



MediaCP RPC Manual

RPC / API Documentation for the Media Control Panel

Table of Contents

RPC Overview	2
RPC Example – New User Account.....	2
RPC Example – New Shoutcast 198 Service.....	3
RPC Command & Function Listing.....	4
admin.service_create.....	4
admin.service_remove	5
admin.service_suspend	5
admin.service_unsuspend	5
admin.service_update	6
admin.user_create	7
admin.user_update.....	8
service.status	9
service.restart	9
service.start.....	9
service.stop	10
source.status.....	10
source.start	10
source.stop.....	11
source.restart.....	11
system.build.....	11



RPC Overview

The Media Control Panel is built with an advanced and simple **XML-RPC for PHP**.

To utilize the Media Control Panel RPC you will be required to download the PHP class from <http://scripts.incutio.com/xmlrpc/>

RPC Example – New User Account

This example will create a new user account (not server)

1. Download **XML-RPC for PHP** from <http://scripts.incutio.com/xmlrpc/>
2. Extract the **XMLRPC.class.php**
2. Create a new php file (for example) **createuser.php** with the following content

```
<?php
require_once("XMLRPC.class.php");

$url = 'http://yourdomain.com/mediapanel/'
$command = "admin.user_create.php";
$arguments = array(
    "auth" => "REPLACE_WITH_AUTH_KEY",
    "username" => "myuser@domain.com",
    "hash" => SHA1("myuser@domain.com" . "mypassword"),
    "user_email" => "myuser@domain.com"
);

$client = new IXR_Client($url, '/system/rpc.php');
$client->debug = false; /* Set to true if you have difficulties */

if ( !$client->query( $command, $arguments ) ) {
    die('An error occurred - ' . $client->getErrorCode() . ": " . $client->getErrorMessage());
}

$return = $client->getResponse();

if ( $return['status'] == 'success' ) {
    echo $command . ' executed successfully.';
} else {
    echo 'failed executing ' . $command;
}

/** Output debugging information */
print_r( $return );

?>
```



RPC Example – New Shoutcast 198 Service

This example will create a new shoutcast 198 service account with no source plugin.

1. Download **XML-RPC for PHP** from <http://phpxmlrpc.sourceforge.net/#download>
2. Extract the **XMLRPC.class.php**
3. Create a new php file (for example) **createuser.php** with the following content.
4. Be sure to replace crucial settings such as **auth** and **username**.

```
<?php
require_once("XMLRPC.class.php");

$url = 'http://yourdomain.com/mediapanel/'
$command = "admin.service_create.php";
$arguments = array(
    "auth" => "REPLACE_WITH_AUTH_KEY",
    "username" => "myuser@domain.com",
    "plugin" => "shoutcast198",
    "sourceplugin" => "", /** Service Source? **/
    "password" => "mypassword", /** Service Password **/
    "maxuser" => "25", /** Maximum Users **/
    "bitrate" => "48", /** Bitrate Limit in Kbps **/
    "bandwidth" => "10240", /** Transfer in MB **/
    "quota" => "1024", /** Disk Allowance in MB **/
    "system_id" => "1", /** Cluster/System **/
    "expire" => "0", /** Expiry UNIX Timestamp **/
);

$client = new IXR_Client($url. '/system/rpc.php');
$client->debug = false; /** Set to true if you have difficulties **/

if ( !$client->query( $command, $arguments) ) {
    die('An error occurred - '.$client->getErrorCode().": ".$client->getErrorMessage());
}

$return = $client->getResponse();

if ( $return['status'] == 'success' ) {
    echo $command. ' executed successfully.';
} else {
    echo 'failed executing '. $command;
}

/** Output debugging information **/
print_r( $return );

?>
```



RPC Command & Function Listing

admin.service_create

Description

Creates a streaming service through the MediaCP

Parameters

Param	Type	Description
username or userid	STRING or INT	The user account to create service upon.
plugin	STRING	The plugin to base the service upon, use the following examples or refer to your control panel new service page for more information. NoService, shoutcast198, shoutcast2, icecast, windowsMediaServices, wowzaMedia
source	STRING	A source plugin to use with the service, leave blank to ignore. ices04, ices20, sctransv1, sctransv2, streamtranscoderv3
unique_id	STRING	A unique string to use to identify the service. For example: myShoutcastServer1
portbase	INT	Base network port of service, set to "auto" to automatically select a port based on currently assigned in the MCP.
planid	INT	Apply settings directly from a plan created in the system.
SystemID	INT	A specific system or cluster to deploy the service upon. Set to "auto" to automatically determine available system.
rpc_extra	BOOL	If set to <i>true</i> will return an array of the whole serverData
maxuser	INT	Maximum connections to the service
bitrate	INT	Maximum uplink rate for service.
bandwidth	INT	Maximum Data Transfer per month in MB. Set to 0 for unlimited.
quota	INT	Disk space allocation for service. Default being 30. Set to 0 for unlimited.
proxy	STRING	Port 80 Proxy Support. "Enabled" or "Disabled"
messengercontrol	STRING	MSN Messenger Service Controller Support. "Enabled" or "Disabled"
expiry	INT	UNIX Timestamp of expiry date. Set to "0" to never expire.

Return values an array containing the following

status => success, id => {INT}, portbase => {INT}, error=>, restart=>{BOOL}



admin.service_remove

Description

Removes a service from the Media Control Panel

Parameters

Param	Type	Description
serverid or unique_id	STRING or INT	The service ID to be removed. The Unique ID of the service to be removed.

Return values an array containing the following

status => success, error=> {STRING}

admin.service_suspend

Description

Creates a streaming service through the MediaCP

Parameters

Param	Type	Description
ServerID or unique_id	STRING or INT	The service ID to be removed. The Unique ID of the service to be removed.
Reason	STRING	The reason for suspension.
Days	INT	Number of days to suspend service for.

Return values an array containing the following

status => success or failed, error=>, {STRING}

admin.service_unsuspend

Description

Unsuspects a service

Parameters

Param	Type	Description
serverid or unique_id	STRING or INT	The service ID to be removed. The Unique ID of the service to be removed.

Return values an array containing the following

status => success, error=> {STRING}



admin.service_update

Description

Creates a streaming service through the MediaCP

Parameters

Param	Type	Description
serverid or unique_id	STRING or INT	The service ID to be removed. The Unique ID of the service to be removed.
username or userid	STRING or INT	The user account to create service upon.
plugin	STRING	The plugin to base the service upon, use the following examples or refer to your control panel new service page for more information. NoService, shoutcast198, shoutcast2, icecast, windowsMediaServices, wowzaMedia
source	STRING	A source plugin to use with the service, leave blank to ignore. ices04, ices20, sctransv1, sctransv2, streamtranscoderv3
unique_id	STRING	A unique string to use to identify the service. For example: myShoutcastServer1
portbase	INT	Base network port of service, set to "auto" to automatically select a port based on currently assigned in the MCP.
planid	INT	Apply settings directly from a plan created in the system.
SystemID	INT	A specific system or cluster to deploy the service upon. Set to "auto" to automatically determine available system.
rpc_extra	BOOL	If set to <i>true</i> will return an array of the whole serverData
maxuser	INT	Maximum connections to the service
bitrate	INT	Maximum uplink rate for service.
bandwidth	INT	Maximum Data Transfer per month in MB. Set to 0 for unlimited.
quota	INT	Disk space allocation for service. Default being 30. Set to 0 for unlimited.
proxy	STRING	Port 80 Proxy Support. "Enabled" or "Disabled"
messengercontrol	STRING	MSN Messenger Service Controller Support. "Enabled" or "Disabled"
expiry	INT	UNIX Timestamp of expiry date. Set to "0" to never expire.

Return values an array containing the following

status => success, id => {INT}, error=>, restart=>{BOOL}



admin.user_create

Description

Creates a User account through the MediaCP

Parameters

Param	Type	Description
username	STRING	
hash	STRING	A SHA1 hash combination of the username and password. For example: <i>SHA1('username' . 'password')</i>
user_level	INT	Default 0 for customer. 0 Customer 1 Administrator 2 Super Administrator
user_email	STRING	Account email address
name	STRING	
surname	STRING	
address	STRING	
country	STRING	
state	STRING	
postalcode	STRING	
activated	INT	0 or 1
theme	STRING	Default MediaPanel
language	STRING	

Return values an array containing the following

status => success or failed, id => {INT} error=>, {STRING}



admin.user_update

Description

Updates a user account through the MediaCP

Parameters

Param	Type	Description
username or userid	STRING INT	
hash	STRING	A SHA1 hash combination of the username and password. For example: <i>SHA1('username' . 'password')</i>
user_level	INT	Default 0 for customer. 0 Customer 1 Administrator 2 Super Administrator
user_email	STRING	Account email address
name	STRING	
surname	STRING	
address	STRING	
country	STRING	
state	STRING	
postalcode	STRING	
activated	INT	0 or 1
theme	STRING	Default MediaPanel
language	STRING	

Return values an array containing the following

status => success or failed, id => {INT} error=>, {STRING}



service.status

Description

Retrieves the online status of the service.

Parameters

Param	Type	Description
ServerID or unique_id	STRING or INT	The service ID to be removed. The Unique ID of the service to be removed.

Return values an array containing the following

status => offline | online | suspended | expired | unconnectable | failed

** the status => failed is a result of invalid information provided.

service.restart

Description

Restarts a specific service

Parameters

Param	Type	Description
ServerID or unique_id	STRING or INT	The service ID to be removed. The Unique ID of the service to be removed.

Return values an array containing the following

status => success, error=> {STRING}

service.start

Description

Start a service.

Parameters

Param	Type	Description
ServerID or unique_id	STRING or INT	The service ID to be removed. The Unique ID of the service to be removed.

Return values an array containing the following

status => success, error=> {STRING}



service.stop

Description

Stops a service.

Parameters

Param	Type	Description
ServerID or unique_id	STRING or INT	The service ID to be removed. The Unique ID of the service to be removed.

Return values an array containing the following

status => success, error=> {STRING}

source.status

Description

Retrieves the status of source plugin associated with a service.

Parameters

Param	Type	Description
ServerID or unique_id	STRING or INT	The service ID to be removed. The Unique ID of the service to be removed.

Return values an array containing the following

status => success, error=> {STRING}

source.start

Description

Starts a source plugin associated with a service.

Parameters

Param	Type	Description
ServerID or unique_id	STRING or INT	The service ID to be removed. The Unique ID of the service to be removed.

Return values an array containing the following

status => success, error=> {STRING}



source.stop

Description

Stops a source plugin associated with a service.

Parameters

Param	Type	Description
ServerID or unique_id	STRING or INT	The service ID to be removed. The Unique ID of the service to be removed.

Return values an array containing the following

status => success, error=> {STRING}

source.restart

Description

Restarts a source plugin associated with a service.

Parameters

Param	Type	Description
ServerID or unique_id	STRING or INT	The service ID to be removed. The Unique ID of the service to be removed.

Return values an array containing the following

status => success, error=> {STRING}

system.build

Description

Retrieves the version build of the software

Parameters

N/A

Return values an array containing the following

status => success, build=> {STRING}