R1Soft Licensing Portal API for Server Backup 5.0

This chapter explains how to use the Licensing API to get updates on licenses for the Server Backup 5.0 application. This includes querying for creating, returning, upgrading, enabling, disabling, obtaining, and deleting licenses and license pools.

XML-RPC Method Calls

XML-RPC is a Remote Procedure Calling protocol that works over the Internet. An XML-RPC message is an HTTP-POST request. The body of the request is in XML. A procedure executes on the server and the value it returns is also formatted in XML. Procedure parameters can be numbers, strings, etc. They can also be complex record and list structures.

Method Calls	Method Parameters
`licensing.createPooledLicense`	`struct createPooledLicense(string poolID, string product, Map features)`
`licensing.disableLicense`	`struct disableLicense(String licenseID)`
`licensing.enableLicense`	`struct enableLicense(String licenseID)`
`licensing.getLicense`	`struct getLicense(String licenseID)`
`licensing.getLicensePool`	`struct getLicensePool(String poolID)`
`licensing.getLicensePools`	• `array getLicensePools()` • `array getLicensePools(Map filter)`

`licensing.getLicenses`	• `array getLicenses()` • `array getLicenses(Map filter)` • `array getLicenses(String poolID)` • `array getLicenses(String poolID, Map filter)`
`licensing.returnPooledLicenseFeatures`	`struct returnPooledLicenseFeatures(Strin licenseID, Map features)`
`licensing.upgradePooledLicense`	`struct upgradePooledLicense(Strin licenseID, Map features)`
`licensing.deletePooledLicense`	`boolean deletePooledLicense(Strin poolID, String licenseID)`

Licensing API Examples
• 5.0 Licensing API Example - createPooledLicense • 5.0 Licensing API Example - deletePooledLicense • 5.0 Licensing API Example - disableLicense • 5.0 Licensing API Example - enableLicense • 5.0 Licensing API Example - getLicense • 5.0 Licensing API Example - getLicensePool • 5.0 Licensing API Example - getLicensePools • 5.0 Licensing API Example - returnPooledLicenseFeatures • 5.0 Licensing API Example - upgradePooledLicense

Method Parameter Descriptions

- Map features - Key/value map of features and their quantity value
- Map filter - Key/value map of filters and their value
- String poolID - ID (UUID) of the license pool you allocate Server Backup licenses from
- String licenseID - ID (UUID) of the Server Backup license key you created from a pool
- String product - The product type of the license (`Enterprise Edition (Win)`, `Enterprise Edition (Linux)`)

Many of the methods return Maps or take Maps as arguments. These Maps may have different key/value mappings depending on the context. The Maps basically represent different objects. These objects and their corresponding key/value mappings are described below.

Features Map | Filter Map | Product Map

Features Map

Name	Type	Description
Physical Servers	Integer	Number of physical servers
Virtual Servers	Integer	Number of virtual servers

Example of usage: `$VIRTUAL_SERVERS = '1'; $PHYSICAL_SERVERS = '1';`

Filter Map

Name	Type	Description
Type	String	Type of license ("Perpetual," "Heartbeat," "Trial," "Pooled License")
Enabled	Boolean	Whether or not license is enabled
Installed	Boolean	Whether or not license is installed
Server Description	String	Description of server on which license is installed
Product	String	Product type of the license

Product Map

Name	Type	Description
Enterprise Edition (Win)	String	Enterprise Edition Key for Windows
Enterprise Edition (Linux)	String	Enterprise Edition Key for Linux

Example of usage: `$LICENSETYPE = "Enterprise Edition (Win)";`