# Good Control Online Help

Version 4.2

# Table of Contents

# Revision History

*Good Control Online Help*

| Date | Description |
|---|---|
| 2017-09-25 | Corrected Default permissions and web services requests for predefined roles : The Help Desk Admin does not have permission to view users' access keys. |
| 2017-09-19 | Determining whether you should upgrade to BlackBerry UEM |
| 2017-08-28 | • Domain specification in Connectivity Profiles for Clients consists of the bare domain name, like **`qa.bigwebsite.com`** without any special leading characters *. or +.<br>• Corrected configurable agreement message size limit: not 4,000 chars, but 1M chars.<br>• Miscellaneous editorial corrections. |
| 2017-08-23 | Not supported: storing PAC files on UEM or GC |
| 2017-07-18 | Updated for latest release |
| 2017-03-08 | In Summary of Good Control Security Policies , clarified that "Always require password at application startup" and authentication delegation are mutually exclusive. |
| 2017-02-07 | • Corrected Configuring Web Proxy Server Properties for GP: web proxy properties are not editable in the GC console. You must edit the **`C:\good\gps.properties`** file on the Good Proxy server itself.<br>• Removed Good Proxy web proxy properties from list of Communication properties. |
| 2017-02-02 | Added important information for upgrading both Good Control and Good Proxy: Restoring custom (enterprise-issued) certificates from backup |
| 2017-01-31 | Version numbers updated for latest release; no content changes. |
| 2017-01-09 | • MDM not available for new installations of Good Control |
| 2016-12-22 | Updated Installing SSL certificates on GC and GP servers to include the correct **`keytool`** command syntax for creating a certificate signing request (CSR) that contains multiple hostnames/domains (Subject Alternative Name, or SAN, format), using the **`-ext`** option:<br>`keytool -certreq -keyalg RSA -alias new_alias_gc -file csr.csr -keystore`<br>`..\lib\security\cacerts -storepass changeit -ext`<br>`san=dns:`<br>`servername1.example.com`<br>`,dns:servername2.example.com,dns:servername3.example.com` |
| 2016-12-21 | Updated for latest release. See details in What's New in Good Control Online Help . |
| 2016-09-23 | In BlackBerry Access Secure Browser , clarified that in the **`setspn`** command syntax, *ADdomainUser* is the name of the service account that runs Good Control on this particular GC server. |

| Date | Description |
|---|---|
| 2016-08-25 | - Added details about behavior of applications to Optional: restoring BlackBerry Dynamics apps to a new device: discontinue use of old device<br>- Added information about Disabling Zipping of Logfiles |
| 2016-08-01 | Added qualification of the default interval to require passwords in:<br>- Optional: Allowing Android Fingerprint and interval to require password<br>- Allowing Apple Touch ID with Good Apps and Interval to Require Password |
| 2016-07-08 | Added Changing the GC and GP Service Password |
| 2016-07-26 | Added clarification about why application version numbers are retained in GC: About application versions . |
| 2016-07-20 | Clarified the behavior of the security policy **Prevent Screen Capture** in Summary of Good Control Security Policies |
| 2016-07-07 | Added:<br>- Behavior and Model of Disconnected/Inactive Containers<br>- Decommissioning Good Control or Good Proxy |
| 2016-06-30 | Updated for latest release:<br>- Optional: Allowing Apple Touch ID and Interval to Require Password<br>- Optional: Allowing Android Fingerprint and interval to require password<br>- Compliance Policy: Android Hardware Manufacturers or Models<br>- BlackBerry Access Secure Browser<br>- Info: client certificate sharing among BlackBerry Dynamics-based applications and on-Premise Good Control<br>- Info: support for Kerberos PKINIT: user authentication via PKI Certificates<br>- Optional: Bypassing the App Lock Screen<br>- Optional: restoring BlackBerry Dynamics apps to a new device: discontinue use of old device<br>- Good Proxy TCP Session Keep-Alive |
| 2016-06-06 | Added advice to Renew APNS Certificates Before Expiration |
| 2016-06-02 | Added action "Ring Device" for Windows Phone 8.1 devices to Device Management Operational Tasks: Device Status, Lock, Clear Password, Wipe, and Deactivate |
| 2016-05-16 | Added note About unique names for apps |
| 2016-05-13 | Added discrete steps for Entitling or denying end-users via entitlement groups (aka app groups) and Entitling or denying an individual end-user |

| Date | Description |
|---|---|
| 2016-05-13 | Added discrete steps for Entitling or denying end-users via entitlement groups (aka app groups) and Entitling or denying an individual end-user |
| 2016-05-05 | Re-added Policies high-level conceptual description. |
| 2016-04-15 | Corrected text in Installing Google Cloud Messaging API Keys . |
| 2016-04-08 | Clarified in Configuring Compliance Policy Rules that the policy **Base connectivity interval on auth delegate apps** applies only to GD-SDK based apps but does not include GFE, which is not based on the GD SDK. |
| 2016-04-07 | Added Blocking Android or iOS BlackBerry Dynamics apps by native version , which had been omitted in error. |
| 2016-03-17 | General revision of material, primarily relating to SSL/TLS and PKI certificates:<br>• Eliminated the terminology "self-signed certificate" in reference to the SSL/TLS certificate created by the GC at installation. This certificate is issued by the BlackBerry Dynamics Certificate Authority (GD CA) and is not "self-signed". It is now referred to as the "auto-installed certificate".<br>• Significant changes to Installing SSL Certificates on GC and GP Servers.<br>• Clarified usage of Trusted Authorities Tab , App Usage Tab , and Certificate Definitions Tab .<br>• Enhanced discussion of Certificate Management Policies .<br>• Clarified policy of Allowing Client Certificates .<br>• Added Summary of Good Control Security Policies .<br>• Updated steps for obtaining Licensing for BlackBerry Dynamics deployments. |
| 2016-03-14 | Clarified in Applying a Policy Set to an Application that it can take up to 24 hours for the new policy to propagate to GC servers. |
| 2016-03-10 | Truncated revision history to reduce bulk. |
| 2016-03-02 | Added overview to Good Control web services in Good Control Web Services. |
| 2016-02-17 | Updated clickpaths/steps in Create Google Cloud Messaging API keys because Google changed their site again. |
| 2016-02-09 | Clarified several topics related to SSL/TLS certificates:<br>• Installing SSL Certificates on GC and GP Servers relates to replacing the certificates created during installation of the GC and GP.<br>• Certificates relates to trusted Certificate Authorities and end-user PKI certificates. |
| 2016-02-01 | Included cross-reference to document describing Integrating BES12 and BlackBerry Dynamics. |
| 2016-01-15 | Updated for latest release: some limitations now removed: |

| Date | Description |
|---|---|
| | - Apple DEP Profiles can now be assigned to more than 100 devices at a time.<br>- Auto-push of managed apps is no longer limited to the first version of an app. |
| 2015-12-23 | Updated for latest release. |
| 2015-12-14 | Removed sections regarding creating custom roles in Good Control Cloud; this feature is not available. |
| 2015-11-13 | Eliminated conflicting information formerly included under "Security Policy for Root Certificates". |
| 2015-10-30 | Updated the procedure for Updating the Passwords of Private Keys Associated with GC or GP Certificates to remove unecessary references to server.xml file, which is no longer needed. |
| 2015-10-28 | Updated the procedure for Updating the GC or GP Certificate Keystore/Truststore Password . Editing the server.xml file is no longer required; instead, you must obfuscate the password for both the keystore and the truststore with the **`changepwd`** command. |
| 2015-10-08 | Updated the snippet shown for GC **`server.xml`**'s **`<Connector>`** element in:<br>- Installing SSL Certificates on GC and GP Servers<br>- Updating the GC or GP Certificate Keystore/Truststore Password |
| 2015-10-07 | Added Default permissions and web services requests for predefined roles |
| 2015-09-23 | In Managing Application Permissions for a Group , if you are entitling end users to a new app version and denying the old version, be sure to entitle the new version first. |
| 2015-09-21 | Expanded and corrected details about "Route All" in External Web Proxy : You must set the name of the primary GP cluster (at a minimum). |
| 2015-09-16 | Readded missing topic Configuring Web Proxy Server Properties for GC or GP that was mistakenly omitted. |
| 2015-09-10 | Updated for latest release: Connectivity profile overrides can now either add servers/domains or remove servers/domains from the baseline profile, as described in Connectivity Profiles for Clients . |

# Determining whether you should upgrade to BlackBerry UEM

If you require MDM or MAM capabilities, you must manage BlackBerry Dynamics apps using BlackBerry UEM. When you upgrade from Good Control to BlackBerry UEM, you not only get to use the great feature set that Good Control provides but you also get to take advantage of an enhanced feature set such as:

- Support for more policies for operating systems
- Better app management
- More container types

- Improved administration and provisioning
- Advanced connectivity and networking
- Expanded compliance and integrity checking
- Additional email, content, location, and certificate features
- Access to BlackBerry Web Services APIs

For information on how to use BlackBerry UEM to manage BlackBerry Dynamics apps, see the Getting started with BlackBerry UEM and BlackBerry Dynamics content.

For more information on the benefits of using BlackBerry UEM, see Benefits of upgrading from Good Control to BlackBerry UEM.

# Overview

Welcome to BlackBerry Dynamics, brought to you by BlackBerry.

An integral part of BlackBerry Dynamics is the Good Control server. With the Good Control server's console you can create and manage users, provision access keys, control user and device access, policies, and application permissions, and much more.

This is the PDF rendition of the online help for On-premise Good Control.

**Important:** Be sure to expand your browser window wide enough until you see the search text box in the upper right and the navigation on the left.

The help is in general structured in the same order as the menu selections in the Good Control console itself.

BlackBerry offers a number of other sources of information about BlackBerry Dynamics. See the Resource Library on the BlackBerry Developer Network for documentation relating to GD server installation and configuration, specific features or newly announced features, guides for developers with the GD SDK, and more. For details, see BlackBerry Dynamics documentation .

# What's New in Good Control Online Help

## Enterprise SSL certs: essential pre-upgrade planning and best practices

If your enterprise has its own Certificate Authority (CA) to issue SSL certificates that you use to secure network connections, the discussion, best practices, and caveats of working with them and the GC, the details here are essential to you. If not, you do not need these details.

Formerly, Good Control required you to use the Java **`keytool`** command to store enterprise certificates (that is, custom certificates your own enterprise generates as a Certificate Authority (CA) and add those certificates to Good Control keystore.

> **Note:** Direct editing of the keystore is not required or supported in this release. This function is now done via the GC console.

### Best practices with enterprise-CA-issued certs

These recommendations are to avoid problem scenarios described here. The re-upload needs careful planning and timing of execution.

Essentially, you must make sure that your network peripherals and end-user client apps are prepared before you re-upload and that your GP servers are accessible and running.

- Pre-upgrade: save your enterprise CA certs
- Network peripheral setup: certs pre-loaded
- Client app setup: latest versions installed on devices before re-upload of certs
- Client apps running at time of cert re-upload
- Timing the exact re-upload: cert effective date/time
- At time of re-upload, Good Proxy clusters must be accessible and running

#### Pre-upgrade: save your enterprise CA certs

Be sure to save copies of your enterprise certificates before you upgrade. After your upgrade is complete, you will re-upload the enterprise certificates to Good Control.

#### Network peripheral setup: certs pre-loaded

Before re-uploading your enterprise-CA-issued certs into the GC, make sure that *all* your network connection devices are pre-loaded with your certs.

This best practice relates to *all* network devices that you protect via your certs: switches (such as F5), routers, load balancers, web proxies, app servers: all equipment that already relies on your certs.

This best practice includes devices that are configured to use the Direct Connect network configuration.

If you do not prepare these peripherals in advance, the GC cannot establish a secure connection to them.

### Client app setup: latest versions installed on devices before re-upload of certs

Before re-uploading your enterprise-CA-issued certs into the GC, make sure that all BlackBerry Dynamics-based apps are upgraded to the latest versions released along with this version of the GC. Also make sure tha these latest apps are fully deployed to all end-users that require them.

- This applies to apps distributed by BlackBerry, such as BlackBerry Access, BlackBerry work, and so on.
- This advice also includes any BlackBerry Dynamics-based apps your enterprise itself has developed and deployed. They must be re-built with the latest version of the BlackBerry Dynamics SDK and must be re-deployed *prior* to re-uploading certs into the GC.

Without this best practice, the older apps cannot communicate with the GC to receive your re-uploaded enterprise-CA-issued certs, and end-users will have to re-provision them on their devices.

### Client apps running at time of cert re-upload

All the affected apps that have been rebuilt with the latest BlackBerry Dynamics SDK and re-deployed to all end users *must be running (in the background or otherwise) when you re-upload the certificate*.

If not, the app will not receive the new cert, and the next time it starts, it will not be able to secure the connection.

### Timing the exact re-upload: cert effective date/time

Be sure that the effective date of your certs is not less than the GC policy for connectivity checking of client apps.

Be sure that you allow sufficient time for re-uploaded certs to propagate to connected GP clusters and devices.

BlackBerry recommends a minimum interval of 30 minutes. That is, stagger the effective time for all the certs so that they take effect in sequence, at least 30 minutes apart.

> **Important:** The exact length of time you need to wait for propagation depends on which certs you are replacing and how my there are. If you are replacing only one cert for server-to-server communication, 10 minutes might be sufficient, because there are not many servers involved.
>
> However if you are replacing the certs used for communication with apps on devices (those on the GC's **Client Certificates** tab), you should allow much longer than 30 minutes, depending on the number of deployed apps. The greater the number of apps, the longer you should allow between re-uploading certs.

### At time of re-upload, Good Proxy clusters must be accessible and running

Before re-uploading your enterprise-CA-issued certs into the GC, make sure that *all* your GP clusters and servers are accessible via the network and are running.

Otherwise, you will have to manually reload your certs on each GP server that was not accessible and running at the time you re-upload your certs into the GC console.

## Mechanisms to load certs into the GC

**Options to re-upload your enterprise-CA-issued certificates in Good Control**:

- In the GC console, use **Settings** > **Server Certificates**.
- With the SOAP APIs RenewCertificate or RenewCertificatesByCertDefId

### Forcing import certificates into Good Proxy if necessary: install repair

This topic is important for you only if your Good Control server and Good Proxy are not communicating and you suspect a problem with your enterprise-CA-issued SSL certificates.

To avoid problems when you re-upload your enterprise-CA-issued, follow the best practices detailed in Pre-upgrade: save your enterprise CA certs .

Good Control and Good Proxy secure their own communications by way of an SSL certificate. If you load an SSL certificate into GC, the same certificate details are also updated on the GP *unless the GP is not reachable when you load the new certificate into the GC.* In this case, you will have to manually load the pertinent SSL certificate or certificates onto every GP that fails to communicate with its GC.

To correct this problem and to re-load your own certs into the affected GP servers, run the GP installation program and select the repair option.

## Installation and upgrade

**Important:** If your enterprise uses its own SSL certifcates issued by your own private Certificate Authority (CA), *before you upgrade the GC or GP*, be sure to read Pre-upgrade: save your enterprise CA certs to avoid possible problems during upgrade.

### Info: Good Control installation allows non-admin AD account

In the past, the installation program for Good Control accepted only a single administrator account credential to be the default Good Control administrator and to install and run the GC service.

Now, to separate these functions, the Good Control installer as an option allows you to enter two different accounts. (This feature is similar to the Good Proxy installer.)

Separate accounts are not required. You have the option of using a single account.

If separate accounts are used, the installer validates both of them to confirm the accounts have the necessary administrative rights and the **Log on as user** privilege.

If you decide on separate accounts, make sure that the accounts also have the necessary privileges via Windows auth to work with the Microsoft SQL database.

## Administrator interface

### New: Second production license not required

Formerly, you had to obtain a production license for each server you added to a GC cluster.

Now, only one production license is required, which you obtain (as in the past) from community.good.com. Thereafter, you can add more servers to your cluster without having to obtain a license.

#### Change: "License" link removed: Only one license required for entire cluster

Formerly, with Good Control's Licenses page, you generated new production licenses for additional servers in the GC cluster.

Now, the Licenses link and page have been removed from the system.

## Managing GC, GP, and logging server properties

The Good Control console displays links in the navigation bar for managing server properties:

- GC Server Properties
- GP Server Properties
- Logging Properties

#### New: Good Proxy properties now editable in Good Control

Formerly, to change the values of Good Proxy properties or add new properties, an administrator had to edit several files on the GP server itself.

Now, all Good Proxy properties are viewable in the Good Control console itself and those properties that are settable by the administrator can be edited directly in the console.

Changes made to these properties are propagated to all Good Control and Good Proxy servers, regardless of clustering.

A complete list of all Good Proxy properties is in GP property reference .

**To edit Good Proxy properties, in Good Control:**

1. Navigate to **Settings** > **GP Server Properties**.
2. Find the desired property.
3. Change its value.
4. The GC UI gives you the option to limit the property update to a specific cluster.

   > **Note:** This option is not functional at this time. Property updates are sent to all GP servers, regardless of clustering.

5. In the upper right, click **Update** to save your changes.

#### New: GC and GP logging properties now editable in Good Control

Formerly, to change the values of logging properties, an administrator had to edit log4j files on the Good Control or GP servers themselves.

Now, all Good Control and Good Proxy logging properties are viewable in the Good Control console itself and those properties that are settable by the administrator can be edited directly in the console.

Changes made to these properties are propagated to all Good Control and Good Proxy servers in a cluster.

A complete list of all logging properties is in Logging property reference .

**To edit Good Proxy properties, in Good Control:**

1. Navigate to **Servers** > **Logging Properties** .
2. Choose either **Good Control** or **Good Proxy**.
3. Find the desired property.
4. Change its value.
5. In the upper right, click **Update** to save your changes.

# Network

## New: specify Good Proxy clusters for provisioning

In the past, all servers in all defined Good Proxy clusters were used for application provisioning, regardless of their location or other concerns.

Now, in Good Control's **Settings** > **Clusters** > **GP Clusters** tab, you can specify exactly which of your GP clusters should be used to provision applications. This allows you to provision applications from GP clusters that you can ensure are available or to exclude those GP clusters that should not be considered eligible for provisioning, such as GP clusters in your disaster recovery configuration that should not be part of day-to-day operations.

By default, all GP clusters are eligible for provisioning. Only when you set at least one GP clustetr for provisioning, the other defined clusters are then excluded from this function.

**To change whether a GP cluster should be used for application provisioning, in Good Control**:

1. Navigate to **Settings** > **Clusters** > **GP Clusters** tab.
2. Find the desired cluster name.
3. To the right of the cluster name, find the checkbox **Use For Provisioning**.
4. If this cluster should be used for provisioning, check the checkbox. Otherwise, do not check the checkbox.
5. In the upper right, click **Update** to save your changes. To discard your changes, navigate away from this page.

### SOAP API operation

You can also set the new **`provision`** field in the SOAP API operation **`UpdateGPClusters`**.

# Certificates

## New: changes to Certificate Definitions tab

Good Control's **Certificates** > **Certification Definitions** tab has the following changes:

- The **Test Connection** button does *not* save the definition to Good Control's database, as it did in the past.
- To save the definition to Good Control's database, you must click **Save**.
- The list of defined certificates now displays characteristics of the definitions, such as Require user-entered password or OTP.

## New: automatic renewal or deletion of CA-fetched PKI certificates

If you have implemented the PKI certificate "fetching" feature described in *PKI Cert Creation via Good Control: Reference Implementation* at https://community.good.com/docs/DOC-7151, in Good Control:

- You can specify the automatic renewal of these certificates.
- You can cause them to be deleted when they expire.
- You can automatically remove duplicate certificates

**To specify automatic renewal of certificates in Good Control:**

1. Navigate to **Certificates** > **Certificate Definitions** tab.
2. Find the desired certificate definition.
3. Click **Edit**.
4. Find **Enable certificate renewal** *XX* **before expiration**
5. Click the checkbox.
6. Use the pulldown menu to set the number of days prior to expiration for the renewal to occur. Values are as follows:
   - 7
   - 14
   - 30 (default)
   - 60
   - 90
   - 120
   - 180
7. Click **Save** to save the changes or **Cancel** to discard them.

**To specify automatic deletion of expired of certificates in Good Control:**

1. Navigate to **Certificates** > **Certificate Definitions** tab.
2. Find the desired certificate definition.
3. Click **Edit**.
4. Find **Delete certificate upon expiry**
5. Click the checkbox.
6. Click **Save** to save the changes or **Cancel** to discard them.

**To specify automatic remove duplicate certificates in Good Control:**

1. Navigate to **Certificates** > **Certificate Definitions** tab.
2. Find the desired certificate definition.
3. Click **Edit**.
4. Find **Remove duplicate certificate (Certificate that expires first will be removed)**

5. Click the checkbox.
6. Click **Save** to save the changes or **Cancel** to discard them.

## Adminstrator-initiated PKI cert renewal for client apps

In addition to automated cert renewal, the administrator can force certificate renewal for individual users via the Good Control console.

> **Note:** Forced cert renewal operates only with client apps built with the latest BlackBerry Dynamics SDK. Apps built with earlier release cannot be forced. Good Control does not display an error message in this case.
>
> After the older apps have been upgraded, the administrator can then force the renewal.

**Steps to force cert renewal in Good Control:**

1. In the left nav, click **Users and Groups**.
2. Find the affected user name and click the name.
3. Click **Certificates**.
4. To initiate the cert renewal, click the circular arrows at the far right of the certificate.

## Required: update your PKI Connector to support certificate renewal

The reference implementation as delivered does not include the logic necessary to work with the certificate renewal feature.

your PKI Connector must include a function to return values that indicate the capabilities of your connector. Those capabilities are as follows:

- getP12: New cert enrollment only
- getP12, renewCert: Bother new cert enrollment and certificate renewal

The necessary design aspects of certificate renewal are detailed in BlackBerry's User Certificate Management Protocol.

After you modify your PKI Connector and deploy it, you need to inform Good Control that the connector has new capabilities.

The latest version of Good Control includes an **Update connector capabilities** button (under **Certificates** tab) whereby you inform Good Control of your PKI connector's capabilities. The server makes a request to your connector to discover the capabilities based on the values you return.

## Info: PKI Connector notified when certificates are removed if connector supports removal capability

Good Control supports a PKI Connector that allows you to interact with a Certificate Authority server. A reference implementation in Java for a PKI Connector is described at *PKI Cert Creation via Good Control: Reference Implementation*.

The PKI Connector is now notified whenever a certificate has been removed from the GC.

For certificate removal, the PKI connector must be configured to support certificate removal, and the connector details must be updated in GC. Complete details on developing a connector and configuring GC to use it are in *PKI Cert Creation via Good Control: Reference Implementation*.

## New or changed security or compliance policies

### Setting "No password" policy

**Note:** These policies were introduced with Good Control v3.0.50.70 in December, 2016, but were not enforced by the BlackBerry Dynamics SDK at that time. Now with this latest release of the BlackBerry Dynamics SDK, they are enforced.

The security policies below allow your end users to avoid having to set an application password when a BlackBerry Dynamics-based application is activated:

- Do not require user password for Android
- Do not require user password for iOS
- Do not require password for macOS
- Do not require password for Windows (UWP)

By default, these policies are not enabled.

If you enable one of these policies, the following message is displayed in the Good Control console:

> **Warning**
>
> Disabling the BlackBerry application password significantly reduces security of BlackBerry containers and Enterprise Network. Use of this mode is strongly discouraged.

#### Security and design considerations for policy sets

When you plan to enable the "no password" policy, keep these considerations in mind:

1. Consider the security ramifications in your environment carefully. Enabling "no password" is only one of the design options available. Others include authentication delegation, enabling "no password" on only certain devices, device management, and more. You should consider enabling "no password" for only a certain select group of users whose devices are under tight control via device management profiles or other controls.
2. The policy can be enabled for either Android, or iOS, or both, in the same policy set. Thus, for example, in a single policy set, you can enable "no password" for Android while not enabling it for iOS.
3. Do not enable "no password" and authentication delegation in the same policy set. You should consider these two security policies as mutually exclusive. Enabling both policies will cause unnecessary switching among applications without any apparent benefit.
4. Enabling "no password" does not permit authentication in the background, because there is no authentication.

Effects and behavior of enabling "no password"

- For behavior of enabling the "no password policy," see New: Support for "No Password" security policy
- For programming aspects on Android, see New: optional APIs for Android.
- For programming aspects on iOS, see New: optional APIs for iOS.

## New or changed properties

### New: allow new macOS devices

| | | |
|---|---|---|
| allow.new.MAC.device | Allow any new macOS device | Default: true<br>Global: yes<br>Restart: no |

### New: Exporting server properties

You can export all Good Control and Good Proxy server properties to a comma-separated value (CSV) for easier troubleshooting.

The name of the exported file is as follows:

**`Server_Properties_Report_`***`DD`*`-`*`MM`*`-`*`YYYY`***`.csv`**

**To export all properties, in Good Control:**

1. Navigate to **Export Server Properties**.
2. Click **Export**.

### GP property reference

| Property | Description | Editable? |
|---|---|---|
| eacp.command.service.nslookup.srv.ldap | Enables LDAP over TCP for Active Directory servers. Active Directory servers offer the LDAP service over the TCP protocol; therefore, clients find an LDAP server by querying DNS for a record of the form: _ldap._ tcp. DnsDomainName<br>• true = indicates that GP uses LDAP for nslookup of a given service hostname<br>• false = GP uses reverse DNS lookup directly, using the given service hostname<br>Default: false | yes, editable |
| gc.admin.name | Username of Good Control administrator<br>Default: none | not editable |

| Property | Description | Editable? |
|---|---|---|
| gc.auth.token | Secret token to authenticate GC with GP<br>Default: none | not editable |
| gc.server.port | Port of GC server<br>Default: 443 | not editable |
| gc.server.uri | SOAP endpoint of GC server with which this GP should be registered.<br>Default: none, depends on name of server | not editable |
| gd.product.capability | GP server feature set used to compare with GC server feature set during GP registration to make sure that GC and GP are compatible.<br>Default: none. | not editable |
| gd.product.domain | Active Directory domain of the GP<br>Default: none. Set by installer. | not editable |
| gd.product.hostname | GP server name<br>Default: none. Set by installer. | not editable |
| gd.product.licensekey | GC and GP license keys as recorded in GDN<br>Default: none | not editable |
| gd.product.loginkey | GP server login credentials to BlackBerry Dynamics NOC for uploading GP server logs<br>Default: none | not editable |
| gd.product.serialnum | GC and GP serial numbers as recorded in GDN<br>Default: none | not editable |
| gd.product.type | Differentiate between GC service and GP service.<br>• GPS = Good Proxy<br>• GMC = Good Control<br>Default: for GP, GPS | not editable |
| gd.product.version | Version number of this GP<br>Default: none | not editable |
| gd.security.keystore.alias | Alias for the GP's keystore<br>Default: good-proxy | yes, editable |
| gd.security.keystore.file | Location of GP keystore file | yes, editable |

| Property | Description | Editable? |
|---|---|---|
| | Default:*GP_installation_ directory* **\jre\security\lib\cacerts** | |
| gd.security.rootcert.alias | Alias for the root certificate of the GP<br>Default: good-dynamics | yes, editable |
| gps.auth.token | Secret token to authenticate GP with GC | not editable |
| gps.directconnect.port | Port for Direct Connect configuration<br>Default: 17533 | not editable |
| gps.dns.server.ttl.ms | Time-to-live in milliseconds for the DNS server connections., i.e. time to wait for DNS server response.<br>Default: 1.8M milliseconds | yes, editable |
| gps.logfiles.days | Length of time to retain logfiles<br>Default: 10 days | yes, editable |
| gps.product.installdir | Installation directory for GP<br>Default: none. Set by installer | not editable |
| gps.product.registered | Flag for whether this GP has been registered with BlackBerry<br>Default: false | not editable |
| gps.server.fqdn | Fully qualified domain name for this GP server<br>Default: none. Set by installer | not editable |
| gps.server.name | Bare hostname of this GP server<br>Default: none. Set by installer | not editable |
| gps.server.port | Non-secured port for this GP server<br>Default: 17080 | not editable |
| gps.server.secure.port | Secure port for this GP server<br>Default: 17443 | not editable |
| gps.service.name | Name of the GP service on Windows<br>Default: GPS | yes, editable |
| gps.status.request.frequency | Allowable frequency for /status request on this GP | yes, editable |
| gps.tcp.session.timeout | Length of time that a TCP connection can be inactive before it is closed. | yes, editable |

| Property | Description | Editable? |
|---|---|---|
| | **Important:** Do not alter this setting without direct consultation with BlackBerry.<br>Default: 1,800 seconds | |
| gps.unalias.hostname | For DNS lookups of app servers, use either IP address or hostname<br>• true= GP uses reverse DNS lookup with IP address of app server<br>• false = GP uses app server hostname for lookup<br>Default: false | yes, editable |
| gwy.push.connection.timeout | Timeout of persistent connection to MDC server in BlackBerry Dynamics NOC for push notifications<br>Default: 45 seconds | yes, editable |
| gwy.push.port | Port of MDC server in BlackBerry Dynamics NOC<br>Default: 443 | not editable |
| gwy.push.prot | Protocol for communications<br>Default: 1 | not editable |
| gwy.push.register | GP is registered with MDC server<br>true = GC is registered with BlackBerry Dynamics NOC<br>Default: true | not editable |
| gwy.push.request.timeout | Timeout of request to MDC server BlackBerry Dynamics NOC<br>Default: 20 seconds | yes, editable |
| gwy.push.secure | Use SSL for connection to MDC serverBlackBerry Dynamics NOC<br>Default: false | not editable |
| gwy.push.server | Name of MDC server in BlackBerry Dynamics NOC<br>Default: gdmdc.good.com | not editable |
| gwy.push.socket.timeout | Timeout in establishing socket connection to MDC server BlackBerry Dynamics NOC<br>Default: 45 seconds | yes, editable |

| Property | Description | Editable? |
|---|---|---|
| health.check.enabled | Whether to perform additional checks on GP health<br>Default: true | yes, editable |
| health.check.interval | How often to check GP health<br>Default: 3.6M milliseconds (1 hour) | yes, editable |
| log.upload.date.name.format | Date format for timestamp of GP logfile names<br>Default: yyyy-MM-dd | not editable |
| log.upload.dir | Path to directory on server where logs are stored<br>Default: none. Set by installer. | not editable |
| log.upload.url | URL on this GP where logfiles can be uploaded<br>Default: none | not editable |
| mdc.server.name | Name of MDC server in BlackBerry Dynamics NOC<br>Default: gdmdc.good.com | not editable |
| mdc.server.port | Port of MDC server in BlackBerry Dynamics NOC<br>Default: 443 | not editable |
| proxy.auth.domain | Active Directory domain for authentication login to external Web proxy server<br>Default: none | yes, editable |
| proxy.auth.password | Password of username for authenticating to external Web proxy server<br>Default: none | yes, editable |
| proxy.auth.username | User name for connecting to external Web proxy server<br>Default: none | yes, editable |
| proxy.https.host | Name of external Web proxy server<br>Default: none | yes, editable |
| proxy.https.port | Port number for HTTPS connection to external Web proxy server<br>Default: none | yes, editable |
| proxy.urls | URLs that must be proxied<br>Default: none | yes, editable |

| Property | Description | Editable? |
|---|---|---|
| proxy.use | Use an external Web proxy server<br>Default: false | yes, editable |
| relay.gps.key | Key to access relay server in BlackBerry Dynamics NOC<br>Default: none | not editable |
| relay.server.name | Name of relay server in BlackBerry Dynamics NOC<br>Default: gdrelay.good.com | not editable |
| relay.server.port | Port number of relay server in BlackBerry Dynamics NOC<br>Default: 443 | not editable |

### Logging property reference

| Property | Description | Good Control | Good Proxy |
|---|---|---|---|
| Maximum server log file size | Allowable values: from 100 KB to 1 GB | Default: 256 MB | Default: 256 MB |
| Maximum server log file age | In days | Default: 10 days | Default: 10 days |
| Compress server log files | Allowable values: true\|false | Default: on | Default: on |
| Server logging level | Allowable values: Info\|Debug | Default: Info | Default: Info |

# What was new in previous releases

## Installation and upgrade

**Important:** If your enterprise uses its own SSL certifcates issued by your own private Certificate Authority (CA), *before you upgrade the GC or GP*, be sure to read Pre-upgrade: save your enterprise CA certs to avoid possible problems during upgrade.

### Required before upgrade: AD domain name in DB upgrade scripts

If you plan to upgrade from earlier versions of Good Control or Good Proxy, the database upgrade scripts now require the name of the Active Directory domain for user accounts.

The reason for this name is to avoid possible ambiguity during the upgrade.

**Important:** You must add this domain name *before* upgrading.

## Enterprise SSL certs: essential pre-upgrade planning and best practices

If your enterprise has its own Certificate Authority (CA) to issue SSL certificates that you use to secure network connections, the discussion, best practices, and caveats of working with them and the GC, the details here are essential to you. If not, you do not need these details.

Formerly, Good Control required you to use the Java **`keytool`** command to store enterprise certificates (that is, custom certificates your own enterprise generates as a Certificate Authority (CA) and add those certificates to Good Control keystore.

**Note:** Direct editing of the keystore is not required or supported in this release. This function is now done via the GC console.

### Best practices with enterprise-CA-issued certs

These recommendations are to avoid problem scenarios described here. The re-upload needs careful planning and timing of execution.

Essentially, you must make sure that your network peripherals and end-user client apps are prepared before you re-upload and that your GP servers are accessible and running.

- Pre-upgrade: save your enterprise CA certs
- Network peripheral setup: certs pre-loaded
- Client app setup: latest versions installed on devices before re-upload of certs
- Client apps running at time of cert re-upload
- Timing the exact re-upload: cert effective date/time
- At time of re-upload, Good Proxy clusters must be accessible and running

#### Pre-upgrade: save your enterprise CA certs

Be sure to save copies of your enterprise certificates before you upgrade. After your upgrade is complete, you will re-upload the enterprise certificates to Good Control.

#### Network peripheral setup: certs pre-loaded

Before re-uploading your enterprise-CA-issued certs into the GC, make sure that *all* your network connection devices are pre-loaded with your certs.

This best practice relates to *all* network devices that you protect via your certs: switches (such as F5), routers, load balancers, web proxies, app servers: all equipment that already relies on your certs.

This best practice includes devices that are configured to use the Direct Connect network configuration.

If you do not prepare these peripherals in advance, the GC cannot establish a secure connection to them.

### Client app setup: latest versions installed on devices before re-upload of certs

Before re-uploading your enterprise-CA-issued certs into the GC, make sure that all BlackBerry Dynamics-based apps are upgraded to the latest versions released along with this version of the GC. Also make sure tha these latest apps are fully deployed to all end-users that require them.

- This applies to apps distributed by BlackBerry, such as BlackBerry Access, BlackBerry work, and so on.
- This advice also includes any BlackBerry Dynamics-based apps your enterprise itself has developed and deployed. They must be re-built with the latest version of the BlackBerry Dynamics SDK and must be re-deployed *prior* to re-uploading certs into the GC.

Without this best practice, the older apps cannot communicate with the GC to receive your re-uploaded enterprise-CA-issued certs, and end-users will have to re-provision them on their devices.

### Client apps running at time of cert re-upload

All the affected apps that have been rebuilt with the latest BlackBerry Dynamics SDK and re-deployed to all end users *must be running (in the background or otherwise) when you re-upload the certificate*.

If not, the app will not receive the new cert, and the next time it starts, it will not be able to secure the connection.

### Timing the exact re-upload: cert effective date/time

Be sure that the effective date of your certs is not less than the GC policy for connectivity checking of client apps.

Be sure that you allow sufficient time for re-uploaded certs to propagate to connected GP clusters and devices.

BlackBerry recommends a minimum interval of 30 minutes. That is, stagger the effective time for all the certs so that they take effect in sequence, at least 30 minutes apart.

> **Important:** The exact length of time you need to wait for propagation depends on which certs you are replacing and how my there are. If you are replacing only one cert for server-to-server communication, 10 minutes might be sufficient, because there are not many servers involved.
>
> However if you are replacing the certs used for communication with apps on devices (those on the GC's **Client Certificates** tab), you should allow much longer than 30 minutes, depending on the number of deployed apps. The greater the number of apps, the longer you should allow between re-uploading certs.

### At time of re-upload, Good Proxy clusters must be accessible and running

Before re-uploading your enterprise-CA-issued certs into the GC, make sure that *all* your GP clusters and servers are accessible via the network and are running.

Otherwise, you will have to manually reload your certs on each GP server that was not accessible and running at the time you re-upload your certs into the GC console.

### Mechanisms to load certs into the GC

**Options to re-upload your enterprise-CA-issued certificates in Good Control**:

- In the GC console, use **Settings** > **Server Certificates**.
- With the SOAP APIs RenewCertificate or RenewCertificatesByCertDefId

### Info: Windows Server 2016 now supported

You can now install Good Control and Good Proxy on Microsoft Windows Server 2016.

> **Note:** For Windows Server 2016, only new installations of the latest version of Good Control and Good Proxy are supported. New installations of earlier versions or upgrading from earlier versions are not supported. Windows Server 2016 was only recently release and so could not be accommodated by earlier versions of Good Control and Good Proxy.

### Info: Win32 platform now supported

Good Control now supports the Win32 platform, the 32-bit API for Windows management from Microsoft.

## Administrator interface

### New: Second production license not required

Formerly, you had to obtain a production license for each server you added to a GC cluster.

Now, only one production license is required, which you obtain (as in the past) from community.good.com. Thereafter, you can add more servers to your cluster without having to obtain a license.

#### Change: "License" link removed: Only one license required for entire cluster

Formerly, with Good Control's Licenses page, you generated new production licenses for additional servers in the GC cluster.

Now, the Licenses link and page have been removed from the system.

### New: Good Proxy properties now editable in Good Control

Formerly, to change the values of Good Proxy properties or add new properties, an administrator had to edit several files on the GP server itself.

Now, all Good Proxy properties are viewable in the Good Control console itself and those properties that are settable by the administrator can be edited directly in the console.

Changes made to these properties are propagated to all Good Control and Good Proxy servers, regardless of clustering.

A complete list of all Good Proxy properties is in GP property reference .

**To edit Good Proxy properties, in Good Control:**

1. Navigate to **Settings** > **GP Server Properties**.
2. Find the desired property.
3. Change its value.
4. The GC UI gives you the option to limit the property update to a specific cluster.

   > **Note:** This option is not functional at this time. Property updates are sent to all GP servers, regardless of clustering.

5. In the upper right, click **Update** to save your changes.

### New: GC and GP logging properties now editable in Good Control

Formerly, to change the values of logging properties, an administrator had to edit log4j files on the Good Control or GP servers themselves.

Now, all Good Control and Good Proxy logging properties are viewable in the Good Control console itself and those properties that are settable by the administrator can be edited directly in the console.

Changes made to these properties are propagated to all Good Control and Good Proxy servers in a cluster.

A complete list of all logging properties is in Logging property reference .

**To edit Good Proxy properties, in Good Control:**

1. Navigate to **Servers** > **Logging Properties** .
2. Choose either **Good Control** or **Good Proxy**.
3. Find the desired property.
4. Change its value.
5. In the upper right, click **Update** to save your changes.

## Certificates

### New: changes to Certificate Definitions tab

Good Control's **Certificates** > **Certification Definitions** tab has the following changes:

- The **Test Connection** button does *not* save the definition to Good Control's database, as it did in the past.
- To save the definition to Good Control's database, you must click **Save**.
- The list of defined certificates now displays characteristics of the definitions, such as Require user-entered password or OTP.

### New: automatic renewal or deletion of CA-fetched PKI certificates

If you have implemented the PKI certificate "fetching" feature described in *PKI Cert Creation via Good Control: Reference Implementation* at https://community.good.com/docs/DOC-7151, in Good Control:

- You can specify the automatic renewal of these certificates.
- You can cause them to be deleted when they expire.
- You can automatically remove duplicate certificates

**To specify automatic renewal of certificates in Good Control:**

1. Navigate to **Certificates** > **Certificate Definitions** tab.
2. Find the desired certificate definition.
3. Click **Edit**.
4. Find **Enable certificate renewal** *XX* **before expiration**

5. Click the checkbox.
6. Use the pulldown menu to set the number of days prior to expiration for the renewal to occur. Values are as follows:
    - 7
    - 14
    - 30 (default)
    - 60
    - 90
    - 120
    - 180
7. Click **Save** to save the changes or **Cancel** to discard them.

**To specify automatic deletion of expired of certificates in Good Control:**

1. Navigate to **Certificates** > **Certificate Definitions** tab.
2. Find the desired certificate definition.
3. Click **Edit**.
4. Find **Delete certificate upon expiry**
5. Click the checkbox.
6. Click **Save** to save the changes or **Cancel** to discard them.

**To specify automatic remove duplicate certificates in Good Control:**

1. Navigate to **Certificates** > **Certificate Definitions** tab.
2. Find the desired certificate definition.
3. Click **Edit**.
4. Find **Remove duplicate certificate (Certificate that expires first will be removed)**
5. Click the checkbox.
6. Click **Save** to save the changes or **Cancel** to discard them.

## New or changed security or compliance policies

### Info: policy categories collapsed for speed on load

Security and compliance policies in Good Control's **Policy Sets** > *policy name* are categorized by function, such as authentication delegation, password policies, and other categories.

In the past, the **Policy Sets** > *policy name* page was loaded with the individual policies revealed in the browser.

Now, for improved speed, the policies are hidden on page load, revealing just the category headings.

To see the policies in a category, click the triangle to the left of a category heading.

## Change: allow password from 1 to 16 characters long

With the security policy for end-users' password length **Require at least *X* characters**, formerly you could allow a password from 1 to 14 characters long.

Now, the upper limit has been changed to allow a password from 1 to 16 characters long.

## Setting "No password" policy

**Note:** These policies were introduced with Good Control v3.0.50.70 in December, 2016, but were not enforced by the BlackBerry Dynamics SDK at that time. Now with this latest release of the BlackBerry Dynamics SDK, they are enforced.

The security policies below allow your end users to avoid having to set an application password when a BlackBerry Dynamics-based application is activated:

- Do not require user password for Android
- Do not require user password for iOS
- Do not require password for macOS
- Do not require password for Windows (UWP)

By default, these policies are not enabled.

If you enable one of these policies, the following message is displayed in the Good Control console:

> **Warning**
>
> Disabling the BlackBerry application password significantly reduces security of BlackBerry containers and Enterprise Network. Use of this mode is strongly discouraged.

### Security and design considerations for policy sets

When you plan to enable the "no password" policy, keep these considerations in mind:

1. Consider the security ramifications in your environment carefully. Enabling "no password" is only one of the design options available. Others include authentication delegation, enabling "no password" on only certain devices, device management, and more. You should consider enabling "no password" for only a certain select group of users whose devices are under tight control via device management profiles or other controls.
2. The policy can be enabled for either Android, or iOS, or both, in the same policy set. Thus, for example, in a single policy set, you can enable "no password" for Android while not enabling it for iOS.
3. Do not enable "no password" and authentication delegation in the same policy set. You should consider these two security policies as mutually exclusive. Enabling both policies will cause unnecessary switching among applications without any apparent benefit.
4. Enabling "no password" does not permit authentication in the background, because there is no authentication.

### Effects and behavior of enabling "no password"

- For behavior of enabling the "no password policy," see New: Support for "No Password" security policy
- For programming aspects on Android, see New: optional APIs for Android.
- For programming aspects on iOS, see New: optional APIs for iOS.

## New or changed properties

### New: allow new macOS devices

| | | |
|---|---|---|
| allow.new.MAC.device | Allow any new macOS device | Default: true<br>Global: yes<br>Restart: no |

### Info: server properties collapsed for speed on load, categorized by global vs local

Server properties in Good Control's **Servers** > **Server Properties** are now categorized on two separate tabs:

- Global Server Properties
- Local Server Properties

Each tab is further categorized by function:

- Certificate Management
- Communications
- Directory
- Duplicate Containers
- Email Templates
- GC Console Login
- Purge Inactive Containers
- Reporting
- Retention Data Policy
- Troubleshooting

In the past, the **Server Properties** page was loaded with the individual properties revealed in the browser.

Now, for improved speed, the properties are hidden on page load, revealing just the category headings.

To see the properties in a category, click the triangle to the left of a category heading.

### New: Exporting server properties

You can export all Good Control and Good Proxy server properties to a comma-separated value (CSV) for easier troubleshooting.

The name of the exported file is as follows:

**Server_Properties_Report_***DD-MM-YYYY***.csv**

**To export all properties, in Good Control:**

1. Navigate to **Export Server Properties**.
2. Click **Export**.

## GP property reference

| Property | Description | Editable? |
|---|---|---|
| eacp.command.service.nslookup.srv.ldap | Enables LDAP over TCP for Active Directory servers. Active Directory servers offer the LDAP service over the TCP protocol; therefore, clients find an LDAP server by querying DNS for a record of the form: _ldap._ tcp. DnsDomainName<br>• true = indicates that GP uses LDAP for nslookup of a given service hostname<br>• false = GP uses reverse DNS lookup directly, using the given service hostname<br>Default: false | yes, editable |
| gc.admin.name | Username of Good Control administrator<br>Default: none | not editable |
| gc.auth.token | Secret token to authenticate GC with GP<br>Default: none | not editable |
| gc.server.port | Port of GC server<br>Default: 443 | not editable |
| gc.server.uri | SOAP endpoint of GC server with which this GP should be registered.<br>Default: none, depends on name of server | not editable |
| gd.product.capability | GP server feature set used to compare with GC server feature set during GP registration to make sure that GC and GP are compatible.<br>Default: none. | not editable |
| gd.product.domain | Active Directory domain of the GP<br>Default: none. Set by installer. | not editable |
| gd.product.hostname | GP server name<br>Default: none. Set by installer. | not editable |
| gd.product.licensekey | GC and GP license keys as recorded in GDN<br>Default: none | not editable |

| Property | Description | Editable? |
|---|---|---|
| gd.product.loginkey | GP server login credentials to BlackBerry Dynamics NOC for uploading GP server logs<br>Default: none | not editable |
| gd.product.serialnum | GC and GP serial numbers as recorded in GDN<br>Default: none | not editable |
| gd.product.type | Differentiate between GC service and GP service.<br>• GPS = Good Proxy<br>• GMC = Good Control<br>Default: for GP, GPS | not editable |
| gd.product.version | Version number of this GP<br>Default: none | not editable |
| gd.security.keystore.alias | Alias for the GP's keystore<br>Default: good-proxy | yes, editable |
| gd.security.keystore.file | Location of GP keystore file<br>Default:*`GP_installation_ directory`* **`\jre\security\lib\cacerts`** | yes, editable |
| gd.security.rootcert.alias | Alias for the root certificate of the GP<br>Default: good-dynamics | yes, editable |
| gps.auth.token | Secret token to authenticate GP with GC | not editable |
| gps.directconnect.port | Port for Direct Connect configuration<br>Default: 17533 | not editable |
| gps.dns.server.ttl.ms | Time-to-live in milliseconds for the DNS server connections., i.e. time to wait for DNS server response.<br>Default: 1.8M milliseconds | yes, editable |
| gps.logfiles.days | Length of time to retain logfiles<br>Default: 10 days | yes, editable |
| gps.product.installdir | Installation directory for GP<br>Default: none. Set by installer | not editable |
| gps.product.registered | Flag for whether this GP has been registered with BlackBerry | not editable |

| Property | Description | Editable? |
|---|---|---|
| | Default: false | |
| gps.server.fqdn | Fully qualified domain name for this GP server<br>Default: none. Set by installer | not editable |
| gps.server.name | Bare hostname of this GP server<br>Default: none. Set by installer | not editable |
| gps.server.port | Non-secured port for this GP server<br>Default: 17080 | not editable |
| gps.server.secure.port | Secure port for this GP server<br>Default: 17443 | not editable |
| gps.service.name | Name of the GP service on Windows<br>Default: GPS | yes, editable |
| gps.status.request.frequency | Allowable frequency for /status request on this GP | yes, editable |
| gps.tcp.session.timeout | Length of time that a TCP connection can be inactive before it is closed.<br>**Important:** Do not alter this setting without direct consultation with BlackBerry.<br>Default: 1,800 seconds | yes, editable |
| gps.unalias.hostname | For DNS lookups of app servers, use either IP address or hostname<br>• true= GP uses reverse DNS lookup with IP address of app server<br>• false = GP uses app server hostname for lookup<br>Default: false | yes, editable |
| gwy.push.connection.timeout | Timeout of persistent connection to MDC server in BlackBerry Dynamics NOC for push notifications<br>Default: 45 seconds | yes, editable |
| gwy.push.port | Port of MDC server in BlackBerry Dynamics NOC<br>Default: 443 | not editable |
| gwy.push.prot | Protocol for communications<br>Default: 1 | not editable |

| Property | Description | Editable? |
|---|---|---|
| gwy.push.register | GP is registered with MDC server<br>true = GC is registered with BlackBerry Dynamics NOC<br>Default: true | not editable |
| gwy.push.request.timeout | Timeout of request to MDC server BlackBerry Dynamics NOC<br>Default: 20 seconds | yes, editable |
| gwy.push.secure | Use SSL for connection to MDC serverBlackBerry Dynamics NOC<br>Default: false | not editable |
| gwy.push.server | Name of MDC server in BlackBerry Dynamics NOC<br>Default: gdmdc.good.com | not editable |
| gwy.push.socket.timeout | Timeout in establishing socket connection to MDC server BlackBerry Dynamics NOC<br>Default: 45 seconds | yes, editable |
| health.check.enabled | Whether to perform additional checks on GP health<br>Default: true | yes, editable |
| health.check.interval | How often to check GP health<br>Default: 3.6M milliseconds (1 hour) | yes, editable |
| log.upload.date.name.format | Date format for timestamp of GP logfile names<br>Default: yyyy-MM-dd | not editable |
| log.upload.dir | Path to directory on server where logs are stored<br>Default: none. Set by installer. | not editable |
| log.upload.url | URL on this GP where logfiles can be uploaded<br>Default: none | not editable |
| mdc.server.name | Name of MDC server in BlackBerry Dynamics NOC<br>Default: gdmdc.good.com | not editable |
| mdc.server.port | Port of MDC server in BlackBerry Dynamics NOC<br>Default: 443 | not editable |

| Property | Description | Editable? |
|---|---|---|
| proxy.auth.domain | Active Directory domain for authentication login to external Web proxy server<br>Default: none | yes, editable |
| proxy.auth.password | Password of username for authenticating to external Web proxy server<br>Default: none | yes, editable |
| proxy.auth.username | User name for connecting to external Web proxy server<br>Default: none | yes, editable |
| proxy.https.host | Name of external Web proxy server<br>Default: none | yes, editable |
| proxy.https.port | Port number for HTTPS connection to external Web proxy server<br>Default: none | yes, editable |
| proxy.urls | URLs that must be proxied<br>Default: none | yes, editable |
| proxy.use | Use an external Web proxy server<br>Default: false | yes, editable |
| relay.gps.key | Key to access relay server in BlackBerry Dynamics NOC<br>Default: none | not editable |
| relay.server.name | Name of relay server in BlackBerry Dynamics NOC<br>Default: gdrelay.good.com | not editable |
| relay.server.port | Port number of relay server in BlackBerry Dynamics NOC<br>Default: 443 | not editable |

## Logging property reference

| Property | Description | Good Control | Good Proxy |
|---|---|---|---|
| Maximum server log file size | Allowable values: from 100 KB to 1 GB | Default: 256 MB | Default: 256 MB |
| Maximum server log file age | In days | Default: 10 days | Default: 10 days |

| Property | Description | Good Control | Good Proxy |
|---|---|---|---|
| Compress server log files | Allowable values: true\|false | Default: on | Default: on |
| Server logging level | Allowable values: Info\|Debug | Default: Info | Default: Info |

## New: automatic removal of duplicate containers at activation

In addition to other properties for scheduling the removal of duplicate or inactive containers, a new property is available to automatically remove duplicate containers when a new version of an application is activated:

- Property name: Automatically remove older duplicate containers on same device for user after provisioning
- Default:true

### Behavior and precedence

By default, this property is enabled, both for upgrades and new installations, and takes precedence over the other properties to purge duplicate containers.

# Good Control Web Services

## Change: UpdateConnectionProfileRules syntax and usage

With the **UpdateConnectionProfileRules** operation, you can make changes to your defined connectivity profiles.

The change in this operation is that it now allows certain action prefixes in the **domainName** element to add domains to the various sections of a profile, regardless of the section specified with the **actionType** (such as **ADD**) or the specific type (such as **ALLOWED_DOMAINS**).

### Syntax

```
<urn:UpdateConnectionProfileRulesRequest>
  <urn:connectionProfileId>profileId</urn:connectionProfileId>
  <urn:connectionProfileRules>
  <urn:domainServers>
  <urn:type>type_of_connectivityProfile</urn:type>
  <urn:domains>
    <urn:domainName>actionprefix.domain</urn:domainName>
    <urn:primaryRouteName>first_or_second</urn:primaryRouteName>
    <urn:actionType>action</urn:actionType>
  </urn:domains>
  </urn:domainServers>
  </urn:connectionProfileRules>
</urn:UpdateConnectionProfileRulesRequest>
```

where:

| Element | Description and allowable values |
|---|---|
| **connectionProfileId** | Profile IDs can be retrieved with the SOAP API **GetAllConnectionProfiles**. |
| **type** | Required.<br>Connectivity profile type. Allowable values are shown below.<br>**Note:** This field is always overrdden by the value of the **domainName** field (see below) and its *actionrprefix*.<br>• **ALLOWED_DOMAINS**<br>• **DEFAULT_DOMAINS**<br>• **IP_ADDRESS_RANGES**<br>• **ADDITIONAL_SERVERS** |
| **domainName** | Required.<br>The domain you are adding or deleting, in the form: *actionprefix.domain*.<br>• *domain* is the Internet domain, like **.wikipedia.org** or **.ca.gov**. Must begin with "**.**" |

| Element | Description and allowable values |
|---|---|
| | The optional *actionprefix* can be any of the following. The *actionprefix* always overrides any value specified for **`type`** (see above).<br>• *`domain`*<br>With no prefix, specified *`domain`*is added to the "Additional Servers" section of the connectivity profile. Example: **`.turkeywing.org`**<br>• **`+.`***`domain`*<br>With **+**, specified *`domain`*is added to the "Default Domains" section. Example: **`+.duckfoot.org`**<br>• **`*.`***`domain`*<br>With **`*`**, specified *`domain`*is added to the "Allowed Domains" section. Example: **`*.bigfoot.com`**<br>To include the * wildcard in the *`domain`*, use two asterisks, like this: **`<urn:domainName>*.*.somedomain.com</urn:domainName>`**<br>• **`//.`***`ip_address`*<br>With //, specified *ip_address* is added to the "IP Address Range" section. Example: **`//.100.1.1.100`**<br>The *actionprefix* overrides the action type. |
| **`actionType`** | Required.<br>Type of the update:<br>• ADD<br>• DELETE |

#### Example: add a domain to Additional Servers with null action prefix and action type ADD

In this example, the action type **`ADD`** is used to add the domain **`turkeywing.org`** to the "Additional Servers" section of the connectivity profile

> **Note:** Even though the required field type specifies **`ALLOWED_DOMAINS`**, the null action prefix in the domains field overrides type and adds **`turkeywing.org`** to "Additional Servers".

```
<urn:UpdateConnectionProfileRulesRequest>
  <urn:connectionProfileId>2887288105</urn:connectionProfileId>
  <urn:connectionProfileRules>
     <urn:domainServers>
        <urn:type>ALLOWED_DOMAINS</urn:type>
        <!-- Even if ALLOWED_DOMAINS is specified in type,
        the action prefix in domainName field
        determines where domain will be added.
        In this case, the action prefix is null,
        so the update is to ADDITIONAL_SERVERS. -->
        <urn:domains>
           <!-- action prfix is null, so change is to ADDITIONAL_SERVERS -->
           <urn:domainName>turkeywing.org</urn:domainName>
        <urn:primaryRouteName>second</urn:primaryRouteName>
        <urn:actionType>ADD</urn:actionType>
      </urn:domains>
    </urn:domainServers>
  </urn:connectionProfileRules>
</urn:UpdateConnectionProfileRulesRequest>
```

### Example: add a domain to Default Domains with + action prefix and action type ADD

In this example, the action type **`ADD`** is used to add the domain **`duckfoot.com`** to the "Default Domains" section of the connectivity profile

> **Note:** Even though the required field type specifies **`ALLOWED_DOMAINS`**, the **+** action prefix in the domains field overrides type and adds **`turkeywing.org`** to "Default Domains".

```
<urn:UpdateConnectionProfileRulesRequest>
  <urn:connectionProfileId>2887288105</urn:connectionProfileId>
  <urn:connectionProfileRules>
    <urn:domainServers>
      <urn:type>ALLOWED_DOMAINS</urn:type>
      <!-- Even if ALLOWED_DOMAINS is specified in type,
      the action prefix in domainName field
      determines where domain will be added.
      In this case, the action prefix is +
      so the update is to DEFAULT_DOMAINS. -->
      <urn:domains>
        <!-- action prfix is +, so change is to DEFAULT_DOMAINS -->
        <urn:domainName>Example: add a domain to Additional Servers with null actionprefix and action type ADD
```

### Example: delete a domain from Additional Servers

This example request deletes the domain **`.carwash.gov`** from the "Additional Servers" section, using **`actionType`** with a value of **`DELETE`**.

```
<urn:UpdateConnectionProfileRulesRequest>
<urn:connectionProfileId>2887288105</urn:connectionProfileId>
  <urn:connectionProfileRules>
  <urn:domainServers>
  <urn:type>ADDITIONAL_SERVERS</urn:type>
    <urn:domains>
      <!-- The id of the domain carwash.gov is stored in the t-gc-domains
      or can be retrieved with the fetchDomains SOAP API call. -->
      <urn:id>10125</urn:id>
      <urn:actionType>DELETE</urn:actionType>
    </urn:domains>
  </urn:domainServers>
  </urn:connectionProfileRules>
</urn:UpdateConnectionProfileRulesRequest>
```

## App wrapping requests now deprecated

Because application wrapping is no longer supported, the following operations have been deprecated and will be removed from gc.wsdl in a future release:

- GetWrappingProperty
- SetWrappingProperty
- GetWrappingEngineVersion
- UpdateWrappingEngineVersion
- WrapApp

## Global changes: new product names and abbreviations

This release includes major changes in naming to align former "Good Technology" products with BlackBerry.

| Old name | New Name | Old abbreviation | New abbreviation |
|---|---|---|---|
| Good Control | None. This name refers to the standalone server that is not integrated with the BlackBerry Unified Endpoint Manager (BlackBerry UEM).<br>When the server is integrated with the BlackBerry Unified Endpoint Manager, it is called BlackBerry Control. | GC | None |
| Good Proxy | None. This name refers to the standalone server that is not integrated with the BlackBerry Unified Endpoint Manager (BlackBerry UEM).<br>When the server is integrated with the BlackBerry Unified Endpoint Manager, it is called BlackBerry Proxy. | GP | None |
| Good Dynamics | BlackBerry Dynamics | GD | None. We no longer abbreviate the name of this product line. |
| Good Dynamics Software Development Kit | BlackBerry Dynamics Software Development Kit | GD SDK | BlackBerry Dynamics SDK |
| Good Enterprise Mobility Server | BlackBerry Enterprise Mobility Server | GEMS | None. We no longer abbreviate the name of this product line. |

## Add service account for role via GC console

A service account for Good Control is an account that has a limited defined function associated with a GC role. For separation of concern, service accounts allow you to divide work according to these roles that have only a limited set of permissions.

You can create a service account for any predefined or custom role by way of the GC user interface.

**To create a service account for a role in Good Control**:

1. Navigate to **Administrators** > **Roles**.
2. In the list of roles, click the role to which you want to add a service account.
3. Click **Add Service Account**.
4. Enter the name of the service account.
5. Click **Add** to add the service account or **Cancel** to discard your entries.
6. Securely record the password for this service account so it can be given to the human beings who will use the service account.

   **Note:** The password is displayed in the user interface only once; it is never displayed again in the user interface.

7. Click **OK** to confirm.

## Enhancements to Manage User page: container lock status and auth delegates

The **Manage User** page now shows the following information, if applicable:

- Status of lock on containers
- Authentication delegates of containers

## Searching for members of administrative roles

Good Control's **Administrators** > **Edit Role** page now includes a filter textbox to search for the names of members in the role.

**To search administrative roles by member name:**

1. Navigate to **Administrators** > **click a role** > **Edit Role** > **Members** tab.
2. In the **Name filter** textbox, enter a string to search for.
3. Hit return.

The matching member names are displayed.

## Searching for users in local AD domain groups to import to GC

If the GC server property **Enable to allow domain local group to be included in search** is enabled (which is not enabled by default), in the GC **Users and Groups** page, you can search for usernames in AD domain groups that you can then import into the system.

**Note:** AD comes with the Builtin container that includes default local groups, including Users. The Users group in the Builtin container *cannot* be searched.

## Enhancements to GP diagnostics page

Good Control's diagnostic page for its associated Good Proxy servers has been enhanced to display a color-coded status for the server and the following details.

If any of these metrics indicates a problem, the color of the status is yellow. If all metrics indicate problems, the color of the status is red.

| YELLOW<br>if all three of these indicators is failing... | RED status<br>if any of the following is failing... |
|---|---|
| • GC Connectivity<br>• Memory usage<br>• CPU usage | • NOC Last Connected Time<br>• Active sessions count<br>• or if all indicators listed here are failing |

## Good Control health report

Good Control writes a report about its health to the file **`c:\good\gc_health_report.data`** based on a frequency you can control via the system property **Frequency in seconds that job to generate GC health report will run**. Default is every 86400 seconds, or 24 hours.

For every update, the file is overwritten, not appended.

Below is a sample of the data available in the report, which consists of name/value pairs in JSON format. The value of the status **`code`** can be one of OK, WARN, ERROR, INFO, UNKNOWN.

```
"lastReportedTime":1474319313201,
"healthy":true,
"status":
{
        "statuses":
        {
                "MemoryMonitor":
                {
                        "code":"OK",
                        "desc":"769.64MB free,
                        966.00MB total."
                },
                "DBMonitor":
                {
                        "code":"OK",
                        "desc":"Database connection is ok"
                },
                "CPUMonitor":
                {
                        "code":"OK",
                        "desc":"CPU Load: 0.00"
                },
                "HttpConnectivityMonitor":
                {
                        "code":"OK",
                        "desc":"Http connectivity succeeded to URL <URL for GD NOC>
                },
                "DBJobQueueMonitor":
```

```
                {
                        "code":"INFO",
                        "desc":"JobQ Size: 31,
                        Average JobQ Size: 31,
                        Max JobQ Size: 31"
                }
        },
"code":"OK"
}
```

## /status URLs display status of Good Control and Good Proxy

You can navigate to the URL **https://** *fully_qualified_domain_name_of_good_control_host* **/gc/status** to see the general status of the server.

The response looks similar to the following:

```
{"paused":false,"name":"BlackBerry UEM - Good Control Service","ha":{"scheme":"active-
standby","state":"active"},"health":
{"score":100},"serviceID":"GoodControl","version":"2.4.55.8997","connections":
[{"connected":true,"type":"MDC","dest":"https://someserver.company.com/GNP1.0"},
{"connected":true,"type":"PUSHGW","dest":"https://someserver.company.com:443/GDES1.0"},
{"connected":true,"type":"DB","dest":"jdbc:sqlserver://someserver.company.com:1433;databaseName=anu1;sendString
ParametersAsUnicode=false","properties":[{"name":"dialect","value":"com.good.db.util.SQLAddNVarCharDialect"},
{"name":"driver","value":"com.microsoft.sqlserver.jdbc.SQLServerDriver"}]}]}
```

This URL is not access-protected but it is governed by the property **Allowed frequency of /status**, which permits access only within a certain frequency. See New or changed properties for more details.

### Good Proxy /status URL

For GP, the URL is **https://** *fully_qualified_domain_name_of_good_proxy_host_and_port* **/status** , which responds similarly to the following:

```
{"paused":false,"name":"BlackBerry UEM - Good Proxy Server","ha":{"scheme":"active-
standby","state":"active"},"health":
{"score":100},"serviceID":"GoodProxy","version":"2.4.55.6204","connections":
[{"connected":true,"type":"GC","dest":"someserver.company.com","properties":
[{"name":"lastConnectedTimeStamp","value":"2016-09-30T17:28:09.997-04:00"}]},
{"connected":true,"type":"session","properties":[{"name":"maxSession","value":"15000"},
{"name":"activeSession","value":"0"},{"name":"totalSession","value":"0"},{"name":"idleSession","value":"0"},
{"name":"noOfdirectConnectConnections","value":"0"},{"name":"noOfrelayConnections","value":"0"},
{"name":"dest","value":"someserver.company.com"},{"name":"port","value":"443"}]},
{"connected":true,"type":"MDC","dest":"someserver.company.com","properties":
[{"name":"lastConnectedTimeStamp","value":"2016-09-30T17:27:33.700-04:00"},{"name":"port","value":"443"}]}]}
```

## Progress indicator for cluster-wide logfile upload

The status of log file uploading, shown on Good Control's **Upload Server Logs** page, now shows the status of log uploads for all the GC servers in the entire cluster. This same status is also viewable on any server in the cluster.

# Editing connectivity profiles by CSV export/import

Connectivity profiles are Good Control's mechanism for defining networking characteristics that are applied to groups of users via policy sets. There are two types of connectivity profiles: the base connectivity profile and override profiles that modify the base profile. For more details about connectivity profiles and how to create them, see Connectivity Profiles for Clients

After you have created a connectivity profile, you can export it in comma-separated value (CSV) format. You can add new rows, edit existing values, or delete values. This is especially useful for adding large numbers of Internet domains to a profile.

## General steps for export/import: navigation

**To export and import a profile, in Good Control:**

1. Create the connectivity profile with the UI at **Connectivity Profile**.
2. Navigate to **Reporting** > **Export Connectivity Profile**, and export the desired profile to CSV.
3. Edit the exported profile as desired. See A look at an exported profile: which fields can be edited?
4. Navigate to **Reporting** > **Export Connectivity Profile**, import the changed profile within 24 hours.

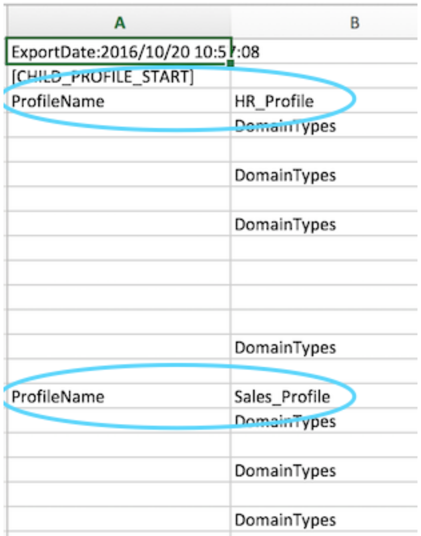
## Guidelines on editing via CSV

Keep these points in mind when editing connectivity profiles via export/import of CSV.

- Before you make large changes to your profiles, as a safeguard, consider backing up the Good Control database table `T_GC_DOMAINS`, in case you want to revert.
- You cannot create a new connectivity profile by importing. You must always first create a profile using Good Control's **Connectivity Profile** menu item and then export to CSV to edit the data. See the advice in General steps for export/import: navigation .
- Likewise, you cannot delete a profile via export/import. You must use the Good Control **Connectivity Profile** menu item.
- Be sure to plan your changes with some care, that you change only those cells described in A look at an exported profile: which fields can be edited? and that you make certain exactly what you want to change.
- The filename of an exported CSV has a certain prefix and a date stamp. Do *not* change the filename in any way.
- The extension of a file you import must be `.csv`. The system rejects any other extension.
- There is a time limit of 24 hours on reimporting the edited profile to avoid conflicts that might occur if the exported profile is edited in the UI before it is imported again.

## A look at an exported profile: which fields can be edited?

When you export a base connectivity profile, the CSV contains only that base connectivity profile, because there is always only one base connectivity profile. With override profiles, the exported CSV files contains *all* of the override profiles you have defined, not just the one you might want to edit. All profiles are identified by name in the second column of the CSV. For example, assume you have two override profiles "HR Profile" and "Sales Profile". In the CSV these are shown like so:



| A | B |
|---|---|
| ExportDate:2016/10/20 10:57:08 | |
| [CHILD_PROFILE_START] | |
| ProfileName | HR_Profile |
| | DomainTypes |
| | |
| | DomainTypes |
| | |
| | DomainTypes |
| | |
| | |
| | |
| | |
| | DomainTypes |
| | |
| ProfileName | Sales_Profile |
| | DomainTypes |
| | |
| | DomainTypes |
| | |
| | DomainTypes |

The exported CSV file contains tabs (worksheets), labels, and identifiers (numbers) that must not be touched.

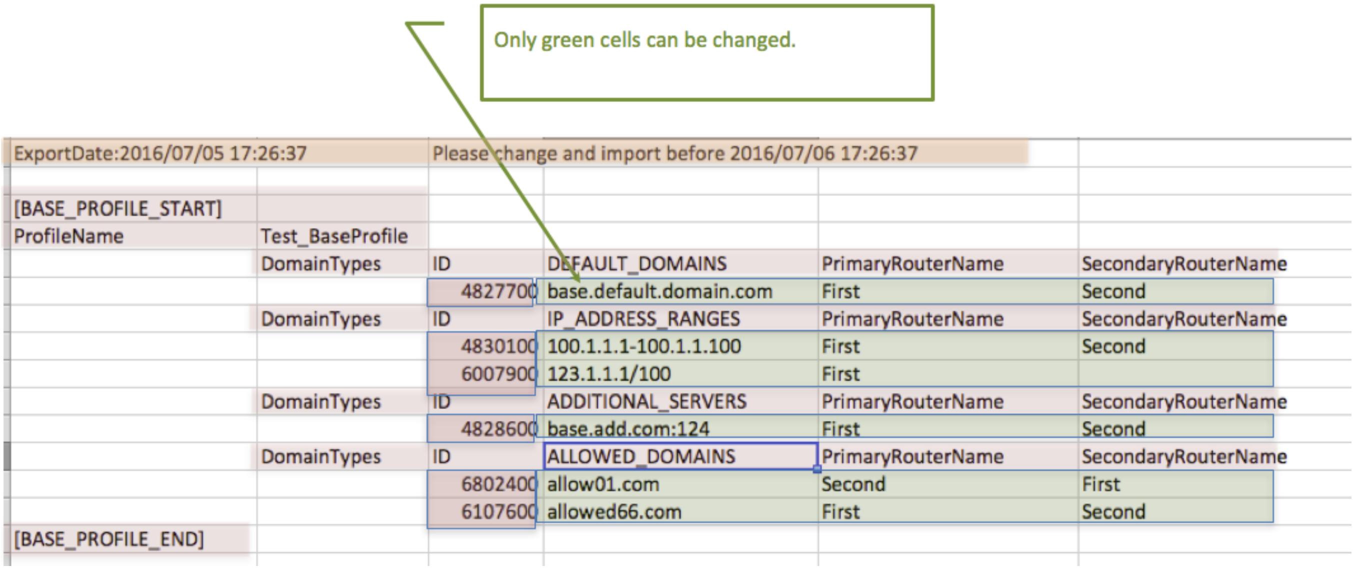
You can edit records only on the **Infrastructure** tab, not the **App Servers** tab.

All changes you make must be in one of the identified blocks:

- ALLOWED_DOMAINS
- DEFAULT_DOMAINS
- ADDITIONAL_SERVERS
- IP_ADDRESS_RANGES

**Note:** You cannot set or change "Route All" via CSV.

As shown in the annotated view below, only the **Infrastructure** tab's cells highlighted in green should be changed. Be careful: because this is CSV format, no cells are protected or locked.



Only green cells can be changed.

| | | | | | |
|---|---|---|---|---|---|
| ExportDate:2016/07/05 17:26:37 | | Please change and import before 2016/07/06 17:26:37 | | | |
| | | | | | |
| [BASE_PROFILE_START] | | | | | |
| ProfileName | Test_BaseProfile | | | | |
| | DomainTypes | ID | DEFAULT_DOMAINS | PrimaryRouterName | SecondaryRouterName |
| | | 4827700 | base.default.domain.com | First | Second |
| | DomainTypes | ID | IP_ADDRESS_RANGES | PrimaryRouterName | SecondaryRouterName |
| | | 4830100 | 100.1.1.1-100.1.1.100 | First | Second |
| | | 6007900 | 123.1.1.1/100 | First | |
| | DomainTypes | ID | ADDITIONAL_SERVERS | PrimaryRouterName | SecondaryRouterName |
| | | 4828600 | base.add.com:124 | First | Second |
| | DomainTypes | ID | ALLOWED_DOMAINS | PrimaryRouterName | SecondaryRouterName |
| | | 6802400 | allow01.com | Second | First |
| | | 6107600 | allowed66.com | First | Second |
| [BASE_PROFILE_END] | | | | | |

- To change existing data, edit the cell that has the data you want to change.
- To delete data, remove the data from the green cells, but do not remove the identifiers.
- To add new data, in the appropriate block, insert new rows, without identifiers. You do not need to add identifiers to new data rows.

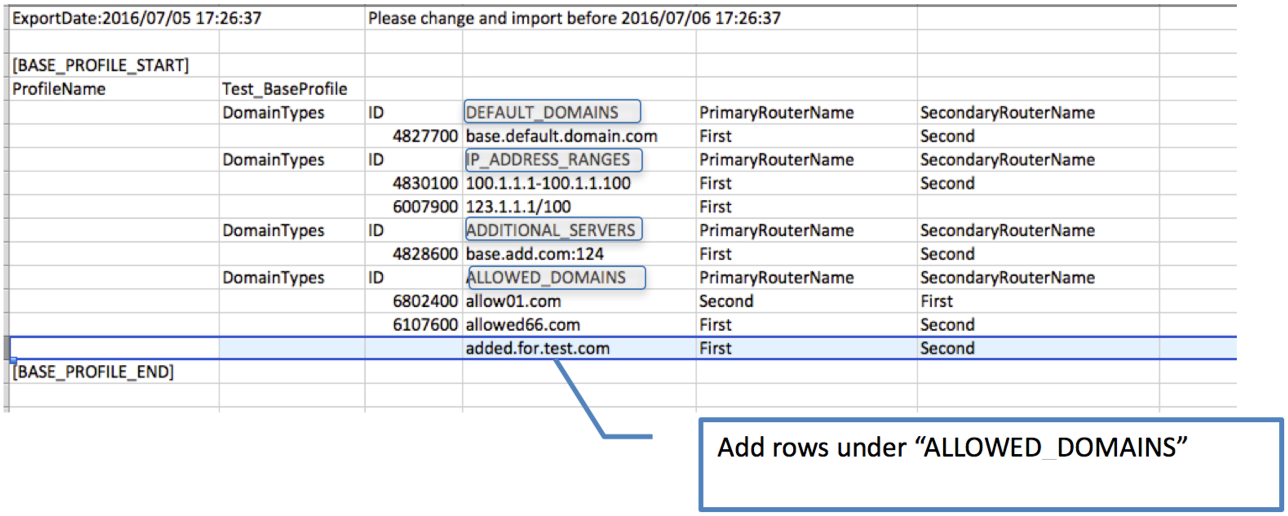
## Examples

Here are some examples of common usage.

### Adding many allowed domains

Suppose you have hundreds of different Internet domains that your users must be allowed to access. Configuring all these domains via the UI might be tedious. Here are general steps for combining the UI and export/import to create the larger profile.

1. In the GC UI at **Connectivity Profiles**, create two connectivity profiles: a base connectivity profile and an override profile.
2. In the override profile, under **Allowed Domains**, add one of the Internet domains and all its necessary fields, such as first or secondary GP cluster.
3. In the GC UI at **Reporting** > **Export Connectivity Profiles**, click the name of the override profile.
4. Click **Export**.
5. Open the downloaded spreadsheet.
6. Find the name of the profile in the CSV file.
7. In the block for **ALLOWED_DOMAINS**, insert a row for each additional allowed domain, its primary route, and itssecondary route. Do not include any ID numbers. Example:



| | | | | | |
|---|---|---|---|---|---|
| ExportDate:2016/07/05 17:26:37 | | Please change and import before 2016/07/06 17:26:37 | | | |
| | | | | | |
| [BASE_PROFILE_START] | | | | | |
| ProfileName | Test_BaseProfile | | | | |
| | DomainTypes | ID | DEFAULT_DOMAINS | PrimaryRouterName | SecondaryRouterName |
| | | 4827700 | base.default.domain.com | First | Second |
| | DomainTypes | ID | IP_ADDRESS_RANGES | PrimaryRouterName | SecondaryRouterName |
| | | 4830100 | 100.1.1.1-100.1.1.100 | First | Second |
| | | 6007900 | 123.1.1.1/100 | First | |
| | DomainTypes | ID | ADDITIONAL_SERVERS | PrimaryRouterName | SecondaryRouterName |
| | | 4828600 | base.add.com:124 | First | Second |
| | DomainTypes | ID | ALLOWED_DOMAINS | PrimaryRouterName | SecondaryRouterName |
| | | 6802400 | allow01.com | Second | First |
| | | 6107600 | allowed66.com | First | Second |
| | | | added.for.test.com | First | Second |
| [BASE_PROFILE_END] | | | | | |

Add rows under “ALLOWED_DOMAINS”

8. In the GC UI at **Reporting** > **Export Connectivity Profiles**, click the name of the override profile.
9. Click **Import**, select the edited spreadsheet, and follow the leading prompts to complete the import.

### Cloning a profile from one GC to another

Building on the example above, suppose you have the same set of hundreds of allowed domains that need to be defined in connectivity profiles in all your Good Control servers. Here are the general steps to clone those allowed domains from one server to another.

1. Follow the steps described in Adding many allowed domains to create the override profile you want to clone on another server.
2. Export again the override profile that includes all the allowed domains.
3. On the second server, follow steps 1 through 5 from the first example to create a skeletal profile and export it.
4. From the first downloaded spreadsheet for the first server, from the correct override profile, copy to your clipboard the domain, primary route, and secondary route for all of the rows for allowed domains. If you like, you can paste these fields to a temporary spreadsheet or a temporary set of rows.
5. Do not copy of the ID numbers from those rows, only the data fields.
6. Open the second downloaded spreadsheet for the second server.
7. Under the block for **ALLOWED_DOMAINS**, paste the ID-less data fields from your clipboard or copy them from the temporary set of rows.
8. Save the second spreadsheet.
9. In the Good Control UI for the second server, navigate to **Reporting** > **Export Connectivity Profile**, open the affected profile, and click **Import** to upload the second spreadsheet.

# New or changed security or compliance policies

## Setting "No password" policy

The security policies below allow your end users to avoid having to set an application password when a BlackBerry Dynamics-based application is activated:

- Do not require user password for Android
- Do not require user password for iOS

By default, these policies are not enabled.

If you enable one of these policies, the following message is displayed in the Good Control console:

> **Warning**
>
> Disabling the BlackBerry application password significantly reduces security of BlackBerry containers and Enterprise Network. Use of this mode is strongly discouraged.

### Effects of enabling "no password"

With a BlackBerry Dynamics application that is protected by security policy to require a password, if the IT administrator changes the security policy to "No Password":

- The user is shown an informational screen stating that a password is no longer required for the application.
- The user is then in "No Password mode" and is never prompted for password again.

Conversely, if the user is in "No Password mode" but the IT administrator changes the security policy to require a password:

- The user is prompted to set a password.
- The user is shown an informational screen stating that a password is now required for the application.

## New: Prevent end-user from enabling detailed logging

To increase supportability of the system, the security policy **Prevent users from turning on detailed logging** controls whether or not the end user of a GD based application can enable detailed logging on the client.

Default: Enabled. Detailed logging is not allowed.

Usage notes:

- When this policy is set to prevent the end user, the end user is not shown any control in a BlackBerry Dynamics application to turn on detailed logging.
- If the policy **Enable detailed logging for GD apps** is enabled, the policy **Prevent users from turning on detailed logging** is grayed out, not settable.

## New: Enable detailed logging for BlackBerry Dynamics apps by policy set/by user group

Detailed logging is controlled by the security policy **Enable detailed logging for GD apps**.

Default: Not enabled.

You can set this policy in those policy sets that you apply to specific groups of end users you want to allow detailed logging.

Usage notes:

- Disabled: When this policy is disabled, the related setting for **Detailed Logging** for a particular user under **Manage Users** can still be used to set the policy for individual users.
- Enabled: When this policy is enabled in a policy set applied to a specific group of users, the related function on the **Manage User** page for a particular user is grayed out, and not settable.

### New: Compliance rule for Android OS versions allows alphanumeric characters

Good Control now allows an Android operating system version that includes both letters and numbers.

### Change to data leak prevention policies

Formerly the Data Leak Prevention (DLP) security policies were presented as independent settings in the Good Control console.

The console has been changed to show that the policies are dependent:

- **Prevent copy from GD apps into non-GD apps** is the primary policy. If it is enabled, then:
  - **Prevent copy from non-GD into GD apps** is a secondary policy that becomes visible.

### Change: Number of disallowed previous passwords increased to 12

The security policy **Disallow** *X* **previously used passwords** has been changed from 1 to 8 to 1 to 12.

## New or changed properties

### New: Allowing search of AD domain local group

The server property **Enable to allow domain local group to be included in search** allows you to search for users in local domain groups.

Default: Disabled.

Global

### New: Frequency of GC on-disk health report

The server property **Frequency in seconds that job to generate GC health report will run** determines how often the report is generated. Requests more frequent than this value will be returned HTTP code 503.

Default: 86400 seconds, or 24 hours.

Global

### New: Allowed frequency of /gc/status URL

The **Minimum time interval between two status requests in milliseconds** property controls how frequently the **`/gc/status`** URL can be accessed. Any accesses more frequent than this value are rejected with HTTP return code 503. See /status URLs display status of Good Control and Good Proxy for more information.

Default: 1,000 milliseconds

Global

## New: Set domain value for Good Control console login

The GC console login page has the **Domain** field, which is used as part of authentication for logging in.

With the property **Domain: Pre-populate domain field**, you can "hard code" the value for the Domain field so that your users and administrators do not have to remember it.

Default: none

Local

## New: Properties for "unlock email"

A new grouping of properties **Email Templates** has been added, and the properties for forgotten passwords formerly listed under **Misc** have been moved here.

For ease of use, the properties related to unlock emails were formerly settable in Good Proxy, but now have been made editable in Good Control.

Email Templates

| Property | Description | Default, Global, Restart |
|---|---|---|
| Forgot password email on/off | Enable sending of forgotten password email | Default: true<br>Global: yes<br>Restart: no |
| Forgot password email body | Body of the mail for forgotten passwords | • Default: see text below.<br>• Global: yes<br>• Restart: no<br>• Text:<br>Good Control received a request to reset your password. If you did not make this request, ignore this email.<br>To reset your password, follow the instructions at this link, which expires after <%DEFINED_EXPIRY_TIME_FOR_THIS_GC%>.<br><%GC_REG_URL%> .<br>Thank you,<br>Good Control |
| Forgot password email sender | Email address for sender of forgotten email | Default: BlackBerry Mobile Administrator<br>Global: yes<br>Restart: no |

| Property | Description | Default, Global, Restart |
|---|---|---|
| Forgot password email subject | Subject line of the forgotten password email | Default: Password from Good Control<br>Global: yes<br>Restart: no |
| Unlock email | Enable/disable sending of "unlock emails" | Default: Enabled<br>Global: yes<br>Restart: no |
| Unlock email body | Active Directory domain specified during installation | Default: Dear <%HELPDESK_REF%>,<br>You can unlock your BlackBerry Dynamics Mobile Application <%APPLICATION_NAME%> provided by your company.<br>This email contains your UNLOCK ACCESS KEY and instructions for unlocking the mobile application.<br>Enter the following information when prompted (not case sensitive):<br>EMAIL ADDRESS: <%EMAIL_ADDRESS%><br>UNLOCK ACCESS KEY: <%PIN_FULL%><br>Your Unlock Access Key expires: <%EXPIRY%><br>For further assistance please contact your IT department.<br>Global: yes<br>Restart: no |
| Unlock email sender | Username of the GC administrator specified at installation | Default: BlackBerry Mobile Administrator<br>Global: yes<br>Restart: no |
| Unlock email subject | Subject line of unlock email | Default: Unlock BlackBerry Dynamics Mobile Application <%APPLICATION NAME%><br>Global: yes<br>Restart: yes |

## Change: Separate properties for duplicate containers and purge inactive containers

The server properties recently introduced for managing duplicate and inactive containers on devices have been divided into separate sets of properties for each function.

### Duplicate containers

| Property | Default, Global, Restart |
|---|---|
| Enable job to automatically remove duplicate containers (on/off)<br>How often to run the job to automatically remove inactive containers. In seconds. | Default: On<br>Global: yes<br>Restart: no |

| Property | Default, Global, Restart |
|---|---|
| Frequency in seconds that job to remove duplicate containers will run. | Default: 86400<br>Global: yes<br>Restart: no |
| Inactivity timeout in seconds before duplicate container is deleted. | Default: 259200<br>Global: yes<br>Restart: no |
| Maximum number of containers to remove in a single job | Default: 100<br>Global: yes<br>Restart: no |

Purge inactive containers

| Property | Default, Global, Restart |
|---|---|
| Container inactivity interval in seconds. | Default: 7776000<br>Global: yes<br>Restart: no |
| Enable job to automatically remove inactive containers (on/off) | Default: Off<br>Global: yes<br>Restart: no |
| Frequency in seconds that job to remove inactive containers will run. | Default: 86400<br>Global: yes<br>Restart: no |
| Interval in seconds that container inactivity times will be adjusted forward by the downtime to allow for reconnection. | Default: 86400<br>Global: yes<br>Restart: no |
| Maximum number of containers to remove in a single job | Default: 100<br>Global: yes<br>Restart: no |

## Miscellaneous changes

### Settings page displays GC version number

The GC **Settings** page now includes the version number of GC.

### Log file includes war version

The GC log file now contains the name of the WAR archive used by the server.

### No more hyphens in activation emails

The access key in activation email to new users does not have hyphens any more, only the key itself.

### Activation email includes timestamp

The activation email to new users now includes a timestamp.

### Audit log processing performance enhancements

Audit log processing has been optimized for improved performance.

### Support for IP v6

Good Control, Good Proxy, and the BlackBerry Dynamics SDK all support Internet Protocol (IP) v6 addresses (AAAA records).

### Info: Certificates now signed with SHA-256

Certificates issued by the BlackBerry Dynamics Certificate Authority (GDCA) for GC and Good Proxy are now signed with the SHA-256 algorithm.

For upgrades, new certificates are issued encrypted with SHA-256 to replace the older certificates encrypted with SHA-1.

### Switching DB authentication in GC from Windows auth to SQL auth

GC supports two kinds of authentication to its database:

- Windows authentication
- SQL authentication

Here are the steps to switch from Windows authentication to SQL authentication. First, you create a new SQL login. Then, you change the necessary GC properties to use that new SQL login.

#### In the SQL database

1. Create a new login account that has the **SQL Server auth** permission. Example: **`gc_sql_auth`**.
2. In the SQL database, give this new login the **`db_owner`** permission for the GC database.
3. Make a note of this new login's username and password, which you will use in later steps.

#### In Good Control

1. Make sure you have backed up the database and the **c:\good** directory. See *BlackBerry Dynamics Server Backup and Restore*.
2. Stop the GC service. See Stopping the GC and GP Servers .
3. Edit the file **`C:\good\gc.properties`**.

4. Find the **`db.connection.url`** property.
5. Remove the **`integrated security\=true`** string. If this string is at the end of the **`db.connection.url`** property, also remove the trailing semi-colon (**`;`**).
6. Find the **`db.connection.user`** property.
7. Change its value to the username of the SQL login that you created above. Example: **`gc_sql_auth`**.
8. Save the **`C:\good\gc.properties`** file.

   Now record the obfuscated database password of the new SQL login:

9. Open a command window as administrator.
10. Change to *`gc_installation_directory`***`\tools`** directory.
11. Enter the following command, where *sql_login_password* is the password for the new SQL login you created above.

    ```
    changepwd sql_login_password
    ```

12. Oprional: check the update time recorded in the **`gc_secure`** file.
13. Restart the GC. See Starting the GC and GP Servers

# Getting started

## Understanding our Terminology

The following terminology is used this help and the Good Control console.

| Term | Definition |
|---|---|
| Access Key | A one-time 15 character code required to activate a GD application for the first time. When an access key is generated for a user, GC sends the key to the email address it has on file for the user. If user self-service is enabled, the access key is also available to the user in the self-service portal. |
| Application Group (or Group) | A collection of users to which the same base application permissions are applied. A user can belong to multiple groups. |
| Application Policy | A collection of application-specific rules, uploaded in XML format. Each GD application can have its own rules, which can be configured for each policy set. |
| Application Service (or Service) | Shared functions provided by a GD mobile application or server-based application that can be used by other GD applications. Developers can refer to service definitions through the GC console. |
| Application<br>(or GD App) | A specific native application developed with the GD SDK and assigned an Application ID. An application can have multiple versions and can offer services. |
| Authentication Delegation and Delegate | The capability for one GD-SDK-based application to delegate its user authentication to another GD-SDK-based application running on the same device.<br>The authentication delegate is the application that assumes responsibility for the authentication. This application can be a GD application or Good for Enterprise. One authenticator can be |

| Term | Definition |
|---|---|
| | specified for each policy set; however, configuring an authenticator is not required. |
| Activation (or Application Activation) | The process of initially setting up a GD-based application. Also known as "provisioning". |
| Certificate | There are two general kinds of certificates used in Good Control:<br>• SSL/TLS certificates<br>• PKI (or PKCS 12) certificate |
| Certificate Authority (CA) | An entity that issues certificates. These can be either well-known, public third-party CAs or enterprise CAs internal to an organization. |
| Compliance Policy | A collection of rules relating to the environment GD applications can run on. For example, rules governing the OS versions or hardware models allowed to run GD applications. |
| Container | A secure storage area on the device that is controlled by the BlackBerry Dynamics framework. Only one application runs in a container, so if a user has multiple applications on a device, multiple containers exist. |
| Device | A phone, tablet, or emulator. The device can be under control of device management. A device usually also runs one or more GD applications, each in a separate container. |
| Device Configuration | Device policies can be grouped according to device configurations, which are types of network access, such as VPN or WiFi, that parallel how user groups access the network. |
| Device Policy | A collection of rules for managing the features and security of a mobile device, as distinct from the security, compliance, and application policies that manage individual application containers. |
| Enrollment (or Device Enrollment) | The process of configuring a mobile device for mobile device management.<br>There are two general kinds of enrollment:<br>• Administrator-initiated enrollment, also known as *Corporate-owned*, in which the administrator does all configuration on the physical device, which is then sent to the end-user.<br>• End-user self-enrollment, also known as BYOD, in which the end-user configures the device himself. |
| Form Factor | In mobile device management, a grouping of similar devices under the headings "phone" or "tablet." Example: An iPhone is counted as a phone, whereas an iPad is counted as a tablet. |
| Good Control Server (GC) | The GD server component that hosts the web-enabled Good Control management console, or GC console, for managing permissions and settings for BlackBerry Dynamics applications. GC resides on a machine belonging to your organization. |
| BlackBerry Dynamics Network Operation Center (or GD NOC) | A collection of Good servers that host databases and MDC, Relay, and Enterprise Gateway services for BlackBerry Dynamics. The NOC controls communication between GD applications and application data and for validating user access to GD applications. |

| Term | Definition |
|---|---|
| Good Proxy Server (GP) | A GD server component that is a secure link between the GC server and your enterprise application servers (if any) and that delivers messages to and from GD applications. GP resides on a machine belonging to your organization. |
| Policy Set | A set of all security, compliance, application, and device policies that all containers on a user's device and the user's device itself must adhere to. It can include policies for individual applications and for devices. |
| Security Policy | A collection of rules relating to the user's password and access to a container. An authentication delegate is also specified here. |
| User | An account imported into GC from Active Directory or created directly in GC. A user can have more than one device or container and is identified in GC by their email address. |

# Activating Your First GD Application

Before your first application can be registered, the following conditions must be met.

- You have installed the Good Control (GC) and Good Proxy (GP) servers using the license you generated on the BlackBerry Developer Network (BDN) portal.
- You have a BlackBerry Dynamics (GD) client application. If your organization does not have any GD applications, sample GD applications supplied by BlackBerry are available for download on the BDN portal.
- The application server, if any, is installed at a known address.

## To set up your first GD application and prepare for activation

Follow these steps in the GC console.

1. Adding applications .

   Adding or "registering" an application means that GC can manage access to it and includes specifying the GD App ID and version configured in the client. Conversely, it also enables the client application to access the application server, when necessary.

2. Entitling end-users to applications or denying them .

   Permits all GC users to install and run this application.

3. Add a user.

   See Adding A Single User Account via Active Directory , Adding Multiple User Accounts via Active Directory , or Importing Multiple User Accounts from CSV File .

4. Provision an access key for the user.

   This sends an email to the user at the email address that was imported from Active Directory. The email contains an access key the user need to activate the application on his device.

### To set up the user's device

1. Download and Install the application.

   In normal operation in production, you can download via the App Store or an enterprise application distribution server, depending on how your organization publishes GD applications.

   But for development testing of **GD App ID and Version Only** applications on, sideloading your application onto the device is the recommended mechanism
2. Launch and activate the Application.

   When the user launches the application, they are prompted for their email address and the access key sent to their email account. If this information is entered correctly, the application is activated.

The GD application is now running on a device. We recommend browsing the rest of this guide, reading documentation available on the BDN portal, and exploring the GC console to learn how you can fine tune access to your applications.

# Users and Groups

Before users in your organization can activate and run GD applications, they must have GC user account.

You have several ways you can create user accounts:

- You can create them one user at a time.
- You can search for Active Directory users and import the users into GC. When the first GC server in your cluster was installed, GC was configured to query for new users from the Active Directory domain of the administrator account that runs the GC service. For details, see Adding Multiple User Accounts via Active Directory .
- You can import user records from a comma-separated value (CSV) file. For details, see Importing Multiple User Accounts from CSV File .
- You can programatically add them using GC's web services. For details, see Good Control Web Services, listed in BlackBerry Dynamics documentation .

**Important:** If you import both from a CSV file and also from your directory service, any user whose information is in both sources is given two unique accounts, with independent account names and passwords. Inform such users that they must keep track of their credentials to use the correct password depending on which account they need to log in.

After a GC user account is created, you can control the following:

- The policy set assigned to the user.
- The list of applications a user is granted or denied access to.
- The application groups a user belongs to.
- The GD applications installed and running on the user's devices.
- The access keys sent to the user's email address that can be used to activate GD applications.

Users are identified primarily by their email address.

# Add users

## Adding A Single User Account via Active Directory

The process for creating a single new user account in GC via Active Directory is identical to the process described in Adding Multiple User Accounts via Active Directory , except instead of selecting multiple records to import, you select only a single record.

The new user is automatically added to the Everyone group and inherits all application permissions from the Everyone group by default. If the user requires access to additional applications, or if you need to restrict the user from activating and running certain products or entitlement versions, you can add the user to other groups that have the required permissions applied, or you can apply the permissions directly to the user.

The new user is also automatically assigned the default policy set. You can change the policy set for a user at any time from the user management screen. For more information, see Viewing an existing user account and Changing the policy set assigned to users.

## Searching for users in local AD domain groups to import to GC

If the GC server property **Enable to allow domain local group to be included in search** is enabled (which is not enabled by default), in the GC **Users and Groups** page, you can search for usernames in AD domain groups that you can then import into the system.

> **Note:** AD comes with the Builtin container that includes default local groups, including Users. The Users group in the Builtin container *cannot* be searched.

## Import Users by AD Group

You can import users from specific Active Directory (AD) groups into Good Control. You can set default characteristics for all users in the AD group, such as default policy set and more.

**To import from a specific AD group, in Good Control:**

1. Navigate to **Users and Groups**.
2. Click **Add Users**.
3. In the search box for **Import from Directory Service**, enter text that matches the name of the desired AD group.
4. Click **Search**.
5. In the displayed list of groups, find the desired group.
6. Under **Actions**, click the right-pointing arrow.
7. On the displayed screen make any setting changes you want:

| Setting | Desription |
|---|---|
| Auto Sync | Check this if in the future you want new users in this AD group to be added to the GC. |

| Setting | Desription |
|---|---|
| Device Enrollment Key | Check this if users in this AD group should be sent a device management enrollment key. |
| Policy Set | From the pulldown menu, select the desired policy set to be applied to the imported users. |
| App Groups | Click the pencil icon. From the displayed list of defined app groups, check the desired groups to which the imported user should be added. |

8. Check **Preview Users** if you want. A list of user information to be imported is displayed.
9. Click **Add Users**.

A job is created and scheduled to import the group.

## Adding Multiple User Accounts via Active Directory

You can create a GC user account for any user in your organization's Active Directory.

When the first GC server in your cluster was installed, GC was configured to query for new users from the Active Directory domain that the administrator account under of the GC service belongs to. You can configure your GC servers to query for users from other trusted domains by adding these domains on the **Server Configuration** > **Settings** screen. For more information, see Configuring properties for querying Active Directory.

**Important:** Make sure that you add only valid email addresses to Good Control; that is, active email address that represent real human beings. Do not add aliases, spam email addresses, or "junk" email addresses to Good Control. If you add such email addresses to Good Control and then later delete them, the human beings who "own" such email addresses who attempt to login will not be able to login.

**To create new GC user accounts from AD:**

1. Go to the **Users** > **Add Users** screen.

   With the search box on this screen, GC finds groups and users in Active Directory that match your search terms. If you do not specify search terms or they match too many Active Directory groups or users, GC displays a truncated list of results and a warning message.

   **Note:** Depending on the amount of memory of your GC server, attempting to import more than 20,000 users at a time can result in an error message; the import does not succeed. Make sure you import users in batches smaller than 20,000 users, or increase the memory on your GC server.

2. Use the search box to find a group in Active Directory. Click the group to view a list of users in that group.

   Users who already have GC user accounts do not appear in this list If you cannot locate a user you want to make an account for, you can search the existing GC user list to see if the user already has an account.

3. Select the users that you need to import into GC by clicking each individual user in the list to toggle its selection, or by pressing Ctrl+A to select all users in the list. Pressing Ctrl+U deselects all users.

At the top of the screen, you can set that policy set assigned to new GC users, the groups the users belong to, and how many access keys are provisioned for each new user. Users are still added to the Everyone group, even if you do not select any groups for the new users.

4. If you want to see a preview of the records to be imported, click the **Preview** checkbox.

   > **Important:** Be careful with preview. If the number of records to be imported is large (thousands of records), your browser might become unresponsive.

5. When you are satisfied with the settings, click **Add Users**.

   The GC server then creates a job for processing the new user accounts and puts the job in a queue. GC then displays the job details screen.

When the job is finished, GC displays a list of new users associated with the job, along with any errors encountered during account creation.

You can return to this screen at any time to view the job status. First, click the **Reporting** > **Server Jobs** main navigation link to view a list of all GC jobs, then find select a job from the list to view more information. For additional details, see Viewing the status of a job.

## Adding User Accounts

You can create individual user accounts, one at a time, and manage the account's assigned policy sets.

> **Note:** Every user in GC must have a valid email address. GC verifies the existence of the email address by contacting your organization's mail (SMTP) service. Make sure that the GC server can communicate with your SMTP server.

> **Important:** Make sure that you add only valid email addresses to Good Control; that is, active email address that represent real human beings. Do not add aliases, spam email addresses, or "junk" email addresses to Good Control. If you add such email addresses to Good Control and then later delete them, the human beings who "own" such email addresses who attempt to login will not be able to login.

**To add a user account:**

1. Go to **Users** > **Add Users.**
2. Enter the required email address and optional first and last names.
3. If you see an error message, check your SMTP settings. See the note above for explanation.
4. Click **Add User**.

The new user account is created. The system displays the policy set management screens for you to alter the assigned policy for the new user.

## Importing Multiple User Accounts from CSV File

You can create GC user accounts by importing comma-separated value (CSV) records from a file you upload to GC.

- Do not import via CSV file any users who are already defined in Active Directory. Active Directory maintains metadata about users that are not included in the CSV file. With import via CSV, user records are created locally to the GC and are not added into Active Directory. Instead, see Adding Multiple User Accounts via Active Directory .
- Make sure that you add only valid email addresses to GC; that is, active email address that represent real human beings. Do not add aliases, spam email addresses, or "junk" email addresses to GC. If you add such email addresses to GC and then later delete them, the human beings who "own" such email addresses who attempt to login will not be able to login.

## CSV Record Layout and Limits

- Your CSV file must start with a header line that includes the following comma-separated field names:

  **`email,firstname,lastname`**

  The **`email`** header field name must be first. The **`firstname`** and **`lastname`** fields can be in any order, as long as that order matches your data.
- The **`email`** field is required for every record. The other fields are optional.

  The **`email`** field must not contain spaces, commas, or any other characters that are illegal in an email address.

  The **`email`** field must conform to the Internet style email address ***`word@word.word`*** with no punctuation other than `@` and `.`.
- Your data rows must follow the header in sequence with no blank records.
- If your first name or last name fields themselves contain commas, the fields must follow standard quoting, like this example: **`"Firstname, Some Other String"`**.
- Limits:
  - 1,000 records per file/import
  - 2 MB file size

## Import Process

To create new GC user accounts by importing from a CSV file:

1. Prepare your CSV file conforming to the layout and limits detailed above.
2. In the GC console, go to **Users** > **Add Users**.
3. Under the heading **Add or Import Custom Users not in a Directory Service**, click **Add Custom Users**.
4. Click the **Import Users** tab.
5. To change the default application policy for the imported users, from the **Policy Set** pulldown menu, select the desired policy set.
6. If you want to set the default application groups for the imported users, next to **Application Groups**, click the pencil icon, from the displayed list select the desired groups, and click **OK**.
7. To start the import process, click **Upload**.
8. Browse your own computer to find the CSV file to import.

9. Click **Import** to continue or **Cancel** to stop.
10. Click **OK** to continue, or **Cancel** to stop.

The GC server queues a job to process the new user accounts and displays the job details screen. When the job is finished, GC displays a list of new users associated with the job, along with any errors encountered during account creation. For additional details, see Viewing the status of a job.

**Note:** Be patient as the job progresses.

The newly created users are notified by email when their accounts have been created.

**Important:** If you import both from a CSV file and also from your directory service, any user whose information is in both sources is given two unique accounts, with independent account names and passwords. Inform such users that they must keep track of their credentials to use the correct password depending on which account they need to log in.

### Possible Error Messages

Errors can occur at different phases of the import.

- **Pre-processing**: The system analyzes ("sanity checks") a portion of the file before starting the batch job to process it entirely.
- **Processing**: The system processes the records and displays other encountered errors, if any.

| Message | When Occurs | Workaround |
|---|---|---|
| Invalid file format: must be CSV. | Pre-processing | Use a CSV file. |
| Limit on number of records to import exceeded. Import terminated. No new records created. | Pre-processing | Only 1,000 records per file are allowed. |
| CSV file has no rows to import. | Pre-processing | Make sure that your file conforms to the heading layout and has data records. |
| Invalid header in CSV file. | Pre-processing | Make sure your file has the required header on the first row. |
| Maximum file size exceeded | Pre-processing | The file size must not be greater than 10MB. |
| User already exists. | Processing | No workaround. No new record created. |
| Invalid data format | Pre-processing | Use a CSV file. |

## AD Synchronization

Information about users frequently changes in your Active Directory database. With AD synchronization, you can have these changes in AD updated into the Good Control database according to a schedule you define. The following attributes are synchronized:

- User Principal Name (UPN)
- Email Address
- Active Directory domain

You can synchronize AD information either for the entire user base or for specific user groups:

- If for the entire user base, new records in GC are not created for users not previously imported from AD. Only changed attributes for existing users are synchronized from AD.
- If for specific user groups, new records are created for previously unknown-to-the-GC users.

By default, AD synchronization is enabled; the server property **`directory.adsync.polling.enabled`** is by default set true.

GC creates jobs to synchronize the data from AD. These jobs are run after Good Control starts and depending on the schedule you define in settings of certain server properties:

- **`directory.adsync.polling.interval`**
- **`directory.adsync.access.enabled`**

See "Properties Enhancements in this guide for details.

# Viewing an Existing User Account

The user management screen is the control and reporting center for the user account. This screen displays the following:

- The groups the user belongs to
- The applicable policy set for the user
- The number of devices the user has installed GD applications on
- A log of messages sent between the GC server and the user's installed GD applications
- The user's permissions for installing and accessing various GD applications
- The number of available access keys for the user

To view an existing GC user, first navigate to the **Users** > **Users and Groups** screen to view a search enabled list of all GC users.

This screen offers two ways to filter the list of users:

- The standard filter to match users by name or email address
- The advanced filter (above the standard filter) to match users by application group assignment, policy set assignment, or Active Directory group

### Using the advanced filter

Click **--Not Set--** to view a pulldown menu with advanced filter types, then click on a filter type to select it.

If you select the Policy Set or Application Group filter type, a secondary pulldown menu appears, containing the items which you can filter by. For example, if you select Policy Set, the secondary pulldown menu contains every policy set in your Good Control. Select an item in the secondary pulldown menu, and Good Control filters the list of users by that criterion.

If you instead select the Directory Group filter type, a secondary text box appears. You can type the name or partial name of an AD group into the box and press the Enter key to search for matching groups, or you can simply press the Enter key to request a list of all groups from your Active Directory. A popup panel containing the list of groups appears on the screen. Click one of the groups in the panel to select it, and click **OK** to filter the list of users by the group you selected.

### Viewing a user account

When you locate the user that you want to view, click the user to select it and click **Edit** to proceed to the account management screen. From this screen, you can view and manage many aspects of the account. For more information, see Managing applications on user devices, Managing application permissions for a user, and Changing the policy set assigned to users.

## Modifying User Accounts

With the GC console you can make changes to user accounts in bulk or to a single user account.

To modify multiple user accounts at once, first navigate to the **Users and Groups** screen to view a search-enabled list of all GC users.

There are two ways to filter the list of users:

- The standard filter to match users by name or email address
- The advanced filter (above the standard filter) to match users by application group assignment, policy set assignment, or Active Directory group

### Using the advanced filter

Click **--Not Set--** to view a pulldown menu with advanced filter types, then click on a filter type to select it.

If you select the Policy Set or Application Group filter type, a secondary pulldown menu appears, containing the items which you can filter by. For example, if you select Policy Set, the secondary pulldown menu contains every policy set in your Good Control. Select an item in the secondary pulldown menu, and Good Control filters the list of users by that criterion.

If you instead select the Directory Group filter type, a secondary text box appears. You can type the name or partial name of an AD group into the box and press the Enter key to search for matching groups, or you can simply press the Enter key to request a list of all groups from your Active Directory. A popup panel containing the list of groups appears on the screen. Click one of the groups in the panel to select it, and click **OK** to filter the list of users by the group you selected.

### Making changes to user accounts

You can click each individual user in the list to toggle its selection, or click the topmost checkbox in the table to select or deselect all users. Alternatively, you can press Ctrl+A to select all users in the list, or press Ctrl+U to deselect all users.

When you are satisfied with the list of users you have selected, click **Edit** to proceed. If only one user is selected, GC takes you directly to the account management screen for the user, where you can view and modify details only for that account. However, if multiple users are selected, GC displays the next screen in the bulk management flow, as shown . The remaining information in this topic assumes that you have selected multiple users.

The filtered user list appears again on this screen, and all users are selected by default. You can click each individual user in the list to toggle its selection, or click the topmost checkbox in the table to select or deselect all users. Alternatively, you can press Ctrl+A to select all users in the list, or press Ctrl+U to deselect all users.

At the top of the screen, you can configure the policy sets assigned to the selected GC users, the groups these users belong to, and how many access keys are provisioned for each user. Because only one policy set can be applied to a user account, if you select a new policy set on this screen, each of the users is reassigned the selected policy set. If you select any application groups, you can configure how the new groups are applied to the users. Beside the Group Assignment label, select Replace to replace the existing application groups for each of the selected users, or select Additive to have GC add your selected groups to the list of groups the users already belong to. For example, if you select an application group named Engineering and choose the Replace option, GC removes the users from all application groups and assigns them only the Engineering group; instead, if you choose the Additive option, GC does not remove any existing group assignments and simply adds the users to the Engineering group if they do not already belong to that group.

**Important:** Remember that changing the policy set or application group assignment for a user can have far-reaching effects. Depending on how your policy sets and application permissions are set up, users can lose access to their GD applications if you modify policy set or group assignment. For example, a user can lose permission to run an application if you remove the user from the only group that permits the application, or a user's applications can be locked or wiped due to a compliance policy violation if you assign the user a new policy set.

You can also use this screen to provision access keys for multiple users at once. Simply select a number of access keys, and GC generates that number of keys for each of the users you have selected.

When you are satisfied with your configuration, click **Update Users**. GC then creates a job for processing the changes to the selected user accounts and displays the job details screen.

For related information, see Managing applications on user devices, Managing application permissions for a user, and Changing the policy set assigned to users.

# Deleting User Accounts

If a person leaves your organization or no longer needs access to GD applications, you can delete their GC user account.

**Note:** Deleting a GC user account has far-reaching effects; *delete an account only if the user no longer requires access to GD applications.* When a person's GC user account is deleted, the person is prevented from running or activating any GD applications, and the data for any GD applications currently on all of their devices is deleted.

With the GC console, you can delete user accounts in two ways:

1. While on the user account management screen, you can click **Delete** at the top of the to remove that user from GC.
2. From the **Users** > **Users and Groups** screen, you can delete a single user or multiple users.

The following information describes how to delete one or multiple user accounts at once from the user management screen.

First, navigate to the **Users** > **Users and Groups** screen, shown , to view a search enabled list of all GC users.

You can filter the list of users in the following ways:

- The standard filter to match users by name or email address
- The advanced filter (above the standard filter) to match users by application group assignment, policy set assignment, or Active Directory group

### Using the advanced filter

Click **--Not Set--** to view a pulldown menu with advanced filter types, then click on a filter type to select it.

If you select the Policy Set or Application Group filter type, a secondary pulldown menu appears, containing the items which you can filter by. For example, if you select Policy Set, the secondary pulldown menu contains every policy set in your Good Control. Select an item in the secondary pulldown menu, and Good Control filters the list of users by that criterion.

If you instead select the Directory Group filter type, a secondary text box appears. You can type the name or partial name of an AD group into the box and press the Enter key to search for matching groups, or you can simply press the Enter key to request a list of all groups from your Active Directory. A popup panel containing the list of groups appears on the screen. Click one of the groups in the panel to select it, and click **OK** to filter the list of users by the group you selected.

### Deleting user accounts

When you have selected all of the users you want to remove from GC, click **Delete** to proceed. Click **OK** in the confirmation box to delete the users.

If a user account is deleted in error or if the person might need GD applications again in the future, you can add the user account to GC again from the **Users** > **Add Users** screen. However, the new user account does not have any of the permissions that had been configured for the deleted account and must be set up from scratch, just as for any other new user account.

## Understanding How Application Permissions are Determined

Good Control has three tiers of application permissions. Each tier in this list overrides the tiers underneath it:

1. User level permissions
2. Application group level permissions
3. Everyone group level permissions

Users can inherit application permissions from various sources, and these permissions might be in conflict with each other. With the GC console you can view the source of each grant or deny permission set and the actual resolved permission applied for the user.

To view resolved permissions, first go to the user account management screen and click the **Applications** tab. This tab shows a list of applications that the user has been granted or denied access to, based on a combination of user level permissions and group level permissions.

Click the name of an application to expand or hide an unresolved list of allowed or denied entitlement versions. To view resolved permissions for an entitlement version, click the version number or the ℹ **Info** icon.

The same application can show up in both the allowed and denied lists. This is because permissions are applied at the version level, not the application level, so some versions of the application can be allowed and others can be denied. In this case, you can view the information for the application in both lists to determine the allowed and denied versions.

A user can inherit permissions from the Everyone group or application groups created by GC administrators. Permissions set for the Everyone group act as default permissions, or a baseline permission set that all GC users automatically inherit.

The next tier of permissions is set at the application group level. If a user belongs to one or more application groups, any permissions applied to these groups override Everyone group permissions if there is a conflict, or add to the list of permissions inherited from the Everyone group. If a user belongs to multiple groups, the groups might have conflicting permissions for a given application or entitlement version. When this happens, the lowest and most restrictive of the inherited permissions is applied at group level for that particular application or version.

The top tier of permissions is set on the user account management screen. These permissions set at the user level override any permissions set at group level and Everyone group level.

### Example

An organization's GC is set to allow access to all versions of "My App" for the Everyone group.

User "David" belongs to seven application groups. Six of the groups allow version 2.0 of "My App" and one of the groups denies version 2.0 of "My App". This entitlement version is **denied** for this user at group level, because the most restrictive permission is applied for the user when there is a conflict. No user level permissions are set for this application on the user's account, so access to version 2.0 of "My App" is effectively **denied** for "David".

However, if a GC administrator with the right to modify users and groups goes to the user's account management screen and then clicks the ⬆ **Allow** icon for "My App" version 2.0, the entitlement version is now **allowed** for user "David" because permissions set at the user level override any set at group or Everyone group level.

# Entitling end-users to applications or denying them

Your end-users must be entitled to view or run the applications defined in the application catalog. You can also deny them the right to applications. You can entitle or deny end-users in several ways:

- WIth app groups
- Per individual end-user

## Sequence of app version entitling and denying: entitle, then deny

**Important:** If you are entitling a new app version and denying an older version, be sure to entitle the new version first before you deny access to the older version. If you deny the older versions first, the app will be wiped from the device.

## Entitling or denying end-users via entitlement groups (aka app groups)

By default, Good Control comes with the Everyone group, to which end-users are added automatically. The easiest way to entitle all your end-users is to entitle the Everyone group.

You might have the need for different end-user groups for finer control over which end-users can use which applications. In this case, entitle the appropriate user groups for just those applications you want them to use.

**To entitle or deny via the Everyone group:**

1. In Good Control, navigate to **App Groups**.
2. Edit the appropriate group by clicking the edit icon (pencil) on the far right of the group name.
3. Under either **Entitled Enterprise Apps** or **Denied Enterprise Apps**, click **Add More**.
4. From the displayed dialog, you can select applications in several ways, some combinations of which are mutually exclusive. Choose the desired ways:
   - From the **View** pulldown, select the type of application to show: **All**, **Organization**, **Partner** or **Good**.
   - If desired, click the **Show dev versions** checkbox.
   - In the text box, enter the name of the application you are looking for.
5. After finding the desired application, you can click the triangle left of its name to see the registered versions of the application.
6. Click the checkbox for **ALL** or the individual checkboxes for only the desired versions.
7. Click **OK** to save your changes or the **X** in the upper right of the dialog box to discard them.

## Entitling or denying an individual end-user

In Good Control's **Manage Users** screen, you can manage various aspects of users in bulk (that is, more than a single user at a time), but to entitle or deny an end-user an application, you can operate on only a single user at a time.

You can entitle the end user by way of app groups or by entitling the end-user individually.

**To entitle or deny a single end-user:**

1. In Good Control, navigate to **Users and Groups**
2. You can filter users in several ways. Choose the desired ways:
   - From the **Filter users by** pulldown, select **Policy Set**, **Application Group**, or **Directory Group** (Active Directory group). Then from the additional pulldown menu, choose the specific policy set, app group, or enter the name of the specific AD group.
   - In the text box, enter the name or email address of the desired end-user.
3. After finding the desired end-user, click the checkbox left of the end-user's and in the upper right, click **Edit**.
4. If you want to assign the end-user to a previously defined an app group. click **App Group**, scroll to find the desired group, and click Save.
5. 1. Under either **Entitled Enterprise Apps** or **Denied Enterprise Apps**, click **Add More**.
   2. From the displayed dialog, you can select applications in several ways, some combinations of which are mutually exclusive. Choose the desired ways:
      - From the **View** pulldown, select the type of application to show: **All**, **Organization**, **Partner** or **Good**.
      - If desired, click the **Show dev versions** checkbox.
      - In the text box, enter the name of the application you are looking for.
   3. After finding the desired application, you can click the triangle left of its name to see the registered versions of the application.
   4. Click the checkbox for **ALL** or the individual checkboxes for only the desired versions.
   5. Click **OK** to save your changes or the **X** in the upper right of the dialog box to discard them.

## Activating an Application for a User

If a user wants to install and run a GD application on a device, you must first grant the permission to their GC account so they can run that application.

The user also needs an access key, which must be entered correctly on the device when the user runs the application for the first time. The access key is a 15 character code that is sent in an email to the user's company email address. Access keys have the following properties:

- They can only be used one time.
- They are not specific to an application. For example, a user sent four access keys can use them to activate any four applications he is entitled to.
- They do not support re-activation. If a GD application is uninstalled and then reinstalled on the same device, a new activation key is required. This also pertains to new or factory-reset devices and device emulators that do not preserve state. However, a user issued multiple keys can use them to activate the same application multiple times.
- They can be configured to expire after a specified period of time. For any policy set, in the Provisioning Policies section, you can select the option labeled **Access Keys expire after** and use the pulldown to choose the number of days before an access key expires if it is not used.

**To provision an access key for a user, in Good Control:**

1. Navigate to the **Users and Groups** screen.
2. Find the user in the list of accounts,.
3. Click the username to edit the record.
4. Click the **Access Keys** tab.
5. Click **New Access Key**.

A new access key is generated and sent to this user.

If you are logged into Good Control with administrative permissions, the generated access key is displayed directly on screen. You can note it without having to open the sent email.

## Action by the User

Activation keys are then sent to the user's corporate email address. Each email message contains one key. Hashes of the activation keys are also copied to the GD NOC to enable container validation.

When the user receives the activation email, as long as the key has not yet expired, he can activate a GD application on a device with these steps.

1. The user must download the GD application to their device, if they have not already done so.
2. The user must launch the GD application. The BlackBerry Dynamics user activation screen is displayed.
3. The user must enter the activation key and their corporate email address, both in the activation email, into the user activation screen.

The client then sends the activation key to the GD NOC. If the correct key is entered, the application is activated and becomes usable on the device. Because the key is used once only during activation, it is dropped from the account screen's **Keys** tab.

# Resending and Canceling Access Keys

When you provision an access key for a user, Good Control sends an email to the user's corporate email address. This email contains the key that the user must enter into the GD application in order to activate it on a device. If this email becomes lost or is accidentally deleted, you can resend the email. You can also cancel any key not yet used in activation.

To resend or cancel a key, go to the user's account management screen and open the **Keys** tab.

Click **Resend keys** to resend the email to the user.

If you need to revoke or cancel an access key, check the checkbox for a key and click the **Delete**  to remove it from the system.

Expired access keys can be canceled, but they cannot be resent to the user. If a user has lost the provision email and the access key has expired, simply cancel the expired key and provision a new key for the user.

If user self service is enabled, users can log into their own accounts and manage their own access keys through the self service portal. For more information, see Configuring self service settings.

# Apps: Wipe, Unlock, Lock, Upload Logs, and More

The GC console shows you which GD entitlement versions are installed on a given GC user’s devices so you can manage certain actions on the GD applications.

While on the account management screen for a user, click the **Devices and Apps** tab to view a list of devices, if any, that the user has activated GD applications on. During activation, a GD application reports an identifier for its host device, and GC stores this information. When you view the **Devices and Apps** tab, GC displays an organized list of containers grouped by common device identifier. If the user has not yet activated any GD applications, no devices appear on this tab.

Click the toggle for a device to view a list of all GD applications that have been activated on that particular device. Because each activated application resides in its own separate container, each entry in this list represents one container.

A user's containers can be organized in a different manner on this screen if the identifier is changed for one or more of the user's devices. Certain actions, such as upgrading to iOS 7, automatically modify the device identifier. Additionally, the user can manually modify the device identifier. If the user activates a new GD application after the device identifier is changed, the new container is listed under a new device on this screen, instead of being grouped with the other containers associated with the old device identifier. However, if a previously activated container reconnects to GC after the device identifier is changed, it reports the new device identifier, and GC updates the records for other containers on the same device to reflect the new identifier. All containers on the device are then grouped together on this screen again.

### Container actions

GC administrator accounts with the right to manage containers can manage the containers of any GC user. Self-service users can do these actions on their own containers but cannot view or act upon the containers of other users.

**To work with container actions:**

1. Navigate to **Users and Groups** > *select a user* > **Edit** > **Devices and Apps** > *select a device* > **Installed Apps**.
2. Check the checkboxes for the applications you want to change.
3. Use the **App Actions** menu on the right to do the functions you want:

| Menu Selection | Description |
|---|---|
| Lock App/Unlock App | Lock or unlock application containers for selected applications |
| Remove App | Delete the application from Good Control |
| Logging On/Logging Off | Turn on or off application container logging. Logging is always set to "debug logging" for maximum detail. |
| Upload Logs | Upload application container logs from the user device to the GD NOC.<br>**Note:** Be sure that **Logging On** had been previously set so logs have been captured to be |

| Menu Selection | Description |
|---|---|
| | uploaded. |
| Get Info | Display detailed information about compliance and other events associated with the application container |

## User Devices: Wiping, Clearing Passwords, Locking, Deactivating Device Management

See Device Management Operational Tasks: Device Status, Lock, Clear Password, Wipe, and Deactivate  and Unenrolling a Device from MDM .

## User Self Service

With User Self Service, GC users can log in to do a limited set of tasks on their accounts. They have access to a shortened version of the online GC help documentation located here.

Users can do the following tasks:

- View the GD applications activated on their devices and read container history logs
- Lock, wipe, or unlock GD applications on their devices
- Provision, delete, or resend their own access keys
- Upload personal PKCS 12 certfificates

Users can view but not modify the following:

- The policy set assigned to them
- Application groups they belong to
- Applications they are permitted or denied access to

Self service users cannot access or modify any other information in GC.

User self service is initially disabled by default. You can enable user self service on the **Servers** > **Settings** screen. For more information, see Configuring self service settings.

## Security: Close browser on logout

To maintain the security of the Good Control console, after you logout, it is best to close your browser window. This ensures that your session is completely terminated.

# Administrators

## Adding Users as Administrators

> **Note:** Before you can make a user an administrator of GC, the user must already exist in the GC.

To give a GC user administrator rights, you need to add that user to the Administrator predefined role. Adding Users to Predefined Roles

## Add service account for role via GC console

A service account for Good Control is an account that has a limited defined function associated with a GC role. For separation of concern, service accounts allow you to divide work according to these roles that have only a limited set of permissions.

You can create a service account for any predefined or custom role by way of the GC user interface.

**To create a service account for a role in Good Control**:

1. Navigate to **Administrators** > **Roles**.
2. In the list of roles, click the role to which you want to add a service account.
3. Click **Add Service Account**.
4. Enter the name of the service account.
5. Click **Add** to add the service account or **Cancel** to discard your entries.
6. Securely record the password for this service account so it can be given to the human beings who will use the service account.

   > **Note:** The password is displayed in the user interface only once; it is never displayed again in the user interface.

7. Click **OK** to confirm.

## Enhancements to Manage User page: container lock status and auth delegates

The **Manage User** page now shows the following information, if applicable:

- Status of lock on containers
- Authentication delegates of containers

## Deleting Administrator Accounts

Deleting an administrator account involves several steps:

- Remove the user account, just as you would for any user who is no longer active. See Deleting User Accounts
- Remove the user from the Administrator role. See Removing Users from Roles

# Understanding Administrator Rights

With Role-Based Access Control (RBAC), your organization can easily restrict access to GC functions and offload tasks (particularly those related to container management) from IT administrators to help desk support specialists or other administrators without compromising internal policies and requirements. Role privileges are enforced globally across all GC servers in your cluster, so administrators have the same rights and access for any GC server they log into.

An administrator can have multiple roles. In this case, the administrator inherits the cumulative rights granted to all roles to which the administrator's account belongs. For example, if an administrator's account belongs to a role that allows members to modify user account information and another role that does not, GC combines the rights of both roles to determine that the administrator is allowed to modify user account information.

Administrators can also have GC user accounts, although this is not a requirement. Administrators without user accounts can log in to GC to do the administrative activities they have been granted, but they do not have application permissions or policy sets assigned to them, and they cannot generate access keys or activate GD applications for their accounts. If an administrator needs to install and use GD applications, the administrator simply creates a GC user account with the **Users and Groups** > **Add Users** screen.

Use the following information as a reference when determining the rights to grant an administrator or role. Each item described in the following sections identifies a right configurable from this screen and indicates the GC functions available to an administrator who has been granted the right.

You cannot modify a predefined role, but you can create a custom role that has the permissions you need. See the following topics:

- Default permissions and web services requests for predefined roles
- Creating and Configuring a Custom Role

**Note:** Your username in the Good Control must be a member of a role that has permission to view reports. For instance, the Help Desk Administrators predefined role does not have permission to view reports. Follow the steps in Creating and Configuring a Custom Role to create a role with the Reports and Troubleshooting permission that the Help Desk people need.

## User and Group Management

This right does not include container management.

- Create, modify, or delete users and groups
- Change the policy set assigned for any user

## App Groups

- Manage application permissions for users and groups

## Container and Device Management

- Lock, unlock, or delete users' containers
- Enable or disable detailed logging for user's containers
- Send the command for any user's container to upload its logs to the GD NOC
- Generate, resend, or revoke access keys for any user
- View, without modifying, the policy set assigned to a user, the user's group membership, and any application permissions for the user

When Good device management is enabled, the following additional permissions are included in the Container and Device Management permission. Any predefined role such as Global Administrator and Help Desk obtains these permissions. In addition, any custom role that has the Container and Device Management permission also obtains these permissions:

- Lock Device
- Clear Device Password
- Wipe Device
- Installed Apps
- Deactivate Device
- Add Device Enrollment Key

The Container Management right has several child rights:

- **Create New Access Key.** This right allows administrators to generate new access keys.
- **View Full Access Keys for All Users**. This right allows administrators to view all characters in the access keys generated for all users. Otherwise, GC only displays the final five characters of users' access keys.

## Policy Sets

- Create and delete policy sets
- Modify all policy set information, including Security Policies, Compliance Policies, and Application Specific Policies
- Assign an authentication delegate for a policy set

## Applications, Shared Services, and Application Wrapping

- Register, modify, or delete organization applications and application services
- Configure application servers for any GD application
- Apply a policy override or upload application specific policies for any GD application
- Modify all settings related to application wrapping
- Wrap applications and store signing certificates for later use

### Roles

- Create, modify, or delete custom roles
- Add or remove members for any role

### Server Configuration

- Modify settings for all GC servers in the server cluster
- Create, modify, or delete GP server clusters
- Assign primary and secondary GP clusters to GC servers
- Configure the domains, subnets, and servers that can be accessed by your users' GD applications
- Generate licenses to install new GC servers into the cluster
- View the status of all GC and GP servers in your deployment
- Unregister GC and GP servers

### Reporting and Troubleshooting

- Export or purge the audit trail logs for all GC servers in the cluster
- Export container and compliance violation data to CSV file for reporting
- View the status of server jobs
- Upload server logs to Good for analysis

## Default permissions and web services requests for predefined roles

Good Control creates the following predefined roles, which are granted specific rights. These roles have certain permissions to perform functions in the Good Control UI or with the GC web services.

- **Good Control Global Administrators** - Administrators with this role are granted the privilege to all functions, modify settings for all GC servers, and make changes to any user account. Additionally, these administrators can create, delete, and modify any other roles. The first Good Control Global Administrator is created from the Active Directory user specified during the installation of the first GC server in your server cluster.
- **Help Desk Administrators** - This role has limited access to GC data and functions. Administrators with this role are able only to view user account information, including application permissions, and to manage containers for all GC users. For example, they can delete, lock, or unlock any GD application for any GC user. Administrators with this role can generate an access key for any user, but as a security measure, they are not allowed to view the entire access key. Instead, GC displays only the final five characters of the access key.
- **Service Accounts** - This role is for use by third-party server monitoring and reporting tools. These administrators can do all functions except role management.

## About permissions for web services requests

Based on the specific permissions listed below for the various roles, you can correlate with the GC's SOAP request names or HTTP API names to determine if a role has the necessary permission to execute a particular request.

For example, the Help Desk Administrator role can execute the **`GenerateAccessKeysRequest`** but cannot execute the **`GetUsersRequest`** or **`GetPolicyDetailRequest`**.

## Specific permissions for Global Administrators role

The following are the permissions for the Good Control Global Administrators role: all permissions.

The Global Administrator role can execute any of GC's web services requests.

- Users and Devices: All Access
  - Devices
- Entitlement Groups: All Access
- Container/Device Management: All Access
  - Create New Access Key
  - View Full Access Keys for All Users
- Policy Sets: All Access
  - Apple DEP Profiles
- Applications, Shared Services, and Application Wrapping: All Access
- Roles: All Access
- Server Configuration: All Access

## Specific permissions for Help Desk Administrators role

The following are the permissions for the Help Desk Administrators role.

The Help Desk Administrator role can execute web services requests that relate to container and device management and user roles.

- Users and Devices: No Access
  - Devices
- Entitlement Groups: No Access
- Container/Device Management: All Access
  - Create New Access Key
- Policy Sets: No Access
  - Apple DEP Profiles
- Applications, Shared Services, and Application Wrapping: No Access

- Roles: All Access
- Server Configuration: No Access

### Specific permissions for Service Accounts role

The following are the permissions for the Service Accounts role: all permissions except roles.

The Service Account role can execute all web services requests except those related to roles.

- Users and Devices: All Access
  - Devices
- Entitlement Groups: All Access
- Container/Device Management: All Access
  - Create New Access Key
  - View Full Access Keys for All Users
- Policy Sets: All Access
  - Apple DEP Profiles
- Applications, Shared Services, and Application Wrapping: All Access
- Roles: No Access
- Server Configuration: All Access

## Adding Users to Predefined Roles

With Role-Based Access Control (RBAC), your organization can easily restrict access to GC functions and offload tasks (particularly those related to container management) from IT administrators to help desk support specialists or other administrators without compromising internal policies and requirements. Role privileges are enforced globally across all GC servers in your cluster, so administrators have the same rights and access for any GC server they log into.

You cannot modify the right assignments for predefined roles, but you can associate administrators with one or more of these roles. To add one or more administrators to a role:

1. Navigate to the **Administrators** screen.
2. Click the name of the role or on its associated pencil icon to view the role management screen.
3. Select the **Members** tab.
4. Click **Add**  to view the Add Admins to Role panel.
5. Use the search box to find the account or accounts you want to associate with the role. GC displays a list of matching user accounts.
6. Select the desired account or accounts.
7. Click **Add**. GC then adds the selected user account or accounts to the administrator role.

An administrator can have multiple roles. In this case, the administrator inherits the cumulative rights granted to all roles to which the administrator's account belongs. For example, if an administrator's account belongs to a role that allows members to modify user account information and another role that does not, GC combines the rights of both roles and determines that the administrator is allowed to modify user account information.

You can use any predefined role as a template for a fully configurable custom role. For more information on administrator rights and roles, see Understanding administrator rights and Creating and configuring a custom role.

## Searching for members of administrative roles

Good Control's **Administrators** > **Edit Role** page now includes a filter textbox to search for the names of members in the role.

**To search administrative roles by member name:**

1. Navigate to **Administrators** > **click a role** > **Edit Role** > **Members** tab.
2. In the **Name filter** textbox, enter a string to search for.
3. Hit return.

The matching member names are displayed.

## Removing Users from Roles

Removing a user from a role is the inverse of adding that user to the role:

1. On the left, under **Roles**, click **Administrators**.
2. On the right, click the name of the affected role from which the user must be removed.
3. Click the **Members** tab.
4. Find the name of the user to remove from the role.

   > **Note:** You cannot delete the last user who has been added to the Global Administrator role. The system must always have at least one user who can administer it.

5. On the right, click the trash can icon.
6. Click **OK** to confirm or **Cancel**.

## Creating and Configuring a Custom Role

Any administrator who has been granted the role management right can create a new custom role by copying an existing predefined or custom role. The new role inherits only the right assignments of the original role, not the role members. The administrator can then add members to the new role and modify its rights. Custom roles that were created in this manner can be updated or deleted by any administrator who has been granted the right to modify roles.

To create a new role, simply navigate to the **Roles** > **Administrators** screen and click the 🗐 **Copy** icon for the existing role you want to use as the foundation for the new role GC then creates a new role with the same rights as the original role and forwards you to the role management screen.

To modify the rights for a role:

1. Navigate to the **Roles** > **Administrators** screen.
2. Click the name of the role or its corresponding **Edit** button to view the role management screen.
3. Select the **Permissions** tab.
4. Choose the **All Privileges** option to automatically grant all rights to the role, or choose the **Custom** option to configure the access level for individual rights.
5. If the **Custom** option is selected, choose to grant either All Access or No Access for each right. For a detailed description of the GC functions associated with each administrator right, see Understanding Administrator Rights .
6. Click **Update** to save your new configuration. The GC console displays an alert message if any issues are encountered.

To add one or more administrators to a role:

1. Navigate to the **Roles** > **Administrators** screen.
2. Click the name of the role or its corresponding ✏ **Edit** icon to view the role management screen.
3. Select the **Members** tab.
4. Click **Add** to view the Add Admins to Role panel.
5. Use the search box to find the account or accounts you want to associate with the role. GC displays a list of matching user accounts in Active Directory.
6. Select the desired account or accounts. You can do this by clicking on each individual account in the list to toggle its selection, or by pressing Ctrl+A to select all accounts in the list. Pressing Ctrl+U deselects all accounts.
7. Click **Add**. GC then adds the selected user account or accounts to the administrator role.

## Viewing the Resolved Rights for an Administrator

Because an administrator can belong to multiple roles, if your organization maintains multiple roles, remembering exactly which administrators have been granted which rights can be difficult. Remember that an administrator with more than one role inherits the cumulative rights granted to all roles to which the administrator's account belongs. For example, if an administrator's account belongs to a role that allows members to modify user account information and another role which does not, GC combines the rights of both roles and determines that the administrator is allowed to modify user account information.

For your convenience, in GC you can view the resolved rights granted to any administrator. To view resolved rights:

1. Navigate to the **Roles** > **Administrators** screen.
2. Find one of the roles to which the administrator belongs, and click the name of the role or its corresponding ✏ **Edit** icon to view the role management screen.
3. Select the **Members** tab.
4. Locate the administrator whose information you want to view and click the associated ℹ **Info** icon. GC then displays the roles to which the administrator belongs and the resolved list of rights for the administrator.

# Working with DEP-Enrolled Devices

On Good Control's **Apple DEP Devices** page, you can work with your DEP-enrolled devices in several ways.

**Important:** In general, you should perform all actions with DEP-enrolled devices in Good Control itself, not in Apple's portal.

## Filtering and Searching

**To filter and search Apple DEP devices, in Good Control:**

1. Navigate to **Apple DEP Devices**.
2. Use the **Filter** pulldown menu to narrow the displayed devices:

- All DEP Devices
- DEP Profile Assigned
- MDM Enrolled
- No DEP Profile Assigned
- Pending DEP Profile Change
- Filter based on CSV file

## Filtering by CSV File from Apple

Good Control does not have knowledge of your order numbers from Apple, Inc. You can use "Filter by CSV" to get the device serial numbers by order number. Your CSV file to filter the display of DEP devices requires only a single column: the exact serial numbers you want to see. All other columns are ignored.

1. From Apple DEP's site, download a CSV file of the serial numbers for a given order.
2. Use this CSV to filter in Good Control.

There is no partial string matching. Your column 1 must include the full, exact serial numbers, as it does when you download from Apple.

## Synching with Apple

Your inventory of devices on file with Apple is synchronized with Good Control once an hour.

**To force the synchronization of the device records in Good Control with Apple's inventory of your devices, in Good Control:**

1. Navigate to **Apple DEP Devices**.
2. Click **Sync Now.**

## DEP Device Actions

**To perform various administrative action on Apple DEP devices, in Good Control:**

1. Navigate to **Apple DEP Devices**.
2. Select the desired device records. See Filtering and Searching .
3. From the **Device Actions** pulldown menu, select the desired action:
   - Wipe
   - Reset Password
   - Deactivate Device
4. Follow the leading prompts to complete the action.

## Export to CSV

**To export the selected Apple DEP device records in comma-separated value (CSV) format from Good Control:**

1. Navigate to **Apple DEP Devices**.
2. Select the desired device records. See Filtering and Searching .
3. Click **Export**.
4. Follow the leading prompts to complete the action.

# Manage Apps

## Key concepts

Some of the more important concepts underlying application management are described here.

### Types of applications

BlackBerry application management categorizes applications for management under several headings.

| Type | Description |
|---|---|
| Public store application | Public store applications are those that are posted to either Apple App Store or Google Play Store. |
| Custom application | Application binaries not in the public stores can be uploaded to Good Control. |
| Web application | Applications that are accessed via a URL on either the public Internet or the private intranet |
| BlackBerry Dynamics Entitlement ID and Version Only | For development and testing, when the actual executable application binaries are not yet available. |

## About BlackBerry Dynamics entitlement ID and version

In the Good Control console and the BlackBerry Developer Network, BlackBerry Dynamics-based applications are identified by a *BlackBerry Dynamics entitlement ID* and *entitlement Version*. A primary purpose of the BlackBerry Dynamics entitlement ID and entitlement Versions is for you to manage end-user entitlement to your BlackBerry-provided applications; in this context you might hear the BlackBerry Dynamics entitlement ID referred to as "entitlement ID"; for BlackBerry Dynamics-based applications, the terms are equivalent.

A single BlackBerry Dynamics entitlement ID must be used to represent the same application across all platforms. Other restrictions also apply.

By default, access to applications varies by type of application:

- All versions of Partner/ISV applications are by default permitted to all to authorized users of any organization to which the application has been published.
- Each version of a BlackBerry Dynamics-based application by default requires the BlackBerry Dynamics administrator's explicit granting of access on the GC console to run.

BlackBerry recommends that you devise a naming scheme to meet your needs. Use these guidelines to help you formulate that naming scheme.

A simple example: assume we have a BlackBerry Dynamics-based application from a company called Acme, Inc. The native version number is completely independent of the BlackBerry Dynamics entitlement version.

- BlackBerry Dynamics entitlement ID: com.acme.gd
- BlackBerry Dynamics entitlement version: 1.0.0.0
- Native version number: 2.0

Other variations on naming schemes for BlackBerry Dynamics entitlement ID and entitlement Versions are also possible, but keep these details in mind when you devise your own BlackBerry Dynamics entitlement ID naming scheme.

### BlackBerry Dynamics entitlement and entitlement version both required for all BlackBerry Dynamics-based apps

You need to define both the BlackBerry Dynamics entitlement ID and the entitlement Version for all your BlackBerry Dynamics-based applications, regardless of whether or not you use the BlackBerry Dynamics Shared Services Framework.  Developers and administrators should ensure that the value specified for the GDApplicationVersion key in an app's application configuration files is the same as the value the administrator specifies in Good Control.

The entitlement Version is independent of any native version identifier; see more information in Distinction from and use with native language identifiers .

### When to change the BlackBerry Dynamics entitlement version?

The BlackBerry Dynamics entitlement Version is distinct from any visible version number you might use for your application. For example, your BlackBerry Dynamics entitlement Version might be "1.0.0.0" while at the same time you publicly show a native version number "2.1".

Because each new BlackBerry Dynamics entitlement Version of your BlackBerry Dynamics-based application requires "publishing" it to your existing customers, it is recommended to change the BlackBerry Dynamics entitlement version number as infrequently as possible. There are three primary reasons to change the BlackBerry Dynamics entitlement version number:

1. To provide early access, beta, or limited access to a new version for specific customers.
2. For Partners/ISVs, to monetize new functionality differently from your existing version.
3. To represent large level differences in BlackBerry Dynamics functionality (not your own functionality). For example, you might update a service definition, that is, publish a service update that is not supported on an older entitlement Version.

When a new version is to be made available per above (which is usually rare), ensure that the new version is listed on the Marketplace by a partner or on the GC console for custom applications *well before* an application reporting that BlackBerry Dynamics version is ever available in the App Store, Play Store or elsewhere. If the new version of the application is downloaded to a device before the version is published on GDN or in GC, the application is blocked. You should never delist a version unless it is to enforce payment, force end-of-life, or remove a version with a fatal security issue. If a BlackBerry Dynamics entitlement ID or entitlement Version is ever unpublished or an end-user unentitled from an a previously entitled application, the container is wiped from end-user devices for all end-users who installed the application.

## Format of BlackBerry Dynamics entitlement ID and version values

The general form of a BlackBerry Dynamics entitlement ID is:

***your_company_name.your_application***

The value of your BlackBerry Dynamics entitlement IDs must follow these rules:

- Must be in reverse domain name form, like com.*yourcompany*.*something*.
- Must not begin with any of the following:
  - **com.blackberry**
  - **com.good**
  - **com.rim**
  - **net.rim**
- No uppercase letters.
- In addition, the string must conform to the **<subdomain>** format defined in section 2.3.1 of RFC 1035, as amended by Section 2.1 of RFC 1123.

**Note:** In the BlackBerry Dynamics SDK for Microsoft Windows 8.1, the value of BlackBerry Dynamics entitlement ID (Application ID) cannot be longer than 35 characters. This does not apply to the BlackBerry Dynamics SDK for UWP.

The value of your entitlement Versions must follow these rules:

- From one to four segments of digits, separated by periods, like **100** or **1.2.3.4**.
- No leading zeroes in the numeric segments. For example, these are *not* allowed: **0100** or **01.02.03.04**.

- The length of the numeric segments can be from one to three characters. This is an allowable example: **`100.200.300.400`**.

### Distinction from and use with native language identifiers

The BlackBerry Dynamics Entitlement ID and Entitlement Version are Good-specific metadata and are independent of the identifiers needed by the application platforms themselves. The key point is that the BlackBerry values and the native language identifiers' values *can* be the same but they do not necessarily *have* to be. Listed below by platform are the equivalent native identifiers, which are where the values of BlackBerry Dynamics Entitlement ID and version are stored.

| Platform | Location | Platform-specific Names |
|---|---|---|

### Mapping BlackBerry Dynamics entitlement ID to native identifiers

To take advantage of many BlackBerry Dynamics features, such as Easy Activation, multi-authentication delegation, and the BlackBerry Dynamics shared services framework, developers need to set up a map in Good Control between your defined BlackBerry Dynamics Entitlement ID and the native identifiers on the platforms for which your application is distributed. The native platforms have no knowledge of the BlackBerry Dynamics Entitlement ID; thus the mapping is needed for the operating systems to take over the actual function of the app.

- This same Native Bundle ID must be registered with BlackBerry to match the app's specific GDs App ID. Without this mapping your app cannot take advantage of Easy Activation.

  Contact your BlackBerry Dynamics administrator to have this mapping recorded in the GC console or in GDN. In the GC console, the steps are as follows. For each application that requires the native Bundle ID:

- Go to Manage Applications.
- Click the name of the application.
- Go to the Advanced tab. (The Advanced tab is available only for custom applications developed by an organization or to Independent Software Vendors (ISVs).)
- Set the identifier for the appropriate devices.

### Native version identifiers: * wildcard allowed for blocking app

The BlackBerry Dynamics SDK supports use of native version identifiers in keeping with the conventions described by the major vendors. These same conventions apply to the use of the * wildcard in Good Control to deny apps by native version.

| Platform | Definition | Reference |
|---|---|---|

The * character can be used in native version identifiers, but must always be preceded by a period (.) and must be the last character in the native version string. Examples:

- Allowed: 2.3.*
- Not allowed: 2.*.3
- 2.* includes 2.*.*

## Enforcement of BlackBerry Dynamics entitlement ID and version in Good Control

The following are the basic rules that application developers must comply with. In this discussion, the terms "BundleIdentifier" and "BundleVersion" are used to cover all similar platform-specific identifiers, such as package name or Application ID.

1. Application name is unique with in the organization.
2. Bundle Version, Bundle Identifier combination is unique for a platform.
3. Change in BlackBerry Dynamics Version enforces change in Bundle Version. The other way round is not true.
4. An application (family of binaries) is either BlackBerry Dynamics (all binaries under it are BlackBerry Dynamics) or non-BlackBerry Dynamics (all binaries under it are non-BlackBerry Dynamics). This rule derives from that entitlement ID is locked at the time of creation. The entitlement ID is the BlackBerry Dynamics entitlement ID if the application is a BlackBerry Dynamics-enabled app.
5. BlackBerry Dynamics entitlement ID is unique throughout the system.
6. Bundle Identifier for a platform is unique for a BlackBerry Dynamics entitlement ID and vice versa. Therefore, a change in BlackBerry Dynamics entitlement ID requires a change in Bundle Identifier, and vice versa.
7. Non-BlackBerry Dynamics and BlackBerry Dynamics versions of same binary have different Bundle Identifiers.

### Common errors

The following are errors in usage of the BlackBerry Dynamics entitlement ID and Entitlement Version that are checked by Good Control when BlackBerry Dynamics-based applications are added.

| Use Case | Explanation of Error |
|---|---|
| Administrator submits an app with an existing BlackBerry Dynamics entitlement ID for an app with some other org. | The BlackBerry Dynamics entitlement ID must be unique across organizations. |
| Administrator submits an app with an existing BlackBerry Dynamics entitlement ID, Bundle Identifier, Bundle Version but different BlackBerry Dynamics Entitlement Version. | The Bundle Version must be changed when there is a change in BlackBerry Dynamics version. |
| Administrator submits an app with an existing Bundle Identifier and Bundle Version but different BlackBerry Dynamics entitlement ID. | The Bundle Identifier should be different for different BlackBerry Dynamics entitlement ID. |
| Administrator submits an app with an existing BlackBerry Dynamics entitlement ID, but different Bundle Identifier for an existing platform. | The Bundle Identifier for the same platform should be unique within a BlackBerry Dynamics App. |
| Administrator submits a BlackBerry Dynamics-enabled app with same Bundle Identifier as an existing non-BlackBerry Dynamics app | Upgrading a non-BlackBerry Dynamics app to a BlackBerry Dynamics app binary requires a change in Bundle Identifier. |
| Administrator submits a non-BlackBerry Dynamics enabled app with same Bundle Identifier as an existing BlackBerry Dynamics app | Downgrading a non-BlackBerry Dynamics app to a BlackBerry Dynamics app requires a change in Bundle Identifier |

## Application catalog

This document uses the term *application catalog* to refer to the display of per-user entitled applications from which end-users can access approved, managed applications via a Good-based application, such as BlackBerry Access. The applications displayed by the catalog are defined by the Good Control administrator, but the application catalog itself is served by the BlackBerry Dynamics NOC:

- The application catalog always serves the latest version of an application to be uploaded or defined.
- The application catalog is sometimes referred to as the "app store", which is not to be confused with the public app stores from Apple or Google.
- End-users must be entitled to the application catalog; see Essential one-time setup tasks .
- In BlackBerry Access, the application catalog is accessed via the **Applications** shopping bag icon, as described in Viewing the BlackBerry Application Catalog in BlackBerry Access.

## Form factor or "platform"

What type of hardware does the application run on? This is called the *form factor* or "platform" of the application. application management distinguishes the following types:

- For iOS:
  - Phone
  - Tablet
- Android, for all types of devices

# Blacklisting or whitelisting applications on devices

In Good Control's **Manage Apps**, the **Blacklist** and **Whitelist** tabs give you large-grained control over the applications not allowed or allowed to run on end-user devices:

- Blacklist: Applications not allowed to run on the device
- Whitelist: The only applications allowed to run on the device

## Behavior

The precise effect on a device depends on the operating system and type of application.

If your device policy checks compliance against the blacklist, then applications on the blacklist cannot be run on the device, subject to the device's operating system constraints.

| OS | How Enforced |
|---|---|
| • iOS<br>• Android | The iOS and Android (without Samsung KNOX) operating systems do not have any programmatic mechanism to enforce the restrictions.<br>If email notification is configured, non-compliance is reported in email. For details about compliance emails, see Configuring compliance emails . |

| OS | How Enforced |
|---|---|
| Android with Samsung KNOX | Disallowed applications (either blacklisted or not whitelisted) are blocked or removed from the device.<br>If email notification is configured, non-compliance is reported in email. For details about compliance emails, see Configuring compliance emails . |

If your device policy checks compliance against the whitelist, then applications not on the whitelist cannot be run on the device, subject to the device's operating system constraints.

The behavior of blacklisting or whitelisting is different for Good-based applications (those that have a GD App ID) and non-Good-based applications:

- Apps added to the whitelist are displayed in the user-accessible application catalog and in case of Good-based apps are permitted to run.
- However, apps added to blacklist are *not* displayed in the user-accessible application catalog and in the case of Good-based apps are *not* permitted to run.

## Steps for blacklisting or whitelisting

The steps for blacklisting and whitelisting are nearly identical. You need to know the following:

- Android: The package name of the application
- iOS: The bundle ID of the application
- The device policy you want to use to apply the lists

The steps have the following general parts:

- Defining the blacklist or whitelist: **Manage Apps** > **Blacklist** tab or **Whitelist** tab
- Applying the list in a device policy: **Device Policies** > *edit a policy* > **General** > **Check compliance against App Blacklist or App Whitelist**

1. Navigate to **Manage Apps**.
2. Click either the **Blacklist** or the **Whitelist** tab.
3. Click **Add App**.
4. Click either **Android App** or **Apple iOS App**.
   - For Android applications, enter the package name.
   - For iOS applications, enter the bundle ID.
5. Click **Blacklist** or **Whitelist**, or click **Cancel** to discard your changes.
6. Navigate to **Device Policies** > *edit a policy* > **General**
7. Click **Edit**.
8. Find the setting: **Check compliance against**
9. Make sure the **ON** radio button is active.

10. From the pulldown select either **App Blacklist** or **App Whitelist**.
11. Click **Save** to save your changes or **Cancel** to discard them.

### Steps for removing apps from blacklist or whitelist

1. Navigate to **Manage Apps**.
2. Click either the **Blacklist** or the **Whitelist** tab.
3. On either the **Blacklist** or the **Whitelist** tab, to select all Android applications or all iOS applications, click the appropriate checkbox above the list, or scroll through the list to checkmark the desired applications.
4. Click **Remove App**.

## Essential one-time setup tasks

Here are administrator's tasks in preparation for implementing application management. In general, you need to do these tasks only once.

### Whitelisting app stores and web servers in Good Control

To allow your end-users' device to access applications on the Google Play Store, Apple App Store, or web servers, you need to "whitelist" the hostnames and ports for these resources in Good Control.

To whitelist these stores, in Good Control, add the hostname and port values to the `proxy.urls` property in **Servers** > **Server Properties** tab. For more details, see the Good Control online help.

| Required? | Resource | Hostname and Port | Notes |
|---|---|---|---|
| Required | Apple App Store | store.apple.com:80 | For retrieving applications' associated images |
| Required | Google Play Store | play.google.com:443 | For retrieving applications' associated images |
| Required if you are serving Web applications | Web Applications | Exact hostnames and ports depend on the web servers host your applications, either on the public Internet or on your private intranet | When you add new web applications, check that their details have been whitelisted. |
| Optional depending on networking configuration | Good's App Store | appstore.good.com:443<br>good.com:80 | Needed if you have enabled the Route All feature, which directs all network traffic throght the Good Proxy. For more info, see the Good Control help topic External Web Proxy . |

## Entitling users to the application catalog

For your end-users to see the application management application catalog, they need to be entitled to it. This entitlement is done via a "placeholder application" name in Good Control's application policies and application groups:

- Application Name: **Feature – AppStore**
- BlackBerry Dynamics App ID: **`com.good.feature.appstore`**

**Note:** Unless you want to allow access to the application management catalog to only a subset of your end-users, BlackBerry recommends that you entitle all end-users via **App Groups** > **Everyone**, to which all end-users are automatically added. Otherwise, entitle only the groups you want.

**To entitle end-users to the application management "virtual application" in Good Control:**

1. Navigate to **App Groups.**
2. Checkmark the group you want to entitle, such as Everyone.
3. Click the pencil icon on the far right of the group name to edit it.
4. Under **Allowed Applications**, click **Add More**.
5. From the displayed list of applications, find and select the "placeholder application" named:

   **Feature – AppStore**

6. Click **OK** to save your changes or the large **X** in the upper right to discard them.

# Adding applications

These are the steps for adding an application to application management.

## About unique names for apps

In general, be sure you have unique names (display names or other) for all your applications. The name of an app is used to distinguish it from other applications and in many cases its uniqueness is the only mechanism available to Good Control to make this distinction.

## App description or "Notes" field visible to all end-users

An optional description is one of the fields you can enter when you add an application version. This is displayed in the UI as the **Notes** field.

> **Note:** Be aware that any text you enter in the description of **Notes** field is visible in the GC console to all end users entitled to the application.

## Adding a public store application

You need the following:

- For Good-Dynamics-based applications, the BlackBerry Dynamics App ID and application version for the application must have been compiled into the application binary.
- The URL to the application's "landing page" or "preview page" in either Apple App Store or Google Play Store

> **Important:** If the public app store is down or its interfaces are not available or not responsive, Good Control cannot retrieve details from it.

**In Good Control:**

1. Navigate to **Manage Apps**.
2. Click the **Enterprise** tab.
3. Click **Add App**.
4. From the dialog, click the radio button for **Public App Store**.
5. Click **Next**.
6. Enter the URL to the public app store for this public app.

   > **Important:** The URL for a public store app must be unique by platform. You cannot reuse the same URL.

7. Click **Cancel** to discard or **Next** to continue.

   GC displays information about the application: its version, operating system, form factor, size, and (for Good-based applications) BlackBerry Dynamics App ID and application version.

8. Click **Back** to select a different URL, **Add App** to finish, or **Cancel** to discard.

### About adding GFE

BlackBerry for Everyone (GFE) is a popular BlackBerry application that was created before BlackBerry Dynamics. As such, GFE does not have a BlackBerry Dynamics App ID or application version number, as do BlackBerry Dynamics applications.

Because of this, for distribution via Good Control, GFE must be added as a public store app.

## Adding a custom application

You need the following:

- The BlackBerry Dynamics App ID and application version for the application, if it is Good-based
- The application's compiled binary file, either Android package (.apk) or Apple bundle (.ipa).

In Good Control:

1. Navigate to **Manage Apps**.
2. Click the **Enterprise** tab.
3. Click **Add App**.
4. From the dialog, click the radio button for **Custom**.
5. Click **Next**.
6. Click **Choose File**.
7. Navigate your computer to select the desired binary: Android package (.apk), Apple bundle (.ipa).
8. Click **Add App** to upload or **Cancel** to discard.

   GC displays information about the uploaded binary: its version, operating system, form, size, BlackBerry Dynamics App ID and application version.

9. Click **Back** to select a different file, **Cancel** to discard, or **Add App** to finish.

## Adding a web application

You need the following:

- The URL to the application's "landing page" or "preview page" on a web server

In Good Control:

1. Navigate to **Manage Apps**.
2. Click the **Enterprise** tab.
3. Click **Add App**.
4. From the dialog, click the radio button for **Web**.
5. Click **Next**.

6. Enter the URL to the application, with the protocol either **`http://`** or **`https://`** (default).
7. Click **Cancel** to discard or **Next** to continue.
8. Verify the displayed details:
   - Edit the displayed text, if desired.
   - To upload your own icon in place of the displayed one, under the icon click UPLOAD and follow the leading prompts.
9. Click **Back** to specify a different URL, **Cancel** to start over, **or Add App** to finish.

## Adding BlackBerry Dynamics app IDand version only

You need the following:

- The BlackBerry Dynamics App ID and application version for the application

In Good Control:

1. Navigate to **Manage Apps**.
2. Click the **Enterprise** tab.
3. Click **Add App**.
4. From the dialog, click the radio button for **BlackBerry Dynamics Entitlement and Version Only**.
5. Click **Next**.
6. Enter the values for the following fields.
   - Display name. This name must be unique among the apps that you manage.
   - BlackBerry Dynamics Entitlement ID. For details on acceptable values, see Key concepts .
   - BlackBerry Dynamics Entitlement Version. For details on acceptable values, see Key concepts .
   - Custom Description displayed in the GC console.
7. Click **Add App** to finish or **Cancel** to discard.

### Specifying app servers

If you have a BlackBerry Dynamics-based application (one with a BlackBerry Dynamics App ID and version) that is served from an application server or web server, you can specify the name of that application server and the priority of the Good Proxy clusters used for communication with it.

**To specify an application server and its GP cluster priority for a BlackBerry Dynamics-based application, in Good Control:**

1. Navigate to **Manage Apps** > *edit an application* > **BlackBerry Dynamics** tab.
2. For **Host Name**, specify the fully qualified domain name of the application server where this application is.
3. Specify any required port number.
4. For **Priority**, select one of **Primary**, **Secondary**, or **Tertiary**.
5. For **Primary GP Cluster**, from the pulldown menu, select the name of the desired cluster.

6. For **Secondary GP Cluster**, from the pulldown menu, select the name of the desired cluster.
7. To add another row, under **Action**, click the plus sign (+), and repeat the steps above as many times as needed.
8. For the **Configuration** field, see the discussion below.
9. Click **Save** to retain your changes or **Cancel** to discard them.

#### Configuration field

In the **Configuration** field you can add text in the format reqquired by the application developer (typically JSON/XML). This configuration is sent to all the clients for any user; that is, it is a global setting.

The **Configuration** field is an older mechanism for passing initialization or other information that should be passed to the application when it starts. The preferred mechanism is application-specific policies, described in Configuring Application Specific Policy Rules . Application-specific policies allows for configuration to be user-group-, the administrator does not have to worry about formatting in JSON/XML.

## Managed apps: enabling app auto-push, exempting policy sets

With Good Control's auto-push feature, you can enforce changes to apps on your end-users' devices, such as automatically pushing the latest version of an app or removing disapproved versions of apps:

- The app auto-push feature is available for all app types except web apps. Note that for other types of apps, the auto-push option is displayed in Good Control only if the app has an associated binary executable file uploaded either to the GC (a custom app) or to one of the public app stores ( a public app store app).
- About auto-push of purchasable applications: GC does not prevent you from auto-pushing an app that must be purchased (from an app store or otherwise). The status in the GC of a purchasable app that has been pushed to a device is: **Payment Required**.
- GC permissions that include the auto-push feature: Applications, Shared Services, and Application Wrapping permissions.
- Except for specific policy sets that you exempt, the auto-push of an app is applied to all policy sets that include devices policies for the desired platform.
- Supported device operating systems:
  - With BlackBerry Agent for iOS: iOS 8.0 or later.
  - Android (minimum API Level 14) with Samsung KNOX(minimum KNOX 2.1)

### Prerequisites

For auto-push, the end-user must have been associated with a policy set in Good Control that includes at least one device policy for the desired platform, and the end-user’s device must be enrolled in BlackBerry device management. If the device is not enrolled, apps cannot be pushed to it:

- To use app auto-push, BlackBerry device management must be in effect in Good Control. See Device Management Administrator's Workflow.
- The policy sets that enforce the auto-push must have an associated device policy that includes the platforms to which you want to auto-push the app.

- You must have already done the basic set-up of a public store or custom app in Good Control, as shown in Application Management Administrator's Workflow.
- To configure auto-push, a user of Good Control must have the **Applications, Shared Services, and Application Wrapping** permission.
- If you want to exempt certain policy sets from auto-pushing the app, determine the names of those policy sets.

**To enable auto-push of an app to end-users' devices, in Good Control:**

1. Navigate to **Manage Apps** > **Enterprise tab** > *edit an app* > **General tab** > **Auto-Push Settings**.
2. Click **Edit**.
3. Check the **Auto-Push Enabled** checkbox.
4. If you want to exempt policy sets from enforcing this auto-push, click **Add Policy Set**.
5. In the displayed list of policy sets, check those that you want to exempt from auto-pushing this app to devices.
6. Click **Add** to exempt these policy sets, or **Cancel** to discard your selection.
7. Click **Save** to retain your changes to this app, or **Cancel** to discard them.

## Behavior on iOS

Noted here are some behaviors of auto-pushed apps on iOS.

### Cannot auto-push on top of unmanaged app

If a version of an application that is not managed by the GC is already installed on a user's iOS device, an attempt to auto-push a later, managed version of the app will fail.

**Workaround:** Delete the previously installed, unmanaged version of the app from the device, and then auto-push the later, managed version.

### Duplicate Apple ID on multiple devices

If a user uses his Apple ID (the ID for logging into the Apple Store) on more than one device, one of which is enrolled in BlackBerry device management, and with sync enabled, managed apps that are auto-pushed to the enrolled device are also pushed to the other device.

If the enrolled device is subsequently unenrolled, the auto-pushed apps are removed from formerly enrolled device but not removed from the other device.

### Remove iOS MDM profile: auto-pushed apps are deleted

On iOS, the end-user always has the ability to remove any device management profile imposed the device.

If the iOS end-user removes the installed profile, any apps that have been auto-pushed to the device are also removed.

# Entitling end-users to applications or denying them

Your end-users must be entitled to view or run the applications defined in the application catalog. You can also deny them the right to applications. You can entitle or deny end-users in several ways:

- WIth app groups
- Per individual end-user

## Sequence of app version entitling and denying: entitle, then deny

**Important:** If you are entitling a new app version and denying an older version, be sure to entitle the new version first before you deny access to the older version. If you deny the older versions first, the app will be wiped from the device.

## Entitling or denying end-users via entitlement groups (aka app groups)

By default, Good Control comes with the Everyone group, to which end-users are added automatically. The easiest way to entitle all your end-users is to entitle the Everyone group.

You might have the need for different end-user groups for finer control over which end-users can use which applications. In this case, entitle the appropriate user groups for just those applications you want them to use.

**To entitle or deny via the Everyone group:**

1. In Good Control, navigate to **App Groups**.
2. Edit the appropriate group by clicking the edit icon (pencil) on the far right of the group name.
3. Under either **Entitled Enterprise Apps** or **Denied Enterprise Apps**, click **Add More**.
4. From the displayed dialog, you can select applications in several ways, some combinations of which are mutually exclusive. Choose the desired ways:
   - From the **View** pulldown, select the type of application to show: **All**, **Organization**, **Partner** or **Good**.
   - If desired, click the **Show dev versions** checkbox.
   - In the text box, enter the name of the application you are looking for.
5. After finding the desired application, you can click the triangle left of its name to see the registered versions of the application.
6. Click the checkbox for **ALL** or the individual checkboxes for only the desired versions.
7. Click **OK** to save your changes or the **X** in the upper right of the dialog box to discard them.

## Entitling or denying an individual end-user

In Good Control's **Manage Users** screen, you can manage various aspects of users in bulk (that is, more than a single user at a time), but to entitle or deny an end-user an application, you can operate on only a single user at a time.

You can entitle the end user by way of app groups or by entitling the end-user individually.

**To entitle or deny a single end-user:**

1. In Good Control, navigate to **Users and Groups**
2. You can filter users in several ways. Choose the desired ways:
   - From the **Filter users by** pulldown, select **Policy Set**, **Application Group**, or **Directory Group** (Active Directory group). Then from the additional pulldown menu, choose the specific policy set, app group, or enter the name of

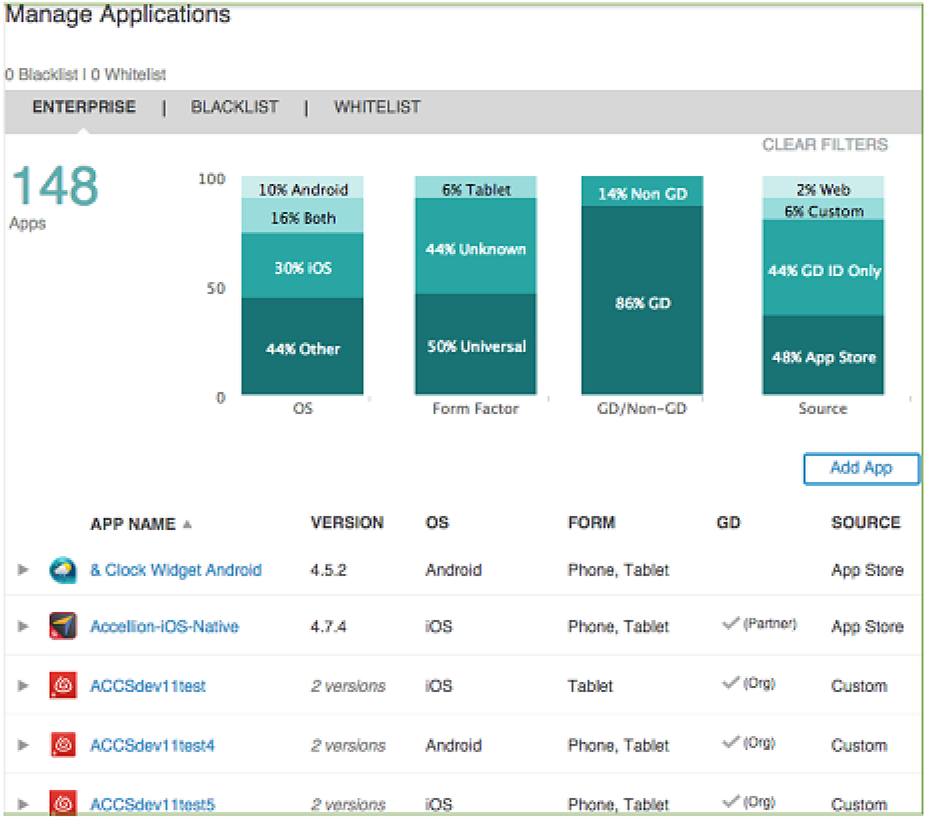
the specific AD group.

- In the text box, enter the name or email address of the desired end-user.

3. After finding the desired end-user, click the checkbox left of the end-user's and in the upper right, click **Edit**.
4. If you want to assign the end-user to a previously defined an app group. click **App Group**, scroll to find the desired group, and click Save.
5. 1. Under either **Entitled Enterprise Apps** or **Denied Enterprise Apps**, click **Add More**.
   2. From the displayed dialog, you can select applications in several ways, some combinations of which are mutually exclusive. Choose the desired ways:
      - From the **View** pulldown, select the type of application to show: **All**, **Organization**, **Partner** or **Good**.
      - If desired, click the **Show dev versions** checkbox.
      - In the text box, enter the name of the application you are looking for.
   3. After finding the desired application, you can click the triangle left of its name to see the registered versions of the application.
   4. Click the checkbox for **ALL** or the individual checkboxes for only the desired versions.
   5. Click **OK** to save your changes or the **X** in the upper right of the dialog box to discard them.

## Filtering the list of applications, viewing the bar chart

In Good Control's **Manage App** screen, some details about applications that have been put under management are listed and summarized in graphic form.



| APP NAME ▲ | VERSION | OS | FORM | GD | SOURCE |
|---|---|---|---|---|---|
| & Clock Widget Android | 4.5.2 | Android | Phone, Tablet | | App Store |
| Accellion-iOS-Native | 4.7.4 | iOS | Phone, Tablet | ✓ (Partner) | App Store |
| ACCSdev11test | *2 versions* | iOS | Tablet | ✓ (Org) | Custom |
| ACCSdev11test4 | *2 versions* | Android | Phone, Tablet | ✓ (Org) | Custom |
| ACCSdev11test5 | *2 versions* | iOS | Phone, Tablet | ✓ (Org) | Custom |

## Details in list view

For each managed application, the following fields are shown in the list:

- **App Name**: field is sortable in ascending or descending order.
- **Version**: latest version to have been uploaded or put under management.
- **OS**: operating system, either **Android** or **iOS**.
- **Form**: the form factor (that is hardware types the app runs on), **Phone** or **Tablet**.
- **BlackBerry Dynamics**: Whether the app is Good-based, with details as follows;
  - A blank means that the application is not Good-based.
  - A **checkmark (Org)** means that your own organization originated the Good-based app.
  - A **checkmark (Partner)** means that a BlackBerry partner company or Independent Software Vendor (ISV) originated the Good-based app.
- **Source**: Either **App Store**, **Custom**, **Web** or BlackBerry Dynamics App ID only. In the case of the Apple App Store (not the Google Play Store), applications that are purchasable show the price, or if not purchasable, show **Free**.

## Filters

At the top of the **Manage Apps** page are several filters that you can use to restrict the data summarized in the bar chart and the list of applications beneath the graphic, from left to right.

| Filter Name | Description |
|---|---|
| Filter: Name/BlackBerry Dynamics App ID | Enter either the name of the application or its BlackBerry Dynamics App ID |
| All OSs | Select from:<br>• All OSs<br>• Android<br>• iOS |
| All Form Factors | Select from:<br>• All Form Factors<br>• Phone<br>• Tablet |
| All Apps | Select from:<br>• All Apps<br>• BlackBerry Dynamics apps<br>• Non-BlackBerry Dynamics Apps |
| All Sources | Select from:<br>• App Store<br>• Custom<br>• Web<br>• BlackBerry Dynamics App ID |

The result of your selection is:

- The bar chart is redrawn to show the percentages of data that match the selections you made.
- The list of applications beneath the bar chart is constrained to include only data that matches the selections you made.

To clear the selections after you have selected filters, in the upper right, click **Clear Filters**.

# Updating apps

See the recommendations in About Updating Applications.

Described here are the steps for updating applications that have been added to application management. "Updating applications" means changing the application itself, such as uploading new binary executables, as opposed to changing details about the application, which is described in Editing application details .

**Note:** After updates are done in Good Control, it can take up to five minutes for the updates to appear in the end-user accessible application catalog.

## Updating a public store app: work in public store, refresh in Good Control

Because the binary executable files for public store applications are stored in the public stores themselves, your work related to updating public store apps has two general parts:

1. Upload new binary versions to the affected public app store and supply other details required by the store.
2. In Good Control, refresh the metadata for the affected application. See details in Editing application details .

**Important:** If the public app store is down or its interfaces are not available or not responsive, Good Control cannot retrieve details from it.

## Updating a custom app: upload new binary

Avoid uploading older application versions. The system allows you to upload older versions of an application (one whose BlackBerry Dynamics application version is older than the application version already under managed control). Although this is possible, it is not best practice. You should use BlackBerry application management to distribute the latest version of an application, not old versions.

To update a custom application's binary executable file:

1. Make sure you have the new binary executable file you want to upload and that it has a different version number than the binaries already in the system.
2. In Good Control, navigate to **Manage Apps** > **Enterprise** tab.
3. Scroll to find the application you want, or use the BlackBerry Dynamics App ID or application name in the filter in the upper left, or sort the list of applications by descending or ascending application name.
4. Click the name of the application.
5. In the upper right, click **Update App**.
6. In the displayed dialog, click **Choose**, and navigate your computer to find and select the desired binary executable file.
7. Click **Cancel** to discard the update, or **Update App** to continue.

## Updating a web app: add new web app

A web application has no manageable binary executable associated with it.

If the URL for a previously added web application changes, define a new web application for it, as described in Adding a web application . Each unique URL is considered a unique web application.

## Updating a BlackBerry Dynamics-app-ID-only app: convert to public store or custom app

By definition in Types of applications a BlackBerry Dynamics-App-ID-Only initially has no associated binary executable file. However, after the executable binary is ready, you can update the previously defined BlackBerry Dynamics-App-ID-Only application to be a public store or custom app.

**To convert a BlackBerry Dynamics-App-ID-Only application to public store or custom app:**

1. Build the executable binary for the BlackBerry Dynamics-App-ID-Only app, using the same BlackBerry Dynamics App ID and application version values that you originally defined with the application was created in Good Control.
2. For public store applications, post your application to the appropriate store.
3. For custom application, have the executable binary ready to upload.
4. In Good Control, navigate to **Manage Apps**.
5. Find the previously defined BlackBerry Dynamics-App-ID-Only application in the list.
6. Click the name of the application.
7. In the upper right click **Update App.**
8. Click the radio button for either Public Store App or Custom App.
9. Click **Next**.
10. For a public store app, specify the details required.
11. For a custom app, click Upload, browse your computer to find the executable binary, and upload it.

### About application versions

Versions of applications (either native bundle version or BlackBerry Dynamics Entitlement version, as described in About BlackBerry Dynamics entitlement ID and version ) that you have published are listed under an application's name in the GC console. Even if you delete the binary executable associated with an app, the version numbers are retained. This provides a historical record of your publishing and is a parallel to similar behavior for versioning in the public stores.

## Adding multiple platforms for public store apps

Imagine you have an application available for two or more different "platforms" (hardware types, or form factors), such as the same application for the iPhone and the iPad or the iPhone and Android devices. You want to give your users access to the applications of both platforms or form factors.

Prerequisites:

- Make sure you have added at least one of the platforms or form factors to BlackBerry application management.
- You need the URL to the public app store for each of the desired platform-specific versions of the application.

**In Good Control:**

1. Navigate to **Manage Apps**.
2. In the list, find the desired application you want to add form factors for and click its line in the list.
3. In the upper right, click **Add URL**.
4. In the displayed dialog box, enter the appropriate URL to the public app store.
5. For BlackBerry Dynamics-based apps, choose either one of the already existing BlackBerry Dynamics application

versions or click the radio button for **New BlackBerry Dynamics App Version** and enter the new version number.

6. Click **Next** to continue or **Cancel** to discard your changes.

# Blocking Android or iOS BlackBerry Dynamics apps by native version

In Good Control's **Manage Apps** screen, you can selectively block access to specific versions of your BlackBerry Dynamics-based application on Android or iOS. The blocking of the app on the device is sent from Good Control in the form of a compliance policy. To uniquely identify an app, the GC admin denies via the app's BlackBerry Dynamics Entitlement ID (also known as "BlackBerry Dynamics App ID") and a native version identifier, which can include a wildcard. For definitions, see Distinction from and use with native language identifiers .

> **Note:** Only Android or iOS apps can be blocked by native version. Windows is not supported.

The blocking only affects the BlackBerry Dynamics Runtime of the app on the device, blocking it so it cannot run. It does not prevent the end user from downloading and installing the latest binaries of the app that are allowed on the device. An end user who attempts to install a blocked version of an app sees the following message:

**`The version of`** *`<appname>`* **`is blocked. An updated version is available.`**

Unless you want to completely block access to the app regardless of its version, be sure your end users are entitled to a later version of the app *before* you deny access to an older version they might also have on their devices. If you deny the older first before entitling, the app is wiped from the device.

## Wildcarding native versions

You can use the * wildcard character with the native version identifier to deny a certain range of versions. Follow the vendor recommendations in Distinction from and use with native language identifiers .

The * wildcard character:

- Must come last in the native version string. **Invalid usage:** 1.*.8
- The closer the * is to the left, the more versions it masks. **Example:** 2.* denies 2.1, 2.2, and 2.3.

## Steps

**Prerequisites**

- You need to know the exact BlackBerry Dynamics Entitlement ID (BlackBerry Dynamics App ID) of the Android or iOS app whose native version you want to block
- You need to know the native versions of the app you want to block.

**To deny specific versions of an application, in Good Control:**

1. Navigate to **Manage Apps** > **Enterprise tab** > *edit the appropriate app* > *platform-specific* tab.
2. For the heading **Blocked Versions**, click **Edit**.
3. Enter the native versions to deny, separated by commas, be sure that any * wildcard you use comes last in the version identifier.
4. Click **Save** to retain your change or **Cancel** to discard them.

## Editing application details

There are several different tabs where you can edit details about the application. The displayed tabs depend on the type of application.

You can edit the details for all application types, including the BlackBerry Dynamics App ID and application version.

The app stores are the source of details for public store apps; after updating details in the app store, in Good Control, you refresh the metadata for the affected app, as described in the steps below.

BlackBerry application management uses APIs from Apple to retrieve details about iOS applications in the App Store. However, because Google does not provide a callable API to retrieve details from the Google Play Store, BlackBerry application management attempts to collect these details by analysis of the Google Play Store pages themselves.

The complete set of tabs is as follows.

| Tab Name | Editable Details |
|---|---|
| General | Application name, icon, and description. The fields vendor, source, and minimum OS are also displayed. |
| Android | Description, release notes, package name and versions, and screenshots |
| iOS | Description, release notes, package name and versions, and screenshots |
| BlackBerry Dynamics | • BlackBerry Dynamics App ID, and corresponding fields for iOS, Android, and Microsoft Windows.<br>See also About BlackBerry Dynamics entitlement ID and version .<br>• Android Package ID<br>• Apple iPad Bundle ID<br>• Apple iPhone Bundle ID<br>• Windows Phone Application ID<br>• Windows Application ID<br>• Policy Set Override<br>• Server configuration for primary and secondary GP clusters |

| Tab Name | Editable Details |
|---|---|
| | • Versions, including:<br>  • Release status: development or production<br>  • Alternate URL for Welcome email<br>  • Service names and bindings |
| Configuration | Upload application policy in XML format. For details, see the Good Control online help. |

### General steps

This guide does not detail the exact steps for updating all available fields, because their meanings are clear. The general process to edit details for an application in the application catalog:

1. For public store apps, be sure to update details about the app in the public store itself, where the details are stored.
2. In Good Control, navigate to **Manage Apps** > **Enterprise** tab.
3. Scroll to find the application you want, or use the BlackBerry Dynamics App ID or name in the filter in the upper left, or sort the list of applications by descending or ascending application name.
4. Click the name of the application.
5. On the **General** tab:
6. For public store applications, click **Refresh Metadata** to pull the latest details from the appropriate app store.
7. On the **General** tab for all other application types, find the block that includes the details you want to change, and click **Edit**.
8. Click **Cancel** to discard your changes or **Save** to save them.
9. Repeat the previous two steps for the other tabs.

## XML Format for Application Policies

Application-specific policy rules must be written in XML format. For details about the XML format, see this technical paper, this sample XML file, this XML schema definition you need to validate your policies, and this explanation of the schema and application policies in general.

## Deleting a managed application

To remove the accessibility to an application, delete the application.

The exact effect of removing an application from managed apps depends on its type. If the application is Good-based, and thus has a defined BlackBerry Dynamics Entitlement ID and version, the application is removed from the application catalog (appstore) and wiped from users' devices. Otherwise, the application is merely removed from the catalog but left intact on end-users' devices. Likewise, removing a web application from the catalog has no effect on the web application itself.

**Note:** If your GC is in development mode, you cannot delete a production app. This restriction is to prevent the inadvertent deletion of a production app by a development team.

Good Control operates in two modes: development and production. (By default, at installation, a GC runs in development mode. A production GC is one in which the administrator has set a production license.) Likewise, the status of an apps is marked as production or development.

Non-BlackBerry Dynamics apps (apps without a BlackBerry Dynamics Entitlement ID) are always considered as production.

**To wipe app data, in Good Control:**

1. Navigate to **Manage Apps** > **Enterprise** tab.
2. Scroll to find the application you want to remove from the catalog or sort the list of applications by descending or ascending application name.
3. Click the name of the application.
4. On the displayed **General** tab, in the upper right click **Remove App**. The **Remove App** button is displayed only if the conditions described in the note above are met.
5. Click **OK** to confirm the deletion or **Cancel** to keep it.

# Manage Services

GD application developers can save time and effort by taking advantage of functions already provided by other application services. With the GD SDKs for iOS and Android, application developers can expose aspects of their GD applications that other developers can use in their own GD applications. In addition, server-based applications can offer shared functions that GD application developers can use. Shared functions offered by a mobile application or server-based application are referred to in the GC console as an “application service”. For an application to properly use a service from another mobile application, both applications must be installed on the same device.

Developers in your organization can publish an application service in the GC console, supply a service definition that describes the service in JSON format, and bind the service to the applications that provide them. Other developers in your organization can then read the service definition and make use of the service in their GD applications.

Some Good and Partner applications can also offer application services for your developers to use. The full list of available application services can be found on the **Manage Services** screen of the GC console.

For more information on registering and configuring new application services, see Registering a new service and Binding a service version to an entitlement version.

Information on application service development is available on the BlackBerry Developer Network (BDN) portal.

# Viewing Registered Services

In the main navigation, click **Apps** > **Manage Services** to view a list of all application services currently registered with GC. Services provided by your organization's mobile and server-based applications are displayed first, followed by the services provided by Good and Partner applications. From this screen, you have the option to delete any of your organization's registered services by clicking the corresponding ✖ **Delete** icon or to register a new service by clicking the ⊕ **Add** icon beside the total service count. For more information, see Registering a new service and Removing a service.

Click the ✎ **Edit** icon for an application service to modify it or view more information.

You can edit most of the information for your organization's application services except for the ID, because the service ID is cannot be changed after it is registered. You also return to this screen if you need to add another version of the service.

On the edit screen for a Good or Partner application service, no details can be modified.

Related information can be found in the Managing service versions and Binding a service version to an entitlement version topics.

# Registering a New Service

The following information is required to register an application service:

- Service type. You must specify whether the service is offered by a GD mobile application or by an application on a server.
- Name of the service.
- ID for the service. The ID is a unique string in reverse DNS notation and must consist of all lowercase letters separated by dots (e.g., com.good.service.print).
- Version identifier of the service. Versions consist of digits only, and are period delimited when applicable to show build numbers or other information. For example, valid version numbers include **2**, **2.3**, **2.3.0**, and **2.3.0.1**. Leading zeros are not allowed, so **2.03** is not a valid version number.
- Service definition in JSON.

**To register a new service in Good Control:**

1. Click **Manage Services** in the main navigation. A list of all application services currently registered with GC is displayed.
2. In the upper right, click the + icon. GC then displays the following screen.
3. Enter the required information for the new service. You can also specify optional information such as the description of the service and version or the interface format if the service is provided by a server-based application.
4. On completion, click **Add Service**. You are directed to the screen to manage the new service.

The service is now registered for your organization. Your next step is to bind the service version to an entitlement version so the GC console can advertise that the entitlement version provides this particular service. For more information, see Binding a service version to an entitlement version.

# Managing Service Versions

### Adding a Version

You can add versions only for services that have been developed and registered by your organization. Good and Partner services cannot be modified through the GC console.

When you initially register a service, you must specify some information about the first version. If you need to add a new version later, return to the screen to manage the service.

Above the list of versions, click **Add a Version**. On the next screen, shown , enter the new version number identifier and supply the service definition in JSON format. You can also specify an optional description, or an interface format definition if the service is provided by a server-based application.

Versions consist of digits only, and are period-delimited when applicable to show build numbers or other information. For example, valid version numbers include **2**, **2.3**, **2.3.0**, and **2.3.0.1**. Leading zeros are not allowed, so **2.03** is not a valid version number.

The service definition must be in JSON format. Other developers refer to this definition when developing applications that relies this particular service.

### Modifying Version Information

You can modify information only for versions of application services registered by your organization. Good and Partner services cannot be modified through the GC console.

At any time, you can return to the version management screen to modify the description and JSON definition for the service. The ID is fixed when the service is registered and cannot be edited.

### Deleting a Version

You can delete a version of a service registered by your organization in two ways:

1. While on the edit screen for the service, click the trash can icon for a version to remove it from the system.
2. While on the edit screen for the service version, click the **Delete** at the top of the screen.

Deleting a service version has no effect on the applications that offer the service or the applications that use the service. Only the definition of the service version is deleted, so that developers cannot refer to it and so that the GC console cannot advertise that any entitlement versions provide that version of the service.

Only versions of services that are registered by your organization can be deleted. Good and Partner service versions cannot be deleted through the GC console.

## Binding a Service Version to an Entitlement Version

The Good Control console advertises the entitlement versions that provide services so that application developers who want to take advantage of services can see exactly which entitlement versions provide those services. Remember that registering services and binding them to applications through the GC console is merely a convenient method of association for developers to refer to; whether or not the services are registered and bound has no effect on the applications' behavior.

Both the entitlement version and service version must be registered through the GC console before you can bind them together. For more information, see Managing Service Versions .

After both are registered, you can do the following steps to bind them together.

**To bind a service to an application in Good Control:**

1. Navigate to **Manage Apps** > *edit an application* > **BlackBerry Dynamics** tab.
2. Go to the edit screen for the entitlement version that provides the service under the **Versions** heading.
3. Under the **Bind** heading, GC displays a list of services already bound to the entitlement version, if any.
4. Click the + **Bind Service** icon to view a list of registered service versions not yet bound to the application.
5. Select the applicable service versions from the list and click **OK**. The service versions you selected are now bound to the entitlement version and are added to the list under he the **Bind** heading.

## Removing a Service

Deleting a service has no effect on the applications that offer the service or the applications that use the service. Only the definition of the service version is deleted, so that developers can not reference it and so that the GC console cannot advertise that any entitlement versions provide that version of the service.

Only application services registered by your organization can be deleted. Public (from BlackBerry or Partner) application services cannot be deleted through the GC console.

**To delete an application service in Good Control:**

1. Navigate to **Manage Services**.
2. Find the desired service.
3. Click its associated trash can icon.
4. Click **OK** to delete or **Cancel** to retain the service.

# App Groups

Application groups provide an easy way to apply the same base application permissions to many users.

When GC is installed, an Everyone application group is automatically created. All GC users belong to this group, so a quick way to grant or deny permission to an application for all users is to set the application permissions for the Everyone group.

You can also make a new group, set allowed and denied applications for the group, and add users to the group in bulk; each user immediately inherits permissions of the new group.

The following rules apply to application groups:

- Users can belong to multiple groups.
- If a user belongs to more than one group, the most restrictive permission applies.
- User-level permissions always override group-level permissions.

### Users in multiple groups

Because users can belong to multiple groups, a user might inherit conflicting permissions. When this happens, the most restrictive permission applies. For example, if a user belongs to three groups, and a certain entitlement version is: a) denied by one of the groups, b) allowed by the second group, and c) has no permission explicitly set for the third group, then the application is denied for the user at the group level.

Explicitly set user level permissions always override group level permissions.

## Viewing and Deleting Groups

Click the **Application Groups** navigation item to view the current list of application groups.

From this screen, you can ✏ **Edit** or ✖ **Delete** groups. You can edit details for any group. You can delete any group created by a GC administrator, but you cannot delete the Everyone group because it is the default group all users belong to.

Be mindful of how application permissions are applied when deleting a group, because this can have a large impact on your users. For example, if a number of users are members of a certain group and access to a GD application has been granted to the group, after the group is deleted, those users lose access to the application.

## Creating a New Application Group

Application groups are an easy way to apply the same base application permissions to many users. You can make a new group, set allowed and denied applications for the group, and add users to the group in bulk; each user immediately inherits permissions of the new group.

**To create a new application group in Good Control:**

1. Navigate to **App Groups** to view the current list of application groups.
2. In the upper right, click the + icon.
3. Enter a name for your new group
4. Click **Create Group**.

5. GC then displays a screen for the new group, where you can add users to the group and apply application permissions.

Read the following topics for more information on how to configure your application groups: Managing application permissions for a group and Managing the list of users in a group.

# Managing Application Permissions for a Group

Application groups are an easy way to apply the same base application permissions to many users.

In Good Control, on the **App Groups** edit screen for a group, make sure the **Apps** tab is active.

## Sequence of App Version Entitling and Denying: Entitle, Then Deny

**Important:** If you are entitling a new app version and denying an older version, be sure to entitle the new version first before you deny access to the older version. If you deny the older versions first, the app will be wiped from the device.

## Entitling

To grant permission to an product or entitlement version, click the + **Add More** icon for the **Entitled Enterprise Apps** list. A panel appears with a list of applications and entitlement versions not yet permitted or denied for the group. If the list is long, you can use the filter to limit the list. You can also use the pulldown to view only Organization applications or Good or Partner Applications. The following image shows an example of this panel.

Click an application or version to view its description in the Details box, and select it by checking its checkbox. Select the **- ALL** item for an application to grant permission for all versions of the application (including all future versions), or select each required version if you do not want to grant access for all versions. Click **OK** to apply your changes.

## Denying

To deny an entitlement version, click instead on the + **Add More** icon for the **Denied Enterprise Apps** list and follow the same instructions.

**Note:** The same application can show up in both the allowed and denied lists. This is because permissions can be applied at the version level, in addition to the application level, so some versions of the application can be allowed and others can be denied. You can expand the application in both lists to view both allowed and denied versions.

# Managing the List of Users in a Group

Application groups are an easy way to apply the same base application permissions to many users.

You cannot add or remove users from the Everyone group, because Everyone always contain every GC user. The following information involves managing the list of users in application groups that you or another GC administrator has created.

You can add users in bulk to a group, and each of the users you add immediately inherits the permissions applied to the group. You can remove individual users from a group if the user no longer requires the group's set of permissions.

There are several ways you can manage the list of users in an application group. You can modify group membership for multiple users at once, you can directly modify a group to include a new list of users, or you can go to a user's account management screen and modify the list of groups that user belongs to.

### Modifying group assignment for multiple users

This information has been moved to a separate topic: Modifying user accounts.

### Directly modifying the list of users in a group

While on the edit screen for a group, select the **Members** tab.

To add group members, click the + **Add** icon, and a panel appears with all GC users not already in the group. If the list is long, you can use the filter to limit the list. Check the checkboxes next to each of the users you want to add to the group, and click **OK**. The screen is updated to display the new list of group members, and the new group members now inherit application permissions from the group.

To remove a group member, click the trash can  icon for the user you want to remove. The user is removed from the group and loses any application permissions previously inherited from the group.

### Directly modifying the groups assigned to a single user

While on the account management screen for a user, find where the user's groups are listed, near the top of the screen. Click the pencil **Edit** icon to view a list of all GC application groups (minus the Everyone group). Check the box for the groups you want the user to belong to, and uncheck the box for groups you want to remove the user from. Click **OK** to commit your changes. The user immediately inherits application permissions from the groups they belong to.

### Users in multiple groups

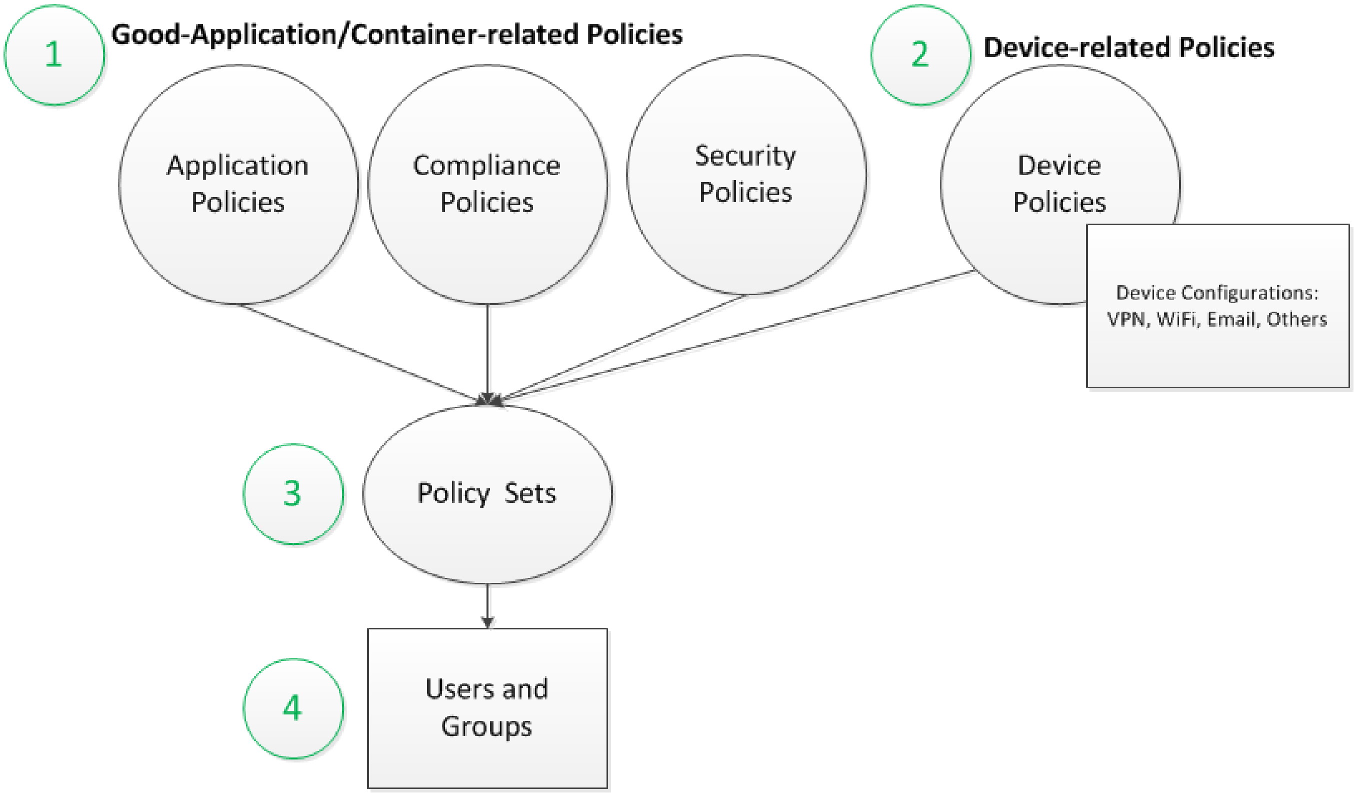
Because users can belong to multiple groups, a user might inherit conflicting permissions. When this happens, the most restrictive permission applies. For example, if a user belongs to three groups, and a certain entitlement version is: a) denied by one of the groups, b) allowed by the second group, and c) has no permission explicitly set for the third group, then the application is denied for the user at the group level.

Explicitly set user level permissions always override group level permissions.

# Policy Sets

# Policies

The diagram below shows the relationships of the various types of policies in BlackBerry Dynamics and the general sequence of working with them. At the highest level (the circled green numbers), there are *BlackBerry-application-related container policies*, *device policies*, *policy sets*, *users and application groups*.



The application/container policies control the behavior of the containers on the device, while device policies control the features of the device itself. Thus, you have layers of control. For instance, with the container security policies, you might require that an application password be six characters long, while with device policies, you might require that the device password be seven characters long.

Device configurations give you even finer-grained capabilities with device policies. For example, you might want one policy for users who access your systems with VPN and another for users who access your systems with WiFi.

### Security Policies

These policies govern the security of GD application passwords and access keys and define security related application behavior.

- With Password Policies, you control the required format of GD application passwords and how often users must change their passwords.

- You cannot deselect the **Require at least X characters** option or set a minimum length of zero characters, because passwords are required for GD applications.
- The setting **Do not allow more than one password change per day** affects the behavior of the GD SDK APIs by which users can be allowed to change application passwords. The specific APIs involved are showPreferenceUI on iOS and openChangePasswordUI on Android.

- With Lock Screen Policies, you control when the GD applications on users' devices ask for a password. You can also configure whether to lock or to wipe applications after a number of authentication failures.
- You can choose to prevent data from being copied from GD applications to other applications on the device.
- You can configure the GD application, if any, that serve as the authentication delegate on devices for all users assigned this policy set.
- With Provisioning Policies, you configure the text for provision emails. These emails contain the access keys your users use to activate GD applications on their devices. You can also configure how long the access keys are valid.

### Compliance Policies

Compliance policies include rules that are specific to mobile device platforms. You can set how often the compliance rules are enforced.

For each platform, you can set compliance rules for device connectivity, jailbroken/rooted devices, and allowed device OS versions, hardware models, and GD Library versions. If a user's device is out of compliance with one or more of the rules, the specified failure action is triggered for GD applications on the device. For example, if you have specified the Wipe Container failure action for devices that have not connected in the last 7 days, and a user's device is out of compliance with that rule, GC sends the command to the user's device to wipe data for any installed GD applications the next time it connects.

### Application Policies

You can configure policy rules specific to GD applications configured for each policy set. Applications that have configurable policies are each displayed in a collapsible section under this tab.

### Device Policies and Device Configurations

Device policies represent accessible settings on the managed device. These include but are not limited to device passcode requirements, device restrictions, and mandatory support, such as device encryption.

You can associate device policies with device configurations, which you can think of conceptually as representing groups of users who access your network in common ways.

### Policy Sets and Policy Reconciliation

A policy set defines a common set of rules that are applied to a collection of users. These policies affect every GD application installed by all of the users in the collection, across all of their devices that have been enrolled in mobile device management.

You can also assign a policy set to a GD application. If you do this, the application's policy rules override the rules in all users' policy sets only for the given application.

Periodically, the GC server retrieves policies from the NOC to ensure that the latest are being enforced. This is called *policy reconciliation*.

## Creating a New Policy Set

Because policy sets are extensive and take time to set up, copy an existing policy set and modify the copy however you wish. This way, you can choose the policy set you want to start with for your new policy set, without having to start from scratch.

When you copy a policy set, only the policies are copied to the new policy set; the list of users assigned to the original policy set is not copied or reassigned to the newly created policy set.

To copy a policy set, click **Policy Sets** in the main navigation to view a list of all current policy sets.

Find the policy set you wish to copy, and click the **Copy** icon. GC then creates a duplicate of that policy set and displays the edit screen for your new policy set.

## Modifying the Rules of a Policy Set

To view or modify a policy set, click **Policy Sets** in the main navigation, then click the **Edit** for the target policy set. GC then displays the policy management screen.

Policy rules are divided into three major sections, each with its own tab: **Security Policies**, **Compliance Policies**, and **Application Policies**. Provisioning Policies are a subset of the Security Policies. Descriptions of the rules in each of these sections are included in corresponding topics of this guide. For more information, see Configuring security policy rules, Configuring provisioning policy rules, Configuring compliance policy rules, and Configuring application specific policy rules.

Some policies are not just simple toggles, while others can have additional settings you can configure. Look for an underlined value in the policy definition, and position your cursor over it to view a dropdown list of possible settings for the rule. If you select a new value, the value is highlighted with a blue background. This is a visual reminder that the value has been changed since the last time the policy set was saved.

Changes you make to policy rules are not automatically saved. When you are finished making changes to the policies on a tabbed section, you must click **Update** on that tab to commit your changes.

## Assigning the Default Policy Set

When you import a single user from Active Directory into GC, the account is by default assigned that policy set that has been designated as the default policy set. The default policy set is always displayed at the top of the list on the **Policy Sets** screen, shown , and indicated by a small **Star**.

You cannot delete the default policy set, but you can designate any existing policy set as the default at any time. Click the ⭐ **Default** icon for any policy set to designate that one as the default. The previous default policy moves down the list and can now be deleted, if necessary.

## Adding Device Policies to Policy Sets

Device policies are created with the **Device Policies** screen (see Working with Device Policies ) see and added to policy sets with the **Policy Sets** > **Device Management** tab.

**To add a device policy to a policy set:**

1. Go to **Policy Sets** > **Device Management**.
2. Next to the label **DEVICE POLICIES**, click the triangle to reveal the existing policy sets.
3. If you want two different policy sets, one for "admin-enrolled" (or "Corporate-owned") devices and another for "employee-enrolled" (or "BYOD") devices, checkmark the checkbox.
4. To create a new policy set, click the large plus (⊕) sign.
5. Otherwise, scroll in the list to find the policy set you want and:
   - Click the pencil icon to edit it, or
   - Click the trashcan icon to remove the policy set and confirm the deletion.
6. You can use the up and down arrows on the far right of the policies to change their priority.
7. For **OS**, from the menu, select All (default), Android, or iOS.
8. For **Form Factor** (the general class of the device), from the menu, select **ALL**, **Phone** or **Tablet**.
9. Under **Device Policy**, from the menu select the name of the desired device policy.
10. The **Devices** fields shows you how many devices this policy set has affected.
11. When finished, in the middle right click **Update**.
12. Click **OK** to confirm or **Cancel** to discard your changes.

## Changing the Policy Set Assigned to Users

Users can only have one policy set at a time.

When you import users from Active Directory into GC, you can specify the policy set to apply to the new users. You can modify the policy set for one or multiple users at any time from the **Users and Groups** screen. For more information, see Modifying user accounts.

Keep in mind that when you change a user's policy set, all of the user's GD applications are checked for compliance with the new policy rules. If a GD application is out of compliance with the new rules, the rule's failure action is performed; depending on your policy configuration, the application might be locked or wiped.

# Deleting a Policy Set

**Note:** A policy set cannot be deleted when it is assigned to users. Move users with this policy set to another policy set before deleting this policy set.

In the GC console, you can delete a policy set: in two ways:

1. Click the **Policy Sets** navigation link. Click the ⊗ **Delete** icon for a policy set to remove it from the system.
2. While on the edit screen for a policy set, click **Delete** at the top of the screen.

**Note:** the default policy set cannot be deleted.

# Applying a Policy Set to an Application

Policy sets contain rules that govern the security of GD applications and rules that are specific to mobile device platforms, such as the devices and OS versions that GD applications are allowed to run on or whether GD applications can run on jailbroken or rooted devices.

Each user is assigned a policy set that enforces security and compliance policy rules universally across all applications the user activates. However, you can define finer-grained control over policy rules for a specific application. For example, you can apply a policy set that enforces strict password rules but might want access to a specific application to not require a password.

With the GC console you can apply an override for individual applications. If you do this, the application's policy rules override the rules in all users' policy sets only for the given application; users' policy set rules still apply for all other GD applications. A complete discussion with many examples is in More about Policy Overrides.

To apply a policy set override for a GD application:

1. Create a policy set on the **Policy Sets** screen that defines the rules you want to apply to the application.
2. Go to the application management screen by navigating to **Manage Applications** and selecting the application from the list.
3. From the pulldown menu next to the Policy Set Override label, select the policy set you want to apply to the application.

**Note:** It can take up to 24 hours for the new policy to propagate to GC servers.

# Configuring Security Policy Rules

Security policies govern the strength of GD application passwords and define security related application behavior.

To modify security rules for a policy set, click **Policy Sets** in the main navigation, then click **Edit** for the policy set you want to update. GC then displays the policy management screen. Make sure the **Security Policies** tab is active.

## Summary of Good Control Security Policies

Last updated: 9/25/2017

| Policy | Description | Notes, Examples, Caveats |
|---|---|---|
| Do not require user password for Android<br>Do not require user password for iOS | Default: off.<br>If you enable one of these policies, the following message is displayed in the Good Control console:<br>Warning Disabling the BlackBerry application password significantly reduces security of BlackBerry containers and Enterprise Network. Use of this mode is strongly discouraged. | With a BlackBerry Dynamics application that is protected by security policy to require a password, if the IT administrator changes the security policy to "No Password":<br>• The user is shown an informational screen stating that a password is no longer required for the application.<br>• The user is then in "No Password mode" and is never prompted for password again.<br>Conversely, if the user is in "No Password mode" but the IT administrator changes the security policy to require a password:<br>• The user is prompted to set a password.<br>• The user is shown an informational screen stating that a password is now required for the application. |
| Expire Password after *X* days | Default: not enabled | |
| Disallow *X* previously used passwords | Default: not enabled | Ranges from 1 to 12 |
| Require at least *X* characters | Set the password length from 1 to 14 characters.<br>Default: 4 characters | You cannot deselect this policy because passwords are required for GD applications, nor can you set the minimum length to 0 (zero) characters. |
| Allow at most *X* occurrences of any given character | Allow from 1 to 5 occurrences<br>Default: 3 | |

| Policy | Description | Notes, Examples, Caveats |
|---|---|---|
| Do not allow more than one password change per day | Default: not enabled | |
| Allow Touch ID (iOS only) | Allow the user to authenticate and Fingerprint Authentication<br>Default: not enabled | See also *BlackBerry Dynamics and Fingerprint Authentication*. |
| Enable Touch ID from Cold Start | Policy appears if Touch ID is allowed.<br>Default: not enabled | See also *BlackBerry Dynamics and Fingerprint Authentication*. |
| Require Password not Fingerprint After *X* Period since Password last used | Policy appears if Touch ID is allowed.<br>Default: 1 day | See also *BlackBerry Dynamics and Fingerprint Authentication*. |
| Android Fingerprint Authentication | Allow the user to authenticate with Android Fingerprint<br>Default: not enabled | See also *BlackBerry Dynamics and Fingerprint Authentication*. |
| Enable Android Fingerprint from Cold Start | Policy appears if Android Fingerprint is allowed.<br>Default: not enabled | See also *BlackBerry Dynamics and Fingerprint Authentication*. |
| Require Password not Fingerprint After *X* Period since Password last used | Policy appears if Android Fingerprint is allowed.<br>Default: 1 day | See also *BlackBerry Dynamics and Fingerprint Authentication*. |
| Do not allow personal information | Imposes constraints on the use in a password of the following personal information:<br>• The user's first and last (or personal) names (excluding initials) as recorded in Active directory.<br>• The part of an email address to | For example, for a user named "Abraham Q. Lincoln-Jones" with an email address "apljones@whitehouse.gov", the following are disallowed in the password: aljones, Abraham, Lincoln, Jones, and Lincoln-Jones. The example passwords are *not* valid for Mr. Lincoln:<br>• Invalid: **`Abraham77##`**: Invalid because it contains "Abraham".<br>• Invalid: **`11jones`**: Invalid because it contains "jones". |

| Policy | Description | Notes, Examples, Caveats |
|---|---|---|
| | the left of the @ sign (the "username" part of an email address).<br>Default: enabled | |
| Require both letters and numbers<br>Require both upper and lower case<br>Require at least one special character<br>Do not allow more than 2 numbers in sequence | Defaults: not enabled | |
| **Lock Screen Policies** | | |
| Always require password on application startup | Default: not enabled | This option is mutually exclusive with Authentication Delegation (see below). |
| Require password when idle for more than *X* | Default: enabled, 1 hour | Timeout range is from 3 minutes to 1 day |
| After *X* invalid password attempts *action* | Default attempts: 10<br>Default *action*: Lock Out User | Range of attempts: 1 to 12<br>Possible actions:<br>• Lock Out User<br>• Wipe Data |
| **Wearables Policies** | | |
| Allow wearables | Default: not enabled | |
| **Authentication Delegation** | | |
| See discussion in Assigning Authentication Delegates . | Default: not enabled | This option is mutually exclusive with "always require password on application startup (see above). |
| **Data Leak Prevention** | | |
| Prevent copy from non-GD apps into GD apps | Default: not enabled | See also Enabling Secure Cut-Copy-Paste, or Data Leak Prevention . |
| Prevent copy from GD apps into non-GD apps | Default: enabled | |
| Prevent Android Dictation | Default: enabled | |
| Prevent Screen Capture (Android, Windows) | Default: enabled | • On Android, this setting also |

| Policy | Description | Notes, Examples, Caveats |
|---|---|---|
| | | blocks the application UI display in the task switcher (also known as **Recent**) .<br>- The behavior on Windows RT and UWP is the same as on Android.<br>- For iOS, a device policy governs this behavior. See Functionality . |
| Prevent iOS Dictation | Default: enabled | |
| Prevent Custom Keyboards (iOS only) | Default: enabled | See also Allow Third-Party Keyboards with Good Apps on iOS. |
| Enable FIPS | Enforce compliance with U.S. Federal Information Processing standard 140-2<br>Default: not enabled | See Enabling FIPS Compliance for a Security Policy . |
| **Certificate Management** | | |
| Trusted Certificates | Where should trusted certificates be stored?<br>Default: GD and Device Certificate Store | Allowable settings:<br>- GD and Certificate Store<br>- GD Certificate Store Only<br>- Device Certificate Store Only<br>See also Certificate Management Policies . To set the trusted authorities, see Trusted Authorities Tab . |
| Allow use of client certificates | Rely on PKCS 12 certificates for user authentication<br>Default: not enabled | See alsoAllowing Client Certificates and PKCS 12 Certificate Management . |
| **Provisioning Policies** | | |
| Access keys expire after *X* | Application activation keys (also called "access keys) cannot be used after X time period. | Time period ranges from 1 day to 90 days |

| Policy | Description | Notes, Examples, Caveats |
|---|---|---|
| | Default: enabled, 30 days | |
| Sender, Subject, Message | Default text sent to user with activation key | See also Configuring Provisioning Policy Rules . |
| **Agreement Message** | | |
| Enable agreement message | Enable a message to display in client applications, once or after every unlock<br>Default: not enabled | See also Configurable Agreement Message . |

## New: Prevent end-user from enabling detailed logging

To increase supportability of the system, the security policy **Prevent users from turning on detailed logging** controls whether or not the end user of a GD based application can enable detailed logging on the client.

Default: Enabled. Detailed logging is not allowed.

Usage notes:

- When this policy is set to prevent the end user, the end user is not shown any control in a BlackBerry Dynamics application to turn on detailed logging.
- If the policy **Enable detailed logging for GD apps** is enabled, the policy **Prevent users from turning on detailed logging** is grayed out, not settable.

## New: Enable detailed logging for BlackBerry Dynamics apps by policy set/by user group

Detailed logging is controlled by the security policy **Enable detailed logging for GD apps**.

Default: Not enabled.

You can set this policy in those policy sets that you apply to specific groups of end users you want to allow detailed logging.

Usage notes:

- Disabled: When this policy is disabled, the related setting for **Detailed Logging** for a particular user under **Manage Users** can still be used to set the policy for individual users.
- Enabled: When this policy is enabled in a policy set applied to a specific group of users, the related function on the **Manage User** page for a particular user is grayed out, and not settable.

## Setting "No password" policy

The security policies below allow your end users to avoid having to set an application password when a BlackBerry Dynamics-based application is activated:

- Do not require user password for Android
- Do not require user password for iOS

By default, these policies are not enabled.

If you enable one of these policies, the following message is displayed in the Good Control console:

> **Warning**
>
> Disabling the BlackBerry application password significantly reduces security of BlackBerry containers and Enterprise Network. Use of this mode is strongly discouraged.

### Effects of enabling "no password"

With a BlackBerry Dynamics application that is protected by security policy to require a password, if the IT administrator changes the security policy to "No Password":

- The user is shown an informational screen stating that a password is no longer required for the application.
- The user is then in "No Password mode" and is never prompted for password again.

Conversely, if the user is in "No Password mode" but the IT administrator changes the security policy to require a password:

- The user is prompted to set a password.
- The user is shown an informational screen stating that a password is now required for the application.

## Optional: Allowing Android Fingerprint and interval to require password

Use of Android Fingerprint for user authentication in BlackBerry Dynamics-based applications is governed by Good Control policy setting.

By default, Android Fingerprint is not allowed.

To set Android Fingerprint policy, in Good Control:

1. Navigate to **Policy Sets** > **edit a policy** > **Security Policies** tab.
2. Scroll to find the heading Fingerprint Policies under Password Policies.
3. Check the policy **Allow Android Fingerprint for Idle Unlock** to enable it. Default is "not allowed".
4. If desired, check the policy **Enable Android Fingerprint from Cold Start**.
5. Set the interval after which end users are required to enter the application password: **Require Password not Fingerprint after N period since Password last used**. Interval can range from 1 hour to 7 days. Default is minimum of every *1* days.

   This setting honors authentication delegation, so that only the password for the delegate application is required.

> **Note:** For client applications built with GD SDK v2.3.xxxx for use with versions of GC *before v2.3.xx.yy* , the interval is 48 hours. For client applications built with GD SDK v2.3.xxxx for use with GC v2.3.xx.yy release, if this policy is not explicitly set, no interval is enforced.

6. Click **Update** to save your changes or navigate away from the page to discard them.

## Optional: Allowing Apple Touch ID and Interval to Require Password

Apple Touch ID is a fingerprint recognition system for some iOS devices.

Touch ID can be allowed for user authentication in BlackBerry Dynamics-based applications, in addition to standard password authentication. One effect of allowing Touch ID is that, if the end user disables then re-enables the device's password, the user is required to first re-authenticate via password, not Touch ID; after password re-authentication, Touch ID is allowed again. Other behaviors of Touch ID and complete details about BlackBerry's support for it are in the white paper BlackBerry Dynamics with Apple Touch ID.

**In Good Control, to allow/disallow Apple Touch ID:**

1. Navigate to the **Policy Sets** > **Security Policies** tab.
2. Scroll to find the heading **Fingerprint Policies** under **Password Policies**.
3. Scroll to find the checkbox for **Allow Touch ID**, and check the checkbox to allow Touch ID.
4. If you want to enable Touch ID when an application starts, click the checkbox next to **Enable Touch ID from Cold Start**.
5. Set the interval after which end users are required to enter the application password: **Require Password not Fingerprint after N period since Password last used**. Interval can range from 1 hour to 7 days. Default is minimum of every day.

   This setting honors authentication delegation, so that only the password for the delegate application is required.

   > **Note:** For client applications built with GD SDK v2.3.xxxx for use with versions of GC *before v2.3.xx.yy* , the interval is 48 hours. For client applications built with GD SDK v2.3.xxxx for use with GC v2.3.xx.yy release, if this policy is not explicitly set, no interval is enforced.

6. In the upper right, click **Submit**.
7. Click **Cancel** to uncheck **Enable Touch ID from Cold Start**, or click **OK** to enable Touch ID when an applications starts and update the policy.

## Allowing Wearable Devices

The terms *wearables* or *wearable devices* refer to small computers intended to be worn on the human body, in distinction from *handhelds* or *handheld devices* like smartphones or tablets. BlackBerry's support for wearable devices includes a Good Control policy to allow or disallow them with Good-enabled applications and the BlackBerry Dynamics Wearable Framework.

- The Good Control administrator can specify via the console's **Security Policies** > **Wearable Policies** if wearable devices are allowed or disallowed. The default is "not allowed". If allowed, some other settings define behavior:

**Timeout after disconnect** in minutes and **Enable Auto-reconnect**.

- The GD SDK developer can work with the GD SDK for Android and the BlackBerry Dynamics Wearable Framework, which is packaged with the GD SDK for Android. Currently supported devices include those that strictly adhere to Android Wear guidelines from Google.

For the end-user of a Good-based application, after the standard provisioning process and after setting a password for an application, if allowed, wearable devices manifest themselves in several ways:

- Depending on the setting of **Timeout after disconnect** by the GC administrator, when a wearable device has been disconnected from the handheld device, the wearable application is locked after a specific period of time, either immediately or up to an hour.
- Depending on the **Enable Auto-reconnect** setting by the GC administrator, a wearable application can be allowed to auto-authenticate with its paired handheld application, after the wearable device is reconnected to the handheld device. However, if the handheld application is locked, the user must enter the application password.

**In Good Control, to allow or disallow wearable devices:**

1. Go to **Policy Sets** > **edit a policy** > **Security Policies**.
2. Scroll to find **Wearable Policies**.
3. Find **Allow Wearables**.
4. Check the checkbox to allow wearable devices, or uncheck it to disallow them.
5. If allowed, configure other desired settings:
   - **Timeout after disconnect** from 0 minutes up to one hour. After an Android Wearable has been disconnected, the amount time in minutes before any associated Good-enabled application is locked and requires authentication. Default value is 0; the application is immediately locked.
   - **Auto-reconnect**: Automatically reconnects to a previously disconnected Android Wearable when that Wearable comes again into close proximity of the device.

## Enabling Secure Cut-Copy-Paste, or Data Leak Prevention

You can prevent users of secure GD applications from copying data to other, insecure applications on the device, prevent the user from taking screenshots, and other constraints.

**To set enhanced data leak prevention policies, in Good Control:**

1. Navigate to **Policy Sets** > **edit a policy** > **Security Policies** tab.
2. Scroll to find the heading **Data Leak Prevention**.
3. Check or uncheck the desired policies.

   A. - **Prevent copy from GD apps into non-GD apps** is the primary policy. If it is enabled, then:
        - **Prevent copy from non-GD into GD apps** is a secondary policy that becomes visible.

   B. **Prevent Android Dictation**

   C. **Prevent Screen Capture (Android, Windows)**

   D. **Prevent iOS Dictation**

**About older client applications.** Formerly, the only policy available for data leak prevention was a single policy that governed all the behaviors for which separate, finer-grained policies are now available. Applications built with earlier versions of the GD SDK support the finer-grained controls by mapping the old, single policy to the newer policies in the list above: B. **Prevent copy from GD apps into non-GD apps** and D. **Prevent Android Screen Capture**.

## Certificate Management Policies

This policy sets the trusted authorities to secure communications to the application server.

**To set security policy for client certificate storage:**

1. Navigate to **Policy Sets**.
2. Edit the desired policy set.
3. Click the **Security Policies** tab.
4. Scroll down to find the heading **Certificate Management**.
5. Choose from the following selections:
    - GD and Device Certificate Store
    - GD Certificate Store Only
    - Device Certificate Store Only
6. In the upper right, click **Update** to save your changes.

In addition to setting this policy, you might need to set trusted authorities. See the Trusted Authorities Tab .

## Allowing Client Certificates

This policy enables client certificates, for uses such as S/MIME or user authentication. It allows:

- Uploading of client certificates to Good Control
- Retrieval of user certificates by Good Control when necessary

By default, the security policy that allows the use of certificates is disabled (false).

If this policy is disabled, then the **Certificates** tab is hidden from the end-user's view of the User Self Service portal but not from the GC administrator's view, who can still add, update, and delete certificates even if the security policy is disabled for a particular user.

**To allow client certificates:**

1. Navigate to **Policy Sets**.
2. Edit the desired policy set.
3. Click the **Security Policies** tab.
4. Scroll down to find the heading **Certificate Management**.
5. Check **Allow use of client certificates**.
6. In the upper right, click **Update** to save your changes.

In addition to setting this policy, you might need to create certificate definitions on the Certificate Definitions Tab and set applications on the App Usage Tab .

## Enabling FIPS Compliance for a Security Policy

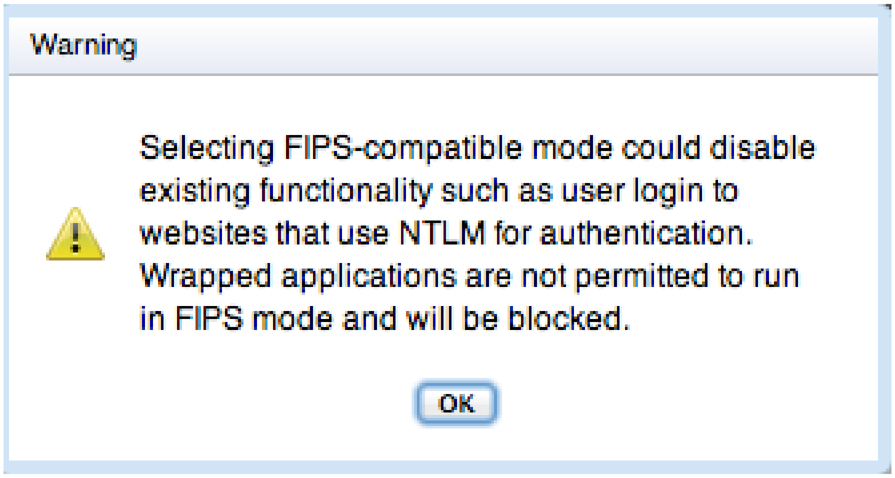
You can enable FIPS compliance for any security policy. Federal Information Processing Standards (FIPS) are U.S. government regulations regarding computing and computing security. When you enable FIPS compliance in a policy, the major effect is on associated applications. Enabling FIPS compliance enforces the following constraints in conformance with FIPS:

- MD4 and MD5 are prohibited by FIPS, which means that access to NTLM- or NTLM2-protected web pages and files is blocked.
- Wrapped applications are blocked.
- In secure socket key exchanges with ephemeral keys, with servers that are not configured to use Diffie-Hellman keys of sufficient length, GD retries with static RSA cipher suites.

### Steps for Enabling FIPS

1. On the **Security Policies** tab for the policy management screen shown above, scroll to the **Authentication Delegation** portion of the screen, and click the **Enable FIPS** checkbox:

   The system displays a warning:



Warning

Selecting FIPS-compatible mode could disable existing functionality such as user login to websites that use NTLM for authentication. Wrapped applications are not permitted to run in FIPS mode and will be blocked.

2. Click **OK** to acknowledge the warning.

### Effect on Applications: Block

Applications that do not conform to the security policy are blocked on user devices. Users must contact an administrator to be unblocked.

You can unblock the user by disabling FIPS compliance in the policy at either the user level or the application level.

### Developing FIPS-compliant Applications on iOS or Android

FIPS compliance is supported on iOS and Android.

For information about how to develop FIPS-compliant applications, see the pertinent GD SDK guide for iOS or Android available from the BlackBerry Dynamics Library.

## Allow Third-Party Keyboards with BlackBerry Apps on iOS

"Third-party keyboards" replace the standard keyboard on iOS devices. BlackBerry's support of third-party keyboards extends to allowing them or disallowing them. The default is "Not Allowed".

- With a policy setting, the administrator of Good Control can allow or not allow the use of third-party keyboards.
- For the BlackBerry Dynamics SDK developer, no extra programming is required to integrate control over third-party keyboards.
- For the end-user:
  - When third-party keyboards are allowed, the custom keyboard option is displayed in the application UI.
  - When third-party keyboards are not allowed, the custom keyboard option is not displayed.

**In Good Control, to allow or disallow third-party keyboards:**

1. Go to the **Policy Sets** > **Security Policies** tab.
2. Scroll to find (iOS only)**Prevent custom keyboards**.
3. Check the checkbox to allow custom keyboards or uncheck the checkbox to disallow them.

## Configurable Agreement Message

You can create a message in Good Control that is displayed in GD-based applications:

- The message is displayed in GD-based applications before the password or authentication prompt.
- If authentication delegation is enabled for your end users' application, the message is displayed in the authenticator application, not in the individual applications.
- The message can be displayed according to a frequency you specify: every time an application unlocked or when you have changed your message.
- You yourself are responsible for the contents of the message, including localizing it for the desired languages:
  - Unicode text is supported.
  - HTML formatted text is not supported.

**Note:** When an application receives this policy from the GC, an event is recorded in the GC's container event history as a "Security Policy ACK". This means only that the policy was received by the application. It does not mean that the end-user agreed to the message, although in order to proceed, the end-user must tap in agreement.

**To set an agreement message, in Good Control:**

1. Navigate to **Policy Sets** > *edit a policy* > **Security Policies**.
2. Find the **Agreement Message** heading.
3. Checkmark **Enable agreement message**.

4. If you like, checkmark **Display message every time the app is unlocked**.
5. In the **Message** box, enter your text. Limit: 1MB.
6. Click **Cancel** to discard or **Save** to save your changes.

## Configuring Provisioning Policy Rules

Provisioning policy rules allow you to set a validity period for access keys and determine how provision emails are formatted.

To modify provisioning rules for a policy set, click **Policy Sets** in the main navigation, then click the ✎ **Edit** for the policy set you want to update. GC then displays the policy management screen. Make sure the **Security Policies** tab is active, then scroll down the list of rules until you see the Provisioning Policies section.

First, you can specify a number of days that access keys remain valid. If a user is provisioned an access key but does not use it to activate an application during the specified period, the key expires and is no longer usable. A new access key must then be provisioned for the user before they can activate a GD application.

To set a validity period for access keys, check the **Access Keys expire after** checkbox and select a number of days from the pulldown menu. To create access keys that do not expire, uncheck this checkbox. If you make changes to these settings, your changes affect only access keys generated from that point onward; all access keys generated prior to your modifications are not affected by the new settings.

When an access key is generated for a user, GC sends the user a provision email that contains the access key. You can configure the text for the sender name, subject line, and body of these emails.

The following tokens are used in the provision email message.

- <**%APPLICATION_LOCATIONS%**> - The names (and download URLs, if specified) of all the products and entitlement versions that the user has permission to activate. This list is current as of the moment the provision email is generated.
- <**%HELPDESK_REF%**> - The user’s display name, retrieved from Active Directory when the user's GC account was created.
- <**%EMAIL_ADDRESS%**> - The user’s email address, also retrieved from Active Directory when the user's GC account was created.
- <**%PIN_FULL%**> - The access key that was generated.
- <**%EXPIRY%**> - The access key's expiration date. This date is determined by taking the current date and advancing it by the number of days that access keys are configured to remain valid.

By default, the email templates are formatted as plain text.

You can also format your email templates as HTML. Any valid HTML 4 or HTML 5 can be used.

Keep the following in mind:

- The size of the template is limited to 4,000 characters, including both tags and text.
- To specify HTML formatting, add this as the first line: **`<!DOCTYPE html>`**. Otherwise the system treats the template as plain text.

- Follow the HTML document type declaration, with **`<head>`**, **`<body>`** and any other desired HTML tags. Be sure to use closing tags (like **`</body>`**) for normalized HTML.
- Before you enter the HTML into the template form in Good Control, be sure to make sure it is valid HTML. Good Control does not validate the HTML. If GC encounters invalid HTML in the template, the message is sent as plain text.

  - Be careful to keep the embedded variable names that GC requires in the text, but you can format them however you like. Formatting example: a single paragraph with bold email address: **`<p>Your email address is <b><%EMAIL_ADDRESS%></b></p>.`**
- All links to CSS, images, or other resources on the internet must be absolute and must be reachable by your end-users' browsers or email clients. That is, the HTML in the template is not relative to a document root, as it would be on a standard web server:

  Images can be base64-encoded and included in the template's **`<img>`** tags, as in the following example snippet: **`<img src="data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAADIA...">`**

  - Any CSS must be defined in the **`<head>`** or inline in the template.

Changes you make to policy rules are not automatically saved. When you are finished making changes to the policy rules, you must click **Update** for the tab to commit your changes.

## Assigning Authentication Delegates

In Good Control, a fundamental design decision for application user authentication is whether to rely on users to enter a password for each individual application or to configure authentication delegation. These options are mutually exclusive. You can configure either one of these options, but not both, on the **Policy Sets** > **Security Policy** tab for an application.

**To assign authentication delegates for a policy set:**

1. Click **Policy Sets** in the main navigation, then click the ✎ **Edit** icon for that policy set to view the policy management screen.
2. On the **Security Policies** tab, go to the **Authentication Delegation** heading.
3. Click **Add Applications** to display a list of registered applications.

   You can filter the list of applications by the **Type** field. From the pulldown menu, select the desired type.
4. From the list, click the plus sign associated with each application you want to act as an authenticator on the devices of all users assigned the policy set.
5. Use the up and down arrows to change the priority of the delegates, or use the circled, red **X** to remove an application from the list. In addition, if desired, click the checkbox **Allow self-authentication when no authentication delegate application is detected**; this is the fallback authentication mechanism described below.
6. When finished, in the upper right, click **Update** to save your changes or **Cancel** to discard them.

A BlackBerry Dynamics application can delegate its user authentication to other BlackBerry Dynamics applications. When the user launches a BlackBerry Dynamics application, the device displays the password screen for the authentication delegate, not the password screen of the application initially launched. After the user enters the password for the authenticator application, the user is then returned to the originally launched application.

**Terminology**

- An application that delegates the authentication task to another application is called a *delegating application*.
- An application that handles the authentication task for other BlackBerry Dynamics applications on a device is called the *authentication delegate* (informally, the "auth delegate") or *authenticator*.
- The Good Control administrator can define up to three applications that are allowable authentication delegates. This is called *multi-authentication delegation*.
- Authentication delegation is allowed among applications only for a single user. That is, an application associated with a user cannot delegate to another application associated with a different user.
- Any BlackBerry Dynamics application can be designated as an authentication delegate. Applications that serve as authentication delegates must have a native bundle identifier defined in Good Control; for more information see About BlackBerry Dynamics entitlement ID and version
- In addition, the Good Control Administrator can enable a *fallback delegate* when designated delegates have been tried without successful authentication for some reason. The fallback delegate is the application itself.
- If no authentication delegates have been set, the system default is that the application itself is its own delegate.

### Purpose and Recommended Use of Multi-authentication Delegation

The purpose of multi-authentication delegation is to allow the Good Control administrator to designate authentication delegate applications across platforms (operating systems) in a BlackBerry Dynamics deployment in which all users do not have the software necessary to use only a single delegate.

BlackBerry recommends the following:

1. *Assign only one auth delegate per policy set, and consider defining only a single auth delegate per platform.* This prevents unnecessarily complex and undesirable auth delegate switching by the end user and simplifies your own administrative work. If a user accidentally deletes an auth delegate, they are guided to reinstall it.
2. If a user deletes an auth delegate, *the first recourse should be to re-install the deleted application*, not to switch to the secondary delegate.
3. If a user already has a secondary auth delegate installed and in use and then later installs the primary auth delegate (perhaps when it becomes available for the platform or if a new primary is configured by the administrator), then the end user needs to be carefully guided through the process. *End users must not delete the currently installed auth delegate.* Instead, each delegating app will automatically switch to the new auth delegate when the delegating app is next launched in online mode.
4. In the rare case that the selected primary auth delegate does not exist for a given platform (for instance, a trusted authenticator that is only available for iOS), either “None” or an alternate auth delegate should be selected as a secondary delegate in Good Control.

### Enable Auto-Push for Auth Delegates

Be sure to enable automatic application push (auto-push) to user devices of your designated delegate applications. This prevents the user from having to download the delegate apps and allows you to manage these applications like any other managed app.

For details on auto-push see the Good Control online help topic "Enabling Auto-Push, Exempting Policy Sets".

### Some Effects of Changes in Authentication Delegates

There are several conditions that can affect how multi-authentication delegates applications function on a device. The key point is that all applications that rely on the authenticators must be in the unlocked state to set up a new authenticator.

| Condition | State or Effect |
|---|---|
| Initial setup of application | The provisioning of an application. By definition, the app is in unlocked state. |
| When the policy changes (that is, the formerly defined authenticators or their sequence are changed). | The previously defined authenticator is still present on the device, so applications that might be locked can be unlocked to apply the new policy. |
| A higher priority authenticator application is installed. | The existing authenticator is still present on the device, so applications that might be locked can be unlocked. |
| The current authenticator application is deleted from the device. | If the current application or the other authenticators in the defined sequence are locked, the end-user is now blocked.<br>**Important:** To remedy this, the end-user must reinstall the original authenticator application on the device.<br>In the other cases, as long as the applications are in the unlocked state, authentication delegation can be set up with the new delegates or with a password. |

## Configuring Compliance Policy Rules

Compliance policies include rules that are specific to mobile device platforms. For each platform, you can set compliance rules pertaining to device connectivity, jailbroken/rooted devices, and allowed device OS versions, hardware models, and GD Library versions. The device platforms are as follows, which correspond to the groupings on the compliance policies in the user interface:

- Android
- iOS
- MacOS, also known as OS X
- Microsoft Windows:

  The version number "6.3" shown in the GC console, which is actually the version of the underlying NT kernel, corresponds to Windows OS version 8.1.

**To modify compliance rules for a policy set:**

1. Click **Policy Sets** in the main navigation.
2. Click the **Edit** for the policy set you want to update. Good Control then displays the policy management screen
3. Click the **Compliance Policies** tab to view compliance rules.
4. Click the desired setting to enable the compliance check.

You can configure rules for all supported mobile platforms according to the requirements of your organization. For each mobile platform, policy rules are grouped into subcategories.

## Android Hardware Manufacturers or Models

See the discussion in Compliance Policy: Android Hardware Manufacturers or Models .

## Failure Actions

Each category of rules has an associated failure action that is triggered if a user's device is out of compliance with the ruleset. You must select one of the following failure actions for each of the rule categories:

- **Application not allowed to run** - This action blocks the user from accessing the GD application, but does not delete or modify application data. This action is reversible; after the user's device is back in compliance with the rules, the GD application is unblocked and can be used normally.
- **Wipe Data** - This is the stricter of the two actions. If this option is selected and a GD application on a user's device is out of compliance with the associated rules, the container and its associated data are wiped from the user's device. If the user wishes to use the GD application again, they must ensure their device is in compliance with all policy rules and then reprovision and reactivate the application; however, the wiped application data is unrecoverable.

You can configure how often the compliance rules are enforced. To do this, find the **Enforce every** setting at the top of the list of compliance policies, then select a time period from the pulldown menu of options. Rules for all platforms are enforced on a schedule determined by this setting. If a GD application on a user's device is out of compliance with a category of rules, the category's specified failure action is performed for the GD application. For example, if you have specified the Wipe Data failure action for devices that have not connected in the last 7 days, and a user's device is out of compliance with that rule, GC sends the command to wipe the GD application from the device the next time it tries to connect.

Compliance policy rules by category

| Category | Description |
|---|---|
| **OS Version Verification** | Located just under each platform heading, the category of rules contains settings for the different OS versions your users' devices are running.<br>**Allow all OS Versions**: Default: Yes, allow.<br>• Set the option to Yes if you want to allow all OS versions to run your GD applications, including future OS versions not yet be released.<br>• Set it to No if you want to disallow specific OS versions from running your GD applications.<br>**Allow previously unknown versions**: Default: checked, allow. |

| Category | Description |
|---|---|
| | - Check this option (default) if you want to allow all previously unknown versions of the allowed versions OS<br>- Uncheck this option if you do not want to allow previously unknown versions of the allowed OS versions.<br>**Failure Action**: Specify a failure action to take if a user's device is out of compliance with the rules in this section:<br>- Default: Application not allowed to run<br>- Wipe data |
| **Hardware Model Verification** | A list of devices that run the platform OS.  For Android hardware, see the discussion in Compliance Policy: Android Hardware Manufacturers or Models . Otherwise, specify the models you want to allow. Specify a failure action if a user's device is out of compliance with the rules in this section. |
| **BlackBerry Dynamics Library Version Verification** | Select which versions of the GD library are permitted. This has a large impact on which GD applications are allowed to run on devices. For example, if you deselect the 1.0 option, no applications that were compiled with version 1.0 of the GD SDK are allowed by the policy set. Specify a failure action if a user's device is out of compliance with the rules in this section. |
| **Connectivity Verification** | Determines if a container has connected at least once within the specified time period. The default time period is 30 days but you can select a stricter or more relaxed value. Specify a failure action if a user's device is out of compliance with this rule.<br>If you want to base the interval on an application's GD-SDK-based authentication delegates, check **Base connectivity interval on auth delegate apps**. This option is displayed only if authentication delegation is enabled. For more information, see Assigning authentication delegates.<br>**Note:** Basing the interval on auth delegate apps applies only to GD-SDK-based apps, and does not include GFE, which is not based on the GD SDK.<br>When selecting a value for the **Connectivity Verification** rule, we recommend you consider the impact your changes might have for users in different scenarios while balancing the security needs of your organization. For example, if you set a value of 8 hours, and a user forgets to charge a device overnight, the device is probably out of compliance with the rule, and the failure action is triggered. |
| **Jailbreak/Rooted Detection** | A platform-specific check to see if devices are jailbroken or rooted. You can enable or disable this rule, according to the security requirements of your organization. If you choose to enable this rule, specify a failure action if a user's device is out of compliance with this rule. |

**Important:** Changes you make to policy rules are not automatically saved. When you are finished making changes to the policy rules, you must click the **Update**  for the tab to commit your changes.

## Compliance Policy: Android Hardware Manufacturers or Models

Good Control's Android hardware compliance policies allow you to enforce compliance either by Android hardware manufacturer or by specific hardware models.

Checkmarking specific models of Android hardware can be time-consuming. In addition, adding new models to Good Control takes a certain amount of time, which can cause delays in deployment of new models. Instead, with compliance by Android manufacturer, you have larger-grained control for quicker deployment of new hardware.

By default, compliance by Android hardware manufacturers is not enabled. You can set the policy to allow either all Android hardware manufacturers or only specific manufacturers.

If you set compliance by Android hardware manufacturers, this compliance is verified.

If compliance by manufacturer passes, hardware-model-specific compliance is skipped.

If compliance by hardware manufacturer fails, compliance by specific hardware models is verified.

If compliance fails in either case, the failure action you set under the **Hardware Model Verification**is taken.

**To set compliance policy for Android hardware manufacturer or model, in Good Control:**

1. Navigate to **Policy Sets** > *edit a policy* > **Compliance Policies** tab.
2. Scroll to find the heading **Android Platform Rules** and again to the heading **Android Hardware Manufacturers**.
3. If you want to permit hardware from all Android manufacturers, from the **Allow all hardware manufacturers** pulldown, select **Yes**.
4. If you want to permit hardware from only selected Android manufacturers:
    a. From the **Allow all hardware manufacturers** pulldown, select **No**.
    b. Check the displayed boxes for the names of those manufacturers you allow.
    c. Under the **Android Hardware Models** heading, click **Uncheck All**. This allows the system to verify against all known models for the checkmarked manufacturers, not just the models you indicate.
5. If you want to check compliance against hardware models:
    a. From the **Allow all hardware manufacturers** pulldown, select **No**.
    b. If you want to check against all hardware models, under the **Hardware Model Verification** heading, from the **Allow all hardware models** pulldown, select **Yes**.
    c. If you want to check only specific hardware models, under the **Hardware Model Verification** heading, from the **Allow all hardware models** pulldown, select **No** and checkmark those models you allow.
6. Whether you want to verify compliance by manufacturer or by model, under the **Hardware Model Verification** heading, set the failure action if compliance fails: wipe the application or block the application.
7. In the upper right, click **Update** to save your changes or **Cancel** to discard them.

## New: Compliance rule for Android OS versions allows alphanumeric characters

Good Control now allows an Android operating system version that includes both letters and numbers.

# Configuring Application Specific Policy Rules

You can configure policy rules for each policy set for GD applications.

To modify application specific rules for a policy set, click **Policy Sets** in the main navigation, then click the ✎ **Edit** for the policy set you want to update. GC then displays the policy management screen. Click the **Application Policies** tab to view application specific policies.

Applications that have configurable policies are each displayed in a collapsible section under this tab.

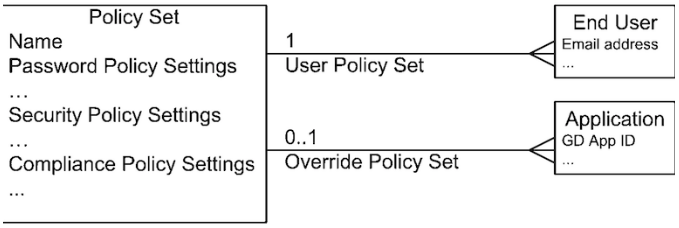
Changes you make to policy rules are not automatically saved. When you are finished making changes to the policy rules, you must click the **Update** for the tab to commit your changes.

## More about Application Policy Overrides, with Examples

The application policy override feature enables the BlackBerry Dynamics policies that apply to end users to be overridden for particular mobile applications. This feature is comprised of a number of configuration options for BlackBerry Dynamics (GD) administrators.

In relation to device policies, a device policy always comes from the user's policy set, not the application policy set override.

Each end user in a GD deployment is assigned exactly one policy set, which applies by default to all GD applications to which the user has access. This is referred to as the user policy set in the following description. Each GD application in a GD deployment can also be assigned a policy set. This is referred to as the override policy set in the following description. Assignment of an override policy set is optional, so a GD application either has one policy set or none. The policy sets that can be assigned as overrides are the same as the policy sets that can be assigned to end users. Both override policy sets and user policy sets are assigned from the same list of policy sets. These terms and relationships are also shown in the following diagram.



### Policy Override Rules

When an end user runs a GD application, a number of policy values apply to their use of the application. For example, the minimum length of the security password is a policy value, so is the true or false value of the data loss prevention flag.

The policy values that will apply are drawn from either the user policy set, or from the override policy set, or from both. The rules for which policies are drawn from which set are as follows.

If the GD application has no override policy set, then the policy values from the user policy set apply. This is the default case and would apply if the enterprise was not utilising this feature.

If the application has defined any application-specific policies, then the values for these policies are drawn from the user policy set, regardless of whether the application has an override policy set.

Otherwise, for policies other than the application-specific policies, if any, the values of the override policy set apply.

These rules are illustrated in the following example scenario, with a diagram.

Scenario:

An end user is running a GD application com.example.gd.app_one

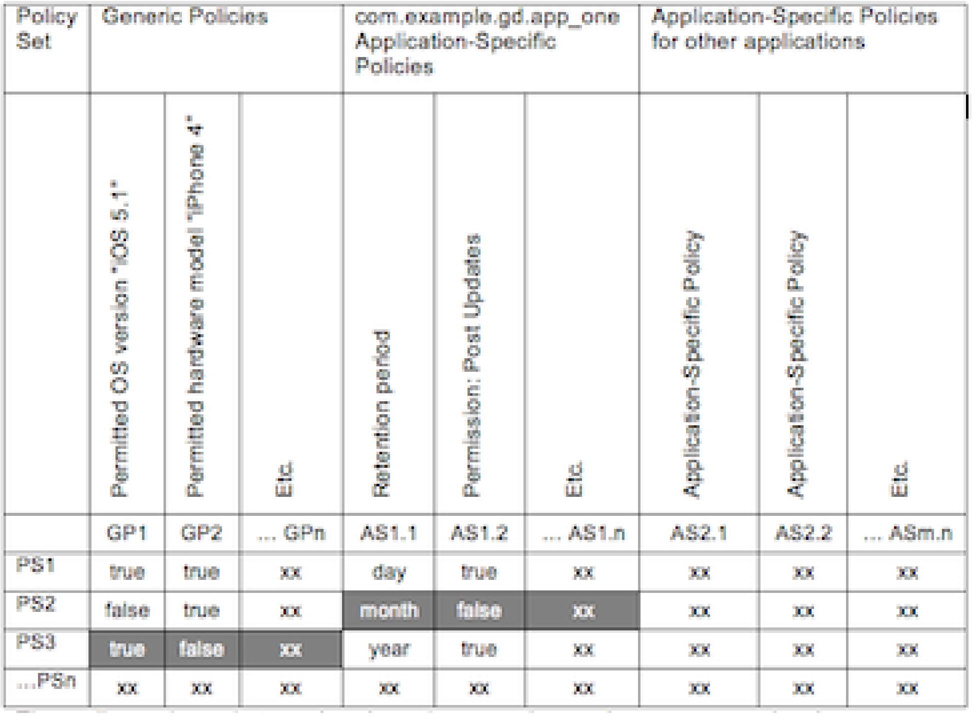
The application has defined application-specific policies

The enterprise has created a number of policy sets, here identified as PS1, PS2 … PSn

The end user has been assigned PS2 as their user policy set by the enterprise

com.example.gd.app_one has been assigned PS3 as its override policy set by the enterprise

The PS2 and PS3 policy sets have a number of differences in their policy values. A subset of these is shown in the following table.



| Policy Set | Generic Policies | | | com.example.gd.app_one Application-Specific Policies | | | Application-Specific Policies for other applications | | |
|---|---|---|---|---|---|---|---|---|---|
| | Permitted OS version "iOS 5.1" | Permitted hardware model "iPhone 4" | Etc. | Retention period | Permission: Post Updates | Etc. | Application-Specific Policy | Application-Specific Policy | Etc. |
| | GP1 | GP2 | … GPn | AS1.1 | AS1.2 | … AS1.n | AS2.1 | AS2.2 | … ASm.n |
| PS1 | true | true | xx | day | true | xx | xx | xx | xx |
| PS2 | false | true | xx | month | false | xx | xx | xx | xx |
| PS3 | true | false | xx | year | true | xx | xx | xx | xx |
| …PSn | xx | xx | xx | xx | xx | xx | xx | xx | xx |

The policy values that apply when the user is running com.example.gd.app_one are highlighted in the table above. For example:

- The end user is permitted to run the application on an iOS 5.1 device, as specified by the override policy set.
- The end user does not have permission to post updates, as specified by the user policy set.
- If the end user also had access to another application, which had no override policy set, then different policy values would apply. For example:
- The end user would not be permitted to run the application on an iOS 5.1 device.

## Override Policy Set Configuration

Policy set assignments are made in the Good Control console, like any other enterprise configuration by the administrator. This applies to user policy sets and to override policy sets.

The setting of an override policy set for an application is made on the Manage Application screen in the Good Control (GC) console user interface. This is the screen used for general configuration of GD applications at the enterprise.

## Authentication Delegation and Policy Override

BlackBerry Dynamics authentication delegation enables one application to have its end user authenticated by another. Authentication delegation is controlled by enterprise policies.

To utilise authentication delegation, the enterprise specifies an application as the authentication delegate in a policy set. When an end user to whom the policy set applies runs any GD application, the specified delegate will be invoked to authenticate the user.

Policy override could, in theory, cause an “authentication loop” problem if used with authentication delegation. An authentication loop is a situation in which two GD applications delegate authentication to each other. Neither application can then authenticate.

Abiding by the following restrictions will prevent authentication loops from arising with policy override:

Do not delegate authentication to a GD application that has an override policy set.

Vice versa, do not configure an override policy set for a GD application that is already specified as the authentication delegate, in any policy set.

## Examples of Policy Override Usage

These are fictional examples of usage of the policy override feature. The examples are for illustration purposes only and are not based on any known requirements, customers or partners.

### Example 1: Shared and Individual Devices

In this example, some devices will be shared between the manager and representatives in a retail outlet. Other devices will be used by individual travelling salespeople and not shared.

#### Example Deployment and Users

In this example, the deployment and applications are as follows:

BlackBerry Dynamics is deployed at an enterprise in the tool retailing business, Esau Drillz.

Esau Drillz also uses Good for Enterprise (GFE) for secure mobile e-mail and PIM.

Esau Drillz has deployed a number of mobile productivity GD applications, such as document viewers and editors, enterprise dashboards, and secure file sharing clients. These are known collectively as the EDproductivity applications.

Esau Drillz also has a custom GD application, EDstockroom, which displays information from the stock control database. EDstockroom gives a simple, read-only view of what items are available in the nearest stock-room to the end user. The EDstockroom display includes retail prices.

Two sets of users feature in the example: Shop Managers (SM) and Shop Representatives (SR).

An SM user has management responsibility in a particular Esau Drillz retail outlet, and also functions as a salesperson.

An SR user is a salesperson working in a particular Esau Drillz retail outlet, managed by an SM user. There are approximately twenty times as many SR users in the organisation as SM users.

### Example Access Requirements

In this example, the access requirements of the users described in the previous section are as follows.

SM users require access to GFE and all the EDproductivity applications. SM and SR users all require access to EDstockroom information.

The Esau Drillz I.T. department will support a mobile device for each SM user. The department will not support or recognise mobile devices for SR users, who are more numerous.

So, SR users require access to EDstockroom, but do not have devices on which to run the application. The policy override feature can be used to deliver this requirement, as described in the following section.

### Example Solution

The following policy configuration and working practices would deliver the requirements in the previous section.

Create a Shop Manager policy set, with the required policy configuration for SM users. This includes:

Delegate authentication to GFE.

Assign the Shop Manager policy set to every SM user.

Create a Stock Room policy set, based on the Shop Manager policy set with the following change:

Does not delegate authentication.

Assign the Stock Room policy set to the EDstockroom application.

With the above configuration, the effective policy sets would be as follows:

EDstockroom application:

Stock Room policy set applies, due to override. No authentication delegation.

Any EDproductivity application:

Shop Manager policy set applies. No override in effect. Authentication delegated to GFE.

With the above policy set configuration in place, the following working practices can be adopted.

Every shop manager sets their own GFE password. This password controls access to GFE, and to all applications in the EDproductivity suite.

Every shop manager also sets a password for the EDstockroom application that is different to their GFE password. The manager informs the sales representatives in their shop of their EDstockroom password. This means that any representative can access stock and pricing information, by using the manager's mobile device, but cannot access the manager's e-mail and PIM or other enterprise data.

If a customer in a shop were to pick up the manager's mobile device, they would not be able to access GFE, the EDproductivity suite, or EDstockroom, since they do not know either password.

As an extension to the above, additional mobile devices could be made available to shop managers, which could be passed around to any representative in the shop. GFE installation would not be required on these additional devices, if they were only used to run EDstockroom.

### Example 2: Offline and Online-only Applications

In this example, one of an enterprise's applications can only be used when the device is on-line. This application stores no data on the device.

#### Example Applications

In this example, the enterprise has deployed the following applications:

Good for Enterprise (GFE) for secure e-mail and PIM.

A suite of mobile productivity GD applications, such as document viewers and editors, and secure file sharing clients.

A custom portfolio management application, PormanMobile, with which users can buy and sell on the commodities market.

PormanMobile is a GD application that mobilises an existing enterprise application, Porman. The Porman application is web-delivered to desktop computers that are behind the enterprise firewall. The PormanMobile application communicates with the same server as the Porman desktop application.

PormanMobile does not store any data on the mobile device.

To access Porman at the desktop, a user must log in with a specific set of credentials. These credentials will be different to the user's general domain login and password. Because Porman can be used to make binding trades with real money, the application is surrounded by very strict and specific security. This security also applies to the PormanMobile application.

#### User Experience Problem and Solution

End users with access to GFE, and the mobile productivity suite, and PormanMobile, see previous section, have a particular user experience problem.

In order to secure the data stored on these users' mobile devices, a security password is required for access to GD applications in the productivity suite. So that end users do not have to remember two security passwords, the GC administrator configures authentication delegation to GFE in their policy set.

The problem is that Porman security requires that the user's credentials are always re-entered when accessing the application, and this applies equally to PormanMobile. This would mean that, in order to access PormanMobile the GD application, end users would have to enter their GFE password, and then enter their Porman credentials. This would be a poor user experience.

The solution is to create a policy that requires no security password, and does not delegate authentication to GFE, and then assign this as the application policy for PormanMobile.

With this policy override in place, users need only enter their Porman credentials to access PormanMobile. This is an absolute requirement of Porman security in any case. End users would use their GFE password to access any other GD application, or GFE.

### Example 3: Sensitive Data and Device Restriction

In this example, one of an enterprise's applications accesses more sensitive data, and only runs on a particular make and model of device.

#### Example Deployment and Users

In this example, the deployment and applications are as follows:

BlackBerry Dynamics is deployed at a hospital, for use by doctors.

The hospital has deployed a number of mobile productivity GD applications, such as document viewers and editors, and a secure browser for the hospital's intranet.

The hospital also has access to a national database of patient records, and has deployed a GD application, NatPatRec, to mobilise this.

#### Example Application Requirements

The productivity GD applications can be run on any mobile device. The data to which they give access requires an ordinary level of protection, like any enterprise data. Therefore, the end user must set a password of at least four characters.

The NatPatRec application is only suitable for use on Apple iPad devices, because of the screen size. The data that is accessed is highly confidential. Therefore, the end user must set a password of at least eight characters, with at least one number, and at least one special character. Using a longer and stronger password makes the data on the device more difficult for an attacker to decrypt.

#### Example Solution

The following policy configuration would deliver the requirements in the previous section.

Create a Doctor policy set, with the required policy configuration for the productivity applications. This includes:

Minimum password length: 4

Create a Patient Records policy set, with the required policy configuration for the patient records application. This includes:

Minimum password length: 8

Require both letters and numbers: True

Require at least one special character: True

Permitted hardware models: Apple iPad, Apple iPad 2 etc.

Any policies whose values are not mandated by the patient records application requirements are set to the same values as the policies in the Doctor policy set. In other words, the Patient Records policy set is a modified copy of the Doctor policy set.

Make the following policy set assignments.

Assign the Doctor policy set to every end user. All end users are doctors.

Assign the Patient Records policy set to the NatPatRec application.

With this configuration in place, the requirements are delivered. Doctors can set a shorter password for their general applications, but must set a longer and more complex password for confidential patient data.

## Not supported: storing PAC files on UEM or GC

A proxy auto-config (PAC) file defines how web browsers and other user agents can automatically choose the appropriate proxy server. For example, by way of application-specific policies, BlackBerry Access can be configured to use PAC files.

**Important:** Do not store PAC files on UEM server or GC server itself. This configuration is not supported.

Store PAC files on a different server that all your users' devices can access, not on the BlackBerry server.

## Device Policies

Device policies are created with the **Device Policies** screen and added to policy sets with the **Policy Sets** > **Device Management** tab.

- Creating, editing, and deleting device policies is detailed in Working with Device Policies .
- Adding devices policies to policy sets is detailed in Adding Device Policies to Policy Sets.

# Connectivity Profiles for Clients

A network *connectivity profile* for client applications is a defined set of allowed network connections, Internet domains, IP address ranges, or specific servers that your end-users' can connect to.

A connectivity profile has two basic parts:

- Settings related to your infrastructure: allowed and default Internet domains and similar settings
- Settings related to application servers: servers need by specific applications or by specific groups of users

## Design Approaches: Restrict? or Open?

The general process for working with connectivity profiles is outlined below.

1. Decide on your design approach: restrictive or open? You can decide to start with either the most restrictive access or the most unrestricted access for the majority of users. This decision is entirely yours.
2. Analyze your end user groups to determine the connectivity is needed. With the restrictive approach, what is the most restrictive, narrowest connectivity need? What specific domains should be accessible? Which groups might require more specific connections?
3. Start by defining your *baseline connectivity profile*. With the restrictive approach, this profile should be the most restrictive access for the majority of your applications and users. With the open approach, it is the inverse. The baseline profile is automatically added to all existing or newly created policy sets.
4. For those user groups that you determine need special connectivity, define *override profiles* that vary from the baseline connectivity profile to satisfy those special connections, such as adding additional domains or specific servers for these specific groups or removing access to domains already defined in the baseline connectivity profile that these user groups must not access.
5. Apply your override profiles to the pertinent policy sets to which end users have been assigned.

# Defining a Baseline Connectivity Profile

The baseline connectivity profile includes the most narrowly applicable connections to serve the majority of your end-users. You define one, and only one, base profile. You vary from the baseline profile with override profiles, discussed in Working with Override Profiles .

The baseline connectivity profile is automatically added to all existing and newly created policy sets.

The baseline profile is divided into several parts: infrastructure connections and application server connections.

## Infrastructure Connectivity

For infrastructure, the following can be configured. The allowed connections are the superset of all these settings.

| Category | Field | Description |
|---|---|---|
| Allowed Domains | Route All Traffic | Cause all network traffic to be routed through Good Proxy. See the discussion in External Web Proxy .<br>**Note: Route All** is mutually exclusive with the remainder of these settings, except you must still set the primary GP cluster for all domains. |
| | Domain | The Internet domain you want to permit access to, in the form *domain*. For example **good.com** allows access to any server in the good.com domain. GD applications are allowed to connect through your organization's firewall to any server in the domains listed here. This includes servers in the subdomains of these domains. |
| | Primary and Secondary GP Clusters | Specify the fully qualified domain name, port and priority of the Good Proxy clusters that must be used to reach the domain. |

| Category | Field | Description |
|---|---|---|
| Default Domains | Domain | The default allowed domains for this profile in the form *`domain`*. For example, **`qa.good.com`**.<br>GD applications may try to connect to an unqualified hostname like "portal" instead of using a fully qualified name like "portal.sales.xyzcorp.com". The domains in this list will be appended to unqualified hostnames to construct fully qualified names. |
| Additional Servers | Host Name | Fully qualified domain name of the desired specific servers. This is a list of specific servers that GD applications can connect to. Add servers to this list instead of using the "Allowed Domains" list if you want GD applications to only connect to certain servers, and not to every server in a domain. |
| | Port | Any required port number |
| IP Address Range | Range | A range of IP addresses to allow, in one of the following formats. GD applications can access all machines with IP addresses that fall within the ranges defined here. Address ranges must be entered with a lower and upper bound address (e.g., 192.168.2.0-192.168.2.255) or in IPv4 CIDR notation (e.g., 192.168.2.0/24).<br>• Discrete addresses: *`start, end`*<br>Example: **`10.2.3.4,10.2.3.10`**<br>• An entire subnet: **`//`***`subnet`***`/24`**<br>Example: **`//.29.102.38.1/24`** |

## Steps

**To create a baseline connectivity profile, in Good Control:**

1. Navigate to **Connectivity Profiles**.
2. Click **Master Connection Profile**.
3. To add a description of the mater connection profile, click **Edit**.
4. Otherwise, under the desired headings described above, click **Edit** then **Add** and fill in the details.
5. Click **Save** to save your changes or **Cancel** to discard them.

The baseline connectivity profile is automatically updated in all affected policy sets.

# App Server Connectivity

The application servers in the baseline connectivity profile are automatically updated with the details from Good-Dynamics-based applications that need application servers. These app servers are defined for individual applications in Good Control's **Manage Apps** > **Enterprise** tab > *edit an application* > **BlackBerry Dynamics** tab. Use of **Manage Apps** for app servers is detailed in Specifying app servers .

This means that all applications that use applications servers are accessible globally to all users in the group (such as Everyone) to which you apply the baseline connectivity profile.

In the connectivity profile, the **App Servers** tab shows those applications and server details that have been defined in **Manage Apps**. From the **App Servers for** pulldown menu, you can select a specific app to view. Also, click **Go to app** to take you to the application management page for this application.

# Working with Override Profiles

For those applications that require special connectivity, you can define override profiles that are variations on the baseline connectivity profile. Override profiles inherit the settings of the baseline profile.

Think of your specific user groups. For example: what additional servers do they need to connect to that are not accommodated by the baseline connectivity profile? Any special IP address ranges or subnets that specific groups need to access? Are their default domains different?

After you create an override profile you need to apply it to the policy sets for these specific user groups.

**To create a connectivity override profile, in Good Control:**

1. Navigate to **Connectivity Profiles**.
2. Click **New Override Profile**.
3. Enter a name and description for this override profile.
4. Determine which category of settings you want to modify, click **Edit**, then, click **Add.**
5. Enter a name and description for this override profile.
6. Edit the desired fields as described in Infrastructure Connectivity . To remove a domain or server from those defined in the baseline connectivity profile, click the circled X on the far right of the domain or server name.
7. Edit the desired fields as described in App Server Connectivity .To remove a domain or server from those defined in the baseline connectivity profile, click the circled X on the far right of the domain or server name.
8. Click **Save** to save your changes or **Cancel** to discard them.

## Applying an Override Profile to a Policy Set

**To specify an override profile for a policy set, in Good Control:**

1. Navigate to **Policy Sets** > **Apps** tab.
2. Find the desired policy set in the list and click it to edit it.
3. Expand the heading **Cross-App Policies**.
4. For **Connectivity Profile**, from the pulldown menu, select the name of the desired override policy.
5. Click **Update** to save your changes or **Cancel** to discard them.

# Managing Connectivity Profiles: Editing and Deleting

**To edit a connectivity profile, in Good Control:**

1. Navigate to **Connectivity Profiles**.
2. In the list of profile overrides, find the override profile you want to edit.

3. Click the profile name.
4. Click the **Edit** button for the appropriate block of settings.
5. Change the profile definitions however you like, with fields described in Infrastructure Connectivity and App Server Connectivity .
6. Click **Update** to save your changes or **Cancel** to discard them.

**To delete a connectivity profile or override profile, in Good Control:**

You cannot delete a connectivity profile that is in active use (that is, which is currently applied to a policy set). You must first remove the connectivity profile from the policy set before you can delete it. Connectivity profiles that are in active use are indicated by an information button (circled **i**) in the list of profiles.

1. Navigate to **Connectivity Profiles**.
2. In the list of profile overrides, find the override profile you want to delete.
3. On the far right, click the **X** associated with the override.

   > **Note:** If there is a circled **i** instead of an **X**, the override is in active use and cannot be deleted. Click the **i** to see which policy sets use the profile.

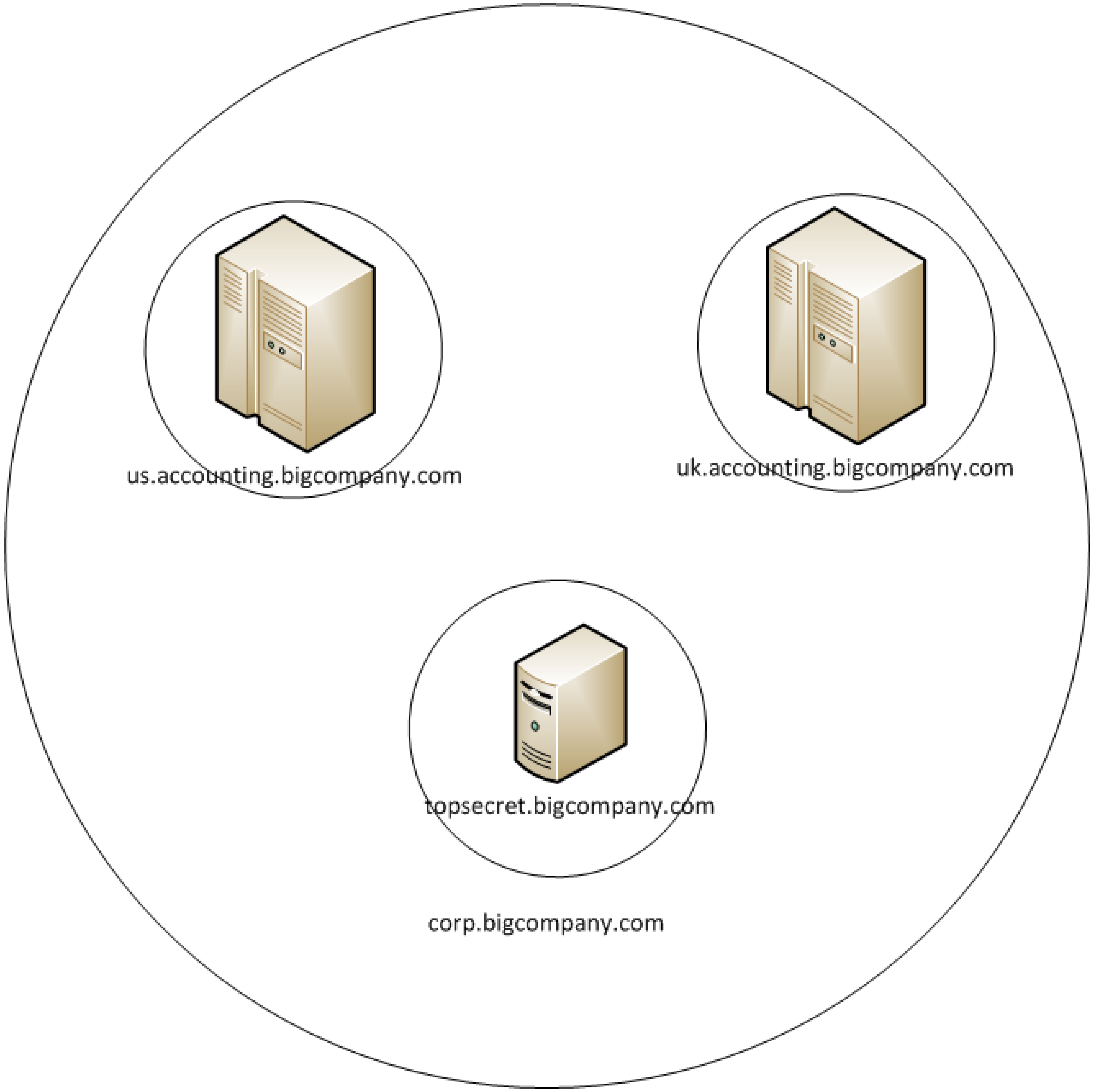
4. Click **OK** to delete the profile or **Cancel** to keep it.

## Example: A Simple Connectivity Profile

Here is some simple example that maps an easy-to-understand network topology to the connectivity profiles, so you can see how they are designed. This design uses the "restrictive" design approach, not the "open" approach.

Imagine a company that has network domains as follows:

- **`corp.bigcompany.com`**: this domain is needed by all users.
- **`us.accounting.bigcompany.com`** and **`uk.accounting.bigcompany.com`**: These domains should be accessible only by the Accounting departments in the United States and the United Kingdom.
- **`topsecret.bigcompany.com`**: this domain should be accessible only by the Top Secret Projects department.



This configuration relates to the **Allowed Domains** portion of the connectivity profile. Here are the high-level steps for creating profiles for this configuration.

## Baseline Connectivity Profile

1. In your baseline connectivity profile, to **Allowed Domains**, add `corp.bigcompany.com`. All users must be able to access this domain. The baseline connectivity profile is always added to the default policy set.

2. Be sure that the policy set has been applied to all applicable user groups.

### Accounting

1. Now create an override profile for the Accounting department. In it, under **Allowed Domains**, add **`accounting.bigcompany.com`**. The accountants in the US and the UK have equal access.
2. Now apply the override profile to the policy set used by the Accounting user group.

### Top Secret

1. The users in the special project need to access their top secret domain. Create another override profile and in its **Allowed Domains**, add **`topsecret.bigcompany.com`**.
2. Then be sure to apply the override policy to the policy set that is used by the Top Secret user group.

# Servers

## Managing GC, GP, and logging server properties

The Good Control console displays links in the navigation bar for managing server properties:

- GC Server Properties
- GP Server Properties
- Logging Properties

## GC Server Property Reference

Last updated: 9/25/2017

This is reference for all GC server properties. The properties are grouped by "area", that is, what they relate to in general. Also included is whether a change to a property values requires a GC service restart to take effect.

### Global Properties

Some properties are global in scope, which is indicated in the reference table. The values of these properties are used across all of the GC servers in a given cluster and include properties related to the following:

- User self service functions
- Active Directory settings the GC servers use to search for new users
- GD NOC server locations and connection configurations
- Most settings related to Kerberos constrained delegation (KCD)

## Certificate Management

| Property | Description | Default, Global, Restart |
|---|---|---|
| gc.user.keystore.ttl.seconds | For the GC server, time-to-live in seconds for the keystore for individual end-users PKCS 12 certificates. | Default: 86400<br>Global: yes<br>Restart: yes |

## Communication

| Property | Description | Default, Global, Restart |
|---|---|---|
| cap.soap.url | Endpoint for SOAP requests that use the cap.wsdl file<br>**Note:** Not editable. | Varies by release |
| cntmgmt.external.port | Port for container management service | Default: 17317<br>Global: yes<br>Restart: yes |
| cntmgmt.internal.port | Internal binding for above | Default: 17317<br>Global: true<br>Restart: true |
| cntmgmt.max.active.sessions | Maximum number of active sessions for container management | Default: 10000<br>Global: yes<br>Restart: yes |
| cntmgmt.max.conns.above.limit | Number of connections allowed above the stated limit in property cntmgmt.max.conns.persec<br>**Important:** Do not alter this setting without direct consultation with BlackBerry. | Default: 3<br>Global: yes<br>Restart: yes |
| cntmgmt.max.conns.persec | Maximum number of connections per second for container management<br>**Important:** Do not alter this setting without direct consultation with BlackBerry. | Default: 30<br>Global: yes<br>Restart: yes |
| cntmgmt.max.idle.count | Maximum number of Allowed idle connections for container management<br>**Important:** Do not alter this setting without direct consultation with BlackBerry. | Default: 0<br>Global: yes<br>Restart: yes |

| Property | Description | Default, Global, Restart |
|---|---|---|
| cntmgmt.max.read.throughput | Maximum number of concurrent read operations for container management<br>**Important:** Do not alter this setting without direct consultation with BlackBerry. | Default: 500<br>Global: yes<br>Restart: yes |
| cntmgmt.max.write.throughput | Maximum number of concurrent write operations for container management<br>**Important:** Do not alter this setting without direct consultation with BlackBerry. | Default: 500<br>Global: yes<br>Restart: yes |
| cntmgmt.ssl.external.enable | Enable SSL for external container management | Default: True |
| cntmgmt.ssl.internal.enable | Enable SSL for internal container management | Default: True |
| gc.event.push.count | For the GC server, count of pushes of events<br>**Important:** Do not alter this setting without direct consultation with BlackBerry. | Default: 10<br>Global: yes<br>Restart: yes |
| gc.event.push.interval | For the GC server, interval between event pushes<br>**Important:** Do not alter this setting without direct consultation with BlackBerry. | Default: 5 seconds<br>Global: yes<br>Restart: yes |
| gc.krb5.debug | Whether or not GC is configured to log additional information for debugging purposes. Check this box to enable additional logging, or uncheck the box if you do not need extra logging. | Default: false<br>Global: no<br>Restart: yes |
| gc.krb5.enabled | Whether or not Kerberos Constrained Delegation (KCD)support is enabled in the GC server. Check this box if you want your GC servers to use KCD. | Default: false<br>Global: no<br>Restart: yes |
| gc.krb5.kdc | The fully qualified domain name of the server where the Kerberos Key Distribution Center (KDC) service resides. | No default<br>Global: no<br>Restart: yes |
| gc.krb5.config.file | The location of the krb5.conf file on the GC host machine. See Kerberos Constrained Delegation. | No default<br>Global: no<br>Restart: yes |
| gc.krb5.keytab.file | The location of the keytab file on the GC host machine. | No default<br>Global: no<br>Restart: yes |

| Property | Description | Default, Global, Restart |
|---|---|---|
| gc.krb5.principal.name | The Kerberos principal account used in the steps above. Specify the username without the domain or realm. | No default<br>Global: no<br>Restart: yes |
| gc.krb5.realm | The realm of the Kerberos principal account. | No default<br>Global: no<br>Restart: yes |
| gc.smtp.email | The email address that sends your users' activation emails. If this value is No default, GC sends emails from the do_not_reply@yourdomain mailbox. However, some mail servers are configured to reject all emails that originate from an invalid email account, so with this property you can supply a valid email address. | Default: do_not_reply@yourdomain<br>Global: no<br>Restart: yes |
| gc.smtp.host | The fully qualified domain name for your mail server. | No default<br>Global: no<br>Restart: yes |
| gc.smtp.password | A secure property that contains the password for the mail server user. If your mail server does not require authentication, this property is not used. The value is obfuscated. | No default<br>Global: no<br>Restart: yes |
| gc.smtp.port | The mail server port number. | Default: 25<br>Global: no<br>Restart: yes |
| gc.smtp.ssl | Boolean whether the SMTP server runs SSL or not. | Default: false<br>Global: no<br>Restart: yes |
| gc.smtp.user | If needed, the account the GC server uses to log into the mail server. | No default<br>Global: no<br>Restart: yes |
| gcsvc.max.reqs.persec | For the GC server, maximum number of requests per second<br>**Important:** Do not alter this setting without direct consultation with BlackBerry. | Default: 30<br>Global: yes<br>Restart: yes |
| gcsvc.throttle.wait | For the GC server, throttling interval in seconds for request processing<br>**Important:** Do not alter this setting without direct | Default: 10000<br>Global: yes<br>Restart: yes |

| Property | Description | Default, Global, Restart |
|---|---|---|
| | consultation with BlackBerry. | |
| mdc.server.url | URL of the Mobile Data Conduit.<br>**Note:** Not editable. | Global: yes |

## Directory

| Property | Description | Default, Global, Restart |
|---|---|---|
| authenticator.adsi.domains.additional | Names of additional domains you want to search | Default: none<br>Global: no<br>Restart: yes |
| authenticator.adsi.domains.undesired | List Active Directory domains you want to avoid adding users from.<br>Extended example: When an administrator for Xyzcorp first installs the GC server, she sets the GC service to run under an account in the admins.xyzcorp.com domain but does not check the “Use Trusted Domains” checkbox. Consequently, the value of this property is false. GC searches for new users only in the admins.xyzcorp.com domain. If the administrator decides to add users from additional domains, she modifies the server properties as follows:<br>directory.adsi.trusted.domains = (checked) authenticator.adsi.domains.additional = sales.xyzcorp.com,chicago.xyzcorp.com,boston.xyzcorp.com<br>If the administrator needs to restrict GC from finding users in certain domains, she modifies the server properties as follows:<br>authenticator.adsi.domains.undesired = test.xyzcorp.com,qa.xyzcorp.com,mars.xyzcorp.com | Default: none<br>Global: no<br>Restart: yes |
| authenticator.type | For user authentication, type of directory.<br>**Note:** Not editable. | Default: ADSI<br>Global: yes<br>Restart: yes |
| directory.adsi.domain.name | Name of the Active Directory domain for a GC server | Default: value entered during installation |

| Property | Description | Default, Global, Restart |
|---|---|---|
| | | Global: no<br>Restart: yes |
| directory.adsi.email.domain | Determines the single domain where GC searches for users' email addresses. This is the one and only domain searched. | Default: none<br>Global: no<br>Restart: yes |
| directory.adsi.search.emailsearchorder | A comma-delimited, case-sensitive, ordered list of Active Directory attributes that GC searches to find user email addresses. GC searches the attributes in this list in order until it finds a valid email address for the AD user. The default value for this property is comprised of the attributes proxyAddresses,targetAddress,userPrincipalName,mail. If you modify this list, make sure each attribute is entered correctly; otherwise, your GC servers cannot properly search for new users. | Default: proxyAddresses,targetAddress,userPrincipalName,mail<br>Global: no<br>Restart: yes |
| directory.adsi.search.fetchmore | To decrease load on your AD server, you can set this property to the number of "batches" to use to get data from AD. This is particularly useful when the size of your data in AD is too large to retrieve in a single transaction.<br>Maximum: 8 | Default: 3 |
| directory.adsi.search.mb.enabled | Reflects whether or not your Active Directory users have Exchange mailboxes. Check this checkbox if your Active Directory users have Exchange mailboxes. Within Exchange, the email address is always bound to the proxyAddress attribute. An Exchange mailbox can have multiple proxyAddress attributes, and GC selects the first valid one it finds. Uncheck this checkbox if you do not use Exchange for your users' mailboxes, or if your users' email addresses are located in different attributes. Configure the following property if you want GC to look for user email addresses in specific attributes. | Default: false<br>Global: no<br>Restart: yes |
| directory.adsi.search.sizelimit | The maximum number of users that GC retrieves at one time when a GC administrator wants to add all users from an AD group into GC. This includes users in the subgroups of the selected group. Only users who do not already have GC accounts are returned in the results. | Default: 10000<br>Global: no<br>Restart: yes |

| Property | Description | Default, Global, Restart |
|---|---|---|
| | For example, if a GC administrator selects a group named “Sales” and this property has a value of 5,000, GC returns up to 5,000 users from that group and its subgroups that have not already been imported into GC. You can increase or decrease this number, but be aware that increasing this number can increase the time to display the list of users. The default value is 10000. | |
| directory.adsi.search.sizelimitname | The maximum number of hits that GC displays per domain it searches. Only users who do not already have GC accounts are included in the results. For example, if this property is set to 100 and GC searches for users in 3 trusted domains, the GC console displays up to 100 matching users per domain, for a maximum of 300 users. You can increase or decrease this number, but be aware that increasing this number can noticeably extend the amount of time you have to wait before results are displayed. The default value is 100. | Default: 100<br>Global: no<br>Restart: yes |
| directory.adsi.search.timelimit | This is the maximum number of seconds that GC waits for results before displaying the available results. If GC reaches this time limit but has not yet received a full set of results, it displays a truncated list of results and an alert message. | Default: 10<br>Global: no<br>Restart: yes |
| directory.adsi.trusted.domains | By default, a GC server searches for new users in the same Active Directory domain as the account running the GC service. This property determines where GC searches for new users. The value is initially set during installation; it is checked if the “Use Trusted Domains” checkbox was checked in the installer interface. Uncheck this property's checkbox if you want GC to only search for new users in the same Active Directory domain as the account running the GC service. Check this property's checkbox if you need to add users to GC from additional Active Directory domains. | Default: false<br>Global: no<br>Restart: yes |
| directory.adsync.access.interval | Throttling Interval between two Active directory queries when syncing the changes from AD to GC | Default: 200 ms<br>Global: yes<br>Restart: yes |
| directory.adsync.polling.ena | Whether to synchronize user records and | Default: true |

| Property | Description | Default, Global, Restart |
|---|---|---|
| bled | their attributes with Active directory | Global: yes<br>Restart: yes |
| directory.adsync.polling.interval | Scheduled interval after which GC will contact Active directory to sync new changes. Example: If the GC spends 30 minutes scanning, this property is set to 2 hours, and a scan starts at 7:00, the next scan will occur at 9:30, and the next at 12:00, and so on. | Default: 7200 seconds<br>Global: yes<br>Restart: yes |
| directory.type | For adding users, type of directory.<br>**Note:** Not editable. | Default: ADSI<br>Global: yes<br>Restart: yes |
| directory.updateEmailAddressTask.interval | Deprecated. Do not use this property. | Do not use this property. |

## Duplicate Containers and Purge Inactive Containers

The server properties for managing duplicate containers and inactive have been separated into sets of properties for each function.

| Property | Default, Global, Restart |
|---|---|
| Enable job to automatically remove duplicate containers (on/off)<br>How often to run the job to automatically remove inactive containers. In seconds. | Default: On<br>Global: yes<br>Restart: no |
| Frequency in seconds that job to remove duplicate containers will run. | Default: 86400<br>Global: yes<br>Restart: no |
| Inactivity timeout in seconds before duplicate container is deleted. | Default: 259200<br>Global: yes<br>Restart: no |
| Maximum number of containers to remove in a single job | Default: 100<br>Global: yes<br>Restart: no |

### Purge Inactive Containers

| Property | Default, Global, Restart |
|---|---|
| Container inactivity interval in seconds. | Default: 7776000<br>Global: yes<br>Restart: no |
| Enable job to automatically remove inactive containers (on/off) | Default: Off<br>Global: yes<br>Restart: no |
| Frequency in seconds that job to remove inactive containers will run. | Default: 86400<br>Global: yes<br>Restart: no |
| Interval in seconds that container inactivity times will be adjusted forward by the downtime to allow for reconnection. | Default: 86400<br>Global: yes<br>Restart: no |
| Maximum number of containers to remove in a single job | Default: 100<br>Global: yes<br>Restart: no |

## GC Console Login

| Property | Description | Default, Global, Restart |
|---|---|---|
| Enable Kerberos Single Sign-On | Enables Kerberos SSO for GC console login. | Default: not enabled<br>Global: yes<br>Restart: no |
| Single Sign-On is required. Entering a password will not work. | Disallows fallback to passwords if Kerberos SSO does not succeed. | Default: not enabled<br>Global: yes<br>Restart: no |
| Domain: Pre-populate domain field | The GC console login page has the Domain field, which is used as part of authentication for logging in.<br>You can "hard code" the value for the Domain field so that your users and administrators do not have to remember it. | Default: none<br>Global: yes<br>Restart: no |

## Email Templates

| Property | Description | Default, Global, Restart |
|---|---|---|
| Forgot password email on/off | Enable sending of forgotten password email | Default: true<br>Global: yes<br>Restart: no |
| Forgot password email body | Body of the mail for forgotten passwords | - Default: see text below.<br>- Global: yes<br>- Restart: no<br>- Text:<br>Good Control received a request to reset your password. If you did not make this request, ignore this email.<br>To reset your password, follow the instructions at this link, which expires after <%DEFINED_EXPIRY_TIME_FOR_THIS_GC%>.<br><%GC_REG_URL%> .<br>Thank you,<br>Good Control |
| Forgot password email sender | Email address for sender of forgotten email | Default: BlackBerry Mobile Administrator<br>Global: yes<br>Restart: no |
| Forgot password email subject | Subject line of the forgotten password email | Default: Password from Good Control<br>Global: yes<br>Restart: no |
| Unlock email | Enable/disable sending of "unlock emails" | Default: Enabled<br>Global: yes<br>Restart: no |
| Unlock email body | Active Directory domain specified during installation | Default: Dear <%HELPDESK_REF%>,<br>You can unlock your BlackBerry Dynamics Mobile Application <%APPLICATION_NAME%> provided by your company.<br>This email contains your UNLOCK ACCESS KEY and instructions for unlocking the mobile application.<br>Enter the following information when prompted (not case sensitive):<br>EMAIL ADDRESS: <%EMAIL_ADDRESS%><br>UNLOCK ACCESS KEY: <%PIN_FULL%> |

| Property | Description | Default, Global, Restart |
|---|---|---|
| | | Your Unlock Access Key expires: <%EXPIRY%><br>For further assistance please contact your IT department.<br>Global: yes<br>Restart: no |
| Unlock email sender | Username of the GC administrator specified at installation | Default: BlackBerry Mobile Administrator<br>Global: yes<br>Restart: no |
| Unlock email subject | Subject line of unlock email | Default: Unlock BlackBerry Dynamics Mobile Application <%APPLICATION NAME%><br>Global: yes<br>Restart: yes |

## Miscellaneous

| Property | Description | Default, Global, Restart |
|---|---|---|
| access.email | Deprecated. Do not use this property. Instead, use the gc.disable.email property. | Do not use this property. |
| allow.new.android.device | Allow any new Android device | Default: true<br>Global: yes<br>Restart: no |
| allow.new.iOS.device | Allow any new iOS device | Default: true<br>Global: yes<br>Restart: no |
| allow.new.MAC.device | Allow any new macOS device | Default: true<br>Global: yes<br>Restart: no |
| allow.new.Windows.device | Allow any new Windows devices other than Windows Phone, such as Windows tablet | Default: true<br>Global: yes<br>Restart: no |
| Comma separated list of tables to be uploaded in log during diagnostic upload | List of database tables<br>**Note:** Do not change these values. | Default: t_gc_servers,t_gc_gp_routes,t_gc_gp_route_servers<br>Global: yes<br>Restart: yes |

| Property | Description | Default, Global, Restart |
|---|---|---|
| Enable upload of additional diagnostic info | Include more information over and above what is normally uploaded for diagnostics | Default: disabled<br>Global: yes<br>Restart: yes |
| Enable upload of table names in diagnostics | Include names of database tables in uploaded diagnostic information | Default: disabled<br>Global: yes<br>Restart: yes |
| gc.admin.domain | Active Directory domain specified during installation | Default: none<br>Global: yes<br>Restart: no |
| gc.admin.user | Username of the GC administrator specified at installation | Default: none<br>Global: yes<br>Restart: no |
| gc.all_device_rules_migrated | For MDM, device rules that have been migrated into core GC<br>**Note:** Not editable | Default: false<br>Global: yes<br>Restart: yes |
| gc.disable.emails | Disable sending of email from the GC to end-users | Default: false<br>Global: yes<br>Restart: no |
| gc.entgw.report.userinfo | Whether user display names are reported to GD NOC | Default: false<br>Global: yes<br>Restart: no |
| gc.health.check.enabled | Whether to perform additional checks on GC health | Default: false<br>Global: yes<br>Restart: yes |
| gc.health.check.interval | How often to check GC health | Default: none<br>Global: yes<br>Restart: yes |
| gc.logs.dir | Where the GC stores its logfiles | Default: c:\good\gclogs<br>Global: no<br>Restart: yes |
| gc.security.realms |  | Default: ADSI,GC<br>Global: yes |

| Property | Description | Default, Global, Restart |
|---|---|---|
| | | Restart: yes |
| gc.server.name | Name of the GC server | Default: Canonical hostname<br>Global: no<br>Restart: yes |
| gc.user.keystore.ttl | | Default: 86400000<br>Global:<br>Restart: ? |
| gcs.logfile.days | Maximum number of days for log file retention | Default: 10<br>Global: yes<br>Restart: yes |
| gd.product.domain | Domain of the GC | Default: same as Active Directory domain<br>Global: no<br>Restart: yes |
| gd.product.enterprise.name | Name of the enterprise where this GC is installed | Default: none<br>Global: yes<br>Restart: yes |
| gd.product.host.url | URL of this to use in user self service email | Default: none<br>Global: yes<br>Restart: yes |
| gd.product.version | Version number of this GC | Default: none<br>Global: no<br>Restart: yes |
| gd.security.keystore.alias | Alias for the GC's keystore | Default: gc<br>Global: no<br>Restart: yes |
| gd.security.keystore.file | Location of GC keystore file | Default: C:\BlackBerry\Good Control\jre\lib\security\cacerts<br>Global: no<br>Restart: no |
| gd.security.rootcert.alias | Alias for the root certificate of the GC | Default: gdca<br>Global:<br>Restart: |

| Property | Description | Default, Global, Restart |
|---|---|---|
| Minimum time interval between two status requests in milliseconds. | Allowable frequency of access to /gc/status URL. Any request more frequent than this is rejected with HTTP code 503 | Default: 1000<br>Global: yes<br>Restart: no |
| policy.app.interval | Frequency of the GC retrieval from the GD NOC application policies for all policy sets. | Default: 1440 minutes<br>Global: yes<br>Restart: yes |
| policy.compliance.interval | Frequency of the GC retrieval from the GD NOC compliance policies for all policy sets. | Default: 1440 minutes<br>Global: yes<br>Restart: yes |
| policy.compliance.url | | Default: https://*fqdn_of_host*/depot/policy<br>Global: yes<br>Restart: no |
| allow.new.iOS.device | Allow any new iOS device | Default: true<br>Global: yes<br>Restart: no |

## Duplicate Containers and Purge Inactive Containers

To ease the administrative burden of managing containers on devices, Good Control automatically identifies inactive ("stale") or duplicate containers and schedules batch jobs to remove them. This relieves the IT administrator from having to deal with this housekeeping task.

A container is considered duplicate if there is another container on the same device with the same combination of user ID and GD Entitlement ID (also known as GD App ID).

By default, a container is considered inactive if it has not connected to the GC in 90 days. Also by default, the container management batch job runs once a day to determine if a container's last connection time exceeds the inactivity threshold and thus should be removed. Deletions are recorded in the GC log.

As a "safety factor" to account for system downtime in the calculation of inactivity, you can set a certain amount of time to adjust the calculation forward to accommodate devices that might have attempted to reconnect during that downtime. By default, this "drift" is one day.

### Duplicate containers

| Property | Default, Global, Restart |
|---|---|
| Enable job to automatically remove duplicate containers (on/off)<br>How often to run the job to automatically remove inactive containers. In seconds. | Default: On<br>Global: yes<br>Restart: no |

| Property | Default, Global, Restart |
|---|---|
| Frequency in seconds that job to remove duplicate containers will run. | Default: 86400<br>Global: yes<br>Restart: no |
| Inactivity timeout in seconds before duplicate container is deleted. | Default: 259200<br>Global: yes<br>Restart: no |
| Maximum number of containers to remove in a single job | Default: 100<br>Global: yes<br>Restart: no |

Purge inactive containers

| Property | Default, Global, Restart |
|---|---|
| Container inactivity interval in seconds. | Default: 7776000<br>Global: yes<br>Restart: no |
| Enable job to automatically remove inactive containers (on/off) | Default: Off<br>Global: yes<br>Restart: no |
| Frequency in seconds that job to remove inactive containers will run. | Default: 86400<br>Global: yes<br>Restart: no |
| Interval in seconds that container inactivity times will be adjusted forward by the downtime to allow for reconnection. | Default: 86400<br>Global: yes<br>Restart: no |
| Maximum number of containers to remove in a single job | Default: 100<br>Global: yes<br>Restart: no |

## Reporting

| Property | Description | Default, Global, Restart |
|---|---|---|
| gc.reports.limit | For limiting the lines in reports to prevent out of memory condition. | Default: 5000 |

| Property | Description | Default, Global, Restart |
|---|---|---|
| | Maximum: 1,000,000 | Global: yes<br>Restart: no |

## Discussion of miscellaneous server properties

### Customizing the Forgotten Password Email Template

Users can request a password reset if they have forgotten their passwords.

At **Servers** > **Server Properties** tab, you can edit the template for the "forgotten password" email.

In the list of properties, find the property named **`forgotpassword.email.body`**.

> **Note:** Be careful not to change any of the variables embedded in the email template.

By default, the email templates are formatted as plain text.

You can also format your email templates as HTML. Any valid HTML 4 or HTML 5 can be used.

Keep the following in mind:

- The size of the template is limited to 4,000 characters, including both tags and text.
- To specify HTML formatting, add this as the first line: **`<!DOCTYPE html>`**. Otherwise the system treats the template as plain text.
- Follow the HTML document type declaration, with **`<head>`**, **`<body>`** and any other desired HTML tags. Be sure to use closing tags (like **`</body>`**) for normalized HTML.
- Before you enter the HTML into the template form in Good Control, be sure to make sure it is valid HTML. Good Control does not validate the HTML. If GC encounters invalid HTML in the template, the message is sent as plain text.

  - Be careful to keep the embedded variable names that GC requires in the text, but you can format them however you like. Formatting example: a single paragraph with bold email address: **`<p>Your email address is <b><%EMAIL_ADDRESS%></b></p>.`**
- All links to CSS, images, or other resources on the internet must be absolute and must be reachable by your end-users' browsers or email clients. That is, the HTML in the template is not relative to a document root, as it would be on a standard web server:

  Images can be base64-encoded and included in the template's **`<img>`** tags, as in the following example snippet: **`<img src="data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAADIA..."">`**

  - Any CSS must be defined in the **`<head>`** or inline in the template.

### Routing All Traffic Through Good Proxy: "Route All"

In Good Control, in **Connectivity Profiles** > **Allowed Domains**, you can specify the servers your users' GD applications are allowed to access through your firewall.

With the **Route All** configuration, all traffic, regardless of domain or subnet, is routed through the Good Proxy server.

**Route All** is useful for two particular needs (among others):

1. Enabling free access for web browsers on devices (as opposed to applications). There is no easy way to configure access for web browsers.
2. Enforcing security: routing all traffic through the GP allows for easier monitoring.

### Setting Route All

**To route all traffic, in Good Control:**

1. Navigate to **Connectivity Profiles**.
2. Click the name of the base connectivity profile.
3. Under **Allowed Domains**, click the **Route All** checkbox.
4. Under the **Domains** heading, for the *** (All Domains)** entry, from the pulldown menus, select the name of the primary and secondary GP clusters. You must set at least the name of the primary GP cluster.
5. Click **Add**.
6. Click **Save** to save your changes or **Cancel** to discard them.

### Effects of Route All

Be advised that enabling Route All can have an adverse impact on your deployment, especially if you have previous network configurations or a web proxy for external access already configured. These points are detailed under Additional Considerations

Setting **Route All** has the following effects:

- Any configurations you had previously defined are grayed out in the GC console to indicate that **Route All** is in effect but these previous configurations are still active. To change those other configurations, uncheck **Route All**, make your changes, and then re-check **Route All**.
- GD clients on mobile devices can connect to any servers behind the enterprise firewall that are reachable by the GP server.
- Establishing connections to servers on the external Internet can take longer.
- Older applications that were not built with the latest version of the GD SDK (at least v1.8.x) do not have the **Route All** feature. Such old applications still rely on any specific routing configurations you have in place. You should recompile with the latest version of the GD SDK, but to accommodate such older applications, the GD service includes rules for many, but not all, of the Internet's top-level domains, as shown below. These are specified in the GC property **`gc.route.all.domains`**, which you can edit to include other domains (see Updating GC Server Properties for details):

  com, org, net, int, edu, gov, mil, us, uk, de, fr, nl, cn, jp, in, au, nz, eu

- Your service might be in a part of the world that these rules do not cover. In this case, for your older clients, in addition to **Route All**, you should also create specific configurations for the domains you need to accommodate (see

External Web Proxy ).

### Additional Considerations

Consider the following points before you enable **Route All**.

#### External Web Proxy

If you are using a web proxy to allow access to external sites and have restrictions already configured in your proxy to restrict certain sites, when you enable Route All, you need to set the proxy properties in Good Proxy.

> **Important:** Without these changes to GP, your applications will not connect. Access to external web sites will break.

Specifically, you need to edit the GP file `C:\good\gps.properties` to set the **`proxy.use`** property and specify the accessible external sites or Internet domains in the **`proxy.urls`** property, among other properties such as port numbers and so on. For details, see the Good Control console help topic **Basic Server Settings** > **Configuring Web Proxy Server Properties for GC or GP**, subsection on the GP properties file.

#### BlackBerry Access vs Other Applications

BlackBerry Access can be configured with a Proxy Access Control (PAC) file that determines allowable sites. In this case, Route All has no effect; the PAC file determines the proxy settings.

Other applications without the PAC file, however, require that the GP proxy properties be set to allow or deny access to external sites, as described above.

## Configuring GC for Kerberos Constrained Delegation

Good Control can be configured for Kerberos Constrained Delegation (KCD). A prerequisite for KCD in GC is that your organization is already set up to use KCD; described here is how to configure your GC servers for KCD.

> **Note:** This feature does not relate to Kerberos SSO, which deals with authentication for login to the Good Control console itself. For details about Kerberos SSO, see BlackBerry Access Secure Browser .
>
> KCD relates exclusively to user authentication in client applications.

These are steps to set certain properties related to KCD. We do not explain KCD authentication itself nor how to implement it in your organization. Consult our published guide on possible configuration/deployment options: Kerberos Constrained Delegation.

For this procedure, GCSvc is the suggested value for the Service Principal Name (SPN); if you choose to use a different value, replace GCSvc with your value throughout all of the following steps.

## Configuring GC for Kerberos Single Sign-On (SSO) to Console

For logging into the Good Control console, you can configure Good Control to rely on Kerberos Single Sign-On (SSO).

> **Note:** This feature does not relate to Kerberos Constrained Delegation (KCD), which deals with user authentication in GD-based applications. For details about KCD, see Kerberos Constrained Delegation.
>
> Kerberos SSO relates exclusively to login to Good Control itself.

You have granular control by way of GC server properties:

- Allow SSO but fallback to password if SSO does not succeed for some reason.
- Optionally, allow SSO exclusively, without fallback to password. This is considered the most secure configuration.

### Behavior and Recommendations

The behavior of Kerberos SSO for the user logging into the GC is extremely different than logging in with username and password: it appears "instantaneous", with no interaction by the user (which is the entire reason for using Kerberos SSO). This "instantaneousness" persists even if the web browser is closed and reopened during a session.

This behavior can be alarming to end users who are not made aware of it.

> **Important:** Because of this behavior, you should be sure to lock your workstation whenever you leave it unattended, whether your browser is running or not.

### Setup of Kerberos SSO

Your Active Directory system running Kerberos must be configured with Microsoft's **`setspn`** tool on your Active Directory servers to recognize your Good Control servers as Service Principal Names (SPNs).

**On your AD system, for each Good Control server in the cluster, issue the following commands:**

1. Set the fully qualified domain name of the Good Control server as an SPN.

   > **Note:** In this command, *ADdomainUser* is the name of the service account that runs Good Control on this server.

   ```
   setspn.exe -A HTTP/gcHostname.someDomain.com ADdomainName\ADdomainUser
   ```

   **Example:** `setspn.exe -A HTTP/mygc.europe.bigCompany.com europe\euadmin`

2. Set the bare hostname of the Good Control server as an SPN.

   > **Note:** In this command, *ADdomainUser* is the name of the service account that runs Good Control on this server.

   ```
   setspn.exe -A HTTP/gcBareHostname ADdomainName\ADdomainUser
   ```

   **Example:** `setspn.exe -A HTTP/mygc europe\euadmin`

### Setup in Good Control for Kerberos SSO

You need to enable Kerberos SSO in Good Control's server properties.

**To enable Kerberos SSO in Good Control:**

1. Navigate to **Servers** > **Server Properties** tab.
2. Scroll to find the following properties:

| GC Server Property | Description |
|---|---|
| Enable Kerberos Single Sign-On | Enables Kerberos SSO for GC console login. |

| GC Server Property | Description |
|---|---|
| Single Sign-On is required. Entering a password will not work. | Disallows fallback to passwords if Kerberos SSO does not succeed. |

3. Check the associated check box to enable Kerberos SSO and the desired optional features.
4. In the upper right, click **Submit** to save your changes, or navigate away from the page to discard them.

### Browser Setup for Kerberos SSO

The administrators who intend to login to Good Control with Kerberos SSO must configure their browsers.

> **Note:** Due to a Microsoft limitation, Kerberos SSO does not work if your run your browser on the same machine as the Kerberos service.

The steps below depend on the version of your browser and might not match the actual clickpaths you need to follow.

#### BlackBerry Access Secure Browser

With BlackBerry Access secure browser for mobile devices or desktop, no special configuration is required as long as the GC server is running securely with HTTPS.

#### Google Chrome

In the commands below, *domain.com* is the same as the Internet domain you specified with `setspn` in Setup of Kerberos SSO . For example: `*.europe.bigCompany.com`.

- On Microsoft Windows, you need to determine the full path to the installed executable file and run this command in a command window: `\path\to\installed\chrome.exe --args --auth-server-whitelist="*.domain.com"`
- On Mac OS X, in a shell: `cd /Applications; open -n -a 'Google Chrome.app' --args --auth-server-whitelist="*.domain.com"`

#### Mozilla Firefox

1. In the address bar, type **about:config** to display the list of current configuration options.
2. Acknowledge the warning.
3. In the **Filter** field, type **negotiate** to narrow the list of options.
4. Double-click the **network.negotiate-auth.trusted-uris** entry to display the **Enter string value** dialog box.
5. Enter the fully qualified domain name of the Good Control with the domain against which you want to authenticate. Do not enter the `https://` protocol portion of the URL. For example, `mygc.europe.bigCompany.com`
6. Repeat the above step for the **network.negotiate-auth.delegation-uris** entry, using the same domain.

#### Microsoft Internet Explorer

1. Click **Tools** > **Internet Options**.
2. Click the **Security** tab.
3. Click **Local Intranet** icon.

4. Click **Sites**.
5. Click **Advanced**.
6. In the **Add this website to the zone** field, enter the URLs of the fully qualified domain names of the Good Control servers. For example: `*.europe.bigCompany.com` or `https://myGc.europe.bigCompany.com:8443/`
7. Repeat the previous step for all GC servers in your GC cluster.
8. Click **Close** and follow the other prompts to save the change.
9. Click **Custom Level**, scroll to find **User Authentication** > **Logon**.
10. Make sure that **Automatic logon only in Intranet zone** is selected.
11. Click **OK** to save the change and return to the main Security page.
12. Click the Advanced tab.
13. Scroll to find **Security**.
14. Make sure that **Enable Integrated Windows Authentication** is selected.
15. Click **OK** to save the changes.
16. Restart Internet Explorer.

### Logging In As a Service or Non-personal Account

If you need to bypass the Kerberos SSO in an emergency (such as failure of your Kerberos service) or to login with a service account or other non-personal account, you can append the query string name/value pair **`?nosso=1`** highlighted below to the URL for accessing the GC console:

**`https://`***`yourGcServer.yourDomain.com`*`:`*`yourPort`***`/?nosso=1`**

This causes the GC system to prompt for username and password to login to the console.

## Installing Additional GC Servers in the Cluster

To deploy additional GC servers in your server cluster, you must first generate a license through your GC console. The installer requires this license in order to properly register the new GC server to the cluster.

If you are using your own enterprise-CA issued certificates, see the caution at the end of this topic.

### Steps

**To install an additional GC server in the cluster:**

1. Log into the console of a GC server already in the cluster.
2. Navigate to the **Server Configuration** > **Licenses** screen, and click **Generate License** to request a new license. GC then creates and displays a server license.

   This license can be used to install one additional GC server in the cluster.

3. Launch the GC installer on the target machine and install the new server. The installation of a cluster GC server follows the same procedure as the installation of a non-clustered GC server, with the following exceptions:
   - On the Administrator Information panel, enter the credentials for an account that is designated as a GC administrator. If the account is not already in the list of GC administrators, you can add it on the **Roles** >

**Administrators** screen of the GC console. For more information, see Understanding Administrator Rights and Creating and Configuring a Custom Role.

- On the Database Information panel, enter the information for the database used by the other GC servers in the cluster. The installer retrieves a list of GC servers from the database and displays them in a confirmation prompt. If you want to associate the new GC server with the servers listed in the prompt, confirm that you want to install the new GC into the cluster.
- Enter the license obtained in step 2 when prompted for a server license.

Repeat the steps above to install additional GC servers in the cluster.

For information on how to prioritize connections to different GC servers, see Assigning cluster priority to GC servers.

If you have installed SSL server certificates issued by your own enterprise Certificate Authority (CA) in Good Control or Good Proxy, you need to take care with adding more systems to the cluster and with upgrading to a new version of Good Control:

*A system being added to the cluster expects that at least one of the already installed GC or GP servers in the cluster is configured with the auto-installed certificate that is initially installed with the system. The same is true for upgrading a system to a new version of Good Control or Good Proxy.*

1. Before you add the new system or before you upgrade, on at least one server in the cluster, copy back the original **`server.xml`** and the original certificate store files. You should have a backup of the original files, which you made according to the procedure detailed in the Good Control Online Help topic "Installing SSL Certificates on GC and GP Servers". The exact paths to the files are documented in that topic. You must restart the service, as detailed in the Good Control Online Help topic "Starting the GC and GP Servers".
2. Add the new systems to the cluster or upgrade the system by following the documented procedure.
3. After the new servers are added to the cluster, or after the upgrade to a new version of Good Control or Good Proxy, undo what you did in step #1, *including the systems newly added to the cluster*: Copy back the **`server.xml`** and certificate store files that include your enterprise-CA-issued certificates.
4. After copying the files, you must restart the service, as detailed in the Good Control Online Help topic "Starting the GC and GP Servers".

## Uninstalling a GC or GP Server from the Cluster

You can remove a GC or GP server from your server cluster in two ways:

1. You can run the product uninstaller on the host machine, which removes the product from the file system and automatically sends the command to unregister the server from your database and the GD NOC.
2. You can manually unregister the server through the GC console with the intention of uninstalling it from the host machine's file system later.

If you need to reinstall a GC or GP server, follow the directions in either section before attempting to install the new server. This ensures that the records of the old server are removed both from your cluster database and from the GD NOC so you are able to install a new server on the same machine.

## Uninstalling a GC or GP server

To uninstall a GC or GP server, simply log into the server host machine as an administrator and run the product uninstaller from the Windows Start menu or the Control Panel. Alternatively, you can launch the installer for the version of GC or GP currently installed - or an installer for any later version of the product - and select the Uninstall option.

The uninstaller requires the credentials of a GC console administrator before proceeding. When asked for this information, you can supply the credentials of any Good Control Global Administrator, including the name of the Windows user account that runs the product's Windows service.

If the GC uninstaller detects that the GC server is the last remaining Primary GC server in the cluster, it prompts you for confirmation. It is highly recommended that, at all times, at least one GC server in the cluster is assigned the Primary priority. For information on how to determine the priority of a GC server, see Assigning cluster priority to GC servers.

Likewise, the GP uninstaller prompts you for confirmation if the GP server is the last remaining member of a server cluster defined as the Primary or Secondary GP cluster for a GC server, an application server, or any servers or domains listed on the **Server Configuration** > **Client Connections** screen. If you choose to continue with the uninstallation and no other GP servers are reachable, the GD clients of all users associated with the cluster are disconnected from all resources and can no longer receive policy updates. Make sure you understand how your organization's server clusters are designed for use before uninstalling any servers. For more information on GP server clusters, see Prioritizing GP server connections for GC servers and other domains and servers.

## Unregistering a GC or GP server through the GC console

This process removes records of the server from your database and from the GD NOC, but does not uninstall the server from its host machine; all files are left intact on the host machine until an administrator runs the product uninstaller. When records of a server are wiped in this manner, the server is rendered unable to connect to any other GD servers. This action is not reversible, so unregister a server only if you intend to uninstall it from its host machine.

To unregister a server:

1. Navigate to the **Server Configuration** > **Status and Diagnostics** screen. The console renders a server cluster diagram with your GC servers and all associated GP servers.
2. Find the server you want to unregister and click it to view a table of detailed server information.
3. Click the **Unregister Server** button, then confirm your action in the warning prompt, to unregister the server.

You cannot unregister the GC server you are currently logged into.

You are also prevented from unregistering the last remaining Primary GC server for the cluster. For information on how to determine the priority of a GC server, see Assigning cluster priority to GC servers.

Additionally, GC does not allow you to unregister the last remaining GP server in a server cluster defined as the Primary or Secondary GP cluster for a GC server, an application server, or any servers or domains listed on the **Server Configuration** > **Client Connections** screen. This ensures that clients are always able to connect to resources and receive policy updates.

However, if you do need to unregister the last GP server in a server cluster, first make sure that the GP server's cluster is not assigned as the Primary or Secondary cluster for any servers or domains, and then attempt to unregister the GP server. For more information, see Prioritizing GP server connections for GC servers and other domains and servers.

## Defining and Managing GP Server Clusters

Each GP server in your enterprise can belong to a GP server cluster. When your organization's first GC server was installed, the installer created a default cluster named "First". Each GP server you install automatically joins this cluster until your deployment has two defined GP clusters; after this condition is met, each new GP server you install is no longer assigned to any cluster by default.

Through the GC console, you can define a cluster for these unassigned GP servers to join, or you can change cluster membership of any GP server at any time. You can also create, modify, or delete GP clusters.

To view or modify GP clusters, first navigate to the **Server Configuration** > **Clusters** screen and make sure the **GP Clusters** tab is active. GC displays a diagram of GP clusters and the GP servers they contain. If any GP servers do not belong to a cluster, they are displayed at the top of the list in an area labeled Unassigned GP Servers.

> **Note:** If you modify any information on this screen, your changes are not saved automatically; you must click Update to commit your changes.

Here is a list of common tasks and how to accomplish them on this screen.

- To create a new cluster, click **Add New Cluster**.
- To update the name of a cluster, modify the text in the displayed field. Each cluster must have a unique name.
- To change the cluster membership of a GP server, drag it from its current cluster to another cluster.
- To delete a cluster, first find the cluster on the screen. Before the cluster can be deleted, you must move all of the cluster's GP servers to other clusters. Additionally, GC prevents you from deleting the cluster if it is currently assigned as the Primary or Secondary GP cluster for any GC server or application server. After the cluster is empty and is not associated with any GC or application servers, click its **Delete** button to mark it for deletion.
- Click **Revert** to undo your changes and reload the screen.
- Click **Update** to save your new configuration. The GC console displays an alert message if any issues are encountered.

Each GP server must belong to exactly one cluster; otherwise, it is considered unassigned.

## Assigning Cluster Priority to GC Servers

For fine grained control over your server cluster, you can designate each GC server to act as a **Primary**, **Secondary**, or **Tertiary** server for the cluster. GP servers attempt to connect to **Primary** GC servers first. If no **Primary** GC servers respond, GP servers then attempt to contact **Secondary**, then **Tertiary** GC servers.

Each new GC server you install is automatically given the **Primary** priority, but you can assign a new priority to any GC server at any time.

> **Note:** A server cluster must have at least one GC server with the **Primary** priority.

**To configure the priority order of GC servers in the cluster:**

1. Navigate to **Clusters**.
2. Click the **GC Clusters** tab to view a list of all GC servers in the cluster.
3. Find a server whose priority you want to change, and click the ✏ **Edit** icon for that server.
4. Hover over the underlined value listed for the server in the Priority column. The value becomes a pulldown list of options. Select a priority for the GC server.
5. Repeat steps 3 and 4 for any other GC servers whose priority you want to change.
6. Click **Update** to save your new configuration.

## Configuring GP Servers for Direct Connect

Some users and organizations that are physically distant from the GD Network Operations Center (NOC) servers might experience a large network round-trip time (RTT) between GD applications and the NOC, or between GP servers and the NOC. Because the establishment of a GD client connection involves multiple trips between the GD application and the NOC, and potentially between an organization's GP servers and the NOC, latency in connection establishment can be much larger. Additionally, TCP windowing can cause a large RTT and result in low overall throughput over an otherwise high bandwidth connection.

To mitigate these issues, with Direct Connect an organization's GD clients can establish direct connections to GP servers behind the internal firewall, completely bypassing the NOC servers. Assuming that an organization's GD clients are probably physically closer to its GP servers than to the NOC, Direct Connect can improve performance and reduce latency for the GD platform.

The following sections include information on how to configure your organization's environment for Direct Connect.

### Enterprise firewall and server configuration

Organizations can optionally use a web proxy server in the demilitarized zone (DMZ) to handle connections from GD clients to GP servers. If your organization chooses to set up this proxy server, you must ensure that the following conditions are met.

- The proxy server supports the HTTP Connect command and does not require authentication
- Port 17533 is open in your internal firewall in order for the proxy server to reach your GP servers

However, if your organization chooses to use Direct Connect without a proxy server in the DMZ, you must ensure that the following condition is met.

- Port 17533 is open in both internal and external firewalls in order for GD clients to reach your GP servers

### GP server configuration

To enable Direct Connect for one or more GP servers, navigate to the **Server Configuration** > **Settings** screen, and click the **Direct Connect** tab.

This tab lists your organization's GP servers, grouped by cluster. For more information on GP clusters, see Defining and managing GP server clusters.

You can configure Direct Connect for any GP server shown on this screen with the following steps.

1. Click the **Edit** icon for the GP server.
2. Change the value of the Direct Connect column cell to Yes.
3. The GC console automatically fills in the Host Name column field with the fully qualified hostname of the GP server. This field is configurable, so you can correct the value as needed. However, if you change to this value, the GP server generates a new certificate using the new value as the server's fully qualified hostname and then sends a request to GC's GD CA to sign the certificate. This certificate is for client connections, so the value you supply must be the correct fully qualified hostname for the GP server.
4. Optionally, if you have set up a web proxy server in your DMZ for the GP server to connect through, change the value of the Web Proxy column cell to Yes, and specify the fully qualified hostname and port of the proxy server in the Proxy Host and Proxy Port fields.

Click **Submit** to commit your changes after you are finished modifying information on this screen.

### Good Proxy TCP Session Keep-Alive

This is a property for Good Proxy, not Good Control. The property is located in the gps.properties file.

| Property | Description | Default, Global, Restart |
|---|---|---|
| gps.tcp.session.timeout | Set the length of time that a TCP connection can be inactive before it is closed.<br>**Important:** Do not alter this setting without direct consultation with BlackBerry. | Default: 1,800 seconds<br>Global: yes<br>Restart: yes |

## GP property reference

| Property | Description | Editable? |
|---|---|---|
| eacp.command.service.nslookup.srv.ldap | Enables LDAP over TCP for Active Directory servers. Active Directory servers offer the LDAP service over the TCP protocol; therefore, clients find an LDAP server by querying DNS for a record of the form: _ldap._ tcp. DnsDomainName<br>• true = indicates that GP uses LDAP for nslookup of a given service hostname<br>• false = GP uses reverse DNS lookup directly, using the given service hostname<br>Default: false | yes, editable |
| gc.admin.name | Username of Good Control administrator | not editable |

| Property | Description | Editable? |
|---|---|---|
| | Default: none | |
| gc.auth.token | Secret token to authenticate GC with GP<br>Default: none | not editable |
| gc.server.port | Port of GC server<br>Default: 443 | not editable |
| gc.server.uri | SOAP endpoint of GC server with which this GP should be registered.<br>Default: none, depends on name of server | not editable |
| gd.product.capability | GP server feature set used to compare with GC server feature set during GP registration to make sure that GC and GP are compatible.<br>Default: none. | not editable |
| gd.product.domain | Active Directory domain of the GP<br>Default: none. Set by installer. | not editable |
| gd.product.hostname | GP server name<br>Default: none. Set by installer. | not editable |
| gd.product.licensekey | GC and GP license keys as recorded in GDN<br>Default: none | not editable |
| gd.product.loginkey | GP server login credentials to BlackBerry Dynamics NOC for uploading GP server logs<br>Default: none | not editable |
| gd.product.serialnum | GC and GP serial numbers as recorded in GDN<br>Default: none | not editable |
| gd.product.type | Differentiate between GC service and GP service.<br>• GPS = Good Proxy<br>• GMC = Good Control<br>Default: for GP, GPS | not editable |
| gd.product.version | Version number of this GP<br>Default: none | not editable |
| gd.security.keystore.alias | Alias for the GP's keystore<br>Default: good-proxy | yes, editable |

| Property | Description | Editable? |
|---|---|---|
| gd.security.keystore.file | Location of GP keystore file<br>Default:*GP_installation_ directory* **\jre\security\lib\cacerts** | yes, editable |
| gd.security.rootcert.alias | Alias for the root certificate of the GP<br>Default: good-dynamics | yes, editable |
| gps.auth.token | Secret token to authenticate GP with GC | not editable |
| gps.directconnect.port | Port for Direct Connect configuration<br>Default: 17533 | not editable |
| gps.dns.server.ttl.ms | Time-to-live in milliseconds for the DNS server connections., i.e. time to wait for DNS server response.<br>Default: 1.8M milliseconds | yes, editable |
| gps.logfiles.days | Length of time to retain logfiles<br>Default: 10 days | yes, editable |
| gps.product.installdir | Installation directory for GP<br>Default: none. Set by installer | not editable |
| gps.product.registered | Flag for whether this GP has been registered with BlackBerry<br>Default: false | not editable |
| gps.server.fqdn | Fully qualified domain name for this GP server<br>Default: none. Set by installer | not editable |
| gps.server.name | Bare hostname of this GP server<br>Default: none. Set by installer | not editable |
| gps.server.port | Non-secured port for this GP server<br>Default: 17080 | not editable |
| gps.server.secure.port | Secure port for this GP server<br>Default: 17443 | not editable |
| gps.service.name | Name of the GP service on Windows<br>Default: GPS | yes, editable |
| gps.status.request.frequency | Allowable frequency for /status request on this GP | yes, editable |
| gps.tcp.session.timeout | Length of time that a TCP connection can be | yes, editable |

| Property | Description | Editable? |
|---|---|---|
| | inactive before it is closed.<br>**Important:** Do not alter this setting without direct consultation with BlackBerry.<br>Default: 1,800 seconds | |
| gps.unalias.hostname | For DNS lookups of app servers, use either IP address or hostname<br>• true= GP uses reverse DNS lookup with IP address of app server<br>• false = GP uses app server hostname for lookup<br>Default: false | yes, editable |
| gwy.push.connection.timeout | Timeout of persistent connection to MDC server in BlackBerry Dynamics NOC for push notifications<br>Default: 45 seconds | yes, editable |
| gwy.push.port | Port of MDC server in BlackBerry Dynamics NOC<br>Default: 443 | not editable |
| gwy.push.prot | Protocol for communications<br>Default: 1 | not editable |
| gwy.push.register | GP is registered with MDC server<br>true = GC is registered with BlackBerry Dynamics NOC<br>Default: true | not editable |
| gwy.push.request.timeout | Timeout of request to MDC server BlackBerry Dynamics NOC<br>Default: 20 seconds | yes, editable |
| gwy.push.secure | Use SSL for connection to MDC serverBlackBerry Dynamics NOC<br>Default: false | not editable |
| gwy.push.server | Name of MDC server in BlackBerry Dynamics NOC<br>Default: gdmdc.good.com | not editable |
| gwy.push.socket.timeout | Timeout in establishing socket connection to MDC server BlackBerry Dynamics NOC | yes, editable |

| Property | Description | Editable? |
|---|---|---|
| | Default: 45 seconds | |
| health.check.enabled | Whether to perform additional checks on GP health<br>Default: true | yes, editable |
| health.check.interval | How often to check GP health<br>Default: 3.6M milliseconds (1 hour) | yes, editable |
| log.upload.date.name.format | Date format for timestamp of GP logfile names<br>Default: yyyy-MM-dd | not editable |
| log.upload.dir | Path to directory on server where logs are stored<br>Default: none. Set by installer. | not editable |
| log.upload.url | URL on this GP where logfiles can be uploaded<br>Default: none | not editable |
| mdc.server.name | Name of MDC server in BlackBerry Dynamics NOC<br>Default: gdmdc.good.com | not editable |
| mdc.server.port | Port of MDC server in BlackBerry Dynamics NOC<br>Default: 443 | not editable |
| proxy.auth.domain | Active Directory domain for authentication login to external Web proxy server<br>Default: none | yes, editable |
| proxy.auth.password | Password of username for authenticating to external Web proxy server<br>Default: none | yes, editable |
| proxy.auth.username | User name for connecting to external Web proxy server<br>Default: none | yes, editable |
| proxy.https.host | Name of external Web proxy server<br>Default: none | yes, editable |
| proxy.https.port | Port number for HTTPS connection to external Web proxy server<br>Default: none | yes, editable |
| proxy.urls | URLs that must be proxied | yes, editable |

| Property | Description | Editable? |
|---|---|---|
| | Default: none | |
| proxy.use | Use an external Web proxy server<br>Default: false | yes, editable |
| relay.gps.key | Key to access relay server in BlackBerry Dynamics NOC<br>Default: none | not editable |
| relay.server.name | Name of relay server in BlackBerry Dynamics NOC<br>Default: gdrelay.good.com | not editable |
| relay.server.port | Port number of relay server in BlackBerry Dynamics NOC<br>Default: 443 | not editable |

## Logging property reference

| Property | Description | Good Control | Good Proxy |
|---|---|---|---|
| Maximum server log file size | Allowable values: from 100 KB to 1 GB | Default: 256 MB | Default: 256 MB |
| Maximum server log file age | In days | Default: 10 days | Default: 10 days |
| Compress server log files | Allowable values: true\|false | Default: on | Default: on |
| Server logging level | Allowable values: Info\|Debug | Default: Info | Default: Info |

# Certificates

With the **Certificates** screen, you define the GD certificate store and create the definitions for on-demand retrieval of client certificate by the GD Runtime.

**Note:** These screens do not relate to Good Control's own SSL/TLS certificates created at installation. For information about changing those certificates, see Installing SSL Certificates on GC and GP Servers.

## Trusted Authorities Tab

On the **Certificates** > **Trusted Authorities** tab, you add the trusted certificate authorities in the GD certificate store for client applications to communicate with application servers.

You upload Certificate Authorities (also known as "root certificates") that sign the SSL/TLS certificates used by your application servers.

**To upload a certificate:**

1. Obtain all required certificates from either a well-known third-party, trusted Certificate Authority (CA) or from your own enterprise CA. Certificates must be X.509 in the DER encoding format.
2. In the system, navigate to **Certificates** > **Trusted Authorities**.
3. Click **Upload New Certificate**.
4. Locate the certificate file on your local machine.
5. Complete the upload.

   The results of the upload are displayed.

6. Repeat these steps for all required CAs.

## App Usage Tab

On the **Certificates** > **App Usage** tab, you can specify which GD-based apps are allowed to have client certificates (for use such as S/MIME or user authentication).

**To specify GD-based apps for certificate synchronization, in Good Control:**

1. Navigate to **Certificate Management** >**App Usage**.
2. Click **Add App**.
3. In the displayed dialog box, find the Good-based app that will synchronize, and checkmark its name.
4. Click **OK** to save your addition, or click the upper right **X** in the dialog box to discard it.

## Certificate Definitions Tab

On the **Certificate Management** > **Certificate Definitions** tab, you can configure connections to a *PKI Connector* system so that your users or your client applications can obtain client certificates.

BlackBerry has specified a protocol for exchange of certificate-related information between an enterprise CA and Good Control. See https://community.good.com/dGood Dynamics User Certificate Management Protocol. In addition, BlackBerry also offers a reference implementation in Java to programatically obtain PKCS 12 certificates from a Certificate Authority. See PKI Cert Creation via Good Control: Reference Implementation for information.

## Fields for Certificate Definitions

The following fields are included in a certificate definition.

| Field | Description |
|---|---|
| Name | A mnemonic name of your own devising for this CA |
| Server Address | A URL including protocol (http:// or https://), IP address or FQDN of the CA server, port, and program that furnishes certificates to the GC. Example: `https://caserver.enterprise.com:9090/create`<br>**Note:** Be sure that your GCs can reach this server and port. Use the **Test Connection** button to verify. This button attempts to make a connection to the server address you have specified. |
| Authenticate with username and password | For the GC to connect to the CA server, specify username and password required by the CA server.<br>Mutually exclusive with next setting. |
| Authenticate with client certificate | Default. Mutually exclusive with above setting. For the GC to connect to the CA server:<br>1. Click **Upload** to upload a PKCS 12 certificate with .pfx or .p12 filename extension.<br>2. Specify the password to that file.<br>**Note:** GC cannot validate the password until it attempts to decrypt the file after you click **OK**. If the password you entered is invalid, an error message is displayed next to the affected certificate in the entire list. You must re-edit the definition to enter the correct password. |
| Use following to trust SSL connection from Good Control to PKI connector | How have you confd the connection to your PKI connector?<br>- Default Public CAs<br>- CA certificate, which you must upload<br>- Server SSL certificate, which you must upload |
| Require user-entered password or OTP | Refers to behavior on the client end-user devices: the password or OTP as needed by the PKI connector defined in the **Server Address** field above. |
| Enable certificate renewal *XX* before expiration | Use the pulldown menu to set the number of days prior to expiration for the renewal to occur. Values are as follows:<br>- 7<br>- 14 |

| Field | Description |
|---|---|
| | - 30 (default)<br>- 60<br>- 90<br>- 120<br>- 180 |
| Delete certificate on expiry | Mutually exclusive with above setting |
| Remove duplicate certificate (Certificate that expires first will be removed) | Self-explanatory |

## Adding a Certificate Definition

**To define a connection to an internal or external certificate authority, in Good Control:**

1. Navigate to **Certificate Management** > **Certificate Definitions**.
2. Click **Add Definition**.
3. Complete the fields described in Fields for Certificate Definitions .
4. Click **OK** to configure the CA details, or **Cancel** to allow them.

## New: changes to Certificate Definitions tab

Good Control's **Certificates** > **Certification Definitions** tab has the following changes:

- The **Test Connection** button does *not* save the definition to Good Control's database, as it did in the past.
- To save the definition to Good Control's database, you must click **Save**.
- The list of defined certificates now displays characteristics of the definitions, such as Require user-entered password or OTP.

## Required: update your PKI Connector to support certificate renewal

The reference implementation as delivered does not include the logic necessary to work with the certificate renewal feature.

your PKI Connector must include a function to return values that indicate the capabilities of your connector. Those capabilities are as follows:

- getP12: New cert enrollment only
- getP12, renewCert: Bother new cert enrollment and certificate renewal

The necessary design aspects of certificate renewal are detailed in BlackBerry's User Certificate Management Protocol.

After you modify your PKI Connector and deploy it, you need to inform Good Control that the connector has new capabilities.

The latest version of Good Control includes an **Update connector capabilities** button (under **Certificates** tab) whereby you inform Good Control of your PKI connector's capabilities. The server makes a request to your connector to discover the capabilities based on the values you return.

## Info: PKI Connector notified when certificates are removed if connector supports removal capability

Good Control supports a PKI Connector that allows you to interact with a Certificate Authority server. A reference implementation in Java for a PKI Connector is described at *PKI Cert Creation via Good Control: Reference Implementation*.

The PKI Connector is now notified whenever a certificate has been removed from the GC.

For certificate removal, the PKI connector must be configured to support certificate removal, and the connector details must be updated in GC. Complete details on developing a connector and configuring GC to use it are in *PKI Cert Creation via Good Control: Reference Implementation*.

## New: changes to Certificate Definitions tab

Good Control's **Certificates** > **Certification Definitions** tab has the following changes:

- The **Test Connection** button does *not* save the definition to Good Control's database, as it did in the past.
- To save the definition to Good Control's database, you must click **Save**.
- The list of defined certificates now displays characteristics of the definitions, such as Require user-entered password or OTP.

## New: automatic renewal or deletion of CA-fetched PKI certificates

If you have implemented the PKI certificate "fetching" feature described in *PKI Cert Creation via Good Control: Reference Implementation* at https://community.good.com/docs/DOC-7151, in Good Control:

- You can specify the automatic renewal of these certificates.
- You can cause them to be deleted when they expire.
- You can automatically remove duplicate certificates

**To specify automatic renewal of certificates in Good Control:**

1. Navigate to **Certificates** > **Certificate Definitions** tab.
2. Find the desired certificate definition.
3. Click **Edit**.
4. Find **Enable certificate renewal** *XX* **before expiration**
5. Click the checkbox.
6. Use the pulldown menu to set the number of days prior to expiration for the renewal to occur. Values are as follows:
   - 7
   - 14
   - 30 (default)
   - 60
   - 90
   - 120
   - 180
7. Click **Save** to save the changes or **Cancel** to discard them.

**To specify automatic deletion of expired of certificates in Good Control:**

1. Navigate to **Certificates** > **Certificate Definitions** tab.
2. Find the desired certificate definition.
3. Click **Edit**.
4. Find  **Delete certificate upon expiry**
5. Click the checkbox.
6. Click **Save** to save the changes or **Cancel** to discard them.

**To specify automatic remove duplicate certificates in Good Control:**

1. Navigate to **Certificates** > **Certificate Definitions** tab.
2. Find the desired certificate definition.
3. Click **Edit**.
4. Find  **Remove duplicate certificate (Certificate that expires first will be removed)**
5. Click the checkbox.
6. Click **Save** to save the changes or **Cancel** to discard them.

## Adminstrator-initiated PKI cert renewal for client apps

In addition to automated cert renewal, the administrator can force certificate renewal for individual users via the Good Control console.

**Note:** Forced cert renewal operates only with client apps built with the latest BlackBerry Dynamics SDK. Apps built with earlier release cannot be forced. Good Control does not display an error message in this case.

After the older apps have been upgraded, the administrator can then force the renewal.

**Steps to force cert renewal in Good Control:**

1. In the left nav, click **Users and Groups**.
2. Find the affected user name and click the name.
3. Click **Certificates**.
4. To initiate the cert renewal, click the circular arrows at the far right of the certificate.

# PKCS 12 Certificate Management

Good Control supports the use of public/private key (PKCS 12) certificates for signing email and for client authentication.

With the Self Service Portal, end-users supply their own password-protected certificate files to Good Control. There is no limit on the number of certificates per user. When the end-user activates an application, all certificates on file with GC are sent to that application's container. Certificates are sent only one time. If the end-user deletes a certificate, GC removes that certificate from the affected containers. If the end-user adds more certificates, they also are sent to the application containers.

In the client applications, the end-user must enter the password for the certificates that were uploaded; with that password, GC decrypts the certificates for use and can then display characteristics of the certificate in the GC user interface.

Setting up certificates for these needs includes the following general parts:

- Certificate requirements
- Enabling the Good Control security policy for PKCS 12 certificates
- Whitelisting the applications allowed to use the PKCS 12 certificates
- End-users uploading their certificates

## Certificate requirements and troubleshooting

Make sure your certificates conform to these requirements:

- Certificates must be in PKCS 12 format: Certificate Authority (CA), public key, and private key, all in the same file.
- The PKCS12 file must end with the extension **`.p12`** or **`.pfx`**.
- The PKCS 12 file must be password-protected.
- The minimum keylength for the certificates must be 2,048 bytes.

There are many sources of certificates:

- Your own internal certification authority (CA)
- A well-known public CA

- Tools from the Internet, such as OpenSSL's **`keytool`** command. For example, the following is sufficient to generate a PKCS 12 certificate that is usable with Good Control; subsitute your own vlues for alias the keystore name and the keystore password. If in doubt consult information on the Internet about all the possible options on the keytool command:

  ```
  keytool -genkeypair -alias good123 -keystore good123.pfx -storepass good123 -
  validity 365 -keyalg RSA -keysize 2048 -storetype pkcs12
  ```

### Beware of weak ciphers from export

Personal Information Exchange files are encrypted, and therefore must be encrypted with FIPS-strength ciphers if to be used when FIPS is enabled on the employee’s security policy.

**Note:** For their own maximum interoperability with other systems, it is common for third-party applications, for example the Mac OSX keychain, to export identity material (credentials) using weak ciphers.

The administrator or employee can use a tool such as the OpenSSL command line to re-encrypt the file with a FIPS-strength cipher like so, which re-encryts with the AES-128-CBC cipher:

```
openssl pkcs12 -in weak.p12 -nodes -out decrypted.pem
<enter password>
openssl pkcs12 -export -in decrypted.pem -keypbe AES-128-CBC -certpbe AES-128-CBC -out strong.p12
<enter password>
rm decrypted.pem
```

## Setting Certificate Expiry Time

By default PKCS 12 certificates uploaded to the GC must be used within a time period you can define before the GC deletes them for security. The default is 24 hours.

To change the default, edit the GC server property gc.user.keystore.ttl.seconds. See the steps and details for all properties in GC Server Property Reference.

## Allowing Client Certificates

This policy enables client certificates, for uses such as S/MIME or user authentication. It allows:

- Uploading of client certificates to Good Control
- Retrieval of user certificates by Good Control when necessary

By default, the security policy that allows the use of certificates is disabled (false).

If this policy is disabled, then the **Certificates** tab is hidden from the end-user's view of the User Self Service portal but not from the GC administrator's view, who can still add, update, and delete certificates even if the security policy is disabled for a particular user.

**To allow client certificates:**

1. Navigate to **Policy Sets**.
2. Edit the desired policy set.
3. Click the **Security Policies** tab.
4. Scroll down to find the heading **Certificate Management**.
5. Check **Allow use of client certificates**.
6. In the upper right, click **Update** to save your changes.

In addition to setting this policy, you might need to create certificate definitions on the Certificate Definitions Tab and set applications on the App Usage Tab .

## Important: Whitelisting Applications Allowed to Use the PKCS 12 Certificates

By default, Good Work and BlackBerry Access applications are already whitelisted for use of PKCS 12 certificates.

> **Important:** Any applications you want to allow must be added to Good Control's **Certificates** > **App Usage** tab. Otherwise, these apps cannot use PKCS 12 certificates and they cannot be activated.

**To add or remove an application for PKCS 12 use, in Good Control:**

1. Navigate to **Certificates** > **App Usage** tab.
2. To add an application, click **Add App**.
3. In the displayed dialog, find the application you want to add, checkmark it, and click **OK**.
4. To remove an application, scroll in the list to find the application to remove.
5. Click the circled **X** on the right of the application name.
6. Click **OK** to remove the application or **Cancel** to retain it.

## Uploading PKCS 12 Certificates for End-users

**To upload a PKCS12-format certificate file with either .p12 or .pfx file extension on behalf of an end-user, in Good Control:**

1. Navigate to **Users and Groups** > *checkmark a user to edit* > **User Actions** menu selection **Edit User**.
2. Click the **Certificates** tab.
3. Click **Upload**.
4. Navigate your computer to find the PKCS 12-format file with either .p12 or .pfx filename extension.
5. Select or open the file.
6. Follow the leading prompts to finish the upload.

GC then displays the date of the upload. GC cannot display more information about the certificate until the end-user uses the certificate at least once by entering the password to the certificate file. Until that password is supplied, the certificate is encrypted and details cannot be obtained from it.

## Deleting Certificates for End-users

1. Navigate to **Users and Groups** > *checkmark a user to edit* > **User Actions** menu selection **Edit User**.
2. Click the **Certificates** tab.
3. Checkmark the certificat you want to delete.
4. Click **Delete**.

## Lifecycle and states of a PKCS 12 certificate

In Good Control, a PKCS 12 certificate can have any of the following states.

| State | Description |
|---|---|
| Uploaded | Certificate has been stored on the GC |
| Delivered | When the certificate has been sent to a GD application container |
| • Verified<br>• Expired<br>• Failed | After a GD application container has used the certificate.<br>**Note:** Hover your cursor over the "Failed" state to see the reason for the failure. |

# Info: support for Kerberos PKINIT: user authentication via PKI Certificates

The following platforms of the BlackBerry Dynamics SDK support Kerberos PKINIT for user authentication via PKI certificate:

- Windows (UWP)

The remainder of this discussion is for the administrator who configures Kerberos PKINIT.

No extensive programming is required use Kerberos PKINIT. For considerations on application programming, see Client applications .

## For the admin: distinction from KCD and behavior of Kerberos PKINIT

Kerberos terminology is notoriously obscure and confusing. For example, do not confuse KDC (Key Distribution Center) with KCD (Kerberos Constrained Delegation). See Short list of acronyms for many of the common terms.

**Important:** Kerberos PKINIT is completely distinct from Kerberos Constrained Delegation (KCD).

| Kerberos PKINIT | Kerberos Constrained Delegation |
|---|---|
| Kerberos PKINIT authentication is between the BlackBerry Dynamics-enabled client | **Note:** For PKINIT, Kerberos Constrained Delegation must *not* be enabled. |

| Kerberos PKINIT | Kerberos Constrained Delegation |
|---|---|
| application and the Windows Key Distribution Center (KDC), which communicate directly, and user authentication is based on certificates issued by Active Directory Certificate Services. | If Kerberos Constrained Delegation has been configured, a BlackBerry Dynamics-based application does *not* use Kerberos PKINIT to access the defined KCD realms. Instead, when Kerberos Constrained Delegation is in effect, a trust relation has been previously established between the GC and the Key Distribution Center, and the GC communicates with the service on behalf of the client application.<br>Kerberos Constrained Delegation takes precedence over Kerberos PKINIT, even if the user has a valid certificate. |

## Background on PKINIT and FAQ

Consider the interactions in this drawing: http://www.ibm.com/developerworks/ibmi/library/i-sso/figure1.jpg

Kerberos PKINIT authentication requires the client (In the drawing, the human John, running a BlackBerry Dynamics-enabled application) to be able to contact:

- When initializing the user session, the user's Key Distribution Center (KDC) Authentication Service (AS) to obtain a Ticket-Granting Ticket (TGT),
- When establishing a connection to a resource (in the drawing, Service "A"), the resource’s KDC Ticket-Granting Service (TGS).

In a large organization users and resources might belong to various realms and there may be many KDCs, so how does BlackBerry Dynamics find the right one?

1. How does the client locate the user’s KDC Authentication Service when initializing the user’s session?

- Password-based authentication

  The realm in the user name must contain the host name of the KDC AS. For example:

  User: user@MY.REALM.COM

  Password: myPassword

- Certificate-based authentication: This is PKINIT.

  The realm in the UPN of the user’s certificate must contain the host name of the KDC AS. For example:

  UPN (OID 1.3.6.1.4.1.311.20.2.3): user@MY.REALM.COM

2. How does the client locate the resource's KDC Ticket-Granting Service (TGS) when retrieving the resource?

BlackBerry Dynamics attempts to obtain a TGS from the host in the domain of the resources URL.

> For example,
>
> URL: http://resource.myrealm.com/index.html
>
> The client will connect to KDC TGS running on host myrealm.com on TCP port 88.

The following are key points to note when integrating BlackBerry Dynamics and Kerberos infrastructure:

- The KDC host must be in the **Allowed Domains** of the Connectivity Profile applied to the affected users' policy sets in Good Control.
- The KCD host must be listening on TCP port 88 (Kerberos default port).
- BlackBerry Dynamics does *not* support KDC over UDP.
- BlackBerry Dynamics does not use Domain Name System (DNS) records such as **`SRV`**, **`CNAME`**, or **`TXT`** to locate the correct KDC. That is, the KDC must have an **`A`** record (IPv4) or **AAAA** record (IPv6) in your DNS.
- BlackBerry Dynamics does *not* use Kerberos configuration files (such as krb5.conf) to locate the correct KDC.
- The KDC can refer the client to another KDC host. BlackBerry Dynamics will follow the referral, as long as the referred-to KDC host is reachable by BlackBerry Dynamics: defined in the the **Allowed Domains** of the Connectivity Profile applied to the affected users' policy sets in Good Control.
- The KDC can obtain the TGT transparently to BlackBerry Dynamics from another KDC host.

## Response on failure

If a valid certificate is not found or if Kerberos PKINIT authentication does not succeed for some reason, the response **`401 Authorization Required`** is returned.

Depending on the client application implementation, the user might be prompted for Kerberos password-based domain credentials.

## Required configurations for PKINIT

Organizations that want to take advantage of Kerberos PKINIT for BlackBerry Dynamics-based applications need to adhere to the following requirements.

### Servers

- Kerberos Constrained Delegation must *not* be enabled.
- Windows Key Distribution Center (KDC) services for KDC server certificates issued by a Microsoft Certificate Authority (CA) via the Active Directory Certificate Services must come only from the following Windows Server versions. No other server versions are supported.
  - Internet Information Server with Windows Server 2008 R2
  - Internet Information Server with Windows Server 2012 R2
- In Good Control:
  - The KDC hosts must be in the **Allowed Domains** of the Connectivity Profile applied to the affected users' policy sets.
  - Valid KDC service certificates must be located either in the **BlackBerry Dynamics Certificate Store** or the **Device Certificate Store**, as described for the **Trusted Certificates** security policy described in "Certificate Management Policies" in the Good Control Online Help.
  - Valid client certificates must be located *only* in the **BlackBerry Dynamics Certificate Store**, as described for the **Trusted Certificates** security policy described in "Certificate Management Policies" in the Good Control Online Help.

- Client certificates need to be enabled and uploaded to Good Control, just as for certificates for S/MIME. See the Good Control Online Help topic "PKCS 12 Certificate Management for Email and Client Authentication".

### Client certificates

- Client certificates must include the User Principal Name (UPN, such as user@domain.com) in the Subject Alternative Name (SAN) of object ID (OID) **`szOID_NT_PRINCIPAL_NAME`** 1.3.6.1.4.1.311.20.2.3, as specified by Microsoft at https://support.microsoft.com/en-us/kb/287547.
- The domain of the UPN must match the name of the realm of the Windows Key Distribution Center (KDC) service.
- The Extended Key Usage (EKU) property of the certificate must be Microsoft Smart Card logon (1.3.6.1.4.1.311.20.2.2), as specified by Microsoft at https://technet.microsoft.com/en-us/library/ff404293(v=ws.10).aspx.
- Certificates must be valid. Validate them against the servers listed in Servers .

### Client applications

- Applications must *not* send any password in the HTTP/HTTPS request.
- Applications must either set the HTTP/HTTPS header **`WWW-Authenticate: Negotiate`** or *not* set any authorization method in the HTTP or HTTPS request, to which the server has responded with **`401 WWW-Authenticate: Negotiate`**, as detailed in https://www.ietf.org/rfc/rfc4559.txt.

## Short list of acronyms

| Acronym | Expansion |
|---|---|
| AS | Authentication Service of a KDC. |
| KCD | Kerberos Constrained Delegation, which must *not* be enabled for Kerberos PKINIT. |
| KDC | Key Distribution Center |
| PKINIT | Public Key Infrastructure Initialization |
| TGS | Ticket Granting Service |
| TGT | Ticket-Granting Ticket, that is a ticket that allows A TGS to give you more tickets. |

# Info: client certificate sharing among BlackBerry Dynamics-based applications and on-Premise Good Control

In conjunction with on-premise Good Control (not with Cloud GC), the BlackBerry Dynamics SDK supports the "sharing" of a single client certificate among all BlackBerry Dynamics-based applications for an end-user. That is, if authentication via client certificates is enabled in Good Control and one or more client certificates have been uploaded to Good Control, those certificates are used for user authentication by all BlackBerry Dynamics-based applications on the user's device.

## Requirements

- There is no setting in Good Control for this feature. It is permanently enabled.
- Client certificates must be enabled in Good Control and at least one PKCS 12 certificate for a user must be uploaded to Good Control. See the Good Control Online Help topic "PKCS 12 Certificate Management for Email and Client Authentication".
- You need to set the discovery scheme **`gd-sc3.certificate.sharing`**, as described in New discovery scheme: gd-sc3.certificate.sharing
- No programming is required.

## Behavior of apps

Certificate sharing among BlackBerry Dynamics-based applications simplifies the set up by the end user, who does not need to manage certificates for each individual application. However, during application activation, end users might notice additional interaction among applications (so-called "flips" between apps) because an application being activated must retrieve a certificate from an application that already has it.

# New: automatic renewal or deletion of CA-fetched PKI certificates

If you have implemented the PKI certificate "fetching" feature described in *PKI Cert Creation via Good Control: Reference Implementation* at https://community.good.com/docs/DOC-7151, in Good Control:

- You can specify the automatic renewal of these certificates.
- You can cause them to be deleted when they expire.
- You can automatically remove duplicate certificates

**To specify automatic renewal of certificates in Good Control:**

1. Navigate to **Certificates** > **Certificate Definitions** tab.
2. Find the desired certificate definition.
3. Click **Edit**.
4. Find **Enable certificate renewal** *XX* **before expiration**
5. Click the checkbox.
6. Use the pulldown menu to set the number of days prior to expiration for the renewal to occur. Values are as follows:
   - 7
   - 14
   - 30 (default)
   - 60
   - 90
   - 120
   - 180

7. Click **Save** to save the changes or **Cancel** to discard them.

**To specify automatic deletion of expired of certificates in Good Control:**

1. Navigate to **Certificates** > **Certificate Definitions** tab.
2. Find the desired certificate definition.
3. Click **Edit**.
4. Find **Delete certificate upon expiry**
5. Click the checkbox.
6. Click **Save** to save the changes or **Cancel** to discard them.

**To specify automatic remove duplicate certificates in Good Control:**

1. Navigate to **Certificates** > **Certificate Definitions** tab.
2. Find the desired certificate definition.
3. Click **Edit**.
4. Find **Remove duplicate certificate (Certificate that expires first will be removed)**
5. Click the checkbox.
6. Click **Save** to save the changes or **Cancel** to discard them.

## Adminstrator-initiated PKI cert renewal for client apps

In addition to automated cert renewal, the administrator can force certificate renewal for individual users via the Good Control console.

**Note:** Forced cert renewal operates only with client apps built with the latest BlackBerry Dynamics SDK. Apps built with earlier release cannot be forced. Good Control does not display an error message in this case.

After the older apps have been upgraded, the administrator can then force the renewal.

**Steps to force cert renewal in Good Control:**

1. In the left nav, click **Users and Groups**.
2. Find the affected user name and click the name.
3. Click **Certificates**.
4. To initiate the cert renewal, click the circular arrows at the far right of the certificate.

# Licensing

The licenses you need for a BlackBerry Dynamics deployment:

- For a new deployment, obtain the license from the BlackBerry Developer Network (BDN) for the first Good Control server

- For additional servers in the same cluster, use the same license as you first obtained.

## First License from BDN

The BlackBerry Developer Network (BDN) portal is the place for you to obtain your first GD server cluster license. Log into the BDN portal and navigate to either the Developer Center or the IT Admin Center, or the screen for your Organization Group, and click **Create Good Control Server License** to manage your licenses.

The BDN has two types of licenses: Production (PROD) and Development (DEV). BlackBerry Dynamics servers for developers need Development licenses, and when you deploy your GC and GP servers into production, use a Production license. Your BDN Organization Group role determines the types of licenses you can manage. Group members can only add and remove Development server licenses, while Group administrators can add Production server licenses and delete both Development and Production licenses.

**Note:** Before you can generate a production license, you must be an administrator of your Enterprise group on the BlackBerry Developer Network (BDN). Contact the members of your Enterprise group who can grant you administrator rights on BDN or if necessary, contact BlackBerry Dynamics technical support for assistance.

To create a license, in your BDN account, click **Add New License**, then enter a friendly name for your new GC server and click **Add License**.

To delete a license, click the delete icon for the license you wish to delete.

**Important:** Do not delete the server license for any GD servers currently in use. When you delete a GD license, the associated GC and GP servers are rendered inoperative. Furthermore, all users managed by that GC server lose their GD application data, and the GD applications themselves are irrevocably wiped from those users' devices.

When you are logged into the console of a GC server, it visually indicates the type of license it uses.

If the GC uses a Development license, the **development** designation is displayed in red text next to the friendly server name.

If the GC uses a Production license, no additional text is displayed next to the server name.

## Second and Subsequent Licenses

You can use Good Control itself to generate additional licenses for the other servers you add to your BlackBerry Dynamics deployment.

To generate additional licenses for new Good Control servers in the cluster:

1. Navigate to **Licenses**.
2. Click **Generate license**.
3. Copy the generated license so you can enter it into the Good Control installer for the new server.

# Export data and reporting

## Enhancements to GP diagnostics page

Good Control's diagnostic page for its associated Good Proxy servers has been enhanced to display a color-coded status for the server and the following details.

If any of these metrics indicates a problem, the color of the status is yellow. If all metrics indicate problems, the color of the status is red.

| YELLOW<br>if all three of these indicators is failing... | RED status<br>if any of the following is failing... |
|---|---|
| • GC Connectivity | • NOC Last Connected Time |
| • Memory usage | • Active sessions count |
| • CPU usage | • or if all indicators listed here are failing |

## Good Control health report

Good Control writes a report about its health to the file **`c:\good\gc_health_report.data`** based on a frequency you can control via the system property **Frequency in seconds that job to generate GC health report will run**. Default is every 86400 seconds, or 24 hours.

For every update, the file is overwritten, not appended.

Below is a sample of the data available in the report, which consists of name/value pairs in JSON format. The value of the status **`code`** can be one of OK, WARN, ERROR, INFO, UNKNOWN.

```
"lastReportedTime":1474319313201,
"healthy":true,
"status":
{
        "statuses":
        {
                "MemoryMonitor":
                {
                        "code":"OK",
                        "desc":"769.64MB free,
                        966.00MB total."
                },
                "DBMonitor":
                {
                        "code":"OK",
                        "desc":"Database connection is ok"
                },
                "CPUMonitor":
                {
                        "code":"OK",
                        "desc":"CPU Load: 0.00"
                },
                "HttpConnectivityMonitor":
                {
                        "code":"OK",
                        "desc":"Http connectivity succeeded to URL <URL for GD NOC>
                },
                "DBJobQueueMonitor":
                {
                        "code":"INFO",
                        "desc":"JobQ Size: 31,
                        Average JobQ Size: 31,
                        Max JobQ Size: 31"
                }
        },
"code":"OK"
}
```

## /status URLs display status of Good Control and Good Proxy

You can navigate to the URL **https://***fully_qualified_domain_name_of_good_control_host***/gc/status** to see the general status of the server.

The response looks similar to the following:

```
{"paused":false,"name":"BlackBerry UEM - Good Control Service","ha":{"scheme":"active-
standby","state":"active"},"health":
{"score":100},"serviceID":"GoodControl","version":"2.4.55.8997","connections":
[{"connected":true,"type":"MDC","dest":"https://someserver.company.com/GNP1.0"},
{"connected":true,"type":"PUSHGW","dest":"https://someserver.company.com:443/GDES1.0"},
{"connected":true,"type":"DB","dest":"jdbc:sqlserver://someserver.company.com:1433;databaseName=anu1;sendString
ParametersAsUnicode=false","properties":[{"name":"dialect","value":"com.good.db.util.SQLAddNVarCharDialect"},
{"name":"driver","value":"com.microsoft.sqlserver.jdbc.SQLServerDriver"}]}]}
```

This URL is not access-protected but it is governed by the property **Allowed frequency of /status**, which permits access only within a certain frequency. See New or changed properties for more details.

### Good Proxy /status URL

For GP, the URL is **https://***fully_qualified_domain_name_of_good_proxy_host_and_port***/status** , which responds similarly to the following:

```
{"paused":false,"name":"BlackBerry UEM - Good Proxy Server","ha":{"scheme":"active-
standby","state":"active"},"health":
{"score":100},"serviceID":"GoodProxy","version":"2.4.55.6204","connections":
[{"connected":true,"type":"GC","dest":"someserver.company.com","properties":
[{"name":"lastConnectedTimeStamp","value":"2016-09-30T17:28:09.997-04:00"}]},
{"connected":true,"type":"session","properties":[{"name":"maxSession","value":"15000"},
{"name":"activeSession","value":"0"},{"name":"totalSession","value":"0"},{"name":"idleSession","value":"0"},
{"name":"noOfdirectConnectConnections","value":"0"},{"name":"noOfrelayConnections","value":"0"},
{"name":"dest","value":"someserver.company.com"},{"name":"port","value":"443"}]},
{"connected":true,"type":"MDC","dest":"someserver.company.com","properties":
[{"name":"lastConnectedTimeStamp","value":"2016-09-30T17:27:33.700-04:00"},{"name":"port","value":"443"}]}]}
```

## Progress indicator for cluster-wide logfile upload

The status of log file uploading, shown on Good Control's **Upload Server Logs** page, now shows the status of log uploads for all the GC servers in the entire cluster. This same status is also viewable on any server in the cluster.

## Exporting or Purging Audit Trail Logs

GC keeps several different types of logs. Server and console log files, located at C:\good\gclogs on the server's host machine, contain a large amount of information pertaining to server processes, requests, and responses. They are useful sources of information if you need to troubleshoot a GC console or server issue. However, information on specific events, such as the name of the administrator who issued a request to wipe a particular container or to unregister a GP server, can be difficult to locate in these logs. For you to easily find information on a specific event, GC servers store audit trail records in the cluster database.

Audit trail log records indicate the administrator or process that initiated a particular action and include the response from the server after the action is completed. GC stores requests and responses for actions such as:

- Logging into a GC console
- Modifying GC and GP cluster configurations
- Creating, modifying, or deleting a policy set
- Adding, modifying, or deleting a user
- Wiping or locking a container
- Generating or deleting an access key
- Assigning a new policy set to a user

Because all GC servers in your cluster use the same database, the combined audit trail logs of all your GC servers are stored in the cluster database. From the console for any GC server, you can export and download a copy of the cluster's audit logs or delete old audit records.

### Downloading audit trail records

With the GC console you can export audit trail logs for a specified date and time range from the database to a comma-separated values (CSV) file. The file's data is easily sortable in spreadsheet software, because each record in the CSV file identifies who requested the action, when the request was sent, and which GC server performed the action.

When you export these logs, the audit trail records in the database are kept intact.

To export the combined audit logs for all GC servers in your cluster as a CSV file:

1. Navigate to the **Reporting** > **Audit Trail Logs** screen.
2. Configure the start and end range for the records you want to view.

3. Click **Export**.
4. Wait for GC to process your request.
5. Open or save the CSV file when your browser prompts you that it is ready.

Your audit trail logs can become large if many users and administrators actively log into and use your GC servers. When you export the audit logs, GC queries for all audit trail records with timestamps that fall within the specified date range and prints only the first 30,000 records into the CSV file if the response is too large.

If the exported CSV file does not contain the records you are looking for, simply select a more narrow date range, or change the date range such that the start date and time matches or slightly overlaps the timestamp of the last record in the exported CSV file, and export the audit logs again.

### Purging old audit trail records

Audit logs can grow large, particularly in deployments that have multiple GC servers or where many users log in to the self-service portal. With the GC console you can delete audit records that are more than thirty days old.

To purge old audit log records:

1. Navigate to the **Reporting** > **Audit Trail Logs** screen.
2. From the pulldown menu labeled **Purge older than**, select a number of days. This value gives GC the starting point for deleting older records.
3. Click **Purge**.

GC then removes all audit records that have an age greater than the number of days you have selected the purge to extend.

**Note:** GC cannot undo this action, so purge a log only as absolutely required. It is recommended that you export the logs before you delete them, so you can have a copy of the purged logs.

## Exporting Usage Data: Container Activity and Compliance Violations

**Note:** Your username in the Good Control must be a member of a role that has permission to view reports. For instance, the Help Desk Administrators predefined role does not have permission to view reports. Follow the steps in Creating and Configuring a Custom Role to create a role with the Reports and Troubleshooting permission that the Help Desk people need.

You can export usage data to comma-separated value (CSV) format for use in spreadsheet or other programs. The data is of two types:

- Container Activity by User/Device/App: metrics about number of activated containers and other data.
- Compliance Violations: Metrics about policies enforced on containers and violations of those policies. Enforced events (that is, those initiated by the administrator), such as wipe or lock, are not included.

**To export data:**

The exported data is always the complete history of your account from the day it was first created to the moment you export

> **Note:** The system writes the first 30,000 records to the file, after which no more data is written. To workaround this limitation you can change the value of the server property **`gc.reports.limit`**. See Updating GC Server Properties.

1. Navigate to **Reporting** > **Usage Data.**
2. Click the appropriate radio button:
   - Container Data
   - Compliance Data
3. Click **Export**.

Be patient as the system retrieves your data. When the data are ready, the system downloads a CSV file directly to your computer.

## Descriptions of Data, Fields, and the Reports

The data are encoded in UTF-8 character set, which can cause it to be displayed strangely in Microsoft Excel. Many solutions to this problem are described on the Internet. There are also other spreadsheet programs that deal with UTF-8 more gracefully.

The file name of the exported data is one of the following:

- **`GD_Containers_Report_`*`datestamp`*`.csv`**
- **`GD_Compliance_Report_`*`datestamp`*`.csv`**

where ***`datestamp`*** is the date the data were exported.

The data are derived from the following sources, as indicated for each metric:

1. An application or device
2. The directory service, such as Active Directory
3. The GC database

The following fields are exported from the GC database.

| Field | Description | Derived from | Report |
|---|---|---|---|
| Application Activation Date | Date the application was activated. | Application or device | Containers |
| Application ID | GD application identifier | Application or device | Containers, Compliance Violations |
| entitlement version | GD entitlement version number | Application or device | Containers, Compliance |

| Field | Description | Derived from | Report |
|---|---|---|---|
| | | | Violations |
| Carrier | Name of telephone carrier company, like AT&T or British Telecom | Application or device | Containers |
| Compliance Violation Date | Date that a policy was violated | GC | Compliance Violations |
| Container ID | Unique container identifier | Application or device | Containers, Compliance Violations |
| Department | User's department name | Directory service | Containers |
| Device Model Name | Model name form the manufacturer, like iPad Air or Samsung Galaxy 5S | Application or device | Containers |
| Device Name | Any identifier the user might have entered on the device itself | Application or device | Containers, Compliance Violations |
| Device OS Version | Operating system version | Application or device | Containers |
| Device Platform | Operating system name, like iOS or Android | Application or device | Containers |
| Device Type | Type of device:<br>• IPHONE<br>• IPAD<br>• ANDR | Application or device | Containers |
| Display Name | User's full name | Directory service | Containers, Compliance Violations |
| Domain | Name of directory service domain | Directory service | Containers |
| GD SDK Version | The version of the GD SDK used to build the application | Application | Containers |
| Last connection time to GC | Date/time of device's most recent connection to GC | GC | Containers |
| Phone Number | Device's phone number | Application or device | Containers |
| Policy Rule Failure Type | Type of policy failure. Any of the following or | GC | Containers, |

| Field | Description | Derived from | Report |
|---|---|---|---|
| | combinations of them:<br>• OS Version<br>• GD Library Version<br>• Jailbroken/Rooted<br>• Connectivity<br>• Device Model | | Compliance Violations |
| Policy Set | Name of policy set | GC | Containers |
| Serial Number | GC-generated unique identifier for the device | GC | Containers, Compliance Violations |
| User Email | User's email address | Directory service | Containers, Compliance Violations |

# Device Management App Inventory Reports

**Note:** Your username in the Good Control must be a member of a role that has permission to view reports. For instance, the Help Desk Administrators predefined role does not have permission to view reports. Follow the steps in Creating and Configuring a Custom Role to create a role with the Reports and Troubleshooting permission that the Help Desk people need.

**To generate the App Inventory report:**

1. Navigate to **App Inventory**.
2. From the pulldown menu, select the time to generate the report.
3. Click **Schedule**.

**To export the app inventory reports:**

You can export the following kinds of data to comma-separated value (CSV) format:

- App inventory
- App summary

1. Navigate to **App Inventory**.
2. Click **Export App Inventory List**.
3. Click the report with the data you want.

## Device Management Inventory Reports

**Note:** Your username in the Good Control must be a member of a role that has permission to view reports. For instance, the Help Desk Administrators predefined role does not have permission to view reports. Follow the steps in Creating and Configuring a Custom Role to create a role with the Reports and Troubleshooting permission that the Help Desk people need.

**To generate the Device Inventory report:**

1. Navigate to **Device Inventory**.
2. From the pulldown menu, select the time to generate the report.
3. Click **Schedule**.

**To export the device inventory reports:**

You can export the following kinds of data to comma-separated value (CSV) format:

Export the following kinds of data to CSV file:

- Device inventory list
- Device inventory change audit
- Device policy and configuration audit

1. Navigate to **Device Inventory**.
2. Click **Export Device Inventory List**.
3. Click the report with the data you want.

## Server Jobs

For relatively simple operations that do not require much time, such as assigning application permissions to a group or importing a single Active Directory user, GC processes such requests immediately.

However, complex operations like adding users in bulk require a different method of processing. GC servers create jobs to handle these types of requests.

Your GC servers maintain a queue of jobs, and the jobs are processed in order of submission. If you have only a single GC server in your cluster, it must process all jobs in the job queue. Therefore, if other unprocessed jobs are still waiting in the job queue, a newly created job can be delayed for some time. However, if you have multiple GC servers in your cluster, any of the servers can pick up the next job from the queue and process it.

Navigate to the **Reporting** > **Server Jobs** screen at any time to view a list of jobs that your GC servers have created.

This list displays details such as job type, state, start and end times, and whether errors were encountered while the job was running. Click any job to view more information.

## Viewing the Status of a Job

Navigate to the **Reporting** > **Server Jobs** screen to view a list of jobs that your GC servers have created.

This list displays details such as:

- Name of GC server associated with the job
- Job Type
- Status
- Start and end times

If a job has completed, the screen indicates whether errors were encountered while the job was running. Click any job to view more information.

On this screen, GC displays the configuration for the job and a progress bar. For a job that is currently running, the progress bar advances as the associated job tasks are completed. The task list, located beneath the progress bar, indicates the state of a task by color and icon: gray text indicates tasks that have not yet been processed, green text indicates the task has been completed successfully, red text indicates that errors were encountered, and blue text indicates the task currently being processed.

For a job that has been completed, this screen also shows a report on the output of the job. The report shows more information on any errors encountered.

# Miscellaneous security tasks or topics

## Changing the GC and GP Service Password

These are the general steps to change the password of the Windows service for the GC or GP servers.

To change the GC or GP service's password, you use the Windows console (sometimes called the Microsoft Management Console, or MMC) to operate on the service, either on each individual GC or GP computer or on the AD system.

**To change the service password for GC and GP:**

1. Stop the service. See Stopping the GC and GP Servers .
2. Go to the **Services** window.
3. Select the **GC** or **GP** server from the list of services and right-click it.
4. Select **Properties**.
5. Click the **Log On** tab.
6. Change the password.
7. Save or apply your changes.
8. Start the service. See Starting the GC and GP Servers .

# Updating the Web Proxy Server Password Used by a GC or GP Server

If your web proxy server account password changes after you have installed the GC and GP servers, you must update the password for all affected GC and GP servers by following the procedure detailed here.

**For each affected GC server:**

When a GC server is installed, the web proxy server information, if any, is stored in the GC database for that server. Consequently, each GC server in your cluster can potentially use a different web proxy server.

1. Log into the GC console and navigate to the **Server** > **Settings** screen and click the **Server Properties** tab.
2. Type the new password into the **proxy.auth.password** field and click **Submit** to commit the change.
3. Log into the GC server host machine and restart the GC server service. See Starting the GC and GP Servers .

**For each affected GP server:**

In the same way, any web proxy server information specified to a GP server installer is stored in the GP server property file at C:\good\gps.properties, and the web proxy account password is obfuscated and stored in the password file at C:\good\prot\gps_secure.properties.

1. Log into the GP server host machine and open a command prompt as an administrator.
2. Navigate to the *Good_Proxy_Install_Directory*\tools\password directory.
3. Run the following command to update the GP web proxy password:
   ```
   changepwd GP_PROXY new_password
   ```
   where *new_password* is the updated password.
4. Restart the GP service. See Starting the GC and GP Servers .

# Updating the SMTP Server Password Used by GC Servers

When a GC server is installed, the SMTP server information is stored in the GC database for that server. Consequently, each GC server in your cluster can potentially use a different SMTP server.

If the password for any SMTP account in use by any GC server is changed, you must update the password through the GC console for each affected GC server by following this procedure:

1. Log into the Good Control console of an affected GC server and navigate to the **Server Configuration** > **Settings** screen, then click the **Server Properties** tab.
2. Type the new password into the **gc.smtp.password** field and click **Submit** to commit the change.
3. Log into the GC server host machine and restart the GC server service.
4. Repeat steps 1-3 for each affected GC server.

# Updating the GC Database Password for Oracle and Windows Authentication to SQL Server

When installing a GC server, an administrator specifies connection information for a database. The database password is obfuscated and stored in the C:\good\prot\GC_secure.properties file on the GC server's host machine. All GC servers in the cluster use this password to connect to the database. If you change the password for the GC database user, you must also update the database connection password stored in this property file on the host machine of every GC server in your cluster.

The following sections describe how to change the GC database user's password for Oracle and SQL Server databases and update the password files for your GC servers.

If you do not have corporate security policies that govern the strength of passwords, we recommend the following minimum requirements for password strength:

- At least 8 characters in length
- At least one numeric character
- At least one special character
- No character used more than twice

### Updating an Oracle database password

Before you begin, you must know the username of the account your GC servers use to access the cluster database. If you do not know this information, you can locate it by logging into the host machine for one of your GC servers and finding the value of the **db.connection.user** property in the C:\good\GC-server.properties file.

> **Important:** Shut down all GC servers in your server cluster before changing the database user's password. This ensures that the database user's account is not locked due to failed login attempts from GC servers that you have not yet updated to use the new password.

Change the GC database user's password using the following procedure.

1. Log on to the machine that hosts the database.
2. Launch the Oracle **Run SQL Command Line** application and log in with the system user credentials.
3. Run the following command, where *GC_db* is the database user and *new_password* is the new password value.
   ```
   alter user GC_db identified by new_password;
   ```

You must also update the password property file for every GC server in the server cluster. The password in this file is obfuscated, so you need to use the password changing tool as described in the following steps.

1. Log on to the host machine of a GC server in the cluster.
2. Open a command prompt as an administrator.
3. Navigate to the *Good_Control_Install_Directory*\tools\password directory.
4. Run the following command, where new_password is the new password value.
   ```
   changepwd GC_DB new_password
   ```

5. Restart the GC service.
6. Repeat steps 1-5 for each GC server in the cluster.

### Updating a SQL Server database password

Before you begin, you must know the username of the account your GC servers use to access the cluster database. If you do not know this information, you can locate it by logging into the host machine for one of your GC servers and finding the value of the **db.connection.user** property in the C:\good\GC-server.properties file.

> **Important:** It is highly recommended that you shut down all GC servers in your server cluster before changing the database user's password. This ensures that the database user's account is not locked due to failed login attempts from GC servers not yet updated to use the new password.

Update the database user's password by using the SQL Server Management Studio tool as described in the following steps.

1. Launch the **SQL Server Management Studio** application and connect to the database as an administrator with the sysadmin server role.
2. Expand the GC database in the Object Explorer pane. Go to **Security** > **Logins**.
3. Right-click the database user and select the **Properties** option to view the General properties screen.
4. Enter the new password in both the Password and Confirm Password input fields.
5. Click **OK** to commit the change.

You must also update the password property file for every GC server in the server cluster. The password in this file is obfuscated, so you need to use the password changing tool as described in the following steps:

1. Log on to the host machine of a GC server in the cluster.
2. Open a command prompt as an administrator.
3. Navigate to the *Good_Control_Install_Directory*\tools\password directory.
4. Run the following command:
   ```
   changepwd GC_DB new_password
   ```
   where *new_password* is the new value.
5. Restart the GC service.
6. Repeat steps 1-5 for each GC server in the cluster.

# Updating the GC or GP Certificate Keystore/Truststore Password

> **Note:** You might also want to see the related topic Updating the Passwords of Private Keys Associated with GC or GP Certificates .

When a GC or GP server is installed, it is configured to use the default Java keystore password **changeit** to access the keystore and trust store located at <*Good_Product_Install_Directory*>\jre\lib\security\cacerts. The keystore contains information about the certificate used by the GC or GP server; the same file by default is used to store information about trusted Certificate Authorities (CAs). The keystore password is obfuscated and stored as the value of the

**gd.security.keystore.password** property in the C:\good\prot\gc_secure.properties file for a GC server and the C:\good\prot\gps_secure.properties file for a GP server. It is not necessary to change the keystore password from the default value, but if you wish to do so, follow the procedure outlined here.

This procedure includes instructions for changing the Java keystore password for a GC or GP server and updating the password that the server uses to access the Java keystore. Follow all steps for the type of server whose keystore password you are changing. Otherwise, the server cannot access the Java keystore and retrieve certificate information.

The default password for the keystore/trustore is **`changeit`**.

## Updating the keystore password for a GC server

To update the Java keystore/truststore password for a GC server:

1. Log into the server host machine.
2. Open a command prompt as an administrator.
3. Navigate to the <*Good_Control_Install_Directory*>\jre\bin directory.
4. Run the following command to update the keystore password, where *new_password* is the new password you have chosen for the keystore.
   ```
   keytool -storepasswd -new new_password -keystore ..\lib\security\cacerts
   ```
5. When you are prompted for the current keystore password, enter *changeit* if this is the first time you are changing the keystore password; otherwise, enter your current custom password for the keystore.

To configure a GC server to use the new Java keystore/truststore password:

1. Log on to the server host machine.
2. Open a command prompt as an administrator.
3. Navigate to the <*Good_Control_Install_Directory*>\tools\password directory.
4. Run the following command, where *new_password* is the new password you have chosen for the keystore/truststore files. The command sets the password on both.
   ```
   changepwd GC_KEYSTORE new_password
   ```
5. Restart the GC service.

## Updating the keystore password for a GP server

To update the Java keystore password for a GP server:

1. Log into the server host machine.
2. Open a command prompt as an administrator.
3. Navigate to the <*Good_Proxy_Install_Directory*>\jre\bin directory.
4. Run the following command to update the keystore password, where *new_password* is the new password you have chosen for the keystore.
   ```
   keytool -storepasswd -new new_password -keystore ..\lib\security\cacerts
   ```

5. When you are prompted for the current keystore password, enter *changeit* if this is the first time you are changing the keystore password; otherwise, enter your current custom password for the keystore.

To configure a GP server to use the new Java keystore password:

1. Log on to the server host machine.
2. Open a command prompt as an administrator.
3. Navigate to the <*Good_Proxy_Install_Directory*>\tools\password directory.
4. Run the following command, where *new_password* is the new password you have chosen for the keystore.
   ```
   changepwd GP_KEYSTORE new_password
   ```
5. Restart the GP service.

# Updating the Passwords of Private Keys Associated with GC or GP Certificates

**Note:** You might also want to see the related topic Updating the GC or GP Certificate Keystore/Truststore Password .

When a GC or GP server is installed, it is configured with a default value *changeit* as the password of the private keys associated with the GC or GP server certificates. It is not necessary to change the private key password from the default value, but if you wish to do so, follow the procedure outlined here.

The procedure for updating the private key password involves two main concepts. First, you must use a Java keytool command to change the password of associated private keys in the Java keystore for the GC or GP server. Then, you must update the value of the stored password that the GC or GP server uses for the private keys.

The steps of the procedure are separated into two sections: one section for GC servers and one section for GP servers. Follow the steps applicable for the type of server whose private key password you are changing.

If you change the password of one private key, you must change the passwords of all other private keys in the server's keystore to the same new value.

## Updating the private key password for a GC server

1. Obtain the password for the Java keystore, located at <*Good_Control_Install_Directory*>\jre\lib\security\cacerts. If the keystore password has not been modified since the GC server was installed, replace **`keystore_password`** with *changeit* in the steps detailed here. Otherwise, replace **`keystore_password`** with the value of the current Java keystore password.
2. Log into the server host machine.
3. Open a command prompt as an administrator.
4. Navigate to the <*Good_Control_Install_Directory*>\jre\bin directory.
5. Run the following command to update the password of the **gcca** private key, where **`new_password`** is the new password chosen for the private key.
   ```
   keytool -keypasswd -new new_password -alias gcca -keystore
   ..\lib\security\cacerts -storepass keystore_password
   ```

6. Run the following command to update the password of the **gc** private key, where **new_password** is the new password value.
   ```
   keytool -keypasswd -new new_password -alias gc -keystore ..\lib\security\cacerts
   -storepass keystore_password
   ```
7. If your organization has imported additional private keys into the keystore, you must also update the passwords for those private keys. For each private key, run the following command to update the private key's password, where **key_alias** is the alias of the private key and **new_password** is the new password value.
   ```
   keytool -keypasswd -new new_password -alias key_alias -keystore
   ..\lib\security\cacerts -storepass keystore_password
   ```
   Repeat this step for each private key.
8. Navigate to the <*Good_Control_Install_Directory*>\tools\password directory.
9. Run the following command, where **new_password** is the new password value. This command updates the password the GC server uses for the private key.
   ```
   changepwd GC_PRIVATEKEY new_password
   ```
10. Restart the GC service.

### Updating the private key password for a GP server

1. Navigate to the <*Good_Proxy_Install_Directory*>\jre\bin directory.
2. Run the following command to update the password of the **good-proxy** private key, where **new_password** is the new password chosen for the private key.
   ```
   keytool -keypasswd -new new_password -alias good-proxy -keystore
   ..\lib\security\cacerts -storepass keystore_password
   ```
3. If your organization has imported additional private keys into the keystore, you must also update the passwords for those private keys. For each private key, run the following command to update the private key's password, where **key_alias** is the alias of the private key and **new_password** is the new password value.
   ```
   keytool -keypasswd -new new_password -alias key_alias -keystore
   ..\lib\security\cacerts -storepass keystore_password
   ```
   Repeat this step for each private key.
4. Navigate to the <*Good_Proxy_Install_Directory*>\tools\password directory.
5. Run the following command, where **new_password** is the new password you have chosen for the keystore. This command updates the password the GP server uses for the private key.
   ```
   changepwd GP_PRIVATEKEY new_password
   ```
6. Restart the GP service.

# Maintenance & troubleshooting

## BlackBerry Marketplace  Org ID Displayed in Good Control

When you become a BlackBerry partner or customer, your organization is assigned an organization ID (or org ID) by the BlackBerry Marketplace system. This org ID is displayed on **Overview** page of your BDN account.

For ease of administration, this same org ID is now displayed in the Good Control console heading itself at the top of the page, so you can corelate your BlackBerry Dynamics deployments with your BDN account details.

# Monitoring GC and GP Server Health

With the GC console you can quickly view the health of all GC and GP servers in the server cluster.

Navigate to the **Server Configuration** > **Status and Diagnostics**.

Basic information and a color-coded status indicator are displayed for each server in the cluster. Server status is represented by one of the following:

- 🟢 **Connected** - The server is successfully connected to other GD platform servers.
- 🔴 **Disconnected** - The server is disconnected from one or more GD platform servers.
- ⚪ **Unknown** - The server has not yet responded to the request for status or the server's status is unknown.

For GP servers, this page also displays the date and time of the **Last heartbeat received from NOC**.

If any of your GD servers have a 🔴 **Disconnected** status and you need to troubleshoot the issue, see Issue: A GC or GP server is disconnected from other GD servers.

Click a server to view more detailed information about the server, such as the version of the GD software the server is currently running. An example of this is shown .

From this screen, you have the option to unregister a server and remove it from your GD server cluster. For more information, see Uninstalling a GC or GP server from the cluster. You cannot unregister the GC server you are currently logged into.

# Behavior and Model of Disconnected/Inactive Containers

When a device or container has been decommissioned, the GD Network Operations Center (GD NOC) is notified that a container is no longer in use by way of the user (who removes or deletes the container), the Good Control administrator's explicit actions, or by GC's inactivity purge model:

- When the GD Runtime starts on a device, it first connects to the GD NOC to obtain its current entitlement status. If the user, container, or device has been deleted or the user has lost entitlement to the application through an administrative action, the GD Runtime immediately wipes the container.
- If the GD Runtime is already running, as soon as an administrative action changes the user's entitlement to the application, a GD notification (via the GNP, or Good Notification Protocol) is sent to the running GD Runtime to force an immediate wipe. Containers that have been wiped can no longer connect to any GPs because they no longer have access to keys, addresses, or application data that are needed in order to connect.

## Model for Disconnected or Inactive Containers

The BlackBerry Dynamics model has multiple ways to handle the case where a given container has not connected to either Good Proxy or to Good Control within a configured period of time. This model is based on the following principles:

- We do want to require containers to connect to Good Control within a policy-controlled period of time. This is called *Connectivity Verification*.
- We do *not* want to force containers to always have to connect to Good Control if they otherwise have a valid path from Good Proxy to the applicable application server.
- This allows for the case where GC is temporarily unavailable (such as planned maintenance or unplanned downtime), but the container otherwise has valid key to connect to Good Proxy or the application server.
- Otherwise, such downtime would always impact end users' ability to continue to access application servers.
- However, we do want to require containers to connect to GC at least once within an administrator-configured period of time to continue accessing GPs and application servers. Failure of a container to connect in a specified period of time triggers the purging of that container. This is called "purging inactive containers" or *Inactivity Purge*.

The combination of these principles ensures that:

- All containers have to connect to GP and through GP to GC within a specified period of time to remain in compliance and continue to have access to application servers.
- Planned or unplanned GC downtime on its own does not automatically and always result in user-impacting downtime because users can still access GPs and application servers, even if GC is temporarily down.

Interrupting this normal operation, such as unexpectedly removing GP servers, can affect these functions.

## Connectivity Verification

The Connectivity Verification method is implemented by the GD Runtime (the GD SDK), is explicitly designed to operate independently of the GC, and is applied when a Connectivity Verification compliance policy has been set by the administrator. If the policy is set and a container does not connect to a GC within the specified verification period, the GD Runtime will immediately take the action to either ‘block’ container access, or ‘wipe’ the container, as specified in the policy. Having Connectivity Verification operate independently in this manner guards against cases where an app is designed to operate in ’offline‘ mode and somebody with malicious intent purposely keeps the device in a disconnected state for an unusually long period of time to avoid the application of new policies and/or remotely initiated ‘block’ or ‘wipe’ commands.

The default Connectivity Verification period is 30 days, but it can be set higher or lower. Setting it significantly lower, however, may have unintended consequences as there are many legitimate scenarios where a given device or app will remain unused and not connect for multiple days or weeks at a time. For more information, see Configuring Compliance Policy Rules .

## Purge Inactive Containers

The Purge Inactive Containers method is implemented by GC, where it identifies inactive containers and schedules batch jobs to remove or purge them. The process of purging a container involves removing it from GC and the GD NOC and the revocation of keys it uses to connect to GC and through GP to application servers. This ensures that any container not actively connecting to GC within the configured period of time will be purged. This relieves the IT administrator from having to deal with this task manually and also handles edge cases where a container continues to connect to GP using valid keys and within the Connectivity Verification period (because we do not want GC downtime to always and automatically lead to service disruption), but still has not been able to connect to a GC to receive policy updates or ‘block’ or ‘wipe’ commands over an extended period of time. By default, the length of time before a container is considered inactive is 90 days, and by default the container management batch job runs once a day to determine if a container's last connection time exceeds the inactivity threshold and thus should be removed. Deletions are recorded in the GC log. As a safety factor to account for system downtime in the calculation of inactivity, you can set a certain amount of time to adjust the calculation forward to accommodate devices that might have attempted to reconnect during that downtime. When GC starts, it checks its last activity time stamp (updated every minute) and by default, if that time stamp is older than one day (by default), the GC adjusts all containers' last activity stamp by the time difference. For example, if the GC is down for three days, then containers are given an additional three days to connect. This ‘drift’ design accounts for unlikely lengthy downtime and database restores.

For more information on these functions, see Duplicate Containers and Purge Inactive Containers .

# Decommissioning Good Control or Good Proxy

For background on the behavior of inactive containers, see Behavior and Model of Disconnected/Inactive Containers .

A container might continue to connect to a GP and through it to an application server, even if does not also and always connect to a GC. This ability is limited, however, and is only possible if the container is still within the container inactivity/purge threshold. Given this intentional ability of the GD Runtime and GP to operate independently of GC for limited periods of time, before fully decommissioning a GP cluster:

> **Important:** It is best practice to bring up the new cluster and keep the old cluster running for a transitional period of time equal to the configured container inactivity/purging period.

Doing so ensures that all actively used containers will both connect to GP and to GC within the transitional period and, by doing so, obtain the updated Connectivity Profile from GC that subsequent container connections to the ‘new’ GP cluster.

Meanwhile, any remaining containers that do not connect to the old GP cluster within that transitional period and from there to GC will (and should) be purged anyway as a matter of security practice and to guard against the possibility that a container continues to connect to GP, within Connectivity Verification period, but without connecting to GC itself.

For example, if you set Connectivity Verification period to 30 days and Inactivity threshold to 60 days, then you should allow 60 days for transition to complete. Active containers connecting to GC within that day period will obtain updated Connectivity Profiles, inactive containers will be purged anyway and will receive Connectivity Profile on re-activation.

The reason that the Connectivity Verification period is less than the Inactivity threshold is that this allows for n>where certain apps are more infrequently by their nature. In such case, the administrator wants to guard against where someone maliciously keeps offline, but the administrator also wants a legitimate user to be able to unblock without loss of data by using the ‘block’ vs. ‘wipe’ action. This is less impactful on the end-user than ‘wiping’ or ‘purging’.

It does not make sense to have a Connectivity Verification period that is greater than Inactivity threshold. By definition, a container that reaches GC through GP in first place will have satisfied Connectivity Verification compliance rule. So, it is not possible to have actively connected to GC within last 60 days, for instance, and not have also connected to GP within that same time.

# Disabling Zipping of Logfiles

Good Control and Good Proxy are configured by default to compress (zip) their logfiles to conserve space on disk.

You can disable the zipping of logfiles with these steps.

> **Note:** The change described here does not take effect until the service is restarted. Logfiles created before this change are not affected.

## Steps for Good Control

These are the configuration files involved. You need to change each of them:

- *gc_installation_directory*`\webapps\gc-server\WEB-INF\classes\log4j.properties`
- *gc_installation_directory*`\webapps\good-control\WEB-INF\classes\log4j.properties`

1. Make a backup copy of the configuration file.
2. Edit the original file.
3. Change the value of the properties as shown below:

| File | Change Property Value From true... | To New Value false |
|---|---|---|
| *gc_installation_directory* `\webapps\gc-server\WEB-INF\classes\log4j.properties` | `log4j.appender.tofile.Compress=true`<br>`log4j.appender.axisLog.Compress=true` | `log4j.appender.tofile.Compress=false`<br>`log4j.appender.axisLog.Compress=false` |
| *gc_installation_directory* `\webapps\good-control\WEB-INF\classes\log4j.properties` | `log4j.appender.tofile.Compress=true` | `log4j.appender.tofile.Compress=false` |

4. Save the file.
5. Restart the GC as described in Starting the GC and GP Servers .

## Steps for Good Proxy

1. Copy the file *gp_installation_directory*`\log4j.properties` to `c:\good\gpslog4j_override.properties`.

> **Note:** The new file name must be exactly as shown: `c:\good\gpslog4j_override.properties`.

2. Edit **`c:\good\gpslog4j_override.properties`**.
3. Change the following property:

   From **`log4j.appender.tofile.Compress=true`**

   To: **`log4j.appender.tofile.Compress=false`**
4. Save the file.
5. Restart the GC as described in Starting the GC and GP Servers .

# Sending Server Logs to Good for Analysis

If an issue arises and you need to contact BlackBerry for technical support, the support person might request server logs for a certain date range to help pinpoint the problem. You can authorize this through via GC console; the system automatically sends the GC and GP logs for the requested dates.

## Audit Logs Not Uploaded By Default

Good Control is configured by default to not upload the audit logs because the size of these logs can impact performance.

If you need to include the audit logs, edit a configuration file to enable the upload.

### Steps

1. Make a backup copy of *`gc_installation_directory`***`\webapps\gc-server\WEB-INF\classes\log4j.properties`**.
2. Edit the original file.
3. Change the value of the property **`log4j.appender.axisLog.Uploadable`**:
   - From: **`log4j.appender.axisLog.Uploadable=false`**
   - To: **`log4j.appender.axisLog.Uploadable=true`**
4. Save the file.
5. Restart the GC service as described in Starting the GC and GP Servers .

## Automatic GC-Cluster-wide Server Log Uploading

When you upload Good Control server logs from one of your GC servers to the GD NOC, all log information from all GC servers in the cluster is automatically included in the upload.

**Note:** There is no need to upload logs from each individual server.

If you create a scheduled task to upload logs on one of the Good Control servers in a cluster, you will see that scheduled task in the UI of all the servers in the cluster.

## Steps

If an issue arises and you need to contact Good for technical support, the Good technician might request server logs for a certain date range to help pinpoint the problem. You can authorize this through via GC console; the system automatically sends the GC and GP logs for the requested dates.

**To immediately upload cluster-wide logs:**

1. Navigate to **Troubleshooting** > **Upload Server Logs**.
2. For immediate upload, under the **Upload Logs Now** heading, select the desired number of days
3. Click **Upload Now**. The Upload Status Messages box displays the status of your upload.

**To set a schedule for uploading logs:**

1. Under the **Set an Upload Schedule**, click the **From** field to set the start date.
2. Click the **To** field to set an end date.

   You can supply any dates, including those in the past or future. If the end date is today or a future date, the **Send Every** selection appears. Use it to select a time interval from the options in the pulldown. This value determines how often the GC and GP logs are sent to Good. Default is every 8 hours.
3. Click **Set Schedule** to begin upload. The Upload Status Messages box displays the status of each scheduled upload. The current schedule is displayed on this screen until the end date and time is reached.

**To cancel the schedule:**

You can cancel the log transfer schedule at any time by. Click **Cancel Schedule**.

# Summary of GC and GP Logs and Locations

This is a summary description of the Good Control and Good Proxy logs and their default locations.

The filenaming convention for most of the logs follows this pattern:

*logtype_hostname*_GD*serialnumber_YYYY-MM-DD_hh.mm.ss*.log

Example: gc_server_mymachine_GD110093934_2016-06-17_10.26.25.log

| Server | Log Name | Default Location | Description | Example of Logfile Name |
|---|---|---|---|---|
| Good Control | | | | |
| | Installation Logs | C:\good\ialogs | Record of installation | good_control_install_log.txt<br>good_control_ia_error_log.txt |
| | Server Logs | C:\good\gclogs | Logs all the operations and command executions for GC server:<br>• Login/Logout | gc_server_mymachine_GD0987654321_2016-06-17_10.26.25.log |

| Server | Log Name | Default Location | Description | Example of Logfile Name |
|---|---|---|---|---|
| | | | • User management<br>• Policy management<br>• Roles management<br>• Apps management<br>• GP/Cluster setup<br>• Background Jobs execution<br>• Container/GC interactions and messages<br>• GC/NOC/GP interactions<br>• DB Poller execution to monitor the DB change<br>• GC Properties and their values<br>**Note:** Passwords, access keys, and device enrollment keys are not logged. | |
| | Audit Logs | C:\good\gclogs | Logs all SOAP actions for auditing purpose | gc_audit_trail_mymachine_GD0987654321_2016-06-17_10.27.31 |
| | Good Control UI Logs | C:\good\gclogs | Logs all GC UI interactions | good_control_hostname_GD0987654321-2016-06-16_17.20.09.log |
| | Startup Logs | C:\good\gclogs | Record of startup of integrated application server | goodcontrol-stderr.log.2016-06-16<br>goodcontrol-stdout.2016-06-16 |
| | Database Logs | C:\good | Record of database operations | gc-db-2016-06-17-07-07-2 |
| Good Proxy | | | | |
| | Installation Logs | C:\good\ialogs | Record of installation | good_proxy_install_log.txt<br>good_proxy_ia_error_log.txt |
| | Server Logs | C:\good | Logs all GP server operations and communications to NOC, GC, and application servers:<br>• GP to NOC interactions | gpslog_hostname_2016-06-17_07.12.36 |

| Server | Log Name | Default Location | Description | Example of Logfile Name |
|---|---|---|---|---|
| | | | • MDC Heartbeat<br>• GP and app servers interactions<br>• GP and GC communications<br>**Note:** Email contents and application data are not logged. | |
| | Heap information log | C:\good | Information about the Java heap (memory) of Good Proxy | GP_gc.log |
| | Startup Logs | C:\good | Record of startup of integrated application server | gps-stderr.log.2016-06-16<br>gps-stdout.2016-06-16 |

# Enabling Debug Logging for a GP Server

GP server log files, stored in C:\good\gpslogs on each GP server's host machine, are useful resources when troubleshooting client connection related issues. To keep log s smaller, GP servers by default do not log all debug statements; however, you can change the logging level for each individual GP server as needed.

Follow these instructions for each GP server you want to enable debug logging on:

1. Log into the GP server host machine.
2. Navigate to the root directory where GP is installed.
3. Make a copy of the log4j.properties file and rename the copy to gpslog4j_override.properties.
4. Open the gpslog4j_override.properties file in a text editor and set the values of the following properties as indicated:
   - log4j.rootLogger=DEBUG, tofile
   - log4j.logger.com.good=DEBUG
   - log4j.logger.org.apache=DEBUG
   - log4j.logger.httpclient.wire=DEBUG
5. Save your changes and move the gpslog4j_override.properties file to the C:\good directory.
6. Restart the GP server.

**Note:** After you have gathered enough detailed debugging information to solve your problem, set the logging level back to INFO so the size of the logfile does not get large.

# Setting GP Server logfile size limit

GP server log files, stored in C:\good\gpslogs on each GP server's host machine, are useful resources when troubleshooting client connection related issues. To keep log s smaller, GP servers by default do not log all debug statements; however, you can change the logging level for each individual GP server as needed.

Follow these instructions for each GP server you want to enable debug logging on:

1. Log into the GP server host machine.
2. Navigate to the root directory where GP is installed.
3. Make a copy of the log4j.properties file and rename the copy to gpslog4j_override.properties.
4. Open the gpslog4j_override.properties file in a text editor and set the values of the following properties as indicated:
   - **`log4j.appender.tofile.MaxFileSizeInKB=256000`**

     Default is 256 MB. Allowable range: 100 KB to 1 GB.
5. Save your changes and move the gpslog4j_override.properties file to the C:\good directory.
6. Restart the GP server.

# Enabling info logging for a GC server

GC relies on the well-known log4j software to set the density (amount of detail) of its logging.

The highest level of detail, debug logging (or "DEBUG"), is useful for finding the cause of problems. It can also make your logfiles' size extremely large. For normal operations, it is best to run with logging set to "info" level to keep the logfile size to a minimum.

You can change the logging level for each individual GC server as needed.

Follow these instructions for each GC server you want to enable "info" logging on:

1. Log into the GC server host machine.
2. Navigate to the directory where GC keeps the logging properties; for example, on Windows, **`C:\Program Files\BlackBerry\Good Control\webapps\gc-server\WEB-INF\classes`**.
3. Make a backup of the **`log4j.properties`** file.
4. In the **`log4j.properties`** file, change all occurrences of **`DEBUG`** to **`INFO`**.
5. Save the file.
6. Restart the GC service.

# Configuring GC and GP Server Log Retention

### Good Control server logs

GC console and server log files, located in C:\good\gclogs on each GC server’s host machine, contain a large amount of information pertaining to server processes, requests, and responses. They are useful sources of information if you need to troubleshoot a GC console or server related issue. These logs can grow large, depending on the load of activities on the GC server. Factors affecting the size of the logs include the frequency that GC users and administrators log into the portal and request operations, and the number of GD applications your users activate and actively use.

All GC servers in the server cluster have a default configuration to retain logs for 10 days. This global setting resides in the cluster database and can be changed through the GC console. GC servers automatically delete log files older than the number of days specified as the value of this setting.

To modify the number of days for GC server log retention, follow these instructions:

1. Log into the console of any GC server in your server cluster.
2. Navigate to the **Server Configuration** > **Settings** screen and select the **Server Properties** tab.
3. Find the **gcs.logfiles.days** property.
4. Set the value of this property equal to the number of consecutive days for your GC servers to retain logs.
5. Click **Submit** to commit the change.
6. Restart the Good Control service for each GC server in the cluster.

### Good Proxy server logs

GP server log files, stored in C:\good\gpslogs on the GP server’s host machine, are useful resources when troubleshooting client connection issues. These logs can grow large as more users in your organization activate and use more GD applications.

All GP servers have an internal default configuration to retain logs for 10 days. This setting is configurable in the C:\good\gps.properties file located on each GP server's host machine. If a value is configured for this setting in the property file, a GP server deletes log files older than the specified value. Otherwise, the GP server deletes logs older than 10 days.

To modify the number of days for GP server log retention, follow these instructions on each individual GP server:

1. Log into the GP server host machine.
2. Stop the Good Proxy Server service.
3. Navigate to the C:\good directory.
4. Open the gps.properties file in a text editor.
5. Search for the **gps.logfiles.days** property.
   - If the file contains this property, modify its value to the number of days to retain logs.
   - If the file does not contain this property, add the following line to the end of the file, where *num_days*:
     **`gps.logfiles.days=`***`num_days`*
6. Save your changes to the gps.properties file.
7. Start the Good Proxy service.

## Starting the GC and GP Servers

These are the steps to start or restart the GC and GP servers.

> **Note:** If you are running a cluster, be sure to start all GC and GP servers in the cluster.

1. Go to the **Services** window.
2. Select the GCserver from the list of services and click **Start** .
3. Select and start the GP server. If the Good Proxy server is installed on a different system, remote-connect to that system and follow the same steps above to start the GP server.

## Stopping the GC and GP Servers

These are the steps to stop the GC and GP servers.

**Note:** If you are running a cluster, be sure to stop all GC and GP servers in the cluster.

1. If necessary, run `services.msc` from the Windows **Start** > **Run** prompt to start the Services window.
2. Select the GC server from the list of services and click **Stop**.
3. Select and stop the GP server. If the Good Proxy server is installed on a different system, remote-connect to that system and follow the same steps above to stop the GP server.

## After Restarting the GC Database

If you need to restart the GC database service for some reason, keep in mind the following:

- You also need to restart the GC service. See Stopping the GC and GP Servers and Starting the GC and GP Servers.
- To login to GC after the restart, you need to reload the GC console login page in your browser.

## Increasing the GC or GP server's Java Heap Size

You might find the need to change the size of the memory heap of the Java Runtime Engine (JRE). These are the steps.

The heap size must be set depending on your hardware configuration, such as amount of physical memory or number of CPUs. Overallocating memory can cause other performance issues.

BlackBerry recommends that you set the size of the GP server's Java heap to 60% of physical memory, with an upper limit not to exceed 5 GB. If you think it necessary to allocate more than this, get guidance from Good Technical Support.

**To increase the GC or GP server's Java heap size:**

You should be familiar and comfortable with using the Windows Registry Editor (`regedit` command). The HKEY entries in the registry for GC and GP are as follows:

| | |
|---|---|
| GC | HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Apache Software Foundation\Procrun 2.0\GoodControl\Parameters\Java |
| GP | HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Apache Software Foundation\Procrun 2.0\GPS\Parameters\Java |

1. Stop the GC or GP server. See Stopping the GC and GP Servers.
2. Login to the GC or GP server as administrator.
3. Start the Windows Registry Editor (**`regedit`** command).
4. Find the desired entry in the registry tree for either GC or GP.
5. Under the **`Java`** node, set the values of following keys to the desired settings. These keys correspond to the similarly named options on the JRE command:

   > **Important:** The initial values for both keys should be the same.

   - **`JvmMs`**: Amount of memory in megabytes to allocate to the Java Virtual Machine at start
   - **`JvmMx`**: Maximum amount of memory in megabytes to allocate to the Java Virtual Machine
6. Save your changes to the registry.
7. Start the GC or GP server. See Starting and Restarting the GC and GP Servers.

# Issue: Data Loss/Dropped Client Connections

Starting with Windows Vista, Microsoft introduced Receive Window Auto-Tuning, which continually monitors network conditions and scales the TCP Receiving Window for optimal performance. Unfortunately, older routers and firewalls that cannot process the window scaling correctly might cause data loss or dropped connections.

To avoid this issue, if your GP server is on a machine with an OS more recent than Vista, we strongly recommend that you do the following steps:

1. Launch a command prompt with elevated credentials.
   You must open a 64-bit command prompt if you are running a 64-bit version of Windows.
   For 32-bit and 64-bit Windows installations, navigate to your Windows System32 directory (usually C:/Windows/System32), right-click the **cmd.exe** file, and select the **Run as administrator** option.
2. Run the following command:
   ```
   netsh interface tcp set global autotuninglevel=disabled
   ```
3. Restart the machine.

# Issue: User cannot Activate an Application

Ensure all of the following conditions are met:

- The application has been registered in the GC console.
- The registered application ID and version in the GC matches with that configured in the client application.
- The user has been added to GC.
- The user has been allowed access to use the application individually or through being a member of an application group.
- The user has not been denied access to use the application.
- The user has installed the corresponding application on their device.

- The user has a valid access key.
- The user has entered their email address and access key when the application was launched.

# Issue: Java Exception CertificateNotYetValidException: NotBefore

After creating a server certificate, you might see the following Java exception when you are trying to install it:

```
java.security.cert.CertificateNotYetValidException: NotBefore
```

This means that the GC server and GP servers are not time-synchronized. There can be several causes for the time not being synchronized:

- Make sure that the time on all servers in your cluster is synchronized via the Network Time Protocol (NTP), if NTP was not set up at installation.
- The GP server might show this exception, even if it is time-synchronized with its GC server. The reason is that when a certificate is created it is time-stamped. Even the slight delay between creating the certificate and installing it on the GP server can cause the time window for installation to be exceeded and cause the installation to fail. The solution is to set the clock on your GP server slightly ahead of the time on the GC server. This allows the certificate to be installed within the time-stamped window.

After you set the GP server's clock ahead and install the certificate, be sure to set its clock back (or resynchronize with NTP) for normal operation.

# Issue: A GC or GP server is disconnected from other GD servers

If any of your GD servers unintentionally have a status on the Server Configuration > Status and Diagnostics screen and you need to troubleshoot the issue, please ensure the following:

- The host machine for the GC or GP server is running and connected to your network.
- The GC or GP service on the server's host machine is running.
- The GC or GP host machine is able to connect to the GC NOC servers.
- The GC or GP server is configured with correct web proxy connection information, if your server connects through a web proxy server.
- All GC host machines are able to listen for inbound connections on ports 443 and 17317.
- All GP host machines are able to listen for inbound connections on ports 17088 and 17433.
- The GC or GP server is not logging other unexpected errors.

GC server logs are located at C:\good\gclogs, and GP server logs are located at C:\good\gpslogs. These server logs are invaluable when troubleshooting issues you may encounter. If you need additional help diagnosing an issue, you can configure your GC to send your server logs to the Good Tech Support team. For more information, see Sending server logs to Good for analysis.

# Migrating the Good Control database

Good Control supports databases from several different vendors. However, BlackBerry recommends that you migrate only among different database servers from the same vendor, such as Microsoft SQL Server to Microsoft SQL Server, not across vendors such as Microsoft SQL Server to Oracle. Migrating across different types of databases requires more extensive work than is discussed here.

## Do not change the Good Control login authentication method

When you migrate from one database to another, do not change the authentication method used for logging in to Good Control. For instance, if the first database was set up to use SQL authentication for login, do not switch to use Windows authentication in the new database.

## About migrating the database of a Good Control cluster

**Important:** If you are running a cluster of Good Control servers, you must:

- Shut down all GC servers in the cluster before migrating the database.
- Follow the procedures detailed here for all GC servers in the cluster.
- After finishing the procedures, restart all GC servers in the cluster.

## Good Control's server properties file and database connections

The property file **C:\good\gc-servers.properties** includes database-related properties.

**Important:** You should make a backup copy of this file before you change the original.

After migrating your data, be sure to verify the following settings.

1. DB connection string. This is the value of **`db.connection.url`** property, as shown in the examples below.
2. DB user/owner. This is the value of the **`db.connection.user`** property, as shown in the examples below.

### gc-servers.properties file with details for Oracle database

**Note:** In this example of **`gc-servers.properties`**, in the value of the **`db.connection.url`** property, the host name is **`localhost`**: **`(HOST\=localhost)`**. However, if your database is installed on a machine separate from your Good Control server, make sure you use the fully qualified domain name of your database server so your GC server can resolve the hostname of the database server.

Also make sure that the value of **`SERVICE_NAME`** (**GC**, in this example) matches what is required by your Oracle instance.

```
db.connection.url=jdbc\:oracle\:thin\:@(DESCRIPTION\=(ADDRESS\=(PROTOCOL\=tcp)(HOST\=localhost)
(PORT\=1521))(CONNECT_DATA\=(SERVICE_NAME\=GC)))
db.connection.user=gc_db
db.dialect=org.hibernate.dialect.Oracle10gDialect
db.driver=oracle.jdbc.driver.OracleDriver
```

```
gd.product.hostname=gc-server-123
gd.product.licensekey=5649-8E49-C9C7-C1D7-78EF-116B
gd.product.serialnum=GD1000001
```

### gc-servers.properties file with details for Microsoft SQL Server database

> **Note:** In this example of **`gc-servers.properties`**, in the value of the **`db.connection.url`** property, the host name is **`localhost`**: **`sqlserver\://localhost`**. However, if your database is installed on a machine separate from your Good Control server, make sure you use the fully qualified domain name of your database server so your GC server can resolve the hostname of the database server.

Also make sure that the value of **`databaseName`** (**`gc`**, in this example) matches name of your Oracle database.

```
db.-
con-
nection.url=jdbc\:sqlserver\://localhost\\\SQLExpress\:1433;databaseName\=gc;selectMethod\=cursor

db.connection.user=gc_db
db.dialect=org.hibernate.dialect.SQLServerDialect
db.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
gd.product.hostname=gc-server-456
gd.product.licensekey=1A51-6D26-8469-6AEF-6361-7298
gd.product.serialnum=GD1000002
```

## Changing the database password

Depending on how it was set up, your new database might have a different password than was set for the database you are migrating from.

BlackBerry supplies a script to update the password that Good Control needs to access the database. This script prompts you for the new database password, obfuscates the password, and then stores the obfuscated password for best security.

**To change the database password:**

1. On the Good Control server host machine, open a command window as administrator and enter the following command:

   *`gc_install_dir`***`\tools\password\changepwd.bat`**
2. You are prompted to select which password you want to change:
   - Enter: **GC_DB**
   - Enter the new password.
3. For changes to these properties to take effect, you need to restart the Good Control service. Go to the **Services** window.
4. Select the GC server from the list of services.
5. Click the **Start** command.

### verification

BlackBerry recommends that with either SQL commands or Microsoft SQL Server management console, log in to the applicable database with the GC database user and password to verify that the account has full access.

### Possible errors

If the above issues have not been addressed, the following errors might be recorded in the GC server logs when the GC is started.

- **`org.hibernate.exception.GenericJDBCException: Cannot open connection at org.hibernate.exception.SQLStateConverter.handledNonSpecificException (SQLStateConverter.java:140)`**
- **`java.sql.SQLException: Connections could not be acquired from the underlying database! at com.mchange.v2.sql.SqlUtils.toSQLException(SqlUtils.java:106)`**
- **`com.mchange.v2.resourcepool.CannotAcquireResourceException: A ResourcePool could not acquire a resource from its primary factory or source.`**

## Optional: restoring BlackBerry Dynamics apps to a new device: discontinue use of old device

If you backup a BlackBerry Dynamics-based application from one device and then restore it to a different device, make sure you remove the copies of the BlackBerry Dynamics-based app from the original device.

> **Note:** On the new device, when you start the restored application, it will be locked. You will need an unlock key from Good Control. See Apps: Wipe, Unlock, Lock, Upload Logs, and More .

On the old device, if you attempt to start the old application, the application will be wiped. Consider wiping the old device via Good Control; see Device Management Operational Tasks: Device Status, Lock, Clear Password, Wipe, and Deactivate .

This recommendation is based on several reasons:

- Using both devices after backup of one and restore to another is not supported.
- The old copies are no longer necessary because all data is now on the new device.
- Leaving the old data on a device you no longer use is not good security practice.

# Device management

## MDM not available for new installations of Good Control

Starting with Good Control 3.x (this release), the mobile device management (MDM) service is no longer available for new installations of Good Control. You are encouraged to move to the BlackBerry Unified Endpoint Manager, which has complete support for mobile device management. See http://global.blackberry.com/en/enterprise/unified-endpoint-management.html

# Create Google Cloud Messaging API keys

These are the details for obtaining keys for the Google Cloud Messaging (GCM) API, which BlackBerry Enterprise Mobility Server uses to send new mail notifications to Android devices. For more information about creating the Google Cloud Messaging API Keys, visit goodpkb.force.com/PublicKnowledgeBase to read article 21187.

## Prerequisites

You must have a Google account. Avoid using your personal account.

## Steps

After getting the API key from Google, you will enter its name and the value of the key into the GEMS Dashboard.

1. In a browser, open https://console.firebase.google.com/ and log in with a valid account.
2. Click **CREATE NEW PROJECT**.
3. In the **Create a project** dialog box, type a project name and select the Country/region you are located in.
4. Click **Create Project**.
5. In the upper left-hand side of the screen, click **Settings** icon.
6. Click **Project settings**.
7. Click **CLOUD MESSAGING**.
8. Copy the value of the **Server key**. The Server key is used as the GCM API Key value in the BlackBerry Enterprise Mobility ServerDashboard
9. Copy the value of the **Sender ID**. The Sender ID is used as the GCM Sender ID value in the BlackBerry Enterprise Mobility ServerDashboard.

## Installing Google Cloud Messaging API Keys

**To enter Google Cloud Messaging API Key details, in Good Control:**

1. In Good Control, **Device Management** > **Android** tab.
2. Under **Google Cloud Messaging**, click **Edit**.
3. For the **Sender ID** field, enter the value of name you specified for the name of the Server Key you created in Google, as detailed in Create Google Cloud Messaging API keys .
4. For the **Key** field enter the value of your API key from Google.
5. Click **Save** to store the values or **Cancel** to discard them.

# Working with APNS certificates

Apple Push Notification Service (APNS) certificates are needed to secure the communications between the system and end-users' iOS devices.

> **Note:** Before you work with APNS certificates, you need to have an account on the Apple Push Certificates Portal at https://identity.apple.com/pushcert/ .

In Good Control's **Device Management** > **iOS** tab, you store certificates needed for communication with end-users' iOS devices. The general process is as follows:

1. Generate a Certificate Signing Request (CSR) to load into to the Apple Push Certificates Portal to obtain your APNS certificates.
2. Upload APNS certificates after you receive them from Apple.

## Generating a CSR

The Certificate Signing Requests (CSRs) from GC are digitally signed by BlackBerry.

**To download a CSR to supply to Apple, Inc.:**

1. Navigate to **Device Management** > **iOS** tab.
2. Click **Generate CSR**.
3. Note the location of and name of the CSR file on your local machine.
4. Log in to your account on Apple's APNS server.
5. Upload the CSR you generated from Good Control.
6. Download the returned certificate from Apple.

## Uploading an APNS Certificate

After you receive from Apple your certificate for use with APNS, upload it on the **Certificates** > **APNS** screen.

**To upload an APNS certificate:**

1. Navigate to **Device Management** > **iOS** tab.
2. On the far right, click **Upload**.
3. Click **Browse** to navigate to and open the desired certificate file that you received from Apple on your local computer.
4. Click **Upload**.

Results of the upload are displayed.

# Renew APNS Certificates Before Expiration

You should renew your APNS certificates before they expire.

Otherwise, with an expired certificate, Apple stops sending notifications to enrolled devices.

# Device Policies

Device policies are created with the **Device Policies** screen and added to policy sets with the **Policy Sets** > **Device Management** tab.

- Creating, editing, and deleting device policies is detailed in Working with Device Policies .
- Adding devices policies to policy sets is detailed in Adding Device Policies to Policy Sets.

## Good Control properties for allowable-new-device platforms

The following server properties in Good Control enable or disable new devices of the indicated platform.

By default, new devices are allowed.

**To set properties in Good Control:**

1. Navigate to **Servers** > **Settings** tab.
2. Find the desired property.
3. Set the property.
4. Click **Save** to retain your changes or **Cancel** to discard them.

| Property | Description |
|---|---|
| allow.new.android.device | Android |
| allow.new.iOS.device | iOS |
| allow.new.Windows.device | All Windows devices other than Windows Phone, such as Windows tablet |
| allow.new.WindowsPhone.device | Windows Phone |

## Enabling device management in Good Control

If Device Management has not been enabled in your Good Control server, you cannot see the user interface related to it.

**To enable Device Management in Good Control:**

1. Navigate to **Servers** > **Server Properties** tab.
2. Scroll to find the property **`gc.mdm.enabled`**.
3. Check the property's checkbox.
4. In the upper left, click **Submit**.
5. Click **OK** to the acknowledgment that the properties have been updated.

6. Log out of Good Control.
7. Allow approximately 30 seconds to pass while the property change takes effect.

After you log back in, you will see additional DM-related entries in the navigation and elsewhere, as documented in these topics.

## Working with Device Policies

For background, see Policies .

**To create a new device policy:**

1. Go to **Device Policies**.
2. On the far right, click **New Device Policy**.
3. Enter a name for the policy.
4. Enter its description.
5. If you want to base this policy on an existing one, from the **Copy From:** menu, select the name of the policy to copy from.
6. Click **OK** to create the policy or **Cancel** to discard it.
7. Continue with editing the policy to set the desired restrictions.

**To edit an existing device policy:**

1. Go to **Device Policies**.
2. Scroll in the list to find the desired policy.
3. On the far right of the line for the policy, click the pencil icon to edit it.
4. Click one of the following tabs, depending on what you want to do, and make the desired settings.

| Tab | Description |
|---|---|
| General | High-level Device Management device features |
| Password | Allowable characters, length, and more relating to device passwords |
| Restrictions | The heart of device policies, subdivided by iOS and Android sections. Specific device features to restrict. |
| Assign Configurations | Associate this policy with a particular kind of network access: VPN, WiFi, Webclip, and others. For creating device configurations, see Creating, Editing, and Deleting Device Configurations . |

Save the changes.

**To delete a device policy:**

1. Go to **Device Policies**.
2. Scroll in the list of policies to find the ones you want to delete.

3. On the left, check the checkbox for each policy you want to delete.
4. In the upper right, click **Delete**.
5. Click **OK** to confirm that you want to delete the specified policies or **Cancel** to leave them intact.

## Windows Tablet device management: known limitations

For managing Windows tablets, BlackBerry device management services rely on Microsoft's Windows 8.1 operating system, the Windows Push Notification Service (WNS), and other Microsoft software discussed below.

Described here is some of the behavior of BlackBerry device management of Windows tablets because of this reliance on Microsoft.

### End-user unenrollment cannot be detected

The Windows implementation of the Open Mobile Alliance (OMA) Device Management client does not send meaningful information to the BlackBerry device management service when an end-user unenrolls from BlackBerry device management. In this case, BlackBerry device management services record that the end-user device is still enrolled, although it might not be.

### Scheduled maintenance works only on Surface Pro tablets

Windows' scheduled maintenance feature is supposed to automatically check with the BlackBerry device management service for any new policies or other configuration updates. However, with Windows 8.1 operating system, scheduled maintenance works correctly only on Surface Pro tablets, not other tablet models.

To work around this limitation to communicate with other tablet models, BlackBerry device management relies on Microsoft's Windows Push Notification Service (WNS).

### WNS channel URI errors can cause unenrollment

BlackBerry device management depends on Microsoft's Windows Push Notification Service (WNS) to communicate with enrolled devices, for policy and other updates.

In the unlikely case that Microsoft's WNS servers return an error, BlackBerry device management cannot communicate with the devices. In this circumstance BlackBerry device management unenrolls the device, which is reported in Good Control.

### About the Windows update field in device status in Good Control

On end-users' Windows devices, the Windows operating system's update feature has four different settings:

1. Scheduled
2. Choose
3. Auto
4. Disabled

However, for device management status in Good Control, the Windows operating system does not return the "Scheduled" value to BlackBerry device management. BlackBerry device management treats the "Scheduled" and "Choose" values as equivalent. For "Scheduled", the **Windows Update** field in GC's device status shows **Choose**.

## Behavior of password restrictions on Windows Tablet

The behavior of password restrictions on Windows tablet devices varies from other platforms. A key distinction is whether the device is enrolled by a Microsoft account (one created on a Microsoft service) or an account that is local to the device (called a *local account*).

| Device Setting in Good Control | Microsoft Account on Windows Tablet | Local Account on Windows Tablet |
|---|---|---|
| Require a password | A password is always required. | A password must have been set on the device prior to enrollment.<br>After a password has been set, it cannot be removed or changed. |
| Quality | Windows dos not support the concept of password quality. | Windows dos not support the concept of password quality. |
| • Minimum password contains...<br>• Minimum password length | Allow from 4 to 16 characters | • Allow up to 14 characters<br>• Cannot be set less restrictive.<br>• After length has been set, it cannot be removed or changed on the device. |
| Password expiration | Not applicable | • Allow from zero to 731 days.<br>• Cannot be set less restrictive.<br>• After expiration period is set it cannot be removed or changed on the device. |
| Prevent reuse of last password (password history) | Not applicable | Allow from zero to 24 unique passwords<br>Cannot be set less restrictive.<br>Once enforced on the device, the setting cannot be removed or changed. |
| Device lockout (maximum number of allowed failed attempts) | Allow from four to 10<br>Once set, cannot be made less restrictive.<br>If device does not have encryption enabled, user must restart device.<br>If device has encryption enabled, locked-out user has two options:<br>• Factory-reset the device | Allow from four to 10<br>Once set, cannot be made less restrictive.<br>Locked out device is restarted. |

| Device Setting in Good Control | Microsoft Account on Windows Tablet | Local Account on Windows Tablet |
|---|---|---|
| | - Provide lockout code supplied by Microsoft | |
| Screen locks after X minutes of inactivity (also called inactivity timeout | One to 120 minutes<br>Once set, cannot be made less restrictive. | One to 120 minutes<br>Once set, cannot be made less restrictive. |
| Complex combinations of characters cannot be managed be cause they are not displayed in the GC console. | | |
| Disallow convenience logon is set OFF and cannot be managed via the GC console. | | |

#### Effect of "Reset Security Policies"

The end-user can manually remove them with the “Reset Security Policies” option on the Windows tablet. If the end-user initiates “Reset Security Policies,” the password restrictions are not enforced on the Local Account and the password can be removed.

#### after unenrollment, password restrictions still enforced

After device management deactivation (unenrollment) all password restrictions are still present and enforced on the device. They can be removed with the “Reset Security Policies” option on the device. See Effect of "Reset Security Policies" .

## Enrolling Devices: Administrator's Tasks

The administrator's tasks for enrolling end-users in mobile device management are detailed here.

### Planning: Corporate-Owned Enrollment or End-User Self-Enrollment?

Decide whether you will enroll your end-users' devices ("Corporate-owned" enrollment) or end-users will self-enroll.

In the Good Control interface, these two types of enrollment are distinguished by two different buttons on the **Users and Groups** screen.

| Type of Enrollment | Corporate-Owned | End-User Self-Enroll |
|---|---|---|
| Button Text | **New Device Enrollment Key** | **New Access Key** |
| Result | Displays enrollment URL and device enrollment key directly on the GC | By default, sends application activation information in email to end-user. |

| Type of Enrollment | Corporate-Owned | End-User Self-Enroll |
|---|---|---|
| | screen. | **Note:** Enrollment in device management occurs only if the related policy set contains at least one device policy; otherwise, only application activation occurs. |

**Prerequisites**

1. All end-users whose devices are to be enrolled have been added to Good Control.
2. Device and application policies have been defined in Good Control:
   - Be sure you have at least one device policy in your policy sets that matches the OSs or form factors (tablet, phone) of your end-users' devices; otherwise, enrollment in mobile device management does not occur.
   - In your application policies, you have granted users access to the necessary applications:
     - For enrollment on iOS, access to at least one GD-SDK-based application.
     - For device enrollment on Android, access to Good Agent.
     - For Windows devices, no application is needed.
3. Policy sets including device policies and application policies created in Good Control.
4. Policy sets applied to users or application groups in Good Control.
5. Necessary software installed on end-users' devices:
   - On iOS, Good Agent for iOS, which you have given the users access to.
   - On Android, Good Agent for Android, which you have given the users access to.
   - For Windows devices, no application is needed.

## Admin Steps for Corporate-Owned Enrollment

For each end-user device, follow these steps:

1. All prerequisites described above are ready.
2. In Good Control, go to **Users and Groups**.

3. Check the checkbox associated with the end-user whose devices you want to enroll in device management.
4. Click **Edit**.
5. Click the **Keys** tab.
6. Click **New Device Enrollment Key**.

## iOS

1. With the end-user's device, open Safari.
2. Enter the URL displayed on the screen in Good Control.
3. In the displayed fields, enter the end-user's email address and device enrollment key.
4. Follow the leading prompts to install the profile presented to you and allow the enrollment to complete.

When the Device Management profile has been successfully installed, enrollment is complete.

## Android

1. With the end-user's Android device, open Good Agent.
2. Do *not* tap **Next**.
3. At the bottom of the displayed screen, tap the label **Corporate-Owned Signup**.
4. In the displayed fields, enter the end-user's email address and the device enrollment key.
5. Tap **Done**.
6. Follow the leading prompts and allow the enrollment to complete.

After enrollment, you are prompted to activate the Good Agent application.

1. In Good Control, click **New Access Key**.
2. In the prompts in Good Agent, enter the user's email address and access key.
3. Follow the leading prompts to complete the activation.

After activation is complete, Device Management enrollment is also complete.

## Windows Tablet and Windows Pro

**Important:** Before beginning, in the **Action Center** slide the user settings to lower than **Always Notify**. If **Always Notify** is in effect, many of the fields detailed below do not appear on the device.

1. With the end-user's device, Navigate to **Settings** > **Workplace Settings**.
2. In the **User ID** field, enter the email address of the end-user whose device you are enrolling.
3. Turn off **Automatically detect server address**.
4. In the **Server Address** field, enter the following case-sensitive URL: `https://bxenroll.good.com/`
5. Tap **Turn on**.
6. In the displayed field showing **Device Token**, enter the device enrollment key from Good Control.
7. Tap **Enroll**.

8. Tap **I agree**.
9. Tap **Turn on**.

When the **Turn on** control changes to **Turn off**, enrollment is complete.

## Windows Phone 8.1

1. With the end-user's device, Navigate to **Settings** > **Workplace**.
2. Tap **Add account**.
3. Enter the email address of the end-user whose device you are enrolling.
4. Tap **Sign in**.
5. Turn off **Automatically detect server address**.
6. In the **Server Address** field, enter the following case-sensitive string. Do *not* enter a leading https:// or a trailing :443: `bxenroll.good.com`
7. Tap **Sign in**.
8. In the displayed field under the heading **Device Activation**, enter the device enrollment key from Good Control.

   **Note:** Click to move through the fields of the key. (The cursor is not automatically advanced.)

9. Tap **Enroll**.

   The enrollment process moves through a series of screens and then displays done.

10. Tap **done**.

When you see that the device is under control of GOODMDM, enrollment is complete.

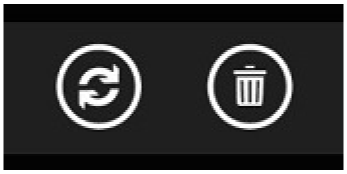
### Viewing Device Management Details on WIndows Phone 8.1

To see the status of device management on a Windows Phone 8.1 device:

1. Navigate to **Settings** > **Workplace**.
2. Tap **GOODMDM**.

The screen displays the name of the user, the Device Management server, and the time of the last policy push from Good DM.

The controls at the bottom:



- Tap the control on the left to force retrieval of policies from Good Control.
- The control on the right unenrolls the device from device management, but this ability is controlled by device policy itself, so the control might not be active.

## Configuring compliance emails

When end-users' device become out compliance with the policies you set, the system can send email to the end-users to advise them of the non-compliant devices

**Important:** Sending compliance emails is not enabled by default. Adding a value for the property **`mdm.compliance.email.admin`** (the administrator's email address) enables compliance emails

Compliance emails are controlled by properties you set on Good Control's **Servers**> **Server Properties** tab. Except for **`mdm.compliance.email.admin`** all properties are templatized and include variables that are populated when email is sent.

| Property | Meaning |
|---|---|
| mdm.compliance.admin.email | Email address of the Good Control administrator in standard Internet email address format, like **`someone@somewhere.com`**. |
| mdm.compliance.email.body | Body of the email message.<br>**Note:** Do not change the variable names embedded in the template. |
| mdm.compliance.email.sender | Display name of sender, like "BlackBerry Mobile Administrator". |
| mdm.compliance.email.subject | Subject line of non-compliance email.<br>**Important:** Do not change the variable names embedded in the template. |

## Device Management Operational Tasks: Device Status, Lock, Clear Password, Wipe, and Deactivate

You can manage end users' device from two general locations in Good Control:

- For devices that are not under control of Apple's DEP, go to the individual end user's information as detailed below.
- For devices under control of Apple's DEP, go to the Apple DEP Devices menu, as described in Working with DEP-Enrolled Devices .

You can see the status of end-users' devices, and you can manage the end-user's device with the buttons described here.

### Actions on Non-Apple DEP Devices

The status details are updated from the Good device management service to Good Control every 60 minutes.

1. In Good Control, navigate to **Users and Groups**.
2. Check the checkbox associated with the end-user whose devices you want to manage.
3. Click **Edit**.
4. Click the **Devices and Apps** tab.
5. Scroll to find the desired end-user's device.

6. Choose the operation you want from the **Device Actions** pulldown:
   - Lock Device
   - Clear Device Password: for iOS only.
   - Ring Device: for Windows Phone 8.1 only.
   - Wipe Device
   - Deactivate Device
   - Installed Apps

   Auto-pushed apps that have been deleted from the GD NOC are not displayed here. See DIsplay of Bundle ID Only: App Removed from GD NOC.
7. Follow the leading prompts to complete the chosen task.

## Reports: Devices and App Inventory

See the following:

- Device Management App Inventory Reports
- Device Management Inventory Reports

## Unenrolling a Device from MDM

As administrator, you can unenroll previously enrolled end-users' devices from MDM.

1. In Good Control, navigate to **Users and Groups**.
2. Check the checkbox associated with the end-user whose devices you want to unenroll from device management.
3. Click **Edit**.
4. Click the **Devices and Apps** tab.
5. On the far right, click **Deactivate Device**.
6. Follow the leading prompts to complete the unenrollment.

# Device policy reference

Included here are the settings that can be configured for device policies. You can use these lists to help plan the device policies you need.

Device policies are organized into several sections:

- General
- Password: Strictness, format, length, and other characteristics of device passwords
- Restrictions: Specific device features that can be managed, grouped by operating system
- **Add Device Configurations**: To associate device policies with previously defined device configurations.

## Disabling US Government notice and consent form

Samsung enforces the U.S. Federal Government's requirement to display a notice and consent form to end-users whenever U.S. government sites or data are accessed.

Samsung enables this notice by default, which might not be desirable outside the USA.

BlackBerry device management includes a device policy setting to disable it.

**To enable or disable the U.S. Government notice and consent device policy, in Good Control:**

1. Navigate to **Device Policies** > *edit a policy* > **Restrictions** tab.
2. Scroll to find **KNOX Standard (SAFE) Restrictions**.
3. Click **Edit**.
4. Scroll again to find **Disable Notice and Consent**.
5. Click the **OFF** radio button.
6. Click **Save** to save your change or **Cancel** to discard it.

## Device policy reference: general

These are the general settings that can be configured.

> **Note:** Always consult the GC **Device Policies** > **General** tab for the latest list of restrictions.

### BlackBerry for KNOX

> **Note:** BlackBerry for KNOX settings are independent from the KNOX Safe restrictions listed in Device policy reference: restrictions .

- BlackBerry for KNOX Enabled
- Attestation trigger

  Periodically every X hours

### Device access controls

You must set at least one of these access control policies.

**Note:** For your initial policy for use with Apple DEP device, be sure that you enable all these settings.

- MDM Enabled: In order for device configurations to be sent to devices, this setting must be ON.
  - Allow device erase
  - Allow inventory of personal apps
  - Check compliance against:

    Black List / White List
  - Allow query of Device Information (serial number, IMEI, etc) (iOS)
  - Allow query of Network information (carrier network, phone number, etc) (iOS)
  - Allow device lock and passcode removal (iOS)
  - Allow password-related queries
  - Allow restriction-related queries
  - Allow remote app installation/updates
  - Allow inspection of installed configuration profiles (iOS)
  - Allow installation and removal of configuration profiles (iOS
  - Allow inspection of installed provisioning profiles (iOS)
  - Allow installation and removal of provisioning profiles (iOS)
  - Allow manipulation of settings (iOS)

## Device policy reference: passwords

These are settings for device passwords that can be configured in device policies.

**Note:** Always consult the GC **Device Policies** > **Passwords** tab for the latest list of restrictions.

### Require a password and Quality

If a password is required (default), the other settings appear.

The number of settable characteristics of passwords changes depending on your choice for password **Quality**:

- **Simple**
- **Alphanmueric**
- **Complex**

**Note:** On Windows tablet devices, password restrictions have significantly differing behavior. See Password restrictions on Windows Tablet .

#### Quality simple

- Minimum password contains X characters
- Password expiration in X days

- Prevent users from reusing the last X unique passwords
- Device wipes out after X failed attempts
- Screen lock after X minutes of inactivity
- Maximum grace period of X minutes for screen lock (iOS)
- MaximumSequential Characters (BlackBerry for KNOX)
- MinimumChanged Characters (BlackBerry for KNOX)
- Simple password type (Android) Any|Numeric|Alphabetic

#### Quality alphanumeric

Same as Simple, without "Simple Password Type (Android)".

- Minimum password contains X characters.
- Password expiration in X days
- Prevent users from reusing the last X unique passwords
- Device wipes out after X failed attempts
- Screen lock after X minutes of inactivity. X is from 0 to 29, except for iPad, which allows either 2 minutes or 5 minutes.
- Maximum grace period of X minutes for screen lock (iOS)
- Maximum X Sequential Characters (BlackBerry for KNOX)
- Minimum X Changed Characters (BlackBerry for KNOX)

#### Quality complex

- Minimum X Symbols Required
- Minimum X Digits Required (Android)
- Minimum X Lower Case Letters Required (Android)
- Minimum X Upper Case Letters Required (Android)
- Minimum X Letters Required (Android)
- Minimum X non-Letters Required (Android)

#### Password restrictions on Windows Tablet

See Windows Tablet device management: known limitations for details on the behavior of password policies and other limitations.

### Device policy reference: restrictions

This is a list of the settable device restrictions for iOS, Android, Samsung KNOX Standard (SAFE), and Windows.

> **Note:** Always consult the GC **Device Policies** > **Restrictions** tab for the latest list of restrictions.

In these lists:

- ✓ indicates that the restriction is enabled by default,
- — indicates that the restriction is disabled by default.

### iOS restrictable features

In these lists:

- ✓ indicates that the restriction is enabled by default,
- — indicates that the restriction is disabled by default.

### Functionality

✓ Allow use of camera

✓ Allow Facetime

✓ Allow screenshots and screen recording (iOS9+)

✓ Allow Voice dialing

✓ Allow Siri (iOS 5+)

✓ Allow Siri while device is locked (iOS 5.1+)

— Enable Siri profanity filter

✓ Allow installing apps (including Apple Configurator and iTunes)

✓ Allow In-App Purchase

— Require iTunes Store password for all purchases

✓ Allow iCloud backup

✓ Allow iCloud documents & data

✓ Allow iCloud keychain (iOS 7+)

✓ Allow iCloud Photo Library (iOS 9+)

✓ Allow My Photo Stream

✓ Allow Shared Stream

✓ Allow managed apps to store data in iCloud (iOS 8+)

✓ Allow backup of enterprise books (iOS 8+)

✓ Allow notes and highlights sync for enterprise books (iOS 8+)

✓ Allow automatic sync while roaming

— Force encrypted backups

— Force limited ad tracking

✓ Allow Internet results in Spotlight (iOS 8+)

✓ Allow automatic updates to certificate trust settings (iOS 7+)

✓ Allow documents from unmanaged apps in managed apps (iOS 7+)

✓ Allow documents from managed apps in unmanaged apps (iOS 7+)

✓ Treat AirDrop as unmanaged destination (iOS 9+)

✓ Allow untrusted TLS prompt

✓ Allow sending diagnostic data to Apple (iOS 6+)

✓ Allow Touch ID to unlock device (iOS 7+)

✓ Allow HandOff (iOS 8+)

— Require pairing password on incoming AirPlay requests

— Require pairing password on outgoing AirPlay requests

✓ Allow Passbook notifications while locked (iOS 6+)

✓ Show Control Center in lock screen (iOS 7+)

✓ Show Notifications Center in lock screen (iOS 7+)

✓ Show Today View in lock screen (iOS 7+)

### Apps

✓ Allow use of YouTube (iOS 6 and below)

✓ Allow use of iTunes Store

✓ Allow adding Game Center friends

✓ Allow multiplayer gaming

✓ Allow Safari

✓ Enable autofill

✓ Enable JavaScript

— Block pop-ups

— Force fraud warning

Accept Cookies: Always

✓ Trust new enterprise app authors (iOS 9+)

### Media content

Allowed content ratings

Ratings Region: US

**Movies**

Allow All Movies

**TV Shows**

Allow All TV Shows

**Apps**

Allow All Apps

✓ Allow playback of explicit music, podcasts & iTunes U media

✓ Allow explicit sexual content in iBooks Store (iOS 6+)

Apple Watch

— Force Apple Watch wrist detection (iOS 8+)

Supervised mode

## General

✓ Allow AirDrop

✓ Allow iMessage

✓ Show user-generated content in Siri

✓ Allow iBooks store

✓ Allow erase all content and settings

✓ Allow modifying restrictions

✓ Allow installing configuration profiles

✓ Allow modifying account settings

✓ Allow modifying cellular data app settings

✓ Allow modifying Find My Friends settings

✓ Allow pairing with non-Configurator hosts

✓ Allow Define

✓ Allow modifying device passcode (iOS 9+)

✓ Allow modifying Touch ID fingerprints

✓ Allow modifying device name (iOS 9+)

✓ Allow modifying Wallpaper (iOS 9+)

## Keyboard

✓ Allow predictive keyboard

✓ Allow auto correction

✓ Allow spell check

✓ Allow keyboard shortcuts (iOS 9+)

## Apps

✓ Allow installing apps using App Store

✓ Allow Automatic App Downloads (iOS 9+)

✓ Allow removing apps

✓ Allow use of Podcasts

✓ Allow use of Game Center

✓ Allow use of Apple News (iOS 9+)

## Apple Watch

✓ Allow pairing with Apple Watch (iOS 9+)

### Android restrictable features

In these lists:

- ✓ indicates that the restriction is enabled by default,
- — indicates that the restriction is disabled by default.

— Disable camera

— Encrypt internal storage

### KNOX standard (safe) restrictable features

The KNOX Standard (SAFE) restrictions here are independent from the settings for BlackBerry For KNOX listed in Device policy reference: general .

In these lists:

- ✓ indicates that the restriction is enabled by default,
- — indicates that the restriction is disabled by default.

#### General restrictions

— Encrypt SD Card

— Disable SMS

— Disable MMS

— Disable SD Card

— Disable NFC

— Disable Android Beam

— Disable Cellular data

— Disable Lock Screen Widgets

— Disable Factory Reset

— Disable Native Browser

— Disable lock screen shortcuts

— Notice and Consent Banner

### Location & roaming restrictions

— Disable Roaming Data

— Disable Roaming Sync

— Disable Roaming VoiceCalls

### Capture restrictions

— Disable SVoice

— Disable Screen Capture

### WiFi restrictions

— Disable WiFi

— Disable WiFi Auto Connect

### Bluetooth restrictions

— Disable Bluetooth

## Software & update restrictions

— Disable Google Play Store

— Disable Non-Market apps

— Disable OTAOS Update

### USB & tethering restrictions

✓ Disable USB Debugging

— Disable USB Media Player (MTP — also controls USB MS and USB KIES)

— Disable USB Host Storage

— Disable Bluetooth Tethering

— Disable USB Tethering

— Disable WiFi Tethering

### KNOX premium

— Enable Common Criteria Mode (Requires BlackBerry for KNOX)

### About enabling Common Criteria mode

This description is based on documentation from Samsung.

An administrator can enable Common Criteria configuration on a device. When enabled, the following are the effects:

- The bootloader blocks KIES download mode and enforces a check of integrity of the kernel and of the self-test crypto modules.

- The device will verify the additional signature on FOTA ("firmware over-the-air") update using a RSA-PSS signature
- The device will enforce the use of the FIPS 140-2 validated crypto module for EAP-TLS Wi-Fi connections. (For more information about WiFi device configuration in BlackBerry device management, see Wi-Fi configuration .)

To fully enable Common Criteria-evaluated configuration, the following should also be enforced:

1. Enable Device Encryption
2. Enable SD Card Encryption
3. Set Attempts before Wipe.
4. Enable Certificate Revocation (since KNOX 2.2)
5. Disable Password History (since KNOX 2.2)

## Update: Windows device management restrictions

The following are the most recent restrictions for device management of Windows.

### Windows restrictions supported by all Windows OS versions

- Disable Data While Roaming

### Windows Phone 8.1, Windows Phone 10 and Windows Tablet 10 restrictions

- Disable Development Unlock
- Require Device Encryption
- Disable Removable Storage Card
- Disable MDM un-enrollment
- Disable Camera
- Disable Bluetooth
- Disable Wi-Fi
- Disable Location Services
- Disable Microsoft Account Connection
- Disable Custom Email Accounts
- Disable Cortana
- Disable Internet Sharing
- Disable VPN While Roaming
- Disable VPN Over Cellular

### Windows Tablet/Desktop 8.1 restrictions

- Allow Diagnostic Data Submission
- Require SmartScreen in Internet Explorer
- User Account Control
- Microsoft Account Optional to use Modern Applications (Windows 8.1)

### Windows Phone 8.1 and Windows Phone 10 restrictions

- Disable MDM software and hardware factory reset
- Disable NFC
- Disable Microsoft Store
- Disable Copy/Paste
- Disable Share Office File (Windows 8.1 only)
- Disable Save As Office File (Windows 8.1 only)
- Disable Screen Capture
- Disable MTP and IPoUSB
- Disable Manual Installation of Root and Intermediate CAP Certificates
- Disable Manual Wi-Fi Configuration
- Disable Wi-Fi Hotspot Reporting to Microsoft
- Disable Action Center Notifications Above Lock Screen
- Disable Voice Recording
- Disable Browser

#### Windows laptop devices not supported

Device management does not support Windows laptop devices.

If you inadvertently apply device management to a Windows laptop, the device management profile will be installed. In this case, you should deactivate the device to remove the unneeded profile.

#### PPTP VPN not supported for iOS 10

Device management does not support Point-to-Point-Tunneling Protocol (PPTP) VPNs on iOS 10 devices.

## MDM properties for GC 2.x

**Note:** These properties are only available in Good Control 2.x, not the latest versions

| Property | Description | Default, Global, Restart |
|---|---|---|
| gc.mdm.enabled | Enable or disable Good device management | Default: false<br>Global: yes<br>Restart: no |
| MDM Admin Email Address | Email address of device management administrator | Default: none<br>Global: yes<br>Restart: no |
| mdm.compliance.admin.email | Email address for sending out-of-compliance emails | No default |

| Property | Description | Default, Global, Restart |
|---|---|---|
| | | Global: yes<br>Restart: no |
| MDM Out-of-Compliance Email Template<br>mdm.compliance.email.body | Body of out-of-compliance emails | • Default: see text below.<br>• Global: yes<br>• Restart: no<br>• Text:<br>Dear Administrator,<br><%DISPLAY_NAME%>'s <%DEVICE_MODEL%> is out of compliance.<br>Type of Compliance Failure: <%COMPLIANCE_TYPE%><br>Reason for Compliance Failure: <%FAILURE_REASON%><br>Thank you,<br>Good Control |
| mdm.android.agent | Name of Device Management client for Android | Default: com.good.android.gdagent<br>Global: yes<br>Restart: no |
| mdm.compliance.email.sender | Email address of sender of out-of-compliance emails | Default: Good Mobile Administrator |
| mdm.compliance.email.subject | Subject line of out-of-compliance emails | Default: [Out Of Compliance] <%DISPLAY_NAME%>'s <%DEVICE_MODEL%><br>Global: yes<br>Restart: no |
| mdm.enrollment.email.enabled | Enable email of device management enrollment | Default: true<br>Global: yes<br>Restart: no |
| mdm.ios.agent | GD Entitlement ID of Good Agent for iOS | Default: com.good.ios.gdagent<br>Global: yes<br>Restart: no |
| mdm.server.url | URL of the Good MDM server | Default: https://bxenroll.good.com<br>Global: yes<br>Restart: yes |

# Device configurations

In order for device configurations to be sent to enrolled devices, the setting **MDM Enabled** must be ON (which is default). See Device policy reference: general for a list of general policies, including MDM Enabled.

## About Active Directory and "auto-fill username"

BlackBerry device management reads information from the Active Directory service that was associated with Good Control at installation.

Some of the device configurations have the option to "auto-fill username". The behavior of this field varies by platform.

### iOS ActiveSync and autofill username

In the ActiveSync for iOS device configurations, the **Autofill Username** field is set by default and cannot be unchecked.

### Android and autofill username

On Android, this field is not populated for non-Active Directory users.

This setting can sometimes result in improper user names on iOS devices that should be corrected by end-users so that data from Active Directory can be synchronized correctly.

```
%login%
```

The end-user should change this value to his own correct Active Directory username.

## VPN configuration

This section contains settings which configure the Virtual Private Network (VPN), which protects the network connections between devices and their corporate servers.

1. Navigate to **Device Configurations** > **VPN** tab.
2. On the right, click **Add VPN Configuration** and select **Android** or **iOS**.
3. Complete the platform-specific fields described in the remaining sections, by **Connection Type**:

   For iOS only: Layer 2 Tunneling Protocol (L2TP) fields
   For iOS only: Point to Point Tunneling Protocol (pptp) fields
   For iOS only: Cisco IPsec
   Android is supported only for Cisco AnyConnect .

4. Click **Save** to keep your changes or **Cancel** to discard them.

The following sections describe the inputs required for each of the VPN connection types.

### For iOS only: Layer 2 Tunneling Protocol (L2TP) fields

The following table describes the fields for the VPN connection type L2TP.

| Setting | Description |
|---|---|
| Connection Name | A descriptive name for the connection |
| Connection Type | Select **L2TPConfig**. |
| Server | Enter the fully qualified domain name of your VPN server (e.g. secure.mycompany.com). |
| Auto-fill Username | Check this field to have the user's name filled automatically from your Active Directory service |
| User Authentication | Select from:<br>- **Password**<br>- **RSAToken**: The RSA SecurID authentication mechanism assigns a "soft token" to a device which generates an authentication code at fixed intervals. |
| Shared Secret | A pre-shared key for authentication that the VPN must receive before requesting username and password credentials. Must not exceed 100 characters in length |
| Send all traffic | Check this field if you want all network traffic to go over the VPN connection regardless of the user's network services (such as WiFi or other connections in addition to VPN). |
| Proxy Type | Select from:<br>- **None**<br>- **Automatic:**<br>  - Protocol and fully qualified domain name of the proxy server<br>  - Allow direct connection, if PAC is unreachable<br>- **Manual**<br>  - Proxy Server and Port in `servername:port` format<br>  - Auto-fill Username: Do not use this field reserved for future use. |

## For iOS only: Point to Point Tunneling Protocol (pptp) fields

The following table describes the fields for the VPN connection type PPTP.

| Setting | Description |
|---|---|
| Connection Name | A descriptive name for the connection |
| Connection Type | Select **PPTPConfig**. |
| Server | Enter the fully qualified domain name of your VPN server (e.g. secure.mycompany.com). |
| Auto-fill Username | Check this field to have the user's name filled automatically from your Active Directory service. |
| PPTP Authentication Type | Select from:<br>- **Password**<br>- **RSAToken**: The RSA SecurID authentication mechanism assigns a "soft token" to a |

| Setting | Description |
|---|---|
| | device which generates an authentication code at fixed intervals. |
| Encryption Level | Select from:<br>• **None**: Not recommended. Non-encrypted PPTP connections send the PPP frame in plain text and are not secure.<br>• **Auto**<br>• **Maximum**: 128-bit encryption |
| Send all traffic | Check this field if you want all network traffic to go over the VPN connection regardless of the user's network services (such as, WiFi or other connections in addition to VPN). |
| Proxy Type | Select from:<br>• **None**<br>• **Automatic**<br>• Protocol and fully qualified domain name of the proxy server<br>• Allow direct connection, if PAC is unreachable<br>• **Manual**<br>• Proxy Server and Port in `servername:port` format<br>• Auto-fill Username: Do not use this field reserved for future use. |

## For iOS only: Cisco IPsec

These are the fields for the VPN connection type IPSec (Cisco).

| Setting | Description |
|---|---|
| Connection Name | A descriptive name for the connection |
| Connection Type | Select **IPSec (Cisco)**. |
| Server | Enter the fully qualified domain name of your VPN server (e.g. secure.mycompany.com). |
| Auto-fill Username | Check this field to have the user's name filled automatically from your Active Directory service. |
| Machine Authentication | Select from:<br>• **Shared secret/Group name**<br>• **Group Name**: Enter the user group defined by the BlackBerry Administrator for the Device Users.The name must not exceed 64 alphanumeric characters. The following special characters are permitted: ._-~‘!#$%^&(){}’?<br>• **Shared Secret**: A pre-shared key for authentication that the VPN must receive before requesting username and password credentials. Must not exceed 100 characters in length |

| Setting | Description |
|---|---|
| | • **Use Hybrid Authentication**: An extension of Internet Key Exchange (IKE) over IP Security (IPSec) tunneling protocol. A digital certificate is deployed on the VPN server at the central site, while remote users use SecurID to access the network. The client authenticates the server certificate, and the server authenticates the client's credentials.<br>• **Prompt for Password**: The user is challenged for the password.<br>• **Certificate**:<br>• Click **Upload Certificate** and navigate your computer to select and upload the certificate.<br>• **Password**: Enter the password for the certificate.<br>• **Include User Pin**: [means what?] |
| Send all traffic | Check this field if you want all network traffic to go over the VPN connection regardless of the user's network services (such as WiFi or other connections in addition to VPN). |
| Proxy Type | Select from:<br>• **None**<br>• **Automatic:**<br>• Protocol and fully qualified domain name of the proxy server<br>• Allow direct connection, if PAC is unreachable<br>• **Manual**<br>• Proxy Server and Port in `servername:port` format<br>• Auto-fill Username: Do not use this field reserved for future use. |

## Cisco AnyConnect

Your end-users' devices must have the Cisco AnyConnect application for the appropriate platform:

- Android: Cisco AnyConnect for ICS+ from the Google Play Store.
- iOS: Cisco AnyConnect from the Apple App Store.

### GC fields for Cisco AnyConnect for Android

For Android, your end-user's devices must have the Cisco AnyConnect for ICS+ application from the Google Play Store..

**Note:** Certificate authentication is optional. Some notes:

- Using certificate authentication with Cicso AnyConnect for ICS+ only has relevance if authentication mode is manual.
- After a certificate is installed on the Android device, removing the VPN profile from the device does not remove the certificate, which must also be removed manually.

The following table describes the configuration settings for Cisco AnyConnect for Android.

| Setting | Description |
|---|---|
| Server | Enter the fully qualified domain name of your VPN server (for example, secure.mycompany.com). |
| Certificate Authentication Mode | Select from:<br>- Automatic<br>- Disabled<br>- Manual |
| Certificate | Click **Upload Certificate**, navigate your local computer, select the desired certificate file, and complete the upload. Certificate must be in PKCS12 format. |
| Certificate Password | Enter the password associated with the uploaded certificate file. |

### GC fields for Cisco AnyConnect for iOS

For iOS, your end-user's devices must have Cisco AnyConnect from the Apple App Store.

The following table describes the configuration settings for Cisco AnyConnect for iOS.

| Setting | Description |
|---|---|
| Connection Name | Enter the defined name of the VPN connection. |
| Connection Type | Select **Cisco AnyConnect**. |
| Server | Enter the fully qualified domain name of your VPN server (e.g. secure.mycompany.com). |
| Auto-fill Username | Do not use this field reserved for future use. |
| Group | Do not use this field reserved for future use. |
| User Authentication | Select from:<br>- **Password**<br>- **Certificate** |
| Certificate | Click **Upload Certificate** and follow leading prompts.<br>**Note:** If you do not upload a certificate, authentication mode is set to "Automatic". |
| Password | For certificate authentication, enter the password associated with the uploaded certificate. |
| Send all traffic | Check this field if you want all network traffic to go over the VPN connection regardless of the user's network services (such as WiFi or other connections in addition to VPN). |
| Proxy Type | Select from:<br>- **None**<br>- **Automatic:** |

| Setting | Description |
|---|---|
|  | - Protocol and fully qualified domain name of the proxy server<br>- Allow direct connection, if PAC is unreachable<br>- **Manual**<br>- Proxy Server and Port in `servername:port` format<br>- Auto-fill Username: Do not use this field reserved for future use. |

## Wi-Fi configuration

This section contains settings which configure access of managed devices to the corporate Wi-Fi network connection.

**Important:** The Service Set Identifier (SSID) for a WiFi connection is a unique value by platform, with one configuration each for iOS or Android. BlackBerry device management does not create multiple WiFi configurations for the same SSID.

The SSID can be hidden by selecting or deselecting the **Hidden Network** checkbox. When hidden, the SSID (name) will not be echoed to the display of the managed device and will not be broadcast by the Wi-Fi network. Click the **Hidden Network** check box to prevent the SSID from being broadcast.

Once entered and saved, the SSID will appear inside the parentheses of the displayed name of the configuration set, but it will not appear on the device.

WPA/WPA2 provides stronger encryption than WEP but may not be supported by older routers. For more information, contact your network administrator.

**To create a Wi-Fi configuration that uses the identity certificate, in Good Control:**

1. Navigate to **Device Configurations** > **WiFi** tab.
2. On the right, from the pulldown menu, select **Android** or **iOS**.
3. Complete the platform-specific fields described below.
4. Click **Save** to keep your changes or **Cancel** to discard them.

The following table describes the configuration settings for WiFi for both Android and iOS.

| Setting | Description |
|---|---|
| Service Set Identifier (SSID) | Enter the SSID for the WiFi network. |
| Hidden Network | Check this if you want to disable broadcast of this network's information. |
| Auto Join | Check this is devices are allowed to join the WiFi network automatically. |
| iOS only:<br>Proxy Setup | - **Automatic:**<br>- Protocol and fully qualified domain name of the proxy server |

| Setting | Description |
|---|---|
|  | • Allow direct connection, if PAC is unreachable<br>• **Manual**<br>• Proxy Server and Port in two separate fields<br>• Auto-fill Username: Do not use this field reserved for future use. |
| Security Type | Select from:<br>• **None**<br>• **WEP**<br>• **WPA/WPA2**<br>• **ANY**<br>• **WPA/WPA2 Enterprise**<br>• **EAP**, or Extensible Authentication Protocol:<br>• **TLS**: Transport Layer Security<br>• **TTLS**: Tunneled Transport Layer Security<br>• **PEAP**: Protected Extensible Authentication Protocol<br>• **Inner Authentication**:<br>• **MSCHAPv2**: Microsoft's version 2 of Challenge-Handshake Authentication Protocol<br>• **PAP**: Password Authentication Protocol<br>• **MSCHAP**: Microsoft's version of Challenge-Handshake Authentication Protocol<br>• **GTC**: Generic Token Card<br>• **Auto-fill Username**: Check this field to have the user's name filled automatically from your Active Directory service.<br>• **Certificate**:<br>• Click **Upload Certificate** and navigate your computer to select and upload the certificate.<br>• **Password**: Enter the password for the certificate.<br>• **Outer Identity**: This key is only relevant to TTLS, PEAP, and EAP-FAST. Allows the user to hide his or her identity. It can increase security because an attacker can't see the authenticating user's name in the clear. The user's actual name appears only inside the encrypted tunnel. For example, it could be set to "anonymous" or "anon", or "anon@mycompany.net". |

## Email configuration

This section contains settings which configure the secure connection to the Exchange server or another non-Exchange server with ActiveSync capability. It permits the administrator to set the frequency of synchronization between devices and the mail server and the amount of historical e-mail data that will be kept in sync with the devices.

In non-Exchange environments, the administrator must ensure that users have Windows authentication credentials and that Active Directory is populated with the correct user e-mail addresses. BlackBerry will automatically use the e-mail addresses found in Active Directory to push ActiveSync profiles to the appropriate devices, allowing users to log into the non-Exchange corporate mail server.

### Multiple Exchange configurations on a single device

It is possible for an end-user's device to receive more than one e-mail ActiveSync profile. Conditions are described below:

| Description |
|---|
| When multiple e-mail configurations are defined in a single configuration set, members of assigned groups will receive multiple Exchange profiles. |
| When the Default configuration contains an Exchange configuration and a separate configuration also contains an Exchange configuration, members of groups associated with the second configuration set will receive two Exchange profiles (because every device receives a Default configuration). |
| When multiple Active Directory groups each have Exchange configurations, users who are members of multiple groups will receive multiple Exchange profiles pushed to their devices. |

The following situations can result if a single device is pushed multiple Exchange configuration profiles:

- When two identical Exchange profiles are pushed to the device, the device will reject the second configuration, regardless of the profile name; the device rejects the second profile because it has the same CAS server configuration.
- If a second configuration refers to an alias for the CAS server, iOS does not recognize it as a duplicate, and will accept the second configuration. This will lead to two separate Exchange profiles existing simultaneously on the device, both communicating with the same ActiveSync mailbox configuration. This situation will negatively impact the ability to manage mail delivery.

### Creating an Exchange ActiveSync configuration

**To create an Exchange/ActiveSync configuration, in Good Control:**

1. Navigate to **Device Configurations** > **Email** tab.
2. On the right, click **Add Email**, and select **Android**, **iOS**, or **Windows.**
3. Complete the platform-specific fields described below.
4. Click **Save** to keep your changes or **Cancel** to discard them.

### GC fields for email configuration for Android

The following table describes the configuration settings for Email for Android.

| Setting | Description |
|---|---|
| Account Name | The account name for the Exchange server. |
| Exchange Host | The fully qualified domain name of the Exchange server |
| Exchange Password | The password for logging in to the Exchange host |
| Use SSL | Check this box if you want to use SSL for data communications between your Exchange service and BlackBerry Dynamics servers. |
| Use TLS | Check this box if you want to use TLS for data communications between your Exchange service and BlackBerry Dynamics servers. |
| Auto-fill Username | Check this field to have the user's name filled automatically from your Active Directory service. |
| Server Path Prefix | The IMAP path prefix. With the value **INBOX** in this field, all "peer folders" such as Sent, Drafts, Trash, and Junk are not visible to the end-user, leaving only the Inbox visible. |
| Always Vibrate for Email Notification | Check this box to make the user's device vibrate on receipt of new mail. |
| Vibrate for email notification when silent mode | Check this box to make the user's device vibrate on receipt of new mail even in silent mode. |
| Notification for new email | Allow on-screen notification of new mail |
| • Sync Contacts<br>• Sync Calendar<br>• Sync Tasks<br>• Sync Notes | Check the appropriate box to synchronize the listed feature. |
| Peak Period Sync Schedule | Select from:<br>• Never<br>• Automatic<br>• 5, 10, 15 or 30 minutes<br>• 1, 4, or 12 hours |
| Off-peak Period Sync Schedule | Select from:<br>• Never<br>• Automatic<br>• 5, 10, 15 or 30 minutes<br>• 1, 4, or 12 hours |
| Retrieval Size | Select from: |

| Setting | Description |
|---|---|
| | - **All**<br>- **Headers only** |
| Roaming Sync Schedule | Select from:<br>- **Manual**<br>- **Use Sync Setting** |
| Sync Interval | Select from:<br>- Never<br>- Automatic<br>- 5, 10, 15 or 30 minutes<br>- 1, 4, or 12 hours |
| Past Days of Email to Sync | Select from:<br>- 1 or 3 days<br>- 1 or 3 weeks<br>- 1 month |
| Allow Incoming Attachment | Click this box if you want to allow attachments on incoming email |

## GC fields for email configuration for iOS

The following table describes the configuration settings for email configuration for iOS.

| Setting | Description |
|---|---|
| Account Name | The account name for the Exchange server. |
| Exchange Host | The fully qualified domain name of the Exchange server |
| Use SSL | Check this box if you want to use SSL for data communicated between your Exchange service and BlackBerry Dynamics servers. |
| Past Days of Mail to Sync | Select from:<br>- **No Limit**<br>- **1 day**<br>- 3 days<br>- 1 week<br>- 2 weeks<br>- 1 month |

| Setting | Description |
|---|---|
| Allow messages to be moved | Allow messages to be moved from user account to user account |
| Allow Recent address to be synced | Synchronize the user's "Recent Addresses" list |
| Use only in Mail | Synchronize the mail only for the standard mail client, not third-party mail clients |
| Credentials Password | For certificate authentication, enter the password associated with the uploaded certificate. |
| Send all traffic | Check this field if you want all network traffic to go over the VPN connection regardless of the user's network services (such as WiFi or other connections in addition to VPN). |
| Proxy Type | Select from:<br>• **None**<br>• **Automatic:**<br>  • Protocol and fully qualified domain name of the proxy server<br>  • Allow direct connection, if PAC is unreachable<br>• **Manual**<br>  • Proxy Server and Port in `servername:port` format<br>  • Auto-fill Username: Do not use this field reserved for future use. |

## Webclip

This section contains details on configuring custom webclips.

**To upload a custom profile for iOS devices:**

1. Navigate to **Device Configurations** > **Webclip**.
2. Click **Add WebClip**.
3. Complete the necessary fields, as described below.
4. Click **Save** to preserve your changes your changes or **Cancel** to discard them.

### Webclip fields for iOS

| Field | Description |
|---|---|
| URL | The publicly accessible URL to retrieve the webclip |
| Label | The desired label to associate with the webclip. |
| Icon | Click **Upload** to upload a graphic to associate with this webclip. |
| ON/OFF | Click the desired radio button for: |

| Field | Description |
|---|---|
|  | • Webclip can be removed<br>• Show as full screen<br>• Display without visual effect |

## Custom iOS profile

Here are details on uploading a custom device configuration that you have created with Apple Configurator or similar program. For information about how to export profiles from the Apple Configurator, consult the latest Apple documentation.

Some points:

- If there are multiple profiles, some of which are not managed by Good, only the profile directly associated with a given, specific device policy is applied.
- If there are device policies and a custom iOS profile, the device management service sends both to the device. Apple iOS reconciles them and applies the most restrictive settings.
- A new custom profile can be uploaded at any time, which will be applied to all devices that rely on the associated device configuration.

**Important:** Do not encrypt or sign the configuration profile.

Your configuration file name must end with the `.mobileconfig` file extension.

**To upload a custom profile for iOS devices:**

1. Make sure you have exported your profile from Apple Configurator.
2. Navigate to **Device Configurations** > **Other** tab.
3. Click **Upload File**.
4. Navigate your computer to select the exported custom device configuration.
5. Follow the leading prompts to complete the task.

# Apple DEP Profiles and Devices

Apple Inc.'s Device Enrollment Program (DEP, described at http://www.apple.com/business/dep/) is for businesses to manage their devices via Apple's service. Good Control has an interface to Apple DEP so you can manage all your devices through the single Good Control console.

After prerequisite setup with Apple, the general process for working with DEP profiles and policies in Good Control is as follows:

1. You create as many DEP profiles (collections of DEP policies) as necessary for your organization.

   > **Note:** After you create a DEP profile in GC, it cannot be edited.

2. You apply the DEP profile to the desried devices.
3. You use Good Control to manage the device.

**Prerequisites**

- You must be enrolled in Apple's DEP.
- You must have completed all of Apple's required setup, including your virtual MDM servers.
- You have recorded in Good Control your DEP-related keys and information you received from Apple.
- Your devices must be ready for deployment to your end users. In Apple terminology, your devices have been assigned to your virtual MDM server.

## One-time Setup with Apple for DEP Profiles in Good Control

You need to setup your configuration with Apple in Good Control, including your DEP public key and the MDM server token given to you by Apple, Inc.

### Careful: Effect of Changing the GC-Defined Apple MDM Server Token

Be advised that after you have set up your Apple MDM server token in Good Control, if you change the token in GC (to attempt to map a different MDM server), either in the same DEP account or from a different DEP account, the following occurs.

- Devices that are already enrolled in MDM:
  - Will continue to be managed and available in the device view.
  - Admin can take MDM actions – change device policy, password reset, lock & wipe.
  - Any change in DEP Profile will not be applied until the device is factory-reset.
  - Once unenrolled, the device will no longer to accessible.
- Devices that are not already enrolled in MDM:
  - All device serial numbers that were associated with the old MDM Server will be removed and no longer accessible in the device list view in Good Control.

#### Steps

**To setup Apple DEP service in Good Control:**

1. Navigate to **Device Management**.
2. Click the **iOS** tab.
3. Under **DEP Account**, click **Edit**.
4. Enter a description of your DEP account.
5. Click **Generate DEP Public key**.

6. Click **Download Key** to save the generated key to your local computer.
7. Login to Apple's DEP Portal and upload this public key to create your virtual MDM server.

   Apple's portal will give you an MDM server token to save to your local computer.

8. In Good Control, click **Import MDM Server Token**.
9. Navigate your computer to find the MDM server token you downloaded from Apple.
10. Click **Import** to finish or **Cancel** to stop.
11. Checkmark **Auto-assign to new DEP devices** if you want a certain DEP profile to be assigned automatically to all new devices.
12. From the **DEP Profile** pulldown menu, select the name of the DEP profile you want automatically assigned to new devices.
13. From the **Initial Device Policy** pulldown menu, select the name of the defined device policy you want to apply to all new devices.
14. Click **Save** to save your changes or **Cancel** to discard them.

## Defining DEP Profiles in Good Control

The following settings and device policies can be defined in an Apple DEP profile via Good Control.

With a profile, you define sets of characteristics of device management for Apple devices, essentially relieving the end user of any need to decide. You can indicate which parts of the device initialization can be skipped entirely. These settings are the **Skip Setup Screens** policies.

| Group | Policy/Info | Default | Description |
|---|---|---|---|
| Optional Support Information | Department | None | The name of your department |
| | Support Phone Number | None | Phone number users can call for assistance. |
| | Support Email | None | Email address users can contact for assistance. |
| DEP Policies | Supervised Devices | Enabled | A supervised device has been entered into Apple DEP or has been configured using the Apple Configurator.<br>**Note:** Either this setting or **MDM Profile Removable** (see below) must be enabled. |
| | MDM Mandatory | Enabled | Enroll the device in device management. |
| | MDM Profile | Not | If enabled, the user is allowed to delete the device management profile |

| Group | Policy/Info | Default | Description |
|---|---|---|---|
| | Removable | enabled | from the device. Also, see discussion in Effect of Removing MDM Profile, How to Prevent .<br>**Note:** Either this setting or **Supervised Devices** (see above) must be enabled. |
| | Allow Pairing | Not enabled | If enabled, the device can pair with the user's associated wearable devices. |
| Skip Setup Screens | Passcode | Not skipped | If skipped, the user does not need to set a passcode on the device. |
| | Location Services | Skipped | If not skipped, Location Services are enabled. |
| | Restoring from backup | Skipped | If not skipped, backup and restore from backup are allowed. |
| | Apple ID and iCloud | Not skipped | If skipped, user is not prompt for Apple ID for the Apple App Store and iCloud services. |
| | Terms of Use | Not skipped | If skipped, user is not prompted to accept Apple's Terms of Service. |
| | Touch ID | Not skipped | If skipped, user is not prompted to activate and train the fingerprint identification system. |
| | Apple Pay | Skipped | If not skipped, user is prompted to enroll in Apple's payment system. |
| | Zoom | Skipped | Accessibility option. Not skipping Zoom enables a magnifying glass and other features described for Zoom at http://www.apple.com/accessibility/ios/ |
| | Send diagnostic info to Apple | Skipped | If not skipped, diagnostic information is sent to Apple. |
| | Siri | Skipped | If not skipped, user is prompted to enable and train the voice recognition system. |
| | Android Migration iOS 9 | Skipped | If not skipped, enable the moving of files from Android devices to iOS, as described at https://support.apple.com/en-us/HT201196. |

## Important GC Settings Affecting Apple DEP

Be aware that there are some key policy settings and standard device restrictions you can set in GC that affect how Apple DEP operates.

**Important:** Make sure you follow these recommendations for the policy sets and device policies you associate with Apple DEP profiles.

#### Good Agent: Allow Self-Authentication in Auth Delegation, Auto-Push Delegates

Multi-authentication delegation is a standard BlackBerry Dynamics feature that allows the function of authenticating the user to be "shared" among a group of defined GD applications. For details and steps, see the good Control online help topic "Assigning Authentication Delegates".

> **Note:** For the Good Agent application, make sure that you enable the setting **Allow self-authentication when no authentication delegate application is detected**.

Good Agent activation is required for Good MDM to determine a device's user. You should exercise care in setting user policy sets that have authentication delegation enabled. The 'Allow self-authentication when no authentication delegation application is detected' must be set to allow user to complete the activation of Good Agent without the need for additional apps on the DEP device

In addition, make sure that the required authentication delegate applications (defined by the administrator) are configured for auto-push (see Managed apps: enabling app auto-push, exempting policy sets ) so they are loaded on end-users' devices without the users' intervention and so you can manage the multi-auth delegation and other aspects of the proper versions of these delegate apps.

#### Device Access Controls: Allow Inventory of Personal Apps

> **Note:** In your device profiles associated with the policy sets that you associate with your Apple DEP profiles, be sure you set the **Allow Inventory of Personal Apps** in the Device Access Control section of Device policy reference: general
> .

This setting is needed to support the following functions of Good Agent:

- To determine the exact user of a device
- To monitor the state of applications pushed to the device of the app pushes themselves

### Steps for Defining DEP Profiles in Good Control

**To define Apple DEP profiles in Good Control:**

1. Navigate to **Apple DEP Profiles**.
2. Click **New DEP Profile**.
3. If you have already created a profile you want to use as a basis for the new profile, from the **Copy from** pulldown menu, select the name of the base profile.
4. Enter a mnemonic name for this profile.

   > **Note:** The DEP profile name cannot exceed 100 characters.

5. Complete the settings using the information in the table above.
6. Click **Save** to save your changes or **Cancel** to discard them.

### About Errors from Apple

Good Control attempts to verify the settings you specify in a DEP profile for consistency before submitting them to Apple.

Unfortunately, Apple might reject a profile without giving the exact combination of settings that might have been invalid. Testing by BlackBerry has shown that there is often no indication in errors returned from the DEP portal about the precise nature of an error.

### Effect of Removing MDM Profile, How to Prevent

If the DEP profile allows user to remove MDM profile and the user actually does remove it *before activating any application/container*, then subsequent app activation treats the device as a BYO ("Bring Your Own", that is, personal) device.

If such a situation is a security concern, BlackBerry recommends the following:

- In the DEP profile, enable supervised mode, disallow MDM removal and disallow skipping MDM enrollment.
- Set the iOS device restriction to disallow managed app removal and disallow access to the Apple App Store. Disallowing the Apple App Store ensures that only MDM can install apps on the device. See Functionality .

You can further ensure that end-user activates Good Agent (so GC can provide visibility about DEP device's actual user) by making Good Agent the first authentication delegate.

## Assigning DEP Profiles to Devices

Before assigning DEF profiles, you must have completed the details in One-time Setup with Apple for DEP Profiles in Good Control  and Defining DEP Profiles in Good Control .

**To assign Apple DEP profiles in Good Control:**

1. Navigate to **Apple DEP Devices**.
2. Select the devices you want to assign a DEP Profile.

   You have several ways to select:

   - From the **Filter** pulldown menu, select **No DEP Profile Assigned**.
   - Manually checkmark individual serial numbers.
3. Click **Assign DEP Profile**.
4. From the **DEP Profile** pulldown menu, select the desired profile.
5. Click **Assign** to assign the selected profile, or **Cancel** to discard your changes.

## Apple DEP Devices

See Working with DEP-Enrolled Devices .

# Good Control Web Services

Good Control has a web services interface for programatically administering the GC system itself and for device management. There are two main groups of services.

- SOAP/WSDL-over-HTTPS for working with the Good Control System itself, users, policies, and so forth. The web services are based on SOAP (Simple Object Access Protocol) and WSDL (Web Services Definition Language) over HTTPS. This is a long-standing, popular programming paradigm that is familiar to many programmers.
- HTTP (or REST) API for working with device management. This is a more recent programming paradigm than SOAP. The HTTP API has functions for device policies, device configurations, and much more related exclusively to device management.

For more details, with examples of usage, see Good Control Web Services.

# Good Control SOAP: location, request syntax, responses, and errors

Good Control includes a SOAP interface for administrative operations outside of the Good Control console. The GC and CAP WSDL files contain definitions of SOAP requests and their corresponding responses, including all fields, types, and error definitions.

## Location and other required schemas

On every on-premise, installed Good Control server the **gc.wsdl** and **cap.wsdl** files are located as follows:

```
    c:\good\docs\gc.wsdl
    c:\good\docs\cap.wsdl
```

Otherwise, to get a copy of the files for your IDE, contact your BlackBerry representative.

The top of both files also define other required schemas.

**Note:** Do not alter the definitions in the WSDL files.

## endpoints for standalone Good Control SOAP requests

The GC web services have two endpoints, depending on which of the WSDL files you are working with, either gc.wsdl or cap.wsdl.

**Note:** In the endpoints below, *localhost* is the fully qualified domain name of your GC server. Port 443 is implied by the use of the HTTPS protocol.

- gc.wsdl: **https://***localhost***/gc/services/GCService**
- cap.wsdl: **https://***localhost***/gc/soapproxy/cap**

## Request syntax

In general, the request names follow the form:

***verbObject*****Request**

where:

***verb*** is **Get**, **Add**, **Update** , **Delete**, **Remove**, and so on

***Object*** is one of GC's categories of administrative functions or focus, such as users, groups, roles, certificates, logs, and more.

Every request has its own unique fields (or elements) that are required or optional, as defined in the WSDL file. The field names are prefixed with the **<ns6:*fieldname*>** prefix.

## MIME type of request

You should set the **Content-type** in the header of your HTTPS request to **text/html** or you can leave the **Content-type** header out altogether.

**Note:** Do not set the MIME type to **application/xml**. This will result in an error.

## Transaction security

The GC web services rely on the WS-Security (WSSE) schema for protection transactions with your GC administrator credentials. The WSSE security type is username/password protection.

The SOAP header of every request must include the inclusion of the WSSE schema and your username and password, as shown in the example below. Notice that your username must match the AD ***domain**username*** syntax:

```
  .
  .
  .
  <soapenv:Header>
    <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
      <wsse:UsernameToken wsu:Id="UsernameToken-10" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd">
        <wsse:Username>
          someDomain\someAdminUsername
        </wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">
          my.password
        </wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  .
  .
  .
```

**Important:** Make sure that you use the exact version of this schema:

**docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd**

Some IDEs might automatically select an older version or other schemas that appear similar but are not correct.

## Response syntax

Responses for successful requests in general simply return a response body with the defined elements and values for the response. Every response has unique fields (or elements) that generally correspond to the fields on the request but are prefixed with the **`<ns2:`*`fieldname`*`>`** prefix.

Responses for requests that result in an error return a defined error message, as defined in the WSDL and listed in Error types

## Error types

If a request results in an error, the system returns an error message in the body of the response. Here is an example of an error response:

```
Content-Type: application/xop+xml; charset=UTF-8; type="text/xml"
Content-Transfer-Encoding: binary
Content-ID: <0.urn:uuid:B5B451F4DB81FB94A81407454300744@apache.org>

<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Body>
<soapenv:Fault>
<faultcode>soapenv:Server</faultcode>
<faultstring>
com.good.gmc.roles.AuthenticateAndEnforce::
ACCESS_MODULE::INVALID_CREDENTIALS::
Invalid username and/or password.
</faultstring>
<detail>
 <gc:Fault xmlns:gc="urn:fault.gc10.good.com">
   <gc:faultCode>INVALID_CREDENTIALS</gc:faultCode>
   <gc:faultMessage>Invalid username and/or password.</gc:faultMessage>
  </gc:Fault>
</detail>
</soapenv:Fault></soapenv:Body></soapenv:Envelope>
--MIMEBoundaryurn_uuid_B5B451F4DB81FB94A81407454300743--
```

The error types are enumerated near the beginning of the gc.wsdl file and are in general self-explanatory:

- INVALID_CREDENTIALS
- INVALID_USER_OR_PASSWORD
- ERROR_INSUFFICIENT_RIGHTS
- OPERATION_NOT_ALLOWED_FOR_SELF_SERVICE
- ILLEGAL_PARAMETER_FOR_SELF_SERVICE
- INVALID_PARAMETERS
- DB_EXCEPTION
- DATA_TOO_LONG
- SERVICE_EXCEPTION

- DB_CONSTRAINT_VOILATION
- AUTH_DELEGATION_EXCEPTION
- APP_POLICY_OVERRIDE_EXCEPTION
- DIRECT_CONNECT_INFO_EXCEPTION
- OPERATION_NOT_ALLOWED
- INVALID_USERID
- APPLICATION_NOT_FOUND
- USER_NOT_FOUND
- USER_NOT_ENTITLED
- USER_ALREADY_EXISTS
- USER_ACCOUNT_LOCKED

## Important notes about DeviceType

The SOAP API **`DeviceType`** complex type is used by **`GetDevicesRequest`** and other requests. Here are notes about how this type works.

> **Note:** The MDM HTTP API also includes a request that will return information about devices that are managed. See **`GET /mdm/devices`** in Device Details.

For **`DeviceType`**, the SOAP API attempts to gather details about the device via the BlackBerry Dynamics SDK and other sources on the device.

For phone number:

- On iOS, there is no mechanism to retrieve the data.
- On Android there is no reliable mechanism to retrieve the data.

For carrier info:

- On iOS if the device has a configured carrier network, **`DeviceType`** return its value; otherwise, it returns **`unknown`**.
- On Android, first an attempt to retrieve the SIM's operator name is made. If that is unsuccessful, an attempt is made to retrieve the network operator name (when the device is not roaming). If both attempts fail, the **`DeviceType`** returns null.

## Example: adding a user to GC from an Active Directory domain

Here is an example of programming a common need for the GC administrator: adding a user from Active Directory without using the GC console.

Here we show the SOAP calls needed to add a user who already exists in the GC associated AD domains:

1. With **`GetDirectoryUsersRequest`**, we search the Active Directory for a user named "smith".
2. With **`AddUserRequest`**, we add that user to the GC.

## getdirectoryusersrequest

We first need to search for a user. We invoke **`GetDirectoryUsersRequest`** to retrieve a list of users whose names match "smith", as specified in the **`<searchString>`** element:

```
POST https://localhost/gc/services/GCService HTTP/1.1
Content-Type: text/xml; charset=UTF-8
SOAPAction: "urn:gc10.good.com:gcServer:GetDirectoryUsersRequest"
User-Agent: Axis2
Host: localhost
Content-Length: 946

<?xml version="1.0" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
      <wsse:UsernameToken wsu:Id="UsernameToken-10" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd">
        <wsse:Username>
          someDomain\someAdminUsername
        </wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">
          my.password
        </wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <urn:GetDirectoryUsersRequest xmlns:urn="urn:gc10.good.com">
      <urn:searchString>
        smith
      </urn:searchString>
    </urn:GetDirectoryUsersRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

The GC web service returns a response like this:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/xml;charset=UTF-8
Content-Length: 530
Date: Wed, 14 Mar 2012 16:44:07 GMT

<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <urn:GetDirectoryUserResponse xmlns:urn="urn:gc10.good.com">
      <urn:users>
        <urn:displayName>
          John Smith
        </urn:displayName>
        <urn:sessionId>
          jsmith1@somecorp.com
        </urn:sessionId>
        <urn:domain>
          some.domain.com
        </urn:domain>
        <urn:firstName>
          John
        </urn:firstName>
        <urn:lastName>
          Smith
        </urn:lastName>
      </urn:users>
      <urn:isPartialResult>
        false
      </urn:isPartialResult>
    </urn:GetDirectoryUserResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

## adduserrequest

We take the returned values and pass them to **AddUserRequest**. Essentially, we can take the fields and values returned by from **GetUsersResponse**, change the namespace from **<urn:*fieldname*>** to **<urn:*fieldname*>**, and pass the values verbatim to **AddUserRequest**:

```
POST https://localhost/gc/services/GCService HTTP/1.1
Content-Type: text/xml; charset=UTF-8
SOAPAction: "urn:gc10.good.com:gcServer:AddUserRequest"
User-Agent: Axis2
Host: localhost
Content-Length: 1283

<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
      <wsse:UsernameToken wsu:Id="UsernameToken-10" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd">
        <wsse:Username>
          someDomain\someAdminUsername
        </wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">
          my.password
        </wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <urn:AddUserRequest xmlns:urn="urn:gc10.good.com">
      <urn:user>
        <urn:displayName>
          John Smith
        </urn:displayName>
        <urn:sessionId>
          jsmith1@somecorp.com
        </urn:sessionId>
        <urn:domain>
          some.domain
        </urn:domain>
        <urn:firstName>
          John
        </urn:firstName>
        <urn:lastName>
          Smith
        </urn:lastName>
      </urn:user>
    </urn:AddUserRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

On success, the system responds like this:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/xml;charset=UTF-8
Content-Length: 755
Date: Wed, 14 Mar 2012 16:44:10 GMT

<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <urn:AddUserResponse xmlns:urn="urn:gc10.good.com">
      <urn:user>
        <urn:userId>
          7733
        </urn:userId>
        <urn:displayName>
          John Smith
        </urn:displayName>
        <urn:sessionId>
          jsmith1@somecorp.com
        </urn:sessionId>
        <urn:domain>
          some.domain
        </urn:domain>
        <urn:firstName>
          John
        </urn:firstName>
        <urn:lastName>
          Smith
        </urn:lastName>
        <urn:status>
          1
        </urn:status>
      </urn:user>
    </urn:AddUserResponse>
  </soapenv:Body>
</soapenv:Envelope>
```