RoamServer 6.1.0 Windows Admin Guide

VERSION 1.0, NOVEMBER 2015

Corporate Headquarters
iPass Inc.
3800 Bridge Parkway
Redwood Shores, CA 94065 USA

www.ipass.com
+1 650-232-4100
+1 650-232-0227 fx

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

Introduction

The RoamServer 6.1.0 for Windows Server Administrator Guide provides instructions for installing RoamServer 6.1.0 for Windows Server 2008 (both 32 and 64 bit) and Windows Server 2012 (64 bit). It also includes instructions on how to configure RoamServer to use RADIUS, LDAP, TACACS+, and NT as authentication protocols.

These instructions often refer to the directory called **<RS_Home>**. This is the directory in which RoamServer is installed. The default for RoamServer 6.1.0 is **C:/ipass/roamserver/6.1.0**

Important

- RoamServer 6.1.0 for Windows Server replaces RoamServer 6.0.0, 5.2.1 & 5.1.1
- RoamServer 6.1.0 should only be downloaded on Windows Server 2008 and 2012.
  - If you are using Windows Server 2003 or Windows Server 2000, you should download RoamServer 5.2.1 (for Server 2003) RoamServer 5.1.1. (for Server 2000). However, RoamServer 6.1.0 can also be installed in customized path (e.g., d:\ipass)

System Requirements

Server Requirements

- Processor
- 1GHz (x86 processor)
- 1.4GHz (x64 processor)
- Memory
- Minimum 512MB
- Disk Space
- Minimum (32-bit): 500MB or greater
- Minimum (64-bit): 500MB or greater
- Server must have an accessible IP address
- Installer must have administrative privileges on the machine

Additional Requirements

- Connectivity to an authentication database.
- The TCP/IP protocol is required to support the SSL-encrypted connection from the iPass Transaction Centers.
- RoamServer must be installed on at least two separate host machines to insure the iPass redundancy model is enabled
- iPass Transaction Centers must be able to communicate with RoamServer on port 577. Please refer to the help page below for a list of Transaction Center IP addresses.

http://help.ipass.com/doku.php?id=required_configurations_for_open_mobile_access#roamserver1

Preferences

The following are not required, but strongly encouraged:

- **DO NOT** install Device Management and RoamServer on the same physical host since the two applications use different security models.

> *Inbound Internet access to RoamServer is secured by restricting inbound Internet access to a single port, and a small set of IP addresses. However, a Device Management server, by its nature, must allow universal inbound Internet access on standard HTTP/SSL ports, since remote Device Management agent IP addresses will be unknown. Device Management server is also secure, but simply implements more security at the application level instead of the network level. Combining locked-down server model and a wide-open server model on the same physical host results in a wide-open model, because security uses a "weakest link" paradigm.*

Supported Platforms

- Windows Server 2008 Service Pack 2 (32-bit)
- Windows Server 2008 R2 (64-bit)
- Windows Server 2012 R2 (64-bit)

Default Port

The default RoamServer port is 577. Consult with iPass before using another port number.

Installation

Prerequisites

Before installing RoamServer, you will need the following:

- Administrator privileges on the RoamServer host.
- Your iPass Customer ID.
- Your host's private and public IP addresses.
- The port number on which the RoamServer will listen (should be 577).
- The host's operating system version and Service Pack, if any.

General Process

To install RoamServer:

1. Download the installation file.
2. Install the software.
3. Set initial configuration.
4. Certify RoamServer by importing signed certificates into the RoamServer keystore.
5. Set additional properties in the **ipassRS.properties file**, if necessary.
6. Test the installation.
7. Repeat as needed to install RoamServer on additional servers. (See the Failover section for more information).

Installing Behind a Firewall

iPass recommends that you install RoamServer behind a firewall. If you choose to do so, you will need to allow TCP traffic to the external IP of RoamServer on port 577 through to RoamServer. The internet-facing IP must be registered with iPass. You may restrict traffic on that port to incoming packets only from the IP addresses of the iPass Transaction Centers. Please refer to the list of iPass Transaction Centers here:

http://help.ipass.com/doku.php?id=required_configurations_for_open_mobile_access#roamserver1

If your firewall is performing Network Address Translation (NAT), you will need to provide the IP address of your firewall to your iPass Installation Engineer.

Downloading the Installer

Before installing, you will need to download the installation file from the iPass FTP site, ftp.ipass.com.

To download the installation file using FTP:

1. Open a Windows command prompt and Change Directory **(cd)** to **C: \.**

2. Type: **ftp ftp.ipass.com**
3. Enter the username: **roamserver**
4. Enter the password: **pass2roAm**
5. To change to binary mode, type: **bin**
6. To obtain a complete listing of directory contents, type: **dir**
7. To change to the directory containing the software for your platform and region, type: **CD**
8. After locating the file appropriate to your platform and region, type: **get <filename>**. Remember that directory names and filenames are case-sensitive.
9. To exit the FTP application, type: **bye**

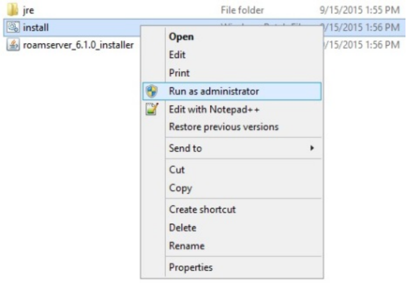
Installing RoamServer (GUI Mode)

These installation instructions are for a machine with Windows Server 2008 Service Pack 2 (32-bit), Windows Server 2008 R2 (64-bit), or Windows Server 2012 R2 (64-bit) that does not have an earlier version of RoamServer.

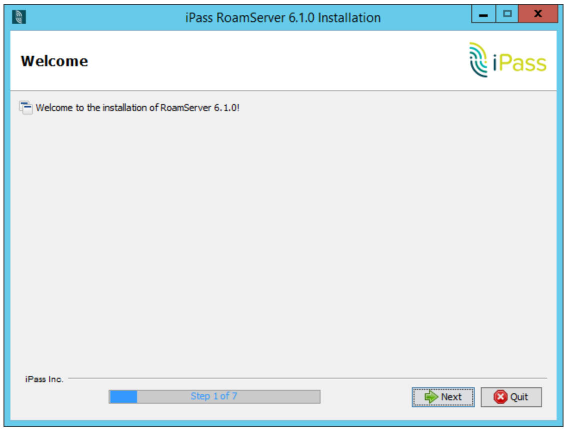
> ***iPass does not recommend*** *installing this binary (roamserver_6.1.0-win-x64.zip, roamserver_6.1.0-win-x86.zip) on any other operating system besides the ones mentioned above.*

To install RoamServer:

1. Extract the **RoamServer zip file** in whichever directory you wish to work.
2. Right-click **install.bat** and select **Run as administrator**.



3. Click **Next** on the installation window that appears.

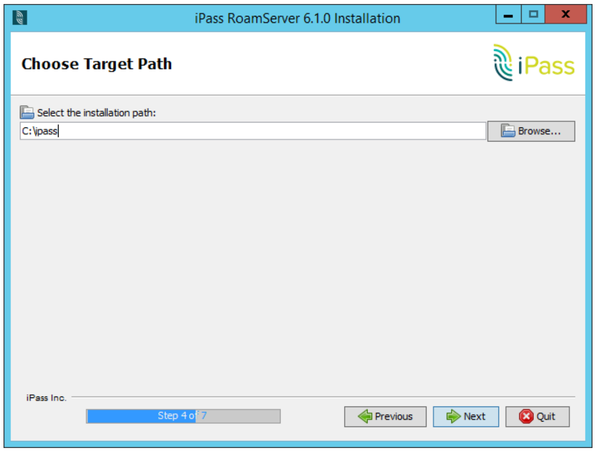


Welcome to the installation of RoamServer 6.1.0!

4. A window will appear asking you to choose a target path. Select the installation path (default path is shown). Alternatively, you can choose a non-default path (**C:\other_dir**). If you provide a non-default path, **other_dir** should be created.

5. Follow further installation prompts for successful installation.

6. Click **Finish** to complete installation.



iPass RoamServer 6.1.0 Installation

Choose Target Path

Select the installation path:

C:\ipass

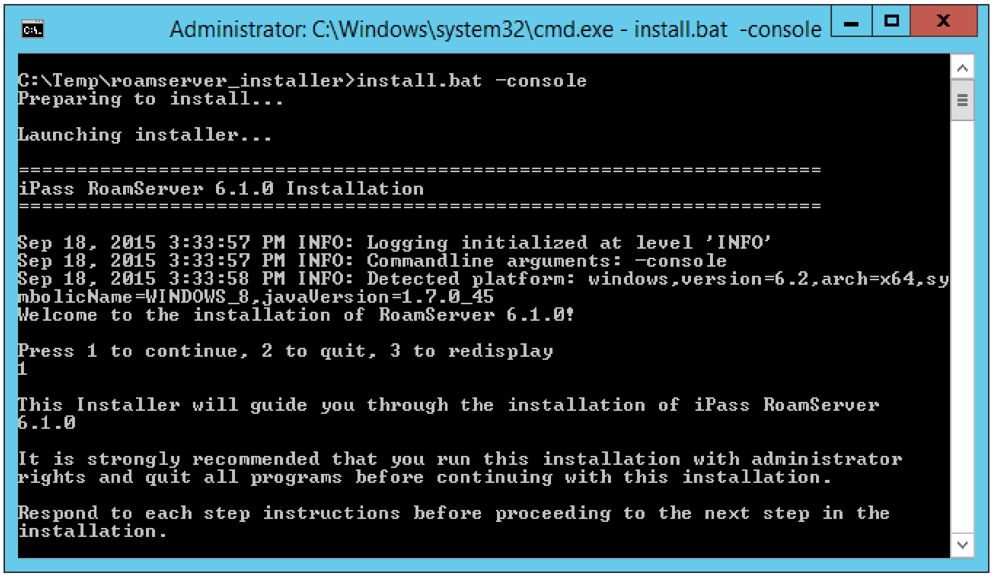
Step 4 of 7

7. After installation is complete, set security permissions on the **<RS_Home>** folder so only Administrators have Write/Execute permissions. Otherwise, non-administrators may be able to change your property files or even shut down RoamServer.

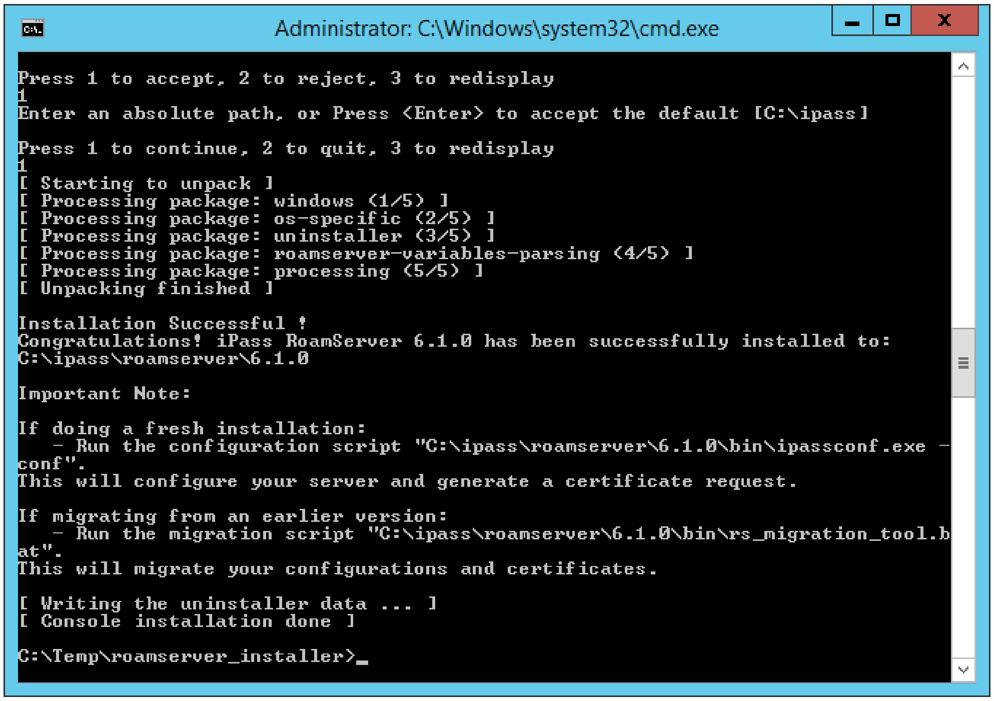
> *If you are not able to install RoamServer using GUI Mode, please try using Console Mode (section below) to install.*

Console Mode Installation (Optional)

1. Open the administrator command line prompt and go to the following path: **C:\temp\roamserver_installer**
2. Execute the following command: **install.bat –console**
3. Choose the default target path C:\ipass, or if you want to install RoamServer under a non-default path then enter proper target path value as **c:\other_dir**
4. After installation is complete, set security permissions on the **<RS_Home>** folder so only Administrators have Write/Execute permissions. Otherwise, non-administrators may be able to change your property files or even shut down RoamServer.


Administrator: C:\Windows\system32\cmd.exe - install.bat -console

```
C:\Temp\roamserver_installer>install.bat -console
Preparing to install...

Launching installer...

=====================================================================
iPass RoamServer 6.1.0 Installation
=====================================================================

Sep 18, 2015 3:33:57 PM INFO: Logging initialized at level 'INFO'
Sep 18, 2015 3:33:57 PM INFO: Commandline arguments: -console
Sep 18, 2015 3:33:58 PM INFO: Detected platform: windows,version=6.2,arch=x64,sy
mbolicName=WINDOWS_8,javaVersion=1.7.0_45
Welcome to the installation of RoamServer 6.1.0!

Press 1 to continue, 2 to quit, 3 to redisplay
1

This Installer will guide you through the installation of iPass RoamServer
6.1.0

It is strongly recommended that you run this installation with administrator
rights and quit all programs before continuing with this installation.

Respond to each step instructions before proceeding to the next step in the
installation.
```



```
Press 1 to accept, 2 to reject, 3 to redisplay
1
Enter an absolute path, or Press <Enter> to accept the default [C:\ipass]

Press 1 to continue, 2 to quit, 3 to redisplay
1
[ Starting to unpack ]
[ Processing package: windows (1/5) ]
[ Processing package: os-specific (2/5) ]
[ Processing package: uninstaller (3/5) ]
[ Processing package: roamserver-variables-parsing (4/5) ]
[ Processing package: processing (5/5) ]
[ Unpacking finished ]

Installation Successful !
Congratulations! iPass RoamServer 6.1.0 has been successfully installed to:
C:\ipass\roamserver\6.1.0

Important Note:

If doing a fresh installation:
    - Run the configuration script "C:\ipass\roamserver\6.1.0\bin\ipassconf.exe -
conf".
This will configure your server and generate a certificate request.

If migrating from an earlier version:
    - Run the migration script "C:\ipass\roamserver\6.1.0\bin\rs_migration_tool.b
at".
This will migrate your configurations and certificates.

[ Writing the uninstaller data ... ]
[ Console installation done ]

C:\Temp\roamserver_installer>_
```

Updating RoamServer

Updating RoamServer 5/6.x to 6.1.0

To update RoamServer 5.x to RoamServer 6.1.0 on Win Server 2008 Service Pack 2 (32-bit), Win Server 2008 R2 (64-bit):

1. Stop the RoamServer 5.x service by opening the command line prompt and running the following command:

   If migrating from RoamServer 5.x,<**RS_Home>\5.x\bin\roamserver_srvc.exe –stop**

   If migrating from RoamServer 6.x **<RS_Home>\6.x\bin\roamserver_srvc.exe –stop**

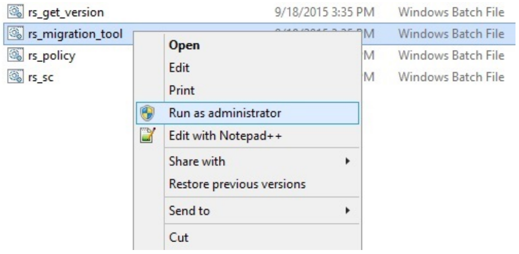
2. Remove the RoamServer 5.x service by running the following command:

   If migrating from RoamServer 5.x **<RS_Home>\5.x\bin\roamserver_srvc.exe –remove**

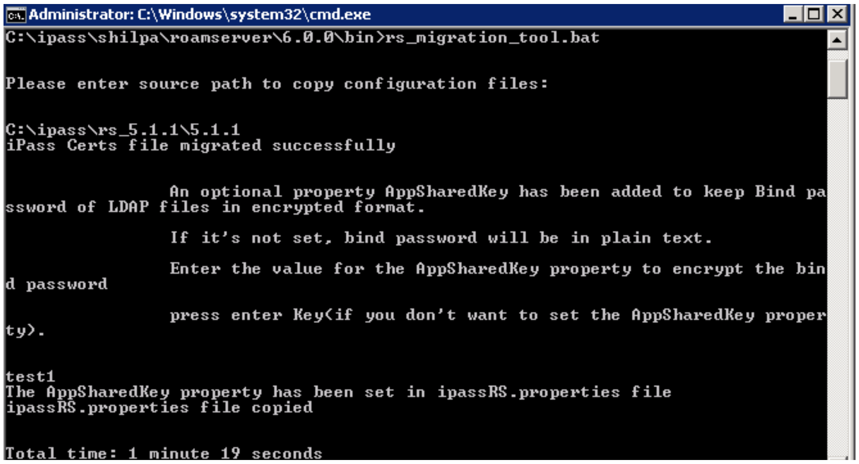
   If migrating from RoamServer 6.x **<RS_Home>\6.x\bin\roamserver_srvc.exe –remove**

3. Install RoamServer 6.1.0 under **C:\ipass**

4. Open the command prompt and run the following command for migration:



- Go to <RS_HOME>\6.1.0\bin\
- Right-click rs_migration_tool.bat and Run as administrator.
- Enter source path as: **<RS_HOME>\5.x**
- If migrating from RoamServer 5.x, and you want encryption for your LDAP bind password configured in the **ipassLDAP.properties** file, then enter a value for the property named **AppSharedKey** (Press Enter at the prompt if you don't want to set the AppSharedKey).
-



```
C:\ipass\shilpa\roamserver\6.0.0\bin>rs_migration_tool.bat

Please enter source path to copy configuration files:

C:\ipass\rs_5.1.1\5.1.1
iPass Certs file migrated successfully

                An optional property AppSharedKey has been added to keep Bind pa
ssword of LDAP files in encrypted format.
                If it's not set, bind password will be in plain text.
                Enter the value for the AppSharedKey property to encrypt the bin
d password
                press enter Key(if you don't want to set the AppSharedKey proper
ty).

test1
The AppSharedKey property has been set in ipassRS.properties file
ipassRS.properties file copied

Total time: 1 minute 19 seconds
```

- These files are migrated in RoamServer 6.1.0:
- ipassRS.properties

- If migrating from RoamServer 5.x, isp_cert.pem and isp_key.pem certificates were converted to the rs.keystore file If migrating from RoamServer 6.x, configured keystore is migrated in respective location.
- Configured LDAP property file(s) (if available in RoamServer 5.x/6.x).
- Configured Policy property file (if available in RoamServer 5.x/6.x).

5. Go to **<RS_HOME>\6.1.0\certs** and verify rs.keystore exists (isp_cert.pem and isp_key.pem got converted to rs.keystore during migration).

6. Open **<RS_Home>\6.1.0\ipassRS.properties** file to check for:

   - new property 'KeyStore'.

   KeyStore=KeyStorePath=C:/ipass/roamserver/6.1.0/certs/rs.keystore

   - Replace version number 5.x with 6.1.0 as listed attributes below (in bold):
   - **AcctServer1**=protocol=AcctFile,localAcctFileName= C:/ipass/roamserver/**6.1.0**/logs/acct.log,IncludeDomainAsWinPrefix=No
   - **AuthServer1**=protocol=**LDAP**,ipaddress=127.0.0.1,port=389,LdapConfigFile= C:/ipass/roamserver/**6.1.0**/ipassLDAP.properties,IdleTimeout=10000,enableSsl=No
   - **PolicyFile**=C:/ipass/roamserver/**6.1.0**/policy.txt

7. AppSharedKey property is added in ipassRS.properties file if value is entered during migration.

   AppSharedKey=test1I

8. Open <RS_Home>\6.1.0\ipassRS.properties file to check for "Keystore" property. For example: KeyStore=KeystorePath, e.g., C:/pass/roamserver/6.1/0/certs/rs.keystore

9. Start RoamServer 6.1.0 using GUI Mode by going to the **Start** menu > **Programs > iPass RoamServer 6.1.0** > **Start RoamServer**.

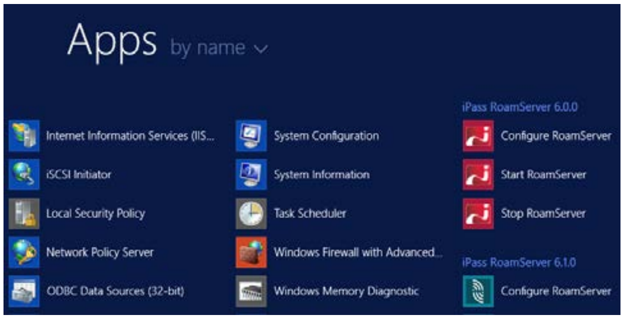
   (Optional) You can also start RoamServer using Console Mode.

   - Open the command prompt and go to:

   <RS_HOME>\6.1.0\bin

   - Execute the following command:

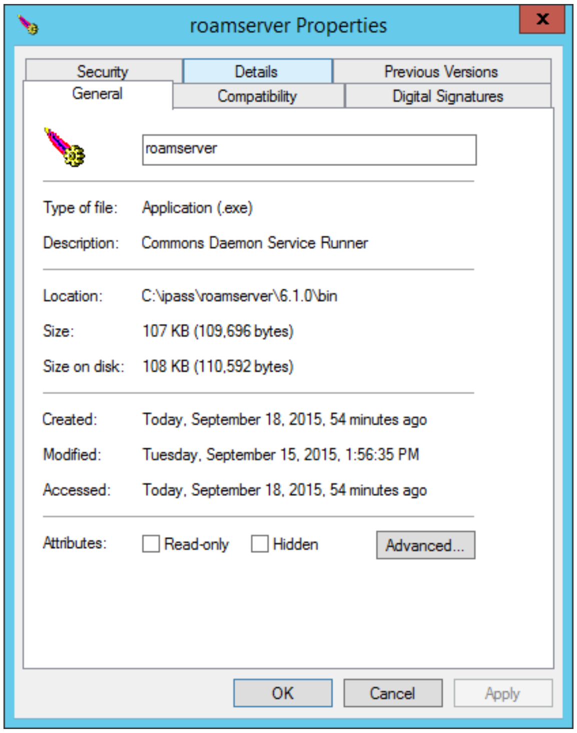
   roamserver_srvc.bat –start



Verification

To determine which RoamServer service installed after an update:

1. Right click on the service name listed under service control manager, **iPass RS Service**.
2. The path under **Path to executable** should be: **<RS_Home>\6.1.0\bin\roamserver.exe**.



roamserver

Type of file: Application (.exe)

Description: Commons Daemon Service Runner

Location: C:\ipass\roamserver\6.1.0\bin

Size: 107 KB (109,696 bytes)

Size on disk: 108 KB (110,592 bytes)

Created: Today, September 18, 2015, 54 minutes ago

Modified: Tuesday, September 15, 2015, 1:56:35 PM

Accessed: Today, September 18, 2015, 54 minutes ago

Attributes: □ Read-only □ Hidden

Note on Admin Privilege

On machines running Windows Server 2008 and 2012, a user can run (**Start/Stop/Configure**) RoamServer and use other RoamServer command line options by right-clicking and selecting **Run as Administrator**. To avoid this, the Administrator needs to disable the **User Access Control** (UAC) on the machine.

Installation Issues

The RoamServer 6.1.0 installer should only be installed on Windows Server 2008 or Windows Server 2012. Attempting to run this installer on another operating system may yield incomplete installation results and is not recommended.

There is another installer file on the iPass FTP site for RoamServer 5.1.1 (for the Windows 2003 Server and Windows 2000 platforms).

Determining RoamServer Installer

If **roamserver_6.1.0-win-x64.zip/roamserver_6.1.0-win-x86.zip** is downloaded and then renamed so that it no longer indicates Windows Server 2008 in the filename, you can use the following methods to determine which installer was downloaded:

JDK Version

- On the command line, type:

```
C:\temp\roamserver_installer\jre\bin\java –version.
```

> *The JRE version should display as: **1.7.0_45**.*

- After installing RoamServer, check the version information by executing:

```
<RS_Home>/6.1.0/bin/rs_get_version.bat
```

> *It will provide build and JRE version of RoamServer 6.1.0*

Uninstalling

To uninstall RoamServer through the Control Panel:

1. Open the **Windows Control Panel**.
2. Select **iPass RoamServer 6.1.0** and follow the prompts to uninstall.

> *You may also need to manually delete any leftover files in the* ***<RS_Home>6.1.0*** *folder that were not created by the installer.*

Setup

When first running RoamServer 6.1.0, you will need to complete the installation by setting it up as a Windows NT service. You must also perform some basic setup tasks and receive a digital certificate from iPass.

Configuring RoamServer

Basic Server Information

To enter basic server information:

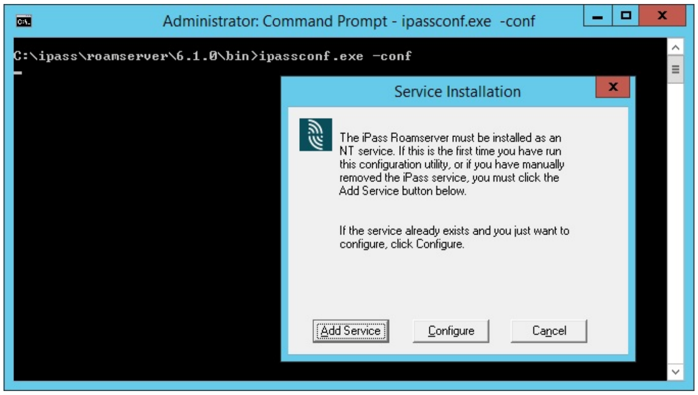
1. After installation, you need to launch the RoamServer configurator through the console in the RoamServer's bin directory. If required, navigate to <RS_HOME\6.1.0\bin.

   cd <RS_HOME>\6.1.0\bin

2. Run ipassconf script:

   ipassconf.exe –conf

3. The **Service Installation** dialog box will open. Click the **Add Service** button.

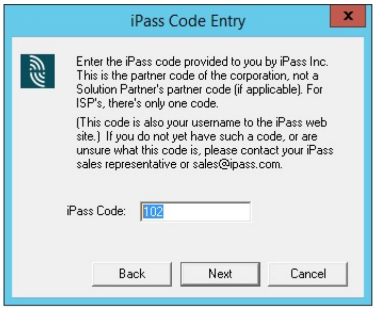


```
C:\ipass\roamserver\6.1.0\bin>ipassconf.exe -conf
```

The iPass Roamserver must be installed as an NT service. If this is the first time you have run this configuration utility, or if you have manually removed the iPass service, you must click the Add Service button below.

If the service already exists and you just want to configure, click Configure.

4. On the iPass Code Entry dialog box, enter the iPass code provided to you, and then click **Next**. This code is also your Customer ID for the iPass Portal. If you do not have such a code, or are unsure what the code is, please contact your iPass representative.

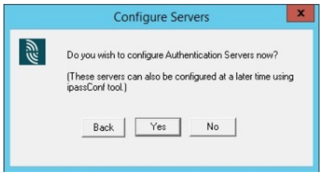


Enter the iPass code provided to you by iPass Inc. This is the partner code of the corporation, not a Solution Partner's partner code (if applicable). For ISP's, there's only one code.

(This code is also your username to the iPass web site.) If you do not yet have such a code, or are unsure what this code is, please contact your iPass sales representative or sales@ipass.com.

iPass Code: 102

5. If you want to configure your authentication servers now, click **Yes** in the Configure Servers dialog box.

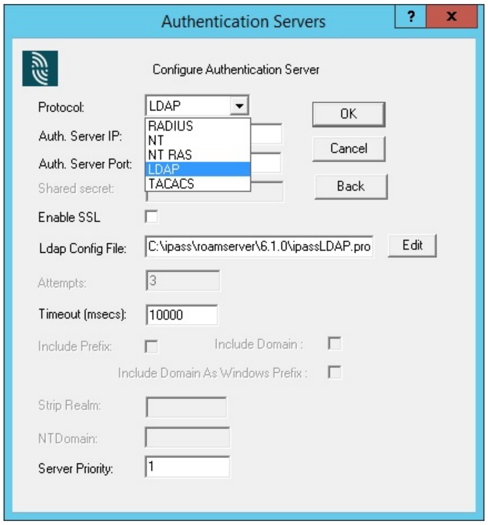


Do you wish to configure Authentication Servers now?

(These servers can also be configured at a later time using ipassConf tool.)

To configure an authentication server,

- Open the Protocol drop down and select an authentication protocol on the Authentication Server dialog box.



Configure Authentication Server

Protocol: LDAP

Auth. Server IP:

Auth. Server Port:

Shared secret:

Enable SSL □

Ldap Config File: C:\ipass\roamserver\6.1.0\ipassLDAP.pro

Attempts: 3

Timeout (msecs): 10000

Include Prefix: □ Include Domain : □

Include Domain As Windows Prefix : □

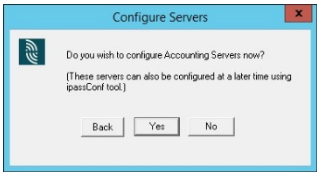
Strip Realm:

NTDomain:

Server Priority: 1

- Enter as much data as possible on the authentication server. (The details of required fields will depend on the protocol you choose. See Authentication Servers, on page 19.)
- When complete, click **OK**.
- If you wish to configure another server, click **Yes** in the Configure Servers dialog box.
- Repeat until all of your authentication servers are configured.

6. If you want to configure your accounting servers now, click **Yes** in the Configure Servers dialog box.

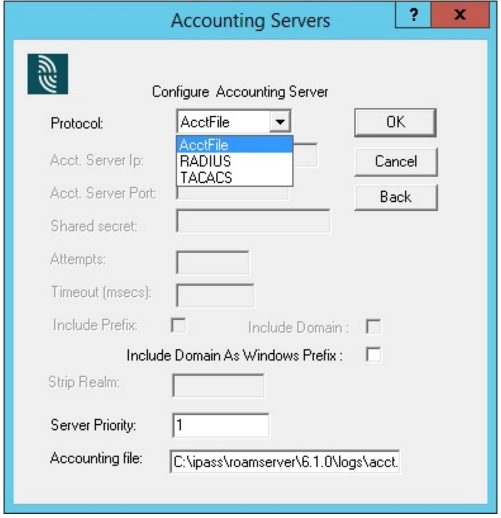


Do you wish to configure Accounting Servers now?

(These servers can also be configured at a later time using ipassConf tool.)

To configure an accounting server,

- Open the Protocol drop down and select an accounting protocol on the Accounting Server dialog box.
- Enter as much data as possible on the accounting server. (The details of required fields will depend on the protocol you choose.)
- When complete, click **OK**.
- If you wish to configure another server, click **Yes** in the Configure Servers dialog box.
- Repeat until all of your accounting servers are configured.
- Complete instructions for configuring your authentication and accounting servers using RADIUS, LDAP or TACACS+ can be found in Authentication Servers. You may configure servers at any time later on by running the **ipassconf** utility, found in your **<RS_Home>\bin** folder.



Configure Accounting Server

Protocol: AcctFile

Acct. Server Ip:

Acct. Server Port:

Shared secret:

Attempts:

Timeout (msecs):

Include Prefix: □ Include Domain : □

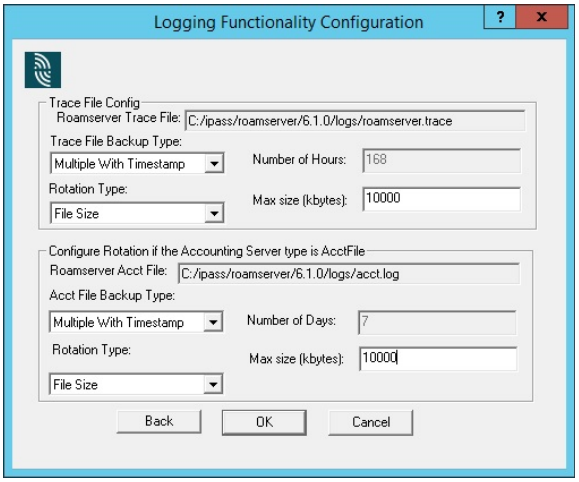
Include Domain As Windows Prefix : □

Strip Realm:

Server Priority: 1

Accounting file: C:\ipass\roamserver\6.1.0\logs\acct.

7. In the Logging Functionality Configuration dialog box, select locations for your trace and log files, as well as the type of rotation you wish to use. Click **OK** when done. For more information, see Configuration Options.



Logging Functionality Configuration

Trace File Config

Roamserver Trace File: C:/ipass/roamserver/6.1.0/logs/roamserver.trace

Trace File Backup Type: Multiple With Timestamp

Number of Hours: 168

Rotation Type: File Size

Max size (kbytes): 10000

Configure Rotation if the Accounting Server type is AcctFile

Roamserver Acct File: C:/ipass/roamserver/6.1.0/logs/acct.log

Acct File Backup Type: Multiple With Timestamp

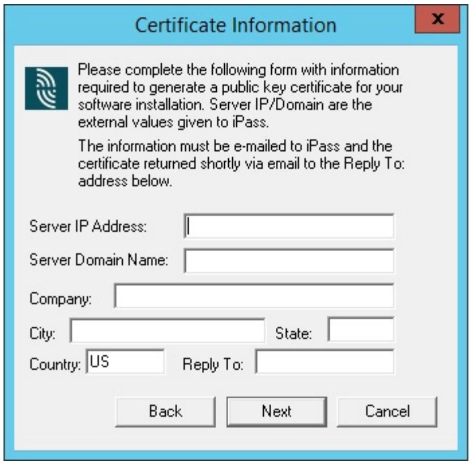
Number of Days: 7

Rotation Type: File Size

Max size (kbytes): 10000

8. In the Certificate Information dialog box, enter your server IP address, domain name, the name of your company, city, state, and country.

9. In the Reply To field, enter an e-mail address, and then click **Next**.



Certificate Information

Please complete the following form with information required to generate a public key certificate for your software installation. Server IP/Domain are the external values given to iPass.

The information must be e-mailed to iPass and the certificate returned shortly via email to the Reply To: address below.

Server IP Address:

Server Domain Name:

Company:

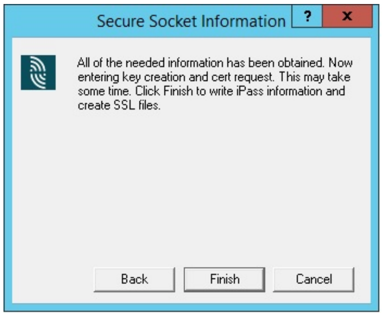
City:

State:

Country: US

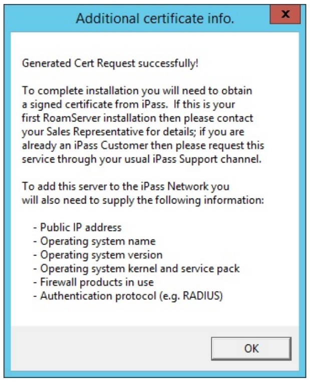
Reply To:

10. The RoamServer keystore and certification request will now be created. Click **Finish** and then **OK**.



Secure Socket Information

All of the needed information has been obtained. Now entering key creation and cert request. This may take some time. Click Finish to write iPass information and create SSL files.



Additional certificate info.

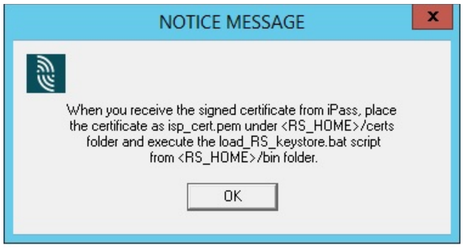
Generated Cert Request successfully!

To complete installation you will need to obtain a signed certificate from iPass. If this is your first RoamServer installation then please contact your Sales Representative for details; if you are already an iPass Customer then please request this service through your usual iPass Support channel.

To add this server to the iPass Network you will also need to supply the following information:

- Public IP address
- Operating system name
- Operating system version
- Operating system kernel and service pack
- Firewall products in use
- Authentication protocol (e.g. RADIUS)

11. You will receive a prompt instructing you where to place the certificate and how to execute the load_RS_keystore.bat file. message is shown to import signed cert into RoamServer keystore.



NOTICE MESSAGE

When you receive the signed certificate from iPass, place the certificate as isp_cert.pem under <RS_HOME>/certs folder and execute the load_RS_keystore.bat script from <RS_HOME>/bin folder.

Certificate Request

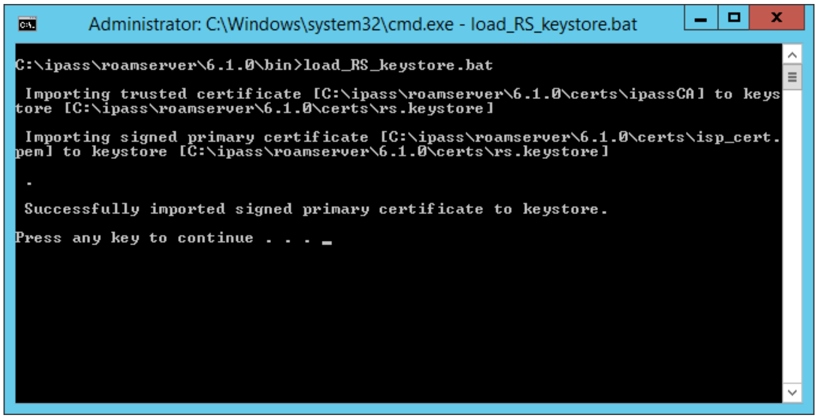
After entering your basic server information, you must submit a request for a signed certificate to iPass. The x509 certificate will allow SSL 128-bit encrypted communication between the iPass transaction server and the RoamServer.

To submit your certification request:

Log into the iPass Portal and open a Support Ticket requesting a signed RoamServer certificate. Please provide **mail_cert_req.data**, which you will find in the RoamServer's **certs** directory.

To finish the certification process:

1. iPass will send you an email containing instructions for downloading a certificate.
2. Import your signed cert into RS keystore to complete certification process.
    - Go to path: <RS_HOME>\6.1.0\bin
    - Execute load_RS_keystore.bat script from <RS_HOME>\6.1.0\bin folder



Administrator: C:\Windows\system32\cmd.exe - load_RS_keystore.bat

```
C:\ipass\roamserver\6.1.0\bin>load_RS_keystore.bat

 Importing trusted certificate [C:\ipass\roamserver\6.1.0\certs\ipassCA] to keys
tore [C:\ipass\roamserver\6.1.0\certs\rs.keystore]

 Importing signed primary certificate [C:\ipass\roamserver\6.1.0\certs\isp_cert.
pem] to keystore [C:\ipass\roamserver\6.1.0\certs\rs.keystore]

 .

 Successfully imported signed primary certificate to keystore.

Press any key to continue . . . _
```

LDAP Bind Password Encryption

By default, the LDAP bind password is stored in clear text in the ipassLDAP.properties file. An optional property can be added to encrypt the bind password.

1. Open ipassRS.properties file in a text editor as **Administrator**.

2. At the bottom of the file add the key **AppSharedKey=** followed by a sting of letters and numbers. This string will be the symmetrical encryption key for the LDAP bind password.

    - **Example**: AppSharedKey=foo123

If this key is not added to the ipassRS.properties file, the LDAP bind password will remain clear text.

ACA Support

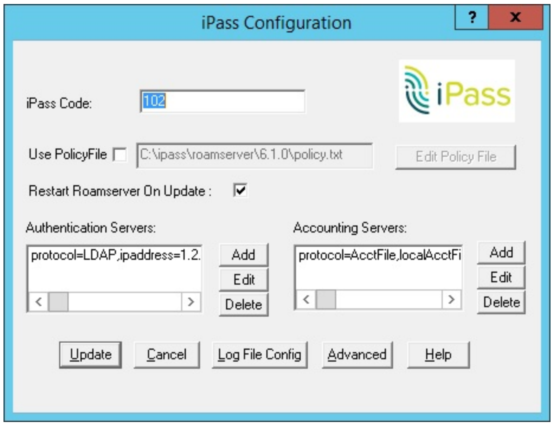
By default, RoamServer supports ACA with Active Directory. For LDAP, **LdapUacAttr** property **must be** configured in the LDAP properties file.
LdapUacAttr=<UserAccountControl attribute name>=<active users UserAccountControl Attribute values>

Example:
LdapUacAttr=ipassStatus=active

For details about LdapUacAttr attribute, refer to the ipassLDAP.properties section.

Starting RoamServer

1. From the Windows Start Menu, go to: **Programs** > **iPass RoamServer 6.1.0** > **Configure RoamServer**.
2. Confirm that the iPass code, hostname, and source IP address is correct.
3. Verify that the box labeled **Restart RoamServer on Update** is checked, and click **Update**. The RoamServer will automatically restart and the changes will take effect. If this box is not checked, you must manually restart the RoamServer before any changes will take effect.



iPass Code: 102

Use PolicyFile ☐ C:\ipass\roamserver\6.1.0\policy.txt

Restart Roamserver On Update : ☑

Authentication Servers: protocol=LDAP,ipaddress=1.2.

Accounting Servers: protocol=AcctFile,localAcctFi

When configuration is complete, you should perform the tests described in the following section.

Post-Configuration Testing

These tests that should be performed during every installation of RoamServer to ensure proper functionality:

- checkipass tool
- RoamServer Test Tool

When testing RoamServer, it is recommended that you perform both of these tests in the order that they are presented here.

checkipass Tool

The **checkipass** test is a request from the RoamServer to the AAA server, which stays local to your network. To test the RoamServer using the checkipass test, you will need to run the checkipass test program as an administrator.

This test verifies that the RoamServer can authenticate a user by communicating with the AAA server. This procedure only tests the RoamServer. No realm should be attached to the user name unless it is required by your AAA. The authentication request goes from the checkipass test to the RoamServer, then to the AAA server for authentication, and finally back to the RoamServer and checkipass program.

checkipass is found in **<RS_Home>\6.1.0\test**. You will need to use a valid user name and password for the host on which the RoamServer is installed.

To run checkipass:

1. Enter the command: **checkipass.bat –u <username>**<Enter>
2. Enter the password when prompted.
3. The results will either be **Accept** or **Reject**.
4. If **status=ack appears at the bottom of the trace**, the RoamServer is properly installed, configured and working, and you may proceed to the next test.

- Possible causes for a **Reject** include:
- Invalid user name or password: The user in this test must have local login privileges to that system.
- Invalid certificate: If the certificate is corrupt, then it will need to be replaced.
- Improper configuration: Verify that you have correctly entered all of the information in the setup program and that your server is running on **port 577**.
- For RADIUS users, invalid shared secret: Verify that your shared secret is entered properly. A shared secret cannot contain the comma (,) or equals sign (=) characters.

RoamServer Test Tool

The RoamServer Test Tool extends the verification performed in the **checkipass** test by sending a authentication request across the iPass network. In this test:

- An authorization request is generated by the tool and sent directly to an iPass Transaction Server, where the domain is used to a discover the Customer ID in the iPass database.

- The Transaction Server forwards this authentication request to the Primary RoamServer at your company on port 577.
- The RoamServer receives the request, and either authenticates locally or forwards the request to your AAA server (RADIUS, LDAP or TACACS+).
- Upon successful authentication, the request is relayed using SSL encryption back to the RoamServer Test Tool.

This test is available as a Web-based tool, and can be reached from openmobile.ipass.com.

To run the RoamServer Test Tool:

1. Log in to openmobile.ipass.com
2. Navigate to the RoamServer Test Tool under the Tools menu
3. Change Test Type to User Authentication Test
4. Change Target Type to point to the IP of the server you wish to test.
5. Enter your iPass user name (with domain name) in the Username field
6. Enter your password in the Password field.
7. Click **Submit**.
8. Scroll to the bottom of the output. Look for an **Accept** or **Reject** response before viewing the rest of the results.

- An **Accept** result means that any user authorized to access your system can now roam on the iPass Network.

In addition to performing this test with a legitimate user name and password, you should also run the test with an invalid user name and password to ensure that the authorization attempt will be rejected.

Running RoamServer

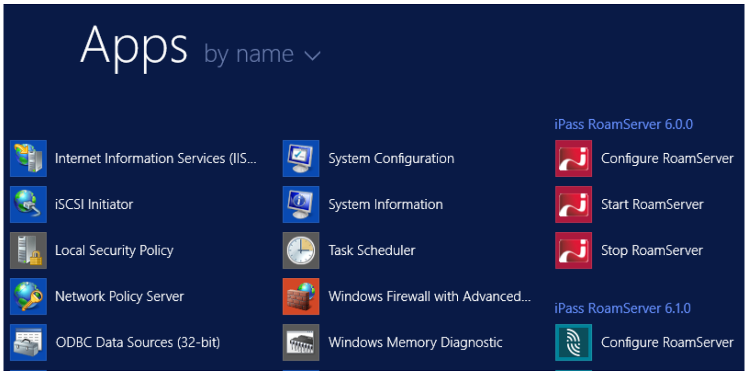
Runtime Commands

Starting RoamServer

On the **Start** menu, navigate to **iPass RoamServer 6.1.0** > **Start RoamServer**.

Shutting Down

Shut down RoamServer by navigating to the **Start** Menu > **iPass RoamServer 6.1.0** > **Stop**.



Restarting After Updates

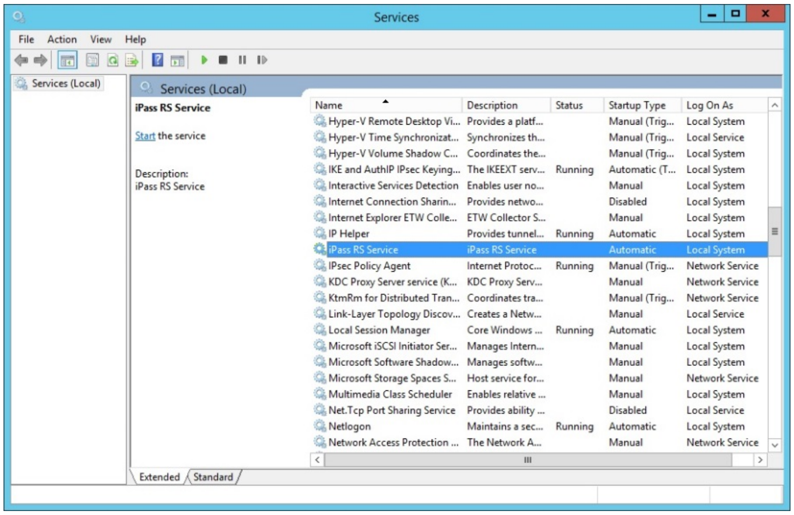
You can restart RoamServer in one of two ways. You can either use the commands above, or alternatively:

1. On the iPass Configuration dialog box, make sure **Restart RoamServer on Update** is selected.
2. Click **Update**. The RoamServer will restart.

This restart method is useful after making configuration changes that you want to implement immediately.

Runtime

The RoamServer will appear in the list of server services. You can verify RoamServer as an active process by navigating to **Start**> **Administrative Tools** > **Services** and locating it in the list as iPass RS Service.



iPass RS Service

Start the service

Description:
iPass RS Service

Name	Description	Status	Startup Type	Log On As
Hyper-V Remote Desktop Vi...	Provides a platf...		Manual (Trig...	Local System
Hyper-V Time Synchronizat...	Synchronizes th...		Manual (Trig...	Local Service
Hyper-V Volume Shadow C...	Coordinates the...		Manual (Trig...	Local System
IKE and AuthIP IPsec Keying...	The IKEEXT serv...	Running	Automatic (T...	Local System
Interactive Services Detection	Enables user no...		Manual	Local System
Internet Connection Sharin...	Provides netwo...		Disabled	Local System
Internet Explorer ETW Colle...	ETW Collector S...		Manual	Local System
IP Helper	Provides tunnel...	Running	Automatic	Local System
iPass RS Service	iPass RS Service		Automatic	Local System
IPsec Policy Agent	Internet Protoc...	Running	Manual (Trig...	Network Service
KDC Proxy Server service (K...	KDC Proxy Serv...		Manual	Network Service
KtmRm for Distributed Tran...	Coordinates tra...		Manual (Trig...	Network Service
Link-Layer Topology Discov...	Creates a Netw...		Manual	Local Service
Local Session Manager	Core Windows ...	Running	Automatic	Local System
Microsoft iSCSI Initiator Ser...	Manages Intern...		Manual	Local System
Microsoft Software Shadow...	Manages softw...		Manual	Local System
Microsoft Storage Spaces S...	Host service for...		Manual	Network Service
Multimedia Class Scheduler	Enables relative ...		Manual	Local System
Net.Tcp Port Sharing Service	Provides ability ...		Disabled	Local Service
Netlogon	Maintains a sec...	Running	Automatic	Local System
Network Access Protection ...	The Network A...		Manual	Network Service

rs_command

You can also perform many runtime functions by using the tool **rs_command, in the <RS_Home>\6.1.0\bin** folder.

Usage: rs_command <command options>

Command Options

-host <IP address>	Specifies the IP address of the machine to send the command to.
-port <port number>	Specifies the server port number to send the command to. Default is the local server's listener port (577).
-shutdown	The server will shutdown.
-restart	The server will restart.
-reload_config	Causes the server to reload many (but not all) of the properties from the ipassRS.properties file. These are: • AutoUpdate flag, used to enable/disable automatic software update. • AAA Servers (AuthServer and AcctServer properties) • Policy Rules, if feature is enabled. • Log Rotation parameters. • DebugLevel of server. For a complete reload, you should use the -restart switch.
-dump_queue	The server will dump the queue elements to a file.
-get <filename> -host <IP address> **-port <port number>**	Get a file from a remote RoamServer. Use filename ipassRS.properties to get the main RoamServer properties file. Use filename RS.trace to get the main RoamServer trace file. Optionally, use any valid filename relative to the RoamServer home folder. <port number> is the port number of the remote host (default port is 577).
-post <Name=value;Name1=value1> -host <IP address> -port <port number>	To post configuration changes on a remote host. where Name=Value pairs are the properties settings separated by a semicolon. (;) <IP address> is the IP address of the remote host, <port number> is the port number of the remote host (default port is 577).
-post_file <file> -host <IP address> -port <port number>	To post configuration changes on a remote host, where <file> contains the configuration changes to be uploaded to the RoamServer, <IP address> is the IP address of the remote host, <port number> is the port number of the remote host (default port is 577).
-version	Print the server version.

Scripts Usage available under <RS_Home>\bin

rs_get_version.bat <options>	To retrieve the version of the roamserver, build number and jre version
config_help.bat <options>	<options> -help <Attribute Name>: Print help/usage for a specific attribute. -list: List the attributes in the server's properties file. -listall : List all of the server's internal attributes.
list_RS_keystore.bat <options>	To list your keystore <options> list_RS_keystore.csh list_RS_keystore.csh <KeyStoreFilePath> list_RS_keystore.csh <KeyStoreFilePath> <Password>
load_RS_keystore.bat	To import the primary signed certificate into rs.keystore
ipass_config_console.bat <options>	<options> -import_cert Use this to import ipass CA certificate and signed primary certificate to a Java keystore -regen_keystore Use this to Regenerate the keystore and certificate signing request

Authentication Servers

This section provides instructions for configuring your iPass RoamServer to communicate with your AAA server.

Once the RoamServer is installed, it can be configured using the iPass RoamServer Configuration tool, ipassconf.exe. To launch the tool, choose **Start** > **Programs** > **iPass RoamServer 6.1.0** > **Configure RoamServer**. You should only have one instance of ipassconf.exe running at any time.

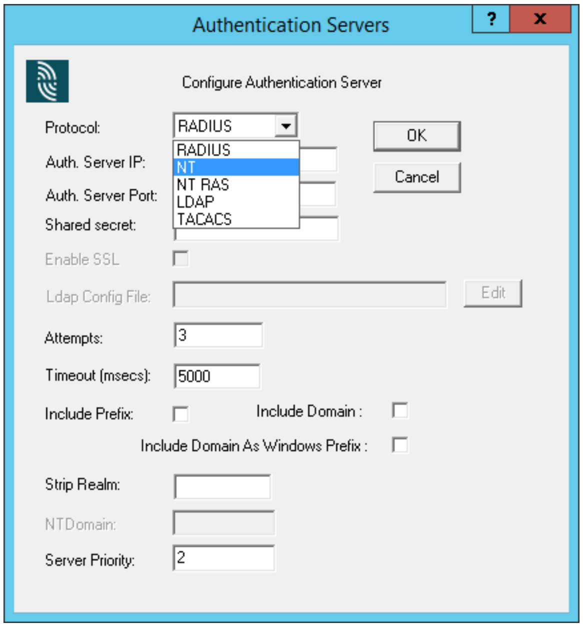
These instructions assume that you are installing RoamServer behind your firewall or on the same host as your AAA server. If you are installing the RoamServer in front of your firewall or even on the firewall, you may need to modify some of these settings. Consult with your iPass RoamServer Installation Engineer for assistance.

Windows Authentication

When authenticating using Windows system passwords, you may use the WinNT or NT RAS protocols. The configuration for either is the same; however, the interaction of the authentication server with the RoamServer will differ between the two. To help you choose the appropriate authentication protocol, consider the following:

- If WinNT is selected as the protocol, Dial-In permissions do not affect the ability for roaming users to authenticate.
- If NT RAS is selected as the protocol, and you are running the system with Dial-In permissions granted, any user with a valid user name and password will be able to authenticate while roaming. However, any user without Dial-In permissions will not be authenticated by the RoamServer.

To configure RoamServer for Windows authentication:



Protocol: RADIUS
Auth. Server IP:
Auth. Server Port:
Shared secret:
Enable SSL □
Ldap Config File:
Attempts: 3
Timeout (msecs): 5000
Include Prefix: □ Include Domain : □
Include Domain As Windows Prefix : □
Strip Realm:
NTDomain:
Server Priority: 2

1. Click **Start** > **iPass RoamServer 6.1.0** > **Configure RoamServer**. The iPass Configuration dialog is displayed.
2. Under Authentication Servers, click **Add**.
3. Under Protocol, select **NT**. (Select **NT RAS** if remote access will be used for authentication.)
4. The following parameters should only be filled in if you are using NT RAS while the RoamServer is installed on a domain member, or if the user's domain is on a different server that is not a trusted domain of the RoamServer host.

- If authentication will include the domain, select the **Include Domain** check box. This may be needed if your organization uses multiple domains.
- If you wish to strip away the realm from the end of the domain before authenticating, enter the realm (e.g. example.com) into **Strip Realm**.
- If authentication will include the NT Domain, enter the domain here. Use this when there is

only one corporate domain for all users.

5. If you use Windows NT authentication, you may also wish to turn on duplicate filtering. See the ipassRS.properties file for more details.

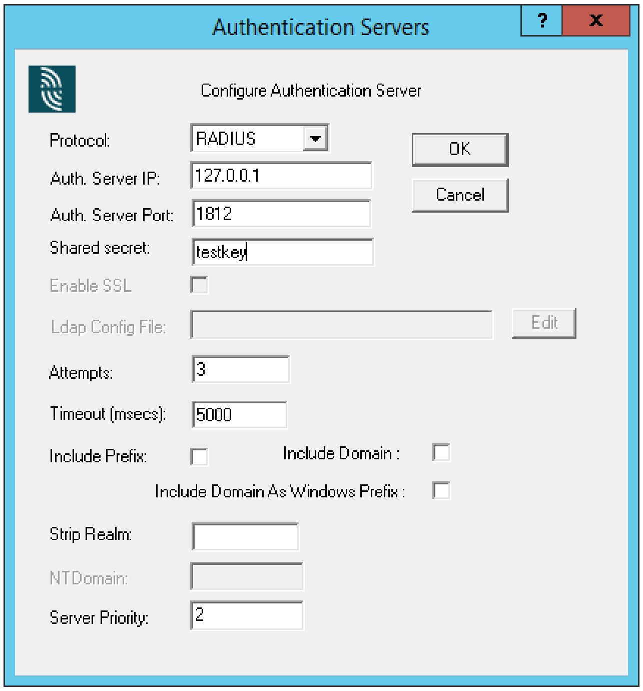
RADIUS Authentication

The iPass RoamServer can forward authentication requests and accounting packets, if desired, to a RADIUS server. The RoamServer will format the request as a standard RADIUS request and forward it to the RADIUS server at the address and port number that is specified during the installation. You must know the IP address and port number that will be used to reach your RADIUS server. Additionally, you must make the RADIUS shared secret available to the RoamServer. The RoamServer uses this shared secret to encrypt the RADIUS packet contents before sending them to the RADIUS server. The RADIUS server then uses the shared secret to decrypt the packet contents.

Your host must have a static, routable IP address, and cannot be blocked by a firewall.

To configure RoamServer for RADIUS authentication:

1. Add the RoamServer as a client of your RADIUS server.
2. Click **Start** > **iPass RoamServer 6.1.0** > **Configure RoamServer**.
3. In the **iPass Configuration** dialogue box, under **Authentication Servers**, click **Add**.
4. In the **Authentication Servers** dialog box, under **Protocol**, select **RADIUS** from the drop-down list.



Authentication Servers

Configure Authentication Server

Protocol: RADIUS

Auth. Server IP: 127.0.0.1

Auth. Server Port: 1812

Shared secret: testkey

Enable SSL □

Ldap Config File:

Attempts: 3

Timeout (msecs): 5000

Include Prefix: □ Include Domain : □

Include Domain As Windows Prefix : □

Strip Realm:

NTDomain:

Server Priority: 2

5. In **Auth. Server IP**, enter the IP address of your RADIUS server.
6. In the **Auth. Server Port** field, enter the port number that the RoamServer will send requests on (usually 1812). If the RADIUS is installed on the same machine as the RoamServer, do not use the loopback address (127.0.0.1). Instead, provide the machine's routable IP just as you would if they were installed in different locations.
7. In **Shared Secret**, enter the same shared secret that you entered into your RADIUS clients file in step 1. (This entry will be used to create a local clients file in <RS_Home>\6.1.0\clients.)
8. In **Attempts**, enter the number of attempts the RoamServer should make to connect with the RADIUS server. (Valid range is between 1 and 3 inclusive, with 3 as the default.)
9. In **Timeout**, enter the duration in milliseconds that RoamServer should wait for a response from the RADIUS server. (Valid range is between 2000 and 15000 inclusive, with 5000 as the default.)
10. If the RoamServer should pass on prefix information to the RADIUS server, select the **Include Prefix** checkbox.

11. If the RoamServer should pass on domain information to the RADIUS server, select the **Include Domain** checkbox.

12. In **Server Priority**, set the priority of this server for failover. (If this is the only server of its kind, enter 1. See the Failover section for more details.

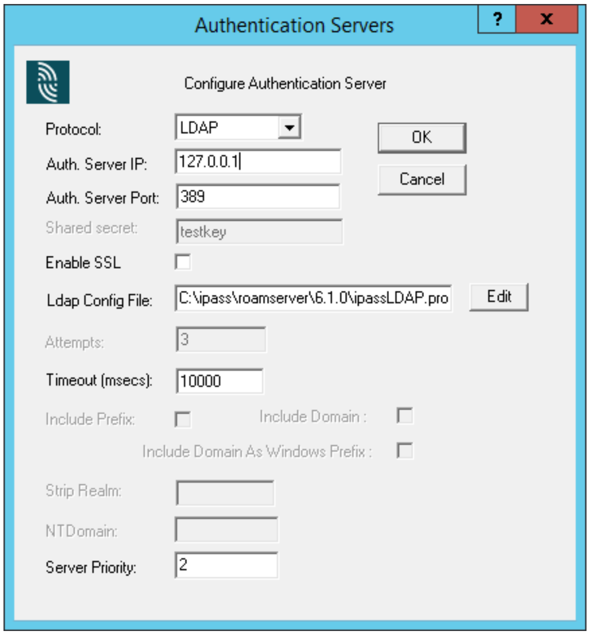
13. Click **OK**.

14. Optionally, to add your RADIUS server as an accounting server, under Accounting Servers, click **Add**. On the Accounting Servers dialog box, enter all the information you entered for the RADIUS Authentication Server (change port to 1813 (default)). Click **OK**. (See Accounting Servers for details.)

15. On the **iPass Configuration** dialog box, select **Restart RoamServers on Update**. Then click **Update**. The RoamServer will restart and the changes will take effect. (If this box is not checked, you must manually restart the RoamServer before any changes will take effect.)

LDAP Authentication

The iPass RoamServer can forward authentication requests to an LDAP server running on the network. The RoamServer will format the request as a standard LDAP request and forward it to the LDAP server at the address and port number that is specified during the installation. You must know the IP address and port number (TCP 389 by default) that will be used to reach your LDAP server.



Configure Authentication Server

Protocol: LDAP

Auth. Server IP: 127.0.0.1

Auth. Server Port: 389

Shared secret: testkey

Enable SSL □

Ldap Config File: C:\ipass\roamserver\6.1.0\ipassLDAP.pro

Attempts: 3

Timeout (msecs): 10000

Include Prefix: □ Include Domain : □

Include Domain As Windows Prefix : □

Strip Realm:

NTDomain:

Server Priority: 2

Additionally, you must configure/customize how the RoamServer will perform authentication at the LDAP server. LDAP specific configurations are set in a file called ipassLDAP.properties. For more information, refer to Appendix 1 on page 43, and the ipassLDAP.properties.example file included in the RoamServer package.

To configure RoamServer for LDAP authentication:

1. Click **Start** > **iPass RoamServer 6.1.0** > **Configure RoamServer**.

2. In the **iPass Configuration** dialogue box, under **Authentication Servers**, click **Add**

3. In the **Authentication Servers** dialog box, under **Protocol**, select **LDAP** from the drop-down list.

4. In the **Auth. Server** field, enter the IP address of your LDAP server. If LDAP is installed on the same machine as the RoamServer, do not use the loopback address (127.0.0.1). Instead, provide the machine's routable IP just as you would if they were installed in different locations.

5. In the **Auth. Server Port** field, enter the port number that the RoamServer will send requests on (usually 389).

6. In **LDAP Config. File**, enter the path to the LDAP configuration file.

7. Select **Enable SSL**, if SSL will be enabled over LDAP connections. (See Secure LDAP on page 22.) <- Not on page 22.
8. In **Timeout**, enter the duration in milliseconds that RoamServer should wait for a response from the LDAP server. (Valid range is between 2,000 and 15,000 inclusive, with 10,000 as the default.)
9. In **Server Priority**, set the priority of this server for failover. If this is the only server of its kind, enter 1. (See Failover on page 29). <- Wrong page
10. You can customize the LDAP configuration file by clicking **Edit**. For more details on this option, see the ipassLDAP.properties file.
11. Click **OK**.
12. In the iPass Configuration dialog box, select **Restart RoamServers on Update**. Then click Update. The RoamServer will restart and the changes will take effect. (If this box is not checked, you must manually restart the RoamServer before any changes will take effect.)

- The RoamServer can contain the IP address of more than one Authentication or Accounting Server for failover purposes. For more information, see the Failover section.
- **To edit an existing LDAP configuration file**, in the Authentication Servers dialog box, click **Edit**. Edit the file as needed.

Secure LDAP

RoamServer can support LDAP over SSL connections. Server-side authentication is performed in the SSL handshake. If enabled, RoamServer will only require a list of certification authority (CA) certificates for validating the LDAP server. <- Wrong! SSL is commonly done over port 636.

The certificate used for secure LDAP is not the certificate issued by iPass. It must be issued by a Certificate Authority both the RoamServer and the LDAP recognize. The certificate must be loaded into the keystore of the Operating System before trying to use it with the RoamServer. There must be a similar certificate on the LDAP.

To list all certificates, run list_CA_certificates.

To import additional CA certificates, run import_CA_certificate <cert-alias-name> <cert-file-name>

To delete a certificate, run delete_CA_certificate <cert-alias-name>.

> *By default, most secure LDAP servers allow client authentication in the SSL handshake but do not require it. To perform only server authentication, RoamServer must have the CA certificate loaded.*

TACACS+ Authentication

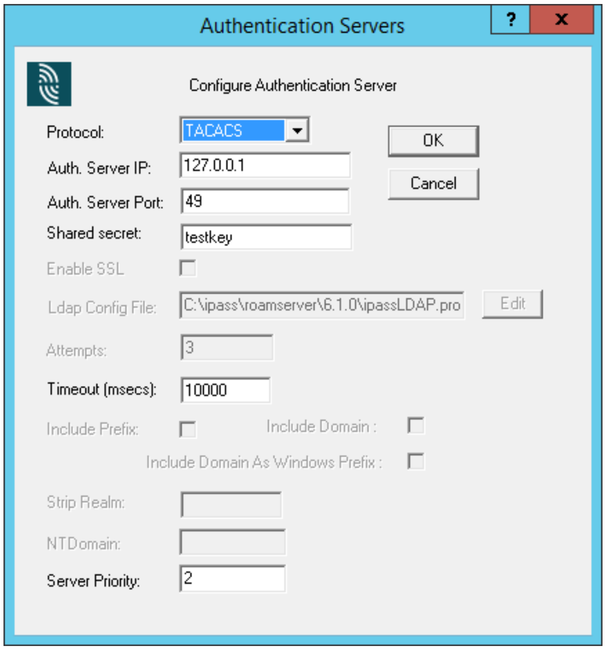
The iPass RoamServer can forward authentication requests to a TACACS+ server running on the network. The RoamServer will format the request as a standard TACACS+ request, and forward it to the TACACS+ server at the address and port number that is configured during the installation.

You must know the IP address and port number that will be used to reach your TACACS+ server. Additionally, you must make the TACACS+ shared secret available to the RoamServer. The shared secret is configured in the TACACS+ configuration file called **tac_plus.conf**. The RoamServer uses this shared secret to encrypt the TACACS+ packet contents before sending them to the TACACS+ server. The TACACS+ server then uses the shared secret to decrypt the packet contents. Please refer to your TACACS+ documentation for more information on the **tac_plus.conf** file and shared secret. The TACACS+ server can be located anywhere with a routable, static IP address, including on the same machine as the RoamServer.

If the TACACS+ server is running on an alternative machine on your network (that is not on the server running RoamServer), you will need to install a copy of the **tac_plus.conf** file on that server or on a network-addressable drive available to that server. You will also need to inform the RoamServer of the location of this file during configuration.

To configure the RoamServer for TACACS+ authentication:

1. Retrieve and copy the TACACS+ key from the configuration file of your TACACS+ Authentication server.
2. Click **Start** > **iPass RoamServer 6.1.0** > **Configure RoamServer**. The iPass Configuration dialog box is displayed.



Configure Authentication Server

Protocol: TACACS

Auth. Server IP: 127.0.0.1

Auth. Server Port: 49

Shared secret: testkey

Enable SSL ☐

Ldap Config File: C:\ipass\roamserver\6.1.0\ipassLDAP.pro

Attempts: 3

Timeout (msecs): 10000

Include Prefix: ☐ Include Domain : ☐

Include Domain As Windows Prefix : ☐

Strip Realm:

NTDomain:

Server Priority: 2

3. Under **Authentication Servers**, click **Add**.
4. On the **Authentication Servers** dialog box, under **Protocol**, select **TACACS+** from the drop-down list.
5. In the **Auth. Server** field, enter the IP address of your TACACS+ server. If the TACACS+ is installed on the same machine as the RoamServer, do not use the loopback address (127.0.0.1). Instead, provide the machine's routable IP just as you would if they were installed in different locations.
6. In **Auth. Server Port**, enter the port number that the RoamServer will send requests on (usually 49).
7. In **Shared Secret**, enter the key that you copied from your TACACS+ configuration file in step 1.
8. In **Timeout**, enter the duration in milliseconds that RoamServer should wait for a response from the TACACS+ server. (Valid range is between 2000 and 15000 inclusive, with 5000 as the default.)
9. Click **OK** to return to the **iPass Configuration** dialog box.
10. Optionally, to add your **TACACS+** server as an accounting server, under **Accounting Servers**, click **Add**. On the **Accounting Servers** dialog box, enter all the information you entered for the **TACACS+** Authentication Server. Click **OK**. (See Accounting Servers for details.)

11. On the **iPass Configuration** dialog box, select **Restart RoamServers on Update**. Then click **Update**. The RoamServer will restart and the changes will take effect. (If this box is not checked, you must manually restart the RoamServer before any changes will take effect.)

The RoamServer can contain the IP address of more than one Authentication or Accounting Server for failover purposes. For more information, see the Failover section.

Accounting Servers

You can configure RoamServer to forward accounting data to three sources: to an accounting file, a RADIUS server, or a TACACS+ server.

Using an Accounting File

If you are using Win NT or LDAP, you don't have an accounting server. The RoamServer is able to record accounting records to a text file called acct.txt.

1. In the **iPass Configuration** dialog box, under **Accounting Servers**, click **Add**.
2. In the **Accounting Servers** dialog box, in Protocol, select **AcctFile**.
3. In **Accounting File**, type the path and filename of the accounting file.
4. Click **OK**.

RADIUS Accounting

The iPass RoamServer can forward accounting information, if desired, to a RADIUS server running on the network. The RoamServer will format the request as a standard RADIUS request and forward it to the RADIUS server at the address and port number that is specified during the installation. You must know the IP address and port number (typically 1813) that will be used to reach your RADIUS server. Additionally, you must make the RADIUS shared secret available to the RoamServer. The RoamServer uses this shared secret to partially encrypt the RADIUS packet contents before sending them to the RADIUS server. The RADIUS server then uses the shared secret to decrypt the packet contents. A shared secret cannot contain the comma (,) or equals sign (=) characters.

To enable RADIUS accounting:

1. Add the RoamServer as a client of your RADIUS server.
2. Click **Start** > **iPass RoamServer 6.1.0** > **Configure RoamServer**. The iPass Configuration dialog box is displayed.
3. On the **iPass Configuration** dialog box, under **Accounting Servers**, click **Add**.
4. On the **Accounting Servers** dialog box, in **Protocol**, select **RADIUS**.
5. In **Acct. Server**, type the IP address of the RADIUS server to which RoamServer will forward accounting data.
6. In **Acct. Server Port**, enter the port number that the RADIUS server will send requests on. If the RADIUS is installed on the same machine as the RoamServer, do not use the loopback address (127.0.0.1). Instead, provide the machine's routable IP just as you would if they were installed in different locations.
7. In **Shared Secret**, enter the same shared secret that you entered into your RADIUS clients file in step 1. (This entry will be used to create a local clients file in <RS_Home>\6.1.0\clients.)
8. In **Attempts**, enter the number of attempts the RoamServer should make to connect with the RADIUS server. (Valid range is between 1 and 3 inclusive, with 3 as the default.)

9. In **Timeout,** enter the duration, in milliseconds, the RoamServer should wait for a response from the RADIUS server. (Valid range is between 2000 and 15000 inclusive, with 5000 as the default.)
10. 1If the RoamServer should pass on prefix information to the RADIUS server, select the **Include Prefix** checkbox.
11. If the RoamServer should pass on domain information to the RADIUS server, select the **Include Domain** checkbox.
12. Click **OK**.

TACACS+ Accounting

The iPass RoamServer can forward accounting data to a TACACS+ server running on the network. The RoamServer will forward this data to the TACACS+ server at the address and port number that is configured during the installation. You must know the IP address and port number that will be used to reach your TACACS+ server. Additionally, you must make the TACACS+ shared secret available to RoamServer. The shared secret is configured in the TACACS+ configuration file called tac_plus.conf. RoamServer uses this shared secret to partially encrypt the TACACS+ packet contents before sending them to the TACACS+ server. The TACACS+ server then uses the shared secret to decrypt the packet contents. Please refer to your TACACS+ documentation for more information on the tac_plus.conf file and shared secret. The TACACS+ server can be located anywhere with a routable, static IP address, including on the same machine as the RoamServer.

If the TACACS+ server is running on an alternative host on your network (that is, not on the server running RoamServer), you will need to install a copy of the tac_plus.conf file on that server or on a network-addressable drive available to that server. You will also need to inform the RoamServer of the location of this file during configuration.

To enable TACACS+ accounting:

1. Retrieve and copy the TACACS+ key from the configuration file of your TACACS+ Authentication server.
2. Click **Start** > **iPass RoamServer 6.1.0** > **Configure RoamServer**. The iPass Configuration dialog box is displayed.
3. On the **iPass Configuration** dialog box, under **Accounting Servers**, click **Add**.
4. On the **Accounting Servers** dialog box, in **Protocol**, select TACACS+.
5. In **Acct. Server**, enter the IP address of your TACACS+ server. If the TACACS+ is installed on the same machine as RoamServer, do not use the loopback address (127.0.0.1). Instead, provide the machine's routable IP just as you would if they were installed in different locations.
6. In **Acct. Server Port,** enter the port number that the RoamServer will send requests on (usually 49).
7. In **Shared Secret**, enter the key that you copied from your TACACS+ configuration file in step 1.
8. In **Timeout**, enter the duration in milliseconds that RoamServer should wait for a response from the TACACS+ server. (Valid range is between 2000 and 15000 inclusive, with 5000 as the default.)
9. Click **OK** to return to the **iPass Configuration** dialog box.

Configuration

This section discusses some of the configurable options in RoamServer.

Configuration Options

Using a Policy File

A policy file allows you to filter the requests being sent to your Authentication Server. This feature may be helpful if you wish the RoamServer to authenticate from a large user database, but only want a small group of those users to be able to roam, or conversely, if you only wish to deny roaming access to a small group. If a policy file is set up, the RoamServer will validate all users against this file before contacting your Authentication Server.

The Policy Tool

The Policy Tool is an application used for creation and maintenance of your Policy File. Although the Policy File is a text file, iPass recommends you use the Policy Tool to ensure proper formatting and correct policy criteria.

- The Policy Tool is located in your **<RS_Home>\6.1.0\bin** folder.
- The Policy File is located at <RS_Home>\6.1.0\roamserver\policy.txt.

To run the policy tool:

1. Open a command window in the **<RS_Home>\6.1.0\bin** folder.
2. At the command prompt, type **rs_policy** and press the Enter key.

To create a policy file:

1. Run the **Policy Tool**.
2. If the tool detects that no Policy File exists, it will create one in the default folder, which is

   <RS_Home>\6.1.0\roamserver\policy.txt .

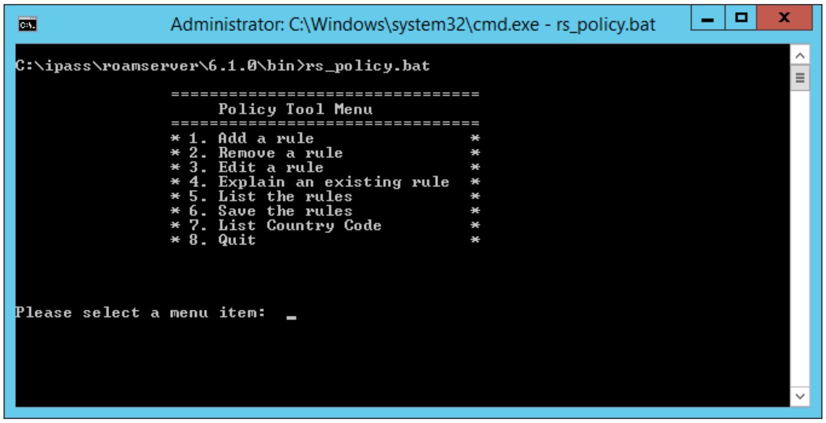
To enable use of a policy file:

1. In the **iPass Configuration** dialog box, check **Use Policy File**.
2. In the text box, type the path to your policy file, or accept the default.
3. Click **Update**.

To edit or manage your policy file:

- In the policy tool, choose your option from the menu:

1. Add a rule
2. Remove a rule
3. Edit a rule
4. Explain an existing rule
5. List the rules



```
C:\ipass\roamserver\6.1.0\bin>rs_policy.bat
                ================================
                     Policy Tool Menu
                ================================
                * 1. Add a rule                 *
                * 2. Remove a rule              *
                * 3. Edit a rule                *
                * 4. Explain an existing rule   *
                * 5. List the rules             *
                * 6. Save the rules             *
                * 7. List Country Code          *
                * 8. Quit                       *


Please select a menu item:  _
```

6. Save the rules
7. List Country Code
8. Quit

- When done, enter 8 to quit the Tool. You must stop and then restart RoamServer so that it can read a new or edited Policy File.

Policy File Pattern Matching

The policy file pattern matching is from most specific to the least, as follows:

#class of service	auth_domain	user_id	country_code
1	1	1	1
1	1	1	0
1	1	0	1
1	1	0	0
1	0	1	1
1	0	1	0
1	0	0	1
1	0	0	0
0	1	1	1
0	1	1	0
0	1	0	1
0	1	0	0
0	0	1	1
0	0	1	0
0	0	0	1
0	0	0	0

All rules are read and the most specific rule to match a given request is used. For example, these entries in a policy file would block all wireless access, except in the US.

#class of service	auth_domain	user_id	country_code	Allow access
WIRELESS	*	*	*	N
WIRELESS	*	*	US	Y

Because the policy file is written with permissions of root/admin, lowering the privileges required to run the policy tool will cause the tool to fail. Accordingly, you may wish to do one of the following to ensure policy file permissions are valid:

- Reset policy file permissions every time the policy tool is run.
- Set up a cron job to periodically reset the file permission regardless of when policy tool is run.

Policy File Mapping

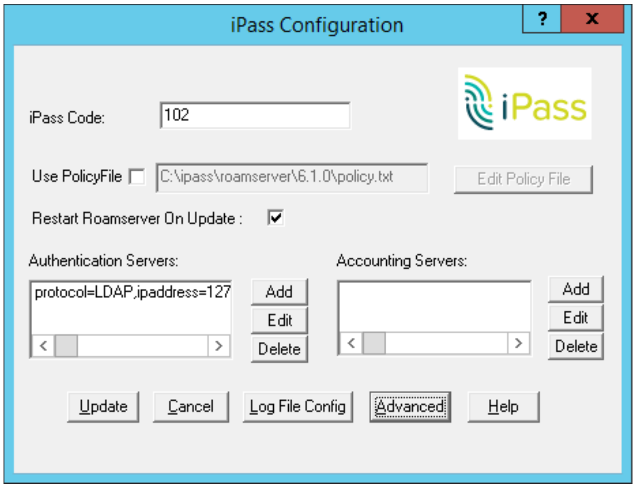
This table shows the mappings of NAS port type numbers to the class of service:

Nas-port-type	Class of service
0	DIAL-UP
1	DIAL-UP
2	DIAL-UP-ISDN
3	DIAL-UP-ISDN
4	DIAL-UP
5	DIAL-UP-PHS
6	DIAL-UP
7	DIAL-UP
8	DIAL-UP
9	DIAL-UP
10	WIRED
11	WIRED
12	WIRED
13	WIRED
14	WIRED
15	WIRED
16	WIRED
17	WIRELESS
18	WIRELESS
19	WIRED
20	WIRED
21	MOBILEDATA
22	MOBILEDATA
23	MOBILEDATA
24	WIRELESS
25	WIRED
ALL OTHERS	DIAL UP

Advanced Configuration

To configure RoamServer advanced features:

1. Click **Start** > **iPass RoamServer 6.1.0** > **Configure RoamServer**. The iPass Configuration dialog box is displayed.
2. On the **iPass Configuration** dialog box, click **Advanced**.



iPass Code: 102

Use PolicyFile ☐ C:\ipass\roamserver\6.1.0\policy.txt

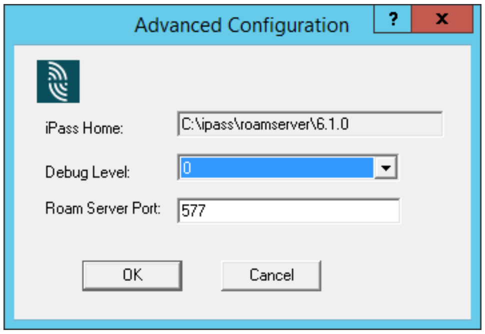
Restart Roamserver On Update : ☑

Authentication Servers: protocol=LDAP,ipaddress=127

Accounting Servers:

3. On the **Advanced Configuration** dialog box, you can set the following:

- **iPass Home**: This value indicates the location of the RoamServer home installation folder. The value shown is established during the installation process and does not require any further modification.



iPass Home: C:\ipass\roamserver\6.1.0

Debug Level: 0

Roam Server Port: 577

- **Debug Level**: This parameter controls the amount of debugging output that is produced to the file <RS_Home>\6.1.0\logs\roamserver.trace. The range for this value is 0 to 5 (inclusive), where 0 will disable debugging except for critical errors, 1 produces the least amount of output, and 5 produces the highest. For normal operation, leave this value set to the lowest level (0) to produce the minimum amount of output. Only critical errors, for example, the inability of the server to startup, will be logged. Error messages are listed in Appendix II. For more information about debugging and the roamserver.trace file, see Trace Log File Configuration.
- **RoamServer Port**: Specifies the Port number on which the RoamServer receives requests from the iPass Transaction Servers. Do not remove this option. The default value is 577.

4. Click **OK** when done.

Failover

If the primary server is unreachable, the iPass RoamServer can fail over to a secondary authentication/accounting server. Failover is configured from the main dialog of the RoamServer Configuration tool. The first server listed in the list box (on the top line of the list box) will be the primary server. The RoamServer will always attempt to contact this server first. If this server is inaccessible, it will then attempt the subsequent servers, in order from top to bottom, in the list.

Your secondary servers do not have to be of the same type as your primary server. For instance, if you had both a RADIUS server and an LDAP server, you could designate your RADIUS server as primary and your LDAP server as secondary, or vice versa.

Due to protocol limitations, the RoamServer Configuration GUI can only be used to configure the failover feature for certain forms of authentication or accounting. The use of this feature is summarized in the table below:

Protocol	Multiple Authentication Servers	Multiple Accounting Servers
RADIUS	Use RoamServer Configuration tool	Use RoamServer Configuration tool
LDAP	Use RoamServer Configuration tool	Server does not support accounting. We suggest you use file-based accounting if needed.
TACACS+	Use RoamServer Configuration tool	Use RoamServer Configuration tool
WinNT	Configure within your Windows Network	Server does not support accounting, configure within your Windows Network. We suggest you use file-based accounting if needed.
WinNT RAS	Configure within your Windows Network	Server does not support accounting, configure within your Windows Network. We suggest you use file-based accounting if needed.

To configure the RoamServer to fail over to a secondary authentication or accounting server:

1. Click **Start** > **iPass RoamServer 6.1.0** > **Configure RoamServer**.
2. In the iPass Configuration dialog box, click Add next to the list box of the server you wish to add, either authorization or accounting.
3. Configure the server just as you would a primary server, adding all relevant data as described under Basic Configuration, above.
4. Repeat steps 2-3 as needed until all secondary servers are added. The RoamServer will contact these servers in the vertical order that they appear on the list.

5. In the **iPass Configuration** dialog box, check **Restart RoamServer on Update** and click **Update**. The RoamServer will restart and the changes will take effect. (If this box is not checked, you must manually restart the RoamServer before any changes will take effect.)

Server Priority

The dialog boxes used to configure authentication and accounting servers each have a text box labeled **Server Priority.** You can use this field to order the servers on the **iPass Configuration** dialog box, where all your servers are listed, each in its own list. Priority 1 means the server will appear at the top of the list, 2 means it will appear second, and so on. This feature can be used to specify servers for failover, as described above, with the first server listed becoming the primary server, and those lower down on the list becoming secondary servers.

Failover and Local Servers

Since there will always be a response from the local server, if you set one of your failover servers to the local WinNT or WinNT RAS server, there is no need to set any further servers in the sequence.

Trace Log Configuration

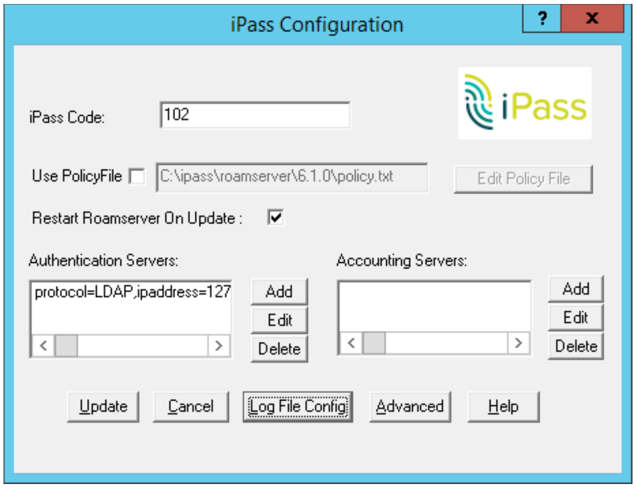
The RoamServer can be configured to write information about access attempts to a log file for debugging purposes. If enabled, debugging information is output to a local log file, **roamserver.trace**, found in the **<RS_Home>\6.1.0\logs\ folder***.* The amount of debugging output can be controlled by changing the value of the Debug Level parameter.

If your Debug value is set to any value greater than 0, you will need to customize the log file rotation and backup process.

To configure the RoamServer to log debugging information to the roamserver.trace file:

1. Click **Start** > **iPass RoamServer 6.1.0** > **Configure RoamServer**.

2. In the **iPass Configuration** dialog box, click **Log File Config**.


iPass Configuration

iPass Code: 102

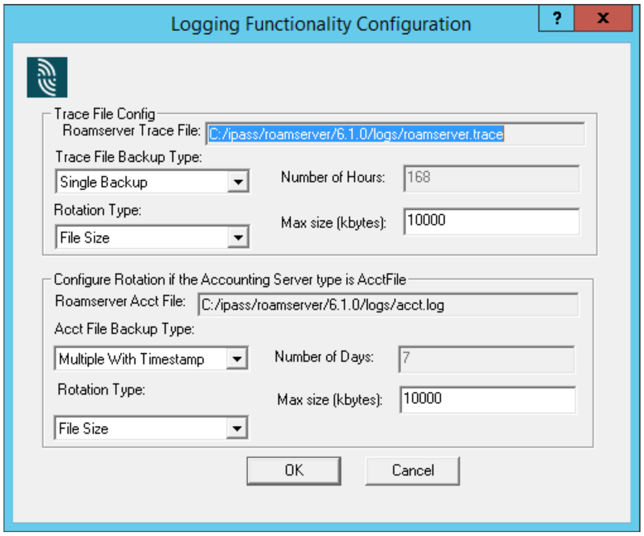
Use PolicyFile □ C:\ipass\roamserver\6.1.0\policy.txt

Restart Roamserver On Update : ☑

Authentication Servers: protocol=LDAP,ipaddress=127

Accounting Servers:

3. In the **Logging File Configuration** dialog box, under Trace File Config, verify the path to the RoamServer trace file. The default is **<RS_Home>\6.1.0\logs\roamserver.trace**



Trace File Config
Roamserver Trace File: C:/ipass/roamserver/6.1.0/logs/roamserver.trace
Trace File Backup Type: Single Backup
Number of Hours: 168
Rotation Type: File Size
Max size (kbytes): 10000

Configure Rotation if the Accounting Server type is AcctFile
Roamserver Acct File: C:/ipass/roamserver/6.1.0/logs/acct.log
Acct File Backup Type: Multiple With Timestamp
Number of Days: 7
Rotation Type: File Size
Max size (kbytes): 10000

4. In **Trace File Backup Type**, select a log file backup type from the drop-down list.

- **Single Backup**: The Single Backup option will keep one backup of the applicable log file and overwrite that backup each time the log is rotated. This technique assures that you have only one additional file in the folder and limits the disk space taken up by backup files. Of course, it also limits your ability to track the log history.
- **Multiple With Timestamp**: The Multiple With Timestamp option allows you to keep an unlimited number of backup logs, each named based on the date and time it was rotated. This allows you unlimited history tracking, but can potentially fill up a partition if left for too long. You should consider your historical tracking needs, as well as your disk space requirements to determine which backup type is right for you.

5. In **Rotation Type**, select the method you wish to use to determine when to rotate the trace file from the drop-down list.

- **File Size**: In File Size rotation, the size of the log file is checked on each incoming request, and it is rotated when it reaches a given size. If this option is selected, indicate the size at which you would like the log to be rotated in the File Size text box.
- **Number of Hours**: In Number of Hours rotation, the log file will be rotated after the specified number of hours has passed. Note that this approach does not take into account the file size, so the log could potentially grow quite large if not rotated on a regular basis. If this option is selected, enter the number of days to pass between rotations in the Number of Hours box.

6. Click **OK**.

7. On the **iPass Configuration** dialog box, verify that the box labeled **Restart RoamServer on Update** is checked, and click **Update**. The RoamServer will restart and the changes will take effect. If this box is not checked, you must manually restart the RoamServer before any changes will take effect.

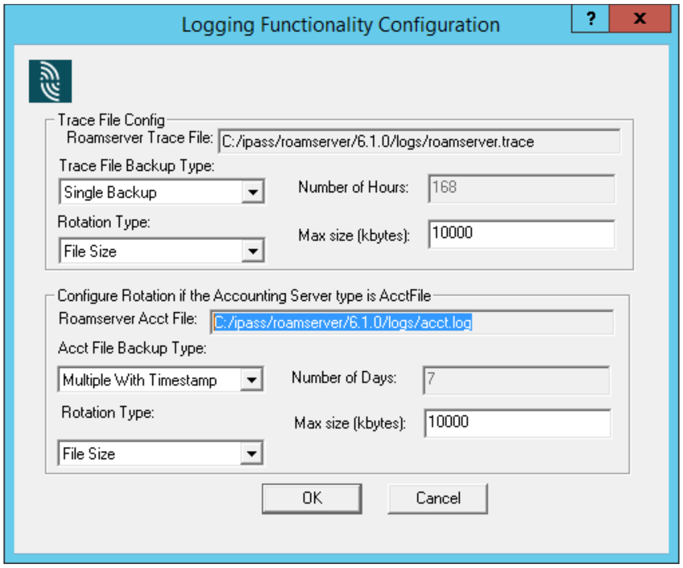
Accounting Log File Configuration

RoamServer can be configured to write accounting information to a log file in a manner similar to that for debugging. The log file rotation and backup process can be customized to suit your networking environment and business needs. Depending on the type of AAA used, the RoamServer can utilize either local accounting logging or remote accounting logging. (Previous versions of the RoamServer could log to both a local server and remote server at the same time, but this feature is not present in RS 6.1.0)

For authentication protocols that do not have a built-in remote accounting server (that is, WIN NT, WIN NT RAS, or LDAP), the RoamServer can be configured to keep detailed local accounting records at a location specified by the user. For authentication protocols which have a remote server capable of handling accounting transactions (that is RADIUS or TACACS+), RoamServer can forward the accounting record to the remote server for logging.

To log accounting records:

1. Click **Start > iPass RoamServer 6.1.0 > Configure RoamServer.** The iPass Configuration dialog box is displayed.
2. On the **iPass Configuration** dialog box, click **Log File Config.**
3. Under **Configure Rotation if the Accounting Server Type is AcctFile**, in the **RoamServer Acct File** field, verify the location of the local accounting file. The default location is `<RS_Home>\6.1.0\logs\acct.log.`



Logging Functionality Configuration

Trace File Config
Roamserver Trace File: C:/ipass/roamserver/6.1.0/logs/roamserver.trace
Trace File Backup Type: Single Backup
Number of Hours: 168
Rotation Type: File Size
Max size (kbytes): 10000

Configure Rotation if the Accounting Server type is AcctFile
Roamserver Acct File: C:/ipass/roamserver/6.1.0/logs/acct.log
Acct File Backup Type: Multiple With Timestamp
Number of Days: 7
Rotation Type: File Size
Max size (kbytes): 10000

4. In **Acct File Backup Type**, select the log file backup type from the drop-down list.

- **Single Backup:** The Single Backup option will keep one backup of the applicable log file and overwrite that backup each time the log is rotated. This technique assures that you have only one additional file in the folder and limits the disk space taken up by backup files. Of course, it also limits your ability to track the log history.
- **Multiple With Timestamp:** The Multiple with Timestamp option allows you to keep an unlimited number of backup logs, each named based on the date and time it was rotated. This allows you unlimited history tracking, but can potentially fill up a partition if left for too long. You should consider your historical tracking needs, as well as your disk space requirements to determine which backup type is right for you.

5. In **Rotation Type,** select the method you wish to use to determine when to rotate the trace file from the drop-down list.

- **File Size***:* In File Size rotation, the size of the log file is checked on each incoming request, and it is rotated when it reaches a given size. If this option is selected, indicate the size at which you would like the log to be rotated in the File Size text box.

- **Number of Days***:* In Number of Days rotation, the log file will be rotated after the specified number of days has passed. Note that this approach does not take into account the file size, so the log could potentially grow quite large if not rotated on a regular basis. If this option is selected, indicate the number of days to pass between rotations in the Number of Days text box.

6. Click **OK**.
7. On the **iPass Configuration** dialog box, verify that the box labeled **Restart RoamServer on Update** is checked, and click **Update**. The RoamServer will restart and the changes will take effect. If this box is not checked, you must manually restart the RoamServer before any changes will take effect.

Ascend Data Filters for Non-VPN Access

Some network providers on the iPass network filter port 25 traffic (SMTP), in an effort to prevent the distribution of spam mail on their networks. Although port 25 traffic is blocked from these providers, they allow port 25 traffic to pass to a limited number of IP addresses to allow users to send SMTP mail to valid mail servers. The IP addresses to which port 25 traffic is allowed is communicated by the Ascend Data Filter attributes, which are sent when the user successfully authenticates. These attributes are configured in `ipassRS.properties`. (The format is similar to how a RADIUS server's `users` file would be configured to return those attributes.)

If users will be connecting through a VPN, this property can be ignored with no effects. If users will not be connecting through a VPN, then iPass strongly recommends you configure these settings to reflect your SMTP servers.

Sample Settings

```
AscendDataFilter1=ip in forward tcp est
AscendDataFilter2=ip in forward dstip xxx.xxx.xxx.xxx/yy
AscendDataFilter3=ip in drop tcp dstport=25
AscendDataFilter4=ip in forward
```

`xxx.xxx.xxx.xxx/yy` would be replaced by an IP mask identifying the customer's mail server IP addresses; for example, `218.239.99.139/32.` Note that most providers only allow masks ranging from 24 to 32.

For example, if your SMTP servers' public IP address is 236.14.5.70, then the settings would look like this:

```
AscendDataFilter1=ip in forward tcp est
AscendDataFilter2=ip in forward dstip 236.14.5.70/32
AscendDataFilter3=ip in drop tcp dstport=25
AscendDataFilter4=ip in forward
```

Note that a either a single IP address (236.14.5.70/32) or a range of IP addresses (236.14.5.0/24) can be specified.

In this second example, there are two entries. The first is a single SMTP server, and the second is a network range. Port 25 traffic will be allowed to the single IP address specified in `AscendDataFilter2`, as well as the entire network specified in `AscendDataFilter3`.

```
AscendDataFilter1=ip in forward tcp est
AscendDataFilter2=ip in forward dstip 236.14.5.70/32
AscendDataFilter3=ip in forward dstip 236.16.6.0/24
AscendDataFilter4=ip in drop tcp dstport=25
AscendDataFilter5=ip in forward
```

Up to 17 different IP addresses or range strings can be specified in this manner.

Log File Deletion

Log files and accounting files can grow to unmanageable sizes. To control this, you can set log files to be deleted after a specified period of time by setting `LogDirFileDeletionAge` to an appropriate value. The default is 90 days.

Routing by Realm

Routing by realm allows routing requests to specific AAA servers, based on the user's realm or domain. Routing can also be done by routing prefix.

This allows you to use different types of authentication servers, if necessary. For example, you could use both a RADIUS server and an LDAP server simultaneously. Requests from one realm can be directed to one AAA server, while requests from another realm can be directed to a second AAA server.

To enable routing by realm, set `RouteByRealm` to YES. If routing by realm is enabled, you will also need to set other properties to specify your other AAA servers.

Sample Settings

```
AuthServer1=protocol=RADIUS,ipaddress=10.10.0.1,port=1812,sharedsecret=foo,Attempts=3,
IdleTimeout=5000,IncludePrefix=No,IncludeDomain=No,IncludeDomainAsWinPrefix=No

AuthServer2=protocol=LDAP,ipaddress=10.10.0.2,port=389,LdapConfigFile=C:/ipass/roamser
ver/6.1.0/ipassLDAP.properties,IdleTimeout=10000,enableSsl=No

AcctServer1=protocol=RADIUS,ipaddress=10.10.0.1,port=1813,sharedsecret=foo,Attempts=3,
IdleTimeout=5000,IncludePrefix=No,IncludeDomain=No,IncludeDomainAsWinPrefix=No

AcctServer2=protocol=AcctFile,localAcctFileName=C:/ipass/roamserver/6.1.0/logs/acct.lo
g,IncludeDomainAsWinPrefix=No
```

```
RouteByRealm=YES

RoutingRealm1=Realm=mydomain.com,AuthServer1=AuthServer1,AcctServer1=AcctServer1

RoutingRealm2=Realm=XY,AuthServer1=AuthServer2,AcctServer1=AcctServer2

RoutingRealm3=Realm=DEFAULT,AuthServer1=AuthServer1,AcctServer1=AcctServer1
```

In this sample there are two Authentication Servers defined. AuthServer1 is a RADIUS. AuthServer2 is an LDAP.

If the customer logs in using the realm mydomain.com, the line which begins, "RoutingRealm1" is in play. It defines the primary Authentication Server as AuthServer1. The rule is written as AuthServer1=AuthServer1. This translates as "The primary authentication server for this realm is the line above that starts with 'AuthServer1='"

But if the customer logs in with the realm XY, the line which begins, "RoutingRealm2" is in play. It defines the primary Authentication Server as AuthServer2. The rule is written as AuthServer1=AuthServer2. This translates as "The primary authentication server for this realm is the line above that starts with 'AuthServer2='"

You'll notice Route-by-Realm also directs traffic to Accounting Servers. When addressing the Accounting Server the key AuthServer becomes AcctServer. Otherwise, the logic of how Route-by-Realm works is the same.

In the examples, when routing realm mydomain.com is used, AcctServer1 is employed to send the accounting records to the RADIUS.

When routing realm XY is used, AcctServer2 is used. Please note, in this case, a text file is being populated because LDAP does not record accounting records.

The final line, where the realm is DEFAULT is required to catch any requests that contain malformed realms. This line gives the RoamServer an avenue to forward the request. Not having this line can cause the RoamServer to crash if a malformed realm is used.

Security Best Practices

These suggestions for best practices will help improve RoamServer security.

Firewall:

- Lockdown the firewall to only allow access to RoamServer's NAT routable address or already-routable DMZ address from an official iPass Transaction Center IP (listed on page 7) through port 577.
- Use the built-in Windows Firewall as an additional layer of defense.

Restrict Access:

- Put a limit on who can log on to RoamServer. This is best done by making it a standalone server (not part of Windows domain structure).
- Restrict access to RoamServer's configuration, log files, and keys to only accounts that need it.

Monitor:

- Create scripted remote log backups and audit the logs periodically.
- Monitor the status of iPass RoamServer Service through SNMP (Simple Network Management Protocol), scheduled WMI (Windows Management Instrumentation) scripts, or other monitoring agents to ensure that the application is running.
- Use an HIDS (Host-based Intrusion Detection System) or create digests of sensitive files to detect any changes to the system

Other Suggestions:

- Internally, allow RoamServer to communicate only through the required ports for the authorized protocol of your choosing (for example, RADIUS, LDAP, LDAP/SSL).
- Configure IPsec policies between RoamServer and other Windows communication peers to protect against entities that access the local wire through a network device takeover, such as a MiTM (Man in The Middle) attack.
- In LDAP configurations, create a standard Domain User account for RoamServer without any other privileges, such as Terminal Server rights or Remote Access rights.
- Do not use EFS (Encrypted File System) on the RoamServer directory because RoamServer service cannot read configuration files that have been encrypted by a specific user's EFS certificate.

ipassRS.properties

The ipassRS.properties file allows customization of RoamServer features. By setting properties in the file, you can enable important RoamServer functions. Enabling some features may involve setting more than one property.

Property names are case-sensitive, but property values are not. Valid values for Boolean properties are: true, false, yes, no, y, n.

See the Setup section for information on setting values in ipassRS.properties.

Property Help

You can obtain help on any property, including those listed here, by using a tool called `config_help.bat`, found in your `<RS_Home>\6.1.0\bin` directory.

- **To list all server properties:** open a command prompt and run config_help.bat –listall (or ipass_config_console.bat –listall).
- **To describe usage of a property:** open a command prompt and run `config_help.bat -help <property name>`

Property Glossary

This glossary defines all properties found in `ipassRS.properties,` including configurable parameters for each property.

Property	Description
`AcctLogBackupType`	`AcctLogBackupType=<backupType>` where <backupType> is either MultipleWithTimestamp or SingleBackup. The default is MultipleWithTimestamp. `AcctLogBackupType` sets the accounting log's backup file name when rotation is to be performed on local accounting files.
`AcctLogRotationDays`	`AcctLogRotationDays=<days>` Valid range is: 1 to 30 days. The default is 7 days. `AcctLogRotationDays` control how often the local accounting file is rotated.
`AcctLogRotationMaxSize`	`AcctLogRotationMaxSize=<max size>` Minimum value is 100 kbytes. Maximum value is 20000 kbytes. The default is 10000 kbytes. `AcctLogRotationMaxSize` limits how large (in kbytes) the local accounting file can get before it is rotated..

`AcctLogRotationType`	`AcctLogRotationType=<rotationType>`
	Where <rotationType> is either FileSize or NumberOfDays.The default is FileSize.
	`AcctLogRotationType` sets the type of rotation to be performed on the local `accounting files.`

`AcctServer`	Provides accounting server information, for example
	`AcctServer1=name11=value11,name12=value12,name13=value13.....`
	`AcctServer2=name21=value21,name22=value22,name23=value23.....`
	AcctServer parameters:
	■ **Protocol**: The server's protocol. Values can be: NT/Radius/LDAP/TACACS\\
	■ **EnableSsl**: Flag used to enable/disable SSL connections to the LDAP servers. It is ignored when used for other Acct servers.
	■ **IpAddress**: The server's IP address.
	■ Port: The server's port number.
	■ **LocalIpAddress**: The Local IP address to bind the socket to. (Optional and Only for RADIUS)
	■ Attempts: The number of attempts made to communicate with a server.
	■ **IdleTimeout**: Timeout (in milliseconds) to wait for a response from a server for a given communication attempt.
	■ **SharedSecret**: The shared secret used by a RADIUS/TACACS+ server.
	■ **IncludeDomain**: Include the user's domain in the request sent to the server.
	■ **IncludeDomainAsWinPrefix**: Include the user's domain, as Windows style prefix,in the request sent to the server. For example, user@ntdomain would become ntdomain\user
	■ **IncludePrefix**: Include the user's routing prefix in the request sent to the server.
	■ **IncludeNasPortType**: Include the NAS-Port-Type in the request sent to the RADIUS AAA server.
	■ **StripRealm**: Specifies a realm suffix to strip away from the user's domain. For example, with StripRealm=example.com and IncludeDomainAsWinPrefix enabled, the login of user@ntdomain.example.com would become user@ntdomain
	■ **NTDomain**: The NT domain used to authenticate window users.
	■ **NTRasMode:** The NT RAS mode to use. 1=WINNT RAS mode, 0=WINNT.
	■ **SiteFile:** The file used in Site (Unix Shadow file) authentication
	■ **LdapConfigFile:** The file used to load LDAP specific properties for an LDAP server.
	■ **ValidateAuthenticator**: Specifies in the RADIUS Authenticator should be validated. Values are YES or NO. Default is YES.

	■ **ProtocolVersion**: Used by the TACACS+ server to specify the Minor Version.Values are 1 or 0. Default is 1. ■ **EnableLocalAcct:** Used by an AcctFile server to enable/disable local accounting. Values are YES or NO. Default is NO. ■ **RetryDelay**: The time delay, in minutes, before retrying a server that recently failed a connection request. When a connection fails to a server, it is reordered to the end of the list. Once the RetryDelay expires, that server is brought back to the top of the list. The default value is 15 minutes. Valid range is: >=0.
`AppSharedKey`	`AppSharedKey= <Secret key> for encryption and decryption` This entry determines the salt used for encrypting and decrypting the LDAP bind password. The passwords in LDAP property file will not be encrypted if it is not set.
`AuthCacheEnabled`	`AuthCacheEnabled=yes/no.` Determines if the caching of successful authentication requests is enabled. Default is set to YES.

`AscendDataFilter`	`AscendDataFilter1=<valid string for ascend-data-filter>`
	This is used as an anti-Spam feature for some providers and will block the email port (25) at the provider. If the AAA server does not send it to us, we will use the AscendDataFilter(s) specified to send back in the auth accept packet.
	An example entry is:
	`AscendDataFilter1=ip in forward tcp est`
	`AscendDataFilter2=ip in forward dstip xxx.xxx.xxx.xxx/yy`
	`AscendDataFilter3=ip in drop tcp dstport=25`
	`AscendDataFilter4=ip in forward`
	The string `"ip in drop tcp dstport=25"` is a mandatory AscendDataFilter attribute. When no AscendDataFilter is configured, this feature is disabled. See page 33 for more information.

`AuthServer`	Provides authorization server information, for example:
	`AuthServer1=name11=value11,name12=value12,name13=value13.....`
	`AuthServer2=name21=value21,name22=value22,name23=value23.....`
	AuthServer parameters:
	■ **Protocol**: The server's protocol. Values can be: NT/Radius/LDAP/TACACS
	■ **EnableSsl**: Flag used to enable/disable SSL connections to the LDAP servers. It is ignored when used for other Auth servers.
	■ **IpAddress**: The server's IP address.
	■ **Port**: The server's port number.
	■ **LocalIpAddress**: The Local IP address to bind the socket to.

	(Optional and Only for RADIUS) ■ **Attempts**: The number of attempts made to communicate with a server. ■ **IdleTimeout**: Timeout (in milliseconds) to wait for a response from a server for a given communication attempt ■ **SharedSecret**: The shared secret used by a RADIUS/TACACS+ server. ■ **IncludeDomain**: Include the user's domain in the request sent to the server. ■ **IncludeDomainAsWinPrefix**: Include the user's domain, as Windows style prefix,in the request sent to the server. For example, user@ntdomain would become ntdomain\user ■ **IncludePrefix**: Include the user's routing prefix in the request sent to the server. ■ **IncludeNasPortType**: Include the NAS-Port-Type in the request sent to the RADIUS AAA server. ■ **StripRealm**: Specifies a realm suffix to strip away from the user's domain. For example, with **StripRealm=example.com** and **IncludeDomainAsWinPrefix** enabled, the login of user@ntdomain.example.com would become **user@ntdomain** ■ **NTDomain:** The NT domain used to authenticate window users. ■ **NTRasMode:** The NT RAS mode to use. 1=WINNT RAS mode, 0=WINNT. ■ **SiteFile:** The file used in Site (Unix Shadow file) authentication ■ **LdapConfigFile**: The file used to load LDAP specific properties for an LDAP Server. ■ **ValidateAuthenticator**: Specifies in the RADIUS Authenticator should be validated. Values are YES or NO. Default is YES. ■ **ProtocolVersion**: Used by the TACACS+ server to specify the Minor Version.Values are 1 or 0. Default is 1. ■ **EnableLocalAcct:** Used by an AcctFile server to enable/disable local accounting. Values are YES or NO. Default is NO. ■ **RetryDelay:** The time delay, in minutes, before retrying a server that recently failed a connection request. When a connection fails to a server, it is reordered to the end of the list. Once the RetryDelay expires, that server is brought back to the top of the list. The default value is 15 minutes. Valid range is: >=0.
`CustomerId`	`CustomerId=<iPass Code>.` This is the same number as your iPass portal customer ID. If you do not yet have such

	code, or are unsure what this code is, contact your iPass representative.
`DebugLevel`	`DebugLevel=<level>.` Debug level determines if debug and error messages are logged to the trace file. The following levels are supported. Debug Level 0 - Only severe messages are logged. Debug Level 1 - Error messages are logged. Debug Level 2 - Error and Debug messages are logged. Debug Level 3 - Error, Debug, and Packet parsing information is logged. Debug Level 4 - Error, Debug, Packet parsing, and Packet dumping is logged. Debug Level 5 - Detailed Packet and debug information is logged. The default value for this property is 0 Note: Production servers should normally run with debug level 0.
`FailedAcctLogDir`	`FailedAcctLogDir=<Failed Accounting Directory>` If an accounting record cannot be sent to the AAA server due to a communication error, the RoamServer writes the record to this file. The RoamServer writes one file per failed record. The file name of these files would have the timestamp as the suffix. Use the `AcctLog` tool to retransmit these records to the RoamServer, which will then resend it to the Accounting Server. The failed account directory should specify either the full path to the directory or the path relative to the iPass server home via the `$ipass.server.home` macro. Default value for this property is set to `$ipass.server.home/logs/failedAcct/`
`FilterRequest`	`FilterRequest=<filter time in minutes>` This property determines the amount of time to keep users in the local authentication cache. This cache is used to filter duplicate request and authenticate cached users. Valid range is 0 to 10 minutes. A value of 0 will turn off local authentication cache. The `FilterRequest` default is 0 minutes.
`HeartBeatInterval`	`HeartBeatInterval=<number of minutes>` This entry determines the time interval between heartbeat messages. This is an advanced setting. The server may not function properly if this value is set incorrectly. Default value for this property is set to 15 minutes.
`HeartBeatMessage`	`HeartBeatMessage=yes/no.` This entry determines if the heartbeat is turned on or off. This is an advanced setting. The server may not function properly if this value is set incorrectly. Default value for this property is set to no (heartbeat messages are turned off).
`IMonServer`	Provides IMonServer information. The IMonServers are central iPass servers used to receive HeartBeat Messages from this server. Sample format of the entries:

	IMonServer1=name11=value11,name12=value12,... IMonServer2=name21=value21,name22=value22,... IMonServer attributes: IpAddress: The IMonServer's IP address. Port: The IMonServer's port number. Do not change the default values set internally, unless instructed by iPass. Refer to iPass NetServer Documentation for more details.
Listener	List of the Listeners for this server. Expected format: Listener1=Type=<protocol>,Port=<port number>,IpAddress=<local IP address> Listener2=Type=<protocol>,Port=<port number>,IpAddress=<local IP address> Default Listeners are: Listener1=Port=577 NumOfThreads: You can improve connectivity to a RoamServer by increasing the number of threads accepting requests on port 577. This can be helpful for if your RoamServer is under heavier stress, such as 10 or more requests per second. For example: Listener1=Port=577,NumOfThreads=10 This is an advanced setting. The server may not function properly if this value is set incorrectly.
KeyStore	Provides KeyStore information. Sample format of this entry: KeyStore=name11=value11,name12=value12,... Below are the list of various KeyStore attributes: (These are advanced settings. The server will not start properly if these values are set incorrectly.) KeyStorePath: This entry determines the java keystore path. Default value for this property is set to C:\ipass\roamserver 6.1.0\certs\[rs.keystore] KeyPassword: This entry determines the password required to get keys from java keystore. Default value for this property is set to changeme KeyAlias: This entry determines the java keystore private key Alias. Default value for this property is set to rs CertAlias: This entry determines the java keystore trusted certificate alias. Default value for this property is set to ipassca KeyStorePassword: This entry determines the password required to open java keytore. Default value for this property is set to changeme Salt: This entry determines the salt used for encrypting KeyPassword and KeyStorePassword. Default value for this property is set to iPassRS
LogDirFileDeletionAge	LogDirFileDeletionAge=<age in days> Valid range is: 0 to 180 days. The default is 90 days. A value of 0 means deletion is

	DISABLED. LogDirFileDeletionAge determines how old files in the directory <iPass ServerHome>/logs must be before they are deleted. The check for file age is done only when the log file rotation happens. See page 34 for more information.
PolicyFile	PolicyFile=<Policy file Name> This entry, when present enables policy management (access control). The policy file contains a list of access control rules. Each rule can identify a country, class of service, a username, and whether roaming access is allowed or denied. This file can be created using the Policy Tool.
ReplyClass	ReplyClass=yes/no Configuration to enable passing Class attribute coming from the AAA server. When enabled, Roamserver will pass the Class attribute coming from AAA server. Default value is no (disabled). When disabled, Roamserver will block the Class attribute coming from AAA server. However, Roamserver may add its own Class attribute values even if ReplyClass is disabled.

RouteByRealm	RouteByRealm=yes/no
	Configuration to enable routing based on user realms (domains). When enabled, the RoutingRealm1, RoutingRealmX... are used to specify the servers to route to for a given realm. Default value is no.
	Routing by realm allows routing requests to specific AAA servers, based on the user's realm or domain. Routing can also be done by routing prefix. This allows you to use different types of authentication server, if necessary. For example, you could use both a RADIUS server and an LDAP server simultaneously. Requests from one domain, or with one prefix, can be directed to one server while requests from another domain or with another prefix can be directed to a second server. If routing by realm is enabled on your RoamServer, you will also need to set other properties to specify your other AAA servers, including RoutingRealm, Realm, AuthServer, AcctServer
	Example
	RouteByRealm=YES
	RoutingRealm1=Realm=example.com,AuthServer1=AuthServer1,
	AcctServer1=AcctServer1
	RoutingRealm2=Realm=XY, AuthServer1=AuthServer2,
	AcctServer1=AcctServer2
	RoutingRealm3=Realm=DEFAULT,AuthServer1=AuthServer1,AcctServer1= AcctServer 1

RouteByRealmScheme	RouteByRealmScheme=<scheme> Where <scheme> is either EndsWith or StartsWith. The default is EndsWith. RouteByRealmScheme indicates how the RoutingRealm properties are matched up with the domain (or realm) of the incoming user request. See page 34 for more information on routing by realm.
RoutingRealm	RoutingRealm=<valid domain or routing prefix> See also RouteByRealm for examples of proper use and formatting.
ServerInfold	This feature is not currently in use.

`StartUpMessage`	`StartUpMessage=yes/no.` This entry determines if a message is generated by the server on startup. This is an advanced setting. The server may not function properly if this value is set incorrectly. Default value for this property is set to no (startup messages are turned off)
`StoreFailedAcct`	`StoreFailedAcct=yes/no  or true/false.` Determines if the RoamServer will store accounting to a local file if it fails to communicate with any and all of the AAA accounting servers. The `resendacct` tool can then be used to resend each of those accounting records to the RoamServer once the AAA is back up. Default setting is: `false`
`StripDeviceInfo`	StripDeviceInfo=yes/no or true/false.When StripDeviceInfo is set to true and the request is ACA accounting request then RoamServer will strip the Device info from userid. Default value for this property is set to true.
`TraceLogBackupType`	`TraceLogBackupType=<backupType>` Where <backupType> is either MultipleWithTimestamp or SingleBackup. The default is SingleBackup. `TraceLogBackupType` sets the trace log's backup file name when rotation is to be performed on the local trace files.
`TraceLogRotationHours`	`TraceLogRotationHours=<hours>` Valid range is: 1 to 720 hours. The default is 168 hours (1 week). `TraceLogRotationHours` controls how often the local trace file is rotated.
`TraceLogRotationMaxSize`	`TraceLogRotationMaxSize=<max size>` Minimum value is 100 kB. Maximum value is 20000 kB. The default is 10000 kB. `TraceLogRotationMaxSize` limits how large (in kilobytes) the local trace file can get before it is rotated.
`UpdateInterval`	`UpdateInterval=<DayOfWeek Hour:Minute>` Where `DayOfWeek` ranges from Sunday to Saturday and `Hour` is between 0-23. Default value for this property is set to Monday 2:00. This entry determines when RoamServer contacts the update server. Note: The `UpdateInterval` mechanism synchronizes with the system clock every sixty minutes. See also `AutoUpdate`.

`UpdateServer`	Provides iPass software Update Server information. Sample format of the entries:
	`UpdateServer1=name11=value11,name12=value12,...`
	`UpdateServer2=name21=value21,name22=value22,...`
	`UpdateServer` attributes:
	■ `IpAddress:` The URL of the iPass software update server
	■ `RetryDelay:` The time delay, in minutes, before retrying a server that recently failed a connection request. When a connection fails to a server, it is reordered to the end of the list. Once the `RetryDelay` expires, that server is brought back to the top of the list. The default value is 15 minutes. Valid range is: >=0.
	■ `FailureThreshold:` Once the failure count exceeds the `FailureThreshold`, the server is reordered to the end of the list. The default value is 0.
	Refer to iPass NetServer Documentation for more details.

`UploadServer`	Provides iPass software Upload Server information. Sample format of the entries:
	`UploadServer1=name11=value11,name12=value12,...`
	`UploadServer2=name21=value21,name22=value22,...`
	`UploadServer` attributes:
	■ `IpAddress:` The URL of the iPass software update server

	■ `RetryDelay:` The time delay, in minutes, before retrying a server that recently failed a connection request. When a connection fails to a server, it is reordered to the end of the list. Once the `RetryDelay` expires, that server is brought back to the top of the list. The default value is 15 minutes. Valid range is: >=0. ■ `FailureThreshold:` Once the failure count exceeds the `FailureThreshold`, the server is reordered to the end of the list. The default value is 0. Refer to iPass NetServer Documentation for more details.
`UsePolicyFile`	`UsePolicyFile=y/n` This property determines if the server uses policy file for authentication. Default value for this property is set to n. This is an advanced setting. The server may not function properly if this value is set incorrectly
`ZipLogFilesEnabled`	`ZipLogFilesEnabled=true/false.` Determines whether or not trace and log files are zipped. Default is set to `true`.

ipassLDAP.properties

When configuring LDAP authentication, you can specify a path to a text file containing special LDAP settings named `ipassLDAP.properties`. This section discusses configuration for this file.

User-Configurable Options

This table summarizes the configurable options in `ipassLDAP.properties`. When an `ipassLDAP.properties` file is not present, or if an option is not specified, the default values will be used.

Property	Default Value	Comments
LdapBaseDn	NULL	Specifies base DNs to be used during LDAP authentication. When configured, it will be appended to the LdapExactMatchRdn during exact match bind and used as a search base during the LDAP search operation. Any variables supplied in the format of $VARIABLE will be replaced with the actual value of that variable. The current variables supported are $USERID, $PREFIX and $DOMAIN. If no LdapBaseDn is configured, then no anonymous bind and search will be performed. Multiple base DNs (more than one line) are permitted in the ipassLDAP.properties file. When multiple base DNs are configured, the authentication process will use them in the order they appear in the ipassLDAP.properties file. If authentication fails using the first LdapBaseDn, authentication will be re-attempted using the second LdapBaseDn and so on. Since a base DN is added on to the login name when an exact match bind is performed, if a user logs on using a full DN (uid=Joe,ou=people,o=example.com), LdapBaseDn should not be because performance will be reduced. Examples: LdapBaseDn=ou=people,o=example.com LdapBaseDn=o=example.com LdapBaseDn=dc=company,dc=com
LdapBindDn	NULL	For LDAP servers that do not support anonymous binds, this configuration will set a specific DN to be used for binding to the LDAP server, before performing a search operation. When anonymous binds are supported,

		omit this configuration and the default value of NULL will be used. Example: LdapBindDn=uid=bindmaster,ou=people,o=example.com
LdapBindPasswd	NULL	For LDAP servers that do not support anonymous binds, this configuration will set a password to be used for binding to the LDAP server before performing a search operation. When anonymous binds are supported, omit this configuration and the default value of NULL will be used. Example: LdapBindPassword=bindUserPassword
LdapBindPasswdEncr	NULL	This property has been added to keep the bind password of LDAP files in encrypted format. After server startup, the LdapBindPassword attribute will change to LdapBindPasswdEncr attribute with password encrypted.
LdapCompareAttr	NULL	Configuration to enable comparison of user passwords against a specific user attribute in the LDAP directory as a means of authentication. The user attribute specified must contain a password saved in clear text in the LDAP directory for LdapCompareAttr to work. This compare replaces the final user bind to authenticate the user. The user bind authenticates against the standard password attribute (usually userpassword), which may or may not be encrypted in the LDAP directory. Example: LdapCompareAttr=roamingPassword
LdapDetectBaseDn	YES	When LdapDetectBaseDn is enabled, and no LdapBaseDn is configured, it will detect all the available BaseDn (a.k.a. namingContexts) of the LDAP server. Valid values: YES or NO.
LdapDoExactMatch	NO	Disables or enables binding directly to the LDAP server for user authentication using only the user's login id, password, and any base DN by the LdapBaseDn configuration. Accepted values are YES or NO. Example: LdapDoExactMatch=YES
LdapExactMatchRdn	uid=$USERID	The DN used for the exact match bind is comprised of two parts: the relative DN (RDN) and the base DN. The base portion can be

		specified by the LdapBaseDn configuration. The relative DN format can be specified by the LdapExactMatchRdn. The RDN is by default uid=$USERID, where the variable $USERID is replaced by the username specified at login time. The current variables supported are $USERID and $DOMAIN. For example: User *joe* exists in a LDAP tree with a DN of uid=joe,ou=people,o=example.com, and he logs in as *joe@example.com*. For a successful exact match bind, leave the LdapExactMatchRdn as default and set the LdapBaseDn=ou=people,o=example.com. User *Mary* exists in a LDAP tree with a DN of cn=Mary,dc=company,dc=com, and she logs in as *Mary@example.com*. For a successful exact match bind, set the LdapExactMatchRdn=cn=$USERID and set the LdapBaseDn=dc=company,dc=com. The exact match bind can be disabled by setting LdapDoExactMatch=NO. Only one LdapExactMatchRdn (one line) is allowed in the ipassLDAP.properties file. Examples: LdapExactMatchRdn=cn=$USERID LdapExactMatchRdn=$USERID
LdapExcludeWildcards	NULL	This property is used to exclude wildcards (special characters) from the default set of LDAP wildcards &|~=^!<>*()+. These wildcards could be used for LDAP Blind Injection, so excluding them is not recommended. Default is none. Example: LdapExcludeWildcards=& or LdapExcludeWildcards=&|~=^!<>*()+
LdapGroupDepth	3	Can be used in conjunction with LdapMemberOfGroup to limit the depth of the search for nested groups. Valid values are from 1 to 10. A value of 1 would avoid any nested group search and only look for direct group memberships.

`LdapIgnoreExpiredAdPassword`	NO	If set to YES, RoamServer will allow access by ignoring expired Active Directory (AD) passwords.
`LdapMemberOfGroup`	NULL	This property will enable verification that a user is a member of a given group in Active Directory. RoamServer compares the given group DN to the attribute and any subsequent nested groups, up to a maximum depth of 10 nested groups. Example: `LdapMemberOfGroup=CN=CompanyUsers,CN=Users,DC=CorporateHQ,DC=company,DC=com`

`LdapSearchFilter`	uid=$USERID	Specifies a custom filter when searching an LDAP server for a user. If this option is not set, the default filter (`uid=$USERID`) will be used. When an exact match bind is disabled or is unsuccessful, an anonymous bind and search will be used. A custom filter may be supplied for the search. Any variables supplied in the format of `$VARIABLE` will be replaced with the actual value of that variable. The current variables supported are `$USERID, $PREFIX` and `$DOMAIN.`
		Only one filter (one line) is presently allowed in the `ipassLDAP.properties` file.
		The variables' values are taken from the user's login. For example if someone logs in as *joe@example.com*, the variable `$USERID` would be replaced by *joe* (that is, everything to the left of the leftmost @-sign, not including any prefix such as *iPass/*). The variable `$DOMAIN` would be replaced by *example.com* (that is, everything to the right of the leftmost @-sign).
		For example: if the search filter is (&(mail=$USERID@$DOMAIN)(dialup=true)), when joe from example.com logs on, the search filter will be converted to (&(`mail=joe@example.com`) `(dialup=true))`
		Examples:
		`LdapSearchFilter=uid=$USERID`
		`LdapSearchFilter=mail=$USERID@$DOMAIN`
		`LdapSearchFilter=(&(uid=$USERID)(dialup=true))`
		`Class_of_service_str` can also be used as a valid attribute for the search query. Valid values for this attribute are: `DIAL-UP, DIAL-UP-ISDN, DIAL-UP-PHS, WIRED, WIRELESS, MOBILEDATA.`

		Example: `LdapSearchFilter=(&(sAMAccountName=$USERID)(memberOOf=CN=$(class_of_service_str),CN=Users,DC= company,DC=com))`
`LdapSearchMoreServers`	NO	Uncomment and customize the `LdapSearchMoreServers` line to enable/disable searching other LDAP servers when the user is not found on the current LDAP server. Valid values are YES or NO. Default value is NO. Note to Active Directory (AD) users: you will, in most cases, need this enabled to YES.
`LdapSearchScope`	2	Determines the scope of the LDAP search. Valid values are: 0=Object Scope, 1=One Level Scope , 2=Subtree Scope
LdapUacAttr	userAccountControl=512,544,66048,66080,262656,262688,328192,328224	To enable ACA support with LDAP. Given default value only applicable for Active Directory. For LDAP, configure your customized user account control attribute with value associated for active users. example: LdapUacAttr=ipassStatus=active or LdapUacAttr=userStatus=enable or LdapUacAttr=userEnabled=true

Using Active Directory

When using Active Directory, configure RoamServer to point to any domain controller server when setting up your authentication server. AD listens on TCP port 389, but for large AD 'forests', you may consider configuring RoamServer to point to Global Catalog DCs on TCP port 3268.

Normal LDAP traffic on port 389 to AD DCs will not support 'referral chasing' for object binds outside of the resident domain which the DC resides in. To be able to authenticate users in other domains in your organization, RoamServer needs to authenticate against a Global Catalog DC in any domain, preferably at the root of the forest.

The error codes returned by Active Directory are the hexadecimal numbers of the Microsoft System Error Codes. You can convert a hex number to a decimal number and look up the corresponding error code on the Microsoft Website at:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/debug/base/system_error_codes.asp

Here is an example of an iPassLDAP.properties file configured for use with Active Directory. All lines in ipassLDAP.properties prefaced with a space or # sign are ignored.

```
# File: ipassLDAP.properties.example
#
# Description: Contains configurations for customizing LDAP authentication.
#       The AuthServer's LdapConfigFile property must be set
```

```
    #       in ipassRS.properties for RoamServer to use this file.
    #
    # Blank lines and lines beginning with # or spaces are ignored.
    #
    ############################################################
    # Sample for Active  Directory (AD) users:#
    ############################################################
    LdapBaseDn1=dc=company,dc=com
    LdapSearchFilter=sAMAccountName=$USERID
LdapBindDn=cn=bindUser,cn=Users,dc=dev,dc=company,dc=com
LdapBindPassword=bindUserPassword
    LdapDetectBaseDn=YES
    #LdapSearchMoreServers=YES
    #LdapCompareAttr=someUserAttribute
    #LdapDetectBaseDn=YES
    #LdapMemberOfGroup=CN=iPassUsers,CN=Users,DC=company,DC=com
    #LdapGroupDepth=3
    #LdapSearchScope=2
    ############################################################
    # Sample for LDAP:      #
    ############################################################
    #LdapBaseDn1=o=company.com
    #LdapSearchFilter=uid=$USERID
    #LdapDetectBaseDn=YES
    #LdapSearchScope=2
    #LdapDoExactMatch=NO
    #LdapExactMatchRdn=uid=$USERID
    #LdapBindDn=uid=bindUser,ou=people,o=company.com
    #LdapBindPassword=bindUserPassword
    #LdapCompareAttr=someUserAttribute
    #LdapSearchMoreServers=YES
    ############################################################
    # More  Documentation on the settings above         #
    ############################################################
    ############################################################
    #
    # Uncomment and customize the 'LdapBaseDn' line to set a search base.
    # Important: a minimum of 1 'LdapBaseDn' is required for a search to occur.
    # Supported variables are USERID, PREFIX and DOMAIN.
    # Default is no base DN.
    #
    #       LdapBaseDn1= o=company1.com
    #       LdapBaseDn2= o=$DOMAIN
    #
    # Sample for Active  Directory (AD) users:
    #       LdapBaseDn1= dc=company,dc=com
    #
    ############################################################
    ############################################################
    #
    # Uncomment and customize the 'LdapSearchFilter' line to set a search filter.
    # Supported variables are USERID, PREFIX and DOMAIN.
    # Default is "uid=$USERID".
    #
    #       LdapSearchFilter=uid=$USERID
    #
    # NOTE to Active  Directory (AD) users: you will
    # need to configure this property for searches.
    #
    # Most common filter is:
    #
    #       LdapSearchFilter=sAMAccountName=$USERID
    #
    # Search filter to find a member of a group
    #
    #
    LdapSearchFilter=(&(sAMAccountName=$USERID)(memberOf=CN=iPassUsers,CN=Users,DC=co
mp any,DC=com))
    #
    # Search filter to find a member of a group using the class_of_service_str
    # iPass attribute (wrapped with ${} ) from the incoming auth_request packet:
    #
    #
    LdapSearchFilter=(&(sAMAccountName=$USERID)(memberOf=CN=${class_of_service_str},CN=
    Users,DC=company,DC=com))
    #
    # Valid values for class_of_service_str are:
    #
    # DIAL-UP,DIAL-UP-ISDN,DIAL-UP-PHS,WIRED,WIRELESS
    #
    ####################################################################
    ####################################################################
    #
    # When LdapDetectBaseDn is enabled, and no LdapBaseDn is configured,
    # it will detect all the available BaseDn (a.k.a. namingContexts) of the LDAP server.
    # Options: NO or YES
    # Default: YES
    #
    #       LdapDetectBaseDn=YES
    #
    ####################################################################
    ####################################################################
    #
    # Property to enable verifying that a user is a member of
    # a given group in Active  Directory.
    # This is a special feature to handle nested groups.
    # It compares the given <Group DN> to the memberOf attribute of the user
    # and any subsequent nested groups, up to a max depth of 10 nested groups.
    # Default is none.
    #
    #       LdapMemberOfGroup=CN=iPassUsers,CN=Users,DC=company,DC=com
    #
    ####################################################################
    ####################################################################
    #
    # This property can be used in conjunction with the LdapMemberOfGroup feature
    # to limit the depth of which we search for nested groups.
    # The valid range is from 1 to 10.
    # A value of 1 would avoid any nested group search and only look at the
    # user's memberOf attribute for direct group memberships.
    # The default depth is 3.
    #
    #       LdapGroupDepth=3
    #
    ####################################################################
    ####################################################################
    #
    # Search Scope.
    # Valid values are:
    #       0 (Object Scope)
    #       1 (OneLevel Scope)
    #       2 (Subtree Scope)
    # Default is 2 (Subtree Scope).
    #
    #       LdapSearchScope=2
    #
    ############################################################
    ############################################################
    #
    # Uncomment the following to enable the exact match bind. This is
    # recommended when the LDAP search is not needed. Options: NO or YES.
    # Default is NO.
    #
    #       LdapDoExactMatch=YES
    #
    ############################################################
    ############################################################
    #
    # Uncomment and customize the 'LdapExactMatchRdn' line to specify the RDN
    # format for the exact match bind. Supported variables are USERID, PREFIX and DOMAIN.
    # Note that the LdapExactMatchRdn will be concatenated with the LdapBaseDn
    # to formulate the exact match DN.
    # Default is "uid=$USERID".
    #
    #       LdapExactMatchRdn=uid=$USERID,o=company.com
    #
    ############################################################
    ############################################################
    #
    # Uncomment and customize the 'LdapBindDn' and 'LdapBindPassword' lines
    # if your LDAP server does not support anonymous binds.
    # Default is none.
    #
    #       LdapBindDn=uid=test,ou=people,o=company.com
    #       LdapBindPassword=test
    #
    # NOTE to Active Directory (AD) users: you will
    # need to configure these properties for binding.
    #
    #       LdapBindDn=cn=bindUser,cn=Users,dc=company,dc=com
    #       LdapBindPassword=bindUserPassword
    #
    ############################################################
    ############################################################
    #
    # Uncomment and customize the 'LdapCompareAttr' line to specify a user attribute
    # to compare the password with when authenticating. NOTE: This will replace
    # the final user bind for authenticating.
    # Default is none.
    #
    #       LdapCompareAttr=someUserAttribute
    #
    ############################################################
    ############################################################
    #
    # Uncomment and customize the 'LdapSearchMoreServers' line
    # to enable/disable searching other LDAP servers
```

```
    # when the user is not found on the current LDAP server.
    # Valid values are YES or NO. Default value is NO.
    #
    # NOTE to Active Directory (AD) users: you will,
    # in most cases, need this enabled to YES.
    #
    #       LdapSearchMoreServers=YES
#
####################################################################
####################################################################
#
# Uncomment and customize the 'LdapUacAttr' line to enable
# ACA support with LDAP.
# Default value is (for AD):
#           LdapUacAttr=userAccountControl=512,544,66048,66080,262656,262688,328192,328224
#
# For LDAP, configure your customized user account control attribute
# with value associated for active users.
# example:
#           LdapUacAttr=ipassStatus=active
#                       or
#           LdapUacAttr=userStatus=enable
#                       or
#           LdapUacAttr=userEnabled=true
#
####################################################################
####################################################################
#
# Uncomment and customize the 'LdapExcludeWildcards' line to
# exclude wildcards (special characters) from the default set of
# LDAP wildcards &|~=^!<>*()+. These wildcards could be used for
# LDAP Blind Injection, so excluding them is not recommended.
# Default is none.
# example:
#           LdapExcludeWildcards=&
#                       or
#           LdapExcludeWildcards=&|~=^!<>*()+
#
####################################################################

####################################################################
#
# Uncomment the following to allow access to Active Directory
# (AD) by ignoring expired passwords.
# Default is NO.
#
#           LdapIgnoreExpiredAdPassword = YES
#
####################################################################

####################################################################
#
# Uncomment the following to enable the printing of detailed
# debug information for the various phases of SSL/TLS handshaking
# between the RoamServer and the Secure LDAP server.
# Note: The RoamServer must be restarted for this property to take effect.
# Default is NO.
#
#           LdapSslDebug = YES
#
####################################################################
```

```
#
```

LDAP Authentication and RoamServer

Action 1: Exact match is used to authenticate the user, which means a bind to the LDAP server is performed using an exact match DN (Domain Name) and the user's password. The exact match DN is attached with any base DN specified in the `ipassLDAP.properties file.` The user portion (Relative DN) of the exact match DN is by default `uid=username,` but it can be customized with the `LdapExactMatchRdn` configuration in the `ipassLDAP.properties file.` The exact match operation can be disabled bysetting `LdapDoExactMatch=no` in the `ipassLDAP.properties file.`

Action 2: Anonymous bind and search is used to authenticate the user, which means a bind to the LDAP server is performed using a `NULL` userid and password. If anonymous binds are not supported bythe LDAP server, a `LdapBindDn` and `LdapBindPasswd` can be specified in the `ipassLDAP.properties file.`

After a successful bind, search the LDAP directory for the user starting from a base DN as specified by the `LdapBaseDn` and filtering with the `LdapSearchFilter.` If a user (and only one user) is found during the search, a simple bind to the LDAP server will be performed to authenticate the user. This last authentication will be done using the DN of the user found during the search and the password supplied at login time.

The anonymous bind and search will not be performed if the user was authenticated during the exact match, or if no `LdapBaseDn` was specified in the `ipassLDAP.properties` file.

Appendix I: Error Messages

This section lists error messages that can be returned by the RoamServer at Debug Levels 0, 1 and 2. Although other debug levels are possible, they are used only for packet dumps and no error messages are associated with them.

Variables denoted in the list by + (for example, `+ioe.getMessage()`) will be replaced at runtime with specific data.

Feature	Debug Level	Message
Tacacs+		
	1	Error occurred while trying to communicate to the TACACS+
	1	Failed to convert TACACS+ packet to bytes
	1	"Failed to open TCP socket to TACACS+ server: IO Error, "+ioe.getMessage()
	1	"Failed to open TCP socket to TACACS+ server:
	1	Failed to send packet to TACACS+ server" +ioe.getMessage()
	1	Unexpected NULL clientSocket, socket could be closed.
	1	Timed Out reading packet from TACACS+ server "
	1	"Failed to read packet from TACACS+ server "
	1	Cannot parse raw TACACS+ packet
	1	"Error closing socket to TACACS+ " +ioe.getMessage()
	1	"ERROR parsing header of packet received from TACACS+
	1	"Unsupported reply packet type " +this.hdr_type +" received from
	1	"ERROR decrypting TACACS+ packet"
	1	"ERROR: missing TACACS+ packet type"
	1	"parse() not supported for this reply packet type " +pktType
	1	"ERROR: missing TACACS+ packet type"
	1	"ERROR: toBytes() not supported for packet type " +pktType
	1	"ERROR encrypting TACACS+ packet"
	1	"CHAP challenge conversion failed."
	1	"CHAP password conversion failed."
	1	"ERROR encrypting TACACS+ packet"
	2	Error or Timeout in getting reply from TACACS+ server
	2	Password is NULL, TACACS+ Minor Version 0 does not support CHAP
	2	Error/Timeout getting first auth reply from TACACS+ server
LDAP		
	0	"Server's LDAP Info is Missing "
	0	"Unexpected return code (" +rc +")"
	0	"Internal Error: LDAP server address not set"
	1	"Illegal LDAP Configuration: Must configure an "+LdapInfo.LDAP_BASE_DN +" or Enable "+LdapInfo.LDAP_DO_EXACT_MATCH
	1	"Error creating RDN from ldapExactMatchRdn"
	1	"ExactMatchBind failed " +ne.getMessage()
	1	"Error creating Search Filter."
	1	"LDAP Authentication failed " +reason
	1	"Error, LDAP search found multiple matches "+entryCount +" found
	1	"LDAP Search found multiple matches for this user " +slee.getMessage()

	1	"LDAP Search exceeded " +searchTimeout +" millisecond time limit:
	1	"LDAP Search Error: " +ne.getMessage()
	1	"LDAP Compare of (" +name +") attribute with password failed."
	1	"LDAP Compare of (" +name +") attribute failed: " +ne.getMessage()
	1	"Unexpected NULL ldap context"
	1	"Invalid attribute name: "+attrName+", in line: "+origString
	1	"Could not authenticate user at this LDAP server"
	1	"TIMEOUT while talking to LDAP server after " +sInfo.NumRetry +"
	2	"Error while closing connection to LDAP server" +ne.getMessage()
SSLPost		
	0	fileDesc+fileName+" does not exist"
	0	"Cannot read "+fileDesc+filename
	0	"Failed to instanciate SSLPostCommunicator: "+cce.getMessage()
	0	"Could not instantiate SSLSocketImpl"
	0	"ERROR: Missing IpassDictionaryEntry"
	1	"Socket receive timed out"
	1	"Failed to receive data from server: " + serverInfoRec.IpAddress
	1	"IOEXCEPTION: while talking to server: " + serverInfoRec.IpAddress + ":" +
	1	"received null Communicator object"
	1	"received null serverInfoRec"
	1	"received null requestPkt"
	1	"received null replyPkt"
	1	"Could not create sslSocket: doHandshake failed"
	1	"Could not create sslSocket: Instantiation failed"
	1	"sslSocket null for ServerSide communicator"
	1	"Could parse post packet: " +replyStr
	1	Malformed Post Packet
	1	"Malformed post packet header"

	1	"Unexpected NULL sslSocket."
	1	"Error parsing MultiInstance attribute "+name+", of type " +de.getType()
	1	"Error parsing attribute "+name+", of type " +de.getType()
	1	"Error in converting the packet to bytes: " + e.toString()
	1	"Error for attribute "+name+ ": "+i.getMessage()+" Ignoring it"
	1	"Dropping attribute for ipassCode " +ipassCode+" value "+value+",
	1	Base64 Decode ERROR: Dropping OBJECT of ipassCode " +ipassCode+"
	1	"Dropping OBJECT of ipassCode " +ipassCode+" value "+value+", OptionalDataException: "+o.getMessage()
	1	"Dropping OBJECT of ipassCode " +ipassCode+" value "+value+",
	1	"Dropping OBJECT of ipassCode " +ipassCode+" value "+value+", IOException: "+i.getMessage()
	1	"Dropping attribute for ipassCode " +ipassCode+" value "+value+",
	2	"NULL sslServerSocket, listener socket could be closed."
	2	"SSL handshake failed, closing accepted socket."
	2	"Listeners are shutdown, closing accepted socket."
	2	"Rejecting packet from: " +sslSocket.getHost()
	2	"Error: No ipassPkt to send"
	2	"Unexpected NULL sslSocket, socket could be closed."
	2	"Could parse post packet: " +packetStr
	2	"Error parsing IpassPostPkt: Unknown URI/request type " +uri
	2	"Error parsing IpassPostPkt: missing empty string."
	2	"Error parsing IpassPostPkt."
	2	"Unknown PostPkt attribute (" +name +"): ignoring it."
Handlers		
	0	"Software update failed"
	0	"Download failed"
	0	"Error occurred while trying to instantiate RSPolicyRules: " + i.getMessage()
	0	"Error occurred while adding policy rule: An entry with the same rule:" + id + “ exists!"
	0	"File "+policyFile+" not found"

	0	"Failed to Shutdown due to policy errors as the TransactionController is null"
	0	"Failed to Shutdown due to policy errors as the TransactionContext is null"
	0	Cannot find TRANSACTION CONTROLLER
	0	Cannot find exceptionHandler
	0	Could not get LOCAL_HOST_IP
	0	Error occurred while trying to instantiate " + s.toString()
	0	Error occurred while trying to send the reply packet
	0	No Server found for the following transaction type: "+ reqTypeName
	0	No valid handler found for the request of type "+type);
	0	ERROR occurred while trying to save the acct record in a file: "+i.getMessage())
	0	Error occurred while trying to instantiate RSAcctReqHandler: " +
	0	Unexpected ERROR: "+Config.FAILED_ACCT_LOG_DIR+" property not set!
	0	Could not create directory \"" + failedDirPath + "\" to store
	0	ERROR, expected "+Config.FAILED_ACCT_LOG_DIR+" to be a directory, got \"" + failedDirPath + "\" instead.
	1	"Software Update Failed due to failure to load the Server's Version Table."
	1	"Unable to copy "+this.serverJarFileName +" to "+this.updatefilesJarFileName
	1	"User " + user_id + " is denied access based on the policy rule:
	1	"IO error in loading policy File "+policyFile
	1	"Error loading the policy file"
	1	"Cannot get SSLPOST listener port, defaullting to:" + UNKNOWN PORT
	1	"Failed to handle Heartbeat message!"
	1	"Failed to load RS Policy Rules: "+se.getMessage()
	1	"Policy Restriction. Verify Policy Failed."
	1	"Authentication Rejected: Invalid Reply Packet"
	1	"ERROR: list lock is NULL. Cannot check for duplicates in our
	1	"exception ocurred: " + e.toString()
	1	"ERROR: list lock is NULL. Cannot add entry to our accessList"
	1	"No such hashing alrorithm error: "+nsae.getMessage()

	1	handleRequest-Communicator object is null
	1	Error: File: " + fileName + " does not exist on the server
	1	Error: File: " + fileName + " content is empty!
	1	failed to get file contents
	1	Invalid Request: Failed to get the path of the file: " + fileName
	1	Invalid Request: Cannot return the files in the keys folder!
	1	Invalid Request: filename is not from the $ipass.server.home: " +
	1	Invalid Request: File:" + fileName + " does not exist on the server!
	1	"Invalid Request: File name not specified!
	1	handleRequest-Communicator object is null
	1	Failed to reload the new config file, reverted to the old config file...
	1	Invalid request, Failed to Reload the new config file, and failed
	1	Invalid request, Failed to Reload the new config file, and failed to find the " + fileName + ".bak in order to
	1	Invalid request, Failed to Reload the new config file, and failed to delete it.\nPlease copy the " + fileName + ".bak
	1	Failed to rename " + fileName + " to " + fileName + ".bak"
	1	Failed to delete " + fileName + ".bak"
	1	Error, Config Filename could not be obtained!
	1	source Ip is null, not a valid CTRL_MSG_IP
	1	netSourceIp +" is not a valid/configured CTRL_MSG_IP
	1	Invalid Request: File contents are empty!
	1	Invalid Request: Failed to load the config changes: " + e.getMessage()
	1	Protocol is not supported by current version of software: Server ID=" + serverInfoRec.ServerInfoId + ", Server
	1	ERROR: Cannot get communicator for server IP: " + serverInfoRec.IpAddress + ", of Protocol: " +
	1	"No Servers found: Null returned from getRoute()"
	2	netSourceIp +" is not a valid/configured CTRL_MSG_IP");
RADIUS		
	0	Failed to open DatagramSocket

	0	Cannot get LOCAL_HOST_IP, unable to set NAS_IP in RADIUS packet
	0	IOException on listener for port "+serverPort+": "+e.getMessage();
	0	IOException on listener for port is due to RADIUS Listeners being
		shutdown
	0	ERROR creating the UDP socket at port "+port+". (Port may be in use)");
	0	Failed to instanciate SharedSSLPostCommunicator
	1	Unexpected NULL socket, socket could be closed
	1	IOException on DatagramSocket
	1	Error occurred while trying to talk to AAA server
	1	Failed to communicate with radius server after " +sInfo.NumRetry
	1	RADIUSPkt parsing errors
	1	Input not a byte array
	1	Empty RADIUS data
	1	Illegal type in RADIUS packet
	1	Missing identifier in the RADIUS packet
	1	Missing Length in the RADIUS packet
	1	Missing authenticator in the RADIUS packet
	1	Missing code in the RADIUS packet
	1	Missing length in the RADIUS packet
	1	ERROR: Invalid CHAP_PASSWD length of "+dataLen
	1	ERROR: Invalid MESSAGE_AUTHENTICATOR length of "+dataLen
	1	Missing IpassDictionaryEntry for radius code " + code
	1	Illegal data type
	1	Malformed radius packet (When data length is longer than the packet header specified)
	1	ERROR: missing MESSAGE_AUTHENTICATOR to validate EAP-Message
	1	ERROR: missing Request Authenticator to validate EAP-Message
	1	ERROR: failed to re-calculate Message-Authenticator"
	1	ERROR: Invalid Message-Authenticator

	1	ERROR: missing Request Authenticator
	1	ERROR: failed to generate test Authenticator
	1	ERROR: missing Response Authenticator
	1	ERROR: Invalid Response Authenticator
	1	No such algorithm
	1	Digest Exception
	1	No valid RADIUS code for Ipass Packet Type "+getPktType()+" Status "+status
	1	Missing IDENTIFIER header attribute, using value of "+ident+"
	1	Error: CHAP Identifier missing from packet
	1	CHAP password conversion failed.
	1	CHAP challenge conversion failed.
	1	ERROR: missing Shared Secret to calculate the Message Authenticator
	1	ERROR: when calculating HMAC digest of Message Authenticator
	1	ERROR: Request Authenticator is missing.
	1	Unsupported encoding exception
	1	NoSuchAlgorithmException
	1	Exception: " + e.toString());
	1	ERROR: missing Shared Secret
	1	ERROR: Base64 Decode of iPass Attribute " +ipassAttrCode +" Failed
	1	WARNING: Unable to get Dictionary entry for iPass Attribute
	1	ERROR: UTF8 conversion of iPass Attribute " +ipassAttrC
	1	ERROR: Base64 Decode of iPass Attribute " +ipassAttrCode +" failed
	1	ERROR: Base64 Decode Vendor Specific Attribute " +vendorId+":"+vendorType +"
	1	ERROR: Invalid Vendor Specific Attribute format
	1	Vendor ID missing from Vendor Specific Attribute
	1	Vendor Type missing from Vendor Specific Attribute (VendorID="+vendorId+
	1	Vendor Length missing from Vendor Specific Attribute (VendorID="+vendorId+", VendorType="+vendorType+
	1	Value missing from Vendor Specific Attribute (VendorID="+vendorId+", VendorType="+vendorType

	1	Value from Vendor Specific Attribute is corrupted. (VendorID="+vendorId+", VendorType="+vendorType
	1	expected len was "+vendorValueBytes.length
	1	Cannot convert attribute "+attr +", RADIUSType type of IPADDRESS
	1	Cannot convert attribute "+attr +", RADIUSType of Integer to iPassType " +iPassType
	1	Unsupported iPass attribute " +attr +", with radius value " +radiusValue
	1	NULL input: key is null
	1	NULL input: text is null
	1	Hashing error
	1	No such hashing algorithm error
	2	Cannot parse raw packet
	2	Receive timeout set to " +sInfo.IdleTimeout milliseconds
	2	RADIUSBufferSize error
	2	NULL serverSocket, listener socket could be closed.
	2	Started RADIUS Listener "+i +" on port "+listenerThreads[i].getServerPort());
	2	Cannot convert attribute "+attr +", RADIUSType of TEXT to iPassType " +iPassType
	2	Unsupported String Encoding: " +attr +", with radius Type " +radiusType
	2	Cannot convert attribute "+attr +", RADIUSType of String to iPassType " +iPassType
	2	Cannot convert to Integer: "+attr +", with radius Type " +radiusType
	2	Cannot convert attribute "+attr +", RADIUSType Time to iPassType
	2	Cannot convert attribute "+attr +", RADIUSType BYTEARRAY to iPass type " + iPassType
	2	Illegal data type " + radiusType
Site		
	0	Failed to load SiteCommunicator library
	1	Error occurred while trying to do Site file authentication
	2	Failed talking to SITE server
Unix		
	0	Failed to load UnixCommunicator library
	1	Error occurred while trying to do UNIX authentication

	2	Failed talking to Unix server
NT and NT RAS		
	2	Received authentication accept packet from Windows Server
	2	Received authentication reject packet from Windows Server
AcctFile		
	1	Failed to write to local AcctFile
	1	Error occurred while trying to talk to Windows server
	1	Failed talking to Windows server
	2	Received unexpected null packet when writing to local AcctFile

Appendix II: RADIUS Attributes

When using RoamServer 5.x with RADIUS authentication, check your RADIUS logs to verify your RFC attributes. If an attribute is not shown in the tables here, then you need to re-configure your RADIUS to eliminate the attribute.

RADIUS Authentication Attributes

This table shows which attributes may be found in which kinds of packets, and in what quantity. On the table:\

- **0:** This attribute must not be present in the packet.
- **0+:** Zero or more instances of this attribute may be present in the packet.
- **0-1:** Zero or one instance of this attribute may be present in the packet.
- **1:** Exactly one instance of this attribute must be present in the packet.

Request	Accept	Reject	Challenge	#	Attribute	Notes
0-1	0-1	0	0	1	`User-Name`	
0-1	0	0	0	2	`User-Password`	An Access-Request must contain either a `User-Password` or a `CHAP-Password` or `State`. An Access-Request must *not* contain both a `User-Password` and a `CHAP-Password`. If future extensions allow other kinds of authentication information to be conveyed, the attribute for that can be used in an Access-Request instead of `User-Password` or `CHAP-Password`.
0-1	0	0	0	3	`CHAP-Password`	An Access-Request must contain either a `User-Password` or a `CHAP-Password` or `State`. An Access-Request must *not* contain both a `User-Password` and a `CHAP-Password`. If future extensions allow other kinds of authentication information to be conveyed, the attribute for that can be used in an Access-Request instead of `User-Password` or `CHAP-Password`.
0-1	0	0	0	4	`NAS-IP-Address`	An Access-Request must contain either a `NAS-IP-Address` or a `NAS-Identifier` (or both).
0-1	0	0	0	5	`NAS-Port`	
0-1	0-1	0	0	6	`Service-Type`	An Access-Request must contain either a `NAS-IP-Address` or a `NAS-Identifier` (or both).
0-1	0-1	0	0	7	`Framed-Protocol`	
0-1	0-1	0	0	8	`Framed-IP-Address`	
0-1	0-1	0	0	9	`Framed-IP-Netmask`	
0	0-1	0	0	10	`Framed-Routing`	
0	0+	0	0	11	`Filter-Id`	
0-1	0-1	0	0	12	`Framed-MTU`	
0+	0+	0	0	13	`Framed-Compression`	

0+	0+	0	0	14	Login-IP-Host	
0	0-1	0	0	15	Login-Service	
0	0-1	0	0	16	Login-TCP-Port	
0	0+	0+	0+	18	Reply-Message	
0-1	0-1	0	0	19	Callback-Number	
0	0-1	0	0	20	Callback-Id	
0	0+	0	0	22	Framed-Route	
0	0-1	0	0	23	Framed-IPX-Network	
0-1	0-1	0	0-1	24	State	An Access-Request must contain either a `User- Password` or a `CHAP-Password` or `State`. An Access-Request must *not* contain both a `User- Password` and a `CHAP-Password`. If future extensions allow other kinds of authentication information to be conveyed, the attribute for that can be used in an Access-Request instead of `User-Password` or `CHAP-Password`.
0	0+	0	0	25	Class	
0+	0+	0	0+	26	Vendor-Specific	
0	0-1	0	0-1	27	Session-Timeout	
0	0-1	0	0-1	28	Idle-Timeout	
0	0-1	0	0	29	Termination- Action	
0-1	0	0	0	30	Called-Station-Id	
0-1	0	0	0	31	Calling-Station-Id	
0-1	0	0	0	32	NAS-Identifier	
0+	0+	0+	0+	33	Proxy-State	
0-1	0-1	0	0	34	Login-LAT-Service	
0-1	0-1	0	0	35	Login-LAT-Node	
0-1	0-1	0	0	36	Login-LAT-Group	
0	0-1	0	0	37	Framed-AppleTalk-Link	
0	0+	0	0	38	Framed-AppleTalk-Network	
0	0-1	0	0	39	Framed-AppleTalk-Zone	
0-1	0	0	0	60	CHAP-Challenge	
0-1	0	0	0	61	NAS-Port-Type	
0-1	0-1	0	0	62	Port-Limit	
0-1	0-1	0	0	63	Login-LAT-Port	

0-1	0	0	0	77	Connect-Info	
0+	0+	0+	0+	79	EAP-Message	
0-1	0-1	0-1	0-1	80	Message-Authenticator	
0	0-1	0	0	85	Acct-Interim-Interval	

RADIUS Accounting Attributes

This table shows the attributes found in Accounting-Request packets. No attributes should be found in Accounting-Response packets except Proxy-State and possibly Vendor-Specific. On the table:

- **0:** This attribute must not be present in packet.
- **0+:** Zero or more instances of this attribute may be present in packet.
- **0-1**: Zero or one instance of this attribute may be present in packet.
- **1**: Exactly one instance of this attribute must be present in packet.

#	Attribute	Notes
0-1	User-Name	
0	User-Password	
0	CHAP-Password	
0-1	NAS-IP-Address	An Accounting-Request must contain either a NAS-IP-Address or a NAS- Identifier (or both).
0-1	NAS-Port	
0-1	Service-Type	
0-1	Framed-Protocol	
0-1	Framed-IP-Address	
0-1	Framed-IP-Netmask	
0-1	Framed-Routing	
0+	Filter-Id	
0-1	Framed-MTU	
0+	Framed-Compression	
0+	Login-IP-Host	
0-1	Login-Service	
0-1	Login-TCP-Port	
0	Reply-Message	
0-1	Callback-Number	
0-1	Callback-Id	
0+	Framed-Route	
0-1	Framed-IPX-Network	
0	State	
0+	Class	
0+	Vendor-Specific	
0-1	Session-Timeout	
0-1	Idle-Timeout	
0-1	Termination-Action	
0-1	Called-Station-Id	
0-1	Calling-Station-Id	
0-1	NAS-Identifier	An Accounting-Request must contain either a NAS-IP-Address or a NAS- Identifier (or both).

0+	Proxy-State	
0-1	Login-LAT-Service	
0-1	Login-LAT-Node	
0-1	Login-LAT-Group	
0-1	Framed-AppleTalk-Link	
0-1	Framed-AppleTalk-Network	
0-1	Framed-AppleTalk-Zone	
1	Acct-Status-Type	
0-1	Acct-Delay-Time	
0-1	Acct-Input-Octets	
0-1	Acct-Output-Octets	
1	Acct-Session-Id	
0-1	Acct-Authentic	
0-1	Acct-Session-Time	
0-1	Acct-Input-Packets	
0-1	Acct-Output-Packets	
0-1	Acct-Terminate-Cause	
0+	Acct-Multi-Session-Id	
0+	Acct-Link-Count	
0	CHAP-Challenge	
0-1	NAS-Port-Type	
0-1	Port-Limit	
0-1	Login-LAT-Port	
0 -1	Acct-Input-Gigawords	
0 -1	Acct-Output-	
0 -1	Event-Timestamp	
0+	Connect-Info	

Trademarks

iPass, iPassConnect, ExpressConnect, iPassNet, RoamServer, NetServer, iPass Mobile Office, DeviceID, EPM, iSEEL, iPass Alliance, Open Mobile, and the iPass logo are trademarks of iPass Inc.

All other brand or product names are trademarks or registered trademarks of their respective companies.

Warranty

No part of this document may be reproduced, disclosed, electronically distributed, or used without the prior consent of the copyright holder.

Use of the software and documentation is governed by the terms and conditions of the iPass Corporate Remote Access Agreement, or Channel Partner Reseller Agreement.

Information in this document is subject to change without notice.

Every effort has been made to use fictional companies and locations in this document. Any actual company names or locations are strictly coincidental and do not constitute endorsement.