RLM Activation Pro Manual

RLM v9.4

July, 2012

RLM Activation Pro Manual

v9.4

July, 2012

Contents

RLM Activation Pro Manual

RLM Activation Pro Manual v9.4, July, 2012 ***(v9.4BL2)***

Welcome

This manual, the *RLM Activation Pro Manual*, is intended for organizations that have licensed the optional RLM Activation Pro software or the Hosted Activation Service from Reprise Software. This manual explains how to configure and run the activation software.

The RLM documentation is divided into 5 manuals:

Standard RLM Components

- *RLM Getting Started Guide* - an introduction to the basic concepts of license management and RLM
- *RLM Reference Manual* - the complete reference to all core RLM components (including the end-user manual)
- *RLM End-User Manual* - The stand-alone end-user manual, suitable for shipment to your customers

Optional RLM Components

- *RLM Activation Pro Manual* **(this manual)** - Reference for the Optional RLM Activation Pro software

The *RLM Getting Started Guide, RLM Activation Pro Getting Started Guide, RLM Reference Manual* and **RLM Activation Pro Manual** (**this manual**) are available as PDF downloads at the Reprise Website at:

http://www.reprisesoftware.com/kits/RLM_Getting_Started_Guide.pdf

http://www.reprisesoftware.com/kits/RLM_Activation_Pro_Getting_Started_Guide.pdf

http://www.reprisesoftware.com/kits/RLM_Reference_Manual.pdf

http://www.reprisesoftware.com/kits/RLM_Activation_Pro.pdf

The enduser manual is in wiki format. This manual is also available at the Reprise Website:

http://www.reprisesoftware.com/kits/RLM_Enduser.html

What's New in RLM Activation PRO v9.4

This section lists the new features in RLM Activation Pro v9.4 along with pointers to the relevant sections in the manual.

What's new

- ***The Administer Database form (available to admin users only) now allows you to back up the database.*** Selecting this option will create a file of SQL commands to restore your database.
- ***The Activation Pro kit no longer requires anything to be compiled as part of the installation process.*** The only reason you would have to compile anything now is if you need to build a license generator to support an ISV-defined hostid.
- ***The activation database now contains customer/contact info*** which can be associated with activation keys.   See Customer List on page 27 for more information.
- ***When creating activation keys, the keys created are now displayed on the screen.***
- ***RLC login passwords can now contain uppercase characters.*** Prior to Activation Pro v9.4, passwords could consist only of numbers and lowercase letters. Uppercase letters are now allowed.
- ***Product Definitions can now be deleted.*** If a product definition has no associated activation keys, a delete icon will appear in the list of product definitions; clicking this will delete the product definition.
- ***Activation Keys can now be deleted.*** If an activation key has no associated fulfillments, a delete icon will appear in the list of activation keys; clicking this will delete the activation key.
- ***The "View Product Definitions", "View Activation Keys", and "View Fulfillments" buttons now allow you to filter the product definitions which appear***. See the documentation on the particular command for more information.
- ***The activation PRO license generator now collects usage statistics.*** These can be displayed in the new "Statistics" report under "Reports".
- ***The activation PRO license generator now monitors bad key requests by IP address and will automatically blacklist an IP address if too many incorrect activation keys have been presented.*** See Appendix K – Automatically blacklisting IP addresses on page 66 for more information.
- ***The "login" subdirectory has been removed from the kit. The login functions are now in the main actpro directory.***
- ***The underlying actpro code has been updated to use the mysqli API rather than the older mysql API.***
- ***The ActPro installation now auto-detects the location of the cgi-bin directory, rather than having it hard-coded to ../../cgi-bin.***
- ***The "administer database" menu now has a "Check Database" button to check the database consistency.*** Several internal tables will be checked for missing product definitions or activation keys when this command is executed.

Section 1 – RLM Activation Pro Basics

This section of the manual contains the information you need to set up and deploy the RLM Activation Pro software.

Activation Overview

Overview

RLM Activation Pro allows you to deliver a pre-generated **activation key** to your customer, and when they are ready to use your product, a transaction with the activation server running at your site allows the license to be fulfilled without manual intervention. When using activation, there is no need for you to get your user's hostid information - this is transmitted to the activation server automatically.

In the case of a node-locked product, a typical scenario would be that your customer runs the product on the desired machine, and if the license had not been fulfilled earlier, the product asks for an activation key. Once the activation key is supplied, the license is retrieved transparently. From this point on, the product runs with its license in place.

Floating licenses would operate in a similar manner, except that the number of floating licenses to be activated is required.

RLM Activation Pro utilizes a *MySQL* database for storing all activation data for quick access to large amounts of activation data. The rlc back office GUI is written in PHP and runs against the same databases as the activation license generator.

The RLM kit contains a demo of the activation capability. For more information on this demo and how to run it, see ….... This demo applies to either standard Internet Activation or Activation Pro.

How Activation Works

RLM Activation is driven by two main data items - **product definitions** and **activation keys**. The RLM License Center tool (rlc) allows you to define various products to be activated.

Product definitions consist of a product definition name, a product name, version, expiration specification, and node-locked/floating indication. In addition, product definitions can contain additional attributes for the license. All product definition attributes are described in detail in the Database Setup chapter on page 17.

Once a product is defined, **activation keys** can be created. An **activation key** specifies the product to be fulfilled, the type of fulfillment, and how many fulfillments are allowed by this activation key. In addition, **activation keys** allow you to specify the license expiration and/or additional attributes for the license. If any of the attributes specified in the **activation key** are the same as attributes specified in the **product definition**, the **activation key** version of these attributes will be used. All activation key attributes are described in detail in the Database Setup chapter on page 17.

Once activation is set up, you give your customer an **activation key**, which they then use to retrieve the license for the particular machine which they are using. This happens via the internet, using the license generator (isvname_mklic) on your website.

The request is made by your customer via either calls you make to the **activation request API** or the Reprise-supplied **activation GUI** built into the rlm webserver.

If you use the **activation request API** (rlm_act_request()), you will get another opportunity to add additional attributes to the generated license. These attributes will override any specified in either the **activation key** or the **product definition**.

Note that the system date/time on the requesting computer must be within 7 days of the system clock on the activation server. If the clocks differ by more than 7 days, an RLM_ACT_CLIENT_TIME_BAD error status will be returned, and no license will be generated.

Activation Pro Components

RLM Activation Pro consists of 5 components:

- activation admin tool (rlc)
- activation server license generator (actpro_mklic + isvname.gen, or isvname_mklic)
- activation data
- activation request API (rlm_activate())
- optional activation GUI (built into the rlm webserver)

The activation admin tool, **RLM License Center** (rlc), allows you to set up the parameters for activation: **product definitions**, **activation keys**, **list of blacklisted hosts**, and other **setup data**, as well as view records of license fulfillments.

The activation server **license generator** (actpro_mklic + isvname.gen, or isvname_mklic) runs on the website (in the cgi-bin directory) and processes activation requests, recording the results in the activation database.

The **activation data** consists of product and activation key definitions, blacklisted host definitions, rlc setup data, as well as records of license fulfillments. The data is described in the **Activation Data** section below, and is all stored in a MySQL database.

The **activation request API** (rlm_activate() or the older rlm_act_request()) allows you to embed activation requests within your application or your installation tools.

The **optional activation GUI** is built into the RLM webserver (using the activtion request API) to provide a generic activation user interface.

You will always be able to provide a simpler activation user interface to your customers by using the activation request API, since you know the answers to many of the questions which the Reprise-supplied activation GUI must ask. However, you can use the supplied activation GUI to get started without modifying your applications.

Rehostable Licenses

Activation can be set up to support rehosting of licenses by the end user without ISV involvement. This involves the use of the rehostable hostid type, which is new in RLM v9.3. A rehostable hostid is requested at the time that rlm_activate() is called, and the activation server issues the license to this hostid. Later, if the user wishes to move this license to another system, the rlm_revoke() call can be used to revoke the rehostable license and allow it to be activated on another system. Note that the rehostable hostid is a copy-protected file, and if the system should crash, the hostid cannot be retrieved. As an ISV, you will need a policy to deal with this situation – e.g., to give the customer another activation in order to continue. Since this situation should be rare, a non-automated request procedure might be quite sufficient. If you use the **Normal** fulfillment type with Rehostable licenses, the expiration date should be fixed. If you want an expiration date that is relative to the time of activation, you should use the **ReActivate** fulfillment type, and set the count to the # of times you want to allow your customer to revoke and re-activate the license.

Why should **Normal** fulfillments use fixed license expirations? Well, when your customer revokes a rehostable license, the activation server returns the fulfillment count back to the activation key. If, for example, you issue 60-day licenses with **Normal** fulfillment type, your customer could fulfill the license, then on day 59 revoke that license and re-activate it and get a new 60-day license. If the activation key specifies a fixed expiration date, that is not an issue. Also, if you use a **ReActivate** fulfillment type, the original expiration date will be preserved on a new activation.

What happens if my customer doesn't have an internet connection?

For cases such as this, you can create a webpage so that your customer support people can activate the license for your customer, then email it to them. The RLM kit has an example html file that can be customized for this use. It is called ***activation_example.html*** and it is in the ***examples*** directory of the RLM kit. Modify this file for your use, and put it on an internal site for your support people to use.

Getting Started

First, you should read the Activation Overview section of this manual on page 7. Next, you should read the three setup chapters:

- Activation Pro Setup on page 12 (Note: if you are using Reprise Hosted Activation, you can skip this chapter – we have already set up Activation Pro for you).
- Database Setup on page 17 and
- Client Setup on page 30.

When you have performed these steps, you are ready to issue licenses for your product with RLM Activation Pro.

To use the RLM Activation Pro administration tool (**rlc**), point your browser (RLM Activation Pro supports Firefox v3.6 or later) to the URL where your system administrator set up the RLM Activation Pro software. The webserver will prompt you for your username and password. Once you provide these, you will see the initial **rlc** screen. Login to **rlc** using the username and password you received from your system administrator (the login area is in the bottom left-hand side of the screen).

Upgrading from an earlier version of Activation Pro

If you are currently running Activation Pro, you can easily upgrade to the new version. (Hosted Activation customers do not need to worry about software upgrades – we do these for you).

You should follow these steps:

1. Back up your MySQL database. This can be done with the phpMyAdmin program available as part of XAMPP and on most web hosting companies' admin pages. Select your database, click on the "export" tab, and press "GO" in the bottom-right hand corner. Save the file. While this step isn't strictly required, it is ***strongly recommended*** by Reprise Software.
2. Install the new Activation Pro software. Do this the same way you installed Activation Pro initially:
   - extract the kit.
   - Copy your isv generator settings file to the main directory (e.g. actpro.x86_l2) **and** the *actpro_setup* directory on the kit (see the next chapter for details).
   - Remove the actpro_setup directory on your webserver.
   - If your web server is local, then edit the HTDOCS variable and run "make". If your web server runs at an ISP, ftp the "actpro_setup" directory and all it's contents to your ISP, as you did in the initial installation.
   - Browse to …../actpro_setup/setup.php on the webserver, just like when you first set up Activation Pro. The setup program will detect that you have an installation present, and will prompt you to upgrade. If you select "yes", it will then back up the program directory (actpro), install the new software into a new actpro directory, and upgrade the database from the old version to the new version.

Activation Pro Setup

In order to set up activation, you do 3 things:

1. install the Activation Pro software on your webserver (If you are a Reprise Hosted Activation customer, you can skip this chapter and move straight to step 2).
2. setup the Activation Database (described in the Database Setup chapter on page 17)
3. Add activation request calls to your application (described in the Client Setup chapter on page 30)

Requirements

Setting up *RLM Activation Pro* involves setting up a webserver with PHP and MySQL support, as well as the *RLM Activation Pro* software. Ideally, you should be familiar with downloading and installing open-source tools and using Web infrastructure.

RLM Activation Pro requires:

- The Apache Web server, version 2.2 or greater
- PHP version 5, with MySQL extensions
- MySQL server, version 5.1 or greater
- MySQL Connector/C development kit (you will only need this for linking your license generator and you only need to do this if you are using ISV-defined hostids)
- Mozilla Firefox v3.6 (for running *rlc*, the administration tool)

To get you started as quickly and easily as possible with RLM Activation Pro we suggest that you install a pre-configured Apache/MySQL/PHP (AMP) stack. Reprise Software recommends the XAMPP stack from Apache Friends. This is the stack which we use to test Activation Pro. It is free, and it comes pre-built for all platforms on which we support RLM Activation Pro (although it is not appropriate for production, see below). While it is certainly possible for you to install Activation Pro on an existing Apache/MySQL/PHP stack, please understand that Reprise cannot support every combination of these components. We are familiar with the XAMPP stack, and will be able to offer help on XAMPP. We strongly urge you to install Activation Pro on an XAMPP stack to become familiar with the product, then, once you are familiar with RLM Activation Pro, move your installation to you ultimate web server or web hosting company. This way, you will have a working reference implementation. Note that XAMPP is not recommended for production installations, it is not secure. See here: http://www.apachefriends.org/en/xampp.html#300

If you are not comfortable setting up your own webserver, you might consider Reprise Software's hosted activation service, which will save you the trouble of installing even XAMPP.

Setting Up Activation Pro

To set up RLM Activation Pro, there are 6 steps:

1. Install a pre-configured Apache/MySQL/PHP (AMP) stack (described in this chapter).
2. Install the Activation Pro software on your webserver (described in this chapter).
3. Configure RLM Activation Pro (described in this chapter).
4. (Optional) Configure your License Generator if you use an ISV-defined hostid.
5. Setup the Activation Database (described in the Database Setup chapter on page 17).
6. Add activation request calls to your application, or build a stand-alone activation utility (described in the Client Setup chapter on page 30)

If you are using Activation Pro at a web hosting company, a different set of installation steps will be required. In particular, step 1 above isn't required. The differences will be outlined in the following sections.

If you are using Reprise Hosted Activation, we have done steps 1-3 for you, so unless you have an ISV-defined hostid (step 4), you can skip the remainder of this chapter and go straight to Database Setup on page 17.

Step 1. Install a pre-configured Apache/MySQL/PHP (AMP) stack

If you are installing *RLM Activation Pro* at your ISP, you can skip this section. If you are evaluating, however, Reprise Software recommends installing on a local AMP stack before you attempt to run *RLM Activation Pro* at your ISP.

To start, you need the stack of web services to support *RLM Activation Pro*. A pre-configured AMP stack called XAMPP is available for Windows and Linux systems at the Apache Friends website. To install it, do the following:

- For Windows:
  - Browse to: http://www.apachefriends.org/en/xampp-windows.html
  - Below the text "Jump-off point", select the XAMPP download.
  - select the "Installer" link. Save the file, then run it.
- For Linux:
  - Browse to http://www.apachefriends.org/en/xampp-linux.html
  - Follow the 4 steps on the download page
  - after installing, the package will be in /opt/lampp, execute the following commands (as root):
    - chmod 777 /opt/lampp/cgi-bin
    - chmod 777 /opt/lampp/htdocs

    (Note: these directories would not normally have 777 permissions, but we are doing this to make the rest of the setup easier. When you are finished, reset the permissions on cgi-bin and htdocs to their original settings).
    - Now you can log out as root.

Step 2. Install the *RLM Activation Pro* kit from the Reprise website

RLM Activation Pro operates on a particular RLM binary platform, and is supported on windows, and linux systems (x86_w and x86_l):

On all Platforms

First, download the Activation Pro kit: go to the Reprise Website Download area, enter your username and password, and select the "RLM Activation Pro" kit. Pick the appropriate platform, and save this on your system.

Note: When downloading Unix kits using Internet Explorer on Windows XP systems, the files are incorrectly named as 'actpro.platform.tar.tar', rather than 'actpro.platform.tar.gz', once downloaded. This is a browser issue - after transfer, please rename the file before installation. Or, better yet, use Firefox.

Also note: if you are installing on your ISP's webserver, be sure to chose the *RLM Activation Pro* kit appropriate for your ISP's operating system.

Continuing the Installation on Linux

Extract the kit files:

Use gunzip/tar to extract the archive:

```
 % gunzip actpro.platform.tar.gz
 % tar xvf actpro.platform.tar
```

Continuing the Installation on Windows

Extract the kit files:

On Windows, the kit is in a ZIP file. Unzip the file in the desired location.

Step 3. Configure *RLM Activation Pro*

- **Copy your License Generator Settings file from your RLM kit.**

  You can create the license generator settings file (*isvname*.gen) on any supported RLM platform. To create it, execute the following command in the configured RLM kit area:

  ```
  rlmsign -generator
  ```

  Copy the resulting file(*isvname*.gen) to the main actpro directory (e.g. actpro.x86_l2) **and** the *actpro_setup* directory. If you are copying between Unix and Windows systems, be sure that your copy program doesn't insert carriage-control characters. The easiest way to make sure of this is to check the file size to make sure it is the same after copying.

- **Install the *RLM Activation Pro* files on your webserver**

  If you are running on a local AMP stack, follow these instructions:

  - Run the command “make” (or “nmake” for Windows) in this directory.
    - The “make” command will copy all the Activation Pro files to your webserver space as specified by the $(HTDOCS) variable in the makefile. The files will reside in $(HTDOCS)/actpro_setup ready for the next step.

  If you are running at your ISP, follow these instructions:

  - run the command “make setup” (or “nmake setup” for Windows)
  - FTP the contents of the actpro_setup directory to your ISP's webserver space, e.g. *//www.yourdomain.com/actpro_setup*.
  - Make sure that the *actpro_setup* directory is mode rwx to the web user.
  - Make sure that *actpro_setup/rlmact.mysql* is writable by the web user.

- **Run the Activation Pro setup program.**
  - On Windows: Run the XAMPP control panel. Start the Apache and MySQL services. On Linux, this happened as part of installation.
  - Point your browser (Firefox recommended) to:

    http://localhost/actpro_setup/setup.php

  - The defaults in the form should be correct, so press "Setup activation database". ***REMEMBER the admin account username and password – you will need this later***. (The default is admin and admin).

NOTE: if you are going to host your servers at your ISP, many ISPs have naming conventions for MySQL databases, and, in addition, many ISPs will not allow you to create a new database from a script, the way that *Activation Pro's* setup.php software does it. If this is the case, simply create the database manually using your ISP's administration tools, then run setup.php after the database is created.

Security Considerations

Once your Activation Pro software is installed, you will want to check the permissions on all files and perform other security checks on your website. While this list is not complete, it is a good start:

1. Remove the directory *actpro_setup* on your webserver. You no longer need this.
2. Change the modes of the following files, as shown (note: the setup program attempts to do this, but on some servers the file modes are not set correctly):

   all .php files – mode 0400

   rlmact.mysql – mode 400

   all other files (including all files in the images directory) – mode 0444
3. Make sure that your cgi-bin/isvname_mklic program (or isvname.gen file) is read-execute to owner, ie, mode 500.
4. Consider setting up a .htaccess file to protect access to the *actpro* directory. Many ISPs have a tool to password-protect the directory.
5. Reset permissions of the cgi-bin and htdocs directories to their original settings.

Step 4 (OPTIONAL) - Configuring your license generator if you use an ISV-defined hostid

If your software uses an RLM ISV-defined hostid, you will need to build a custom license generator for *Activation Pro*. In order to do this, you need the MySQL Connector/C development environment.

To install the Connector/C development environment:

```
1. Browse to: http://dev.mysql.com/downloads/connector/c/

2. Select the appropriate platform from the drop-down list

3. Select the most appropriate distribution for your operating system, and
download that version.  Follow the directions for installation. Note that
you will have to register to download the MySQL software).  Select a mirror
site and download the binaries.  Save the file.

4. Install the connector kit in a convenient location.
```

For example, on windows, you might download the Windows (x86, 32-bit), MSI Installer. Select the mirror site, then download. Once downloaded, start the installer by double-clicking on it's icon. Click past the security warning, accept the license agreement, then click “next”, and select a “custom” installation, unless you want to put the kit in the standard location. Click “Next”, then “Install”.

On Linux, you might pick the “Linux-Generic” platform, and the Linux ver. .3.glibc2 (x86, 32-bit), Compressed TAR Archive. Select the mirror site, then download, save the file. Once downloaded, gunzip the installer, and change directory to where you want to install the MySQL software. Extract the installer with tar. Click past the security warning, accept the license agreement, then click “next”, and select a “custom” installation (Unless you want to use the default location). Click “OK”. Click “Next”, then “Install".

Once the Connector/C development kit is installed, do the following:

```
1. edit the makefile in the Activation Pro kit and change the definition of
MYSQLDEV to be the directory where you installed the Connector/C kit.

2, edit the makefile and change the definition of ISV_DEF_HOSTID_OBJS to
point to your object files that implement your ISV-defined hostid.

3.Copy the RLM library (rlm.a or rlmclient.lib on windows) from a production
RLM kit of the correct platform.

4. create the custom generator by typing "make isvname_mklic" where isvname
is your ISV name.

5. Copy the isvname_mklic binary to your webserver cgi-bin directory,
replacing the generic generator which was installed earlier when the kit was
installed, or, if you have not installed the files yet, copy isvname_mklic
to the actpro_setup directory before ftping the actpro_setup directory to
your webserver.  (Note: Additionally, the rlc admin account has an
Administer Database menu item which contains an option to upload the license
generator from your local system to the webserver cgi-bin directory.)
```

Using rlc with https

rlc can be configured to use the https:// protocol. To do this, copy the file ***htaccess.https*** in the actpro directory on your webserver to ***.htaccess*** in the same directory. You may get a certificate warning message the first time you connect to the site, which you can ignore. If you want to fix the certificate warning, you will need to purchase an SSL certificate for your website.

Note that the license generator itself, ISV_mklic, will not work with https. The traffic from the rlm_act_request() call to the license generator is already encrypted (as of RLM v9.0), so activation keys are not sent in the clear.

Database Setup – using rlc

In order to set up activation, you do 3 things:

1. install the Activation Pro software on your webserver (described in the Activation Pro Setup chapter on page 12). (Not necessary for Reprise Hosted Activation customers).
2. setup the Activation Database (described in this chapter)
3. Add activation request calls to your application (described in the Client Setup chapter on page 30)

All RLM activation activities are controlled by a database of **product definitions** and **activation key definitions**. In addition, you can create a **blacklist** of domains which are not allowed to activate licenses.

Before you can begin to use RLM activation, you must set up at least one product definition and one activation key definition, as described below. You should also set up your activation URL for REFRESH licenses if you are planning on using REFRESH-type activations (deprecated in RLM v9.3).

These operations are performed by **rlc** - RLM License Center. **rlc** is a web-based application which allows you to maintain your activation database.

rlc uses user access permissions to control access to the three main functions:

- viewing and reporting on the activation database (**View** Access)
- editing the activation database (**Edit** Access)
- administering users/permissions/database (**Admin** Access)

Running rlc

You run rlc by pointing your browser to the URL where you installed the Activation Pro software. The server uses a MySQL database to store all the activation data.

rlc Access Control

Access control is performed by logging in as the **admin** user and creating individual user accounts. An account has one of 3 levels of access:

- View
- Edit
- Admin

An account with **View** access can view the activation data, but cannot create or edit any data.

An account with **Edit** access can do anything an account with **View** access can do, as well as edit any data except user accounts.

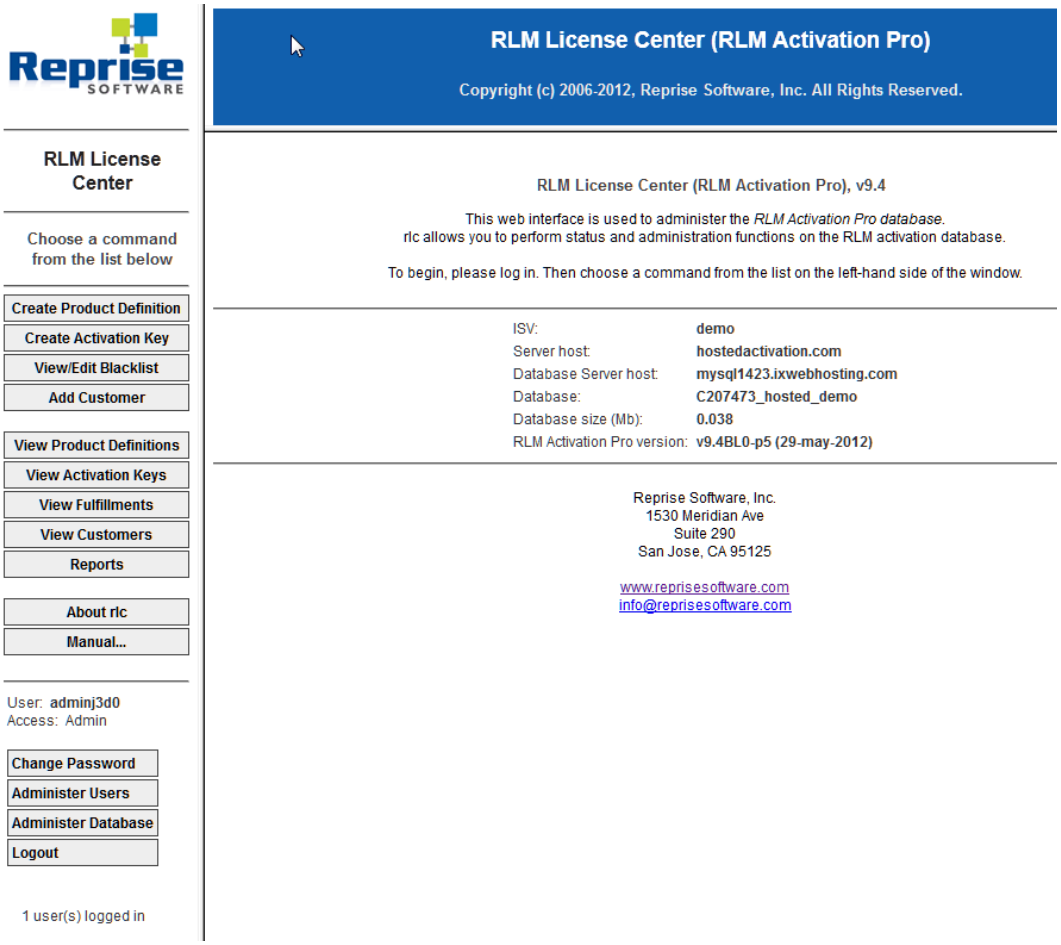
An account with **Admin** access can do anything an account with **Edit** access can do, as well as create and delete users, upload the activation license generator settings and perform other administration functions.

Using rlc

There are 3 main activities you will perform with **rlc**:

- initial setup of your activation database
- viewing and/or updating the activation data
- generating individual licenses

When you run **rlc**, your browser will display a page that has a title area at the top, along with a column of command buttons along the left, and a main display area on the bottom right hand side, as shown below:



RLM License Center (RLM Activation Pro)

Copyright (c) 2006-2012, Reprise Software, Inc. All Rights Reserved.

RLM License Center (RLM Activation Pro), v9.4

This web interface is used to administer the *RLM Activation Pro database.*
rlc allows you to perform status and administration functions on the RLM activation database.

To begin, please log in. Then choose a command from the list on the left-hand side of the window.

ISV:	demo
Server host:	hostedactivation.com
Database Server host:	mysql1423.ixwebhosting.com
Database:	C207473_hosted_demo
Database size (Mb):	0.038
RLM Activation Pro version:	v9.4BL0-p5 (29-may-2012)

Reprise Software, Inc.
1530 Meridian Ave
Suite 290
San Jose, CA 95125

www.reprisesoftware.com
info@reprisesoftware.com

User: adminj3d0
Access: Admin

1 user(s) logged in

The command buttons are arranged in 3 groups, top to bottom:

- Setup commands - "Create Product Definition", "Create Activation Key", “View/Edit Blacklist”, and “Add Customer”.
- Database viewing/updating commands - "View Product Definitions", "View Activation Keys", "View Fulfillments", “View Customers”, “Reports”, "About rlc", and “Manual...”
- A Login area, with some account modification and administration commands.

Logging In

At the bottom of the menu username and password boxes are displayed, along with a **Login** button. Enter the username and password assigned by your administrator (when Activation Pro is installed, an admin user is created with password "admin".)

Once logged in, this area changes to a set of administration commands appropriate for the user.

The commands are:

- **Change Password**,
- **Administer Users**,
- **Administer Database**, and
- **Logout**

Change Password allows you to change your password. (Note: passwords can consist of numbers, lowercase letters and uppercase letters only).

Administer Users [administrator users only] allows the administrator to create and delete users, as well as modify the access permission of any user.

Administer Database [administrator users only] allows the administrator to upload the license generator settings, and perform backup operations on the database. In addition, you can configure the automatic blacklisting of domains and check database consistency with commands in this screen.

Logout logs you out.

You will want to have non-administrator users for the day-to-day operations of managing your activation server. To create these user(s) log in to the activation server as the *admin* user, then create a second user who can make changes to the activation database. Perform the following steps:

1. Point your browser to your installed version of Activation Pro, for example: http://localhost/actpro/index.php
2. On the left-hand side of the screen, near the bottom, type "admin" in the Username box, and "admin" in the password box. Press "login". (This assumes you did not change the default admin user's password).
3. Select "Administer Users" from the lower-left menu.
4. Press "Create New User" in the view area, below the list of users.
5. Select a username, password, and "Edit" access in the "Create New User" page, then press the "Create User" button. Go back to the admin page by clicking on the "Admin Page" link. You should now see your new user, with Edit access.
6. Log out (at the bottom left).

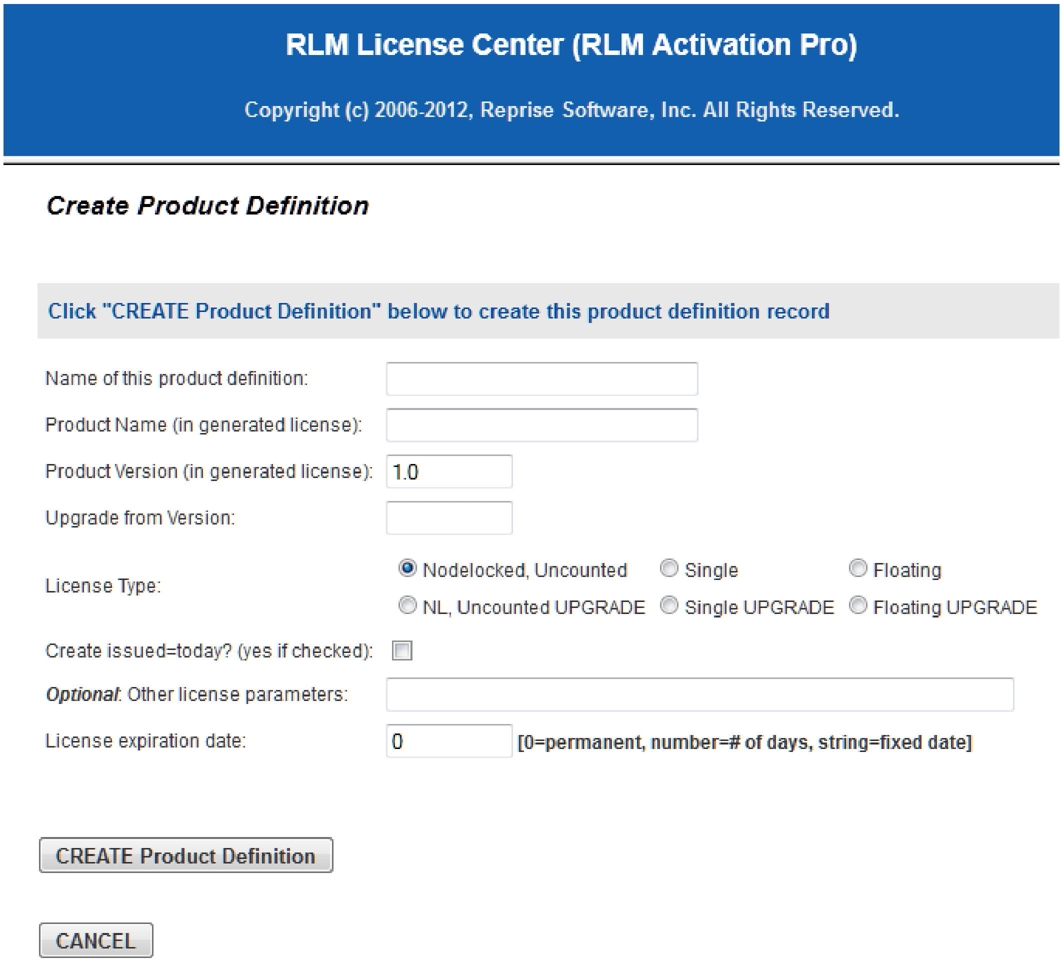
Setup Commands

The setup commands are:

- **Create Product Definition**
- **Create Activation Key**
- **View/Edit Blacklist**
- **Add Customer**

Product Definitions

These commands are used to set up the database for activation and/or license generation. Everything in rlc is driven from *product definitions*, so you should create at least one product definition first. Product definition allows you to specify a name for the "product" which is then associated with a license product name, version, and several other license attributes. When you press the **Create Product Definition** button, you will see the screen below:



RLM License Center (RLM Activation Pro)

Copyright (c) 2006-2012, Reprise Software, Inc. All Rights Reserved.

Create Product Definition

Click "CREATE Product Definition" below to create this product definition record

Name of this product definition:	
Product Name (in generated license):	
Product Version (in generated license):	1.0
Upgrade from Version:	
License Type:	● Nodelocked, Uncounted ○ Single ○ Floating ○ NL, Uncounted UPGRADE ○ Single UPGRADE ○ Floating UPGRADE
Create issued=today? (yes if checked):	□
Optional: Other license parameters:	
License expiration date:	0 [0=permanent, number=# of days, string=fixed date]

A product definitions consist of:

- a product definition name
- a product (license) name, i.e. what you check out in your application.
- a product version
- an optional upgrade-from version
- a license type
- an issued= flag
- optional additional attributes for the license
- a product expiration specification

The **product definition name** is the name used to refer to this product definition. It is not the name of the product license generated by RLM activation. This might be something like "Evaluation Office License", in which case the actual product name might be "Office".

The **product name** is the name of the RLM license to be generated. This is the name that appears on the LICENSE line in your RLM license file.

The **product version** is the version of the product on the generated LICENSE line.

The **upgrade from version** is the minimum version of the old product which is eligible for upgrades in an RLM UPGRADE license.

The **license type** controls the type of generated license. Choices are:

- node-locked, uncounted
- single
- floating
- upgrade (floating licenses only)

In the case of floating licenses, the requested count is the number of floating licenses generated. The requested count must be less than or equal to the remaining count available in the activation key. (As a special case, a request for 0 licenses will fulfill all remaining licenses from the activation key). For the other license types, the requested count is ignored.

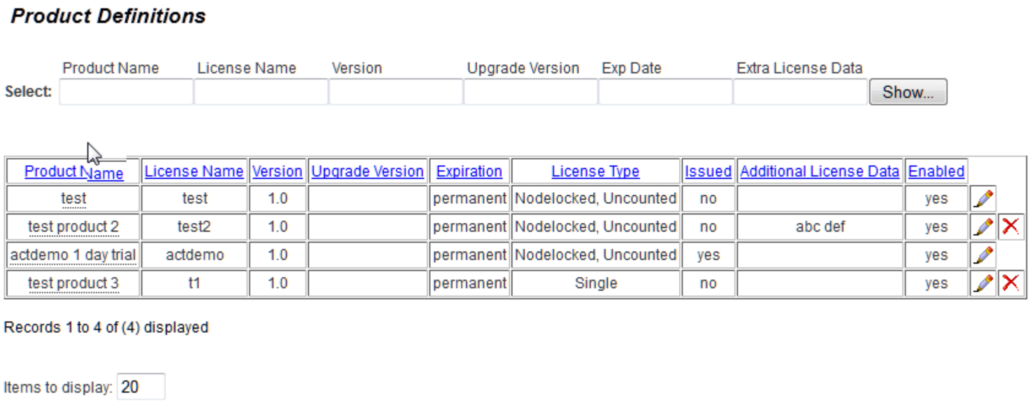
The **Create issued=today?** checkbox controls whether the generated license contains an issued=today date attribute.

The **other license parameters** are additional RLM license attributes which will be placed into the generated LICENSE. These must be valid RLM license keyword/value pairs, e.g. “options=professional”.

The **License expiration date** controls the expiration date for the generated LICENSE line. This is one of:

- 0 (or blank) - for a permanent license
- an integer - the number of days from activation to expiration (ie, 30 means the license expires 30 days after it is activated)
- an actual expiration date, in the form 1-jan-2012 - the actual expiration date, independent of the date of activation, in the standard RLM expiration date format.

Once you have created a product definition, the list will be displayed (this is the same as if you pressed the “View Product Definitions” button on the left-hand menu:



Product Definitions

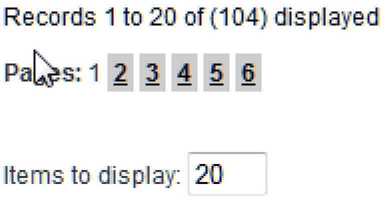
	Product Name	License Name	Version	Upgrade Version	Exp Date	Extra License Data	
Select:							Show...

Product Name	License Name	Version	Upgrade Version	Expiration	License Type	Issued	Additional License Data	Enabled		
test	test	1.0		permanent	Nodelocked, Uncounted	no		yes	✎	
test product 2	test2	1.0		permanent	Nodelocked, Uncounted	no	abc def	yes	✎	✕
actdemo 1 day trial	actdemo	1.0		permanent	Nodelocked, Uncounted	yes		yes	✎	
test product 3	t1	1.0		permanent	Single	no		yes	✎	✕

Records 1 to 4 of (4) displayed

Items to display: 20

This list gives you an overview of the product definitions in your Activation Pro system. If any product definition has no associated activation keys defined, you can delete it by pressing the red "X" in the far right-hand column. In this example, "test product 2" and "test product 3" have no activation keys defined, so they can be deleted. Note that you can control the number of product definitions displayed by changing the "Items to display" count at the bottom of the screen. You can also filter the list by entering text in the "Select:" text boxes at the top of the screen. If the list is longer than the number of items to display, RLC will display a pagination box at the end of the list like the following:



Records 1 to 20 of (104) displayed

Pages: 1 2 3 4 5 6

Items to display: 20

The list filtering and pagination are common to all the browsers: **View Product Definitions**, **View Activation Keys**, **View Fulfillments,** and **View Customers**.

Activation Keys

If you are going to use RLM Activation Pro, you will then need to create activation keys - these are the keys you give to your end-users. When they have the activation key, they can then use Activation Pro to fulfill licenses for their products. Activation keys can be set up to allow unlimited use, or a fixed number of uses.

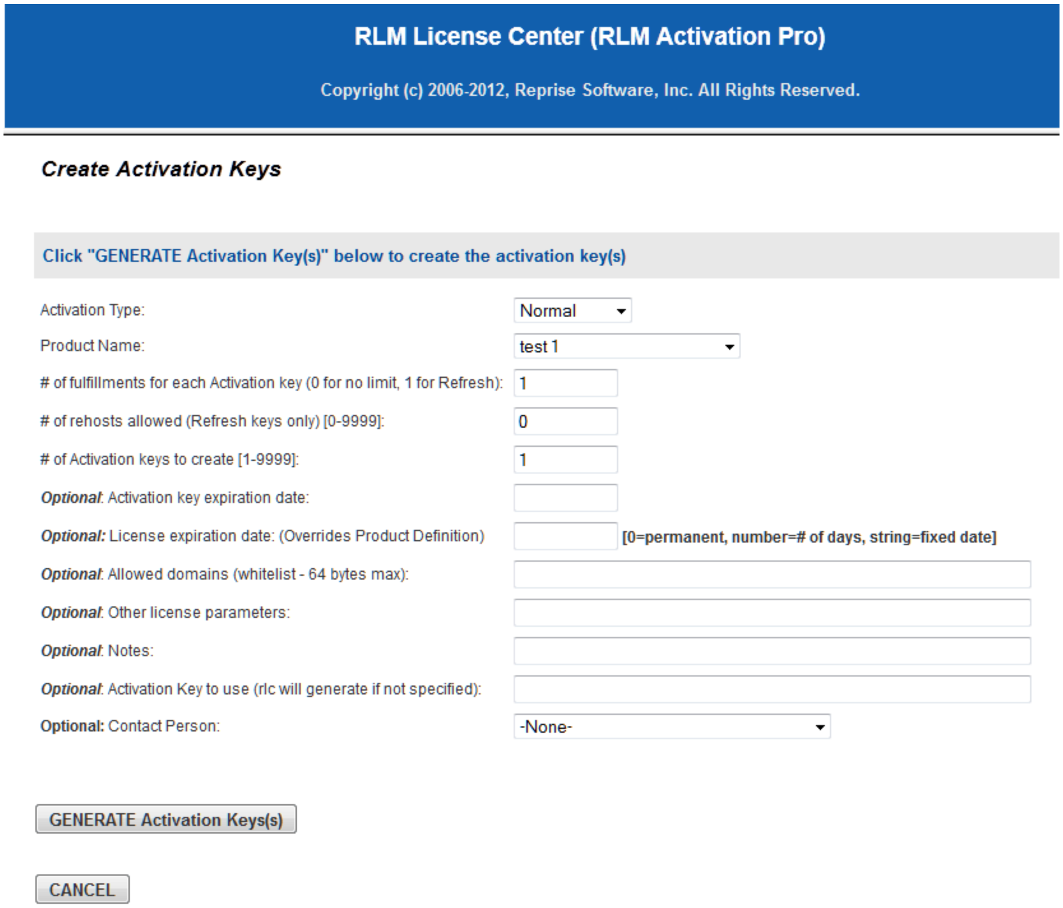
Activation keys consist of:

- an activation type
- a product definition name to be fulfilled
- the fulfillment count
- the hostid change (rehost) count (for Refresh type fulfillments only – deprecated in RLM v9.3)
- the last date valid for this activation key
- the list of allowed domains for this key
- optional additional attributes for the license
- optional notes for the activation key (unused by RLM)

In addition, when you create activation keys, you can also specify:

- the number of independent activation keys to create
- the actual activation key string

When you press the **Create Activation Key** button you will see the screen below:



RLM License Center (RLM Activation Pro)

Copyright (c) 2006-2012, Reprise Software, Inc. All Rights Reserved.

Create Activation Keys

Click "GENERATE Activation Key(s)" below to create the activation key(s)

Activation Type:	Normal
Product Name:	test 1
# of fulfillments for each Activation key (0 for no limit, 1 for Refresh):	1
# of rehosts allowed (Refresh keys only) [0-9999]:	0
# of Activation keys to create [1-9999]:	1
Optional: Activation key expiration date:	
Optional: License expiration date: (Overrides Product Definition)	**[0=permanent, number=# of days, string=fixed date]**
Optional: Allowed domains (whitelist - 64 bytes max):	
Optional: Other license parameters:	
Optional: Notes:	
Optional: Activation Key to use (rlc will generate if not specified):	
Optional: Contact Person:	-None-

The **Activation type** is one of the following:

- Normal
- Reactivate
- Refresh (deprecated in RLM v9.3)
  The meaning of these 3 types is described below.

The **product name** is the name of the product definition to be activated.

The **# of fulfillments** is the number of fulfillments allowed by this activation key. For Refresh keys, this is always 1, and even if you enter a larger number, rlc stores the count as 1. The meaning of count depends on the type of fulfillment, see below.

The **# of rehosts allowed** is used for the Refresh fulfillment type only. This count is the number of times a hostid can be changed while this activation key is valid. After this number of hostid changes, licenses can only be refreshed for the last hostid.

The **# of activation keys to create** controls how many activation keys are created with the data in this form. If the key string is specified (see below), this exact key will be used if the # of activation keys is 1. If the # of activation keys is > 1, then the key string will have a 16-digit string appended for each unique activation key generated.

The **Allowed domains (whitelist)** is a space-separated list of domain names which are allowed to activate using this activation key. If a request comes in from another domain, it is rejected with RLM_ACT_NOT_WHITELISTED. Note that the names in this list can be a substring of the hostname requesting the activation. So, for example, if you put the string "com" into this field, activation will be possible from any domain with "com" in the name (ie, most domains). All names are case-insensitive, and the whole list must be less than RLM_MAX_HOSTNAME (64) bytes. If an activation key specifies allowed domains, the general activation **blacklist** is not used, and the domain requesting the activation must be part of the allowed domains specification.

The **Activation Key Expiration date** for the activation key is the last date when the activation server will generate a license for this activation key. The activation server will continue to send previously-generated licenses if so requested. Note that this is completely different from the license expiration date in the product definition – the last date valid is the last date when the activation server will create a new license for this activation key.

The **License Expiration Date**, if specified, will override the expiration date specified in the product definition. If not specified here, the license expiration from the product definition is used. The date is specified in the same way as in the product definition, i.e., a positive integer indicates a license that expires that number of days from the date of activation, and a fixed date specifies a fixed expiration date.

The **Other license parameters** are additional RLM license attributes which will be placed into the generated LICENSE. Any options specified in the activation key override corresponding attributes in the product definition, thereby allowing you to have default attributes which you modify for a particular customer when you create their activation key. These must be valid RLM license keyword/value pairs, e.g. "options=professional".

The **Notes** field is unused by the activation software, other than in listings and reports.

The **Contact Person** choicelist allows you to associate a customer with this activation key. Create customers with the "**Add Custome**r" button in the Setup Commands area at the top.

The **activation key** generated by **rlc** is a 16-digit number (with embedded dashes for a total of 19 characters) of the form:

```
aaaa-bbbb-cccc-dddd
```

for example:

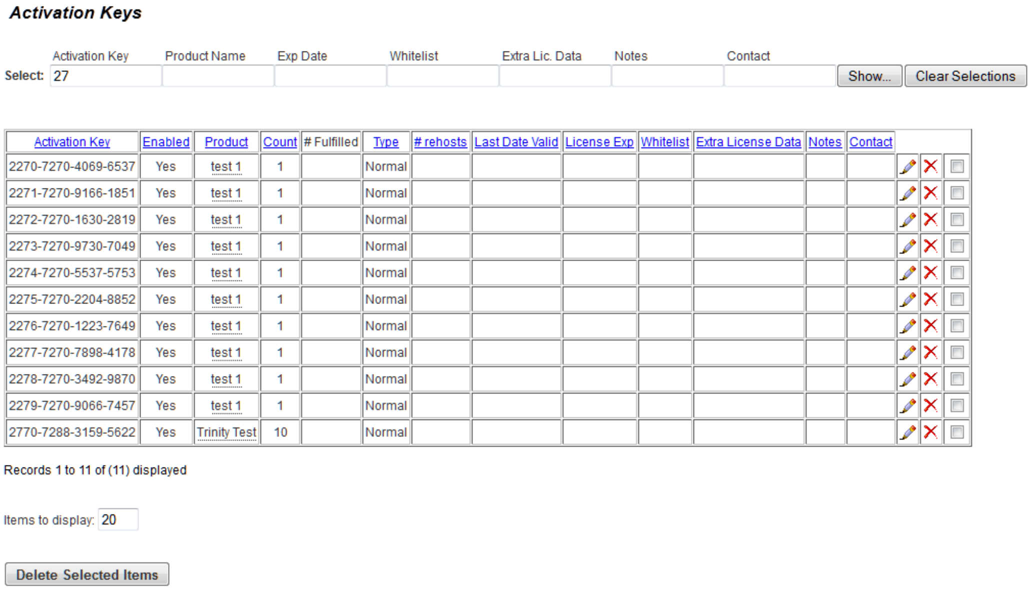
```
4567-4997-7125-5436
```

Alternately, **rlc** will allow you to specify the activation key. If you specify a key and generate a single key, the exact key you specify will be used. If you generate multiple keys, **rlc** will append 16 characters (with 4 dashes) to the string you specified.

This activation key is treated as a string by all RLM activation components.

Once you have created an activation key, the list will be displayed (this is the same as if you pressed the "View Activation Keys" button on the left-hand menu:

This list gives you an overview of the activation keys in your Activation Pro system. If any activation key has no associated fulfillments, you can delete it by pressing the red "X" in the far right-hand column. In this example, "test-second" and "test-third" have no fulfillments, so they can be deleted. Note that this list was filtered to display only activation keys which contain the string "27".



	Activation Key	Product Name	Exp Date	Whitelist	Extra Lic. Data	Notes	Contact
Select:	27						

Activation Key	Enabled	Product	Count	# Fulfilled	Type	# rehosts	Last Date Valid	License Exp	Whitelist	Extra License Data	Notes	Contact			
2270-7270-4069-6537	Yes	test 1	1		Normal								✎	✕	□
2271-7270-9166-1851	Yes	test 1	1		Normal								✎	✕	□
2272-7270-1630-2819	Yes	test 1	1		Normal								✎	✕	□
2273-7270-9730-7049	Yes	test 1	1		Normal								✎	✕	□
2274-7270-5537-5753	Yes	test 1	1		Normal								✎	✕	□
2275-7270-2204-8852	Yes	test 1	1		Normal								✎	✕	□
2276-7270-1223-7649	Yes	test 1	1		Normal								✎	✕	□
2277-7270-7898-4178	Yes	test 1	1		Normal								✎	✕	□
2278-7270-3492-9870	Yes	test 1	1		Normal								✎	✕	□
2279-7270-9066-7457	Yes	test 1	1		Normal								✎	✕	□
2770-7288-3159-5622	Yes	Trinity Test	10		Normal								✎	✕	□

Records 1 to 11 of (11) displayed

Items to display: 20

Fulfillment Type

The **type of fulfillment** controls how the activation server processes requests. In all cases, for a node-locked license, the count field represents the total number of independent licenses which can be generated. For floating licenses, the count field represents the total license count of all floating licenses generated.

The **Normal** fulfillment type is the most straightforward activation type, and represents what most people think of as software activation: each request from a new hostid generates a new license, and this license's expiration date is computed at the time the license is activated. A subsequent request from the same hostid will cause the old generated license to be retrieved, with exactly the same license parameters. If you use the **Normal** fulfillment type for Rehostable licenses, you should always use a fixed expiration date, not a # of days.

For the **Reactivate** fulfillment type, additional activations on the same activation key are considered reactivations. A reactivation will have the same expiration date as the original activation. This can be used to allow your customer to move a time-limited license to a new host one or more times. So, for example, assume you sell a 1-year license, but you would like your customer to be able to move it one time during the year. In this case, set the activation count to 2 and the activation type to **Reactivation**. Then, once your customer creates the first license it will have an expiration date one year out, but he can come back sometime later during that year and reactivate the license on another system. The new license would expire on the original expiration date. If he waited more than one year, no new valid license could be generated. You should use the **Reactivate** fulfillment type for Rehostable licenses if you want to deliver a license which expires some number of days after it is first activated. The count will control how many times they can revoke and re-activate the rehostable license.

For the **Refresh** fulfillment type, the count field must be 1 (rlc will store the count as 1 independent of what you enter). In other words, only a single activation can be supported in a Refresh license activation. This license is then refreshed over time, and the hostid can be changed up to the hostid change count. Refresh licenses will always have the REPLACE option specified as well as an ISSUED date. If multiple Refresh requests are made on the same day, all will return the same license. A new request on a subsequent day will return a new license. You should never use the **Refresh** fulfillment type for Rehostable licenses. (Note: **Refresh**-type activations are deprecated in RLM v9.3).

This processing for each type of activation key is described in the following table:

Activation type	New request from the same hostid (with same count)	New request from new hostid (or same hostid with different count)	expiration date of license
Normal	prior license returned	new license generated if sufficient activation count is available	computed at activation time
Reactivate	prior license returned	new license generated	same expiration date as first activation done with this activation key
Refresh (deprecated in RLM v9.3) – use REHOSTable licenses instead.	new license generated if request is on a new day, otherwise prior license returned	new license generated until hostid change count exceeded, then RLM_ACT_TOOMANY _HOSTID_CHANGES error returned	computed at activation time

In the table above, whenever the action listed is "prior license returned", the remaining activation count is not decremented. Whenever activation generates a new license, the remaining count is decremented by the license count generated.

Activation blacklist

RLM Activation allows you to specify a list of unauthorized domains - domains which cannot activate licenses. If you select the "View BlackList" button on the bottom-left side of the main form, you will see a list of blacklisted domains. The last item in the list is blank and allows you to add a new domain. In addition, each domain in the list has a **Delete** button on the right to remove it from the list.

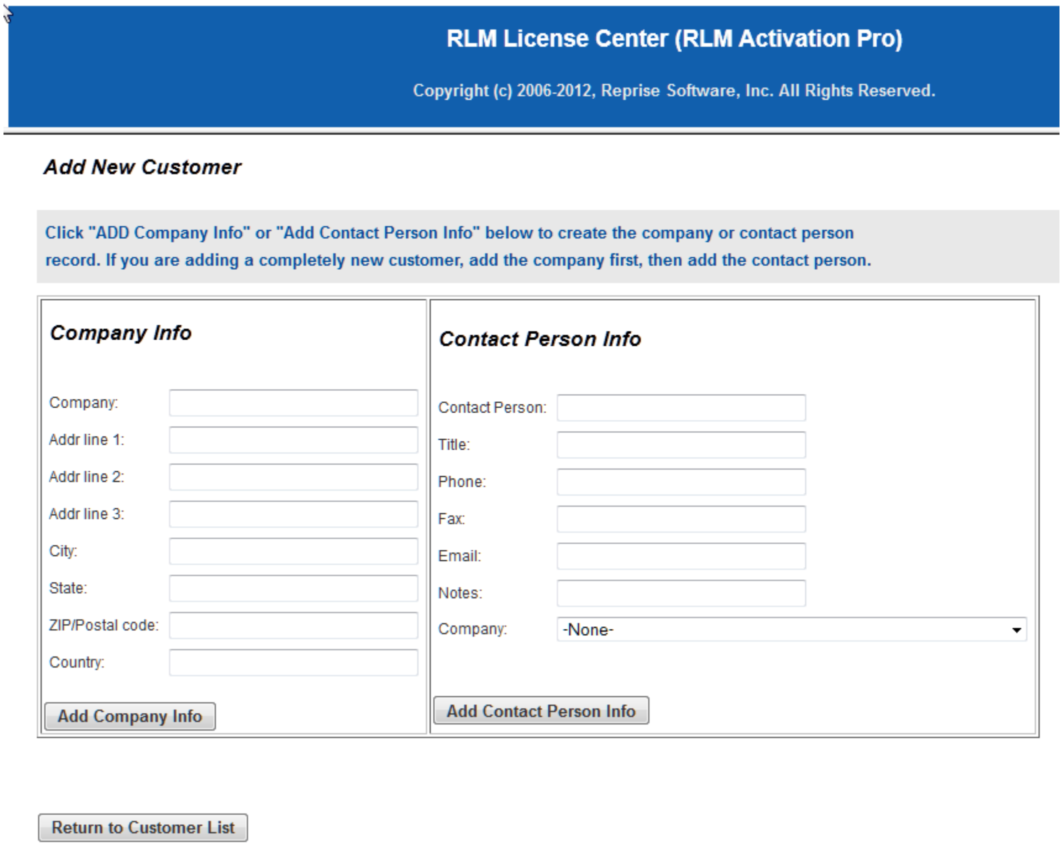
The blacklist is a list of domain names which are not allowed to activate using this activation key. If a request comes in from any domain on this list, it is rejected with RLM_ACT_BLACKLISTED. Note that if any string you enter here is a substring of the domain name then that domain will be rejected. All names are case-insensitive, and each item in the list must be less than RLM_MAX_HOSTNAME (64) bytes. If an activation key specifies allowed domains, the **blacklist** is not used, and the domain requesting the activation must be part of the allowed domains specification in the activation key itself.

Beginning in RLM v9.4, the license generator automatically blacklists IP addresses which attempt a large number of incorrect activation keys. These ip addresses will appear in normal IP address format (a.b.c.d), rather than as domain names, and an exact IP address match is required to blacklist the address, rather than the substring match that happens with manually-blacklisted domain names. When you are manually entering blacklisted domains, do *not* enter them in IP address format. You can delete any automatically-blacklisted IP address in this list by pressing the delete icon on the right-hand side of the list. For more information on the automatic blacklisting of IP address see Appendix K – Automatically blacklisting IP addresses on page 66.

Customer List

RLM Activation Pro allows you to maintain a customer list and associate customers with activation keys.

You create customers with the “Add Customer” button in the Setup Command area near the top of the command list. When you press “Add Customer”, the company/contact form appears:



RLM License Center (RLM Activation Pro)

Copyright (c) 2006-2012, Reprise Software, Inc. All Rights Reserved.

Add New Customer

Click "ADD Company Info" or "Add Contact Person Info" below to create the company or contact person record. If you are adding a completely new customer, add the company first, then add the contact person.

Company Info

Company:

Addr line 1:

Addr line 2:

Addr line 3:

City:

State:

ZIP/Postal code:

Country:

Contact Person Info

Contact Person:

Title:

Phone:

Fax:

Email:

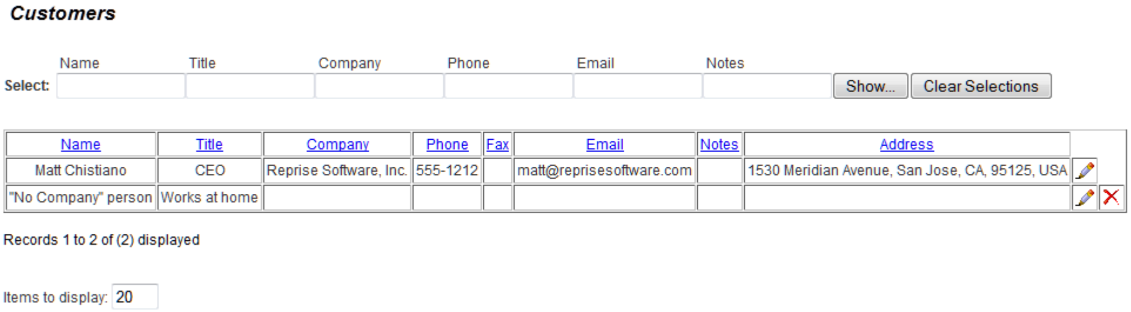
Notes:

Company: -None-

You first add the company, then add contact people for that company. Note that you are not required to add company information for contacts if you prefer to have only the person's name and other info from the Contact Person Info side of the form.

Once you have added some Contacts to the system, you can assign them to activation keys.

After creating contacts, the list is displayed.



Customers

Select: Name, Title, Company, Phone, Email, Notes

Name	Title	Company	Phone	Fax	Email	Notes	Address		
Matt Chistiano	CEO	Reprise Software, Inc.	555-1212		matt@reprisesoftware.com		1530 Meridian Avenue, San Jose, CA, 95125, USA	✎	
"No Company" person	Works at home							✎	✕

Records 1 to 2 of (2) displayed

Items to display: 20

You can edit individual contacts with the edit button on the right. If the contact is not associated with any activation key, pressing the delete button will confirm that you want to delete the contact. If you delete the last contact for a company, you will be prompted to delete the company record itself.

Database Viewing/Updating Commands

These commands are:

- View Product Definitions,
- View Activation Keys,
- View Fulfillments,
- View Customers,
- View BlackList,
- Reports,
- About rlc, and
- Manual...

These commands all allow viewing of the activation database. The first 2 commands also will allow you to edit product definitions and activation keys, by pressing the "Edit" icon (an image of a pencil) at the end of the row in the display. Product definitions and Activation Keys which are disabled are not deleted, but will appear in gray in the list, and can be re-enabled later.

While viewing activation keys, the product name is underlined. Hovering the cursor above the product name will display a popup listing the product's license name, version, and type.

You can also view the fulfillments done for a particular activation key by clicking on the "Show" icon (a small image of a computer screen) at the end of each row in the **View Activation Keys** display.

The **View Fulfillments** option will allow you to look at activation fulfillments from your website. In addition, beginning in v9.2, you can delete an individual fulfillment, in order to re-enable the count consumed on that activation key by that particular fulfillment. This is particularly helpful when you are testing the activation software and you want to be able to regenerate a license for a particular hostid, since a re-activation from the same hostid would return the old license. By deleting the old fulfillment, the activation software will generate a new license (with a new expiration date) for that activation key on the hostid.

The **View Customers** option allows you to view all customer information in the Activation Pro database. This form allows you to edit or delete the customer information as well. Note that you cannot delete contacts if they are associated with activation keys.

View Blacklist allows you to view and edit the list of blacklisted hosts. Note that if an entry in the blacklist matches any part of a hostname, that host will be blacklisted. Thus, "foo.com" in the blacklist will also blacklist the host "eng.foo.com".

Reports are described in a separate chapter. See Reporting on page 32.

About rlc displays information about the rlc program itself.

Finally, **Manual...** will display this manual in your browser.

Refresh License URL setup

If you plan to use REFRESH-type activations (deprecated in RLM v9.3), you will need to set up the URL on which your activation server will run. This is required for REFRESH type licenses only. Select **Administer Database** from the menu, enter the URL in the text box provided, and select "Set REFRESH URL" at the right-hand side of the text box. (Note: you must be logged in as the *admin* user for the **Administer Database** menu item to appear.)

Client Setup

In order to set up activation, you do 3 things:

1. install the Activation Pro software on your webserver (described in the Activation Pro Setup chapter on page 12) (Not necessary for Reprise Hosted Activation customers).
2. setup the Activation Database (described in the Database Setup chapter on page 17).
3. Add activation request calls to your application (described in this chapter)

On the client side, you must prepare your activation code and get the activation key to your customer.

Making the activation request for your customer - using the activation API

Once you are familiar with how to use the activation server and the activation GUI, you are ready to integrate the activation request into your application. To do this, embed a call to rlm_act_request() at the appropriate point (usually after attempting to check out a license which fails), and, if you are successful at activating the license, perform the following steps:

- call rlm_close() on your handle
- write out the activated license to a local license filename of your choice (Reprise Software recommends a file named *something*.lic)
- call rlm_init(), and
- re-attempt the license checkout.

For details on the activation API request call, see Appendix A – Activation API on page 37.

Making the activation request for your customer - using the activation GUI

You can get started right away by using the **activation GUI** supplied by Reprise Software. We recommend that you start this way, so that you can become familiar with the server side of activation without having to embed any code into your application.

In order to proceed, make sure that you have set up the server side as described in Activation Pro Setup on page 12.

The activation GUI is built into the rlm webserver. Run rlm (you don't need a license file), point your browser at port 5054 (or the port number you specified in the -ws port# specification) then select "activate license" from the menu on the left-hand side. The GUI will guide you through the steps in activating a license. (Note: prior to RLM v6.0, the default admin port was 9000, not 5054).

When the rlm webserver activates a license, the resulting license file will be placed in the directory from which rlm was started, and the file name will be "activateN.lic", where N is a sequence number. Each activation performed will get a new number.

Note that RLM activation will only operate with non-zero RLM_HOSTID_32BIT, RLM_HOSTID_ETHER, or ISV-defined hostid types - all other hostid types will return an error.

Activation API or rlm web interface - which to use?

Note that the rlm web interface is completely generic, and you can activate licenses for your products using this interface by setting up activation on your web server and specifying your URL in step 1. However, it will always be simpler for your end-users if you integrate the RLM activation API into your product or installation tools, since you can preset many of the activation parameters, such as URL, ISV name, etc.

Other Client-side activation options

Alternately, you can create a custom stand-alone activation utility for your customers, rather than building it into your application.   Follow the steps above for using the Activation API.

Also, you can create an html page to activate licenses using your activation server. For an example of this, see "activation_example.html" in the RLM kit "examples" directory. This technique, however, will require your customer to cut and paste the resulting license file into the appropriate file on their system.

Using Proxy Servers

The RLM Activation client software (either the rlm_act_request() API call or the RLM admin web interface) will make use of a user-defined proxy server. In order to use a proxy server, set the environment variable **HTTP_PROXY** or **http_proxy** to the hostname and port number of the proxy server.

For example, to utilize the HTTP proxy server running on "myproxyhost" on port 8765, use the following command on unix:

```
% setenv HTTP_PROXY myproxyhost:8765
```

If your proxy server uses authentication, you use the **HTTP_PROXY_CREDENTIALS** environment variable to pass the credentials to the proxy server:

HTTP_PROXY_CREDENTIALS - the username and password to authenticate you to the proxy server, in the format user:password.
For example, if your username is "joe" and password is "joes_password":

```
% setenv HTTP_PROXY_CREDENTIALS joe:joes_password
```

Note that RLM activation supports only the BASIC authentication type.

You can either set these environment variables before running your application, or use putenv() (or rlm_putenv()) to set them inside your application before calling *rlm_act_request()*.

Reporting

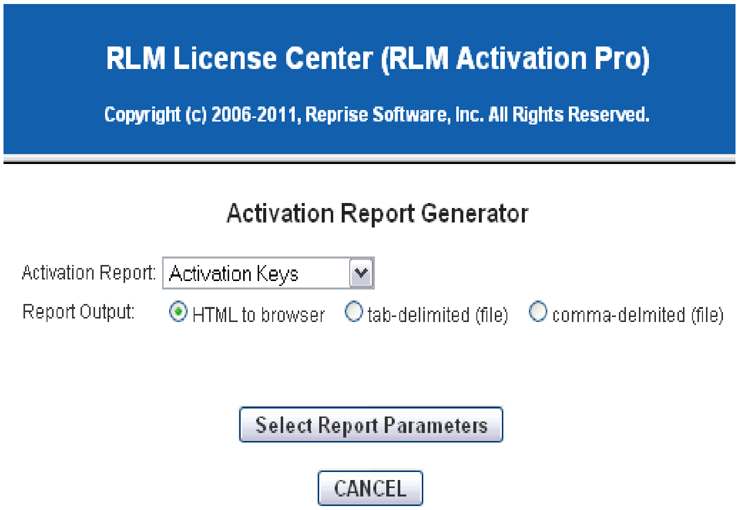
rlc contains a reporting module which allows you to generate reports of products, activation keys, fulfillments, etc.

To use the reporting module, select “Reports” from the menu on the left-hand side of the page. When you select “Reports”, the first Reporting screen appears:

The first choicelist allows you to select the report type: Activation Keys, Permanent Licenses, Fulfilled Licenses, Products, Revoked Licenses, or Statistics.

On the second line, you select the output for the report – either directly to the browser, or to a tab- or comma-delimited file which can be imported to a spreadsheet.

After making these selections, click “Select Report Parameters” which will present a screen allowing you to filter the output of the report. The contents of this screen are report-type dependent. The selection screen for the “Activation Keys” report is shown below:



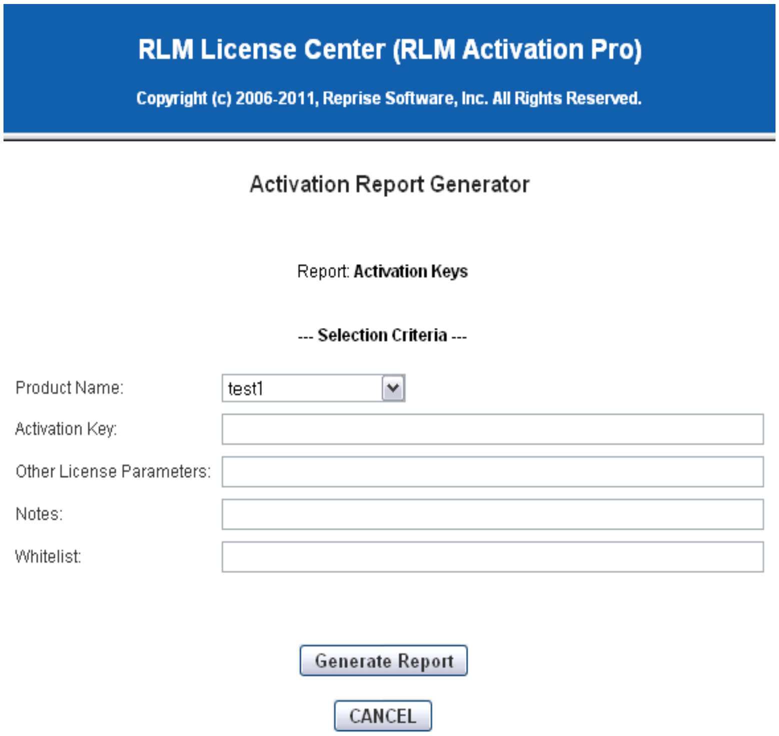
RLM License Center (RLM Activation Pro)

Copyright (c) 2006-2011, Reprise Software, Inc. All Rights Reserved.

Activation Report Generator

Activation Report: Activation Keys

Report Output: ● HTML to browser ○ tab-delimited (file) ○ comma-delmited (file)



RLM License Center (RLM Activation Pro)

Copyright (c) 2006-2011, Reprise Software, Inc. All Rights Reserved.

Activation Report Generator

Report: **Activation Keys**

--- Selection Criteria ---

Product Name: test1

Activation Key:

Other License Parameters:

Notes:

Whitelist:

The “Product Name” choicelist allows you to select an individual product or all products.

The other text boxes allow you to filter the various fields in the activation key table.

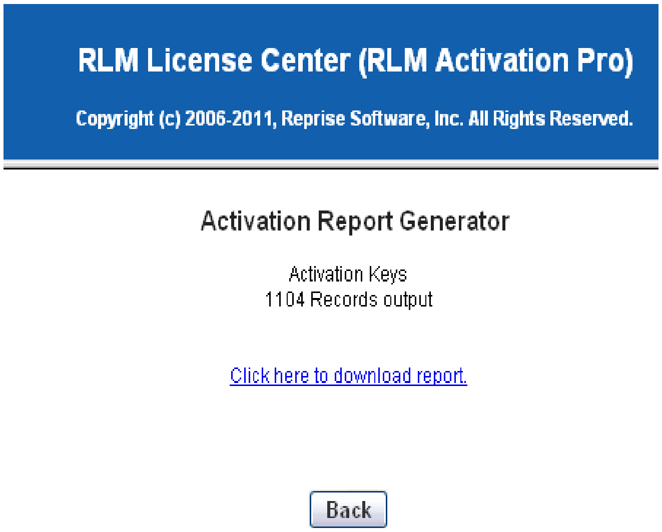
In each text box, the text you type will be used as a regular expression to select the appropriate rows from the activation key table. So, for example, if you have an activation key which is “1234-5678-abcd-efgh”, then typing “1234” in the Activation Key field will select this key (as well as any others with “1234” in the activation key string) for the report.

For the “statistics” report, you can select the date to report. Enter the date as yyyy-mm-dd. If you enter only yyyy-mm, a report for that month will be produced.

Once you have made your selection, click “Generate Report”, and the report will be output.

If you selected report output to the screen, you will see a form very similar to the “View Product Definitions” or “View Activation Keys” forms. You can page through the various pages as well as sort on the columns by clicking on the underlined column header.

If you selected either tab- or comma-delimited output, you will see the a screen similar to the following:



RLM License Center (RLM Activation Pro)

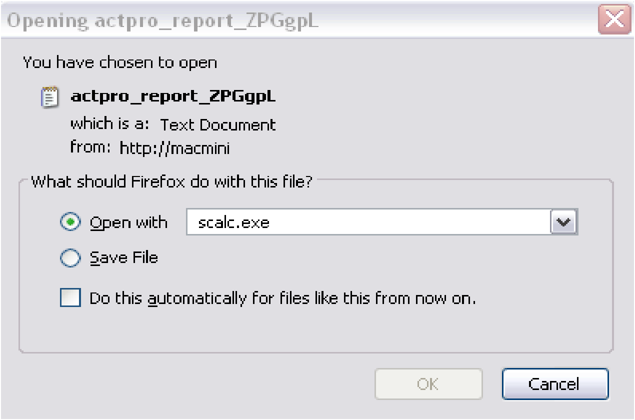
Copyright (c) 2006-2011, Reprise Software, Inc. All Rights Reserved.

Activation Report Generator

Activation Keys
1104 Records output

Click here to download report.

When you click on the underlined “Click here to download report”, your browser will ask you if you want to save the file, with a dialog similar to the following (Firefox shown here):



Opening actpro_report_ZPGgpL

You have chosen to open

actpro_report_ZPGgpL
which is a: Text Document
from: http://macmini

What should Firefox do with this file?

● Open with scalc.exe
○ Save File
□ Do this automatically for files like this from now on.

You can either open the file directly with your spreadsheet, or save the file. Once opened, the standard spreadsheet import functions for tab- or comma-delimited files will work to put your report data into separate columns.

Release Checklist

With RLM Activation Pro, you control the activation server, so you can make changes at any time. However, there are a couple of issues that you need to consider before you ship your application:

- Review the rkn_activate() (or rlm_act_request()) call you make in your application to be sure that you have the correct URL for your activation server.
- Be sure that your activation license generator and data files are protected on your webserver by limiting user access to **rlc**, especially **Admin** access.
- Be sure that your license generator (in the cgi-bin directory) , including any isv generator settings file (isv.gen) is read-write-execute to the web user only.

Importing Activation Keys to ActPro

On occasion, you might want to create activation keys outside of Activation Pro and then import those keys automatically into ActPro. You might do this, for example, if you wanted your order processing system to create an activation key for your customer's purchase, then populate ActPro with this activation key.

The activation pro kit contains an example program to accomplish this. This program is called *import_key.c* and it can be used as an example to write your own code, or it can be used as-is as a command-line utility.

Usage of *import_key* is as follows:

```
% import_key activation-key product-id [-c count] [-t n|r] [-e key-exp-date] [-l lic-exp-date]
                [-w whitelist] [-o other-params] [-n notes]
```

The product-id and activation-key are both required.

To determine the product-id, view the product definitions in the Activation Pro GUI, and hover over the Product name in the list. The product-id will be displayed. The activation-key can be any printable string of less than 60 characters.

All parameters correspond to the parameter in the “Create Activation Key” GUI.

The -t parameter specifies whether this key is a “normal” (n – the default), or ReActivate key (r).

To build *import_key*, you must have the MySQL connector kit installed (instructions for this are in step 4 of the Activation Pro Setup chapter. To compile and link import_key (where ***sql*** is the directory where the mysql connector kit is installed):

```
cc -c import_key.c -Isql/include

cc -o import_key import_key.o  sql/lib/libmysqlclient.a -lm
```

Section 2 – Reference Material

Appendix A – Activation API

The RLM Activation API consists of five calls:

- *rlm_activate()* (the new, preferred activation request call)
- *rlm_act_new_handle()*
- *rlm_act_destroy_handle()*
- *rlm_act_set_handle()*
- *rlm_act_revoke()*

In addition, the following 2 calls are now deprecated:

- *rlm_act_request()* (the old activation request call)
- *rlm_act_refresh()*

These calls are all documented in the RLM Reference Manual.

Appendix B – Example Program

The following is the source code to the activation demo program actdemo: This example program (actdemo.c) is contained on the RLM kit in the *examples* directory.

```
******************************************************************************

            COPYRIGHT (c) 2005, 2012 by Reprise Software, Inc.
        This software has been provided pursuant to a License Agreement
        containing restrictions on its use.  This software contains
        valuable trade secrets and proprietary information of
        Reprise Software Inc and is protected by law.  It may not be
        copied or distributed in any form or medium, disclosed to third
        parties, reverse engineered or used in any manner not provided
        for in said License Agreement except with the prior written
        authorization from Reprise Software Inc.

 ******************************************************************************/
/*
 *      Description:    Example client for RLM internet activation
 *
 *      M. Christiano
 *      6/18/07
 *
 *      $Id: RLM_Activation_Pro.html,v 1.2 2011/02/25 00:24:26 matt Exp $
 */

#include "license.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>


static int getstat(RLM_HANDLE, RLM_LICENSE, const char *);

int
main(int argc, char *argv[])
{
  RLM_HANDLE rh;
  RLM_LICENSE lic;
  int x, stat;
  const char *product = "actdemo";
  int loop = 0;

        while (loop < 2)
        {
            rh = rlm_init(".", argv[0], (char *) NULL);
            if (rh == (RLM_HANDLE) NULL)
            {
                (void) printf("Error initializing license system\n");
                exit(1);
            }

            lic = rlm_checkout(rh, product, "1.0", 1);
/*
 *          getstat() checks the status, and if bad, tries an activation
 *          if the user requests it.  If the returned status is 0 (good) or
 *          < 0 (error), we do not continue.  If the returned status is
 *          positive, the activation succeeded, and we need to re-init and
 *          attempt the checkout again.
 */
            stat = getstat(rh, lic, product);
            if (stat <= 0) break;

            if (lic) rlm_checkin(lic);
            rlm_close(rh);
            loop++;
        }

        if (stat == 0)
        {
/*
 *              We got the license
 */
                (void) printf("Enter <CR> to continue: ");
                x = fgetc(stdin);
                if (lic) rlm_checkin(lic);
                rlm_close(rh);
        }
        else
        {
/*
 *              checkout and/or activation failed.
 */
                (void)
                   printf("Unable to check out/activate \"actdemo\" license\n");
        }
        return(0);
}

static
int
getstat(RLM_HANDLE rh, RLM_LICENSE lic, const char *name)
{
  int stat, x;
  char errstring[RLM_ERRSTRING_MAX];
  char license[RLM_MAX_LINE+1];
  char akey[RLM_MAX_LINE+1];

        stat = rlm_license_stat(lic);
        if (stat == 0)
            (void) printf("Checkout of %s license OK.\n", name);
        else
        {
            (void) printf("Error checking out %s license\n", name);
            (void) printf("%s\n", rlm_errstring(lic, rh, errstring));
            (void) printf("\nWould you like to activate this license now? ");
            gets(akey);
            if (*akey == 'y' || *akey == 'Y')
            {
                (void) printf("Enter Activation key for \"%s\": ", name);
                gets(akey);
                stat = rlm_act_request(rh, "http://www.reprisesoftware.com",
                                                "rlmactdemo", akey, "", "", 1,
                                                license);
                if ((stat == 0) || (stat == 1))
                {
                  char name[100];
                  int try;
                  FILE *f, *fopen();

                    stat = 1;
/*
 *                  Write the license out
 */
                    for (try=0; try<100; try++)
                    {
                        sprintf(name, "a%d.lic", try);
                        f = fopen(name, "r");
                        if (f == (FILE *) NULL)
                        {
                                f = fopen(name, "w");
                                if (f)
                                {
                                    fprintf(f, "%s\n", license);
                                    fclose(f);
                                    break;
                                }
                                else
                                {
                                    printf(
                                          "Error writing license file \"%s\"\n",
                                                                        name);
                                    stat = -1;
                                    break;
                                }
                        }
                    }
                }
                else
                {
                    printf("\n");
                    switch(stat)
                    {
                        case RLM_EH_CANTCONNECT_URL:
                          printf("You were unable to connect to
                                     http://www.reprisesoftware.com.\n");
                          printf("Please make sure that this system is able to\n");
                          printf("access the internet and try again.\n");
                          break;

                        case RLM_ACT_NO_KEY:
                          printf("The activation key you supplied (%s) was \n",
                                                                        akey);
                          printf("not found.  Please check the key and ensure\n");
                          printf("that you have entered it correctly.\n");
                          break;

                        case RLM_ACT_KEY_USED:
                          printf("The activation key you supplied (%s)\n", akey);
                          printf("has already been used to activate a license.\n");
                          printf("Please check the key and ensure that you have entered
                                             it correctly.\n");
                          break;

                        case RLM_ACT_BAD_HTTP:
                          printf("HTTP error on activation server\n");
                          break;

                        default:
                          printf("Error %d requesting activation\n", stat);
                          break;

                    }
                    printf("\n");
                }
            }
        }
        return(stat);
}
```

Appendix C – License Rehosting

Rehostable Licenses

Activation can be set up to support rehosting of licenses by the end user without ISV involvement. This involves the use of the rehostable hostid type, which is new in RLM v9.3. A rehostable hostid is requested at the time that *rlm_activate()* is called, and the activation server issues the license to this hostid. Later, if the user wishes to move this license to another system, the *rlm_revoke()* call can be used to revoke the rehostable license and allow it to be activated on another system. Note that the rehostable hostid is a copy-protected file, and if the system should crash, the hostid cannot be retrieved. As an ISV, you will need a policy to deal with this situation – e.g., to give the customer another activation in order to continue. Since this situation should be rare, a non-automated request procedure might be quite sufficient.

Rehostable license support is limited to only nodelocked, uncounted licenses. In addition, only a single version of any product can exist with a rehostable hostid on a system at any given time. Attempting to activate a 2nd product with a different version number will give an RLM_EH_REHOST_EXISTS (-153) error from activation. Finally, you should always use a fixed expiration date if you are using the **Normal** fulfillment type for Rehostable licenses. If you want an expiration date that is relative to the time of fulfillment, then you should use a **ReActivate** fulfillment type and set the count to the # of times you want your customer to be able to revoke and re-activate the license.

Rehostable License Example code

An example for rehostable licenses is on the RLM kit in the examples directory, called rehost_example.c It is repeated here. The important calls to do the rehosting are highlighted in yellow:

```
/******************************************************************************

            COPYRIGHT (c) 2005, 2012 by Reprise Software, Inc.
        This software has been provided pursuant to a License Agreement
        containing restrictions on its use.  This software contains
        valuable trade secrets and proprietary information of
        Reprise Software Inc and is protected by law.  It may not be
        copied or distributed in any form or medium, disclosed to third
        parties, reverse engineered or used in any manner not provided
        for in said License Agreement except with the prior written
        authorization from Reprise Software Inc.

 *****************************************************************************/
/*
 *      Description:   Example client for RLM license rehosting/revoking
 *
 *      M. Christiano
 *      11/18/11 - modified from the regular activation example
 *
 *      $Id: rehost_example.c,v 1.3 2012/01/04 00:44:04 matt Exp $
 */

#include "license.h"
#include <stdio.h>
#include <stdlib.h>
```

```
#include <string.h>

static int checkstat(RLM_HANDLE, RLM_LICENSE, const char *);
static int doactivation(RLM_HANDLE, const char *);
static int dodeactivation(RLM_HANDLE, const char *);

int
main(int argc, char *argv[])
{
  RLM_HANDLE rh = (RLM_HANDLE) NULL;
  RLM_LICENSE lic = (RLM_LICENSE) NULL;
  int stat;
  const char *product = "actdemo";
  int pass;

        for (pass = 1; pass <= 2; pass++)
        {
            if (pass == 2)
            {
/*
 *             We did not get the license on the first pass.  See
 *             if we can activate it now.
 */
               stat = doactivation(rh, product);
                rlm_close(rh);        /* Close the old handle */
               if (stat < 0) exit(2);
            }
/*
 *          NOTE that rlm_init() is in this loop.  This is necessary
 *          because if the license is not acquired on the first pass,
 *          the activation request (above) will create a new
 *          license file.  rlm_init() must be called to see this
 *          new license file.
 */
            rh = rlm_init(".", argv[0], (char *) NULL);
            stat = rlm_stat(rh);
            if ((pass == 1) && (stat == RLM_EH_READ_NOLICENSE))
            {
/*
 *             We didn't find a license.  Checkout won't work, so just
 *             skip to the 2nd pass where we try to activate it.
 */
               continue;
            }
            else if (stat)
            {
               (void) printf("Error %d initializing license system\n", stat);
               exit(1);
            }

            lic = rlm_checkout(rh, product, "1.0", 1);

            stat = checkstat(rh, lic, product);

            if ((pass == 1) && (stat < 0))
            {
/*
 *             Didn't get the license.  Try a 2nd time to activate it.
 */
                if (lic) rlm_checkin(lic);
               lic = (RLM_LICENSE) NULL;
               continue;
            }
            else if (stat == 0)
            {
/*
 *              We got the license
 */
                break;
            }
        }
        if (stat == 0)
        {
/*
 *              We got the license
 */
                if (lic) rlm_checkin(lic);
                printf("License valid.\n");
                stat = dodeactivation(rh, product);
                rlm_close(rh);
        }
        else if (pass == 2)
        {
/*
 *              checkout and/or activation failed.
 */
                (void)
                   printf("Unable to check out/activate \"actdemo\" license\n");
        }
        return(0);
}

static
int
checkstat(RLM_HANDLE rh, RLM_LICENSE lic, const char *name)
{
  int stat;
  char errstring[RLM_ERRSTRING_MAX];

        stat = rlm_license_stat(lic);
        if (stat == 0)
            (void) printf("Checkout of %s license OK.\n", name);
        else
        {
            (void) printf("Error checking out %s license\n", name);
            (void) printf("%s\n", rlm_errstring(lic, rh, errstring));
        }
        return(stat);
}

#include <time.h>
#ifndef _WIN32
#include <sys/time.h>
#endif
int
doactivation(RLM_HANDLE rh, const char *name)
{
  char license[3 * RLM_MAX_LINE + 1]; /* Allow for HOST, ISV, and LICENSE */
  char akey[RLM_MAX_LINE+1];
  int stat = RLM_EH_READ_NOLICENSE;   /* If they say NO, no license */
  int len;
  RLM_ACT_HANDLE act_handle;

        (void) printf("\nWould you like to activate this license now? ");
        fgets(akey, RLM_MAX_LINE, stdin);
        if (*akey == 'y' || *akey == 'Y')
        {
                 (void) printf("Enter Activation key for \"%s\": ", name);
                fgets(akey, RLM_MAX_LINE, stdin);
                len = ((int) strlen(akey)) - 1;
                if (akey[len] == '\n') akey[len] = '\0';
/*
 *              Request the license.  First make a handle, and tell it we
 *              want a rehostable hostid to activate.
 */
                act_handle = rlm_act_new_handle(rh);
                rlm_act_set_handle(act_handle, RLM_ACT_HANDLE_REHOST,
                                                            (void *) 1);
/*
 *              Note - you would normally never need to do this, but we
 *              do it here so that we can connect to the license generator
 *             on the reprise demo site.
 */
               rlm_act_set_handle(act_handle, RLM_ACT_HANDLE_ISV,
                                                     (void *) "rlmactdemo");
               stat = rlm_activate(
                       rh,            /* RLM handle */
                       "http://www.reprisesoftware.com", /* URL */
                       akey,          /* Activation key */
                       1,             /* Activation count - # of licenses */
                       license,       /* Space for the returned license */
                       act_handle     /* Activation handle data */
                       );

               rlm_act_destroy_handle(act_handle);   /* Done with this */

                if ((stat == 0) || (stat == 1))
                {
                 char name[100];
                 int try;
                 FILE *f;
/*
 *                 Activation was successful.  Write the license out.
 *                 Note in this example, we try the license file name
 *                 aN.lic, and we only try 100 different names.  You
 *                 should change this to whatever naming convention you
 *                 want to use.
 */
                   for (try=0; try<100; try++)
                   {
                       sprintf(name, "a%d.lic", try);
                       f = fopen(name, "r");
                       if (f == (FILE *) NULL)
                       {
                         struct tm *t;
#ifdef _WIN32
                         time_t ltime;

                              time(&ltime);
                              t = localtime(&ltime);
#else
                         struct timezone tz;
                         struct timeval tv;
                         time_t x;

                              gettimeofday(&tv, &tz);
                              x = tv.tv_sec;
                              t = localtime((time_t *) &x);
#endif

                              f = fopen(name, "w");
                              if (f)
                              {
                                  fprintf(f, "This license created by RLM Internet 
Activation\n");
                                  if (t)
                                    fprintf(f, "Created on %02d/%02d/%04d at %02d:
%02d\n",
                                             t->tm_mon+1, t->tm_mday,
                                             t->tm_year+1900, t->tm_hour,
                                             t->tm_min);

                                  fprintf(f, "\n%s\n", license);
                                  fclose(f);
                                  printf(
                       "Activation successful, license file \"%s\" written\n",
                                                            name);
                                  break;
                              }
                              else
                              {
                                  printf(
                                         "Error writing license file \"%s\"\n",
                                                                       name);
                                  stat = -1;
                                  break;
                              }
                      }
                   }
               }
               else
               {
                   printf("\n");
                   switch(stat)
                   {
                     char err[RLM_ERRSTRING_MAX+1];

                      case RLM_EH_CANTCONNECT_URL:
                        printf("You were unable to connect to
http://www.reprisesoftware.com.\n");
                        printf("Please make sure that this system is able to\n");
                        printf("access the internet and try again.\n");
                        break;

                      case RLM_ACT_NO_KEY_MATCH:
                        printf("The activation key you supplied (%s) was \n",
                                                                       akey);
                        printf("not found.  Please check the key and ensure\n");
                        printf("that you have entered it correctly.\n");
                        break;

                      case RLM_ACT_KEY_USED:
                        printf("The activation key you supplied (%s)\n", akey);
                        printf("has already been used to activate a license.\n");
                        printf("Please check the key and ensure that you have entered it
correctly.\n");
                        break;

                      case RLM_EH_BAD_HTTP:
                        printf("Bad HTTP transaction\n%s\n",
                                             rlm_errstring(0, rh, err));
                        break;

                      default:
                        printf("Error %d requesting activation\n%s\n",
                                          stat,
                                          RLM_ACT_ERR(stat) ?
                                             rlm_act_errstring(stat) :
                                             rlm_errstring(0, rh, err));
                        break;

                   }
                   printf("\n");
               }
       }
       return(stat);
}

int
dodeactivation(RLM_HANDLE rh, const char *name)
{
  int stat = RLM_EH_READ_NOLICENSE;   /* If they say NO, no license */
  char x[100];

       (void) printf("\nWould you like to deactivate the \"%s\" license now? ",
                                                                       name);
       fgets(x, RLM_MAX_LINE, stdin);
       if (*x == 'y' || *x == 'Y')
       {
/*
 *              Request the deactivation.
 */
               stat = rlm_act_revoke(rh, "www.reprisesoftware.com",
                                                            (char *) name);
                if (stat == 0)
                {
/*
 *                     revoke (Deactivation) was successful.
 */
                       printf("License successfully revoked\n");
               }
               else
               {
                 char err[RLM_ERRSTRING_MAX+1];

                       printf("\n");
                       printf("Error %d requesting license revoke\n%s\n",
                                         stat,
                                         RLM_ACT_ERR(stat) ?
                                             rlm_act_errstring(stat) :
                                             rlm_errstring(0, rh, err));
                       printf("\n");
               }
        }
        return(stat);
}
```

Appendix D – Activation Database

RLM activation uses a MySQL database to store all data. There are a total of 11 database tables:

- active_users - the currently active users,
- company – table of customer companies
- contact – table of customer contacts
- blacklist - blacklist definitions,
- keyd - The Activation Key definitions,
- prod - The product definitions,
- setup - Activation setup data,
- keyf - The Activation key fulfillment status,
- licf - License Fulfillment data
- report_types – report type definitions
- report_select – selection criteria for reports
- report_cols – database columns which appear in reports
- users - Table of authorized users,

Each table indicates whether the admin tool and/or the license generator read and write to it.

The contents of these tables is as follows:

active_users - active **rlc** users

This table is used only by the admin tool. Unused by the license generator.
Contents:

column	meaning	data type
user	username logged in	string
timestamp	Time user logged in	integer

company – customer company information

This table is used only by the admin tool. Unused by the license generator.
Contents:

column	meaning	data type
company_id	Company ID (primary key)	integer
company	Company name	string
addr1	Address line 1	string
addr2	Address line 2	string
addr3	Address line 3	string
city	City	string
state	State	string

zip	Zip/Postal code	string
country	Country	string
u1	Unused	string
u2	Unused	string
u2	Unused	string

contact – customer contact person information

This table is used only by the admin tool. Unused by the license generator. Contents:

column	meaning	data type
contact_id	Contact ID (primary key)	integer
contact	Contact person name	string
Title	Job Title	string
phone	Phone #	string
fax	Fax #	string
email	Email address	string
notes	General notes	string
company_id	ID of associated company record	integer

blacklist - blacklist definitions, one row per blacklisted host.

This table is written only by the admin tool. Read-only by the license generator. Contents:

column	meaning	data type
id	row id	integer
domain	the hostname/domain of the blacklisted host. If this string matches any part of the requesting host, that host is not allowed to use the activation server	string (255). No spaces.

keyd - activation key definitions, one row per activation key.

This table is written only by the admin tool. Read-only by the license generator.
Contents:

column	meaning	data type
akey	the activation key	string
active	0 if this definition is inactive, 1 if active	integer
product id	product identifier, from product definitions	integer
count	the # of activations allowed, 0 = unlimited. Limited to 1 for Refresh-type activations.	integer
type	0 - Normal, 1 - Reactivate, 2 = Refresh	integer
rehosts	For Refresh, the number of hostid changes allowed	integer
lastdate	Expiration date of activation key - key cannot be used after this date	string
exp	If a number, number of days to expiration of the license after product activation. If == 0, the license does not expire. If a standard RLM date format, it is a fixed expiration date. If specified, this expiration date overrides the expiration date in the product definition.	string
white	Domain whitelist for key - only domain names appearing in this list can use the key. This string can be a space-separated list. If any component of the string matches the requesting hostname, that host is allowed to activate using this key.	string (100)
misc	any optional RLM license attributes	string (255)
notes	optional notes to allow you to identify the key - unused by RLM	string (100)

keyf - activation key fulfillment information, one row per activation key.

This table is written by both the license generator and rlc.
Contents:

column	meaning	data type
akey	the activation key	string
num	Number of fulfillments from this key	integer
date	For Reactvation: original expiration date of the license. For Refresh: last date license refreshed	string
lasthostid	For Refresh type: Last hostid fulfilled	stri

licf - license fulfillment information, one row per activation.

This table is written by both the license generator and rlc.
Contents:

column	meaning	data type
id	row ID	integer
akey	the activation key	string
product_id	product identifier	integer
count	number of licenses activated	integer
time	time() value when activation was done	time_t (integer)
license_hostid	hostid for license	string
hostname	hostname on activated machine	string
remote_host	IP address (typically) of machine requesting activation	string
log	Log info from rlm_act_request() call	string
license	generated license. For floating licenses, this is just the LICENSE line	string
last_fulfill	time() value at last fulfillment. (v9.3)	time_t (integer)
num_fulfills	Number of fulfillments of this license (v9.3)	integer
revoked	“license was revoked” flag (1=revoked) (v9.3)	integer
last_fulfill	time() value when revoked. (v9.3)	time_t (integer)

prod - product definitions, one row per product definition.

This table is written only by the admin tool. Read-only by the license generator.
Contents:

column	meaning	data type
id	activation product identifier	integer
active	0 if this definition is inactive, 1 if active (product definitions are not deleted)	integer
name	Product Definition name	string
version	the RLM product version	string
upgrade_ version	Upgrade-from version number (UPGRADE licenses only)	string
exp	If a number, number of days to expiration of the license after product activation. If == 0, the license does not expire. If a standard RLM date format, it is a fixed expiration date.	string
lictype	0 = floating, 1 = node-locked, uncounted, 2 = node-locked, counted (unimplemented), 3 = single, 4 = UPGRADE (floating)	integer
issued	0 - do not include "issued=" in license, 1 - include "issued=[today's date]" in license	integer
product	License product name	string
misc	any optional RLM license attributes	string

setup - RLC setup definitions, one row per setup item.

This table is written only by the admin tool. Read-only by the license generator.
This table contains data such as the activation URL for refresh, Activation Pro version, etc.
Contents:

column	meaning	data type
what	Description of the data item	string (30)
data	The actual data	string (64)

users - authorized **rlc** users

This table is used only by the admin tool. Unused by the license generator.
Contents:

column	meaning	data type
user	username	string
password	user's encrypted password	string
userid	User Identifier	string
userlevel	User access (1=view, 2=edit, 9=admin)	integer
timestamp	Last time user logged in	integer

report_cols– database columns which appear in report.

This table is read-only by rlc.   Unused by the license generator.
Contents:

column	meaning	data type
id	Primary key	int
report	Report name	string (40)
displayorder	Order to display selections	int (2)
tablecolumn	Column name from table	string (40)
display	Report column header text	string (30)
var	POST variable name	string (20)
is_int	If non-zero, column contains integer data	int (1)

report_types – report type definitions.

This table is read-only by rlc. Unused by the license generator.
Contents:

column	meaning	data type
report	Report name	string (40)
tablename	Table for report data	string (40)
displayorder	Order to display report names	int (2)
report_header	Display header for report	string (80)

report_select – report selection definitions.

This table is read-only by rlc.   Unused by the license generator.
Contents:

column	meaning	data type
id	Primary key	int
report	Report name	string (40)
fixed	Is selection criteria fixed or variable	int (1)
displayorder	Order to display selections	int (2)
display	Text to display in UI	string (40)
var	POST variable name	string (20)
size	Size of text box in UI	int (3)
op	Table for report data	string (20)
value	Value for fixed selection criteria	string (64)

stats – Activation statistics.

This table is read-only by rlc.   Read-write by the license generator.
Contents:

column	meaning	data type
date	Date in the format" yyyy-mm-dd hh". This record collects statistics for 60 minutes starting at the hour indicated. Primary key	char(13)
total	Total transactions with the activation server	int (11)
good	Completed license fulfillments	int (11)
badkey	Unknown activation key supplied	int (11)
checks	Reprise pings the hosted activation servers with an invalid activation key to check that the website is up, the license generator is working and the database is working.. This is a bad key where the key supplied is "rsi-check-act-server"	int (11)
numfulfill	Number of "num fulfill" commands executed. Used by rlm_act_revoke()	int (11)
rmfulfill	Number of "remove fulfill" commands executed. Used by rlm_act_revoke()	int (11)
badtime	Number of otherwise good requests from clients with clocks which are off by more than 7 days	int (11)
prod	Number of "list product" commands executed. Used by rehostable hostid activation.	int (11)

Appendix E – Upgrading from RLM Internet Activation

If you are a current RLM Internet Activation customer, you will have a production activation database. This section describes how to upgrade that database to *RLM Activation Pro*.

The upgrade procedure should be performed ***after*** you install *RLM Activation Pro,* but ***before*** you add any data to the database.

To upgrade your production database, change directory to the directory containing your RLM Internet Activation data files, and perform the following steps:

- cd ***activation* directory**
- run rlmact2sql from the activation pro directory:

  ```
  % activation_pro_dir/rlmact2sql isvname
  ```

  (where *isvname* is your ISV name)

  This will create a database update file named *isvname*_db.sql
- Login to activation pro as the admin user.
- In the login section of activation pro, select "Administer Database"
- In the section of the next screen, under "Upload Activation database data", click to browse to the *isvname*_db.sql file you just created.
- Select "Upload File" to the right-hand side of the filename.

That is it. rlmact2sql will read your existing data from the import file and create the data in your new Activation Pro MySQL database.

Appendix F - Frequently-Asked Questions

Reprise Software maintains a list of frequently-asked questions on our website. For the current list of Frequently-Asked Questions, please see our website.

For ISVs, see the FAQ at our support site at:

http://www.reprisesoftware.net/support_forum

For Your End-Users, see the End-User FAQ at

http://www.reprisesoftware.com/support/faq.php

Appendix G – Document Revision History

V9.3– 15-February-2012 - v9.3 release (BL2)

v9.3 – 21-November-2011 – First v9.3 beta release (BL1).

V9.2 – 28-September-2011 - v9.2 release (BL2)

v9.2 - September-2011 - First v9.2 release (BL1) – first release with PDF manual.

V9.1 – 2-May-2011 - v9.1 release (BL3) (HTML wiki manual)

v9.1 - Mar-2011 - Second v9.1 beta release (BL2) – second beta release (HTML wiki manual)
v9.1 - Feb-2011 - First v9.1 beta release, first release with Activation PRO (BL1) (HTML wiki manual)

Appendix H – Diagnosing Activation Problems

When you call rlm_act_request(), errors can come from either the server side, or locally. This section lists most of these errors and what they mean.

Errors detected and returned by the activation server (ISV_mklic):

Status return	Problem	Solution
RLM_EH_BADPARAM (-123)	Error creating license	If RLM_EH_BADPARAM is returned without a corresponding secondary error, this is an error on the server side generating the license. Most likely, this means that some part of the license contains illegal characters – generally in the "extra" parameters supplied in the product definition, activation key, or in the rlm_act_request() call itself.
RLM_EH_ACT_BADLICKEY (-135)	Bad license key in activation binary/settings file.	Your *ISV_mklic* binary (or generator settings file) has incorrect RLM license keys. Re-build and install the new binary or settings file.
RLM_EH_BAD_HTTP (-136)	An unknown HTTP error was returned to the RLM activation client software.	This is usually an installation problem. Some possibilities are: • you have specified an ISV name which is incorrect • the ISV web server is not configured correctly • the *ISV_mklic* binary is not for the correct architecture
RLM_EH_ACTPRO_UNLICENSED (-160)	Activation Pro license missing or invalid	Check the actpro.lic file in the cgi-bin directory. Make sure it is for "rlm_actpro_ISVNAME" where ISVNAME is your ISV name, and that it hasn't expired.
RLM_ACT_NO_KEY (-1002)	No activation key supplied to the activation server.	This is an error in the configuration of your software
RLM_ACT_NO_PROD (-1003)	Activation key specifies a non-existent product.	Make sure your activation key refers to an existing product. This is a setup problem.
RLM_ACT_CANT_WRITE_KEYS (-1004)	Cannot write the license key fulfillment data.	This is an internal server error. Check to make sure the MySQL server is running correctly.
RLM_ACT_KEY_USED (-1005)	Activation key already used	The supplied activation key has already been used and no more activations are available.
RLM_ACT_BAD_HOSTID (-1006)	No hostid supplied to the activation server.	This is an error in the configuration of your software
RLM_ACT_BAD_HOSTID_TYPE	No acceptable hostid	Activation will only accept hostids of types:

Status return	Problem	Solution
(-1007)	type transmitted to activation server	32-bit, ethernet, disksn, rlmidN, or ISV-defined. In addition hostids with value 0 are rejected as is 32-bit hostid 7f0100 (the default linux hostid). This error will also occur if there is an illegal character in a valid hostid type.

RLM_ACT_CANTREAD_DB
(-1010)
	Cannot initialize MySQL or connect to the database	Make sure that the isvname.mysql file in the cgi-bin directory contains the correct server name/port/username/password.
		This error can also result from a missing or empty generator settings file in the case of RLM Hosted Activation.
		To debug this problem (v9.3 or later), set the environment variable "RLM_ACT_SQL_DEBUG", then run the activation generator as follows:
		`ISV_mklic << EOF`
		`akey=1&hostid=2`
		`EOF`
		The generator will then print (to stdout) the SQL error status.

RLM_ACT_CANT_WRITE_FULFILL (-1011)	Cannot write the license key fulfillment data.	This is an internal server error. Check to make sure the MySQL server is running correctly.
RLM_ACT_CLIENT_TIME_BAD (-1012)	Time on client machine is more than 7 days off from server machine.	Correct the time difference between client and server machines (must be < 7 days)
RLM_ACT_TOOMANY_HOSTID_CHANGES (-1014)	Refresh-type activation has had it's hostid changed too many times.	
RLM_ACT_BLACLISTED (-1015)	Activation attempted from a host which is blacklisted for this server	Update the blacklist if it is not accurate
RLM_ACT_NOT_WHITELISTED (-1016)	Activation attempted from a host not on the whitelist for this activation key	Update the whittelist if it is not accurate
RLM_ACT_KEY_EXPIRED (-1017)	The activation key itself has expired	This would be a normal error for your customer who attempts to use an expired activation key.
RLM_ACT_BAD_GENERATOR (-1020)	License keys in generator (ISV.gen) bad or expired, or the generator is missing.	Check your license generator to make sure it was built with a correct license_to_run.h
RLM_ACT_NO_KEY_MATCH	Activation Key can't be	This would be a normal error for a customer

Status return	Problem	Solution
(-1021)	found in database	who uses an incorrect activation key
RLM_ACT_DB_READERR (-1026)	Error reading data from MySQL	The activation key table has the wrong number of columns (this can happen if you use a v9.1 database with the v9.2 software). The product definition table has the wrong number of columns.
RLM_ACT_GEN_PARAM_ERR (-1027)	Error in generator call to rlm_sign_license()	The license string passed into rlm_sign_license() is incorrect, most likely because something in your extra license parameters is invalid.
RLM_ACT_UNSUPPORTED_CMD (-1028)	Unsupported command	This is an error that a non-Activation PRO server returns when an advanced command is issued to it.
RLM_ACT_REVOKE_TOOLATE (-1029)	Revoke attempted after license expiration.	Licenses cannot be revoked after they expire.

Errors detected and returned locally by rlm_act_request() or by the HTTP server:

Status return	Problem	Solution
RLM_EH_NET_INIT (-103)	Can't initialize the networking code.	Most likely the URL specified in the call is incorrect.
RLM_EH_NET_WERR (-104)	Cannot write data to network.	Most likely the URL specified in the call is incorrect.
RLM_EH_NET_RERR (-105)	Can't read returned message from network.	If the write succeeds and the read fails, then it is likely that there is a proxy server in the middle. Try setting HTTP_PROXY.
RLM_EH_BADPARAM with RLM_ACT_BP_TOOMUCH sub-error (-123)	Request is too long to communicate.	Check the length of the extra license parameters, the hostid list, and the log data. Each of these should be < 80 bytes.
RLM_EH_CANTCONNECT_URL (-132)	Can't connect to the specified URL.	Check that the URL is correct and that the activation server is installed in cgi-bin. Pass the URL without the trailing /cgi-bin, e.g. http://www.reprisesoftware.com, NOT http://www.reprisesoftware.com/cgi-bin
RLM_EH_ACT_BADSTAT (-134)	rlm_act_request() received no status in the returned message from *ISV_mklic*	This is most likely a configuration error on the server side, which causes the *ISV_mklic* process to return an incomplete reply. The HTTP server error logs might well indicate the problem. If your server is known to be running correctly, this could be caused by a proxy server that isn't configured correctly. This can also be caused by the Avast anti-virus software - which interrupts the return from the activation server and returns a partial buffer, causing this error. The workaround is to turn off the Avast "Web Shield". This can be turned off for 10 minutes in the Avast GUI.
RLM_EH_BAD_HTTP (with sub-error equal to HTTP status) (-136)	Other HTTP error	See the full error message for the HTTP status. Use RLM's extended error messages.
RLM_EH_UNLICENSED (-143)	RLM license keys missing or invalid	Check your license_to_run.h include file and ensure your RLM keys are valid.
RLM_EH_ACT_OLDSERVER (-145)	Activation server version too old.	This happens when a pre-v9 activation server is used with a v9+ client which encrypts the request. This can also happen if there is a proxy server installed at the user site.
RLM_EH_CANT_CREATE_REHOST (-151)	The rehostable hostid for this product could not be created.	Verify that the user has write access to either /var/tmp/{isvname} on unix/mac

Status return	Problem	Solution
		systems or “documents and settings/all users/application data/reprise” on Windows
RLM_EH_REHOST_TOP_DIR_EXISTS (-152)	The directory which contains a rehostable hostid for this product already exists, so it cannot be created.	Verify that this product has not already been activated with a rehostable hostid. If it has, it cannot be activated again on the same machine.
RLM_EH_REHOST_EXISTS (-153)	The rehostable hostid already exists for this product, so it cannot be created.	Verify that this product has not already been activated with a rehostable hostid. If it has, it cannot be activated again on the same machine.
RLM_EH_NO_FULFILLMENTS (-154)	No fulfillments exist to revoke	An attempt was made to revoke a license which was not activated by this server.
RLM_ACT_BAD_REDIRECT (-1013)	HTTP redirect does not specify hostname, or can't connect to specified hostname.	This is a webserver redirect error – no redirected hostname was specified, or we can't connect to the redirected hostname.
RLM_ACT_NO_PERMISSION (-1018)	HTTP server did not allow request to be processed.	This is a configuration error – your license generator cannot run on the webserver.
RLM_ACT_SERVER_ERROR (-1019)	HTTP request returned HTTP status 500 - INTERNAL_SERVER_ERROR	The activation server is not configured correctly. Look at the web server error log files to find the source of the error. Without this information, it is impossible to diagnose the error.
RLM_ACT_NO_AUTH_SUPPLIED (-1022)	Proxy requires authentication	Supply proxy credentials via the HTTP_PROXY_CREDENTIALS environment variable.
HTTP_PROXY_AUTH_FAILED (-1023)	Proxy credentials didn't work	Proxy requires authentication and the credentials supplied failed.
RLM_ACT_NO_BASIC_AUTH (-1024)	Proxy does not support BASIC authentication	RLM activation uses only basic authentication with proxy servers.

Appendix I – Refresh-type activations (deprecated)

Rehosting via Activation Refresh (deprecated)

Note: the REFRESH activation type has been deprecated in RLM v9.3. Please use the new rehostable hostid type if you wish to allow users to transfer licenses from one system to another.

Activation can be set up to support rehosting of licenses by the end user without ISV involvement. This involves the use of "refresh" licenses, which are normal RLM licenses, except they are typically issued for a very short duration - on the order of days. Refresh licenses are reactivated frequently, either by a generic utility supplied with RLM, or by a utility supplied by the ISV, using the RLM refresh API. Each time a refresh license is reactivated, it gets a new (later) expiration date, some number of days from the reactivation date specified by the ISV.

Rehost and Refresh Concepts

Note: the REFRESH activation type has been deprecated in RLM v9.3. Please use the new rehostable hostid type if you wish to allow users to transfer licenses from one system to another.

A rehostable license is one that is intended to be reactivated frequently and receive a new expiration date with each reactivation. Rehostable licenses are typically of very short duration. The main purpose of the refresh activation type is to support rehosting of licenses. Rehosting is when the end-user wants to move a license from one system to another. Traditionally, the ISV had no way of knowing if the end-user ceased running the application on the original system after issuing a license for the new system. With very short term rehostable licenses, the ISV is assured that the maximum period during which the end user can run the licensed application on both the old and the new systems is the (short) duration of the rehostable license. Refresh-type licenses must be node-locked, they can never be floating.

For example, if the ISV specifies that the rehostable license for a product should expire 4 days after activation, a rehost of that product will effectively shut off use of the product a maximum of 4 days after the rehost. If the refresh operation at the end user site is run daily and it fails for some reason on a given day, the license is still good for another 3 days - enough time to resolve the refresh failure.

Rehostable licenses also give the ISV a way to revoke a license granted to an end-user should that user fail to meet contractual obligations, for example. The ISV can simply disable refresh of the user's license on the activation server, and refresh attempts of that license will fail from that point forward.

The client side of refreshing is automated, so it can be performed daily without an undue burden on the end-user. Reprise supplies a "Refresh API" for license refreshing, which the ISV can use in a custom-developed refresh utility. Reprise also supplies a generic refresh utility which can be supplied to the end-user by the ISV. The generic utility, "refresh_util", is meant to be set up to run daily as a Scheduled Task on Windows or as a cron job on Unix.

Rehostable Licenses

A license is rehostable if the activation key that was used to activate it is a refresh type activation key. This is set in the rlc page that creates activation keys. The license generator does a couple of special things when it activates refresh licenses:

- It includes the optional attribute "_refresh_url". This is the URL that the refresh utility contacts to refresh the license, that is, the ISV's activation server. This is specified by the ISV in rlc, in the "Setup RLC" section. The activation server includes the refresh URL in every refresh license it creates.
- It includes the optional attribute "_refresh_id", which is the activation key for refreshing the license. This is also used by the activation utility.
- It includes a the options attributes "start_date" and "replace", which are used internally by the rlm library to obsolete previous instances of the same license.

NOTE: At the time of this release (v7.0), the refresh utility supports only nodelocked, uncounted licenses.

Refresh Utility

Reprise supplies both a stand-alone utility (isv_refresh) that can be used as-is at the end-user site to refresh licenses, and an API (Appendix A – Activation API on page 37) around which the ISV may build a custom refresh utility.

rlmutil rlmrefresh takes an argument specifying the directory in which it should process rehostable licenses, and an optional argument specifying the name of an ISV whose licenses are to be refreshed. Without the ISV argument isv_refresh processes rehostable licenses from all ISVs in the specified directory. It prints details about any errors it encounters while refreshing the licenses, unless the optional -q argument is specified. On Windows a batch script can be set up to supply arguments and also to redirect the output to a file. It can be run manually, but it is intended to be run automatically as a cron job on Unix or a Scheduled Task on Windows. The norm is to run it once a day.

In order to avoid all end-users refreshing their licenses at the same time every day and overloading the ISV's activation server, the ISV can request that each end-user run the refresh utility at an assigned time every day. A custom refresh utility could randomize the refresh time.

Rehosting

Rehosting a license with refresh is a matter of turning off the refreshing of it on the original host, and activating it on the new host. Disable refresh by disabling the refresh of the directory containing the license if refresh_util is being used. Then activate the license on the new host, and enable refreshing of the license on the new host.

Product Definition and Activation Key Creation for Rehostable Licenses

- Define the refresh URL in rlc. Select "Setup RLC" from the menu on the left, and set the URL of the Activation server to contact for refreshes (your activation server name, eg, www.reprisesoftware.com). You only have to do this step one time.

- Define the product in rlc. The product must be defined as a nodelocked license (as of RLM v7.0). Choose a short duration for the license in the expiration date field. You do this each time you set up a rehostable product.
- Create Activation Keys for the product in rlc. Under "Activation Type", choose "Refresh". Set the number of rehosts allowed to the quantity desired. You create one activation key for each customer who has purchased this product.

Appendix J – Redirecting Your Activation Website

When you use RLM activation, your application contains the URL of the activation server. If you are running the activation server at your own website, it may be fine to hard-code this URL into your software.

However, if you think that there may be a time when you might want to move the website, there are a couple of options you can employ to make this change more easily:

1. you could create some kind of configuration which your software reads at runtime to decide which URL to use to activate, or
2. You can use the ISVNAME_ACT_URL environment variable to override the activation URL in the rlm_activate() call (see the description of rlm_activate() in the Reference Manual), or
3. you can set up your URL to redirect to a different (presumably newer) URL.

If you use Reprise's hosted activation service, Reprise strongly recommends that you employ one of these techniques. Obviously, the first two techniques are more flexible, because they would allow you to continue operating even if the original URL goes away completely. This is what Reprise Software recommends.

If you wish to redirect your URL, you can do this with an .htaccess file if you are using a linux server with Apache:

Set up a new domain which you use as your activation URL and redirect the whole domain to the actual activation domain. To do this, follow these steps:

- Create the domain (usually a subdomain, e.g. activation.yourcompany.com).
- In the top-level of the domain, create a .htaccess file with the following contents:

```
Options +FollowSymLinks
RewriteEngine on
RewriteRule (.*) http://www.actual-activation-domain.com/$1 [R=301,L]
```

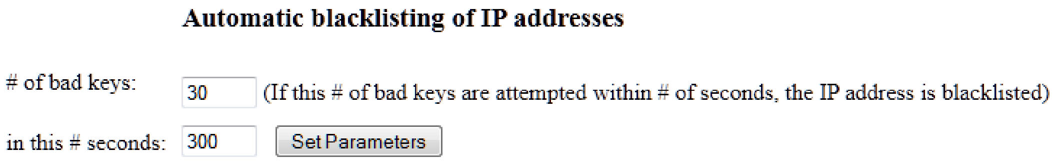
(Note: Change “actual-activation-domain” to your domain name. Also, this .htaccess method of redirection works ONLY on Linux servers having the Apache Mod-Rewrite module enabled.)

Appendix K – Automatically blacklisting IP addresses

If someone writes a program to guess activation keys, they will present a large number of invalid activation keys to the server in a short period of time. RLM Activation Pro v9.4 has the capability to automatically blacklist this IP address based on parameters you specify.

The first time an invalid activation key is presented to the license generator by a particular IP address, the license generator logs the time. After this, if N bad keys are presented within a specified time interval, the IP address is automatically blacklisted. Once this time interval passes without N bad keys being attempted, the whole process resets (in other words, N bad attempts will be required before the IP address is blacklisted).

By default, the interval is 600 seconds (10 minutes) and the number of bad keys which cause the IP address to be blacklisted is 40. However, you can change both of these parameters in the "Administer Database" menu available to admin users, near the bottom of the form:



Automatic blacklisting of IP addresses

# of bad keys: 30 (If this # of bad keys are attempted within # of seconds, the IP address is blacklisted)

in this # seconds: 300

In this example, 30 bad activation keys within 5 minutes will cause the IP address to be blacklisted. To set the parameters, enter the 2 parameters into the text boxes and press "Set Parameters".

Once blacklisted, the IP address will appear in the normal blacklist. The blacklisted IP address will have to match the IP address exactly, rather than the substring domainname match of normal blacklisted hosts.

Appendix L – How Activation Pro chooses a hostid

RLM Activation Pro will chose a hostid for the generated license based on the input from the RLM-licensed application.

If your application does not specify a hostid to the *rlm_act_request()* or *rlm_activate()* call, then the RLM client library supplies the standard set of hostids for the machine on which the software is running. Alternately, you can specify a hostid-list in the rlm_act_request() call, or via the *rlm_act_set_handle()* call if you use *rlm_activate()*.

rlm will only activate licenses with rehostable, non-zero RLM_HOSTID_32BIT, RLM_HOSTID_ETHER, RLMIDn, RLM_DISKSN, or ISV-defined hostids. Any other hostid will return an RLM_ACT_BAD_HOSTID_TYPE status from *rlm_act_request()*. (Note: ISV-defined hostids were added to the list of legal hostids in RLM v4.0). The priority is:

- If a rehostable hostid is present, it will be used.
- If an ISV-defined hostid is present, it will be used.
- If an rlmid hostid is present, it will be used.
- If a Disk Serial Number hostid is present, it will be used
- Otherwise, if an ethernet address is present, it will be used.
- Finally, if none of the hostids above are present in the list, the 32-bit hostid will be used.
- If none of the hostid types above are present, the activation software will return RLM_ACT_BAD_HOSTID_TYPE.

You can override RLM's notion of the hostid and specify an exact hostid to be used by the activation server by calling *rlm_act_set_handle()* with the RLM_ACT_HANDLE_HOSTID_LIST parameter (or passing the hostid-list parameter to *rlm_act_request()*). The *hostid_list* parameter can contain a list of hostids for use in nodelocked licenses. This is specified with the following syntax:

```
list:list-of-hostids
```

For example:

```
list:user=joe host=sam ip=192.16.7.23 3f902d8b0027
```

If a list is supplied, note the following:

- The activation software uses the hostids in the list as you specified, even if they are not "secure".
- If the license to be activated is a served license (floating), only the first hostid in the list is used.
- The number of available activations on the activation key is decremented by 1 regardless of the number of hostids in the license created.
- The hostid list must be less than RLM_ACT_MAX_HOSTID_LIST characters long (205) including the "list:" prefix.
- The hostid list can contain no more than RLM_MAX_HOSTID_LIST (25) hostids.

  This capability can be used to create a license which works on 2 (or more) systems, e.g. to create a license for a primary and a backup system. It can also be used to pass a hostid of a less secure type to be used, e.g. the *hostid-list* "list:ip=172.16.7.12" will cause the activation software to use the IP address as a hostid without returning RLM_ACT_BAD_HOSTID_TYPE.