



# Contact Service API

## Reference

December 2022



# Contents

- What is the BEMS Contact Service API?..... 4**
- Request, create, and update format..... 5**
- Requesting contact list information..... 6**
  - Retrieving contact list information properties..... 6
  - Request contacts..... 8
  - Request contacts using an email address and preset properties..... 10
  - Request contact list information using a specific property..... 11
- Creating and requesting folder and subfolder information..... 12**
  - Creating and updating folder parameters..... 12
  - Create a folder or subfolder..... 12
  - Request all folders and subfolders under the contacts folder..... 13
- Adding contact information..... 14**
  - Properties used when creating a contact..... 14
  - Create a contact using a specific property..... 15
  - Update a contact's list information..... 16
- Appendix: Configure the UserShape properties..... 17**
- Legal notice..... 18**

# What is the BEMS Contact Service API?

The Contact Service API allows third-party BlackBerry Dynamics apps to query, retrieve, create, and update contact information from a user's contact folder and any folders and subfolders the user creates in the contacts folder in their mailbox. For example, you can use the API to do the following:

- Create new contacts
- Create folders and subfolders under the contacts folder
- Retrieve the complete list of contacts from the contacts folder
- Retrieve contacts from a specified folder or subfolder
- Retrieve a single contact
- Retrieve the new and modified contacts since a specified date
- Update existing contact information

# Request, create, and update format

You must specify the BEMS endpoint in the API. The endpoint specifies where the object address is located.

```
Endpoint: <BEMS FQDN>:8443/api/contact
```

The format for the HTTP Request to retrieve contact information from a mailbox in BEMS is:

```
POST <BEMS FQDN>:8443/api/contact  
<request-header>  
<request-body>
```

The format for the HTTP Request to create a contact in a mailbox in BEMS is:

```
POST <BEMS FQDN>:8443/api/contact/create  
<request-header>  
<request-body>
```

The format for the HTTP Request to update a contact in a mailbox in BEMS is:

```
POST <BEMS FQDN>:8443/api/contact/update  
<request-header>  
<request-body>
```

The format for the HTTP Request to create additional folders and subfolders under the contacts folder in a mailbox in BEMS is:

```
POST <BEMS FQDN>:8443/api/folder/create  
<request-header>  
<request-body>
```

The format for the HTTP Request to get all folders and subfolders under the contacts folder in a mailbox in BEMS is:

```
POST <BEMS FQDN>:8443/api/folder/get  
<request-header>  
<request-body>
```

The following is a sample header:

```
Content-Type: application/json  
X-Good-GD-AuthToken: <GDAuthToken>
```

# Requesting contact list information

Third-party BlackBerry Dynamics apps can retrieve the contact information that was added within a period that you specify, a specified contact, or the contact list from a user's contact folder in their mailbox.

## Retrieving contact list information properties

The following table describes the request body properties that you can include in the JSON formatted request when you retrieve contact list information from a user's folder in the mailbox.

Parameter	Type	Description
Account	String	This parameter specifies the user's email account that is used to request contact information (for example, jamie01@ex365.example.com).
ByName	String	<p>This parameter specifies to search the user's local contacts based on a specific name.</p> <p>The search results include all contacts with a set of specified characters. For example, "ByName": "Jane" will return all users that have Jane as their first name, last name, or a part of their name.</p> <p>When you use this parameter, you can also include the UserShape properties "Basic" or "Detail" to retrieve additional information for the returned contacts. See the UserShape parameter below for more information.</p>
ByEmail	String	<p>This parameter specifies to search the user's local contact list with a specific email address.</p> <p>When you use this parameter, you can also include the UserShape parameter properties "Basic" or "Detail" to retrieve additional information for the returned contacts. See the UserShape parameter below for more information.</p>
FolderId	String	This parameter retrieves contacts from the specified FolderId. This is optional. If the FolderId is not specified, BEMS retrieves the contacts from the user's contact folder.
MaxNumber	Integer	This parameter specifies the maximum number of contacts or objects to return in the search query. By default, BEMS can only return up to 512 items at a time. The client must make multiple calls to retrieve more than 512 items by setting the "Offset" parameter. The "MoreAvailable" value tells the client if more items are available. See the API response table below for more information.
Offset	Integer	This parameter specifies the starting point of a batch response. By default, the Offset is 0 (zero).

Parameter	Type	Description
SinceTs	Integer (long)	<p>This parameter specifies the new or modified contacts in the user's personal contact list since a specified time. SinceTs is specified in epoch time format. If you want to retrieve the new and modified contacts, you must specify the SinceTs to begin searching for contacts.</p> <p>If the SinceTs is not specified, BEMS retrieves all of the contacts from the user's contact folder.</p> <p>When you use this parameter, you can also include the UserShape parameter properties "Basic" or "Detail" to retrieve additional information for the returned contacts. See the UserShape parameter below for more information.</p>
UserShape	String array	<p>This parameter specifies the list of attributes to return in the search results (for example, Basic, MobilePhone, JobTitle, Photo).</p> <p>UserShape supports a preset list of common properties list: Basic and Detail. The Basic and Detail property name lists can be configured in BEMS.</p> <ul style="list-style-type: none"> <li>• Basic: By default, this property returns the following list of properties: LastName, DisplayName, EmailAddress, and PhoneNumbers.</li> <li>• Detail: By default, this property returns the following list of properties in addition to the Basic properties: PhysicalAddresses, CompanyName, JobTitle, Department, and Photo.</li> </ul> <p>For instructions on how administrators can view or configure the UserShape properties, see <a href="#">Appendix: Configure the UserShape properties</a>.</p>

## API response

The following table describes the response properties that might appear in the JSON formatted API response when you retrieve contacts from the user's local contact list.

Property	Type	Description
MoreAvailable	Boolean	<p>This parameter indicates that more contacts are available than the response returned.</p> <p>If MoreAvailable is true, then the client continues to call the API changing the "Offset" value to the value received in the previous response. The client performs this call until the MoreAvailable value is false, which indicates that there are no more contacts to be returned.</p>
TotalCount	Integer	This parameter specifies the total number of contacts that match the fetch query.
NextPageOffset	Integer or null	This parameter specifies the starting point of the second batch of contacts that are returned.

Property	Type	Description
Size	Integer	This parameter specifies the number of contacts returned in the response, up to the MaxNumber size specified.
Offset	Integer	This parameter specifies the starting point of a batch response.
Collection	List of MAP	This parameter specifies a list of contacts returned in the request.

## Request contacts

You can retrieve contacts from a user's main contact folder and from subfolders that the user has created. Deleted contacts are not returned. If the FolderId is not provided by the client in the request, contacts are retrieved from the main folder.

In the following sample, BEMS retrieves all the contacts for the user, Jamie01, excluding any deleted contacts, from the specific subfolder. The first response from BEMS includes up to 100 contacts, as specified by MaxNumber. Each contact that is retrieved includes the default Basic properties that are specified in BEMS. If more than 100 contacts are available (for example, this mailbox includes a TotalCount of 150 contacts) the response includes a MoreAvailable is true, so the client application sends additional requests using the NextOffset value to retrieve contacts in batches until MoreAvailable is false. In this example, the Basic properties returns the following information for the contacts:

- Display name
- Email address
- Given name
- Surname

In the following example, the client provides a FolderId and BEMS retrieves the contacts from the specific folder.

```
POST https://localhost:8443/api/contact
HTTP/1.1Content-Type: application/json
X-Good-GD-AuthToken: <GDAuthToken>
{
  "Account": "jamie01@ex365.example.com",
  "MaxNumber": 100,
  "Offset": 0,
  "UserShape": ["Basic"],
  "FolderId": "AQMkADk0YzU4MjdkLWU1YjktNGE0OC1iZDczKI0Kuc7BmRwA..." (optional)
}
```

If the request is successful, BEMS returns the first 100 contacts that meet the query criteria. BEMS also returns a TotalCount of contacts and NextPageOffset to retrieve the next batch of contacts.

```
{
  "LookupInfo":
  {
    "MoreAvailable": true,
    "TotalCount": 150,
    "NextPageOffset": 100,
    "Size": 100,
    "Offset": 0
  },
  "Contacts": [
  {
```



```

    "DisplayName": "user1_02",
    "EmailAddress": "user1_02@example.com",
    "GivenName": "user1",
    "Surname": "02",
    "UniqueId" : "AAMkADg2Y2IxMzAwLWJmZmYtAAMkADg2Y..."
  },
  {
    "DisplayName": "user2_03",
    "EmailAddress": "user2_03@example.com",
    "GivenName": "user2",
    "Surname": "03",
    "UniqueId" : "AAMkADg2Y2IxMzAwLWJmZmYtNGJjMiliN..."
  },
  ...
]
}

```

The client sets the Offset to NextPageOffset from the previous query to receive the next batch.

```

POST https://localhost:8443/api/contact HTTP/1.1
Content-Type: application/json
X-Good-GD-AuthToken: <GDAuthToken>
{
  "Account": "jamie01@ex365.example.com",
  "MaxNumber": 100,
  "Offset": 100,
  "UserShape": ["Basic"],
  "FolderId": "AQMkADk0YzU4MjdkLWU1YjktNGE0OC1iZDczKI0Kuc7BmRwA..." (optional)
}

```

BEMS returns the next 50 contacts for a total of 150 contacts. No additional contacts are available.

```

{
  "LookupInfo":
  {
    "MoreAvailable": false,
    "TotalCount": 150,
    "NextPageOffset": null,
    "Size": 50,
    "Offset": 100
  },
  "Contacts": [
    {
      "DisplayName": "user55_05",
      "EmailAddress": "user55@example.com",
      "GivenName": "user55",
      "Surname": "05",
      "UniqueId" : "tNGJjMiliNAAMkADg2Y2IxMzAwLWJmZmY..."
    },
    {
      "DisplayName": "user56_06",
      "EmailAddress": "user56@example.com",
      "GivenName": "user56",
      "Surname": "06",
      "UniqueId" : "AAMkADg2Y2IxMmZmYtNGJjMiliNzAwLWJ..."
    },
    ...
  ]
}

```

## Request contacts using an email address and preset properties

You can retrieve contacts from a user's contact folder, or folders and subfolders that the user created in the mailbox using multiple properties (for example, retrieve a user based on their email address and include the preset Detail properties for the contact). In this example, the response includes one contact and `MoreAvailable` is false. If more than 512 contacts are identified, the response indicates that `MoreAvailable` is true, and the client sends additional requests to retrieve contacts in batches until `MoreAvailable` is false. If the client provides a `FolderId`, BEMS retrieves the contacts from the specific folder.

```
POST https://localhost:8443/api/contact HTTP/1.1
Content-Type: application/json
X-Good-GD-AuthToken: <GDAuthToken>
{
  "Account": "jamie01@ex365.example.com",
  "ByEmail": "jane_doe@example.com",
  "MaxNumber": 512,
  "Offset": 0,
  "UserShape": ["Details"],
  "FolderId": "AQMkADk0YzU4MjdkLWU1YjktNGE0OC1iZDczKI0Kuc7BmRwA..." (optional)
}
```

If the request is successful, BEMS returns the following response, and the third-party BlackBerry Dynamics apps display the following information for contacts with the email address `jane_doe@example.com`. If a property is not available, BEMS returns a null value and the information is not included in the response. In this example, the following information is displayed for Jane Doe:

- `DisplayName`
- `GivenName`
- `Surname`
- `FullName`
- `EmailAddress`

```
{
  "LookupInfo": {
    "MoreAvailable": false,
    "TotalCount": 1,
    "NextPageOffset": null,
    "Size": 1,
    "Offset": 0
  },
  "Contacts": [{
    "CompleteName": {
      "Title": null,
      "GivenName": "Jane",
      "Surname": Doe,
      "FullName": "Jane Doe",
      "Nickname": null,
      "YomiGivenName": null,
      "YomiSurname": null
    },
    "DisplayName": "Jane Doe",
    "EmailAddress": "jane_doe@example.com",
    "GivenName": "Jane",
    "UniqueId": "AAMkADg2Y2IxMzAwLWJmZmYtNGJjMiliN..."
  }],
}
```

```
}
```

## Request contact list information using a specific property

You can request the user's contact information for contacts that return specific properties (for example, only the first name of the contacts). In the following sample code, BEMS requests the first name of all the contacts in the user's contact folder. The response includes up to 50 contacts. If the client provides a `FolderId`, BEMS requests the contacts from the specific folder.

```
POST https://localhost:8443/api/contact HTTP/1.1
Content-Type: application/json
X-Good-GD-AuthToken: <GDAuthToken>
{
  "Account": "jamie01@ex365.example.com",
  "MaxNumber": 50,
  "Offset": 0,
  "UserShape": ["FirstName"],
  "FolderId": "AQMkADk0YzU4MjdkLWU1YjktNGE0OC1iZDczKI0Kuc7BmRwA..." (optional)
}
```

If the request is successful, BEMS returns the following response, and the third-party BlackBerry Dynamics apps display the first name of the contacts.

```
"LookupInfo": {
  "MoreAvailable": false,
  "TotalCount": 3,
  "NextPageOffset": null,
  "Size": 3,
  "Offset": 0
},
"Contacts": [
  { "GivenName": "Jane",
    "UniqueId": "g2Y2IxMzAwLWJmZmYtNGJjMiliNAAMkAD..."
  },
  { "GivenName": "Lori",
    "UniqueId": "AAMkADMzAwLWJmZmYtNGJjMiliNg2Y2Ix..."
  },
  { "GivenName": "Tammy",
    "UniqueId": "jMiliNAAMkADg2Y2IxMzAwLWJmZmYtNGJ..."
  },
]
}
```

# Creating and requesting folder and subfolder information

Third-party BlackBerry Dynamics apps can retrieve folder and subfolder information that the user created in their contact folder. Subfolders can also be created in a specific folder.

## Creating and updating folder parameters

The following table describes the request body properties that you can include in the JSON formatted request when you create a folder or subfolder in the contact folder of a user's mailbox.

Parameter	Type	Description
FolderName	String	This parameter specifies the name of the folder or subfolder that the user created.
ParentFolderId	String	This parameter creates the contact in the specified ParentFolderId. This is optional. If the ParentFolderId is not provided, BEMS creates the contact in the user's contact folder.

## Create a folder or subfolder

You can create folders and subfolders in the user's contact folder. The ParentFolderId is optional. When it is not provided and a folder is created, the folder appears in the user's contact folder. In the following sample code, the folder called "Support folder" is created as a subfolder in the specified ParentFolderId.

```
POST https://localhost:8443/api/folder/create HTTP/1.1
Content-Type: application/json
X-Good-GD-AuthToken: <GDAuthToken>
{
  "FolderName": "Support folder",
  "ParentFolderId": "AQMkADk0YzU4MjdkLWU1YjktNGE0OC1iZDczLWJjOTeyYjE2M..."
}
```

If the request is successful, BEMS returns a 201 HTTP response code that the contact folder is successfully created.

If a folder with the same name exists in the parent folder, BEMS returns a 200 HTTP response code and the folder is not saved.

```
{
  "ErrorMessage": "Support folder" already exists in this directory"
}
```

## Request all folders and subfolders under the contacts folder

You can retrieve all folders and subfolders that the user created in a user's contact folder. In the following sample code, BEMS retrieves all folders that the user created.

```
POST : https://localhost:8443/api/folder/get HTTP/1.1
Content-Type: application/json
X-Good-GD-AuthToken: <GDAuthToken>
{
  "MaxNumber": 512,
  "Offset": 0
}
```

If the request is successful, BEMS returns the following response, and the third-party BlackBerry Dynamics apps display the retrieved folder.

```
{
  "LookupInfo": {
    "MoreAvailable": false,
    "NextPageOffset": null,
    "TotalCount": 3,
    "Offset": 1
  },
  "FolderId":
  {
    "ChangeKey": "AwAAABYAAAAkmjuXJkcZSLVmVxZy9YrgAAFjhmIy",
    "UniqueId": "AAMkADQyNDc4Mjc1LTMyOTMtNDMzZC1hYmVlLTVm...",
    "FolderName": null,
    "Mailbox": null,
    "XmlElementName": "FolderId",
    "Valid": true,
    "Namespace": "Types"
  },
  "FolderName": "Support folder",
  "ParentFolderId":
  {
    "ChangeKey": "AQAAAA==",
    "UniqueId": "AAMkADQyNDc4Mjc1LTMyOTMtNDMzZC1hYmVlLTVm...", //Folder
    "Support folder" is created under this folderId.
    "FolderName": null,
    "Mailbox": null,
    "XmlElementName": "FolderId",
    "Valid": true,
    "Namespace": "Types"
  }
}
```

# Adding contact information

Third-party BlackBerry Dynamics apps can create and update the contact information in a user's contact folder, or folders and subfolders that the user created in their mailbox.

## Properties used when creating a contact

The following list shows the supported body properties that you can include in the JSON formatted request when you create a contact in a user's folder in the mailbox. Any value which is null or empty in the request body is not saved in the contact.

If the user is updating an existing contact, all values for the request, whether they are changed or not, are submitted to BEMS. The following values can be specified when creating a contact:

- Contact
  - First Name
  - Middle Name
  - Last Name
  - Mobile Phone
  - Home Phone
  - Home Phone2
  - Home Fax
  - Other Fax
  - EmailAddress1
  - EmailAddress2
  - EmailAddress3
  - Business Phone
  - Business Phone2
  - Car Phone
  - CompanyMainPhone
  - ISDN
  - CallBack
  - Radio Phone
  - Primary Phone
  - Assistant Phone
  - Telex
  - TtyTddPhone
- Home address
  - Street
  - City
  - State
  - Country
  - Postal code
- Work
  - Company
  - JobTitle
  - Department

- Office
- Manager
- Assistant
- Business address
  - Street
  - City
  - State
  - Country
  - Postal code

## Create a contact using a specific property

You can create a contact using specific properties. In the following sample code, BEMS creates the contact in the user's contact folder. If a ParentFolderId is included, the contact is created in the specified folder.

In this example, the user creates a contact using the following information for the contact:

- First name
- Last name
- Middle name
- Mobile phone
- Home phone
- Business phone
- Email address
- Company name

```
POST https://localhost:8443/api/contact/create HTTP/1.1
Content-Type: application/json
X-Good-GD-AuthToken: <GDAuthToken>
{
  "FirstName": "NewContact",
  "LastName": "Last",
  "MiddleName": "Middle",
  "MiddleName": "Middle",
  "MobilePhone": "519-555-1234",
  "HomePhone": "519-555-4567",
  "BusinessPhone": "519-555-7890",
  "EmailAddress1": "NewContact.Last@blackberry.com",
  "EmailAddress2": "NewContact.Last2@blackberry.com",
  "EmailAddress3": "",
  "ParentFolderId": "AQMkADk0YzU4MjdkLWU1YjktNGE0OC1iZDczLWJjTVXVDG....",
  (optional)
  "Work": {
    "JobTitle": "",
    "Department": "",
    "Company": "BlackBerry",
    "Office": "",
    "Manager": "",
    "Assistant": ""
  }
}
```

If the contact is created successfully, BEMS returns a `uniqueid` and the third-party BlackBerry Dynamics apps display the provided contact information. If a property is not specified, BEMS returns a null value and the information is not saved in the contact.

```
{
  "UniqueId" : "Dg2Y2IxAAMkAMzAwLWJmZmYtNGJjMiliN..."
}
```

## Update a contact's list information

You can update information for a contact using specific properties. In the following sample code, BEMS updates the contact information for `NewContact Last` in the user's contact folder. The client sends the `UniqueId` for the contact to update. When a contact is updated, the client sends all of the values for the contact to BEMS, whether the values were modified or not. If the client provides a `ParentFolderID`, BEMS updates the contact in the specified folder.

In this example, the contact is updated with their work information. The new and existing information is sent to BEMS for updating.

- Job title
- Department
- Middle name
- Mobile phone
- Home phone
- Business phone
- Email address
- Company name

```
POST https://localhost:8443/api/contact/update HTTP/1.1
Content-Type: application/json
X-Good-GD-AuthToken: <GDAuthToken>
{
  "UniqueId": "AAMkADY2ZjrKzY2E2LWY0ZDQtNGU3NilhMjk3LTAwM....",
  "FirstName": "NewContact",
  "LastName": "Last",
  "MiddleName": "Middle",
  "MobilePhone": "519-555-1234",
  "HomePhone": "519-555-4567",
  "BusinessPhone": "519-555-7890",
  "EmailAddress1": "NewContact.Last@blackberry.com",
  "EmailAddress2": "NewContact.Last2@blackberry.com",
  "Work": {
    "JobTitle": "Manager",
    "Department": "Development",
    "Company": "BlackBerry",
    "Office": "Home"
  },
  "FolderId": "AQMkADk0YzU4MjdkLWU1YjktNGE0OC1izDczKI0Kuc7BmRwA..." (optional)
}
```

If the request is successful, BEMS returns a 200 HTTP response code that the contact is successfully updated.



# Appendix: Configure the UserShape properties



**CAUTION:** Don't modify the UserShape properties unless the changes are required. Modified settings are not retained when you upgrade the BEMS software.

The following values can be specified for the UserShape properties. Other values are ignored.

- Alias
- EmailAddress
- DisplayName
- GivenName
- FirstName
- Surname
- LastName
- CompleteName
- CompanyName
- Company
- Department
- JobTitle
- Title
- Photo
- UserCertificate
- UsersmimeCertificate
- PrUserx509Certificate
- HomePhone
- HomePhone2
- MobilePhone
- Pager
- BusinessPhone
- BusinessFax
- OtherTelephone
- PhoneNumbers
- PhysicalAddresses
- Manager
- DirectReports

1. Open a browser and go to the Apache Karaf Web Console Configuration website located at [https://<fqdn\\_of\\_the\\_bems\\_host>:8443/system/console/configMgr](https://<fqdn_of_the_bems_host>:8443/system/console/configMgr) and log in as administrator with the appropriate Microsoft Active Directory credentials.
2. On the menu, click **OSGi > Configuration**.
3. Search for and click **Directory Lookup Common Configuration**.
4. In the **basicPropertyNames** field, the basic property values are listed.  
Click the + button and the property name to add a common property to the list or click the - button to remove a common property from the list.
5. In the **detailedPropertyNames** field, the detailed property value names of common property names is listed.  
Click the + button and the property to add a common property to the list or click the - button to remove a common property from the list.
6. Click **Save**.

# Legal notice

©2023 BlackBerry Limited. Trademarks, including but not limited to BLACKBERRY, BBM, BES, EMBLEM Design, ATHOC, CYLANCE and SECUSMART are the trademarks or registered trademarks of BlackBerry Limited, its subsidiaries and/or affiliates, used under license, and the exclusive rights to such trademarks are expressly reserved. All other trademarks are the property of their respective owners.

Patents, as applicable, identified at: [www.blackberry.com/patents](http://www.blackberry.com/patents).

This documentation including all documentation incorporated by reference herein such as documentation provided or made available on the BlackBerry website provided or made accessible "AS IS" and "AS AVAILABLE" and without condition, endorsement, guarantee, representation, or warranty of any kind by BlackBerry Limited and its affiliated companies ("BlackBerry") and BlackBerry assumes no responsibility for any typographical, technical, or other inaccuracies, errors, or omissions in this documentation. In order to protect BlackBerry proprietary and confidential information and/or trade secrets, this documentation may describe some aspects of BlackBerry technology in generalized terms. BlackBerry reserves the right to periodically change information that is contained in this documentation; however, BlackBerry 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 websites (collectively the "Third Party Products and Services"). BlackBerry 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 BlackBerry 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 BLACKBERRY 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 BLACKBERRY PRODUCTS OR SERVICES, DOWNTIME COSTS, LOSS OF THE USE OF BLACKBERRY 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 BLACKBERRY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW IN YOUR JURISDICTION, BLACKBERRY 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 BLACKBERRY AND ITS AFFILIATED COMPANIES, THEIR SUCCESSORS, ASSIGNS, AGENTS, SUPPLIERS (INCLUDING AIRTIME SERVICE PROVIDERS), AUTHORIZED BLACKBERRY 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 BLACKBERRY OR ANY AFFILIATES OF BLACKBERRY 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 BlackBerry'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 BlackBerry'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 BlackBerry and BlackBerry 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 BlackBerry.

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

BlackBerry Enterprise Software incorporates certain third-party software. The license and copyright information associated with this software is available at <http://worldwide.blackberry.com/legal/thirdpartysoftware.jsp>.

BlackBerry Limited  
2200 University Avenue East  
Waterloo, Ontario  
Canada N2K 0A7

BlackBerry UK Limited  
Ground Floor, The Pearce Building, West Street,  
Maidenhead, Berkshire SL6 1RL  
United Kingdom

Published in Canada