Kerberos Constrained Delegation With Good Control

Version 4.2

Contents

Revision history

Kerberos Constrained Delegation

Date	Description
2017-09-19	Determining whether you should upgrade to BlackBerry UEM
2017-08-03	Eliminate stray reference to erroneous port 5060. The standard Kerberos port 88 must be configured, not port 5060.
2017-07-18	Updated for latest release, no content changes.
2017-01-31	Version numbers updated for latest release; no content changes.
2017-01-27	Miscellaneous technical corrections in Step 2: changing the Kerberos constrained delegation (KCD) keytab password and Configuring Kerberos-related properties in Good Control
2017-01-24	Step 2: changing the Kerberos constrained delegation (KCD) keytab password now present a design choice: Reuse the GC service account with its hardcoded password or use a separate service account specifically for KCD with a randomly generated password.
2017-01-23	Major revisions to: - Step 1: map the Good Control (GC) service account to a service principal name (SPN) - Step 3: configure constrained delegation for the desired resources **Note:** Formerly, this document recommended the use of **`setspn -a`**. This was in error: do not use **`setspn -a`**.
2016-12-19	Version numbers updated for latest release; no content changes.
2016-11-07	Added note About setspn -U option with service account names
2016-07-28	Corrected Step 5: Good Control: enable the GC service account to act as part of the operating system which incorrectly stated that this permission is given on the AD domain controller. This is a potential security risk. Instead, the permission should be given to the account that is running the GC on the GC itself.
2016-06-29	Version numbers updated for latest release; no content changes.
2016-06-09	Added Distinction from KCD and behavior of Kerberos PKINIT .
2016-05-13	Various clarifications: - Addition to Configuring Kerberos-related properties in Good Control of the GC server property gc.krb5.config.file which specifies the path to krb5.conf file on the GC - Clarification that the step Step 5: Good Control: enable the GC service account to act as part of the operating system is done on the Active Directory server where the GC service account has been defined.

Date	Description
2016-03-10	Truncated revision history to reduce bulk.
2016-01-15	Version numbers updated for latest release; no content changes.
2015-12-23	Version numbers updated for latest release; no content changes.
2015-12-01	Corrected port number in DNSfor GC and GP in separate domains, Kerberos vs. KCD : port 88, not 5060
2015-11-10	Added note about the need to set up a Service Principal Name if you use DNS aliases in Step 3: configure constrained delegation for the desired resources
2015-11-05	Emphasized the need for identical KCD configuration on all GC servers in a cluster in About BlackBerry Dynamics clustering: identical KCD config on all GCs

Kerberos constrained delegation

With Kerberos Constrained Delegation (KCD) end users can access enterprise resources without having to enter their network credentials. KCD uses *service tickets* that are encrypted and decrypted by keys that do not contain the user's credentials.

Part of KCD is a mechanism called *delegation*. When this mechanism is configured, the application delegates authentication to Good Control (GC) to act on its behalf to request access to an enterprise resource.

Another mechanism is the ability to *constrain* the accessed resources. With this mechanism administrators can limit the network resources that are accessible. This is accomplished by configuring the account under which the delegate (the GC) run as trusted only for specific services.

Distinction from KCD and behavior of Kerberos PKINIT

Kerberos PKINIT is distinct from Kerberos Constrained Delegation (KCD).

Kerberos PKINIT	Kerberos Constrained Delegation
Kerberos PKINIT authentication is between the BlackBerry Dynamics-enabled client application and the Windows Key Distribution Center (KDC), which communicate directly, and user authentication is based on certificates issued by Active Directory Certificate Services.	**Note:** For PKINIT, Kerberos Constrained Delegation must *not* be enabled. 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. Kerberos Constrained Delegation takes precedence over Kerberos PKINIT, even if the user has a valid certificate.

Terminology and equivalences

Terminology in Kerberos is notoriously obscure and difficult to understand. Here is some of the Kerberos terminology used in this guide.

- *User impersonation* is the Kerberos term for the configurations discussed in this document. Because the user never presents any credentials in this configuration, but instead relies on the Active Directory and Kerberos systems for authentication, the user is technically "impersonated." (User impersonation is technically distinct from Kerberos resource delegation.)
- *REALM* is the Kerberos term for a collection of "entities," either user realms or resource realms, which are any realm other than a user realm. When entered on Kerberos command lines, the REALM name must always be in uppercase.
- *Domain* in this context means "directory service domain", most frequently from Active Directory.

The terms *realm* and *domain* are conceptually equivalent in KCD. The relationship is as follows:

1 Kerberos realm : *n* Good Control server : *n* Good Proxy

There must be a minimum of one Good Control server for each Kerberos realm. The GC must still reside in the same Kerberos realm as the resource because cross-realm resource delegation is not supported. In addition, you must ensure that all Good Proxy servers and clusters can communicate with all other GC and GP servers that you intend to configuration for multi-realm KCD. Other requirements for multi-realm and forest topologies are detailed below.

About BlackBerry Dynamics clustering: identical KCD config on all GCs

Clustering of BlackBerry Dynamics components fully supports KCD. However, be sure you configure all the servers in your cluster for KCD with the identical configuration files and settings described in this guide. This is specifically called for in the commands documented in Concepts and steps for a typical KCD single realm installation and Concepts and steps for KCD multi-realm configuration .

Standard Kerberos port 88 on AD server must be open

The assigned well-known port for Kerberos is port 88 for both UDP and TCP transport protocols, as defined by the Internet Assigned Numbers Authority (IANA).

Important: Port 88 on your Active Directory service must be accessible by all Good Control servers that participate in KCD.

Be sure that no firewalls or other settings interfere with this communication.

Related Kerberos tools

Your Kerberos environment setup must include the following components:

- **Active Directory (AD) Server** – the directory service that authenticates and authorizes all users and computers associated with your Windows network.
- **Kerberos Key Distribution Center (KDC)** – the authentication service on the AD server that supplies session tickets and keys to users and computers in the Active Directory domain.
- **AD Support Tools** – additional tools that are used to configure, manage, and debug AD.

Video tutorials

There are helpful video tutorials about configuring single-ream and multi-realm KCD:

- Single-realm: https://community.good.com/videos/1443
- Multi-realm: https://community.good.com/videos/1474

- BlackBerry Dynamics Single Realm KCD Configuration Example
- BlackBerry Dynamics Multi-Realm KCD Configuration Example
- BlackBerry Work KCD Configuration Example

The videos also include other helpful, supporting materials.

Domains, realms, forests, trusts, and GC and GP: topologies

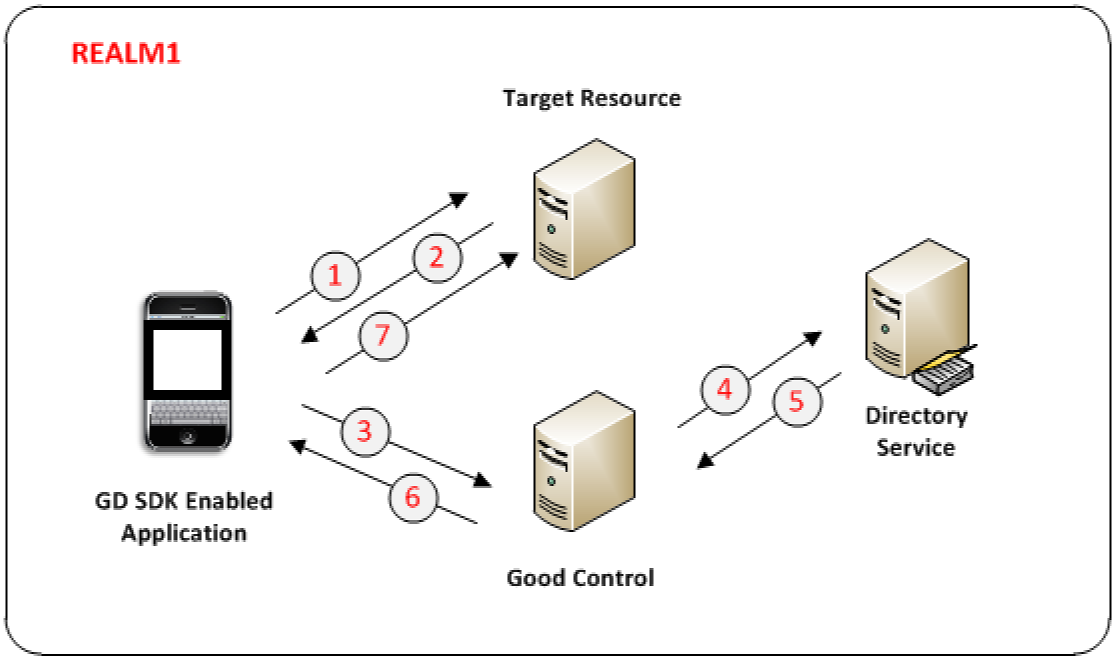
The terms *realm* and *domain* are conceptually equivalent in KCD. The relationship is as follows:

1 Kerberos realm : *n* Good Control server : *n* Good Proxy

There must be a minimum of one Good Control server for each Kerberos realm. The GC must still reside in the same Kerberos realm as the resource because cross-realm resource delegation is not supported. In addition, you must ensure that all Good Proxy servers and clusters can communicate with all other GC and GP servers that you intend to configuration for multi-realm KCD. Other requirements for multi-realm and forest topologies are detailed below.

Example diagram: single Kerberos realm

A typical single realm KCD transaction works as follows.



1. An application makes a request an internal server or service. We call this the *target*.

   The *target* can be either a host name (server name) or an account that is to be protected by Kerberos/BlackBerry Dynamics. For instance, if you are running IIS on a server as the Network service, then the *target* is the computer running IIS as Network. On the other hand, if you run IIS as an actual user (for instance, IISSrvUser), then *target* is that user name, IISSrvUser.

   Thus, if you have 1,000 IIS servers running as Network service, *target* is the names of those 1,000 machines. But if you run IIS as a user on those 1,000 machines, then *target* is the name of that user.

   Another example is WebSphere, in which case *target* is the Kerberos user name of the WebSphere service.
2. The target replies with an authentication challenge that the BlackBerry Dynamics Runtime intercepts.
3. The BlackBerry Dynamics Runtime sends a request to GC for a service ticket to access the target.
4. GC authenticates the user/container (via internal BlackBerry Dynamics protocols) and asks for a service ticket on behalf of the user (this is *delegation*) for the service on the target host.
5. Active Directory (AD) checks its local policy. Then (a) if the user has permission to access the resource on the target host and (b) if the resource on the target host is allowed (this is *constrained*), AD returns to Good Control a service ticket for the resource.
6. Good Control sends the necessary information from the returned service ticket to the BlackBerry Dynamics Runtime.
7. The BlackBerry Dynamics Runtime uses the information from GC to complete the authentication to the target host.

DNSfor GC and GP in separate domains, Kerberos vs. KCD

The Good Control server and the Good Proxy server are often installed in the same Kerberos domain but they do not have to be. You might want to install the GP in your DMZ or "sacrificial" workgroup. If you choose this latter configuration, you need to set-up some required network configuration, as detailed below.

There is a distinction in how BlackBerry Dynamics operates between normal Kerberos and Kerberos Constrained Delegation (KCD) that affects your network configuration.

- In KCD, on behalf of the client applications, the GC service itself requests authentication tickets from the ticketing server (the domain controller).
- In normal Kerberos (without constrained delegation), the client applications themselves make the ticketing requests, not the GC, and the request is passed through (egress)the GP. This means that the GP needs to be able to discover the name of Kerberos domain controller (server). In your Domain Name System (DNS), you need to add an **`SRV`** record specifying the Kerberos service that enables this discovery. This **`SRV`** record must be associated with an **`A`** or **`AAAA`** record, not a **`CNAME`** record. The syntax below is for a Kerberos domain controller in an Internet domain named example.com:

  ```
  _kerberos._tcp.example.com. 86400 IN SRV 0 5 88 kerberos.example.com.
  ```

  This points to a server named kerberos.example.com listening on TCP port 88 for Kerbeors requests. The priority given here is 0, and the weight is 5.

Consult your networking documentation for the exact steps and definitions in creating a **SRV** record for your Kerberos services when the GP is in its own domain.

Requirements and recommendations for multiple Kerberos realms and forests

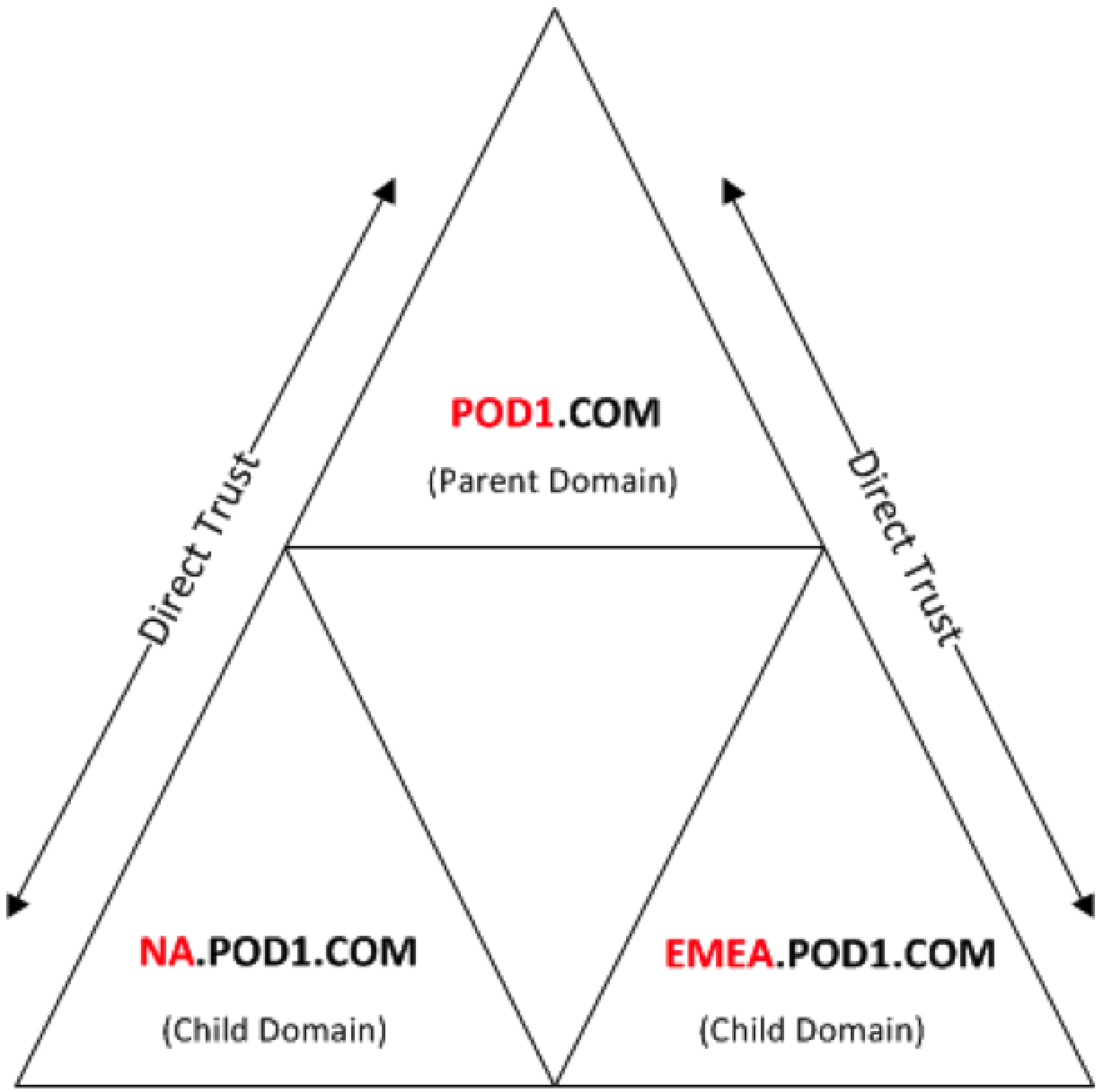
Note: Much of this material is available as a video tutorial from Good's Solutions Architecture group at https://community.good.com/videos/1474.

Multi-realm KCD is supported across any combination of Active Directory 2003, Active Directory 2008, and Active Directory 2012. For multiple Kerberos realms and forests, ensure the following prerequisites before implementation.

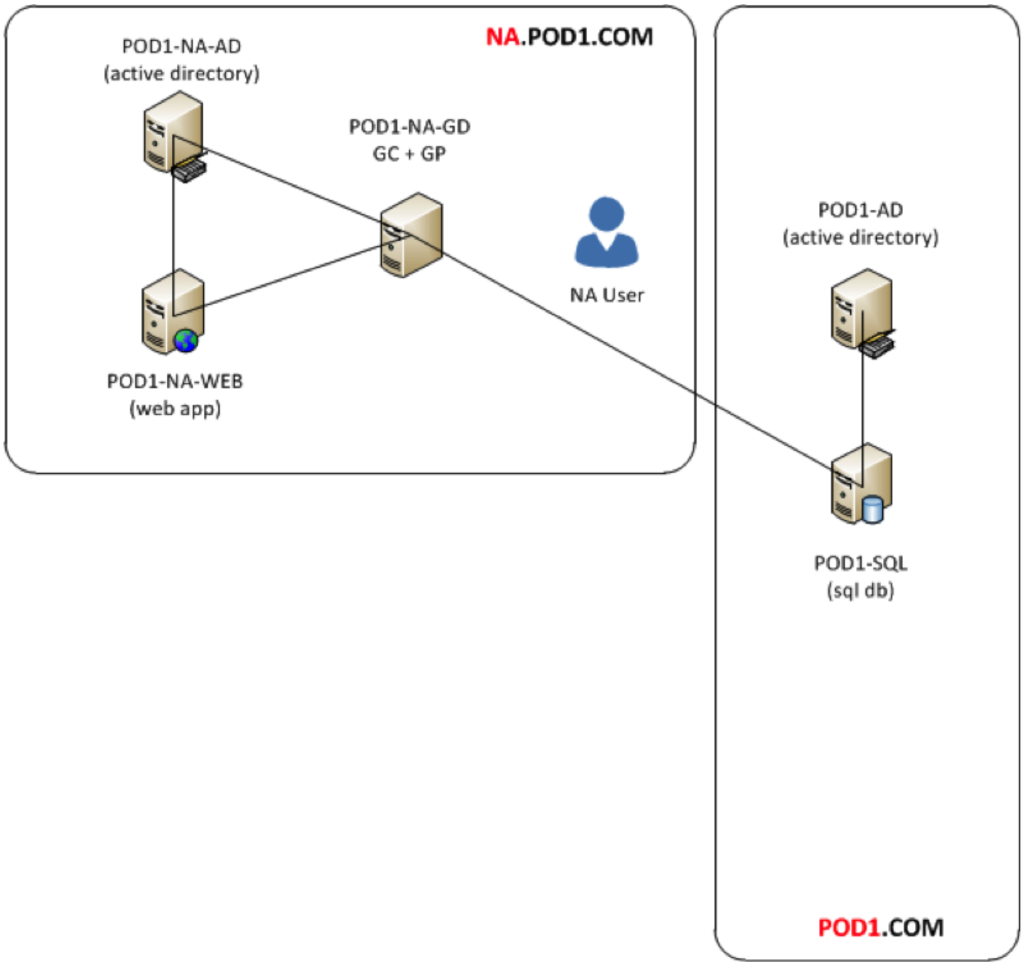
Good Control	SQL Database	KCD	Forests
• At least one Good Control server must be installed in every resource realm that has resources needed by other realms. • All Good Control servers must be in the same cluster. Therefore, they all must share the same database.	If you are using Microsoft SQL Server in the multi-realm configuration, make sure of the following: • Use an SQL account, not the Windows account. • That SQL account must be the owner of the database. • All GCs in the multi-realm KCD configuration must use that same SQL account.	• Ensure that single realm KCD is working before configuring multi-realm KCD. • Ensure that target resources are properly configured for KCD .	• All trust must be bidirectional, transitive forest trust. • If a Kerberos domain is between any two other domains, the trust must be transitive.

Example KCD multi-realm within single forest

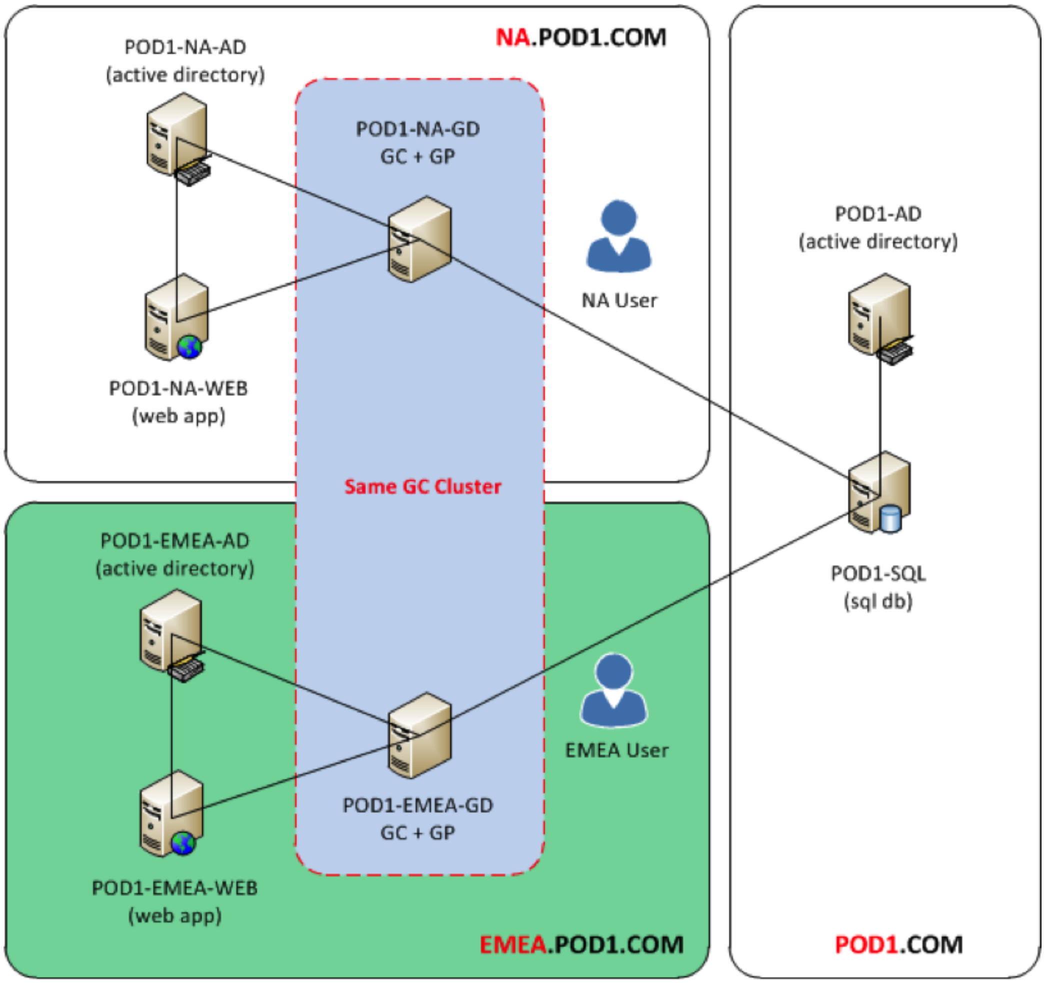
Consider the following diagrams. These are different views of the same deployment: 1. Trust relationships between domains in a single forest (without regard for BlackBerry servers)



2. Verification of the Good Control and Good Proxy connections to the parent domain, POD1.com



3. The logical layout of the multi-realm, single-forest configuration, with GC/GP servers

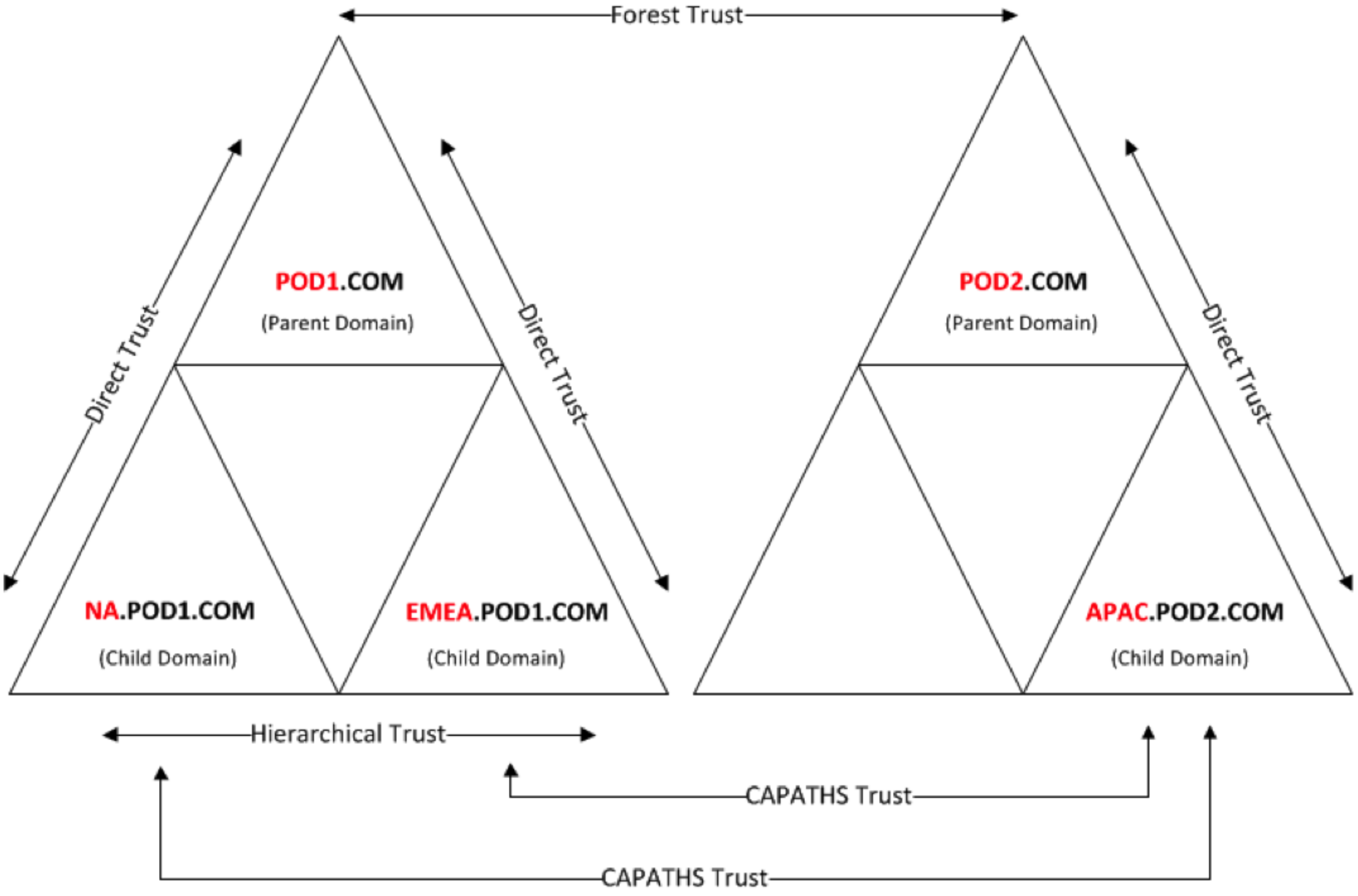


Example KCD Multi-Realm between Two Forests

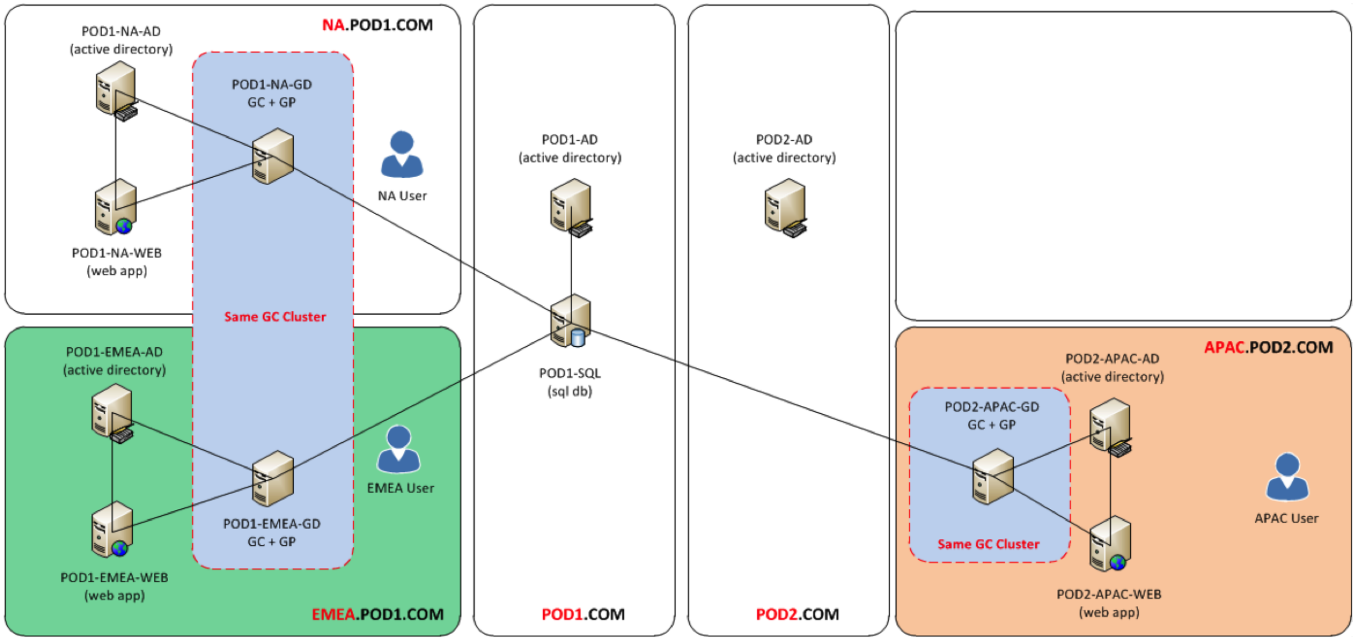
Here are similar diagrams when two forests are involved.

1. Trust relationships between domains in two forests

Domains, realms, forests, trusts, and GC and GP: topologies



2. Logical view of two forests and domains with GC/GP cluster

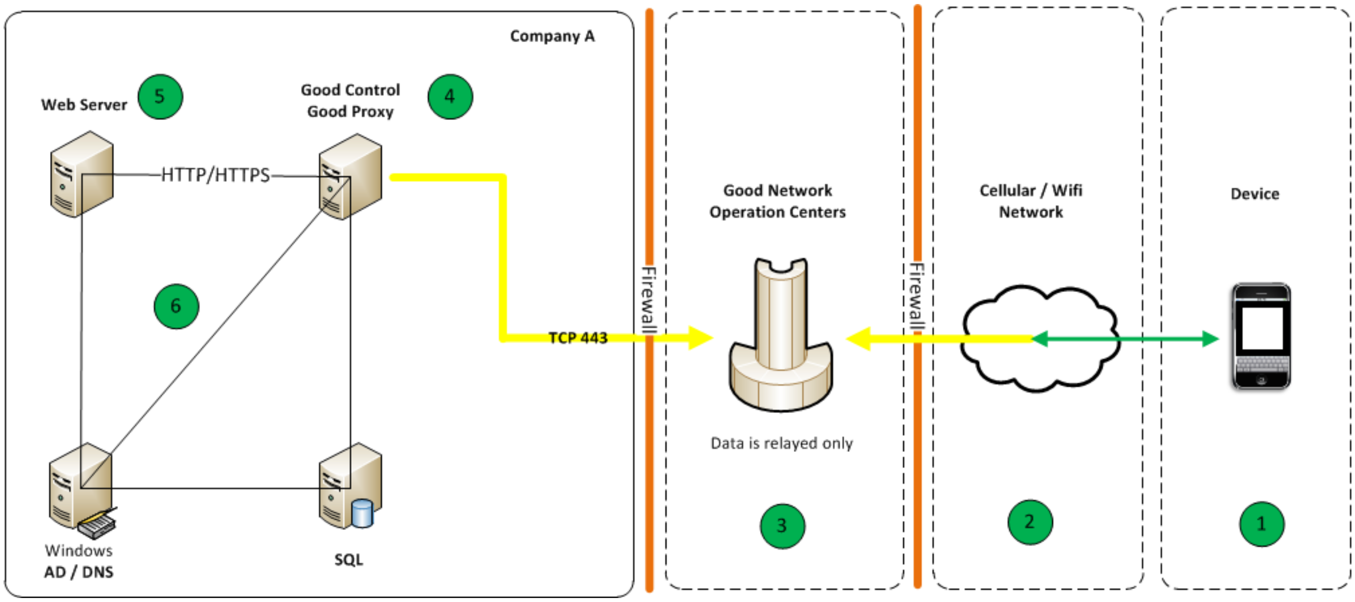


Concepts and steps for a typical KCD single realm installation

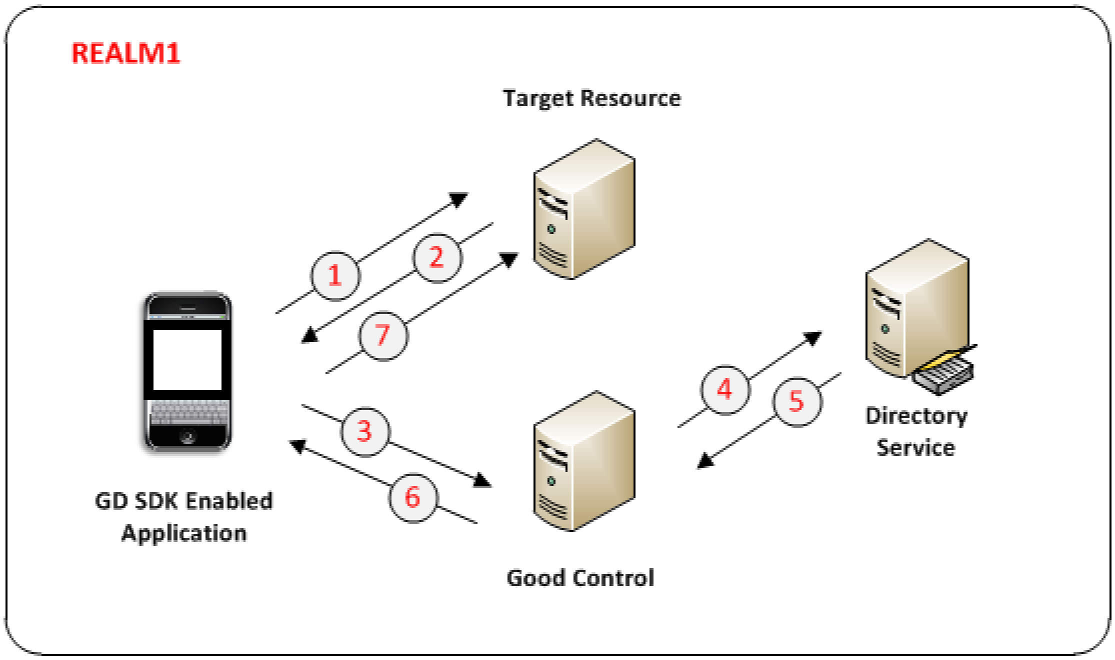
The steps that follow show how to set up a typical installation for a single Kerberos realm with BlackBerry Dynamics.

Example diagrams: single Kerberos realm

Consider the following single-realm KCD topology:



A typical single realm KCD transaction works as follows.



1. An application makes a request an internal server or service. We call this the *target*.

   The *target* can be either a host name (server name) or an account that is to be protected by Kerberos/BlackBerry Dynamics. For instance, if you are running IIS on a server as the Network service, then the *target* is the computer running IIS as Network. On the other hand, if you run IIS as an actual user (for instance, IISSrvUser), then *target* is that user name, IISSrvUser.

   Thus, if you have 1,000 IIS servers running as Network service, *target* is the names of those 1,000 machines. But if you run IIS as a user on those 1,000 machines, then *target* is the name of that user.

   Another example is WebSphere, in which case *target* is the Kerberos user name of the WebSphere service.

2. The target replies with an authentication challenge that the BlackBerry Dynamics Runtime intercepts.
3. The BlackBerry Dynamics Runtime sends a request to GC for a service ticket to access the target.
4. GC authenticates the user/container (via internal BlackBerry Dynamics protocols) and asks for a service ticket on behalf of the user (this is *delegation*) for the service on the target host.
5. Active Directory (AD) checks its local policy. Then (a) if the user has permission to access the resource on the target host and (b) if the resource on the target host is allowed (this is *constrained*), AD returns to Good Control a service ticket for the resource.
6. Good Control sends the necessary information from the returned service ticket to the BlackBerry Dynamics Runtime.
7. The BlackBerry Dynamics Runtime uses the information from GC to complete the authentication to the target host.

Creating SPNs for all HTTP services, like SharePoint

In order to protect the various resources served by Good Enterprise Mobility Server, BlackBerry Share, SharePoint or other systems, you need to create a Service Principal Name (SPN) in Active Directory for each service. These SPNs can then be added to the KCD configuration.

1. Create a dedicated user for running the server service. In this example the user is **`domain\GSSUser`**.
2. Set the password for GSSUser to never expire and do not require a password change for logging on.
3. Create a Service Principle Name (SPN) for each web application that needs to be shared as shown below:

   ```
   setspn -S HTTP/SPHOST:PORT domain\AppPoolUser
   ```

   ```
   setspn -S HTTP/SPHOST.FQDN:PORT domain\AppPoolUser
   ```

   ```
   setspn -S HTTP/SPHOST domain\AppPoolUser
   ```

   ```
   setspn -S HTTP/SPHOST.FQDN domain\AppPoolUser
   ```

   If the port is a default port (80 or 443), omit the first two lines above. Note that some of the lines need just a host name while others need a fully qualified host name. If the application pool identity is for a built-in user such as Network Service, then specify the host name as shown below, instead of **`domain\AppPoolUser`**

   ```
   setspn -S HTTP/SPHOST:PORT domain\SPHOST
   ```

   ```
   setspn -S HTTP/SPHOST.FQDN:PORT domain\SPHOST
   ```

```
setspn -S HTTP/SPHOST domain\SPHOST
```

```
setspn -S HTTP/SPHOST.FQDN domain\SPHOST
```

Note: If you are using SSL, the SPN must refer to HTTP instead of HTTPS.

4. Create a SPN for the BlackBerry Share Server process user as shown below:

```
setspn -S HTTP/GSSHOST domain\GSSUser
```

```
setspn -S HTTP/GSSHOST.FQDN domain\GSSUser
```

GSSHOST is the hostname of the GSS.

Note: An HTTP service (such as IIS) need not be running on the server machine. The lines above are needed just to enable the Delegation tab in the user's properties tab in Active Directory.

About setspn -U option with service account names

In the sections that follow, the **`setspn`** command is used to specify the service principal name to associate with Good Control service accounts cooperating in Kerberos Constrained Delegation.

If your Good Control service account names are the same as your Good Control server names, **`setspn`** assumes that the account you specify on the **`setspn`** command is a computer account, not a user/service account, which is not the correct assumption.

To avoid this confusion, specify **`-U`** on the **`setspn`** command, which indicates that the account specified is a user account.

Step 1: map the Good Control (GC) service account to a service principal name (SPN)

For more information on how to create/modify SPNs, see https://technet.microsoft.com/en-us/library/cc731241.aspx

To use the command line, open an administrator command prompt on the AD machine and enter:

```
setspn -s GCSvc/GC_host_machine DOMAIN\GC_service_account
```

Replace the host machine name, domain, and service account variables with values appropriate to your environment.

Example:

```
setspn -s GCSvc/gchost.gd.qagood.com gd.qagood.com\gdadmin
```

SPNs for other services

If you are working with Good Enterprise Mobility Server, SharePoint or other services, you need to make SPNs for them, too. See the examples in Creating SPNs for all HTTP services, like SharePoint .

Step 2: changing the Kerberos constrained delegation (KCD) keytab password

You have two design options when you create the Kerberos account and set its password:

1. Use the Good Control service account with its hardcoded password
2. Use a separate account specifically for KCD and a randomly generated password.

Design option #2 is a safer choice: to not reuse your service account for KCD, but instead use a separate service account for Kerberos only. You can also generate a random password for this accpunt with the **`+rndPass`** argument on the ktpass command (shown below). When using the GD service account, this option is not possible because the admin needs to know the password. However, when segregating KCD to a new service account, this option can be used and is much more secure.

A new keytab file must be generated and copied to the GC machine whenever the ***`kerberos_account_password`*** is changed.

1. Open a command prompt window on the KDC server
2. Use the **`ktpass`** command to set the Kerberos account password, either randomly generated or hardcoded:

   Syntax for randomly generated password with +rndPass argument

   ```
   ktpass -out outfilename.keytab -mapuser kerberos_account@REALM_IN_ALL_CAPS -princ kerberos_account@REALM_IN_
   ALL_CAPS -ptype KRB5_NT_PRINCIPAL +rndPass
   ```

   Syntax for hardcoded GC service account password with -pass argument

   ```
   ktpass -out outfilename.keytab -mapuser kerberos_account@REALM_IN_ALL_CAPS -princ kerberos_account@REALM_IN_
   ALL_CAPS /ptype KRB5_NT_PRINCIPAL -pass kerberos_account_password
   ```

   where:

outfilename	is the name of the output file.
kerberos_account	is the Kerberos account name.
REALM_IN_ALL_CAPS	is the Kerberos realm in all capital letters.
-pass *kerberos_account_password*	is the same as the existing password for the reused GC Kerberos account. Enclose the value of *kerberos_account_password* in double quotation marks, especially if it contains special characters, such as ^. Mutually exclusive with next option.
+rndPass	sets a random password for the service account and thus the password is never shown in clear text nor will it be known to anyone. Mutually exclusive with the above option.

Example of randomly generated password with +rndPass argument

```
ktpass -outgdadmin.keytab -mapuser separatekcdaccount@GD.QAGOOD.COM -princ separatekcdaccount@GD.QAGOOD.COM
-ptype KRB5_NT_PRINCIPAL +rndPass
```

Example of hardcoded GC service account password with -pass argument

```
ktpass /out gdadmin.keytab /mapuser gdadmin@QAGOOD.COM /princ
gdadmin@GD.QAGOOD.COM /pass gdadmin /ptype KRB5_NT_PRINCIPAL@
```

3. Copy the new keytab file (**gdadmin.keytab** in the examples) saved in this directory to the GC server.

Important: If you have clustered your GC servers, you must copy the keytab file to every GC server in the cluster.

Step 3: configure constrained delegation for the desired resources

To configure constrained delegation, your purposed target needs to authenticate Kerberos (for example, Exchange or SharePoint). Without having this resource authenticating Kerberos, there can be no delegation. For a resource to authenticate Kerberos, it must have a SPN configured (among other configuration). To verify if your target has a SPN, use the following command:

```
setspn -q http/my.resource.fqdn
```

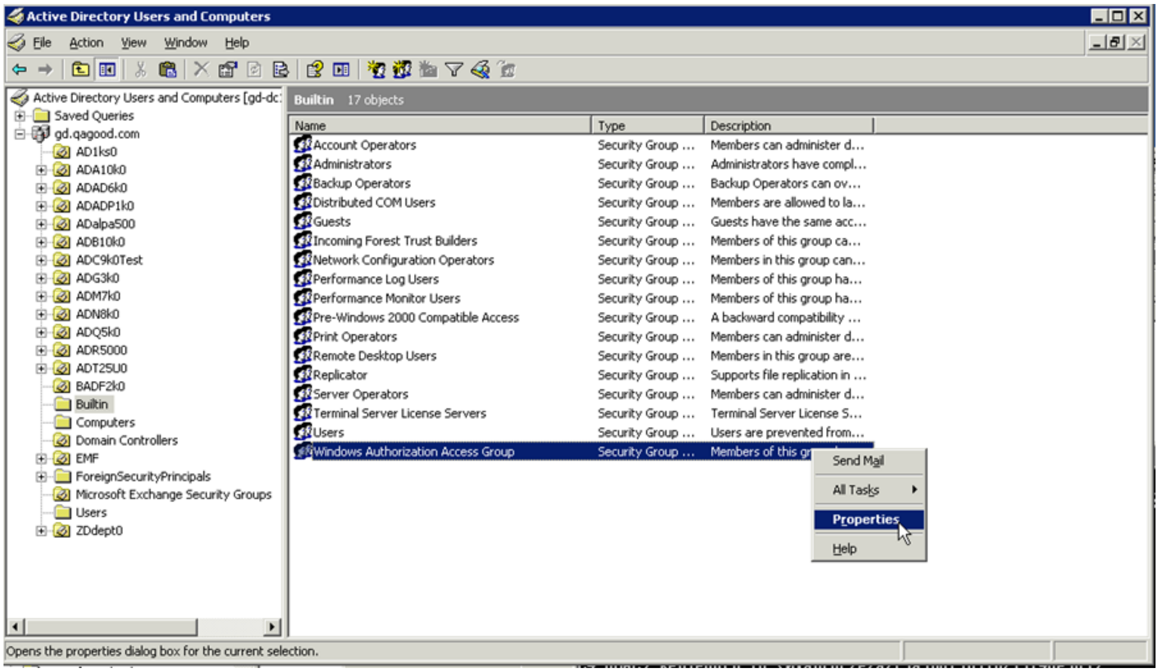
This should return the distinguished name of the service account holding the SPN. If the return is empty, there is no SPN configured and it needs to be registered to the appropriate service account, for example, with the following command:

```
setspn -s http/my.resource.fqdn mydomain\resourceserviceaccount
```

Aside from registering the SPN, additional configuration might be required, depending on the software product of the resource. IIS-based services require “Windows Authentication” to be enabled and the application pool to run under the resource service account.

Step 4: enable enumeration of AD user objects group membership

1. In your **Active Directory Users and Computers** mmc console, select **Builtin** from the list on the left, then right-click **Windows Authorization Access Group** and select **Properties**.



Name	Type	Description
Account Operators	Security Group ...	Members can administer d...
Administrators	Security Group ...	Administrators have compl...
Backup Operators	Security Group ...	Backup Operators can ov...
Distributed COM Users	Security Group ...	Members are allowed to la...
Guests	Security Group ...	Guests have the same acc...
Incoming Forest Trust Builders	Security Group ...	Members of this group ca...
Network Configuration Operators	Security Group ...	Members in this group can...
Performance Log Users	Security Group ...	Members of this group ha...
Performance Monitor Users	Security Group ...	Members of this group ha...
Pre-Windows 2000 Compatible Access	Security Group ...	A backward compatibility ...
Print Operators	Security Group ...	Members can administer d...
Remote Desktop Users	Security Group ...	Members in this group are...
Replicator	Security Group ...	Supports file replication in ...
Server Operators	Security Group ...	Members can administer d...
Terminal Server License Servers	Security Group ...	Terminal Server License S...
Users	Security Group ...	Users are prevented from...
Windows Authorization Access Group	Security Group ...	Members of this gr

Opens the properties dialog box for the current selection.

2. In the **Windows Authorization Access Group Properties** popup, click the **Members** tab, then click **Add...**
3. In the **Select Users**, **Contact**, **Computers** or **Group** popup, enter the name of the GC service account, and click **OK**.

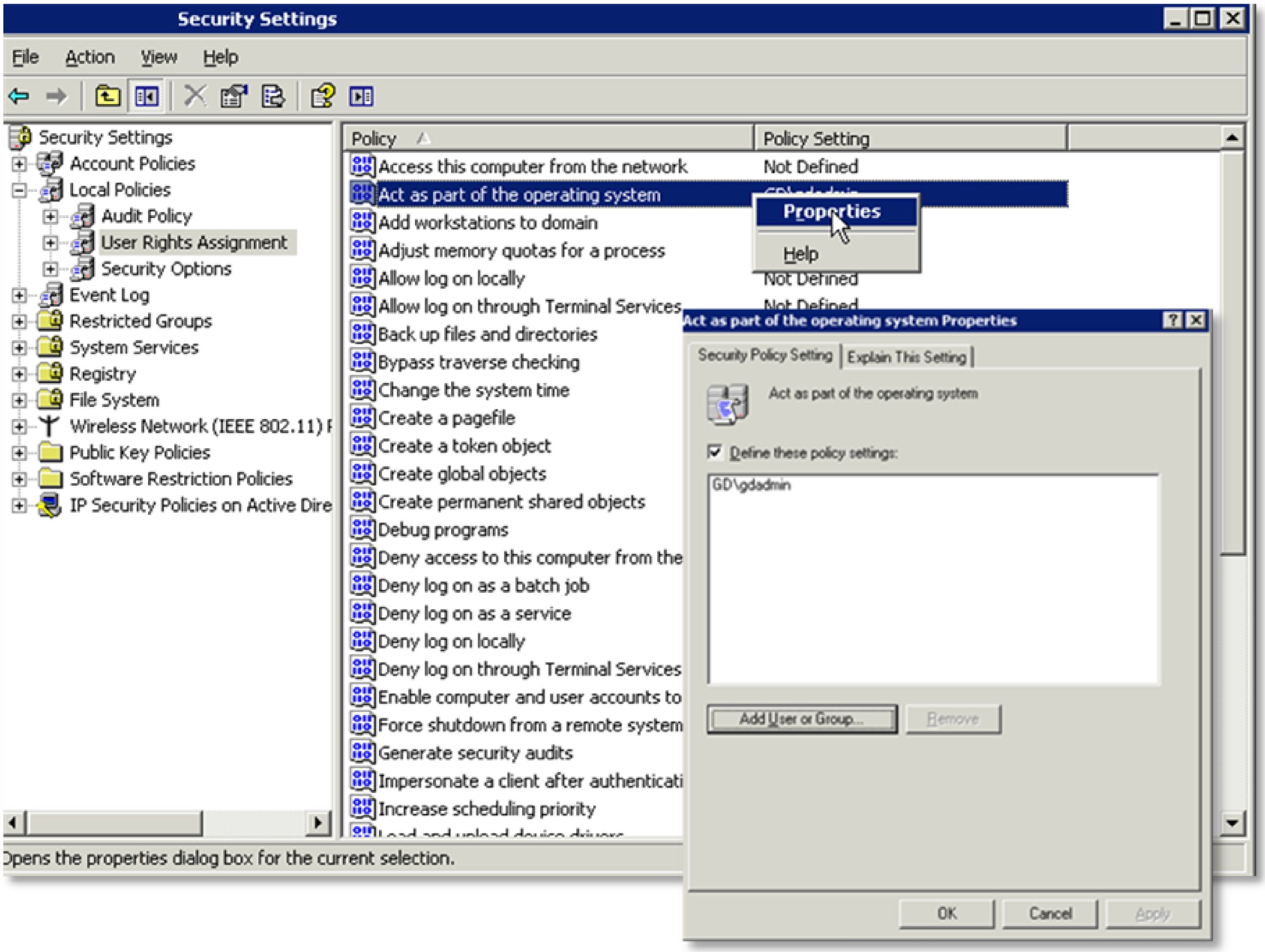
Step 5: Good Control: enable the GC service account to act as part of the operating system

You need to give the GC service account the permission to present the end-user's credentials to the Kerberos system on the end-user's behalf. This permission is **Act as part of the operating system**.

This configuration is on the machine that is running the Good Control service, not on the domain controller (which is not a proper security practice). This is the same account that has the associated Service Principal Name (SPN) you created previously. This is also the same account configured with the ktpass tool to create the keytab file.

1. Open the **Local Security Policy** pane in the Windows console.
2. Under **Local Policies**, select **User Rights Assignments**, then right-click **Act as part of the operating system** in the

right panel and select **Properties**.



Policy	Policy Setting
Access this computer from the network	Not Defined
Act as part of the operating system	GD\gdadmin
Add workstations to domain	
Adjust memory quotas for a process	
Allow log on locally	Not Defined
Allow log on through Terminal Services	Not Defined
Back up files and directories	
Bypass traverse checking	
Change the system time	
Create a pagefile	
Create a token object	
Create global objects	
Create permanent shared objects	
Debug programs	
Deny access to this computer from the	
Deny log on as a batch job	
Deny log on as a service	
Deny log on locally	
Deny log on through Terminal Services	
Enable computer and user accounts to	
Force shutdown from a remote system	
Generate security audits	
Impersonate a client after authenticati	
Increase scheduling priority	
Load and unload device drivers	

Opens the properties dialog box for the current selection.

Act as part of the operating system

☑ Define these policy settings:

GD\gdadmin

3. In the **Properties** popup, click on **Add User or Group...**, then enter the name of the GC service account and click **OK**.

Concepts and steps for KCD multi-realm configuration

Be sure to be familiar with the concepts and steps in Concepts and steps for a typical KCD single realm installation . The steps here assume your familiarity with the commands and steps for a single realm.

With multi-realm configuration, always start by configuring and testing a single realm first. Then proceed to adding the other realms or forests.

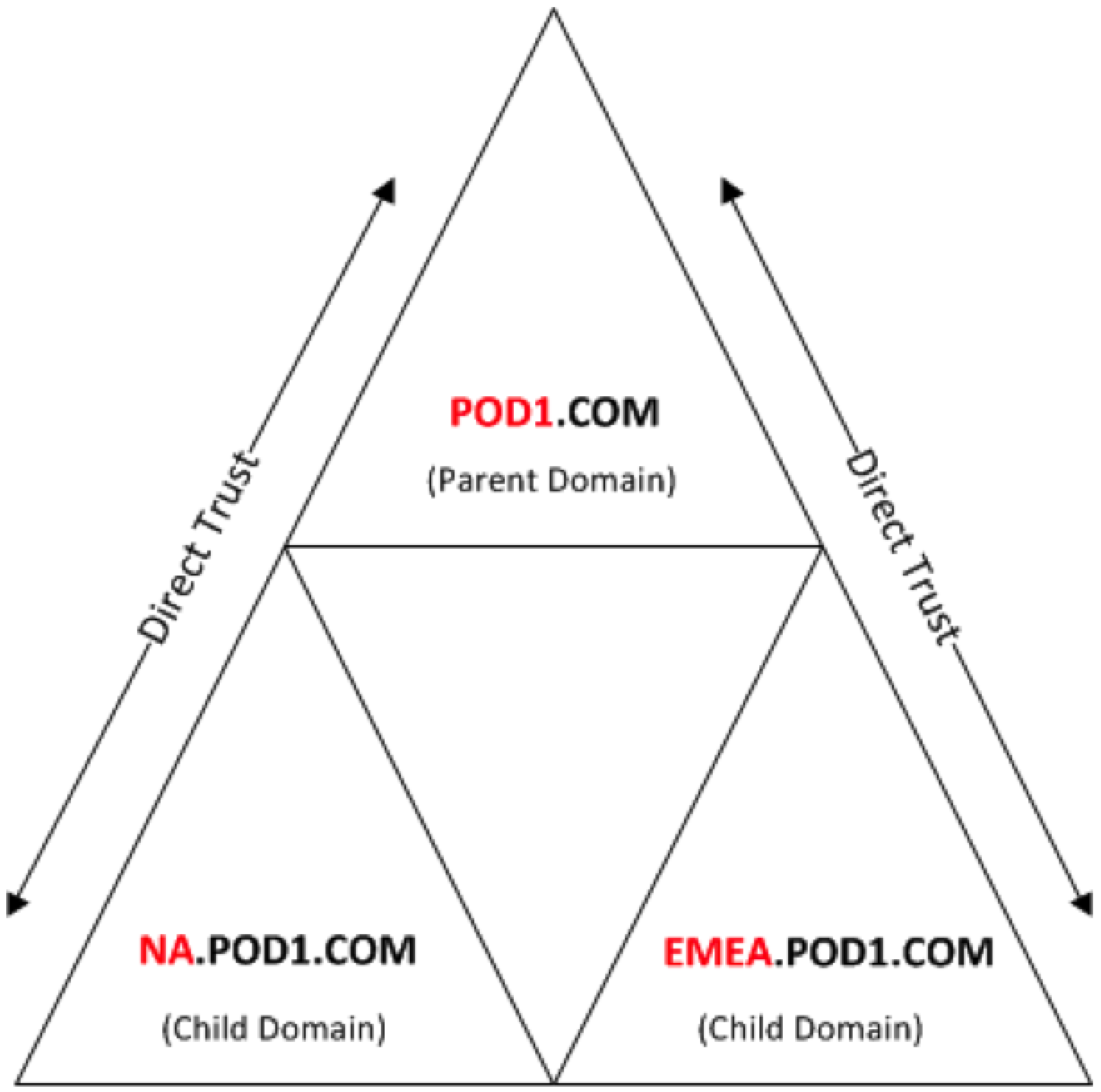
SQL database account for multi-realm KCD configuration

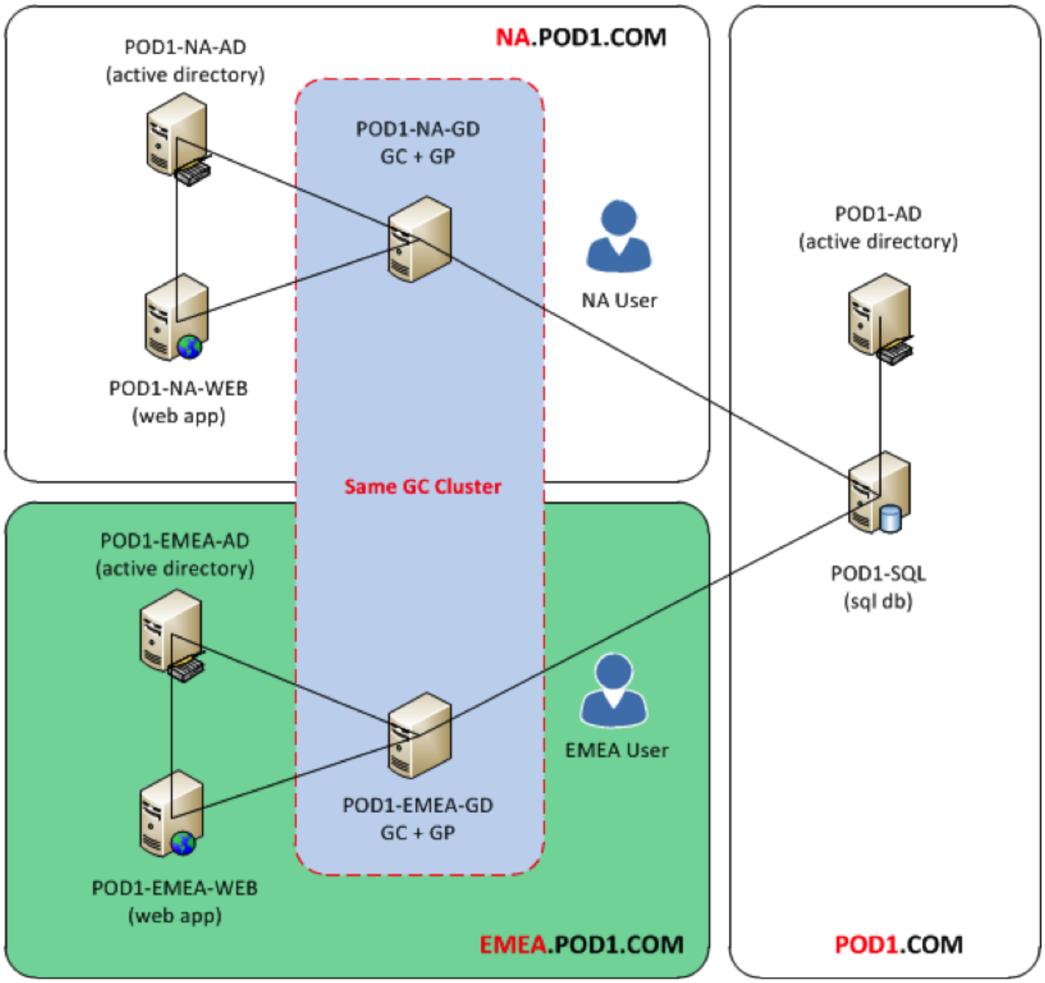
If you are using Microsoft SQL Server in the multi-realm configuration, make sure of the following:

- Use an SQL account, not a Microsoft Windows service account.
- That SQL account must be the owner of the database.
- All GCs in the multi-realm KCD configuration must use that same SQL account.

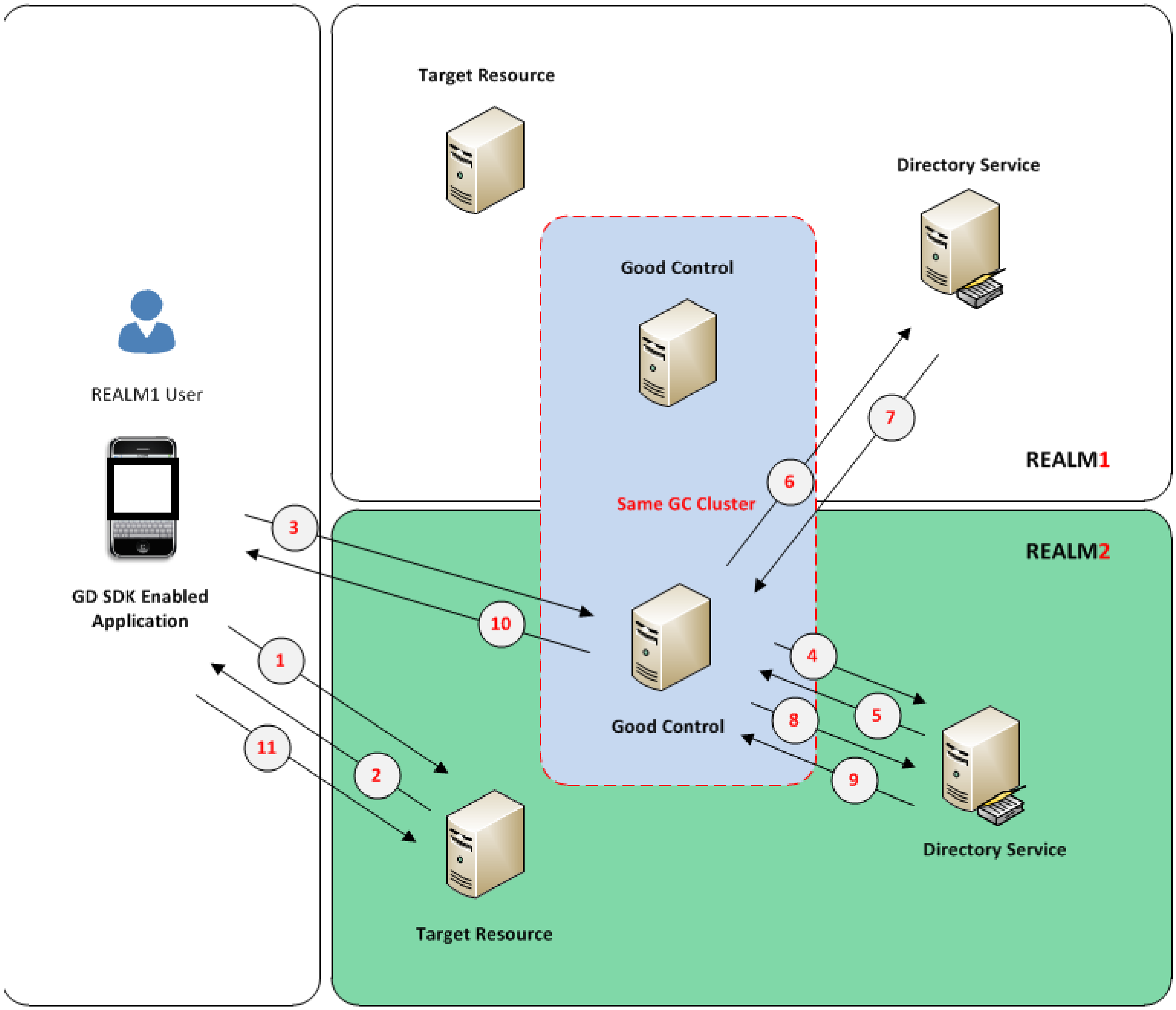
Concepts and high-level steps for single forest

Consider the trust relationships among the domains shown below:


POD1.COM
(Parent Domain)
Direct Trust
Direct Trust
NA.POD1.COM
(Child Domain)
EMEA.POD1.COM
(Child Domain)



The flow with KCD mutli-realm topology is like this:



1. A user in realm 1 with a BlackBerry Dynamics SDK enabled application makes a request to a resource in realm 2
2. The target resource replies with an authentication challenge that the BlackBerry Dynamics runtime intercepts
3. Base on the challenge information, the BlackBerry Dynamics runtime sends a request to a GC that resides in the same realm as the target resource. The request is for a service ticket to access the target resource.
4. GC authenticates the user/container (via internal BlackBerry Dynamics protocols) and asks for a service ticket on behalf of the user (this is delegation) for the service on the target host.
5. Active Directory (AD) checks its local policy and determines that the user is in a remote realm. AD returns a referral to the GC to contact the remote AD for authorization.
6. GC follows the referral to the new AD server for authorization.

7. The realm 1 AD server receives and authorization request. Base on itsí policy, the AD server will return the appropriate authorization response.
8. The GC sends the authorization response to the original AD in realm 2.
9. Base on the authorization response, (a) if the user has permission to access the resource on the target host and (b) if the resource on the target hose is allowed then AD returns to Good Control a service ticket for the resource
10. Good Control sends the necessary information from the returned service ticket to the BlackBerry Dynamics runtime.
11. The BlackBerry Dynamics runtime uses the information from GC to complete the authentication to the target host.

Configuration steps

The steps here are high-level. They rely on information from Concepts and steps for a typical KCD single realm installation and on other sources, as indicated below.

Step	See Also…
Determine the name of your service account to use in the other steps.	
Add Service Account as a administrator in Good Control.	Good Control help
Add Service Account that is actual Owner of the GC database.	
Install GC/GP to the same cluster.	BlackBerry Dynamics Server Installation guide
Create SPN for Service Account and HTTP Services	Step 1: map the Good Control (GC) service account to a service principal name (SPN)
Create Keytab file for Service Account.	Step 2: changing the Kerberos constrained delegation (KCD) keytab password
Configure constrained delegation for Service Account.	Step 3: configure constrained delegation for the desired resources
Add Service Account to AD Group: “Windows Authorization Access Group”.	Step 4: enable enumeration of AD user objects group membership
Add Service Account to Local Security Group: “Act as part of Operating System”.	Step 5: Enable the GC service account to act as part of the operating system
Create krb5.conf file. Only needed if there is a CAPATH trust	Example krb5.conf files for CAPATH trust
Update GC Configuration.	
Configure Test KCD website for testing (optional)	

Configuring Kerberos-related properties in Good Control

In GC's **Server** > **Server Properties** tab, the following settings are required to enable KCD in Good Control.

Property	Optional/Required	Description
`gc.krb5.enabled`	Required	Check this to enable KCD.
The first four properties listed below must be set when gc.krb5.enabled = true.		
	`gc.krb5.kdc=`***`kdc_host_name`*** Required	***`kdc_host_name`*** is the fully qualified name for the KDC. Usually corresponds to the FQDN of an Active Directory Domain controller.
	`gc.krb5.principal.name=`***`gc_service_account`*** Required	***`gc_service_account`*** is the service account name under which the KCD service is running.
	`gc.krb5.realm=`***`REALM`*** Required	***`REALM`*** is the name of the Active Directory realm. Example: **`QAGOOD.COM`**. **Note:** The value must be in all capital letters.
	`gc.krb5.keytab.file=`***`keytab_file_location`*** Required	***`keytab_file_location`*** is the location of the keytab file. Example: **`C:/good/gdadmin.keytab`**. **Note:** Do not use backslashes in this path name. Use normal slashes.

Example krb5.conf files for CAPATH trust

Below is a sample krb5.conf file, which is needed to establish the CAPATH trust relationships of multiple Kerberos domains.

The location of the krb6.conf file on the GC server must be specified in the GC server property gc.krb5.config.file. See Configuring Kerberos-related properties in Good Control .

```
[libdefaults]
default_realm = NA.POD1.COM
[realms]
NA.POD1.COM = {
kdc = pod1-na-ad.na.pod1.com
}
[capaths]
NA.POD1.COM = {
APAC.POD2.COM = POD2.COM
POD2.COM = POD1.COM
POD1.COM = .
}
POD2.COM = {
NA.POD1.COM = POD1.COM
POD1.COM = .
}
APAC.POD2.COM = {
NA.POD1.COM = POD1.COM
POD1.COM = POD2.COM
POD2.COM = .
```

Troubleshooting and diagnostics

BlackBerry Dynamics diagnostic aids are in place to help you troubleshoot real or suspected KCD problems by detecting and/or reporting issues that your system administrator can either fix or else refer the known details to BlackBerry technical support for timely investigation and resolution.

Kerberos and KCD log file error codes

Information captured in your GC server logs can often help to explain Kerberos authentication and KCD issues/errors. Here's an example of a Kerberos error log.

```
   com.good.gmc.security.kerberos.KerberosException: Failed to impersonate userPrincipal tanu100@BlackBerry
Dynamics.QAGOOD.COM;
        krbErrCode: 6;
        krbErrText: Client not found in Kerberos database
        at com.good.gmc.security.kerberos.impl.KerberosServiceImpl.impersonateUser(KerberosServiceImpl.java:211)
        at com.good.kcd.TicketProxy2$ProxyHandler.fetchServiceTicket(TicketProxy2.java:168)
        at com.good.kcd.TicketProxy2$ProxyHandler.messageReceived(TicketProxy2.java:145)
        at org.apache.mina.core.filterchain.DefaultIoFilterChain$TailFilter.messageReceived(DefaultIoFil-
terChain.java:716)
        .
        .
        .
```

The two most important parameters in the error messagesare **krbErrCode** and **krbErrText**, which furnish a description of possible error conditions detected.

Complete documentation for Microsoft's Kerberos error messages is available at http://technet.microsoft.com/en-us/library/bb463166.aspx .