	<ul style="list-style-type: none"> • <code>findSoftwareConfigurationsForUser()</code> • <code>findGroups()</code> • <code>findUsers()</code> • <code>findITPoliciesForUser()</code> • <code>findResolvedUserCapabilities()</code> • <code>findPersonalRedirectionFolderInformation()</code> • <code>gatherEnableBlackBerryUserAttributes()</code>
User and device	Edit a user	<ul style="list-style-type: none"> • <code>createUser()</code> • <code>createUsers()</code> • <code>addUserToGroup()</code> • <code>removeUserFromGroup()</code> • <code>addUsersToGroup()</code> • <code>setUser()</code> • <code>disableBlackBerryServiceForUsers()</code> • <code>moveUser()</code> • <code>setPersonalRedirectionFolderInformation()</code>
User and device	View a device	<ul style="list-style-type: none"> • <code>findUsers()</code> • <code>findDeviceForUser()</code> • <code>isKillHandheldPending()</code>

Tab	Permission	Associated BlackBerry Administration API methods
User and device	Edit a device	<ul style="list-style-type: none"> • <code>setDevicePassword()</code> • <code>cancelKillHandheld()</code>
User and device	View an IT Policy	<ul style="list-style-type: none"> • <code>findVPNConfigurations()</code> • <code>findITPoliciesForGroup()</code> • <code>findITPolicies()</code> • <code>findUsers()</code> • <code>findITPoliciesForUser()</code> • <code>getUser()</code> • <code>findWLANConfigurations()</code>
User and device	Resend data to devices	<ul style="list-style-type: none"> • <code>resendReconciledApplicationsToUser()</code> • <code>resendPeerToPeerKeyToUser()</code> • <code>resendServiceBookToUsers()</code> • <code>resendServiceBookToUser()</code>
User and device	View a software configuration	<ul style="list-style-type: none"> • <code>findSoftwareConfigurationsForUser()</code> • <code>findUsers()</code> • <code>findSoftwareConfigurationsForGroup()</code>
User and device	Add or remove to user configuration	<ul style="list-style-type: none"> • <code>createUser()</code> • <code>createUsers()</code> • <code>detachSoftwareConfigurationFromUser()</code> • <code>detachSoftwareConfigurationFromGroup()</code> • <code>assignVPNConfigurationToUser()</code> • <code>unassignWLANConfigurationFromUser()</code> • <code>unassignVPNConfigurationFromUser()</code> • <code>assignWLANConfigurationToUser()</code> • <code>attachITPolicyToUser()</code>
User and device	Specify activation password	<ul style="list-style-type: none"> • <code>setActivationPassword()</code> • <code>generateActivationPassword()</code> • <code>clearActivationPassword()</code>
User and device	Turn on redirection	<ul style="list-style-type: none"> • <code>enableUserMailRedirection()</code> • <code>enableUsersMailRedirection()</code>
User and device	Turn off redirection	<ul style="list-style-type: none"> • <code>enableUserMailRedirection()</code>

Tab	Permission	Associated BlackBerry Administration API methods
		<ul style="list-style-type: none"> enableUsersMailRedirection()
Topology	View a Server	findServerByHostName()
Topology	View a component	findServices()
Topology	View an Instance	<ul style="list-style-type: none"> findServiceInstancesByService() findServiceInstancesByServer()
Topology	View rules for the BlackBerry MDS Connection Service	findUsers()
BlackBerry Administration Service setup	Send message	<ul style="list-style-type: none"> sendMessageToUser() sendMessageToUsers()
BlackBerry Administration Service setup	View a role	<ul style="list-style-type: none"> findUsers() findResolvedUserCapabilities()
BlackBerry Administration Service setup	Add or remove a role	findCapabilityDefinitions()
BlackBerry Administration Service setup	View BlackBerry Administration Service software management	<ul style="list-style-type: none"> findSoftwareConfigurations() findSoftwareConfigurationApplications()
Organizations	View a group across organizations	<ul style="list-style-type: none"> setGroup() findITPoliciesForGroup() findGroups() findUsers() getUser() removeUserFromGroup() setUser() findSoftwareConfigurationsForGroup()
Organizations	Edit a group across organizations	<ul style="list-style-type: none"> createUser() createUsers() addUserToGroup() setGroup()

Tab	Permission	Associated BlackBerry Administration API methods
		<ul style="list-style-type: none"> removeGroupFromGroup() setUser()
Organizations	Add or remove a role across organizations	findCapabilityDefinitions()
Organizations	View a device across organizations	<ul style="list-style-type: none"> findUsers() findDeviceForUser() isKillHandheldPending()
Organizations	Edit a device across organizations	<ul style="list-style-type: none"> setDevicePassword() cancelKillHandheld()

For more information about the permission requirements for a specific API, see the API reference for the BlackBerry Administration API.

Create a role

You can create a role for an administrator account if existing roles do not match the criteria that your organization specified for a type of administrator account. By default, when you create a role, all permissions are turned off.

1. In the BlackBerry® Administration Service, on the **BlackBerry solution management** menu, expand **Role**.
2. Click **Create a role**.
3. Type a name and description for the role.
4. Click **Save**.
5. In the **Role information** section, click the name of the role that you created.
6. Click **Edit role**.
7. Switch the appropriate tabs to turn on the appropriate permissions.
8. Click **Save all**.

After you finish: Assign the role to an administrator account or group.

Create a role based on an existing role

To create a role for administrator accounts that is similar to an existing role, you can copy the existing role and make the appropriate changes to it.

1. In the BlackBerry® Administration Service, on the **BlackBerry solution management** menu, expand **Role**.
2. Click **Manage roles**.
3. In the list of existing roles, click the role that you want to copy.

4. Click **Copy role**.
5. Type a name and description for the role.
6. Click **Copy role**.
7. In the **Role information** section, click the name of the role that you created.
8. Click **Edit role**.
9. Switch the appropriate tabs to change the appropriate permissions.
10. Click **Save all**.

After you finish: Assign the role to an administrator account or group.

Create an administrator account

You can create an administrator account when you want to assign administrative permissions to an administrator in your organization.

Before you begin: Verify that you can configure the authentication type and roles for an administrator account.

1. In the BlackBerry® Administration Service, on the **BlackBerry solution management** menu, expand **Administrator user**.
2. Click **Create an administrator user**.
3. Type the required information.
4. In the **Role** drop-down list, click the role that you want to assign to the administrator account.
5. Click **Create an administrator user**.

After you finish: To configure the administrator account, provide the login information to the administrator and add the administrator account to a group or assign additional roles to the administrator account.

Add a user account to the development BlackBerry Enterprise Server

You must add at least one user account on the BlackBerry® Enterprise Server to run the code sample *Creating your first application*.

Before you begin:

Contact your messaging server administrator to request some user accounts on the messaging server, that you can use when you develop and test your application.

1. In the BlackBerry Administration Service, on the **BlackBerry solution management** menu, expand **User**.
2. Click **Create user**.
3. Search for a user account.
4. Select the check box beside the display name for the user account.
5. Click **Continue**.

6. If your organization's environment includes multiple BlackBerry Enterprise Server instances, select the development BlackBerry Enterprise Server that you want to add the user account to.
7. Click **Create a user without activation password**.

Configuring your development computer

2

System requirements: BlackBerry Administration API development computers

Make sure that you install the required software on each computer that you use to develop applications for the BlackBerry® Administration API.

Software	Description
Windows® operating system	This the supported operating system.
Microsoft Visual Studio 2005 or 2008	This is the IDE and supporting frameworks that you use to develop and build your application

Add the development BlackBerry Enterprise Server as a trusted authority

If you want to view the WSDL files on a BlackBerry® Enterprise Server, Windows® Internet Explorer® must trust the self-signed SSL certificate that the BlackBerry Administration Service uses by default. If you want to generate the client proxy files by using the wsdl.exe file, Microsoft® Visual Studio requires that the development BlackBerry Enterprise Server is a trusted server.

Before you begin: You must use Windows Internet Explorer 6.0 or 7.0. For more information about downloading certificates by using other browsers or versions, see the help or documentation for the browser.

1. In Windows Internet Explorer, visit the login web page for the BlackBerry Administration Service (for example, <https://<servername>/webconsole/app>, where <servername> is the name of the computer that hosts the BlackBerry Administration Service).
2. On the **File** menu, click **Properties**.
3. Click **Certificates**.
4. Click **Install Certificate**.
5. Complete the instructions on the screen to install the certificate in the **Trusted Root Certification Authorities** certificate store.

After you finish: To verify that you successfully installed the certificate and that your BlackBerry® Administration API is working, visit <https://<servername>/baaws/core/ws?wsdl>. You should see the WSDL definition file for the **BAASer viceCore**, which starts with the following line:

```
<definitions name="BAASer viceCore" targetNamespace="http://www.rim.com/com.rim.bes.bas.baa"
```

Generate the client proxy

You can generate a client proxy by using the proxy generator called `wSDL.exe`, which is installed with Microsoft® Visual Studio®.

1. In the Microsoft Visual Studio installation folder, locate the bin folder that contains the proxy generator called `wSDL.exe` (for example, `C:\Program Files\Microsoft Visual Studio 8\SDK\v2.0\Bin\wSDL.exe`).
2. At the Windows® command prompt, navigate to the bin folder that contains the proxy generator.

```
cd C:\Program Files\Microsoft Visual Studio 8\SDK\v2.0\Bin\wSDL.exe
```

3. Invoke `wSDL.exe` to generate the client proxy, and add as an argument the address of each WSDL file that you want to use.

```
wSDL.exe /sharetypes /out:C:\Temp\proxy.cs  
https://<servername>/baaws/core/ws?wSDL  
https://<servername>/baaws/emailexchange/ws?wSDL  
https://<servername>/baaws/core/wsutil?wSDL  
https://<servername>/baaws/dispatcher/ws?wSDL
```

Creating your first application

3

To verify that you have correctly set up your development BlackBerry® Enterprise Server and your development computer, you can set up a project in Microsoft® Visual Studio® and run the code sample *Creating your first application*.

Create a project

1. In Microsoft® Visual Studio®, on the **File** menu, click **New > Project**.
2. Click **Console Application**.
3. Type a name for the project.
4. Click **OK**.
5. In the **Solution Explorer** pane, right-click the application name and click **Add > Existing Item**.
6. Navigate to the output file that you specified when you created the client proxy (for example, C:\Temp\proxy.cs).
7. Click **Add**.
8. Add the *Creating your first application* code sample to a source file. The default source file is Program.cs.

Code sample: Creating your first application

The following code sample demonstrates how to find an existing user account on your BlackBerry® Enterprise Server. To run the code sample, you must specify a user account and other account information in the `main()` method.

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Security.Cryptography.X509Certificates;
using System.Net;
using System.Net.Security;
using System.IO;

namespace CSClient
{
    class Program
    {
        public static BAAServiceCore coreWebService = new BAAServiceCore();
        public static BAAServiceUtil utilWebService = new BAAServiceUtil();
        public static BAAServiceEmailExchange emailExchangeWebService = new
BAAServiceEmailExchange();

        public static void setup(String basURL, String userName, String password)
        {
            //point proxy to the BAS based on URL
```

```

        coreWebService.Url = "https://" + baseUrl + "/baaws/core/ws?wsdl";
        utilWebService.Url = "https://" + baseUrl + "/baaws/core/wsutil?wsdl";
        emailExchangeWebService.Url = "https://" + baseUrl + "/baaws/emailexchange/
ws?wsdl";

        //set up ssl validation
        //System.Net.ServicePointManager.ServerCertificateValidationCallback =
ValidateServerCertificate;

        //use the utility web service to encode the username
        encodeUsername encodeUsernameRequest = new encodeUsername();
        encodeUsernameRequest.username = userName;
        System.Console.WriteLine("Encoding username");
        encodeUsernameResponse encodeUsernameResponse =
utilWebService.encodeUsername(encodeUsernameRequest);
        System.Console.WriteLine("Done encoding username");

        //set username and password for the core proxy

        String username = encodeUsernameResponse.returnValue;
        coreWebService.Credentials = new NetworkCredential(username, password);
        coreWebService.PreAuthenticate = true;

        emailExchangeWebService.Credentials = new NetworkCredential(username,
password);
        emailExchangeWebService.PreAuthenticate = true;
    }

    //deal with SSL certificate validation
    public static bool ValidateServerCertificate(object sender, X509Certificate
certificate, X509Chain chain, SslPolicyErrors sslPolicyErrors)
    {
        //should check certificate here.
        System.Console.WriteLine(certificate.GetEffectiveDateString());
        System.Console.WriteLine(certificate.GetExpirationDateString());
        System.Console.WriteLine(certificate.Issuer);
        System.Console.WriteLine(certificate.Subject);
        return true;
    }

    public static void testFindUsers(String userDisplayName, String
userEmailAddress)
    {
        //call findUsers api in the core web service
        findUsersResponse response;

        findUsers findUsersRequest = new findUsers();

        UserSearchCriteria searchCriteria = new UserSearchCriteria();
        searchCriteria.userSearchCriteriaServiceAttributes = new
UserSearchCriteriaServiceAttributesType[2];

```

```

        BASUserSearchCriteriaServiceAttributes basServiceAttributes = new
BASUserSearchCriteriaServiceAttributes();
        basServiceAttributes.displayName = userDisplayName;
        searchCriteria.userSearchCriteriaServiceAttributes[0] =
basServiceAttributes;

        EmailCommonUserSearchCriteriaServiceAttributesType emailCriteriaAttribute
= new EmailExchangeUserSearchCriteriaServiceAttributes();
        emailCriteriaAttribute.userEmailAddress = userEmailAddress;
        searchCriteria.userSearchCriteriaServiceAttributes[1] =
emailCriteriaAttribute;

        findUsersRequest.searchCriteria = searchCriteria;

        int pageSize = 3;
        UserSearchResult lastResultElement = null;

        findUsersRequest.locale = "en_US";

        findUsersRequest.sortAscendingOrder = true;

        //set the page size
        findUsersRequest.pageSize = pageSize;

        //prepare and set the sort by
BASUserSearchResultSortByEnum sortBy = new BASUserSearchResultSortByEnum();
        sortBy.@enum = BASUserSearchResultSortByEnumType.DISPLAY_NAME;
        sortBy.enumSpecified = true;
        findUsersRequest.sortBy = sortBy;

        //call findUsers api
        response = coreWebService.findUsers(findUsersRequest);

        //loop through pages
        while (response.returnValue.userSearchResult != null)
        {
            //print metadata
            System.Console.WriteLine("API Status [{0}]; Execution Time [{1}]s",
                response.returnValue.metadata.APIStatusSpecified ?
response.returnValue.metadata.APIStatus : APIStatusEnumType.NOT_YET_IMPLEMENTED,
                response.returnValue.metadata.executionTimeSpecified ?
response.returnValue.metadata.executionTime / 1000000000.0 : -1);

            //check return status
            if (response.returnValue.findUsersReturnStatus.code !=
FindUsersReturnStatusEnumType.SUCCESS)
            {
                System.Console.Error.WriteLine
(response.returnValue.findUsersReturnStatus.message);
                break;
            }
        }

```

```

        //print number of search results on this page
        System.Console.WriteLine("results # [{0}]",
response.returnValue.userSearchResult.Length);

        //print the results:
        for (int i = 0; i < response.returnValue.userSearchResult.Length; i++)
        {
            //Each user response element is a rainbow object
            UserSearchResult searchResult = response.returnValue.userSearchResult
[i];
            UserSearchResultServiceAttributesType[] serviceAttributes =
searchResult.userSearchResultServiceAttributes;

            //loop through the service attributes and determine the concrete
            //datatype of each attribute
            for (int j = 0; j < serviceAttributes.Length; j++)
            {
                if (serviceAttributes[j].GetType().Equals(typeof
(BASUserSearchResultServiceAttributes)))
                {
                    BASUserSearchResultServiceAttributesType
basServiceResultAttributes = serviceAttributes[j] as
BASUserSearchResultServiceAttributesType;
                    System.Console.WriteLine("\tName:" +
basServiceResultAttributes.displayName);
                }
                else if (serviceAttributes[j].GetType().Equals(typeof
(DispatcherUserSearchResultServiceAttributes)))
                {
                    DispatcherUserSearchResultServiceAttributesType
dispatcherResultAttributes = serviceAttributes[j] as
DispatcherUserSearchResultServiceAttributesType;
                    System.Console.WriteLine("\tDispatcher service instance
name:" + dispatcherResultAttributes.dispatcherServiceInstanceName);
                }
                else if (serviceAttributes[j].GetType().Equals(typeof
(EmailExchangeUserSearchResultServiceAttributes)))
                {
                    EmailExchangeUserSearchResultServiceAttributesType
emailResultAttributes = serviceAttributes[j] as
EmailExchangeUserSearchResultServiceAttributesType;
                    System.Console.WriteLine(String.Format("\tEmail Server[{0}];
email address [{1}]; User State [{2}]:",
                    emailResultAttributes.mailServerName,
                    emailResultAttributes.userEmailAddress,
                    emailResultAttributes.userState));
                    if (emailResultAttributes.lastContactDateSpecified)
                    {
                        System.Console.WriteLine
(emailResultAttributes.lastContactDate.ToString());
                    }
                }
            }
        }
    }
}

```

```

        }
        }
        else
        {
            System.Console.WriteLine("\t attribute " + serviceAttributes
[j].GetType().Name);
        }
    }
    System.Console.WriteLine("");
}

//set the last element to be the last search result entry
lastResultElement = response.returnValue.userSearchResult
[response.returnValue.userSearchResult.Length - 1];
findUsersRequest.lastResultElement = lastResultElement;

//get the next page
response = coreWebService.findUsers(findUsersRequest);
}
}

static void Main(string[] args)
{
    try
    {
        //Setup BlackBerry Administration Service url used to connect to the
        //web service.
        //Example: basURL = server01.testnet.rim.net. This is the fully
        //qualified domain name, which you would use to log into the
        //BlackBerry Administration Service.
        String basURL = "Enter your BAS URL";

        System.Console.WriteLine("Setup");
        //example: setup(basURL, "Admin", "password");
        setup(basURL, "<Administrator user name>", "<Administrator password>");

        //call find user to display all current users
        System.Console.WriteLine("Find Users");
        testFindUsers("", "");

        //create a new user by adding it to a dispatcher host instance
        //Example: userDisplayName = "Megan Ball";
        //Example: Megan.Ball@rim.com
        String userEmailAddress = "<user email address to add>";
        String userDisplayName = "<user display name to add>";

        //Example: dispatcherInstanceName = "TestServer";
        String dispatcherInstanceName = "<BlackBerry Dispatcher name>";
    }
}

```

```
        //display the newly added user.  
        testFindUsers(userDisplayName, userEmailAddress);  
        Console.ReadLine();  
    }  
    catch (System.Exception e)  
    {  
        Console.Error.WriteLine(e.Message);  
    }  
}  
}
```

Glossary

4

API

application programming interface

JRE

Java® Runtime Environment

SSL

Secure Sockets Layer

WSDL

Web Services Description Language

Provide feedback

5

To provide feedback on this deliverable, visit www.blackberry.com/docsfeedback.

Legal notice

6

©2009 Research In Motion Limited. All rights reserved. BlackBerry®, RIM®, Research In Motion®, SureType®, SurePress™ and related trademarks, names, and logos are the property of Research In Motion Limited and are registered and/or used in the U.S. and countries around the world.

Microsoft and Visual Studio are trademarks of Microsoft Corporation.. All other trademarks are the property of their respective owners.

The BlackBerry smartphone and other devices and/or associated software are protected by copyright, international treaties, and various patents, including one or more of the following U.S. patents: 6,278,442; 6,271,605; 6,219,694; 6,075,470; 6,073,318; D445,428; D433,460; D416,256. Other patents are registered or pending in the U.S. and in various countries around the world. Visit www.rim.com/patents for a list of RIM (as hereinafter defined) patents.

This documentation including all documentation incorporated by reference herein such as documentation provided or made available at www.blackberry.com/go/docs is provided or made accessible "AS IS" and "AS AVAILABLE" and without condition, endorsement, guarantee, representation, or warranty of any kind by Research In Motion Limited and its affiliated companies ("RIM") and RIM assumes no responsibility for any typographical, technical, or other inaccuracies, errors, or omissions in this documentation. In order to protect RIM proprietary and confidential information and/or trade secrets, this documentation may describe some aspects of RIM technology in generalized terms. RIM reserves the right to periodically change information that is contained in this documentation; however, RIM makes no commitment to provide any such changes, updates, enhancements, or other additions to this documentation to you in a timely manner or at all.

This documentation might contain references to third-party sources of information, hardware or software, products or services including components and content such as content protected by copyright and/or third-party web sites (collectively the "Third Party Products and Services"). RIM does not control, and is not responsible for, any Third Party Products and Services including, without limitation the content, accuracy, copyright compliance, compatibility, performance, trustworthiness, legality, decency, links, or any other aspect of Third Party Products and Services. The inclusion of a reference to Third Party Products and Services in this documentation does not imply endorsement by RIM of the Third Party Products and Services or the third party in any way.

EXCEPT TO THE EXTENT SPECIFICALLY PROHIBITED BY APPLICABLE LAW IN YOUR JURISDICTION, ALL CONDITIONS, ENDORSEMENTS, GUARANTEES, REPRESENTATIONS, OR WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY CONDITIONS, ENDORSEMENTS, GUARANTEES, REPRESENTATIONS OR WARRANTIES OF DURABILITY, FITNESS FOR A PARTICULAR PURPOSE OR USE, MERCHANTABILITY, MERCHANTABLE QUALITY, NON-INFRINGEMENT, SATISFACTORY QUALITY, OR TITLE, OR ARISING FROM A STATUTE OR CUSTOM OR A COURSE OF DEALING OR USAGE OF TRADE, OR RELATED TO THE DOCUMENTATION OR ITS USE, OR PERFORMANCE OR NON-PERFORMANCE OF ANY SOFTWARE, HARDWARE, SERVICE, OR ANY THIRD PARTY PRODUCTS AND SERVICES REFERENCED HEREIN, ARE HEREBY EXCLUDED. YOU MAY ALSO HAVE OTHER RIGHTS THAT VARY BY STATE OR PROVINCE. SOME JURISDICTIONS MAY NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES AND CONDITIONS. TO THE EXTENT PERMITTED BY LAW, ANY IMPLIED WARRANTIES OR CONDITIONS RELATING TO THE DOCUMENTATION TO THE EXTENT THEY CANNOT BE EXCLUDED AS SET OUT ABOVE, BUT CAN BE LIMITED, ARE HEREBY LIMITED TO NINETY (90) DAYS FROM THE DATE YOU FIRST ACQUIRED THE DOCUMENTATION OR THE ITEM THAT IS THE SUBJECT OF THE CLAIM.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW IN YOUR JURISDICTION, IN NO EVENT SHALL RIM BE LIABLE FOR ANY TYPE OF DAMAGES RELATED TO THIS DOCUMENTATION OR ITS USE, OR PERFORMANCE OR NON-PERFORMANCE OF ANY SOFTWARE, HARDWARE, SERVICE, OR ANY THIRD PARTY PRODUCTS AND SERVICES REFERENCED HEREIN INCLUDING WITHOUT LIMITATION ANY OF THE FOLLOWING DAMAGES: DIRECT, CONSEQUENTIAL, EXEMPLARY, INCIDENTAL, INDIRECT, SPECIAL, PUNITIVE, OR AGGRAVATED DAMAGES, DAMAGES FOR LOSS OF PROFITS OR REVENUES, FAILURE TO REALIZE ANY EXPECTED SAVINGS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, LOSS OF BUSINESS OPPORTUNITY, OR CORRUPTION OR LOSS OF DATA, FAILURES TO TRANSMIT OR RECEIVE ANY DATA, PROBLEMS ASSOCIATED WITH ANY APPLICATIONS USED IN CONJUNCTION WITH RIM PRODUCTS OR SERVICES, DOWNTIME COSTS, LOSS OF THE USE OF RIM PRODUCTS OR SERVICES OR ANY PORTION THEREOF OR OF ANY AIRTIME SERVICES, COST OF SUBSTITUTE GOODS, COSTS OF COVER, FACILITIES OR SERVICES, COST OF CAPITAL, OR OTHER SIMILAR PECUNIARY LOSSES, WHETHER OR NOT SUCH DAMAGES WERE FORESEEN OR UNFORESEEN, AND EVEN IF RIM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW IN YOUR JURISDICTION, RIM SHALL HAVE NO OTHER OBLIGATION, DUTY, OR LIABILITY WHATSOEVER IN CONTRACT, TORT, OR OTHERWISE TO YOU INCLUDING ANY LIABILITY FOR NEGLIGENCE OR STRICT LIABILITY.

THE LIMITATIONS, EXCLUSIONS, AND DISCLAIMERS HEREIN SHALL APPLY: (A) IRRESPECTIVE OF THE NATURE OF THE CAUSE OF ACTION, DEMAND, OR ACTION BY YOU INCLUDING BUT NOT LIMITED TO BREACH OF CONTRACT, NEGLIGENCE, TORT, STRICT LIABILITY OR ANY OTHER LEGAL THEORY AND SHALL SURVIVE A FUNDAMENTAL BREACH OR BREACHES OR THE FAILURE OF THE ESSENTIAL PURPOSE OF THIS AGREEMENT OR OF ANY REMEDY CONTAINED HEREIN; AND (B) TO RIM AND ITS AFFILIATED COMPANIES, THEIR SUCCESSORS, ASSIGNS, AGENTS, SUPPLIERS (INCLUDING AIRTIME SERVICE PROVIDERS), AUTHORIZED RIM DISTRIBUTORS (ALSO INCLUDING AIRTIME SERVICE PROVIDERS) AND THEIR RESPECTIVE DIRECTORS, EMPLOYEES, AND INDEPENDENT CONTRACTORS.

IN ADDITION TO THE LIMITATIONS AND EXCLUSIONS SET OUT ABOVE, IN NO EVENT SHALL ANY DIRECTOR, EMPLOYEE, AGENT, DISTRIBUTOR, SUPPLIER, INDEPENDENT CONTRACTOR OF RIM OR ANY AFFILIATES OF RIM HAVE ANY LIABILITY ARISING FROM OR RELATED TO THE DOCUMENTATION.

Prior to subscribing for, installing, or using any Third Party Products and Services, it is your responsibility to ensure that your airtime service provider has agreed to support all of their features. Some airtime service providers might not offer Internet browsing functionality with a subscription to the BlackBerry® Internet Service. Check with your service provider for availability, roaming arrangements, service plans and features. Installation or use of Third Party Products and Services with RIM's products and services may require one or more patent, trademark, copyright, or other licenses in order to avoid infringement or violation of third party rights. You are solely responsible for determining whether to use Third Party Products and Services and if any third party licenses are required to do so. If required you are responsible for acquiring them. You should not install or use Third Party Products and Services until all necessary licenses have been acquired. Any Third Party Products and Services that are provided with RIM's products and services are provided as a convenience to you and are provided "AS IS" with no express or implied conditions, endorsements, guarantees, representations, or warranties of any kind by RIM and RIM assumes no liability whatsoever, in relation thereto. Your use of Third Party Products and Services shall be governed by and subject to you agreeing to the terms of separate licenses and other agreements applicable thereto with third parties, except to the extent expressly covered by a license or other agreement with RIM.

Certain features outlined in this documentation require a minimum version of BlackBerry® Enterprise Server, BlackBerry® Desktop Software, and/or BlackBerry® Device Software.

The terms of use of any RIM product or service are set out in a separate license or other agreement with RIM applicable thereto. NOTHING IN THIS DOCUMENTATION IS INTENDED TO SUPERSEDE ANY EXPRESS WRITTEN AGREEMENTS OR WARRANTIES PROVIDED BY RIM FOR PORTIONS OF ANY RIM PRODUCT OR SERVICE OTHER THAN THIS DOCUMENTATION.

Research In Motion Limited
295 Phillip Street
Waterloo, ON N2L 3W8
Canada

Research In Motion UK Limited
Centrum House
36 Station Road
Egham, Surrey TW20 9LF
United Kingdom

Published in Canada