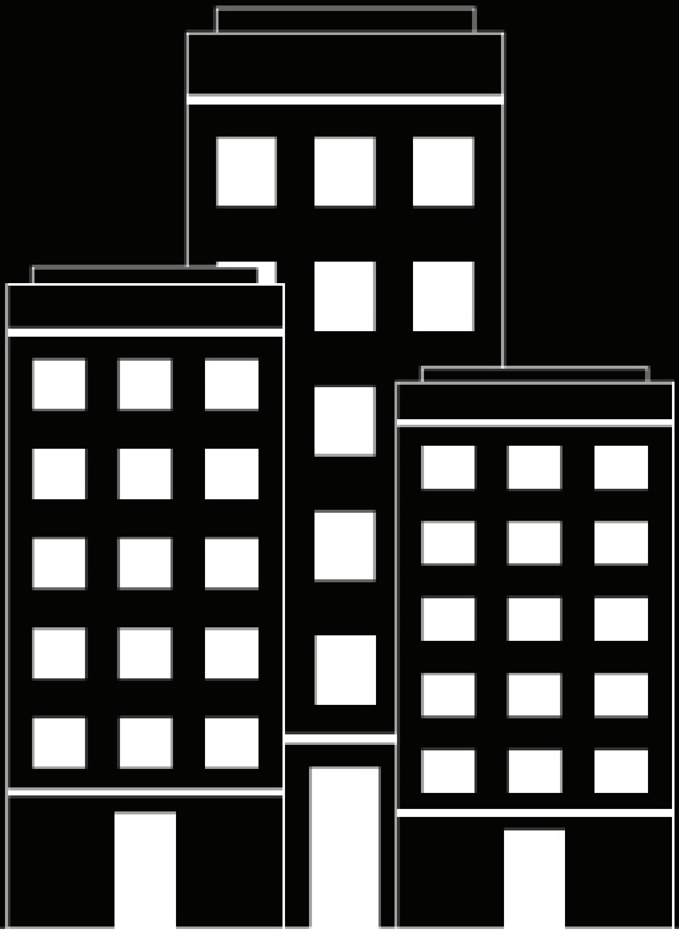

Docs REST API

Reference

March 2021

Contents

What is the Docs REST API?

The Docs REST API uses REST calls to support:

- Document storage in an organization's existing repositories
- Authorized users access to manage their user-defined repositories using the Docs Self-Service web console

REST API

BEMS supports multiple functions for storing data in an organization's existing repositories. This section describes the document, repository, document version, miscellaneous, Microsoft Office Web Apps, generate shared link, and Docs Self-Service web console functions that are supported.

HTTP request headers

The following table lists the request headers used in the Docs REST API.

HTTP Header	Value	Required
One of the following: • X-Good-GD-AuthToken • X-Good-WD-AuthToken	One of the following: • GD AuthToken retrieved from the BlackBerry Dynamics library on device • WD AuthToken retrieved from the BlackBerry Workspaces library	√ One is required depending on which is used for user validation
Authorization	One of the following: • HTTP Basic authentication format For example, BASIC <*username*>:<*password*> in base64 encoding • HTTP Bearer authentication format (bearer token)	√ Required to access enterprise repositories when Kerberos constrained delegation (KCD) is not enabled
Authorization* (Docs Self-Service web console only)	One of the following in base64 encoding: • Windows authentication using Basic credentials • Windows authentication using Negotiate credentials	√ One is required
Content-Type*	application/json	√ Required only when sending a POST request that contains a JSON body
X-GOOD-RMS-ACCESS-TOKEN	ADAL OAuth2 access token	√ Required when: • Retrieving the metadata for the Azure - IP protected document • Downloading an Azure - IP protected document

* The API for the Docs Self-service web console requires only the Authorization and Content-Type headers.

REST call URI values

The following table describes the values that can be included in the REST Call URI.

Variable	Value	Description
version	1	This setting specifies the document version.
repository	datasourceID	This setting specifies the enterprise managed data location, where *datasourceID* is the data source ID.
	datasourceName	This setting specifies the enterprise managed data location, where *datasourceName* is the data source name.
	user/userDatasourceID	This setting specifies the user defined data location, where *userDatasourceID* is the user defined data source ID.
	user/ userDatasourceName	This setting specifies the user defined data location, where *userDatasourceID* is the user defined data source ID.
filename-with-path		This setting specifies any valid subpath. For security purposes, paths with a period (.) or multiple periods (..) for folders are identified and disallowed and return an error message.
folder-path		This setting specifies the folder path. If the folder path is empty, it indicates that the folder is located at the drive root.
repositoryId		This setting specifies the ID of the repository for a specific user.
email		This setting specifies the email address of the specific user of the repository.

REST functions

The following REST functions are supported:

- Document-related functions
- Repository-related functions
- Document version-related functions
- Miscellaneous functions
- Microsoft Office Web Apps support functions
- Generate Share Link support functions
- Docs Self-Service web console functions

Document-related functions

Function	REST Call URI	Verb	Description
Download document	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/{*repository*}/{*filename-with-path*}	GET	This function downloads the document content to the client. The content type is not sent to the client. Supported output HTTP headers: • ETag: This value provides the unique identifier for the current version of the data (file). • Last-Modified: This value specifies the system last modified time for a file. • Accept-Ranges: The server informs the client that it supports the range requests. • Content-Length: The server returns a byte range to the client when responding to a range request. For chunked and resumable downloads, the following HTTP headers are optional: • Input HTTP request headers: • Range: This value specifies the range of bytes for the client to download. • If-Range: This value tells the server to check if the content has changed from the initial download when a downloaded is resumed. If it changed, the entire document is returned. If it hasn't changed, only the requested section of the document is returned. The ETag must match the Range. • Output HTTP response headers: • Content-Range: This value specifies the byte range returned. Optional parameters: • storagePath* • unprotected=true: This parameter specifies if unprotected contents of an RMS protected document are sent to the client.

Function	REST Call URI	Verb	Description
List documents	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/{*repository*}/{*folder-path*}	GET	This function lists the documents in the specified folder. Optional parameters: • recurse: This parameter specifies whether to include the folders and subfolders. • maxitems: This parameter specifies a value for the maximum number of items to output. By default, when maxitems is not specified it is 10,000. • view: This parameter specifies the SharePoint view name used when the library or folder contents are retrieved. • sort: This parameter specifies whether to sort the list output. If the sort parameter is added, use the following format: "sort=*field-name direction*". • The *field-name* can be name, modified, or size. • The *direction* can be asc or desc. By default, when direction is not specified, it is asc. • storagePath** • For chunked listing such as paging, the following parameters can be used: • startIndex: This parameter specifies the starting index of items. • endIndex: This parameter specifies the ending index of items. • refresh: This parameter specifies to delete the cached content when contents are cached. • Output HTTP response header: • X-Good-Content-Range: items startIndex-endIndex/numberOfItems • filter: This parameter specifies to filter the list output to only display files or folders. If adding the filter parameter, use the format: "filter=file|folder". By default, the output is not filtered and both files and folders are displayed. • format: This parameter specifies to format the list output to flat or nested. If you add the format parameter, use the format: "format=flat|nested". By default the format is nested. To use the format option, you must specify recurse=true. If recurse=true and the format is specified as flat, then the "Folder Contents" JSON key in the folders are removed and a "path" JSON key value is added for each file and folder. All of the nested files and folders are displayed at the same level. **Note:** Paging and recursion cannot be specified in the same request if the format is specified as nested. For an example of a list documents output, see File shares folder listing and metadata output and SharePoint folder listing and metadata output.

Function	REST Call URI	Verb	Description
Upload document	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/{*repository*}/{*filename--with#path*}	PUT	This function uploads the document to the Share or SharePoint server. The document contents are sent as the JSON body. • If the file already exists, it is replaced. • If the path does not fully exist, the required folders are created. Optional parameters: • storagePath*
Create Folder	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/{*repository*}/{*folder-path*}/	PUT	This function creates a folder. If a trailing slash is added to the end of the file path, the function automatically creates any intermediate folders if they do not exist. Optional parameters: • storagePath**
Delete document	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/{*repository*}/{*filename-with#path*}	DELETE	This function deletes the specified document. Optional parameters: • storagePath*
Delete folder	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/{*repository*}/{*folder#path*}?recurse	DELET	This function deletes the specified folder if the folder contents are empty. Optional parameters: • recurse: This is required to delete a folder and its content.
Retrieve document metadata	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/{*repository*}/{*filename-with-path*}?metadata	GET	This function returns document metadata. Optional parameters: • storagePath** • includeRMSPermissions=true: This parameter specifies whether RMS permissions information is included when the document is Rights Management Services (RMS) protected.
Retrieve folder metadata	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/{*repository*}/{*folder-path*}?metadata	GET	This function returns folder metadata. If the folder has views associated with them, then these are returned. Optional parameters: • storagePath** For an example of retrieved folder metadata output, see SharePoint library metadata retrieval.

Function	REST Call URI	Verb	Description
Parse storage path	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}?parse&path={*storage-path*}	GET	This function parses the storage path. Optional parameters: • storage: This parameter specifies the underlying storage (for example, FileShare or SharePoint).

Move File	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/{*repository*}/{*filename-with-path*}?action=move	POST	This function moves a file to a new location.
			The required request body is JSON, as shown in the following example:
			`{`
			`"targetPath":"target-path",`
			`"overwrite":"true" Or "false",`
			`"keepOriginal":"true" Or "false"`
			`}`
			• The *targetPath* (string) value is the path relative to repository path where the file needs to be moved to. This is a required parameter.
			• The *overwrite* (boolean) value is true if the existing file at targetPath can be over written. This is an optional parameter. By default, this value is false.
			• The *keepOriginal* (boolean) value is true if the file is copied to the targetPath and false if the file is to move to the targetPath. This is an optional parameter. By default, this value is false

Move Folder	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/{*repository*}/{*foldername-with-path*}?action=move	POST	This function moves a folder to a new location.
			The required request body is JSON, as shown in the following example:
			`{`
			`"targetPath":"target-path",`
			`"overwrite":"true" Or "false",`
			`"keepOriginal":"true" Or "false"`
			`}`
			• The *targetPath* (string) value is the path relative to repository path where the folder needs to moved to. This is a required parameter.
			• The *overwrite* (boolean) value is true if the existing folder at targetPath can be over written. This is an optional parameter. By default, this value is false.
			• The *keepOriginal* (boolean) value is true if the folder is copied to the targetPath and false if folder is moved to the targetPath. This is an optional parameter. By default, this value is false.

Function	REST Call URI	Verb	Description

Protect document	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/{*repository*}/{*filename-with-path*}?action=protect	POST	This function protects a document.
			The required request body is JSON, as shown in the following example:
			`{`
			` "templateId":"templateID",`
			` "expiryDate":"expiryDate"`
			`}`
			• The *templateID* (string) value is the template ID required to protect the document. This is a required parameter.
			• The *expiryDate* (date) value is the date when protection expires. This is a required parameter and you must provide the date and time with the standard time zone or time offset. Use one of the following formats,
			  • "2020:02:25T10:25:00.000-05:00", where 05:00 is the time offset
			  • "2020-02-25T10:25:00.000Z", where Z is UTC time

Unprotect document	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/{*repository*}/{*filename-with-path*}?action=unprotect	POST	This function unprotects a document in the repository.

* Required for documents in flat view and Shared With Me Microsoft SharePoint libraries and specifies the actual path of document

** Required for Shared With Me SharePoint folders and specifies the actual path of the folder

Repository-related functions

Function	REST Call URI	Verb	Description
List repositories	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}?include=offline	GET	This function lists the repositories. Optional parameters • include=offline: This parameter specifies that offline information is included in the output and must be in the request uri. For an example of a repository list response format, see Repository listing response format.

Function	REST Call URI	Verb	Description
Retrieve repository metadata	http(s)://{*FQDN of the BEMS-Docs server*}:8443/ docs/{*version*}/{*repository*}? metadata=repository? include=offline	GET	This function retrieves the repository metadata. Optional parameters • include=offline: This parameter specifies that offline information is included in the output and must be in the request uri.

Add/ edit user defined repository	http(s)://{*FQDN of the BEMS-Docs server*}:8443/ docs/{*version*}? action=addUserRepository	POST	This function allows adding and editing of the user defined repository.
			The required request body is in the JSON format as shown in the following example:
			`{`
			` "name":"name",`
			` "path":"path",`
			` "storage":"storage"`
			` }`
			• The *name* value is the user's display name.
			• The *path* value is the path to the data source.
			• The *storage* value is any supported physical storage. For example, FileShare or Microsoft SharePoint.
			The output is the metadata of the repository added in the JSON format.

Delete user defined repository	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/ {*version*}/user/{*repository*}	DELETE	This function deletes the user-defined repository.

Document version-related functions

These functions are available where document versions is supported (for example, Microsoft SharePoint).

Function	REST Call URI	Verb	Description
Download a specific version of a document	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{version}/{*repository*}/{*filename-with-path*}?version={*doc#version*}	GET	This function downloads a specific version of a document content to the client. The content type is not sent to the client. The supported output HTTP headers are: • ETag: This value provides the unique identifier for the current version of the data (file). • Last-Modified: This value specifies the system last modified time for a file. • Accept-Ranges: The server informs the client that it supports the range requests. • Content-Length: The server returns a byte range to the client when responding to a range request. For chunked and resumable downloads, the following HTTP headers are optional: • Input HTTP request headers: • Range: Specifies the range of bytes for the client to download • If-Range: ETag to match for range downloads • Output HTTP response headers: • Content-Range: Specifies the byte range returned Optional parameters: • storagePath* • unprotected=true: This parameter specifies if unprotected contents of an RMS protected document are sent to the client.
Retrieve document metadata for a specific version of document	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{version}/{*repository*}/{*filename-with#path*} ? metadata&version={doc–version}	GET	This function retrieves the document metadata for a specific version of a document. Optional parameters: • storagePath* • includeRMSPermissions=true: This parameter specifies if RMS permissions information is included when the document is RMS protected.
Retrieve document metadata for all versions of a document	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/{*repository*}/{*filename-with-path*} ? metadata&version=*	GET	This function retrieves the document metadata for all versions of a document. Optional parameters: • storagePath*

Function	REST Call URI	Verb	Description
Check out a document	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/{*repository*}/{*filename-with-path*} ?action=checkout	POST	This function checks out a document from the FileShare or SharePoint server. The optional request body is JSON, as shown in the following example: `{ "comments":"comments" }` The *comments* value can provide information for the checked out document. Optional parameters: • storagePath*

Check in a document	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/{*repository*}/{*filename-with#path*} ?action=checkin	POST	This function checks a document in to the FileShare or SharePoint server.
			The optional request body is JSON, as shown in the following example:
			`{`
			`  "type":"checkin-type",`
			`  "comments":"comments"`
			`}`
			• The *checkin-type* value is any checkin value supported by the underlying storage. For Microsoft SharePoint, this can be one of the following:
			  • major
			  • minor
			  • overwrite
			By default, when the checkin-type is not specified it is major.
			• A complete JSON body as well as the individual elements (type and comments) are optional.
			Optional parameters:
			• storagePath*

Undo checking out a document	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/{*repository*}/{*filename#with#path*} ?action=cancelcheckout	POST	This function cancels a document checkout. Optional parameters: • storagePath*

Function	REST Call URI	Verb	Description
Restore a document version	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/{*repository*}/{*filename#with#path*} ?action=restoreVersion	POST	This function restores the document version to the previous version number. The version to restore to must be specified in the following JSON format: `{ "version":"doc-version" }` The *doc#version* value is the version number of the document. Optional parameters: • storagePath*

* Required for documents in flat view and Shared With Me Microsoft SharePoint libraries and specifies the actual path of document

** Required for Shared With Me SharePoint folders and specifies the actual path of the folder

Miscellaneous functions

Function	REST Call URI	Verb	Description
Get server configuration	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/config	GET	This function retrieves the server configuration. The *version* value is the version number of the document. For a sample output of a server configuration, see Server configuration output.
Validate a user with the Microsof Active Directory usi the BASIC credentials passed in the Authorizatio header	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/validate-user	GET	This function verifies if the user is valid. You can receive one of the following responses: • 200 OK: This response indicates that the user is validated. • 401 Unauthorized: This response indicates that the user could not be validated.

Microsoft Office Web Apps support functions

Function	REST Call URI	Verb	Description
Get list of document types supported	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/config	GET	This function retrieves a list of supported document types. The configuration output includes the supported extensions and the actions that are supported for each extension. For a sample output of a server configuration listing the document types that are supported, see Server configuration output.
Retrieve URL for the Microsof Office Web Apps docum to view and edit	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/{*repository*}/{*filename-with-path*}?metadata&actionURL={*action*}	GET	This function retrieves a document for viewing and editing. Where *action* is one of the following: • view • edit • pdf The output is a JSON body that includes normal file metadata with the addition of the URL for office document to view, edit, or pdf. • If requesting a URL for viewing, the JSON contains a viewActionURL item. • If requesting a URL for editing, the JSON contains an editActionURL item. • If requesting a URL for a .pdf output, the JSON contains a pdfActionURL item.

Generate Share Link support functions

Function	REST Call URI	Verb	Description

Generate a Shared Link	One of the following:
	- http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/{*repository*}/{*folder-path*}?action=generatesharedlink
	- http(s)://{*FQDN of the BEMS-Docs server*}:8443/docs/{*version*}/{*repository*}/{*file-path*}?action=generatesharedlink
		GET	This function generates a link to a file or folder.
			Input: The Request Body contains the JSON body required for generation of the Shared Link as required by underlying storage. For example, the following sample JSON body is used if you are using Box to generate the shared file and allow a user to preview the document, but not download it:
			`{ "shared_link":`
			`  { "access":"company",`
			` "unshared_at":"2016-02-10T12:15:10-8:00",`
			`    "password1":1234",`
			`    "permissions":`
			`    {`
			`      "can_download":false",`
			`      "can_preview":"true"`
			`    }`
			`   }`
			` }`
			Output: The JSON file contains the generated shared link and the password. The password is optional if it is generated by the underlying storage, as shown in the following example:
			`{`
			` "url":"GeneratedSharedLinkUrlString"`
			`}`
			For more information about Box APIs, visit https://developer.box.com/reference#create-a-shared-link-for-a-file.

Docs Self-Service web console functions

These functions are available when the user-defined repositories are enabled in the environment. User-defined repositories allow users to view, add, update, and remove the user-defined repositories. For sample calls of these functions, see Docs Self-Service web console examples.

Function	REST Call URI	Verb	Description
Get User Repositories	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docsconsole-sso/data/docs/users/{*email*}/repositories	GET	This function searches the repositories that are assigned to a specific user. Optional parameters: • name: This parameter searches the user's repositories using the full or partial name of the repository. For an example of user-defined repositories that are assigned to a user, see Get user-defined repositories retrieval.
Get a User Repository	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docsconsole-sso/data/docs/users/{*email*}/repositories/{*repositoryId*}	GET	This function searches for a specific repository that is assigned to a specific user. For an example of a specific user-defined repository that a user is assigned, see Get specific user-defined repository retrieval.

Create User-Defined Repository	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docsconsole-sso/data/docs/users/{*email*}/repositories	POST	This function creates a user-defined repository.
			The required request body is in the JSON format, as shown in the following example:
			`{`
			`"name":"UserDefinedRepository",`
			`"storagetype":"SharePoint",`
			`"path":"<homeDirectory>",`
			`"validatepath":false`
			`}`
			For an example of creating a user-defined repository output, see Create a user-defined repository output.

Delete User-Defined Repository	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docsconsole-sso/data/docs/users/{*email*}/repositories/{*repositoryId*}	DELETE	This function deletes a specific user-defined repository. For an example of deleting a user-defined repository output, see Delete a user-defined repository output.

Function	REST Call URI	Verb	Description

Update User-Defined Repository	http(s)://{*FQDN of the BEMS-Docs server*}:8443/docsconsole-sso/data/docs/users/{*email*}/repositories/{*repositoryId*}	POST	This function updates a specific user-defined repository.
			The required request body is in the JSON format, as shown in the following example:
			`{`
			`"storagetype" : "SharePoint",`
			`"name":"UserDefinedRepositoryModified",`
			`"path":"http://share1.example.com",`
			`"validatepath":false`
			`}`
			For an example of updating a user-defined repository output, see Update a user-defined repository output.

Docs Self-Service web console examples

The following table provides examples of REST Call URIs for user-defined repositories. In these examples, the FQDN of the server that is running BEMS is bems.example.com. The authorized user that these functions are run against is user1. These function call examples use the Windows authentication using Basic credentials. Windows authentication using Negotiate credentials is also supported. If you use Negotiate credentials, you must create a Service Principal Name to allow BEMS to communicate with the Docs Self-Service web console. For more information, see Create a Service Principal Name to access the Docs Self-Service web console.

Function	REST Call URI example	Description
Get User Repositories	GET https://bems.example.com:8443/docsconsole-sso/data/docs/users/user1@example.com/repositories Authorization: Basic ZXhhbXBsZVx1c2VyMTpPcGVuU2VzYW1l	This example searches for and retrieves all of the repositories that are assigned to user1.
Get a User Repository	GET https://bems.example.com:8443/docsconsole-sso/data/docs/users/user1@example.com/repositories/3451 Authorization: Basic ZXhhbXBsZVx1c2VyMTpPcGVuU2VzYW1l	This example searches for and retrieves a specific repository that includes the repositoryId 3451.

Function	REST Call URI example	Description

Create User-Defined Repository	POST https://bems.example.com:8443/docsconsole-sso/data/docs/users/user1@example.com/repositories
	Authorization: Basic ZXhhbXBsZVx1c2VyMTpPcGVuU2VzYW1l
	HTTP Request body in JSON format:
	`{`
	` "name":"SalesPages",`
	` "storagetype":"SharePoint",`
	` "path":"https://<serverY>/`
	`sales",`
	` "validatepath":false`
	`}`
		This example creates the new repository "SalesPages" for the authorized user.

Delete User-Defined Repository	DELETE https://bems.example.com:8443/docsconsole-sso/data/docs/users/user1@example.com/repositories/3451 Authorization: Basic ZXhhbXBsZVx1c2VyMTpPcGVuU2VzYW1l	This example deletes the repository that includes the repositoryId 3451.

Update User-Defined Repository	POST https://bems.example.com:8443/docsconsole-sso/data/docs/users/user1@example.com/repositories/3451
	Authorization: Basic ZXhhbXBsZVx1c2VyMTpPcGVuU2VzYW1l
	HTTP Request body in a JSON format:
	`{`
	` "name":"Marketing",`
	` "storagetype":"SharePoint",`
	` "path":"https://<serverY>/`
	`marketing",`
	` "validatepath":false`
	`}`
		This example modifies the repository that includes the repositoryId 3451 to the new name "Marketing" and new location on the Microsoft SharePoint server.

Enable cross-origin resource sharing support to BEMS-Docs

You must set the AllowedCorsOrigins parameter in BEMS to allow cross-origin resource sharing (CORS) for Docs Self-service API calls.

1. Sign in to the computer that is running the BEMS-Docs service.
2. In a browser, open the BEMS Karaf Console Configuration web site. Type `https://localhost:8443/system/console/configMgr` and log in as an administrator with the appropriate Microsoft Active Directory credentials.
3. On the menu, click **Main > Gogo**.

4. In the command, type the following to add the CORS origin to the list of origins that can access the BEMS-Docs service: `docs:config AllowedCorsOrigins “https://`*`domain1`*`.com:8080,https://`*`domain2`*`.com:8089"`. Separate the CORS URLs with a comma and no space.
5. Close the browser.

Create a Service Principal Name to access the Docs Self-Service web console

You can use Negotiate credentials to allow authorized users to access the Docs Self-Service web console to view, add, update, and remove the user-defined repositories within a domain and between domains and forests. When you use this authentication method, you must create an SPN to provide the BEMS credentials to the web console.

On the computer that hosts BEMS, open a command prompt and type *setspn -S http/<FQDN of the computer hosting BEMS> <Good Technology Common Services service account>*

For example, `setspn -S http/bems.example.com BEMSAdmin`

Response formats

Permission values

The following table defines the permission values that can be included in the JSON response.

Entry	Definition
A	Apps list
AF	Apps followed
C	Cache (offline files)
CF	Create folder
CI	Check in/Check out
CP	Copy/Paste
D	Delete
DS	Data Sources allowed (1 = File Share, 2 = SharePoint)
E	Allow native email
L	List (browse)
LU	Last time
KS	Keep in sync
O	Open in app
P	Print
R	Read (download)
S	Sync
SL	Shared link
W	Write (upload)

Repository listing response format

The following code sample shows a repository listing response.

```
[
{
 "name": "psn1",
 "definedBy":"Admin",
 "providedBy":"Enterprise",
 "permissions":{
   "A":[
        {
         "ID":"com.quickoffice.quickofficeipad",
         "N": "Quick Office Pro"
        },
        {
          "ID":"com.dataviz.DocsToGoEAS",
          "N":"Docs to Go"
        }
       ],
   "AF":[
         1
        ],
   "DS":[
         1,
         2
        ],
   "C":1,
   "CF":1,
   "CI":1,
   "CP":1,
   "D":1,
   "E":0,
   "KS":0,
   "L":1,
   "LU":0,
   "O":1,
   "P":0,
   "R":1,
   "S":1,
   "W":1
   },
   "id":2,
   "path":"http://psn---1c.share1.example.com/",
   "domain":null,
   "storage":"SharePoint",
   "offline":[]
 },
 {
  "name":"path",
  "definedBy":"Internal",
  "providedBy":"Enterprise",
  "permissions":{
    "A":[
      {
       "ID":"com.quickoffice.quickofficeipad",
       "N":"Quick Office Pro"
      },
      {
       "ID":"com.dataviz.DocsToGoEAS",
       "N":"Docs to Go"
      }
     ],
   "AF":[
         1
        ],
    "DS":[
          1,
          2
         ],
    "C":1,
    "CF":1,
    "CI":1,
    "CP":1,
    "D":1,
    "E":0,
    "KS":0,
    "L":1,
    "LU":0,
    "O":1,
    "P":0,
    "R":1,
    "S":1,
    "W":1
   },
   "offline":[]
  },
  {
    "name":"Home Directory",
    "definedBy":"Admin",
    "providedBy":"Enterprise",
    "permissions":{
      "A":[
         {
           "ID":"com.quickoffice.quickofficeipad",
           "N":"Quick Office Pro"
         },
         {
           "ID":"com.dataviz.DocsToGoEAS",
           "N":"Docs to Go"
         }
        ],
      "AF":[
            1
           ],
      "DS":[
            1,
            2
           ],
      "C":1,
      "CF":1,
      "CI":1,
      "CP":1,
      "D":1,
      "E":0,
      "KS":0,
      "L":1,
      "LU":0,
      "O":1,
      "P":0,
      "R":1,
      "S":1,
      "W":1
      },
       "id":4,
       "path":"<homeDirectory>",
       "domain":null,
       "storage":"FileShare",
       "offline":[]
     },
     {
      "name":"psn2",
      "definedBy":"Admin",
      "providedBy":"Enterprise",
      "permissions":{
         "A":[
              {
               "ID":"com.quickoffice.quickofficeipad",
               "N":"Quick Office Pro"
              },
             {
               "ID":"com.dataviz.DocsToGoEAS",
               "N":"Docs to Go"
             }
            ],
         "AF":[
               1
              ],
         "DS":[
               1,
               2
              ],
         "C":1,
         "CF":1,
         "CI":1,
         "CP":1,
         "D":1,
         "E":0,
         "KS":0,
         "L":1,
         "LU":0,
         "O":1,
         "P":0,
         "R":1,
         "S":1,
         "W":1
       },
         "id":5,
         "path":"http://psn---1c.share1.example.com:8080/",
         "domain":null,
         "storage":"SharePoint",
         "offline":[
             {
               "path": "folderA/folderB"
             },
             {
               "path": "folder1/folder2"
             }
        ]
      }
 ]
```

Folder listing and file metadata response format

The following code samples display the folder listings and file metadata formats.

- File shares folder listing and metadata output
- SharePoint folder listing and metadata output
- SharePoint library metadata retrieval
- Server configuration output

File shares folder listing and metadata output

```
[
  {
     "etag":"4785af27-799-13ceb32544e",
     "name":"Assignment.docx",
     "type":"file",
     "date":"2013-02-18T02:46:49.678+0000",
     "size":1945
  },
  {
     "etag":"2611d270-9c8-13c96662ce6",
     "name":"Issues2.docx",
     "type":"file",
     "date":"2013-02-01T15:35:44.102+0000",
     "size":2504
  },
  {
     "etag":"6a4aec51-143c1375c27",
     "name":"Rome",
     "Folder Content":[
        {
        "etag":"24c78e93-b50-13ccf459ab9",
        "name":"Policies.docx",
         "type":"file",
         "date":"2013-02-12T16:38:30.841+0000",
        "size":2896
         }
  ],
  "type":"folder",
  "date":"2014-01-23T22:27:47.367+0000"
  }
]
```

SharePoint folder listing and metadata output

StoragePath is only returned for libraries with flat views and for the Shared With Me library. The storagePath value is the actual path in Microsoft SharePoint, which might be different from the folder whose content is listed. If storagePath returned is present and not empty, it must be included as a URL parameter in subsequent REST API calls that act on the folder or document.

```
[
 {
   "folderCount":1,
   "etag":"d839a970-325e-3ccd-8f4b-7df03fb939cb",
   "created":1383141704000,
   "subtype":"folder",
   "name":"Larry's Documents",
   "type":"folder",
   "fileCount":3,
   "url":"http://sp10.share1.example.com/SharedDocuments/Larry'sDocuments"
   "storagePath":""
  },
  {
   "checkoutRequired":0,
   "etag":"3009d45b-7af4-3c0e-9940-365872f513b7",
   "created":1377095293000,
   "currentVersionId":"3584",
   "checkoutType":[
        ],
    "name":"1.txt",
    "versionId":"3584",
    "isCheckout":0,
    "type":"file",
    "url":"http://sp10.share1.example.com/SharedDocuments/1.txt",
    "version":"7.0",
    "size":13,
    "modified":1410389577000
    "storagePath":""
  },
  {
    "checkoutRequired":0,
    "etag":"210fbffd-fdef-3cee-9af3-8a3b43775db6",
    "created":1377702468000,
    "currentVersionId":"512",
    "checkoutType":[
       ],
 "name":"ViewPG_iPhoneOS.pdf",
 "versionId":"512",
 "isCheckout":0,
 "type":"file",
 "url":"http://sp10.share1.example.com/SharedDocuments/ViewPG_iPhoneOS.pdf",
 "version":"1.0",
 "size":5751793,
 "modified":1377702468000
 "storagePath":""
  }
]
```

SharePoint library metadata retrieval

ViewScope specifies the scope for returning list items and list folders in a list view. ViewScope includes the following values:

- Default
- Recursive (value for flat views)
- RecursiveAll
- FilesOnly

For more information about ViewScope enumeration, visit the Microsoft Developer Network and see ViewScope enumeration.

```
{
  "name":"Shared Documents",
  "etag":"7420bc71-d5d6-3c9d-9ce5-062708183da4",
  "url":"http://sp13.share1.example.com/Shared Documents",
  "created":"2014-05-01T18:46:09.000+0000",
  "modified":"2014-07-31T01:32:17.000+0000",
  "displayName":"Documents",
  "views":[
    {
      "name":"All Documents",
      "default":false,
      "scope":"Default"
    },
    {
      "name":"DannCustomView",
      "default":true,
      "scope":"Recursive"
    }
  ],
 "type":"folder",
 "subtype":"list"
}
```

Server configuration output

```
{
  "SERVER_VERSION":"1.0.0.SNAPSHOT",
  "ALLOW_MEDIA_FILE_PLAYBACK_UNENCRYPTED":false,
  "FORCE_PENDING_UPLOADS":false,
  "DISPLAY_EVENT_INFO_FOR_CALENDAR_EVENTS":false,
  "STORE_PASSWORD":true,
  "ALLOW_MEDIA_FILE_TYPES":"",
  "KCD_ENABLED":false,
  "Actions":{
    "xlsx":["VIEW","EDIT"],
    "xls":["VIEW"],
    "docx":["VIEW", "PDF"]
       },
  "StorageProviders": [
  {
    "id":1,
    "name":"FileShare",
    "params":null,
    "enable":true,
    "sharedLinkInputParameters":null,
    "capabilities":["Standard"],
    "authtype":"windows",
    "storageprovider":"FileShare"
  },
  {
    "id":2,
    "name":"SharePoint",
    "params":null,
    "enable":true,
    "sharedLinkInputParameters":null,
    "capabilities":["Standard","Versioning","CheckInOut"],
    "authtype":"windows",
    "storageprovider":"SharePoint"
  },
  {
     "id":3,
     "name":"Box",
     "params":{"baseurl":"https://app.box.com/api/oauth2",
 "clientid":"xi0tfxmiynx00thohlko0fscsq5s6rqf",…},
     "enable":true,
     "sharedLinkInputParameters":null,
     "capabilities":["Standard","Sharable"],
     "authtype":"oauth2",
     "storageprovider":"Box"
  },
  {
     "id":4,
     "name":"WebDAV",
     "params":{"baseurl": null, "clientid": null, "clientsecret": null},
     "enable":true,
     "sharedLinkInputParameters":null,
     "capabilities":["Standard"],
     "authtype":"windows",
     "storageprovider":"WebDAV"
  }
  ],
"RMS_SERVICE": "AZURE-IP", //If Azure-IP is not enabled, AD-RMS is displayed.
"BEMS_APP_ID": "7a42fdca-a6f1-d4c4-be2d-ecb0beb2dc0c", //The BEMS_APP_ID is
 included when Azure-IP is enabled.
"RMS_SUPPORT":[
       "unprotect"
  ],
 }
```

User-defined repository and response format

The following code samples display the user-defined repositories and outputs:

- Get user-defined repositories retrieval
- Get specific user-defined repository retrieval
- Create a user-defined repository output

Get user-defined repositories retrieval

The following example retrieves all of the user-defined repositories that the user has created. In this example, user1 is assigned to four repositories.

```
{
 "repositories": [{
    "lastExpendedPathChanged": false,
    "id": 5,
    "name": "File Share repository",
    "path": "C:\\example",
    "storagetype": "FileShare",
    "repositorytype": "UserDefined",
    "acl": {
       "acelist": [{
         "user": {
           "name": "user1",
           "email": "user1@docs1.example.com",
           "domain": "DOCS1",
           "tenantId": "0"
            },
        "permissions": {
          "C": 1,
          "CF": 1,
          "CI": 1,
          "CP": 1,
          "D": 1,
          "E": 0,
          "L": 1,
          "O": 1,
          "P": 0,
          "R": 1,
          "W": 1,
          "AR": 1,
          "SL": 1
          }
      }]
   },
   "wdmanaged": false,
   "tenantId": "0"
  },
  {
   "lastExpendedPathChanged": false,
   "id": 6,
   "name": "SharePoint repository",
   "path": "http://docs1.example.com",
   "storagetype": "SharePoint",
   "repositorytype": "UserDefined",
   "acl": {
       "acelist": [{
          "user": {
            "name": "user1",
            "email": "user1@docs1.example.com",
            "domain": "DOCS1",
            "tenantId": "0"
            },
   "permissions": {
     "C": 1,
     "CF": 1,
     "CI": 1,
     "CP": 1,
     "D": 1,
     "E": 0,
     "L": 1,
     "O": 1,
     "P": 0,
     "R": 1,
     "W": 1,
     "AR": 1,
     "SL": 1
     }
  }]
 },
  "wdmanaged": false,
  "tenantId": "0"
  },
  {
   "lastExpendedPathChanged": false,
   "id": 7,
   "name": "SharePoint Online repository",
   "path": "https://"docs.sharepoint.com/personal/123456_docs1_onmicrosoft_com/
Social/Sites.aspx",
   "storagetype": "SharePoint Online",
   "repositorytype": "UserDefined",
   "acl": {
     "acelist": [{
       "user": {
       "name": "user1",
       "email": "user1@docs1.example.com",
       "domain": "DOCS1",
       "tenantId": "0"
      },
   "permissions": {
     "C": 1,
     "CF": 1,
     "CI": 1,
     "CP": 1,
     "D": 1,
     "E": 0,
     "L": 1,
     "O": 1,
     "P": 0,
     "R": 1,
     "W": 1,
     "AR": 1,
     "SL": 1
     }
   }]
 },
  "wdmanaged": false,
  "tenantId": "0"
  },
  {
   "lastExpendedPathChanged": false,
   "id": 57,
   "name": "box repo",
   "path": "https://app.box.com/folder/0",
   "storagetype": "Box",
   "repositorytype": "UserDefined",
   "acl": {
     "acelist": [{
       "user": {
       "name": "user1",
       "email": "user1@docs1.example.com",
       "domain": "DOCS1",
       "tenantId": "0"
      },
  "permissions": {
    "C": 1,
    "CF": 1,
    "CI": 1,
    "CP": 1,
    "D": 1,
    "E": 0,
    "L": 1,
    "O": 1,
    "P": 0,
    "R": 1,
    "W": 1,
    "AR": 1,
    "SL": 1
    }
  }]
 },
  "wdmanaged": false,
  "tenantId": "0"
 }],
 "totalCount": 4
}
```

Get specific user-defined repository retrieval

The following example retrieves a specific repository that the user has defined. In this example, user1 defined the repository called FS repository.

```
{
 "lastExpendedPathChanged": false,
 "id": 54,
 "name": "FS repository",
 "path": "C:\example",
 "storagetype": "FileShare",
 "repositorytype": "UserDefined",
 "acl": {
  "acelist": [{
       "user": {
       "name": "user1",
       "email": "user1@docs1.example.com",
       "domain": "DOCS1",
       "tenantId": "0"
       },
   "permissions": {
     "C": 1,
     "CF": 1,
     "CI": 1,
     "CP": 1,
     "D": 1,
     "E": 0,
     "L": 1,
     "O": 1,
     "P": 0,
     "R": 1,
     "W": 1,
     "AR": 1,
     "SL": 1
    }
 }]
},
 "wdmanaged": false,
 "tenantId": "0"
}
```

Create a user-defined repository output

The following example shows the output when a repository (for example, FileShare repository) is successfully created.

```
{ "message": "User defined repository created successfully with name FileShare
 repository" }
```

Update a user-defined repository output

The following example shows the output when a repository (for example, File Share repository) is successfully updated to a new name.

```
{ "message": "User repository updated successfully with name File Share
 repository-modified" }
```

Delete a user-defined repository output

The following example shows the output when a repository (for example, File Share repository) is successfully deleted.

```
{ "message": "User defined repository deleted successfully with id 3" }
```

Scenario

Consider the following environment:

- The Docs service is installed on a server called gdocsServer.
- You created an enterprise repository named "SPengg" which points to the SharePoint site at `http://SPengg/development` with ID 47.
- You created an enterprise repository named "FSit" which points to a File Share located at `\\shares\it` with ID 53.
- An existing file share exists on the organization's network located at `\\shares\hr\benefits` that users have access to.

The following table lists sample REST calls that can be made:

REST Call URI	Example
Get folder contents in a predefined enterprise repository.	GET http://gdocsServer/docs/1/SPengg/ProductX
Upload a document to a predefined enterprise repository location.	PUT http://gdocsServer/docs/1/53/tickets/screenshot.jpg HTTP Request body is document content
Get a list of all versions of a document.	GET http://gdocsServer/docs/1/47/requirements.docx?metadata&version=*
Check out a document.	POST http://gdocsServer/docs/1/SPengg/ProductZ/status.xlsx HTTP Request body is a JSON file: `{ "comments":"Update Sprint 3 status" }`

Legal notice

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