



## **Platform Documentation**

### **ProVision**

Application Version 4.1.15

#### **Covering:**

- **Resource Manager**
- **IP Address Manager**
- **DNS Manager**
- **DHCP Manager**

1. 6connect ProVision	3
1.1 Installation & System Requirements	3
1.1.1 Backup and Redundancy	4
1.1.2 CentOS Configuration Guide	5
1.1.3 General 6connect Installation Instructions	9
1.2 VM Install Guide - VMware Fusion	11
1.3 Getting Started	17
1.3.1 Working with Resources	18
1.3.1.1 Customizing Resource Types	19
1.3.1.2 Customizing Fields	21
1.3.1.3 Gadgets	22
1.3.1.3.1 XML Specifications	24
1.3.2 Admin Preferences	25
1.3.3 Authentication Options	27
1.3.3.1 LDAP Authentication	27
1.3.3.2 RADIUS Authentication	29
1.3.4 Permissions Overview	32
1.3.4.1 Global Permissions	33
1.3.4.2 Resource Permissions	34
1.3.4.3 Users and Groups	35
1.3.4.4 Verifying Permissions	36
1.3.5 IPAM Administration	37
1.3.5.1 Working with IP Blocks	38
1.3.5.2 IPAM Parameters	41
1.3.5.3 Holding Tank Management	43
1.3.5.4 LIR Management and Use	44
1.3.5.4.1 ARIN LIR Setup and Use	47
1.3.5.4.2 RIPE LIR Setup and Use	49
1.3.6 DNS Administration	51
1.3.6.1 Working with DNS Zones	54
1.3.6.1.1 Editing DNS Zones	55
1.3.6.2 Configuring ISC BIND Support	58
1.3.6.3 Configuring DynECT Support	58
1.3.6.4 Configuring PowerDNS Support	59
1.3.6.5 Configuring Secure64 Support	61
1.3.6.6 Configuring DNSSEC	66
1.3.6.7 Configuring Split Horizon/Views	67
1.3.6.8 Configuring DNS Templates	69
1.3.7 DHCP Administration	70
1.3.7.1 Configuring ISC dhcpd Support	72
1.3.8 Importing Your Data	73
1.3.8.1 Import Aggregate Blocks	75
1.3.8.2 Import DNS Zones	76
1.3.9 Feedback and Feature Requests	77
1.4 Tutorials	77
1.4.1 Common Tasks	78
1.4.2 UI Tours	78
1.5 FAQ	78
1.6 Previous Versions	79
2. CLI (Alpha)	79
3. API v1	81
3.1 1 - Overview	81
3.2 2 - Making API Requests	82
3.3 3 - SDK - PHP	82
3.4 API Module - Admin and Audit	83
3.4.1 Authentication Testing	83
3.4.2 Log Management	86
3.4.3 Zone Templates	89
3.5 API Module - DHCP	93
3.6 API Module - DNS	120
3.7 API Module - IPAM	161

3.8 API Module - LIR	177
3.9 API Module - Resource	180
4. Release Notes	191
4.1 ProVision 4.1.0	191
4.1.1 ProVision 4.1.1	195
4.1.2 ProVision 4.1.2	195
4.1.3 ProVision 4.1.3	195
4.1.4 ProVision 4.1.4	196
4.1.5 ProVision 4.1.5	196
4.1.6 ProVision 4.1.6	196
4.1.7 ProVision 4.1.7	197
4.1.8 ProVision 4.1.8	197
4.1.9 ProVision 4.1.9	198
4.1.10 ProVision 4.1.10	199
4.1.11 ProVision 4.1.11	200
4.1.12 ProVision 4.1.12	200
4.1.13 ProVision 4.1.13	201
4.1.14 ProVision 4.1.14	202
4.1.15 ProVision 4.1.15	202

# 6connect ProVision

6connect - ProVision

## Table of contents

- [Installation & System Requirements](#)
- [VM Install Guide - VMware Fusion](#)
- [Getting Started](#)
- [Tutorials](#)
- [FAQ](#)
- [Previous Versions](#)

## Installation & System Requirements

### 6connect Cloud Hosted Instance

6connect makes every effort to maintain broad compatibility across browser vendors and versions.

Web Browsers Supported:

- Firefox 6+
- Safari 4+
- Chrome 11+
- Internet Explorer 9+(IE 8 works, but there may be some display issues)

### 6connect Virtual Machine

The Virtual Machine has a console with additional information to assist with initial setup.

#### Host Environment:

The optimum resource mix will be based on page views/refreshes. A larger concurrent user base with constant editing may benefit from additional RAM.

The minimum recommended virtual environment is:

- Two processor cores
- 2GB RAM (4GB Recommended)
- 20GB Local storage (local SAS/SSD or iSCSI/FC LUN optional)
- VM format support for VMDK, OVF, OVA (Support for vSphere 5.x)

#### Software Environment:

Operating System: FreeBSD

#### Port Requirements:

Open outbound ports 443 and port 80

- cloud.6connect.com is used for license check
- checkip.dyndns.org validates the IP address of the machine to communicate with the licensing server

### 6connect Locally Hosted Instance

Initial application installation is included with the purchase of a license from 6connect. If modifications need to be made, we recommend contacting 6connect prior to any changes to ensure there is no negative impact to production systems or product functionality.

#### Hardware Requirements:

The optimum resource mix will be based on page views/refreshes. A larger concurrent user base with constant editing may benefit from additional RAM.

The minimum recommended hardware is:

- Dual-core Xeon class processor or equivalent (Quad-core Xeon Recommended)
- 2GB RAM (4GB Recommended)
- Local SATA storage (local SAS/SSD or iSCSI/FC LUN optional)
- Rack mount server chassis with redundant power supplies

\*Virtual instances are also acceptable. We have confirmed functionality with Citrix Xen Essentials, VMware, KVM, etc.

#### Software Requirements:

Operating System: Linux/BSD/OSX

Base Software Needed:

- Apache 2.x: <http://httpd.apache.org/>
- php 5.5.x: <http://php.net/downloads.php>
  - Plugin: Download Source Guardian extension from <http://www.sourceguardian.com/ixeds/> and install to php extensions directory.
- MySQL 5.6+: <http://www.mysql.com/downloads/>



#### MySQL Triggers

6connect does not support custom MySQL triggers at this time - please email [support@6connect.com](mailto:support@6connect.com) if you have any questions.

#### Port Requirements:

Open outbound ports 443 and port 80

- cloud.6connect.com is used for license check
- checkip.dyndns.org validates the IP address of the machine to communicate with the licensing server

## Backup and Redundancy

### Backup and Redundancy

You have several options for backup and redundancy depending on your implementation of your 6connect platform.

#### 6connect Hosted Instance

##### Backup Schedule

6connect backs up your data every hour with a 1 month retention policy. Backups are replicated post transaction flush to a local secondary server.

##### Restoration

Is a phone call or email away. We can spin up a new instance with your preferred data set.

#### Local/VM Instance

##### Backup Schedule

6connect backs up your local data to our cloud server every 48 hours with a 1 month retention policy. The backend of the application is MySQL, so it can be replicated to another server/instance or even tied into your own backup storage infrastructure.

##### Restoration

Is a phone or email away. We can spin up a new instance with your preferred data set, or send you a link to download your database. Optionally, we can even help you set it up and import your data to your new instance or assist with redundant configuration options depending on your RPO/RTO guidelines.



#### Backup your Data

For local customers, you should be backing up the following items:

mysqldump

And system folders off the 6connect root:

/scans

/zones

/keys

```
/archive
```

```
/data
```

## CentOS Configuration Guide

### Install and Configure MySQL

MySQL is included with most CentOS installs, check for it with:

```
yum list installed | grep mysql
```

If its not installed:

```
yum install mysql-server
```



#### Service Startup

Please ensure that the MySQL service has been started after you have installed it!

Set the mysql root password.

```
mysql
\u mysql
SET PASSWORD FOR 'root'@'%' = PASSWORD('newpass');
CREATE USER 'ipam'@'localhost' IDENTIFIED BY 'somelongpassword';
FLUSH PRIVILEGES;
```

Make sure to set both passwords to a minimum of 12 characters with some numbers and punctuation. The default my.cnf is fine for most clients. For large datasets through, the my.cnf will need to be tuned. [Insert tuning guide]

### Install and Configure PHP

PHP is usually included with most CentOS installs too, check for it with:

```
yum list installed | grep php
```

You should see something like php53.x86\_64, php53-mysql.x86\_64, php53-cli.x86\_64 listed. If not:

```
yum install php php-mysql
```



#### PLEASE INSTALL

Depending on your installation - you also need to confirm that **expect** and **unzip** are installed and enabled.

### Install PCNTL

```
yum install php-pcntl
```

## Install and Configure Apache and SSL



### mod\_rewrite REQUIRED

Please note that mod\_rewrite is required! If it is not enabled in Apache, key elements will not work as expected.

If SSL support is not already installed, install it:

```
yum install mod_ssl openssl
```

Generate private key, CSR, and temporary key if one hasn't been provided.

```
openssl genrsa -out ca.key 1024
openssl req -new -key ca.key -out ca.csr
openssl x509 -req -days 365 -in ca.csr -signkey ca.key -out ca.crt
```

Copy the files to the correct locations

```
cp ca.crt /etc/pki/tls/certs
cp ca.key /etc/pki/tls/private/ca.key
cp ca.csr /etc/pki/tls/private/ca.csr
```



Make sure that you copy the files and do not move them if SELinux is enabled (which it is by default)

Edit the apache ssl config and put in the appropriate options:

```
vi /etc/httpd/conf.d/ssl.conf
```

Change - SSLCertificateFile /etc/pki/tls/certs/ca.crt

Change - SSLCertificateKeyFile /etc/pki/tls/private/ca.key

```
/etc/init.d/httpd restart
```

Add 443 virtual hosts as needed in httpd.conf.

## Install and Configure Source Guardian

Download the extensions from <http://www.sourceguardian.com/ixeds/>. Choose either Linux 32 or Linux 64 .tar.gz depending on architecture.

```
tar -xvzf ixedX.xxx.tar.gz /tmp
```

**i** In the new ixed dir in /tmp, there will be many different files. The naming convention is as follows:

- ixed.5.3.lin - for all PHP 5.3.x versions
- ixed-5.0.1.lin - for PHP 5.0.1 only
- ixed.5.3ts.lin - the thread safe version for all PHP 4.3.x versions

Create an extension directory somewhere if there isn't one (/var/www/ext).

```
vi /etc/php.ini
```

Add - extension=/var/www/ext/ixed.5.3.lin

```
/etc/rc.d/init.d/httpd restart
```

## Configure SELinux



### RE-IP WARNING

Please remember - if you change the IP address of the your server, then you will need to update SELinux functions accordingly

Most CentOS install have SELinux enabled by default. One of its protections is to not allow httpd daemon to make network connections, we need to disable this for license checks.

To view the SELinux configuration for http:

```
/usr/sbin/getsebool -a | grep httpd
```

To turn protection off for the httpd daemon for creating network connections:

```
/usr/sbin/setsebool -P httpd_can_network_connect 1
```

## Configure IPTables

IPTables is enabled by default on CentOS. Add a new rule to allow 443 from anywhere. Make sure that this rule is in the chain BEFORE any blanket reject rule:

To list all current IPTable rules:

```
/etc/rc.d/init.d/iptables status
```

To add a rule for 443:

```
/sbin/iptables RH-Firewall-1-INPUT -I 5 -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
```



The -I 5 is what adds the rule to the 5th chain position. You might need to change this depending on existing rules. Look at what rules are there before running.

To save the new config:

```
/etc/rc.d/init.d/iptables save
```

OR (some versions of CentOS have different iptables names, so the above won't work)

```
vi /etc/sysconfig/iptables
```

With the file open for editing, add:

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
```

Once complete - restart the iptables service:

```
/etc/init.d/iptables restart
```



Customers can alter this post install to allow only their IP space, plus the 6connect management space.

## Install Radius

This section only needs to be followed if the customer will be using Radius for authentication. **If pear is not installed, install pear first.** Otherwise, just install the radius extensions:

```
yum install php-pear
pecl install radius
vi /etc/php.ini
Add - extension=radius.so
```

## Install 6connect

Get the latest build (or specific versioned build) from the repository and copy it to the destination. All local installs with -local (full version), or local-free (free version).

```
tar -xvf 6connectSP-xxx-local.tar /var/www/html
```

Will place all files in /var/www/html/IPAM.

Follow the instructions for 6connect installation.

Suggested follow up items for all customers

Tune IPTables rules to allow only your IP space and 67.221.240.0/24 and 2607:FAEO::/36 on ports 22 and 443.

Follow the CIS hardening guide to remove unused applications and their listening ports.

Use a valid SSL certificate if a self signed certificate was generated.

## General 6connect Installation Instructions

### Apache Configuration Requirements

 mod\_rewrite and mod\_ssl are required

ProVision must be run over SSL. Self signed certificates are fine.

 **ssl.conf**

Please note that if SSL is being used, the directory information will need to be present in the ssl.conf as well (location/file name may be different depending on the OS and Apache version)

The web root directory for ProVision must be configured with the following directives:

```
<Directory /<ProVision webroot>>
    Options FollowSymLinks
    AllowOverride All
    Order allow,deny
    Allow from all
</Directory>
```

### MySQL Configuration

```
SET GLOBAL sql_mode='STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION';
SET SESSION sql_mode='STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION';
```

## PHP Configuration



### PHP Compatibility

Please note that as of March 1, 2014, PHP versions <5.5 will no longer be supported

```
display_errors = Off
session.save_handler = files
session.save_path = "/tmp"
```

The session save path can be configured for alternate directories, however, you might need to manually add the folder "imports" and chown/chmod it to be readable and writeable by the web user. The software will try to do this automatically, but permissions could prevent it from being added correctly. This must be configured to import data.

On new versions of PHP, the following may need to be added:

```
session.bug_compat_warn=0
```

SMTP = localhost

smtp\_port = 25



Depending on the OS, the following may need to be added after various php extensions are added:

```
extension=radius.so
```

```
extension=ssh2.so
```

## Source Guardian

php extension - download from <http://www.sourceguardian.com/ixeds/>

```
extension=ixed.5.x.xxx
```

PHP cli binary path must be set in the software Admin section if different from default. By default it is /usr/bin/php.

## Additional PHP Extensions

See configTest.php located in the 6connect tar file for an updated list

## Additional System Packages

memcache

memcached

openssl

cURL

nmap

sendmail (Or any mail software. The correct binary should be specified in php.ini)

## DNS Tools and Packages

named-checkzone  
rndc  
zonesigner  
dnssec-dsfromkey

## NEW INSTALLATION

-----  
1. Install all the packages, extensions, and perform configuration listed above and the Source Guardian extension. To install the Source Guardian extension:

Download the correct Source Guardian loader for your OS/php version from: <http://www.sourceguardian.com/ixeds/>

Place the file in your php extension directory as specified in your php.ini

Add extension=ixed.x.x.y.y to your php.ini

2. Move the tar file in 6connect web root.

```
tar -xof productionBuild-4.0.3.tar
```

This will place all the new files into your web root directory.

3. Go to <http://<web root>/configTest.php>. If there are any configuration errors listed in red, other than in the Database and Configuration Files section, they must be corrected.

4. Click on Setup Wizard or go to <http://<web root>/configBootstrap.php>. Fill in the requested information. Permissions on your web directory may cause the automatic setup of directories to fail. If there are still permissions issues, run configDir.sh from the command line. If there are any other errors listed, follow the instructions to complete these items manually.

5. Carefully note the login credentials provided before continuing.

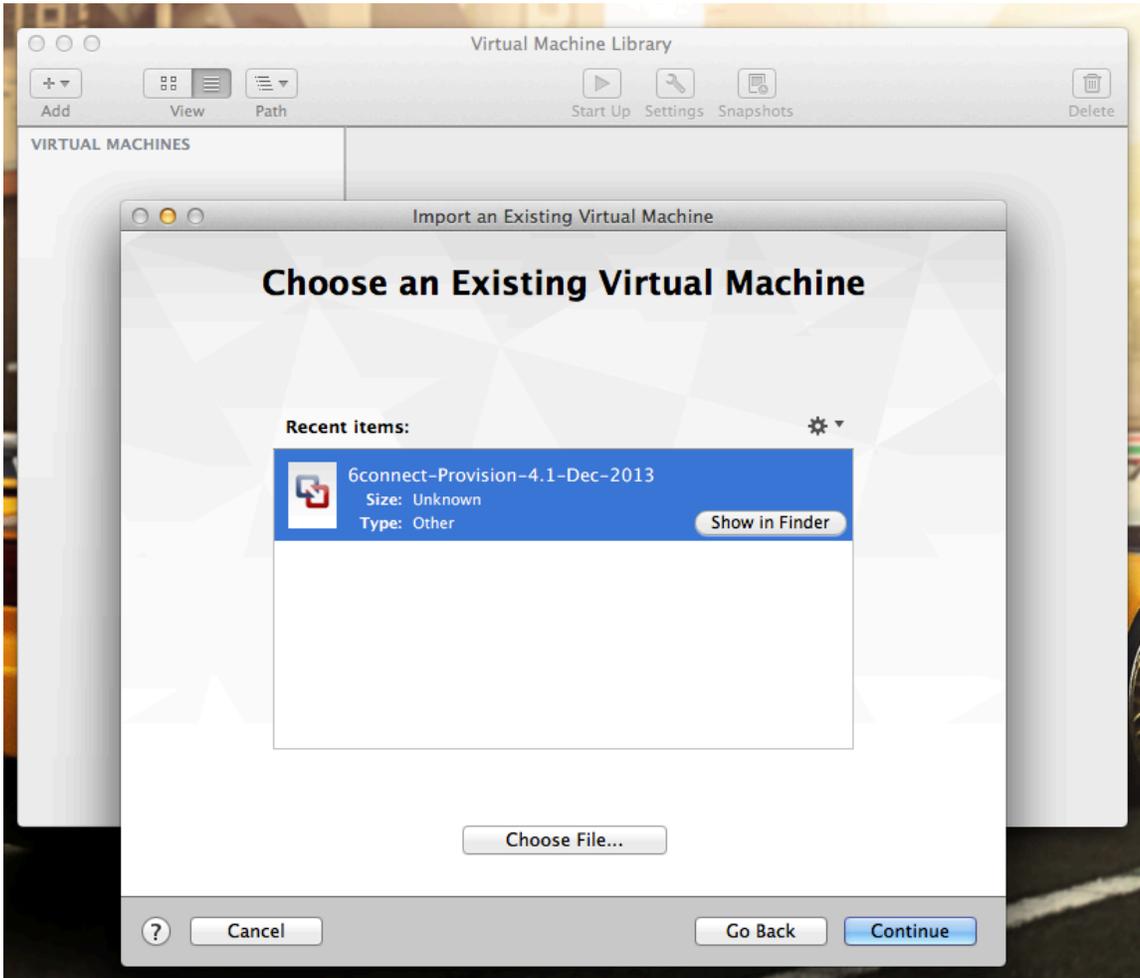
6. Log in and use!

## VM Install Guide - VMware Fusion

### Step 1 - Download the OVF\* tar file

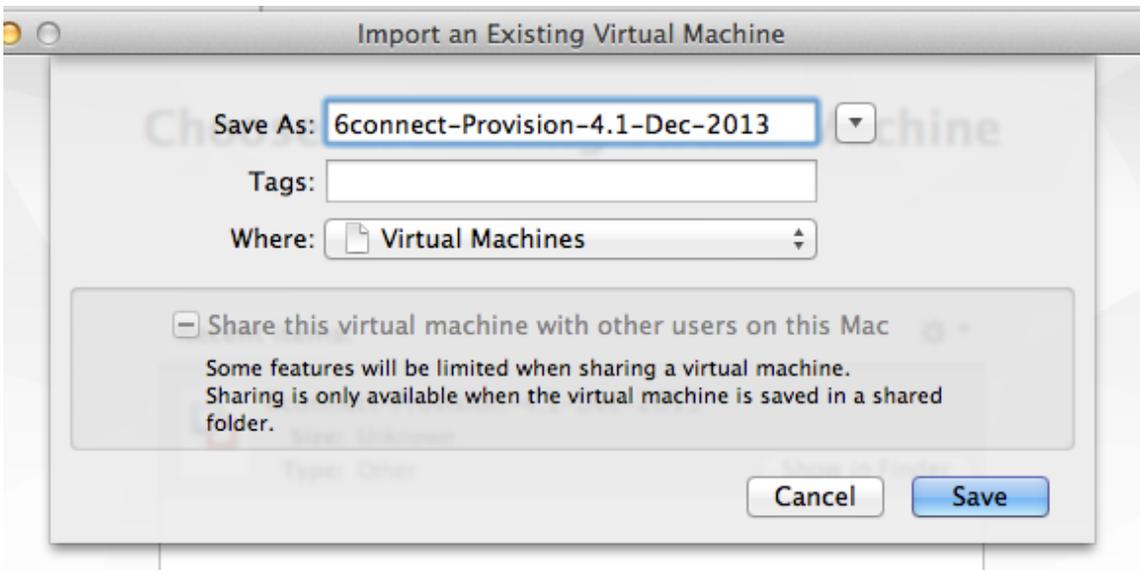
(VMDK is also available)

Select the option to Import an Existing Virtual Machine



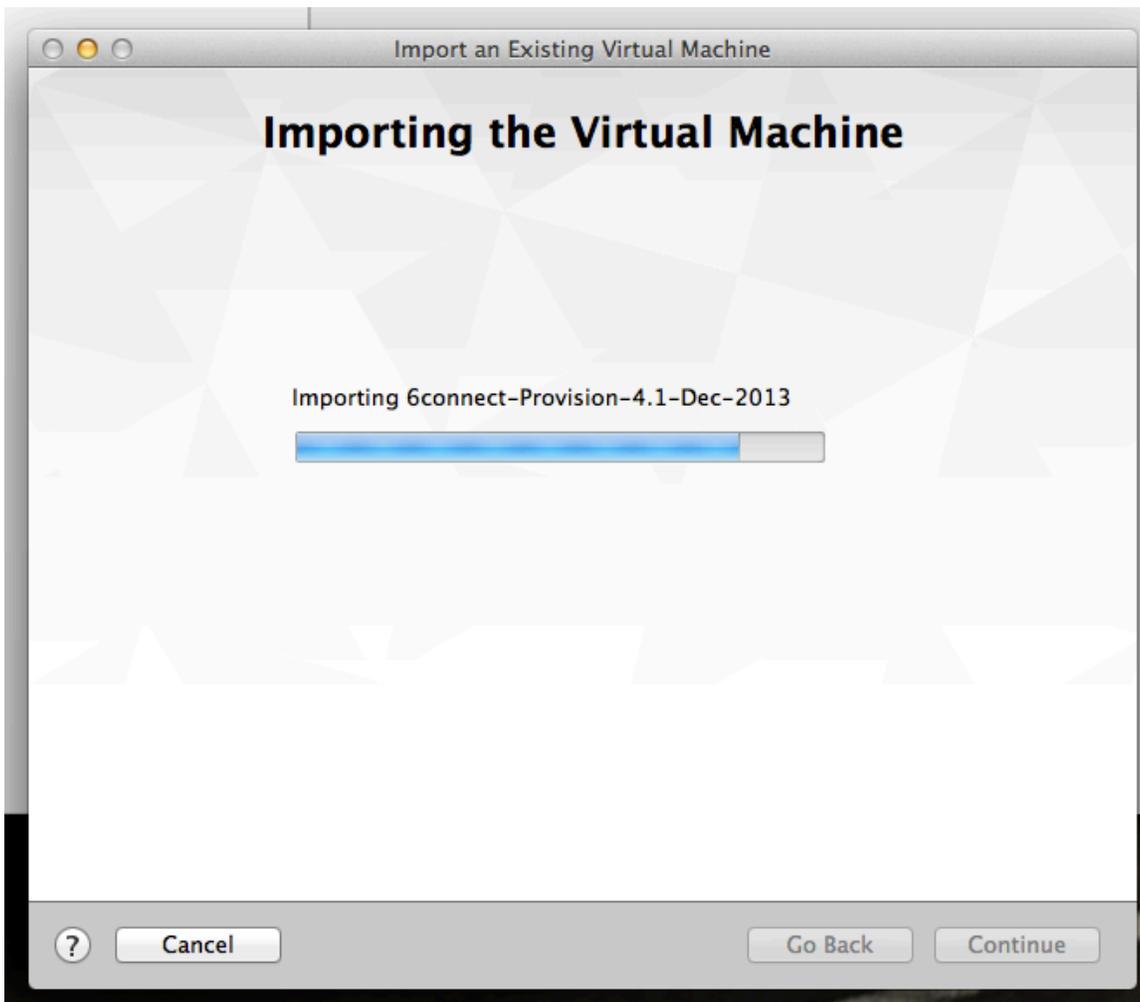
## Step 2 - Select the VM you wish to import

Un-tar the compressed file and select the OVF file under the Choose File dialog box.



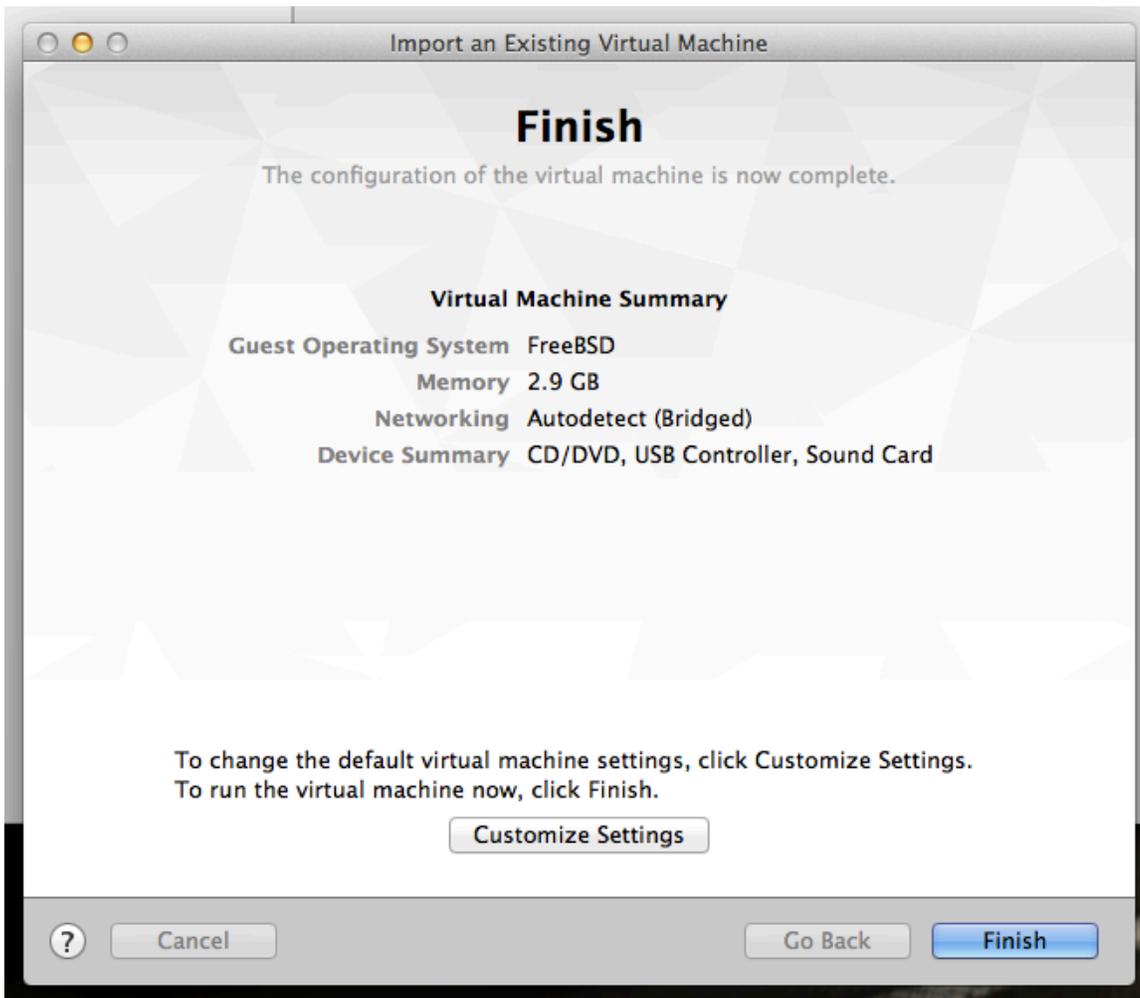
## Step 3 - Import the VM image

This will take a few minutes.



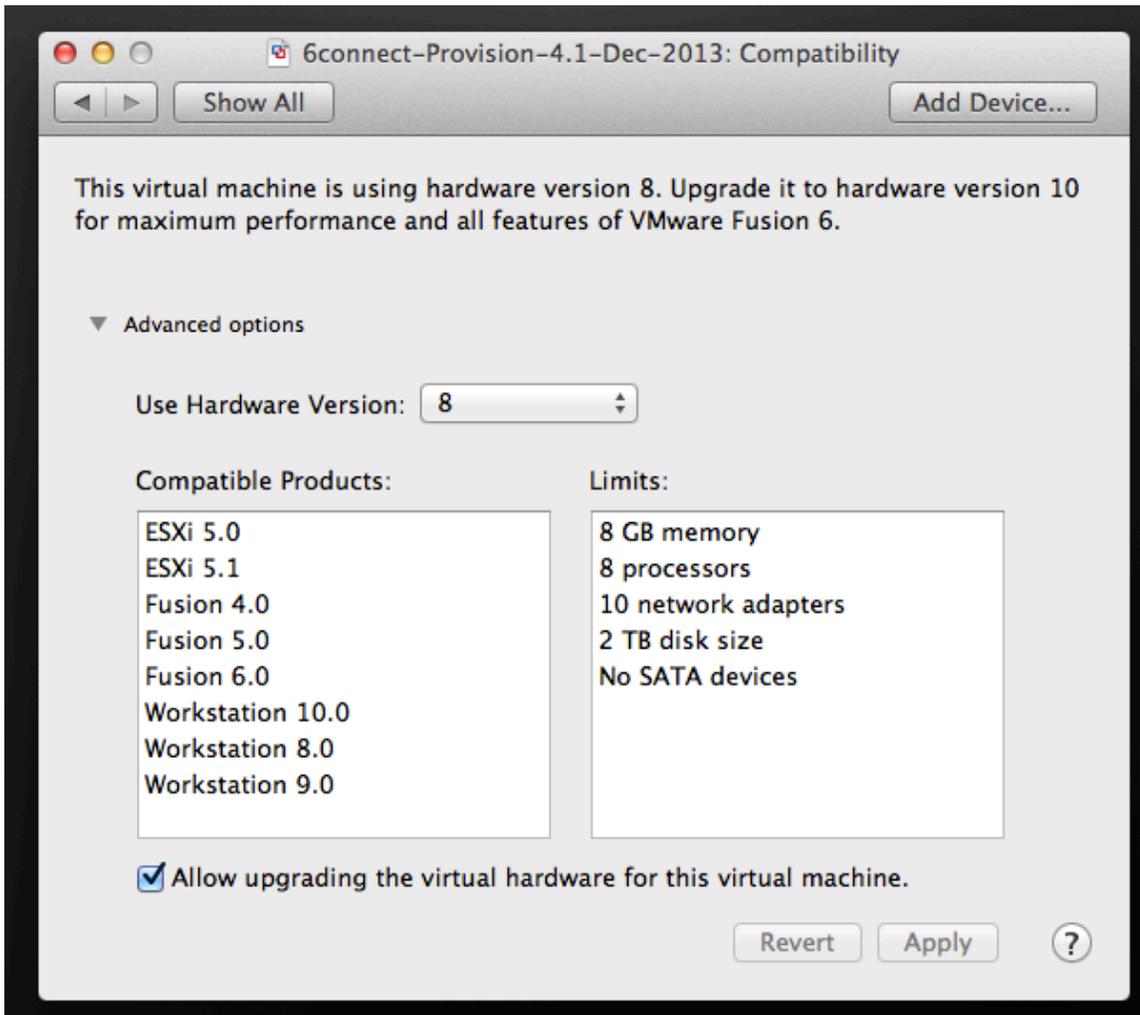
#### **Step 4 - Confirm the VM import configuration**

When completed, you will be prompted with basic information for the setup of the imported VM (network settings, etc.).



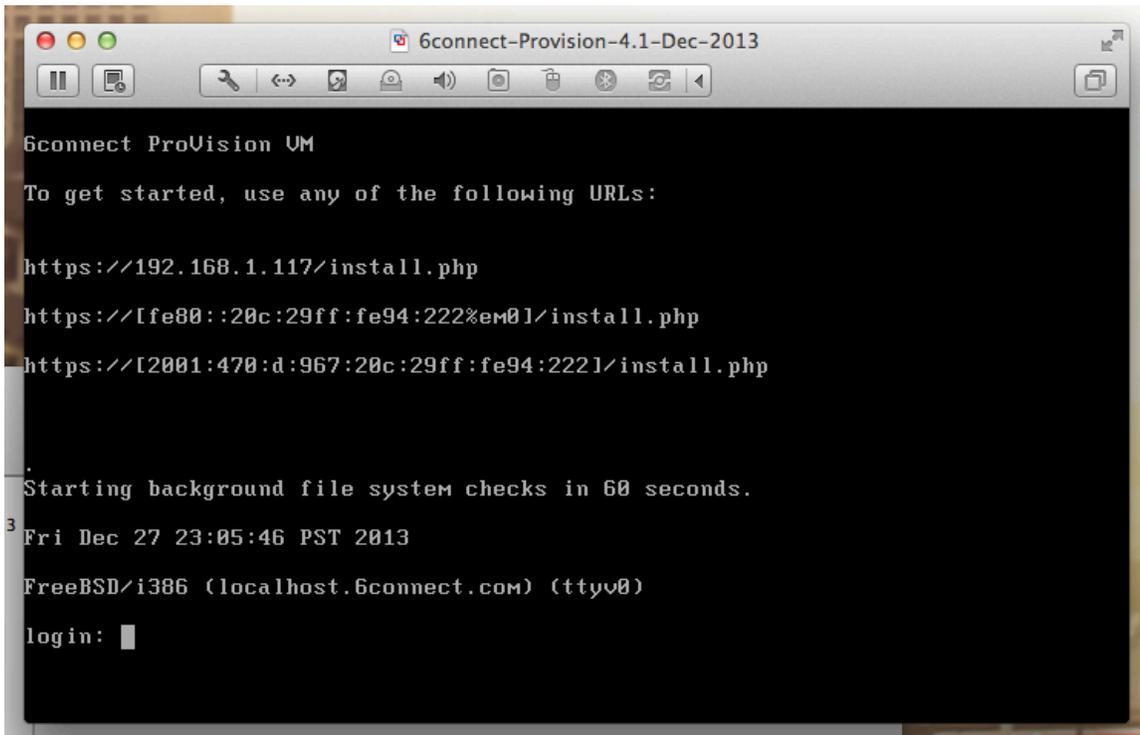
### Step 5 - Optional - Modify VM configuration properties

If needed - you can modify properties for the VM via the usual VMware Fusion settings. Please note that the default configuration for the VM is using hardware version 8 to ensure compatibility with the widest range of VM environments. You can modify/upgrade the VM if you wish to a newer hardware version if needed depending on your production operating environment.



## Step 6 - Setup ProVision environment

When you first boot the VM - you will be prompted with a console display showing the boot process. Once complete, you will be shown URLs to complete the installation process of ProVision software on the VM.



```
6connect ProVision VM
To get started, use any of the following URLs:

https://192.168.1.117/install.php
https://[fe80::20c:29ff:fe94:222%em0]/install.php
https://[2001:470:d:967:20c:29ff:fe94:222]/install.php

Starting background file system checks in 60 seconds.
3 Fri Dec 27 23:05:46 PST 2013
FreeBSD/i386 (localhost.6connect.com) (ttyv0)
login: █
```

## Step 7 - Run ProVision Setup Wizard

The Setup Wizard will confirm that everything in the VM is setup correctly. If there are any issues - please visit [\\$URL/configTest.php](#) - this will run a series of diagnostics to see if there are any configuration issues that should be addressed.

On this screen - enter your Company Name and Registration Email. The Setup Wizard will use the Company Name as a default and will use the registration email to verify the ProVision license and send login credentials.

Press the "Create Defaults" button for the install process to complete.

## Setup Wizard - 4.1.8

Please enter the information below. All of the 6connect specific configuration will be completed. Any other system configuration items with php or mysql need to be complete and verified via the configTest page. The config files and a new default database schema will be loaded configured with the information provided.

### Database and Configuration Files

#### MySQL Configuration

MySQL Found  
MySQL Version Version requirements met.

#### Required Information

##### Database Information

Database User   
Database Password   
Database Name   
Database Host

##### General Information

Company Name   
Registration Email   
Install URL   
Install Directory

[Create Defaults](#)

### Directory/File Ownersip and Permissions

/usr/local/www/apache22/data/ProVision/data	File read successful.
/usr/local/www/apache22/data/ProVision/zones	File read successful.
/usr/local/www/apache22/data/ProVision/keys	File read successful.
/usr/local/www/apache22/data/ProVision/archive	File read successful.
/usr/local/www/apache22/data/ProVision/scans	File read successful.
/usr/local/www/apache22/data/ProVision/data/globals.php	File read successful.

A new globals.php file will be written when creating defaults.

## Step 8 - Use ProVision

Once Setup is complete, you will get the following screen. Here you can find the URL to use your ProVision instance as well as the administrator login credentials.

## Setup Complete!

Please login to <https://192.168.1.117/ProVision>

## Getting Started

### Getting Started

You have got 6connect ProVision and now it's time to set it up! For setup assistance or additional information, you can contact our [Support](#) team at support@6connect.com.

We have broken down our [Getting Started](#) documents into the following steps and will have video tutorials where possible. You can also browse the Tutorials on the [Tutorials](#) page.

[Step 1 - Resources Overview](#)

[Step 2 - Admin Preferences](#)

[Step 3 - User & Group Permissions](#)

[Step 4 - IPAM Administration](#)

[Step 5 - DNS Administration](#)

[Step 6 - DHCP Administration](#)

[Step 7 - Importing Data](#)

## Working with Resources

### What is a Resource?

The "Resource" system is tied to the Permissions structure. What this means is that you get granular control on a resource level and can create groups around a single resource or even groups of resources. Since Resources can inherit permissions from others - it can be an easy way to categorize generic objects.



#### WARNING!

There are key Resources that are used by the System that should not be deleted. We have put in some safeguards in the UI, but the API can delete these resources if prompted. The resources that you should not remove are "Holding" and "Reverse". The "Available" Resource can be renamed - simply not deleted.

### How to Work with Resources?

The Resource is an entity that users can assign Network Resources to (IP blocks, hosts, DNS zones, etc.). You can also create hierarchies between resources which allows you to leverage permissions to control who can view and interact with any given resource and its assigned elements. Please note that you can also have Resources that do NOT have anything assigned to them regarding Network Resources. The result of this flexible architecture is that you can work with Resources in three ways:

- **Resource Entries:** These are the actual Resource names. When you click the "Add Entry" button you can customize various elements of the entry and assign the Parent Resource, Type and Category from their respective dropdown menus. This will pull up the field set for the Type and allow you to enter the data for the given Entry.

**Resources / Entries / Add Entry**

**Fundamentals**

Name (required)  
Some Resource

Parent  
TLR

**Type**  
Resource Holder

- Contact
- Firewall
- LIR
- Migrated Asset Data
- Migrated Device Data
- Physical Interface
- Rack
- Resource Holder
- Router
- Scanlet Result
- Server

- **Resource Types:** These can be anything from "customers" to "firewalls" to "cross-connects". Since you can customize the fields for these elements, and assign them to a Parent Type, you have flexibility in organizing the data. Check out [Customizing Resource Types](#) and [Customizing Fields](#) for more details on how to fit these elements to your business.

**Resources / Type** Add Type

Search Resources...

Name	Entries	Category
<a href="#">Contact</a>	81	<a href="#">Uncategorized</a>
<a href="#">Firewall</a>	2	<a href="#">Uncategorized</a>
<a href="#">LIP</a>	4	<a href="#">Uncategorized</a>
<a href="#">Physical Interface</a>	0	<a href="#">Uncategorized</a>
<a href="#">Rack</a>	2	<a href="#">Uncategorized</a>
<a href="#">Resource Holder</a>	831	<a href="#">Customer</a>
<a href="#">Router</a>	3	<a href="#">Uncategorized</a>
<a href="#">Scanlet Result</a>	0	<a href="#">Uncategorized</a>
<a href="#">Server</a>	7	<a href="#">Uncategorized</a>
<a href="#">Storage Array</a>	2	<a href="#">Storage</a>
<a href="#">Storage Controller</a>	2	<a href="#">Storage</a>
<a href="#">Switch</a>	1	<a href="#">Uncategorized</a>
<a href="#">Virtual Interface</a>	0	<a href="#">Uncategorized</a>
<a href="#">Virtual Machine</a>	3	<a href="#">Uncategorized</a>

- Resource Categories:** Categories can be used to create some filtered views for given Resources and Types. For example, you can create a Resource of Type "Resource Holder" and then assign a Category "Customer". Then you can view a list of Resources that have been assigned to Category "Customer". In the same way, you could also assign a Resource of Type "Router" under the Parent Resource "Corporate Datacenter" and then assign a Category "Infrastructure".

**Resources / Category** Add Category

Search Resources...

Name	Type
<a href="#">Corporate IT</a>	category
<a href="#">Customer</a>	category
<a href="#">Infrastructure</a>	category
<a href="#">Storage</a>	category
<a href="#">VM Infrastructure</a>	category

Want customize Resource Types? Check out [Customizing Resource Types](#) and [Customizing Fields](#) for more details!

Some examples:

- 1) Service Provider
- 2) Managed Service Provider
- 3) Datacenter/Colocation Provider
- 4) Enterprise

## Customizing Resource Types

video overview

Yes! You can create as many Resource Types as you wish (Firewall, Server, VM, Virtual Interface, etc.) and customize the fields that you care about for each Type. For example, you may not need to track the console port for your virtual firewall, so you would simply not use that field for the "Virtual Firewall" Resource Type. This way you can still track the console port for your physical firewalls like normal.

### Step 1: Create a New Resource Type

Create a new **Resource Type** from the **Types** sub-tab under the **Resources** Tab

**Fundamentals**

Name (required)  
My Custom Resource Type

Parent: TLR      Category: Infrastructure

[Create](#)

### Step 2: Add a Custom Field to a Resource Type

Customize the [Customizing Fields](#) for your **Resource Type**. Now you can add New [Customizing Fields](#) of different types (text, dropdown, text area) and also use any existing fields that are available. See the [Customizing Fields](#) page for more details. To add Fields to your Resource Type, just drag and drop the fields.

**Text Input**

**Text Area**

**Choice Box**

**Radios**

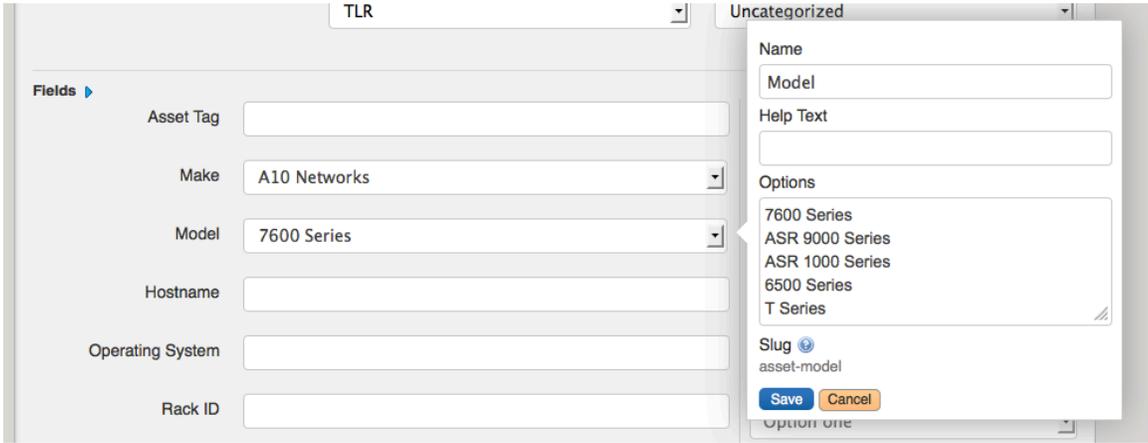
- Option one
- Option two

**Checkboxes**

- Option one
- Option two

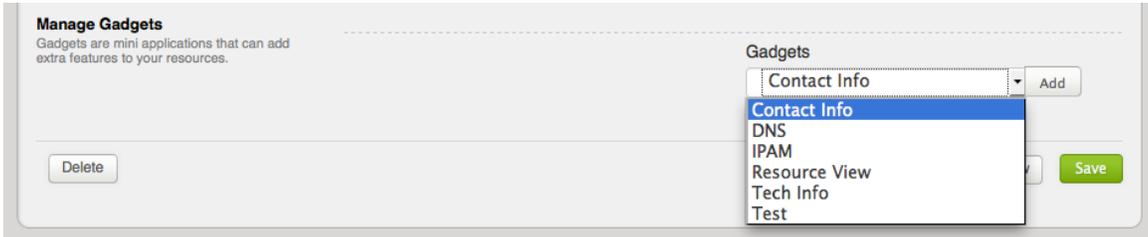
### Step 3: Edit Customize Field data

Select the field and you will get an editing window to modify the parameters of the field.



**Step 4: Add Gadgets to your Resource Type**

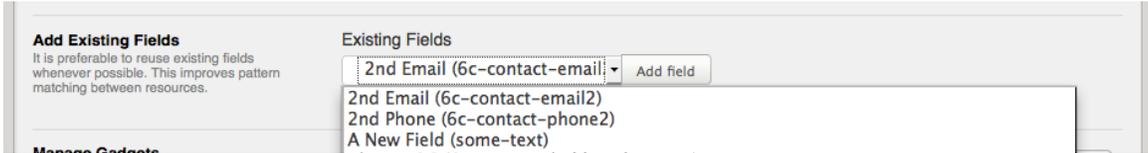
You will notice on this customization screen, you also have an area for **Gadgets**. Gadgets are areas of additional functionality that can be added to the UI of a given Resource.



**Customizing Fields**

**Creating Fields**

To add an Existing Field to a Resource Type, simply select the Field Name from the dropdown menu and click on the "Add Field" button.



To add a New Field to a Resource Type, simply drag the field over to the field list.

## Editing/Removing Fields

Once fields are added to a Resource Type, you can click on the field to make any additional changes to the fields.

To Remove a field, simply drag the field back to the right side of the screen and follow the prompt.

## Gadgets

## Gadgets

- [What are Gadgets](#)
- [Available Gadgets](#)
  - [Resource View](#)
  - [Contact Info](#)
  - [Tech Info](#)
  - [IPAM](#)
  - [DNS](#)

### What are Gadgets

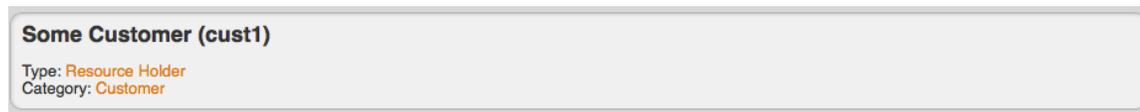
Our gadget system is similar to the Atlassian Gadget system (and Google Gadgets). When creating or editing a Resource type, gadgets can be added to the type in a way similar to how you would add or remove a field. Gadgets are best described as self contained webapps; widgets but with more power. Gadgets can have their own fields, HTML templates, and even accompanying scripts and stylesheets. They can interface with the API to display simple information such as the Type of the Resource, or they can perform much more complex functions as demonstrated with the IPAM gadget in the following section.

The only gadgets currently available are the ones provided with Provision, but it is our hope to provide a simple way for you to create and add your own gadgets in the future.

### Available Gadgets

#### Resource View

This visual element is used on the Resource Holder record Type.



A screenshot of a gadget titled "Some Customer (cust1)". Below the title, it displays "Type: Resource Holder" and "Category: Customer" in orange text.

#### Contact Info

This visual element is used on the Resource Holder record Type.



A screenshot of a gadget titled "Contact Info" with an "edit" link in the top right corner. The gadget displays contact information in two columns:

<b>Phone:</b>	<b>Fax:</b>
<b>Mailing Address</b> 123 Fake St. Santa Clara, CA 95053 US	<b>Billing Address</b> 423 Really Fake St. Suite 120 San Jose, CA 95001 US

#### Tech Info

This visual element is used on the Resource Holder record Type.

### Tech Info edit

**DNS Servers**

ns1: ns1.domain.com	ns2: ns2.domain.com
ns3: ns3.domain.com	ns4: ns4.domain.com
ns5: ns5.domain.com	ns6: ns6.domain.com

**ARIN Info**

<b>Org ID:</b>	ARIN-ORGNAME	<b>Org POC:</b>	ARIN-POC1
<b>Net POC:</b>	ARIN-POC2	<b>Abuse POC:</b>	

**Origin AS:** 23456

**Residential Customer Privacy:** Disabled

## IPAM

This visual element is used on the Resource Holder record Type.

### IPAM IPv4 IPv6

**Assign Block:**

**Browse To Assign**  
List available blocks:

**Direct Assign**

**Smart Assign**

**Filter:**  
      Filter Clear

Address	Hosts	RIR/LIR	Region	Notes	Tags	Assigned	Updated
10.0.1.0/29	8	1918	CHI		Customer	2013-08-05 01:01:02	2013-08-05 01:01:02

## DNS

This visual element is used on the Resource Holder record Type.

### DNS

**New DNS Zone**  Create Zone

**Zone Delegation**

<b>Delegated Zone</b>	<b>Slave IP</b>	<b>Customer</b>
<input type="text" value="Zone name"/>	<input type="text" value="IPv4 or IPv6"/>	<input type="text" value="84"/>

Add Slave

**Zone Records** **Tags** **Entries**

test-domain.com		5
-----------------	--	---

## XML Specifications

### XML Specifications

**THIS IS AN EXPERIMENTAL FEATURE**

User created gadgets are not supported at this time. We recommend waiting until a more user friendly system is implemented before attempting to make custom gadgets.

- [XML Specification](#)
  - [Implemented Tags](#)
  - [Example](#)
  - [Fields](#)

[XML Specification](#)

The XML gadget specification is based on the Atlassian Gadgets.

### Implemented Tags

The implemented tags and corresponding attributes are:

- ModulePrefs
  - Description
    - title
    - width - "full" or "half" are the only options for now
- ContentSources
  - type - "file" uses the file given in src, "html" uses the content in the tag (eg. <Content type="html">This is the content</Content>)
  - src - relative filename or url
- Source
  - Fields
    - type - "css" or "javascript"
    - src - relative filename or url
- Field
  - slug

### Example

```
<?xml version="1.0" encoding="UTF-8" ?>
<Module>
  <ModulePrefs title="Contact Info" width="half" />
  <Description>This gadget adds a field editor for fields related to contact info
(phone, address, ect).</Description>
  <Content type="file" src="template.html" />
  <Sources>
    <Source type="javascript" src="script.js" />
  </Sources>
  <Fields>
    <Field slug="6c-resourceholder-phone-main" />
    <Field slug="6c-resourceholder-phone-fax" />
  </Fields>
</Module>
```

### Fields

If a gadget uses fields, you can optionally add the slug of the field in this section to hide it from the main field list.

This can be very useful and make your Resource Types easier to work with. If the fields are not hidden, this can lead to long lists of redundant data in multiple places and can cause confusion. However, all viewing and editing of the field will have to be done through the gadget. If your gadget uses a field in a read-only manner, then you should **not** add it to the gadget's manifest because that would prevent users from editing the field data through the standard edit page.

## Admin Preferences

### Overview



### Details

[dnsconfig](#)

### License Info

This section provides basic information on your 6connect license including the option to view the *EULA* and check your license status.

## Application Settings

**Time Zone:** Supported Time zones are listed here: [EXT](http://www.php.net/manual/en/timezones.php) <http://www.php.net/manual/en/timezones.php>. Default value is ('America/Los\_Angeles') and can be modified at any time via the drop down menu

**Company Name:** Enter the preferred name for your company to be used.

**Generic Name:** This "short" name is used in abbreviated location for the "Customer" tab label, "Customer" and "Site" are common entries.

## Peering Parameters

**ASN :** Enter the ASN that will be used

## Backup Parameters (local install only)

**Enable mysql offsite backup :** This is enabled by default. Go to the [Backup](#) section for details on this feature.

**Location of mysqldump:** This is the location of the mysqldump directory.

## Logging Options

**Log table size:** This is the maximum number of records to store in the log table. Default value is 50,000,000.

**Rows to remove at limit:** When the value for log\_table\_max is reached, the number of rows to be cut from the table is the number assigned to this variable. Default value is 10,000 rows.

**Local Syslog Enable:** Check the box to enable syslog functionality or for local logging to the database only

**Remote Log IP:** Target IP address that we will send log information to

**Remote Log Port:** Port number for the syslog server you will send log information to

**Remote Log Method:** Select TCP, UDP, SSL from the dropdown for the log delivery method

**Remote Log Backup IP:** Target IP address for the Backup syslog server you will send log information to

**Remote Log Backup Port:** Port number for the Backup syslog server you will send log information to

**Remote Log Backup Method:** Select TCP, UDP, SSL from the dropdown for the log delivery method

**Remote Log Type:** Select SysLog format or JSON output

**Remote Log Facility:** Select the Facility - applies to syslog only

## Authentication Options

**Maximum Session Idle:** This setting (minutes) controls how long a session can stay idle before being forced to log in again.

### ***RADIUS authentication options (local install only) - for implementation details, go here***

**Radius Enable:** Check this box to enable RADIUS functionality.

**Radius Server Address:** Set to the IP address of your radius server. If this is specified, it will force authentication over radius.

**Radius Authentication Port:** Set to the port for authentication. Default port is 1812

**Radius Accounting Port:** Set to the port for radius accounting. Default port is 1813

**Radius Key:** Set to the shared key of your radius server

### ***LDAP authentication - for implementation details, go here***

**LDAP Enable:** check the box to enable LDAP functionality.

**LDAP Server Address:** Set the IP address of your LDAP server.

**LDAP Port:** Set the port for your LDAP server

**LDAP Security:** Select the security method of your LDAP server - SSL, TLS or None

**LDAP Auth DN/Fetch DN:** These strings are used to first authentication the 6connect user and then to retrieve their permissions. The string '%LOGIN%' should be inserted in place of the user's common name both strings. (ex: cn=%LOGIN%,ou=people,dc=6connect,dc=com)

**Mapping Permissions to 6connect schema:** To integrate 6connect permissions with your existing directory structure then you will need the 6connect schema. It should snap in with any existing LDAP structure and allow you to assign 6connect permissions to your existing users. You can download a copy of the schema from this section.

## Templates

This is where you can edit outgoing email templates for IP block assignments

## Authentication Options



Depending on the authentication method chosen by your organization, there may be a separate authentication to login or logout of the application via the drop down menu.

By default, credentials are managed via the local authentication mechanism provided by 6connect. See the [Permissions Overview](#) section for more detail on the local authentication configuration.

- [LDAP Authentication](#)
- [RADIUS Authentication](#)

## LDAP Authentication

### *LDAP Authentication*

Starting in 3.6, ProVision supports LDAP authentication. To an LDAP server for authentication, you must perform the following three procedures:

- [Configure the LDAP Server](#)
- [Test the LDAP Server](#)
- [Configure ProVision for LDAP Authentication](#)

### **Configure the LDAP Server**

SSH into your openLDAP server and create a new 'ldif' file. Example:

```
dn: cn=JoeSmith,ou=people,dc=6connect,dc=com
cn: JoeSmith
sn: JoeSmith
objectclass: top
```

```
objectclass: person
objectclass: sixConnectPermissionsV2
sixConnGroup: "Global Admins"
sixConnGroup: "bonk"
sixConnGroup: "poof"
sixConnGroup: "groot"
userPassword: testpass
```

To create a new user, make a new Idif file and change all instances of "JoeSmith" to whatever username you wish to create and update the password. Keep all of the object class definitions as listed above. Add a sixConnGroup declaration for each ProVision user group a user is in.

After the file is created, run the following command to add the new user to LDAP server:

```
ldapadd -h [SERVER] -x -f [LDIF FILE] -D [ROOTDN] -w [ROOT PW] -v
```

Example:

```
ldapadd -h localhost -x -f 6connect.ldif -D "cn=Manager,dc=6connect,dc=com" -w secret -v
```

The user will now be active in openLDAP and can be used to login to ProVision.

### Test the LDAP Server

To query the LDAP server, run the following command on any server which has openLDAP enabled:

```
ldapsearch -h [IPADDRESS] -D [DOMAIN] -w [PASSWORD] [USER]
```

*Note: We have not been able to use a v6 address at with this tool, even though multiple sources say it should work.*

At the end of the command where [USER] is specified, user or groups can be used (in LDAP format) to query.

Example:

```
ldapsearch -h 50.240.195.129 -D "cn=JoeSmith,ou=people,dc=6connect,dc=com" -w testpass "cn=JoeSmith"
```

### Configure ProVision for LDAP Authentication

To configure the use of LDAP authentication with ProVision, follow the steps below.

- Log into 6connect ProVision
- Go to Admin -> General Settings -> Authentication
- Click the LDAP Enable checkbox.
- Fill in the hostname or ip address, authentication port, LDAP Security, Auth DN, and Fetch DN. An example is below:

LDAP Server Address: 52.240.195.12

LDAP Port: 389 ( or SSL/TLS port is 636)

LDAP Security: None

LDAP Auth DN: cn=%LOGIN%,ou=people,dc=6connect,dc=com

LDAP Fetch DN: cn=%LOGIN%



#### Setting default login authentication options

In the login screen, you would select the authentication method from the dropdown. If you like, you can set the default login option in the following way:

Go to the /data/globals.php and open in vi (or other editor). Add in the following text as the last line of the file (before the closing ?>)

```
define('DEFAULT_LOGIN_TYPE', 'radius');
```

Acceptable values are "local", "radius" and "ldap". If this line is not present in globals.php, the default option is "local".

## RADIUS Authentication

### RADIUS Authentication



- [Add the 6connect VSA to the Radius Installation](#)
- [Configure Radius Accounts](#)
- [Test Radius Accounts](#)
- [Configure ProVision for Radius Authentication](#)

Starting in 3.0, ProVision supports 6connect vendor-specific attributes (VSAs) for use with RADIUS authentication. To use these attributes, you must perform the following procedures:

- [Add the 6connect VSA to the Radius installation](#)
- [Configure Radius accounts](#)
- [Test the Radius account](#)
- [Configure ProVision for Radius Authentication](#)

#### **Add the 6connect VSA to the Radius Installation**

To use the 6connect VSA, the attributes must be defined on the RADIUS server. Add the following RADIUS dictionary file to your RADIUS server and name it dictionary.6connect:

*Important Note: Between version 3.9.3 and 4.0, the permissions structure for ProVision was significantly changed. Make sure you following the version specific instructions below.*

#### **ProVision 3.9.3 and prior:**

▼ [Click here to expand...](#)

### 3.9.3 VSA text file

```
VENDOR          6connect          36009

BEGIN-VENDOR    6connect

ATTRIBUTE      priv_admin          10      integer
#This is used to give a user administrative access to the application

ATTRIBUTE      priv_ipam_c         20      integer
#This allows a user to create IP blocks
ATTRIBUTE      priv_ipam_d         21      integer
#This allows a user to delete IP blocks
ATTRIBUTE      priv_ipam_m         22      integer
#This allows a user to modify IP blocks
ATTRIBUTE      priv_swip          23      integer
#This allows a user to SWIP IP blocks
ATTRIBUTE      priv_email          24      integer
#This allows a user to email IP block information
ATTRIBUTE      priv_ipam_v         25      integer
#This allows a user to view IP block information

ATTRIBUTE      priv_dns_c          30      integer
#This allows a user to create DNS Zones
ATTRIBUTE      priv_dns_d          31      integer
#This allows a user to delete DNS Zones
ATTRIBUTE      priv_dns_m          32      integer
#This allows a user to modify DNS Zones
ATTRIBUTE      priv_dns_v          33      integer
#This allows a user to view DNS Zones

ATTRIBUTE      priv_cust_c         40      integer
#This allows a user to create Customer records
ATTRIBUTE      priv_cust_d         41      integer
#This allows a user to delete Customer records
ATTRIBUTE      priv_cust_m         42      integer
#This allows a user to modify Customer records
ATTRIBUTE      priv_cust_v         43      integer
#This allows a user to view Customer records

ATTRIBUTE      priv_peer_c         50      integer
#This allows a user to create peering sessions
ATTRIBUTE      priv_peer_d         51      integer
#This allows a user to delete peering sessions
ATTRIBUTE      priv_peer_m         52      integer
#This allows a user to modify peering sessions
ATTRIBUTE      priv_peer_v         53      integer
#This allows a user to view peering sessions

ATTRIBUTE      priv_logs           60      integer
#This allows a user to have access to the logs tab in the application

END-VENDOR      6connect
```

#### ProVision 4.0 and greater:

▼ [Click here to expand...](#)

```

VENDOR          6connect          36009

BEGIN-VENDOR    6connect

ATTRIBUTE          6connect_user_group      10      string
#A 6connect User Group to which this user belongs.

END-VENDOR      6connect

```



Make sure to add the following to the primary dictionary file: `$INCLUDE dictionary.6connect`

### Configure Radius Accounts

On the Radius server, configure the user accounts that will have access to the ProVision system.

An example of a ProVision account configuration for the user file on a Freeradius system for version 3.9.3 and prior:

```

#A user with full IPAM privileges and view only DNS privs

joe Cleartext-Password := "testing128"
  priv_admin = 1,
  priv_ipam_v = 1,
  priv_ipam_c = 1,
  priv_ipam_d = 1,
  priv_ipam_m = 1,
  priv_swip = 1,
  priv_email = 1,
  priv_dns_v = 1

```

An example of a ProVision account configuration for the user file on a Freeradius system for version 4.0 and greater:

**Example:** To add a new radius user, edit the 'users' file found at `/etc/raddb/users` and add a block like:

#### Setting up a RADIUS account

```

bobber Cleartext-Password := "hello"
      6connect_user_group = "Global Admins,Group 2,Group 1,Group Nonexistent"

```



#### Note on RADIUS attributes

There are many Radius attributes, but '6connect\_user\_group' is the one used by 6connect ProVision and it is just a comma-separated list of all the group names that the user belongs to.

### Test Radius Accounts

For 3.9.3 and prior, test and response should look like the following:

```
#>radtest test test 50.23.215.162 6connect
  Sending Access-Request of id 179 to 50.23.215.162 port 1812
  User-Name = "test"
  User-Password = "test"
  NAS-IP-Address = 10.124.47.6
  NAS-Port = 0
  Message-Authenticator = 0x00000000000000000000000000000000
rad_recv: Access-Accept packet from host 50.23.215.162 port 1812, id=179, length=68
  priv_admin = 1
  priv_ipam_c = 1
  priv_ipam_m = 1
  priv_ipam_d = 1
```

For 4.0 and higher, test and response should look like the following:

<insert example>

### Configure ProVision for Radius Authentication

To configure the use of Radius authentication with ProVision, follow the steps below.

- Log into 6connect ProVision
- Go to Admin -> General Settings -> Authentication
- Ensure that Radius functions are marked as available. Radius functions are always available on 6connect cloud instances. Radius functions are available on VM Images and Local Installations only if the relevant PHP Pear Radius Libraries have been installed.
- Click the Radius Enable checkbox.
- Fill in the hostname or ip address, authentication ports, accounting port, and shared Radius key as specified.



#### Setting default login options

In the login screen, you would select the authentication method from the dropdown. If you like, you can set the default login option in the following way:

Go to the /data/globals.php and open in vi (or other editor). Add in the following text as the last line of the file (before the closing ?>)

```
define('DEFAULT_LOGIN_TYPE', 'radius');
```

Acceptable values are "local", "radius" and "ldap". If this line is not present in globals.php, the default option is "local".

## Permissions Overview

### Overview

The Permissions structure in ProVision is designed to give you as much flexibility as you need to accommodate most use cases. When mapping out the permissions structure for your organization, keep in mind who you want to access to application:

- Internal Users and Roles (Admins, Read Only, etc.)
- Partners related to multiple specific Resources/Accounts
- Customers/Departments with limited view to only their respective Resources/Accounts



### Permission Levels

#### Global Permissions

When you see a reference to a "TLR" - that is a "Top Level Resource". This Is the primary Resource under which all other resources fall under. ProVision currently only allows a single level of administrator permissions: Global Administrator.

Users with "Admin" access can assign/modify permissions for other users.

See [Global Permissions](#) for more details on configuring these elements.

### Resource Permissions

An administrator can also set respective permissions for a given Resource (single or multiple). These permissions fall under Groups. So a Group is configured for the given group of Resource permissions, and then the User account is added.

See [Users and Groups](#) to learn how Resource Permissions are assigned.

See [Resource Permissions](#) for more details on configuring these elements.

## Global Permissions

Global Permissions apply to the "TLR" or "Top Level Resource" within ProVision.

Administration of these permissions require Administrative privileges. As an Admin, the user can then assign global permissions to groups and users. Depending on the requirement, the user can also have Resource specific permissions depending on how their group is configured.

### Global Permission Details

Global Permission	Description
Create	Ability to create records of a certain type
Read	Ability to read records of a certain type
Update	Ability to update existing records of a certain type
Delete	Ability to delete records of a certain type

Functional Area	Description
IPAM	IP Address Management functionality - this covers the IPAM Tab in addition to the IPAM "Gadget" that can be present in Resources.
DNS	DNS Zone/Zone Record Management functionality - this covers the DNS Tab in addition to the DNS "Gadget" that can be present in Resources.
Peering	Peering functionality - covers the Peering Tab, both the Communication Manager and the Session Manager.

Resources	Resource functionality - this controls access for Resources depending on either the TLR or the individual Resource.
User	User/Group management - this controls access for User and Group functions within the administrative area for ProVision.
SWIP*	This affects the SWIP/RPSL integration for ARIN/RIPE. This way a user can either be enabled to have this capability or not.
Admin*	This controls whether a user is an administrator for the global ProVision application.



\*

SWIP and Admin functions are only visible when [Show Details](#) is selected

## Resource Permissions

Resource Permissions apply to designated Resources within ProVision.

Administration of these permissions require Administrative privileges. As an Admin, the user can then assign resource permissions to groups and users.

### Resource Permission Details

Resource	IPAM				DNS				Peer				Resource				User				SWIP	Admin	
	C	R	U	D	C	R	U	D	C	R	U	D	C	R	U	D	C	R	U	D			
Some Customer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																		
Top-Level (Global Access)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Resource Permission	Description
Create	Ability to create records of a certain type
Read	Ability to read records of a certain type
Update	Ability to update existing records of a certain type
Delete	Ability to delete records of a certain type

Functional Area	Description
IPAM	IP Address Management functionality - this covers the IPAM Tab in addition to the IPAM "Gadget" that can be present in Resources.
DNS	DNS Zone/Zone Record Management functionality - this covers the DNS Tab in addition to the DNS "Gadget" that can be present in Resources.
Peering	Peering functionality - covers the Peering Tab, both the Communication Manager and the Session Manager.

Resources	Resource functionality - this controls access for Resources depending on either the TLR or the individual Resource.
User	User/Group management - this controls access for User and Group functions within the administrative area for ProVision.
SWIP*	This affects the SWIP/RPSL integration for ARIN/RIPE. This way a user can either be enabled to have this capability or not.
Admin*	This controls whether a user is an administrator for the global ProVision application.



\*

SWIP and Admin functions are only visible when [Show Details](#) is selected

## Users and Groups

### User Accounts

A User is defined as a single login account that accesses ProVision.

New Users can be created from the "Manage Users" Tab under the Admin area by clicking the green "Add User" button.

Username	Name	Groups	
admin	Admin User	Global Admins	
alina@6connect.com	Alina Fry	AAA Group	
annac@6connect.com	Anna Claiborne	Global Admins	
brenner@6connect.com	Bill Renner	Global Read-Only	
colin@6connect.com	Colin Robinson	Global Read-Only	
fy@6connect.com	Fry Chen	AAA Group	

[Add User](#)

### Creating/Editing Accounts

When creating or editing User accounts, you will be presented with the following options. Note that membership in multiple permission groups is allowed.

Username:

First Name:

Last Name:

Disable help bubbles?

Groups:

- AAA Group
- test resource
- PJ Test
- test group 1

[Save](#)

## Setting/Resetting User Passwords

When you click on the padlock icon, you will be presented with options to set a new password and/or send a password reset email to the intended user account.

Yes ✕

### Reset Password

New Password:

Send email?

From:

To:

Subject:

Message: 

Dear test mcTest,

6connect Support <ops@6connect.com> has requested your credentials be reset for 6connect IPAM at <https://ops.6connect.com/qa-4.0>.

Your username is: test@6connect.com  
Your new password is: xSEXiHVu

Login at: <https://ops.6connect.com/qa-4.0>  
Go to the "gear" icon in the upper right to reset your password at any time after you have logged in successfully.

Regards,  
6connect Automated Admin

## User Groups

ProVision administrators can also create permission groups to assign users to. This allows more control over user roles. The two default groups are:

- Global Admin
- Global Read-Only

New Groups can be created by ProVision administrators by pressing the green "Add Group" button.

Name	Enabled	Users	
Global Admins	Yes	6	
Global Read-Only	Yes	5	
Global Group 2	Yes	3	
Global Group 3	Yes	4	
Global Group 4	Yes	1	
Global Group 5	Yes	3	



### Overlapping group and user permissions

Permissions are inherited based on the hierarchy of the objects, unless you specify a different permission!

## Verifying Permissions

To verify the permissions of a certain user who is a member of a group, simply select their user account from the dropdown menu and click on the green "Query" button. The resulting output will display the Resources the user has access to along with the specific permissions for each one.

### Check User Permissions

User:  Resource:

IPAM		DNS		Peer		Resource		User		SWIP	Admin		
C	R	U	D	C	R	U	D	C	R	U	D		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>									

Groups effecting this user on this resource: test group 1

## IPAM Administration

### Overview

Intro Video

<b>IPAM Lists Management:</b> <ul style="list-style-type: none"><li><a href="#">Edit IPAM Tags</a></li><li><a href="#">Edit IPAM Regions</a></li><li><a href="#">Edit Contact Roles</a></li><li><a href="#">Edit IPv4 Subnets Dropdown</a></li><li><a href="#">Edit IPv6 Subnets Dropdown</a></li><li><a href="#">Edit RIR List</a></li><li><a href="#">Edit IPv4 Exact Filter</a></li><li><a href="#">Edit IPv6 Exact Filter</a></li></ul>	<b>Holding Tank Management:</b> <ul style="list-style-type: none"><li><a href="#">Process Holding Tank now (Set to 60 days)</a></li></ul>
	<b>DHCP Management:</b> <ul style="list-style-type: none"><li><a href="#">DHCP Admin Home</a></li></ul>
	<b>LIR Management:</b> <ul style="list-style-type: none"><li><a href="#">Add/Edit/Update LIRs</a></li></ul>

### IPAM Parameters

These links are to the respective [IPAM Parameters](#) that are available for customization. Everything from Tags to RIRs - this is where to start. Go to the [IPAM Parameters](#) page for more details and examples for customization.

### IPAM Configuration

**Holding Tank Days:** This is the number of days that a block will be held in "Holding" status before being available to be moved to the Available pool, and thus ready to be assigned. By default this is initially set to 30 days.

**IPv4 Block Scanner Enable:** This is a beta feature that allows a user to scan a block of IPv4 space and show host counts of responding addresses.

**Regions Enable:** Check the box to enable "Region" tags for IP blocks. This will add an additional column to the default IPAM screen. It is treated similarly to a standard tag. You can set the values from the "Edit Tags" function and modify the values list in the IPAM Admin screen "Edit Regions".

**Generic Code Per Block Enable:** Check this box to enable this function. This will enable an additional field per IP Block.

**Generic Code Per Block Display:** Check this box to display this field.

**Generic Code Per Block Name:** This is the label for the Generic Code to be displayed.

**Enable VLAN per Block:** This toggle allows users to specify VLANs via the "Edit Tag" function. With this feature enabled, you can filter by VLAN tag in the primary IPAM interface.

### Holding Tank Management

When IPv4/IPv6 resources are reclaimed, they are placed into the "Holding Tank". This feature allows for a block to stay out of the available address pools until the administrator approves it. Go to the [Holding Tank Management](#) page for more details.

### LIR Management and Use

ProVision supports multiple LIRs from the UI. This allows users to select from various LIRs when they want to update SWIP/RPSL information for

a subnet allocation. Go to the [LIR Management and Use](#) page for more details.

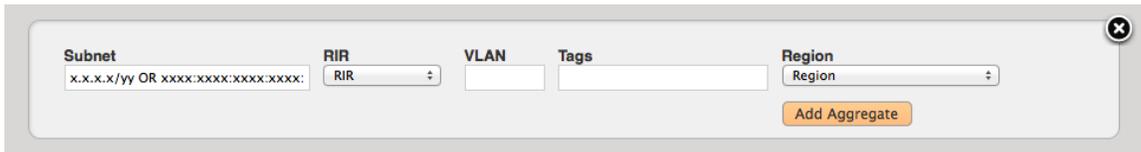
## Working with IP Blocks

### Adding IP Address Aggregates

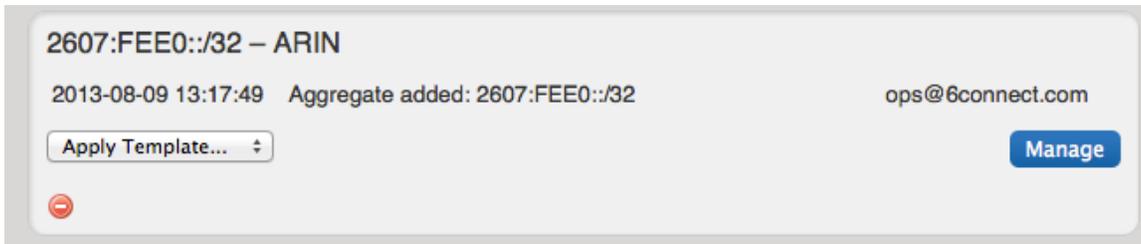
In the standard IPAM page - you should have an option to "Add Aggregate"



Once clicked, you should get a more detailed screen to add an aggregate block



When a block is added, you will be able to see it on the IPAM page



### Architecting IP Address Blocks

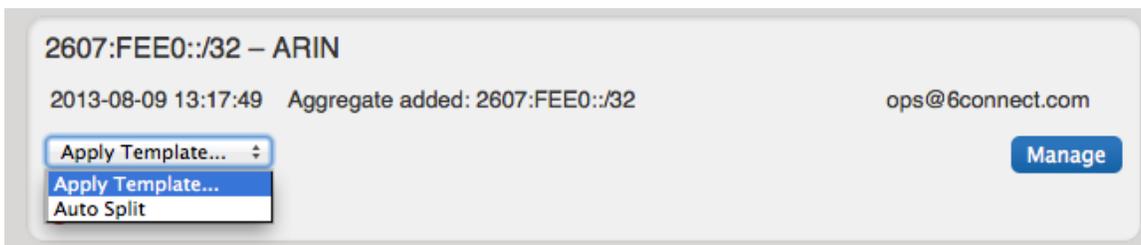
#### Splitting/Aggregating blocks manually

To split a block manually - you can use the functions from the Manage screen for any aggregate

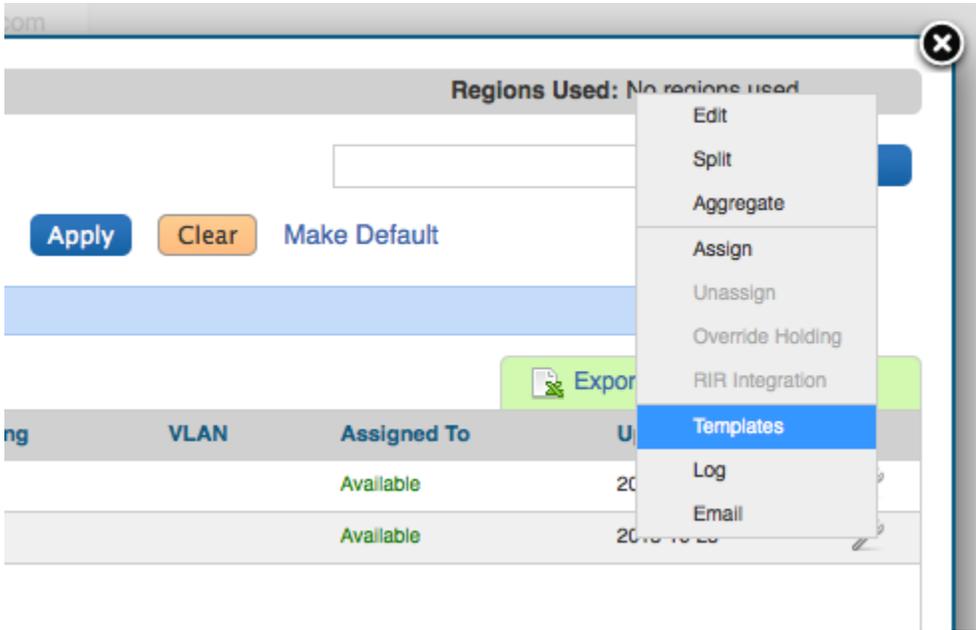


#### Splitting/Aggregating blocks with Templates

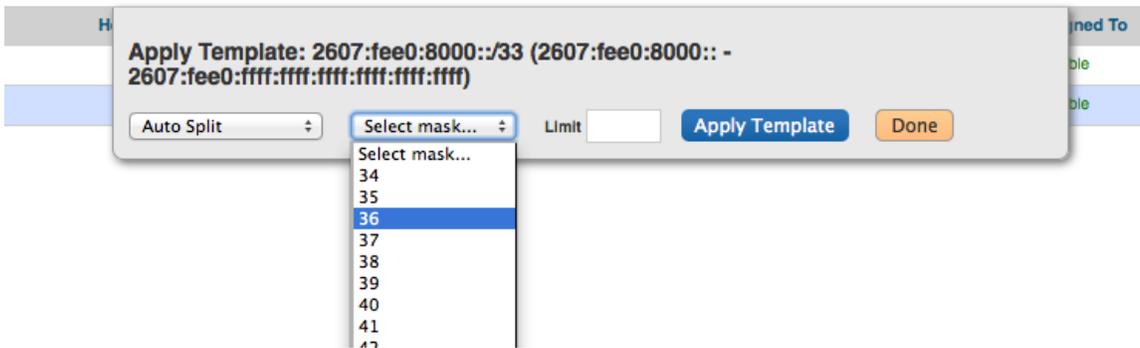
When you first import a block, you can select to use the Auto Splitting function from the main IPAM page



or you can also use the "Templates" on the IPAM Manage screen for the specific block

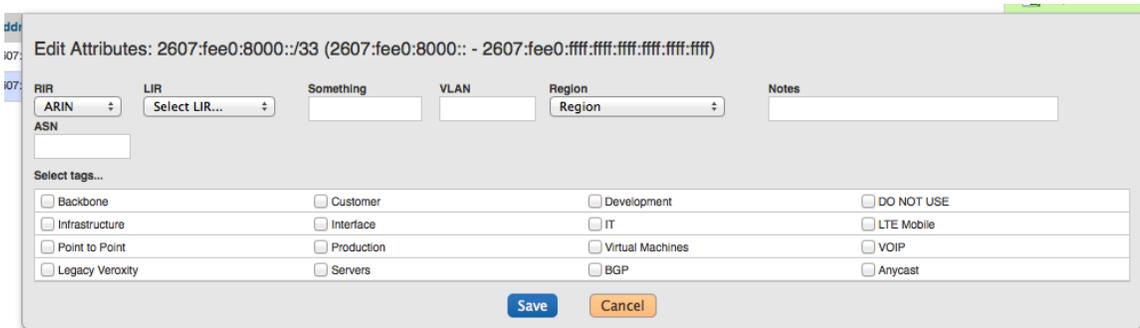


and then select the auto split parameters from there



### IP Block parameters

When you have your IP blocks laid out, you can then modify their attributes, split them further, assign them, etc. Select the "Edit" option for a given block to get the following menu



From here you can set a variety of attributes for a given block. All of these values are also customizable from the Admin screen - IPAM Admin.

## IPAM Lists Management:

- Edit IPAM Tags
- Edit IPAM Regions
- Edit Contact Roles
- Edit IPv4 Subnets Dropdown
- Edit IPv6 Subnets Dropdown
- Edit RIR List
- Edit IPv4 Exact Filter
- Edit IPv6 Exact Filter

## Assigning IP Space

To assign space, you do it using the IPAM Gadget

Address	Hosts	RIR/LIR	Region	Notes	Tags	Assigned	Updated
10.0.0.0/30	4	1918	ATL		Anycast,BB	2013-09-20 13:38:10	2013-09-20 13:38:10
10.0.0.4/30	4	1918	ATL		Anycast,BB	2013-09-24 11:34:15	2013-09-24 11:34:15
10.0.0.16/30	4	1918	ATL		Anycast,BB	2013-10-11 13:50:03	2013-10-11 13:50:03
10.1.0.0/32	1	1918	ATL		Cable,Customer	2013-09-20 13:38:30	2013-09-20 13:38:30
10.1.0.8/29	8	1918	ATL		Cable,Customer	2013-09-20 13:38:40	2013-09-20 13:38:40
67.21.0.0/29	8	ARIN	ASH		Anycast,BB	2013-09-16 18:14:32	2013-09-16 18:14:32
67.21.0.8/29	8	ARIN	ASH		Anycast,BB	2013-09-17 23:13:17	2013-09-17 23:13:17
67.21.0.16/28	16	ARIN	ASH		Anycast,BB	2013-09-17 23:14:36	2013-09-17 23:14:36

Or you can assign blocks manually using the "Assign" function

Assign block

Filter by type: All Types

- All Types
- Firewall
- LIR
- Rack
- Resource Holder
- Router
- Server
- Storage Array
- Storage Controller
- Switch
- Virtual Machine

007 Manufacturing (Resource Holder)	Reverse DNS for HVDN Prefixes (Resource Holder)	123 Enterprises (Resource Holder)	33rd St. Bistro (Resource Holder)
44 Magnum Beats (Resource Holder)	(Resource Holder)	6c-009 (Resource Holder)	6c-026 (Resource Holder)
6c-033 (Resource Holder)	(Resource Holder)	9-All Nine's (Resource Holder)	92ND STREET YOUNG MENS AND YOUNG
A + Technology Solutions (Resource Holder)	of New York (Resource Holder)	ABCAM (Resource Holder)	Acadian Asset Management LLC (Resource Holder)
Acceleron Pharma, Inc. (Resource Holder)	Plus Communications Inc (Resource Holder)	Acronis, Inc. (Resource Holder)	ACS Consultant Company (Resource Holder)
ADA Investment Management (Resource Holder)	ly Inc. (Resource Holder)	Adelphi University (Resource Holder)	ADTRAN, Inc. (Resource Holder)
Advanced Instruments (Resource Holder)	Advanced Physician Services, PC (Resource Holder)	Aer Lingus (Resource Holder)	Affiliated Pathology Service (Resource Holder)

Assign Block Cancel

Please note that once a block is assigned, you will also have other options available, including reverse DNS, and IP subassignments

Edit Attributes: 2607:fee0:8000::/33 (2607:fee0:8000:: - 2607:fee0:ffff:ffff:ffff:ffff:ffff:ffff)

Assigned To: [Some Customer](#)

[Edit Reverse DNS](#)

Allow sub assignments for this block

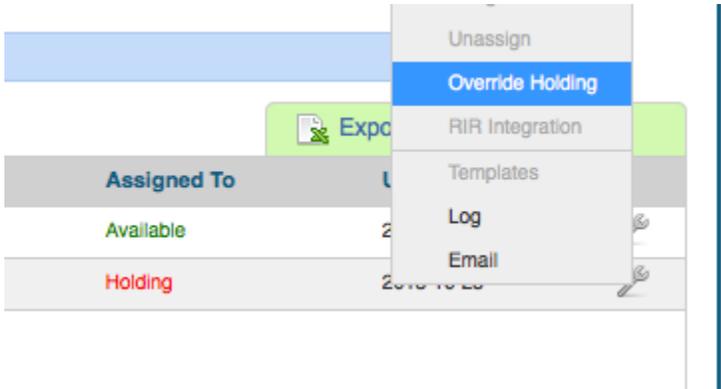
## Sub Assigning IP Space

To allow sub assignments, just check the Allow sub assignments check box under Edit. Once the allow sub assignments box is checked, the block may be further split and assigned to other resources. Split blocks may also be re-claimed to the originally assigned resource and re-aggregated. When allow sub assignments is checked, the block is counted as allocated, but not assigned - various statistics in IPAM, on the dashboard, and reporting will reflect this. Sub assignments can be useful for tracking IPs assigned to a customer with multiple subsidiaries, or locations.

### Unassigning IP Space

When a block is assigned, you will then have the option of unassigning the block from the resource and returning it to the Holding Tank. To return IP space in the Holding Tank to the Available Pool - there are two methods:

1) Manually override the holding tank



2) Process the Holding Tank via the Admin screen (this will only process blocks that were present for the specified number of days)

<p><b>IPAM Lists Management:</b></p> <ul style="list-style-type: none"> <li>Edit IPAM Tags</li> <li>Edit IPAM Regions</li> <li>Edit Contact Roles</li> <li>Edit IPv4 Subnets Dropdown</li> <li>Edit IPv6 Subnets Dropdown</li> <li>Edit RIR List</li> <li>Edit IPv4 Exact Filter</li> <li>Edit IPv6 Exact Filter</li> </ul>	<p><b>Holding Tank Management:</b></p> <p><a href="#">Process Holding Tank now (Set to 60 days)</a></p> <p><b>LIR Management:</b></p> <p><a href="#">Add/Edit/Update LIRs</a></p>
---	---

## IPAM Parameters

### Overview

The elements

**IPAM Lists Management:**

- Edit IPAM Tags
- Edit IPAM Regions
- Edit Contact Roles
- Edit IPv4 Subnets Dropdown
- Edit IPv6 Subnets Dropdown
- Edit RIR List
- Edit IPv4 Exact Filter
- Edit IPv6 Exact Filter

### Editing Tags

When you are applying properties to IP blocks, you have the option to edit tags. Tags are used in a number of ways and can be edited from this screen. You can specify tag values along with sorting options to make it simpler to use. Regions are used by the [IPAM Gadget](#) and the IPAM Management UI).

### Editing Regions

If enabled, Regions can function as a way to further define your network segments (regional tie-downs, etc.). This simply gives you flexibility for allocations and assignments beyond simply using Tags. Regions are used by the [IPAM Gadget](#) and the IPAM Management UI).

### Editing Subnet Dropdowns (used by the IPAM Gadget)

When assigning blocks using the "Smart Assign" function in the [IPAM Gadget](#), the user has an option to assign an IP resource by allocation size. ProVision supports assignments down to a single host level (/32 for IPv4, /128 for IPv6).



#### Note on Editing the Subnet Dropdown

Keep in mind that this is a global edit. If the values in the dropdown are changed, it will affect ALL users of the ProVision application

### Edit RIR List (used by the IPAM Gadget and the IPAM Management UI)

When working with IP aggregates, an editable element is "RIR". The RIR can be assigned on an aggregate level, via the Bulk Editing function, or when editing via the standard "Edit Details" menu.



#### Tracking Overlapping IP Space

This can be used to better track overlapping IP blocks (VRFs, IP space from a merger or acquisition, etc.). For example, you could have an RIR entry of "1918-Corp HQ" and "1918-Company-X". This would allow for an overlapping /8 of IP space, but allocation and assignment tracking would function normally.



#### RIR and SWIP/RPSL functions

ProVision uses the RIR associated with an IP block for SWIP/RPSL functions. As long as the RIR entry starts with "ARIN" - ProVision will know to use the ARIN SWIP functions for those blocks. The same for RIPE blocks. As other RIR API support is added, they will function the same way.

### Edit Exact Filter Dropdowns for Filter by Netmask

On the IPAM Manage screen, you have an option to Filter the view by selected Subnet Mask (dropdown).

The screenshot shows the IPAM Manage interface. At the top, there's a search bar and filters for 'Tags Used' (No tags used) and 'Regions Used' (No regions used). Below the search bar, there's a 'Filter By' section with a dropdown menu currently set to 'All Masks'. The dropdown menu is open, showing options: /24, /25, /26, /27, /28, /29, /30, /31, /32. The main table displays IP blocks with columns: Hosts, RIR, Region, Notes, Tags, Something, Assigned To, and Updated. The table has several rows, with the last row highlighted in green.

	Hosts	RIR	Region	Notes	Tags	Something	Assigned To	Updated	
<input type="checkbox"/>	2	1918					<a href="#">brl_awesome sauce.com</a>	2013-07-24 19:54:13	
<input type="checkbox"/>	2	1918					<a href="#">brl_awesome sauce.com</a>	2013-07-24 19:54:21	
<input type="checkbox"/>	1	1918					<a href="#">brl_totallyrad.com</a>	2013-07-25 16:41:40	
<input type="checkbox"/>	12.11.10.5/32	1	1918				<a href="#">brl_totallyrad.com</a>	2013-07-25 16:41:48	
<input type="checkbox"/>	12.11.10.6/31	2	1918				Available	2013-07-25 16:41:03	
<input type="checkbox"/>	12.11.10.8/29	8	1918				Some Customer	2013-08-05 19:46:21	
<input type="checkbox"/>	12.11.10.18/29	8	1918				Some Customer	2013-08-05 19:46:37	
<input type="checkbox"/>	12.11.10.24/29	8	1918				Available	2013-08-05 19:46:25	

With the Filter By view enabled, the user then gets a simpler view. The user can then click on the red block, and view the additional assignments/allocations underneath it.

12.11.10.0/24 (12.11.10.0 - 12.11.10.255) Tags Used: No tags used Regions Used: No regions used

Search By:  Search

Filter By: /29 All RIRs All Regions All Something All Status All Customers Clear Filters

Export Current List To CSV

<input type="checkbox"/>	Address	Hosts	RIR	Region	Notes	Tags	Something	Assigned To	Updated	
<input type="checkbox"/>	12.11.10.0/29	8	1918					Has Children	2013-07-24 16:22:35	
<input type="checkbox"/>	12.11.10.8/29	8	1918					Some Customer	2013-08-05 19:46:21	
<input type="checkbox"/>	12.11.10.16/29	8	1918					Some Customer	2013-08-05 19:46:37	
<input type="checkbox"/>	12.11.10.24/29	8	1918					Available	2013-08-05 19:46:25	
<input type="checkbox"/>	12.11.10.48/29	8	1918					Available	2013-07-25 16:41:58	
<input type="checkbox"/>	12.11.10.56/29	8	1918					Has Children	2013-07-25 16:41:58	

Here is the view after clicking on the block. The user can also see the SWIP/RPSL status for a given allocation/assignment if applicable.

12.11.10.0/24 (12.11.10.0 - 12.11.10.255) Tags Used: No tags used Regions Used: No regions used

Search By:  Search

Filter By: /29 All RIRs All Regions All Something All Status All Customers Clear Filters

Export Current List To CSV

<input type="checkbox"/>	Address	Hosts	RIR	Region	Notes	Tags	Something	Assigned To	Updated	
<input type="checkbox"/>	12.11.10.0/29	8	1918					Has Children	2013-07-24 16:22:35	
12.11.10.0/30 12.11.10.0/31 - Assigned to br1.awesomesauce.com 12.11.10.2/31 - Assigned to br1.awesomesauce.com 12.11.10.4/30 12.11.10.4/31 12.11.10.4/32 - Assigned to br1.totalrad.com 12.11.10.5/32 - Assigned to br1.totalrad.com 12.11.10.6/31										
<input type="checkbox"/>	12.11.10.8/29	8	1918					Some Customer	2013-08-05 19:46:21	
<input type="checkbox"/>	12.11.10.16/29	8	1918					Some Customer	2013-08-05 19:46:37	
<input type="checkbox"/>	12.11.10.24/29	8	1918					Available	2013-08-05 19:46:25	
<input type="checkbox"/>	12.11.10.48/29	8	1918					Available	2013-07-25 16:41:58	
<input type="checkbox"/>	12.11.10.56/29	8	1918					Has Children	2013-07-25 16:41:58	

Note that as of 4.1, there are more options for managing filter options and the ability to set a view as Default

Dashboard Resources DNS IPAM Peering Log Reporting

2001:db7::/32 (2001:db7:: - 2001:db7:ffff:ffff:ffff:ffff:ffff:ffff) Tags Used: Anycast, BB, BGP, Customer, DSL, Infrastructure, Internal, Loopback, MPLS Regions Used: LON

Filtered by: Mask: 36, 48, 64

Filter By: Mask LIR ASN Tags Region Code VLAN Assigned To Apply Clear Make Default

Export Current List To CSV

<input type="checkbox"/>	Address	Hosts	LIR	Region	Notes	Tags	Router	VLAN	Assigned To	Updated
<input type="checkbox"/>	2001:db7::/36	2^92	LON						Has Children	2013-09-17
+ 2001:db7::/48 + 2001:db7::/64 - Assigned to br1.swisscom.com + 2001:db7:0:1::/64 - Assigned to Acer Worldwide + 2001:db7:0:2::/64 - Assigned to sconnect Labs + 2001:db7:0:3::/64 + 2001:db7:1::/48 - Assigned to Acer Worldwide + 2001:db7:2::/48 - Assigned to br1.swisscom.com + 2001:db7:3::/48 + 2001:db7:900::/48 - Assigned to Acer Worldwide + 2001:db7:901::/48										
<input type="checkbox"/>	2001:db7::/48	2^80	LON			Anycast, BB		101	Has Children	2013-09-19
+ 2001:db7::/64 - Assigned to br1.swisscom.com + 2001:db7:0:1::/64 - Assigned to Acer Worldwide + 2001:db7:0:2::/64 - Assigned to sconnect Labs + 2001:db7:0:3::/64										
<input type="checkbox"/>	2001:db7::/64	2^64	LON			Anycast, BB		101	br1.swisscom.com	2013-09-19
<input type="checkbox"/>	2001:db7:0:1::/64	2^64	LON			Anycast, BB		101	Acer Worldwide	2013-09-20

## Holding Tank Management

How it Works

The "**Process Holding Tank now**" link will move any block assigned to "Holding" to its relevant "Available" pool. This command will process **ALL** addresses assigned to "Holding" depending on their age. The default time for release to "Available" is 30 days. If a block has not been in the holding tank for that specified length of time, it will not be released using this feature (it can be released manually per record at any time) . The threshold for the number of days in the Holding Tank is set in the main [Admin Preferences](#) page and is customizable.

### Process Holding Tank

38 IPv4 blocks to process.  
1 IPv6 blocks to process.  
Processed 36 IP blocks total.  
Assigned all blocks to 81

[Back to IPAM Admin](#)

When an administrator elects to process the Holding Tank, it will show the information above.

**i Pro-Tip!**

If you need to do a bulk "empty" of the holding tank. Set the time for release to "0" days. This will allow you to process the holding tank for all blocks that are in the Holding Tank.

## LIR Management and Use

### Overview

ProVision supports multiple LIRs (Local Internet Registries) in a single instance. This means that you have the ability to update SWIP/RPSL functions for a given allocation with the LIR information that you wish. When you select the "SWIP" function for a given IP block, you will be presented with a menu where you can select the data that you want to use to update the block.

We currently support ARIN and RIPE with support for APNIC, LACNIC and AfriNIC coming in 2014.

### LIR Setup and Use

There is an LIR Manager available from the IPAM Admin page.

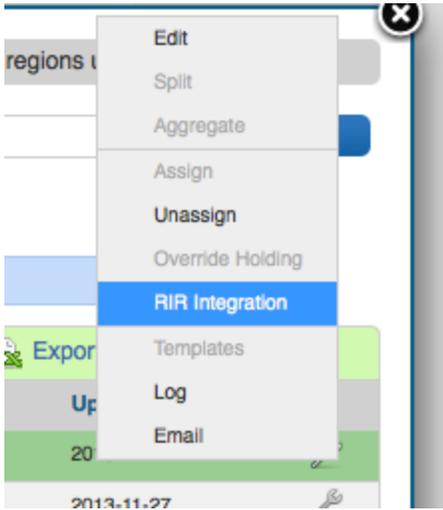
#### LIR Manager

Name	RIR	ASN	MNT-BY/ORG-HANDLE	Other Information		
ARIN Default LIR	ARIN	9498	CONNE-81	Admin POC: 6CONN-ARIN , Tech POC: 6CONN-ARIN		
ipspace 1234	RIPE					

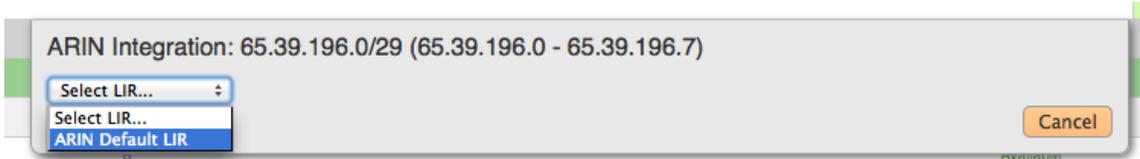
[Add LIR](#)

[Back to IPAM Admin](#)

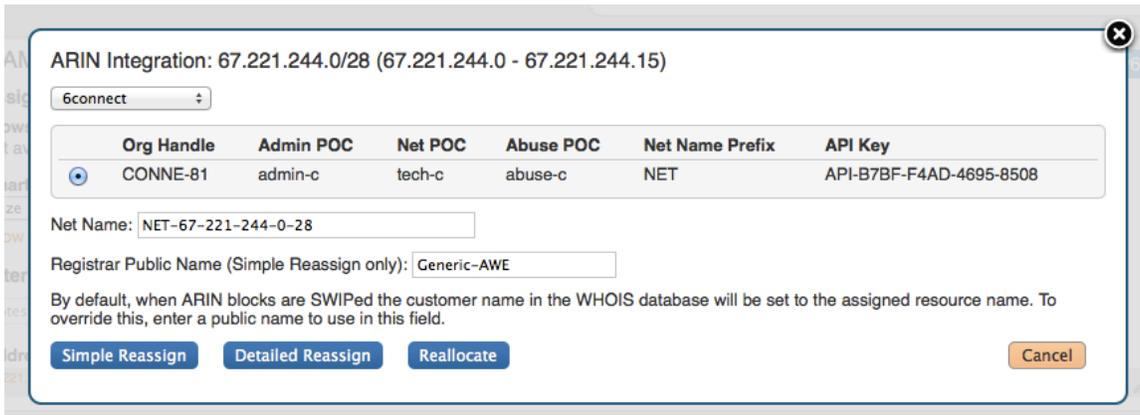
Once these have been configured, you will be able to use the **RIR integration** feature from the **Action** menu on the IPAM Manage screen or IPAM Gadget:



From the menu, you will be prompted to specify the LIR to use:



It will populate the area and then you will have the RIR specific options (see ARIN example below):



After clicking on the **Add LIR** button, you can setup the required data for the specific RIR/LIR combination:

## ARIN

### Update LIR

RIR

Name

ASN

Org Handle  ⊖ Delete

Admin POC

Tech POC

Abuse POC

NET Name Prefix

API Key

⊕ Add Org

**Update**

**i Press UPDATE to SAVE!**  
Make sure to press the Update button or else the LIR data will not save.

### Add LIR

RIR

Name

ASN

Maintainer  Delete

Password

Admin Contact

Tech Contact

+ Add Maintainer

**Update**

**i** **Press UPDATE to SAVE!**  
Make sure to press the Update button or else the LIR data will not save.

### ARIN LIR Setup and Use

#### Step 1: Setup the LIR information via the LIR Manager

You will be prompted to the select the RIR

### Add LIR

RIR

Name

ASN

Delete

**Update**

Add in the requisite Org and POC information

## Add LIR

**RIR** ARIN

**Name**

**ASN**

**Org Handle**  - Delete

**Admin POC**

**Tech POC**

**Abuse POC**

**NET Name Prefix**

**API Key**

+ Add Org

Update



### Multiple Org Support

Note that we support multiple Org Handles per ARIN entry. Simply click on the [Add Org](#) link at the bottom of the Add LIR dialog box.

**Step 2: Assign an IP block to a Resource using the IPAM Gadget or the Assign function from the IPAM Manage screen.**

### Step 3: Update SWIP information

Functions supported:



#### SWIP Update Functionality Details

In the case when a user already has SWIPped blocks to ARIN, 6connect checks prior to actually performing a SWIP. In the process, if the IP block is already SWIPped, it will check for existing ARIN customer data and update the 6connect data to reflect what ARIN has on file. Once that is complete, the user can then perform a de-SWIP function using ProVision.

### Simple Re-assign

From ARIN.net:

Used to subdelegate IP addresses to a customer that does not need to:

- subdelegate the addresses to their own customers
- maintain their own in-addr.arpa delegation
- display their own point of contact (POC) information.

It can also be used to change the customer name and address information (but not the range) on an existing simple reassignment and to remove simple reassignments. It is submitted by an ARIN Online user account linked to the parent organization's Admin or Tech POC, or the Tech POC for the resource.

### Detailed Re-assign

From [ARIN.net](#):

Used to subdelegate IP addresses to a downstream organization that does not need to further subdelegate the IP addresses, but does need to maintain its own reverse name servers and/or display separate point of contact (POC) information. It is submitted by an ARIN Online user account linked to the parent organization's Admin or Tech POC, or the Tech POC for the resource.

### Re-allocate

From [ARIN.net](#):

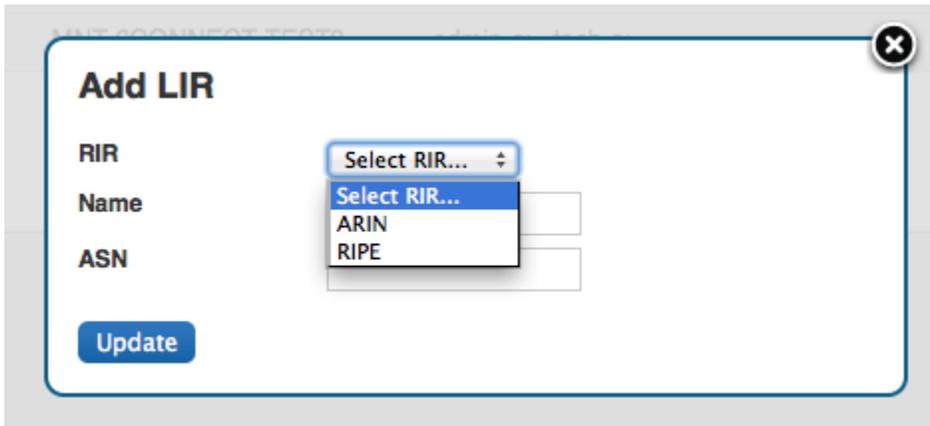
Used to subdelegate IP addresses to a downstream organization that will further subdelegate the IP addresses to their own customers. These requests must be submitted by an ARIN Online user account linked to the parent organization's Admin or Tech POC, or the Tech POC for the resource.

Once completed successfully you will see a confirmation icon with the SWIP details.

## RIPE LIR Setup and Use

### Step 1: Setup the LIR information via the LIR Manager

You will be prompted to the select the RIR:



The screenshot shows a web form titled "Add LIR" with a close button in the top right corner. On the left side, there are three labels: "RIR", "Name", and "ASN". To the right of these labels are input fields. The "RIR" field has a dropdown menu open, displaying the text "Select RIR..." at the top, followed by "ARIN" and "RIPE". Below the input fields is a blue "Update" button.

Then add in the requisite Maintainer Object related information:

**Add LIR**

RIR: RIPE

Name:

ASN:

Maintainer:  Delete

Password:

Admin Contact:

Tech Contact:

[+ Add Maintainer](#)

**Update**

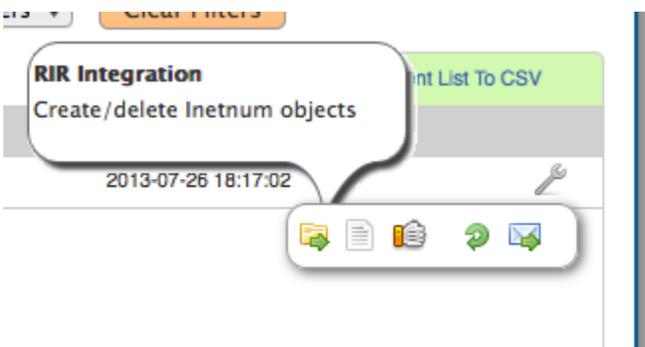
**i Multiple Maintainer Object Support**

Note that we support multiple maintainer objects per LIR entry. Simply click on the [Add Maintainer](#) link at the bottom of the Add LIR dialog box.

**Step 2: Assign an IP block to a Resource using the IPAM Gadget or the Assign function from the IPAM Manage screen.**

**Step 3: Update RPSL information**

When a block is assigned, the user (if they have permissions) can then update the block's maintainer object.



Identify which LIR data you want to use for the netnum update:

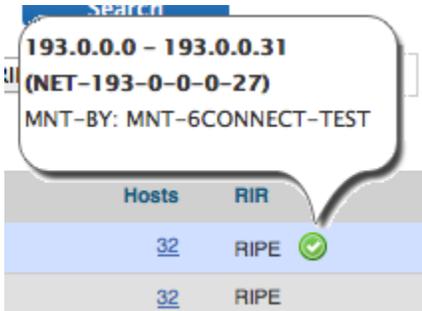
RIPE Integration: 192.162.1.0/24 (192.162.1.0 - 192.162.1.255)

RIPE Test LIR

mnt-by	admin-c	tech-c	API Key
<input checked="" type="radio"/> MNT-6CONNECT-TEST	SIXC1000-TEST	SIXC1000-TEST	

Create Inetnum Cancel

Once the RPSL update is complete, a green checkmark badge will appear next to the RIR field. When you hover over it, you will get a detailed update of the block status.



## DNS Administration

### DNS Admin

The DNS Admin tab contains 5 different functional areas: managing DNS server, performing bulk zone assignments to a resource, performing bulk record changes over all zones, managing default name server, transferring zones, and a collection of links for other useful DNS functions.

### DNS Functions

#### Defaults and Options

- Default TTL: in seconds, default value is 3600
- Default Refresh: in seconds, default value is 14400
- Default Retry: in seconds, default value is 3600
- Default Expire: in seconds, default value is 604800
- Default Minimum: in seconds, default value is 3600
- Default SOA: Server Of Authority and hostmaster contact. E.g. ns1.domain.com. hostmaster.domain.com.
- \$GENERATE IPv4 by default: Set to '1' to generate reverse IPv4 DNS hostnames for non specific PTRs. This is similar to \$GENERATE in standard bind.
- \$GENERATE IPv4 Suffix: Set to forward suffix to append to PTR for \$GENERATE Example: .available.domain.com.
- DNS Server for DNSSEC validation: required to be a non-authoritative name server.

#### Edit DNS Record Types

The "Edit DNS Record Types" will allow you to manage what types of DNS records can be added in the system. The default values are:

- A, AAAA, MX, PTR, CNAME, NS, DIRECTIVE, DNAME, DNSKEY, DS, INCLUDE, IPSECKEY, COMMENT, TXT, KEY, SOA, and SRV
- The complete list of valid record types can be found the RFCs. Wikipedia provides a nice reference: [http://en.wikipedia.org/wiki/List\\_of\\_DNS\\_record\\_types](http://en.wikipedia.org/wiki/List_of_DNS_record_types)

#### Edit DNS Delegations

#### Generate all DS records for DNSSEC

- This link will generate and output all DS records in the database. This is provided to easily bulk upload all DS keys to your domain registrar.

#### Generate zip file of all zones

- This link generates a single .zip file containing all zones for download. Once a zip file has been generated, a quick link is provided at the bottom of this section with datestamp to be downloaded later if needed.

#### Increment All Serials

- Increment all zone serial numbers by one. All zone serials are automatically incremented on a zone push, but if there is ever any other requirement for an increment, it can be performed here.

#### DynECT Zone Import

- Imports and syncs ALL zones on the system with those in your DnyECT instance. This means any zones in ProVision not present in your DynECT instance will be removed and any changes lost.

#### PowerDNS Zone Import

- Option is available after configuring a PowerDNS server with a MySQL backend. Connects to the selected server and imports all zones.

#### DNS View ACL Management

- Manage ACLs for use in DNS Views.

### DNS Zone Transfers

This section lists every server configured in the platform, along with how many zones are assigned to the server.

How to transfer zones:

- Check the boxes and click the Push the button to transfer zones to the target server.

### Manage DNS Servers

This is where you configure DNS servers to transfer zones to from the ProVision platform. ProVision currently supports the following DNS server types: BIND, PowerDNS (using a bind backend), DynECT, and Secure64. The fields available for configuring servers are as follows:

- Server - The FQDN or ip address of the DNS server.

The screenshot shows a web form titled "Manage DNS Servers". It contains the following fields and values:

- Server:** ns1-ns1.6clabs.com (with a dropdown arrow) and a "New Server" button.
- Nick Name:** ns1
- FQDN or IP:** ns1.6clabs.com (with an example: ex: ns1.dns.6connect.net or 216.239.32.10)
- Default:** Do Not Add to New Zones (with a dropdown arrow)
- Transfer Type:** ISC BIND (with a dropdown arrow)
- Server Type:** Master (with a dropdown arrow)
- SOA:** ns1.dns.6connect.net. hostmaster.6c (with an example: ex: ns1.dns.6connect.net. hostmaster.6connect.net.)

- Default - Specify if the server should be added to new zones by default or not.
- Transfer Type - SCP, Secure64, Secure64 Signer, and DynECT. Note that the SCP method should be used for PowerDNS with a Bind backend.
- Server Type - Specify if the server is a master or slave. Different configuration files are created master vs. slave on the Bind, PowerDNS/Bind, and Secure64 platforms.
- SOA - Start of Authority, should be in the format "SRI-NIC.ARPA. HOSTMASTER.SRI-NIC.ARPA.". For more information, see the RFC: <http://tools.ietf.org/html/rfc1033>
- Username - Login/username for the target DNS server. The specified account needs to be valid, and have write permission to the remote directory and execute permission for any pre/post commans.

- Password - Password for the target account. All passwords are stored encrypted in the database.
- Port - Port to contact the target server on. This is port used for SSH on Bind and Secure64 server types.
- Remote Directory - The target directory to transfer zone files to on the DNS system.
- Named Conf Path - The path to other zones on the Bind systems.
- Pre Command - Any valid system command on the target DNS system. This command will be run before any files are transferred.
- Post Command - Any valid system command on the target DNS system. This command will be run after any files are transferred. For example, on a Bind system you would need to run "rndc reload" to reload the zones.

The "Test Config" button will attempt to login to the target system and write to the target directory. If any failures are encountered, an error will be written with some detail. If the test is successful, the word "Success!" will show verifying that files can be transferred. This does not test if the user can execute pre/post commands. This needs to be checked manually.

## Views

Enable Views - Select Yes to enable views on a particular server. You must click "Update Server" to show the view options.

To enable your Bind server to use zones transferred from 6connect, you must add the following to your named.conf.

```
include "/var/named/zones/6connect_named.conf";
```

When views are enabled on a server, all zones/records attached to a server are immediately put into the default view 6connectGeneric that contains a match any rule. For example, here is a sample of the named.conf include generated by ProVision:

```
view "6connectGeneric" in {
    match-clients { any; };
    zone ...
    zone ...
};
```

All views attached to a server are displayed under the "Views" label. **When you enable views on a Bind server, you must wrap all other zones in named.conf or any includes in view statements.** The include line for the 6connect conf file should also be move above any other view statements. An example is below:

```
include "/var/named/zones/6connect_named.conf";

view "hints" {
    match-clients { any; };
    zone "." {type hint; file "named.root";};
};

view "zones-outside-of-6connect" {
    match-clients { some-acl; };
    zone ....
};
```

## Adding a View

To add a view just type in the view name, and a description (for reference only). The config files transferred to the server will automatically be built according to the server type.

<insert image>

## Adding Options to a View

## Adding ACLs to Views

You can select an existing [IP List](#) to create a view ACL. For a Bind server, this creates a corresponding line in the config: `match-clients { 6connect_Internal; }`; The `6connect_` is prefixed to all IP lists inserted by ProVision.

## Bulk Zone Assignments

The Bulk Zone Assignment function allows you to assign multiple zones to a resource in one step. The system will perform a wild card style match for any text in the search box and return all matching zones and display them in a list. You can then assign all the zones found to a resource as either a master or slave.

## Bulk DNS Changes

The Bulk DNS Editor allows an Admin to perform "find and replace" functions across all DNS zones. It will match the host and/or record type and/or record value across the entire zone database. Unless the "Strict Comparison" box is checked, it will use wildcard style matches for the host and record values.

## Nameserver Management

This function controls the list of DNS servers used for pre populating DNS records with NS records.

## Notes

General DNS configuration information is located under the main Admin tab in the DNS section. That is where you can set defaults for other SOA options, generated reverse DNS information, and a DNSSEC validation server.

## System Information for Local Installations

Zones are stored in the 6connect web root under /zones.

DS keys are stored in the 6connect web root under /keys.

# Working with DNS Zones

## Using the DNS Gadget

When you have defined a Resource, you can assign the DNS Gadget to a given Resource Type. This allows you a shortcut to DNS functionality without having to view it in the standard DNS Tab. From this interface, you can create new zones (with or without a [Zone template](#)) or assign Zone delegation specific information.



The screenshot shows a web interface for DNS management. At the top, there is a section titled "DNS". Below this, there is a "New DNS Zone" section with a text input field, a dropdown menu showing "-- no template --", and a blue "Create Zone" button. Below that is a "Zone Delegation" section with three columns: "Delegated Zone", "Slave IP", and "Customer". Each column has a text input field. The "Delegated Zone" field contains "Zone name", the "Slave IP" field contains "IPv4 or IPv6", and the "Customer" field contains "260". There is a blue "Add Slave" button to the right of the "Customer" field.

## Navigating the DNS Tab

Clicking on the main DNS Tab will bring up the following UI.

## Creating/Adding Zones

To create a zone, enter the name of the zone and select the Resource you want to assign the zone to. Click on the green plus sign to be taken to the newly created zone file. There you can edit the zone, assign views, etc.

### Create a DNS Zone

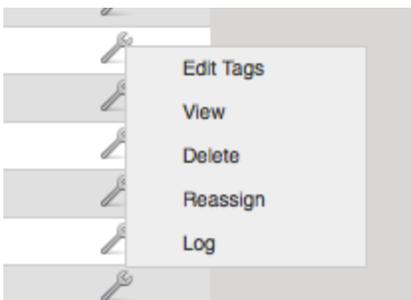
+

### DNS Tab User Interface

Zone	Customer	Tags	DNSSEC	DS	Records	Actions
<a href="#">6clabs.com</a>	<a href="#">6connect Labs</a>				12	
<a href="#">6connect.com</a>	<a href="#">6connect Available</a>				7	
<a href="#">aaron.com</a>	123 Department LAB		DNSSEC	X	11	
<a href="#">anna.com</a>	123 Department LAB				12	
<a href="#">awesome.com</a>	Anna's Test Site		DNSSEC	X	7	

- 1) **Paging** - this allows for easier browsing of large lists of DNS zones
- 2) **Filtering** - this text box allows the user to enter in criteria to filter the list of zones
- 3) The **Zone** list is a click-able list of zone names - if clicked, the user will be directed to the DNS zone editing page
- 4) This column list is a click-able list of Resource names that the zone is assigned to
- 5) The **Tags** column lists the tags associated with the zone
- 6) The **DNSSEC** column will show green if the zone has been signed and pushed successfully, the "X" column will provide a status to acknowledge that the zone was verified by an authenticated DNS server
- 7) The **Records** value is the number of zone records in the given zone

### DNS Zone Action Menu



The Action menu provides a list of options that the user can select for any given zone.

- 1) **Edit Tags:** This allows to assign tag values to a zone for easier filtering. This a free form field and not the same as the IPAM Tags
- 2) **View:** Brings you to the View/Edit screen for the zone
- 3) **Delete:** Deletes the zone from ProVision and removes the entry in ProVision conf file on the remote server(s) (the user will also receive a prompt to confirm they wish to complete the action)
- 4) **Delete & Push:** Deleted the zone from ProVision, removes the entry in ProVision conf file on the remote server(s) **AND** deletes the individual zone file from the remote server(s) (the user will also receive a prompt to confirm they wish to complete the action)
- 5) **Reassign:** Brings up a screen to assign the zone to a new Resource
- 6) **Log:** Brings the user to the Log Tab with the results filtered for the specific zone

### Editing DNS Zones

## Editing a Zone Record

There are two ways to edit a DNS zone:

1. Click on the "Edit Zone" icon. This will take you directly to the Zone Editing screen.
2. Click on the zone name. At the Zone Detail View screen, you can click on the "Edit this zone" hyperlink.

### 1) Zone Management:

This area is at the top of the screen and provides direct access to confirm zone file changes. By clicking the "Check Zone" button, we automatically confirm that your zone is verified and highlight any problem entries. Once verified, you have the option to Push the Zone to the specified server(s) selected.

**\*Note: When zones are written the serial number is incremented and DNSSEC refreshed (if enabled)**



Figure 1: Normal zone with no errors

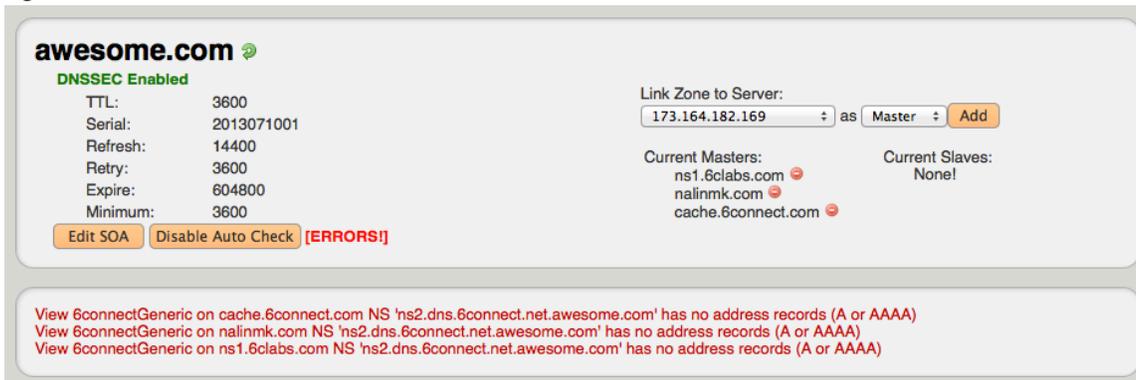


Figure 2: Zone with Errors

If errors are detected, the relevant zone record entries will be highlighted to show the error condition and the user will be prompted to fix them before being able to push the zone. The validation is for RFC compliance.

### 2) DNS Zone Record Data:

You have two modes for viewing/editing Zone Record Data. The **Verbose** view and a **BIND** view allow for varying levels of comfort with DNS editing tools. The **Search** window also allows the user to filter the list by using multiple parameters.

**DNS Records** View:  Verbose  BIND

Search by Record, Value, etc

- 1 **COMMENT** maps to update **A** record based on **turnup date** TTL 3600
- 2 **NS** @ maps to **dns2.mycloud.net.** TTL 3600
- 3 **NS** @ maps to **dns3.mycloud.net.** TTL 3600
- 4 **NS** **amazon.com.** maps to **ns1.dns.6connect.net.** TTL 3600 Automatically Added
- 5 **NS** **amazon.com.** maps to **ns2.dns.6connect.net.** TTL 3600 Automatically Added
- 6 **NS** **amazon.com.** maps to **ns3.dns.6connect.net.** TTL 3600 Automatically Added
- 7 **NS** **amazon.com.** maps to **ns1.dns.bind.com.** TTL 3600 Automatically Added
- 8 **MX** maps to **mx.mycloud.net.** with priority 10
- 9 **MX** maps to **mx2.mycloud.net.** with priority 20
- 10 **A** **veggie.com.** maps to **1.2.3.4** TTL 3600
- 11 **A** **www** maps to **1.2.3.** TTL 3600  
Record value must be an IPv4 address (ex: 127.0.0.1).
- 12 **AAAA** **veggie.com.** maps to **2001:db7::1** TTL 3600
- 13 **AAAA** **www** maps to **2001:db8:** TTL 3600  
Record value must be an IPv6 address (ex: 2001:db10:2001::4).

Add a New  Record

To Edit a Zone Record, simply double-click on it the entry and make any required edits. Use the "Wrench" icon for the action context menu to:

- 1) **Save** your changes to the zone records
- 2) **Delete** the zone record
- 3) **Cancel** your edits to the zone record

Type	Record	Value	Description	TTL	
NS	awesome.com.	ns2.dns.6connect.net.	Automatically Add	3600	
3 A	www maps to 12.12.12.12			TTL 3600	
4 NS	awesome.com. maps to ns3.dns.6connect.net.			TTL 3600	Automatic

Context menu: Save, Delete, Cancel

### Configuring Views per DNS Zone

If Views are enabled on the DNS server assigned to this zone, you will also have the "Glove" icon that will bring up a view assignment menu. You will be able to select the View(s) that you wish to apply to the zone record here.

Type	Record	Value	Description	TTL	
NS	awesome.com.	ns2.dns.6connect.net.	Automatically Add	3600	
<b>DNS Views:</b>					
	ns1.6clabs.com				
	nalinmk.com				

View selection dropdown: All Views, Local Only

For more information on setting up Split Horizon/Views support - go [here](#).

### 3) Show DNS Zone:

This view gives you a "CLI type" view of the zone file. If views are enabled, you will see those zone files as well. Please note that this is a read only screen.

```

Hide Zone File

$TTL 3600
@ IN SOA ns1.dns.6connect.net. hostmaster.6connect.net. (
    12092501 ; Serial
    14400 ; Refresh
    3600 ; Retry
    604800 ; Expire
    3600 ) ; Minimum

; This zone was auto-generated by 6connect, Inc., ProVision.

@ IN COMMENT update A record based on turnup date
@ IN NS dns2.mycloud.net.
@ IN NS dns3.mycloud.net.
amazon.com. 3600 IN NS ns1.dns.6connect.net.
amazon.com. 3600 IN NS ns2.dns.6connect.net.
amazon.com. 3600 IN NS ns3.dns.6connect.net.
amazon.com. 3600 IN NS ns1.dns.bind.com.
IN MX 10 mx.mycloud.net.
IN MX 20 mx2.mycloud.net.
veggie.com. IN A 1.2.3.4
www IN A 1.2.3.
veggie.com. IN AAAA 2001:db7::1
www IN AAAA 2001:db8:

```

#### 4) Show DS Records:

This section displays the DS keys generated for the particular zone.

#### 5) Show Zone History:

The feature allows you to revert/reload previous zone versions. Note that the zone has to actually be pushed for the Zone History area to show up on the screen.



## Configuring ISC BIND Support

### Getting Started

You will need a user who can log in to the DNS server and make changes to the directory in which the zones are being stored. Additionally, it is often useful for this user to have the ability to restart the DNS server. The login and password for this user will be required to configure this server on the DNS Admin page.

6connect Zone files are written out in the following format:

```
/path/to/zone/directory/viewName/zoneFirstLetter/zonefile.zone
```

If no views are configured, or if views are expressly disabled, then the default viewName "6connectGeneric" is used. The zoneFirstLetter is the first letter of the zone name, so the subdirectory 'microsoft.com.zone' is placed in would be /m/.

All 6connect-managed Zones are managed by a dedicated 6connect configuration file named 6connect\_named.conf. This file is created to act a supplementary conf file to work in concert with any existing named.conf which might exist. To include the 6connect configuration file, edit named.conf and append the following line:

```
include "/path/to/conf/directory/6connect_named.conf";
```

You must remember to include the 6connect configuration file or none of the changes managed by 6connect ProVision will take effect!

It is also important to note that if your existing named.conf file contains zones within Split Horizon views, then the 6connect-managed zones must also be view-enabled. Likewise, if existing zones are not grouped into views, then views must be disabled on ProVision.

## Configuring DynECT Support

To use ProVision with DynECT support, first enter your Dyn username, password, and customer name into the New Server dialogue on the DNS Admin page.

Additionally, if you are deploying any DNSSEC-enabled zones, you will also need to provide a valid DynECT DNSSEC contact. See Dyn documentation for details on DNSSEC contacts.

Once ProVision begins managing DynECT zones, only the ProVision tool should be used to make and manage changes to zones. If zone changes are made to DynECT directly they will be overwritten the next time ProVision syncs, causing errors. Only edit zones using ProVision.

## Configuring PowerDNS Support

### Environments supported

PowerDNS version 3.0 or above on the target server(s)

Either BIND or MySQL backend

### Overview



### Step 1: Setup your PowerDNS Server

### Manage DNS Servers

Server:	<input type="text" value="208.39.104.106"/>	<input type="button" value="New Server"/>
Default:	<input type="text" value="Add to New Zones"/>	
Transfer Type:	<input type="text" value="PowerDNS"/>	
Server Type:	<input type="text" value="Master"/>	
Backend Type:	<input type="text" value="MySQL"/>	
SOA:	<input type="text" value="ns1.dns.6connect.net. hostmaster.6c"/> ex: ns1.dns.6connect.net. hostmaster.6connect.net.	
Username:	<input type="text" value="6connect"/>	
Password:	<input type="password" value="....."/>	
DB Username:	<input type="text" value="pdns"/>	
DB Password:	<input type="password" value="....."/>	
DB Port:	<input type="text" value="3306"/>	
DB Name:	<input type="text" value="powerdns"/>	
	<input type="button" value="Update Server"/>	<input type="button" value="Delete Server"/>

### Manage DNS Servers

Server:  **New Server**  
 Default:   
 Transfer Type:   
 Server Type:   
 Backend Type:   
 SOA:  ex: ns1.dns.6connect.net. hostmaster.6connect.net.  
 Username:   
 Password:   
 Port:   
 Remote Directory:   
 Named Conf Path:   
 Pre Command:   
 Post Command:   
**Update Server** **Delete Server**

**Step 2: Import your PowerDNS zones**

This operation will pull all zones on the target server.

This operation may take quite some time.

Choose a server:

**Step 3: Edit/Push your zones to PowerDNS**

### DNS Zone Transfers:

trace.bind.com	4 Zones	<input type="checkbox"/>
208.39.104.106	34 Zones	<input checked="" type="checkbox"/>

**Push Zones to Checked Servers:**

**BIND Backend**

**Note on SSH**  
 The integration does not require a remote database connection, but it does require an SSH account and a writable directory. The SSH account must have access to the server. This account will also be used for DNSSEC functionality within PowerDNS.

**MySQL Backend**



#### Note on SSH

The integration requires a remote database connection, so will need a mysql user with permissions for remote administration. We highly recommend using ACLs to ensure that configuration only occurs from intended sources.

For DNSSEC functionality, you will need a standard SSH user account withing your PowerDNS user group

Please note that Views are not supported with the MySQL backend



#### Database

The integration supports MySQL Backend with Postgres Backend and GenericDB coming soon

## Configuring Secure64 Support



#### A note on Ports

6connect uses port 22 to communicate with Secure64 infrastructure - please ensure that this is addressed in any ACLs/firewalls

The initial setup of the Secure64 Authoritive server is as follows:

### Step 1: Create an nsd.conf file under the root directory / of your S64 Auth server



#### DO THIS

Make sure to add the line include: 6connect\_nsd.conf to the nsd.conf file

#### Output/Input

```
[authdnsadmin@Secure64DNS]# cat nsd.conf
server:
ip-address: 50.198.192.138

axfr-logfile: /axfr_log/axfr.log
axfr-logfile-flush-count: 1
axfr-logfile-max-size: 100000
axfr-logfile-max-size: 10

request-logfile: /request_log/request.log
request-logfile-flush-count: 10
request-logfile-max-size: 1000000
request-logfile-max-files: 10

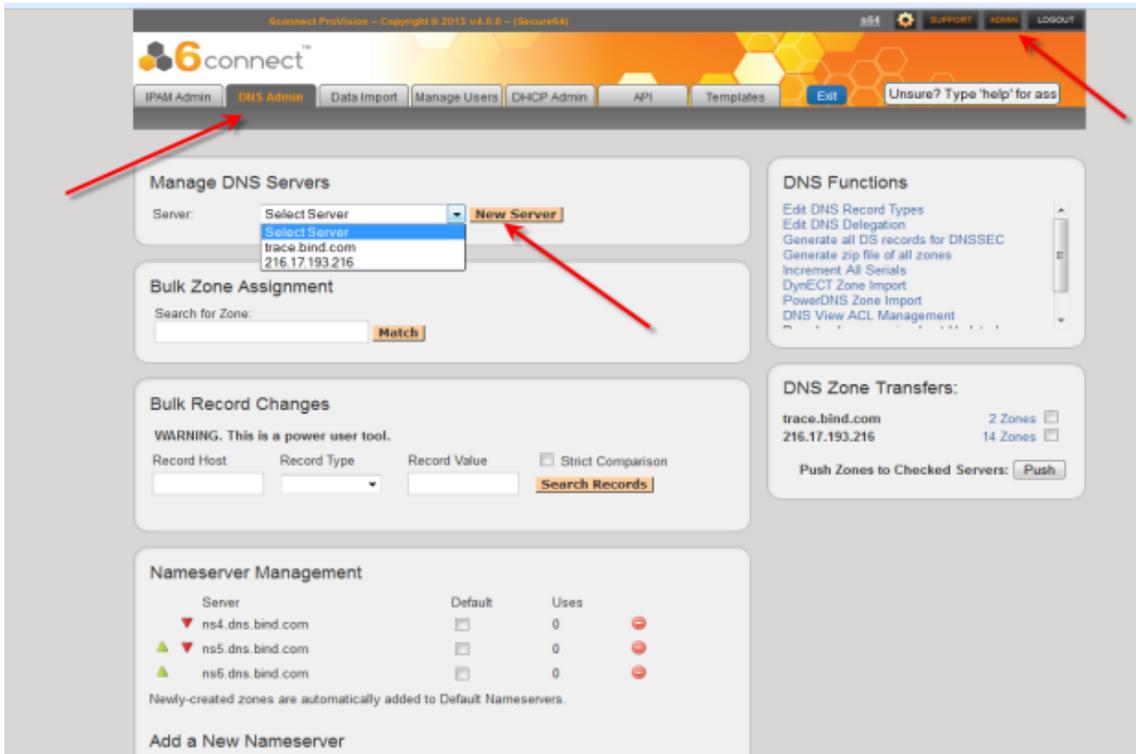
include: 6connect_nsd.conf
```

### Step 2: Make a directory for 6connect ProVison to push zone files to on the Secure64 DNS Server

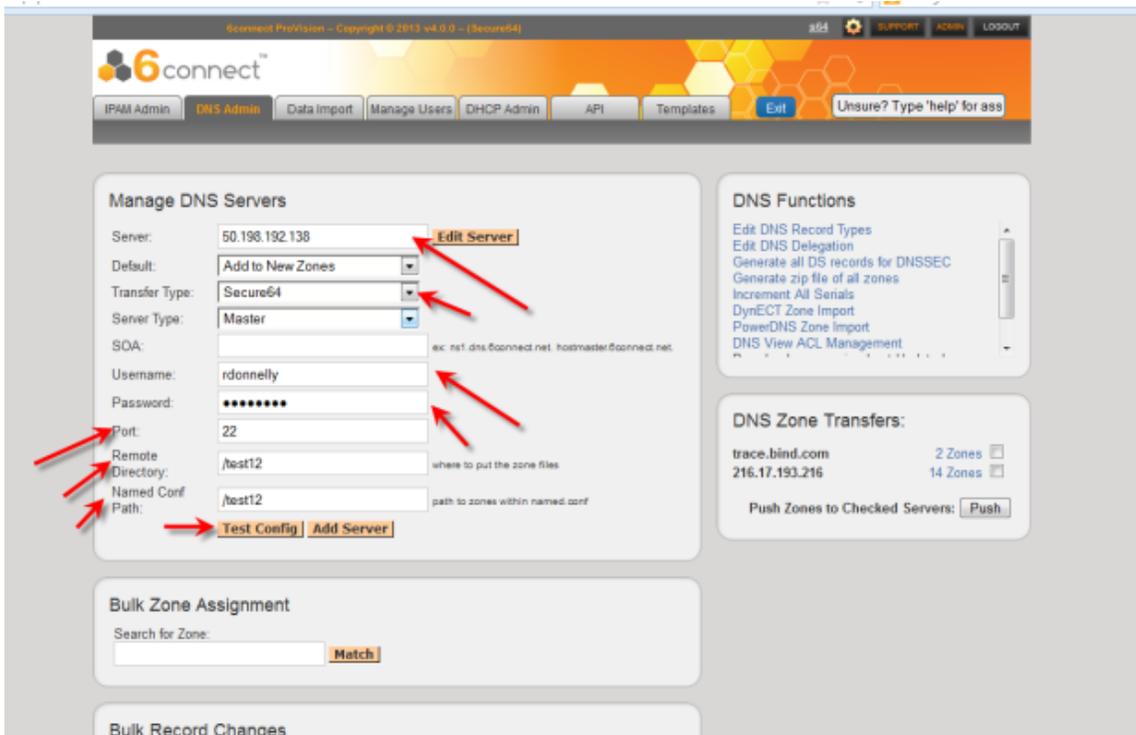
```
[authdnsadmin@Secure64DNS]# mkdir test12
[authdnsadmin@Secure64DNS]# ls
/:
322 2013-08-19 06:07:42 nsd.conf
<DIR> 1024 2013-08-16 17:30:12 test12
```

### Step 3: Setup and Configure 6connect ProVision for your Secure64 DNS Server

Go to the 6connect Admin area and click on the **DNS Admin** Tab. Click on the **New Server** button.



Then fill in the information as follows (including any relevant SOA information):



### Step 4: Test the Secure64 DNS Server configuration

Press the **Test Config** button for the DNS Server you setup.

6connect™

IPAM Admin **DNS Admin** Data Import Manage Users DHCP Admin API Templates [Exit](#) Unsure? Type 'help' for ass

### Manage DNS Servers

Server:  [Edit Server](#)

Default:

Transfer Type:

Server Type:

SOA:  ex: ns1.dns.6connect.net. hostmaster@6connect.net.

Username:

Password:

Port:

Remote Directory:  where to put the zone files

Named Conf Path:  path to zones within named.conf

[Test Config](#) [Add Server](#)

Success!

### DNS Functions

- Edit DNS Record Types
- Edit DNS Delegation
- Generate all DS records for DNSSEC
- Generate zip file of all zones
- Increment All Serials
- DynECT Zone Import
- PowerDNS Zone Import
- DNS View ACL Management

### DNS Zone Transfers:

trace.bind.com	2 Zones	<input type="checkbox"/>
216.17.193.216	14 Zones	<input type="checkbox"/>

Push Zones to Checked Servers: [Push](#)

Success! Will show as depicted above.

Click **Add Server** to add this server as a permanent entry in the dropdown menu. This server will now be available for assigning DNS zones to.

### Step 5: Assign any imported/existing zones to your Secure64 DNS Server(s)

Search for all available zones or enter in a value to find specific existing zones in the system.

6connect™

IPAM Admin **DNS Admin** Data Import Manage Users DHCP Admin API Templates [Exit](#) Unsure? Type 'help' for ass

### Manage DNS Servers

Server:  [New Server](#)

### Bulk Zone Assignment

Search for Zone:  [Match](#)

Matched Zones:

- [epc.mnc016.mcc502.3gppnetwork.org](#)
- [mnc0016.mcc502.gprs.name4](#)
- [mnc016.mcc502.gprs.APN](#)

Assign to:  as  [Assign](#)

### Bulk Record Changes

WARNING. This is a power user tool.

Record Host	Record Type	Record Value	<input type="checkbox"/> Strict Comparison
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>

[Search Records](#)

### DNS Functions

- Edit DNS Record Types
- Edit DNS Delegation
- Generate all DS records for DNSSEC
- Generate zip file of all zones
- Increment All Serials
- DynECT Zone Import
- PowerDNS Zone Import
- DNS View ACL Management

### DNS Zone Transfers:

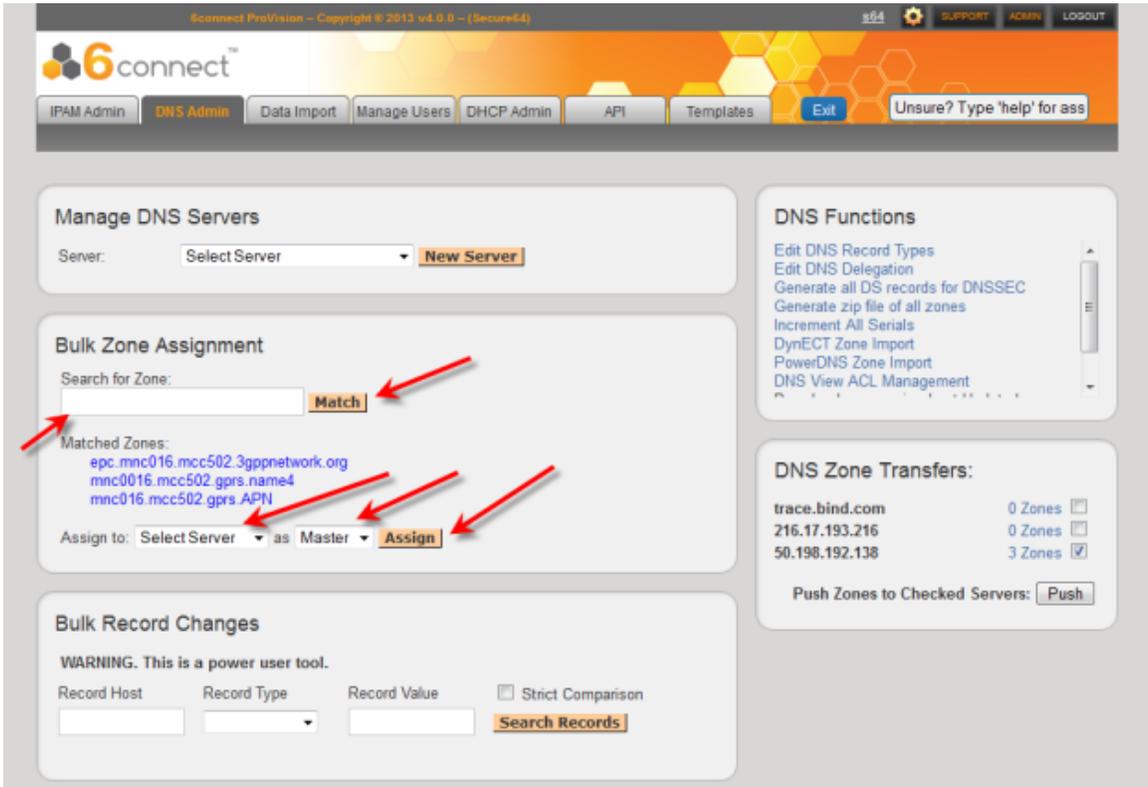
trace.bind.com	0 Zones	<input type="checkbox"/>
216.17.193.216	0 Zones	<input type="checkbox"/>
50.198.192.138	3 Zones	<input checked="" type="checkbox"/>

Push Zones to Checked Servers: [Push](#)



#### Search Tip

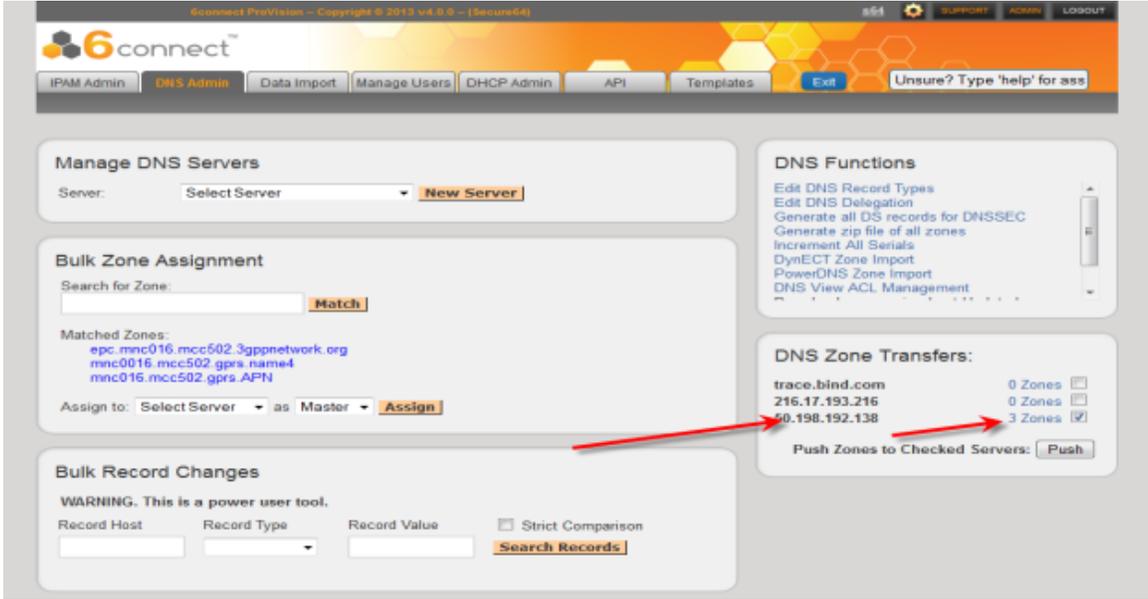
No character in the search area indicates a search for all zones as shown below



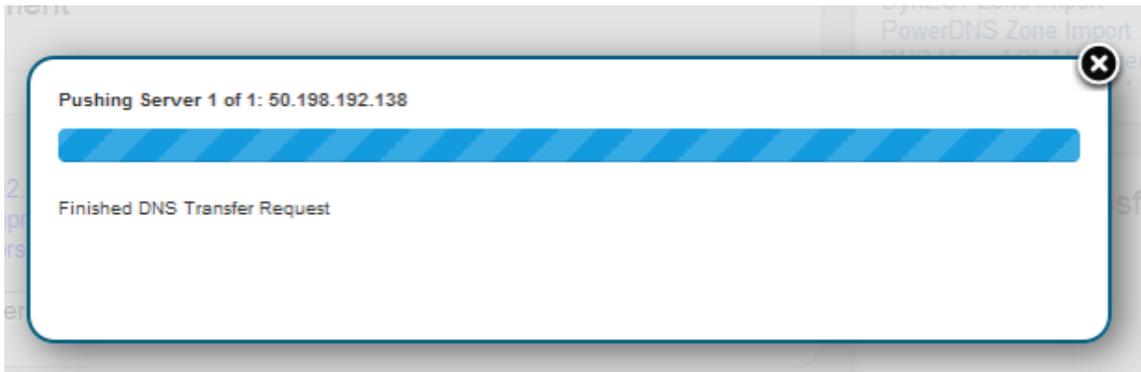
Select the **Select Server** and as **Master** dropdowns and **Assign** the above zones to this server. Verify the DNS Zones Transfers area indicates your server and the # of zones to transfer.

**Step 6: Push Zones to Secure64 Server(s)**

Check the 3 Zones box and click on the Push button to transfer zones to this server.



The system will present the following live progress bar and show as follows when it is finished without errors.



Towards the bottom of the progress status will be the final indication of success or errors to correct.

### Step 7: Verify DNS Zone push on Secure64 Server(s)

The result of the Push can be checked/verified by checking the Secure64 server as follows:



#### Verifying Zone pushes

```
ssh to 50.198.192.138
Login using the designated login account and password
Enable cachednsadmin
ls
```

Now, verify that the "788 2013-08-21 12:35:04" 6connect\_nsd.conf file now exists.

```
[authdnsadmin@eval138.secure64.com]# ls
/:
6728 2013-08-13 00:15:30 nsd.conf
8416071 2013-08-21 12:35:07 nsd.db
788 2013-08-21 12:35:04 6connect_nsd.conf
<DIR> 1024 2013-08-21 12:34:50 test12
```

You can verify the Push contents by doing a cat of the 6connect\_nsd.conf



```
[authdnsadmin@Secure64DNS]# cat 6connect_nsd.conf
AutoGenerated by 6connect ProVision. Do not manually edit.

zone:

name: mnc016.mcc502.gprs.APN

zonefile: /test12/6connectGeneric/m/mnc016.mcc502.gprs.APN.zone

zone:

name: mnc0016.mcc502.gprs.name4

zonefile: /test12/6connectGeneric/m/mnc0016.mcc502.gprs.name4.zone

zone:

name: epc.mnc016.mcc502.3gppnetwork.org

zonefile: /test12/6connectGeneric/e/epc.mnc016.mcc502.3gppnetwork.org.zone
```

In the example above, three Zones have transferred.

To look at the contents of each zone you can cd to the proper directory /test12/6connectGeneric and find the zone files in an alphabetical directory structure as follows:

```
[authdnsadmin@Secure64DNS]# cd 6connectGeneric
[authdnsadmin@Secure64DNS]# cd test12

changed to test12
[authdnsadmin@Secure64DNS]# ls
/test12/:
<DIR> 1024 2013-08-16 19:43:21 6connectGeneric
[authdnsadmin@Secure64DNS]# cd 6connectGeneric
changed to 6connectGeneric
[authdnsadmin@Secure64DNS]# ls
/test12/6connectGeneric/:
<DIR> 1024 2013-08-16 17:30:13 e
<DIR> 1024 2013-08-16 17:30:16 m
<DIR> 1024 2013-08-16 18:49:21 d
<DIR> 1024 2013-08-16 19:43:23 s
[authdnsadmin@Secure64DNS]# cd m
changed to m
[authdnsadmin@Secure64DNS]# ls
/test12/6connectGeneric/m/:
[authdnsadmin@eval138.secure64.com]# ls
5192 2013-08-21 15:35:01 mnc0016.mcc502.gprs.name4.zone
6758 2013-08-21 15:35:02 mnc0016.mcc502.gprs.APN.zone
284 2013-08-21 15:34:11 m2m.mnc0016.mcc502.zone
[authdnsadmin@Secure64DNS]#
```

## Step 8: Validate Zone data in Your Infrastructure

Finally, do a **dig** of the zones to verify the DNS configuration has been successfully deployed.

```
i Using dig to validate your Secure64 Server installation
[authdnsadmin@eval138.secure64.com]# dig @50.198.192.138 mnc0016.mcc502.gprs.name4
; <<<>> DiG SourceT 3.x <<<>> @50.198.192.138 mnc0016.mcc502.gprs.name4
;; Got answer:
;; >>>HEADER<<< opcode: QUERY, status: NOERROR, id: 59591
;; flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0
;; QUESTION SECTION:
;mnc0016.mcc502.gprs.name4. IN A
;; AUTHORITY SECTION:
mnc0016.mcc502.gprs.name4. 3600 IN SOA ns1.dns.6connect.net. hostmaster.6connect.net. (2013082102 10800 3600 604800 38400
)
[authdnsadmin@eval138.secure64.com]#
```

For any questions regarding the integration of Secure64 products into 6connect ProVision, please email 6connect at [support@6connect.com](mailto:support@6connect.com), or Secure64 at [support@secure64.com](mailto:support@secure64.com)

## Configuring DNSSEC



### DNSSEC Implementation

How enable DNSSEC (per zone)

First, we check to see if the signed zone exists, then:

- If it does, archive the existing keys and update the signature for 31536000 seconds (or 1 year)
- If the keys do not exist, sign new keys and create them.

#### For BIND

Coming soon

#### For DynECT

Coming soon

#### For Secure64 and PowerDNS

#### DNSSEC Signatures

In this scenario, 6connect ProVision uses the DNSSEC signing functions of the respective environment we write the zones to.

## Configuring Split Horizon/Views

video coming soon

#### WARNING

If you see a view named "\_6connectDefault" - DO NOT DELETE IT.

### Create a List in the List manager

In the Admin screen, go to the Data Import Tab and click on the "List Management" button. You will be presented with the options to **Create a New List** and also **Manage Lists**. To create a list, enter in the descriptive information and ensure that the **Code** dropdown is marked "IPLIST".

Name	Code	Description
<input type="text" value="Internal Dev"/>	<input type="text" value="IPLIST"/>	<input type="text" value="Dev ACL - RFC 1918"/>

Press the **Eye** icon and you will be presented with an editing area to populate IP data including an option for the data delimiter (you can also do this from the **Manage Lists** section). Click on the **Pencil** icon to save your changes, the List will then be moved to the **Manage Lists** section below.

Name	Code	Description
<input type="text" value="Internal Dev"/>	<input type="text" value="IPLIST"/>	<input type="text" value="Dev ACL - RFC 1918"/>
<b>Initial Population</b>		
<b>Delimiter:</b> <input type="text" value="[space]"/>		
<input type="text" value="192.168.1.0/24 10.10.1.0/24"/>		

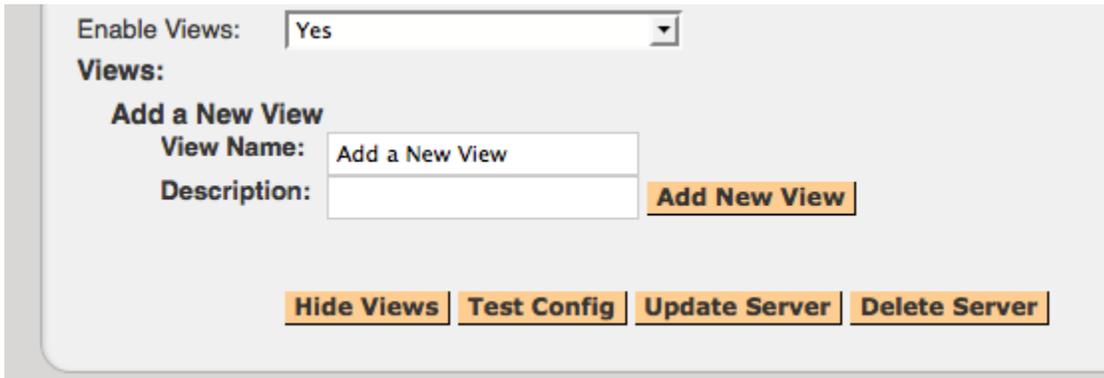
The List will now be available from the **Manage Lists** display area and can now be assigned to a Server View.

Manage Lists			
Name	Code	Description	Actions
Internal Dev	IPLIST	Dev ACL - RFC 1918	  
<b>Item Display</b>	<b>Item Value</b>	<b>Actions</b>	
	192.168.1.0/24	   	
	10.10.1.0/24	   	
<input type="text"/>	<input type="text"/>		

## Define and Assign a View to the DNS Server

In the Admin screen, go to the DNS Admin Tab.

With a DNS server selected and Enable Views marked "Yes", you will then have the option to define a View.



Enable Views: Yes

**Views:**

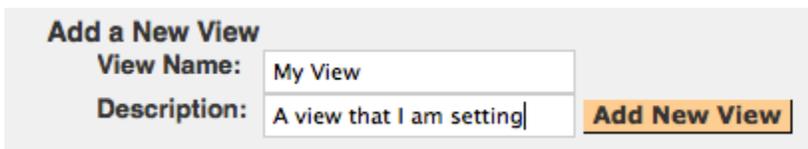
**Add a New View**

View Name: Add a New View

Description: Add New View

Hide Views Test Config Update Server Delete Server

Enter identifying information for the View you are creating and click the "Add New View" button.

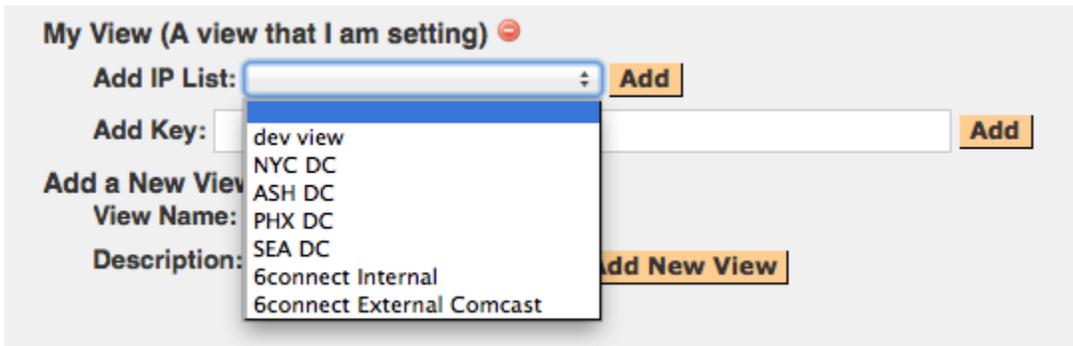


**Add a New View**

View Name: My View

Description: A view that I am setting Add New View

Once the View is created, you can select the IP List that you want to assign to this View by pressing the "Add" button.



**My View (A view that I am setting)**

Add IP List: Add

Add Key: Add

**Add a New View**

View Name: Add New View

Description: Add New View

- dev view
- NYC DC
- ASH DC
- PHX DC
- SEA DC
- 6connect Internal
- 6connect External Comcast

## Assigning other Directives

With the IP List assigned, you can either assign additional Key/Value pairs or add another IP List to apply to the View.



**My View (A view that I am setting)**

Included IPs: 6connect Internal

Add IP List: Add

Add Key: Val: Add



### A Note on Directives

For example, if you wanted to allow recursion, you would simply enter "allow-recursion" as a Key, with a Value of "on".

## Assign a View to a DNS Zone Record

When viewing a DNS Zone, ensure that the Zone is linked to a the server with a DNS View enabled. Double-click on the zone record to edit it.

Click on the **Glove** icon and it will bring up the DNS Views menu where you can select the View to apply to the zone record. Click on the **Pencil** icon for the View and the **Pencil** icon for the Zone record to make sure all changes are saved.

4 NS awesome.com. maps to ns1.dns.bind.com. TTL 3600 Automatically Added

Type	Record	Value	Description	TTL
A	www	12.12.12.12		

DNS Views: 173.164.182.169

- All Views
- All Views
- My View

6 A mail maps to 11.11.11.11

Push the zone out like normal and the View should be applied as expected. You can also preview the zone from the "Show Zone" area of the screen that will be visible once you push the zone out successfully. This will also display the History for the zone if a rollback is necessary.

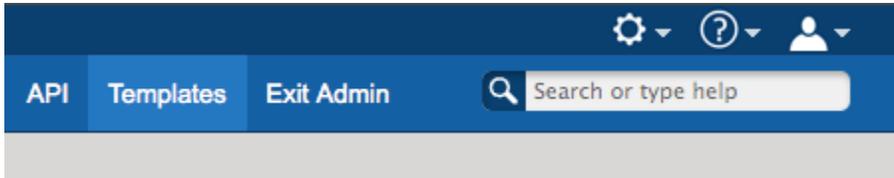
## Configuring DNS Templates

### Overview

When creating a new DNS zone, the user can specify a zone template to use. Templates are setup from the Admin -> Templates Tab.

### Configuring DNS Templates

Go to the Templates Tab in the Admin Menu



The Admin can either create a new template or edit an existing template as listed:

Name	Records	Created By	Modified
Anna's Template	7	ops@6connect.com	2013-05-07 12:20:35
Demo Template	2	pete@6connect.com	2012-08-21 12:38:14

When editing a DNS template, the Admin can specify the data in the fields below:

Editing Demo Template

Name  
Demo Template

SOA Record  
ns1.dns.6connect.net. hostmaster.6connect.net.

Serial	Refresh	Retry	Expiry	Minimum
	14400	3600	604800	3600

Zone record data is specified and can be added/deleted/re-ordered via the icons on the right.

Zone Records

Host:	TTL	Type	Priority	Value
1.2.3.4		A		cnn.com.
8.8.8.8		A		www

Save Template Cancel

As the admin edits entries in the Template screen, the window below will be updated to show the zone file.

```

@           IN      SOA     ns1.dns.6connect.net. hostmaster.6connect.net. (
                <SERIAL> ; serial
                14400   ; refresh
                3600    ; retry
                604800  ; expire
                3600    ; minimum
                )
1.2.3.4     IN      A       cnn.com.
8.8.8.8     IN      A       www

```

## Using DNS Templates

From the DNS Gadget - select the DNS Template from the dropdown that you would like to use.

The screenshot shows the 'DNS' configuration page. At the top, there is a 'New DNS Zone' section with a text input field and a dropdown menu currently set to '-- no template --'. A 'Create Zone' button is to the right. Below this is the 'Zone Delegation' section, which includes a 'Delegated Zone' input field, a 'Slave IP' input field, and an 'Add Slave' button. The dropdown menu is open, showing the following options: '-- no template --', 'Anna's Template', 'Demo Template' (which is highlighted), 'Equinix', 'testing', and 'VM Turnup'.

## DHCP Administration

### Managing Server Configurations

You have the ability to store and manage multiple DHCP server configurations from this interface. The "Push All Configs" link will push out configurations for all DHCP servers.

### Managing DHCP Servers

This section allows you to manage the specific configuration for each DHCP server specified. We support standard commands as specified by the Server Type dropdown. To save the configuration, you need to press the "Add Server" button.

The screenshot shows the configuration page for a DHCP server named 'dhcp2.6connect.com'. The form is organized into two columns. The left column contains: 'Server' (dhcp2.6connect.com), 'Username' (anotheradmin), 'Server Type' (ISC), 'Server Config Path' (/usr/bin/dhcpd/config), 'Server Stop Command' (dhcpd stop), 'Max Lease Time' (2678400), 'Domain Names' (6connect.net), and 'Authoritative' (checked). The right column contains: 'SSH Port' (22), 'New Password' (empty), 'Test Login' button, 'Server Start Command' (dhcpd start), 'Default Lease Time' (28800), 'Name Servers' (dns2.6connect.com, dns3.6connect.c), and 'Log Facility' (empty). At the bottom right, there are 'Use Config File' and 'Save Changes' buttons.

### Managing DHCP Configurations

Once the DHCP server is saved, you now have options for configuration. We provide a standard "config builder" as well as a "config file" option.

#### Configuration Builder

1) The "config builder" builds the ISC configuration file based on the parameters you select - namely the subnets and hosts to be managed by the given DHCP server.

The subnet configuration screen allows for the following parameters:

The host configuration screen allows for the following parameters:

Example configuration

The screenshot shows a configuration interface for a DHCP server named 'dhcp2.6connect.com'. The interface is divided into several sections:

- Server Information:** Server: dhcp2.6connect.com, Username: anotheradmin, Server Type: ISC, Server Config Path: /usr/bin/dhcpd/config, Server Stop Command: dhcpd stop, Max Lease Time: 2678400, Domain Names: 6connect.net, Authoritative:
- SSH Settings:** SSH Port: 22, New Password: (empty), Test Login button
- Commands and Timing:** Server Start Command: dhcpd start, Default Lease Time: 28800
- Name Servers:** dns2.6connect.com, dns3.6connect.c
- Log Facility:** (empty)
- Buttons:** Use Config File, Save Changes, Push
- Subnets on Server:**
  - 192.168.1.0 / 24: 255 IPs Assigned, 1% of Total Available. Options: range 192.168.1.0 192.168.1.255
  - 192.168.2.0 / 24: 255 IPs Assigned, 1% of Total Available. Options: range 192.168.2.0 192.168.2.255
- Hosts on Server:**
  - server1.6connect.com: Options: hardware ethernet 00-AE-32-EE-43-56-FC, fixed-address 192.168.1.145
  - email.6connect.com: Options: hardware ethernet 00-AE-32-AE-33-57-FC, fixed-address 192.168.1.146

## Custom Configuration File

2) The "config file" option allows you to paste a completely customized DHCP configuration file.



### WARNING

Please note that the "config file" option will override all entered server information, including subnets and hosts, for the designated server.

example config

dhcp.6connect.com Push

Server: <input type="text" value="dhcp.6connect.com"/>	SSH Port: <input type="text" value="22"/>
Username: <input type="text" value="6connectadmin"/>	New Password: <input type="password"/>
Server Type: <input type="text" value="ISC"/>	<input type="button" value="Test Login"/>
Server Config Path: <input type="text" value="/dhcpd/config"/>	
Server Stop Command: <input type="text" value="dhcp stop"/>	Server Start Command: <input type="text" value="dhcpd start"/>
Max Lease Time: <input type="text"/>	Default Lease Time: <input type="text"/>
Domain Names: <input type="text" value="dhcp.6connect.com"/>	Name Servers: <input type="text" value="dns1.6connect.com"/>
Authoritative: <input checked="" type="checkbox"/>	Log Facility: <input type="text"/>

Using a DHCP Config text config file will override all entered server information, including subnets and hosts, for this server.

```
# 02-Feb-2012 09:02:17

server-name "dhcp.6connect.com";
option option-252 code 252 = text; option option-252 "http://dhcp.6connect.com/wpad.datn";
option domain-name "6connect.com";
authoritative ;#;
option netbios-name-servers 132.220.21.9, 132.220.21.17;
option space LWAPP ; option LWAPP.controller code 241 = string;
use-host-decl-names off;
boot-unknown-clients on;
default-lease-time 28800;
max-lease-time 2678400;
ddns-update-style interim;
deny client-updates;
ddns-hostname = pick-first-value(config-option host-name, binary-to-ascii(16, 16, "", leased-address));
update-static-leases off;
option option-128 code 128 = text;
```

## Configuring ISC dhcpd Support

### Managing DHCP Configuration - ISC dhcpd

Once a DHCP server is saved, you now have options for configuration. We provide a standard "**config builder**" as well as a "**config file**" option.

1) The "**config builder**" builds the ISC configuration file based on the parameters you select - namely the subnets and hosts to be managed by the given DHCP server.

The subnet configuration screen allows for the following parameters:



The host configuration screen allows for the following parameters:



Example configuration:

**dhcp2.6connect.com** Push

Server: 
 SSH Port:

Username: 
 New Password:

Server Type:

Server Config Path: 
 Server Start Command:

Server Stop Command: 
 Default Lease Time:

Max Lease Time: 
 Name Servers:

Domain Names: 
 Log Facility:

Authoritative:

**Subnets on Server**

192.168.1.0 / 24 
  
 255 IPs Assigned, 1% of Total Available
   
 Options: 
  
 range 192.168.1.0 192.168.1.255

192.168.2.0 / 24 
  
 255 IPs Assigned, 1% of Total Available
   
 Options: 
  
 range 192.168.2.0 192.168.2.255

**Hosts on Server**

server1.6connect.com 
  
 Options: 
  
 hardware ethernet 00-AE-32-EE-43-56-FC 
  
 fixed-address 192.168.1.145

email.6connect.com 
  
 Options: 
  
 hardware ethernet 00-AE-32-AE-33-57-FC 
  
 fixed-address 192.168.1.146

2) The "config file" option allows you to paste a completely customized DHCP configuration file.



**Pro-Tip!**

Please note that the "config file" option will override **ALL** entered configuration information, including subnets and hosts, for the designated server.

Example configuration:

**dhcp.6connect.com** Push

Server: 
 SSH Port:

Username: 
 New Password:

Server Type:

Server Config Path: 
 Server Start Command:

Server Stop Command: 
 Default Lease Time:

Max Lease Time: 
 Name Servers:

Domain Names: 
 Log Facility:

Authoritative:

Using a DHCP Config text config file will override all entered server information, including subnets and hosts, for this server.

```
# 02-Feb-2012 09:02:17

server-name "dhcp.6connect.com";
option option-252 code 252 = text; option option-252 "http://dhcp.6connect.com/wpad.datn";
option domain-name "6connect.com";
authoritative ;#;
option netbios-name-servers 132.220.21.9, 132.220.21.17;
option space LWAPP ; option LWAPP.controller code 241 = string;
use-host-decl-names off;
boot-unknown-clients on;
default-lease-time 28800;
max-lease-time 2678400;
ddns-update-style interim;
deny client-updates;
ddns-hostname = pick-first-value(config-option host-name, binary-to-ascii(16, 16, "", leased-address));
update-static-leases off;
option option-128 code 128 = text;
```

## Importing Your Data

## Step 1 - Normalize your Data

Prior to importing your data, there is a key step of Data Normalization to ensure that information is accurate. If you need assistance with parsing your data prior to importing, 6connect can help with our Data Analyst service. Email us at [support@6connect.com](mailto:support@6connect.com) for more information. You can also use off the shelf tools like Microsoft Excel, MySQL, or [Google Refine](#) if you intend to take on the task of data cleanup in house.



### Data Encoding Format

To ensure correct importing of any special characters, make sure to use UTF-8 encoding for your CSV file!

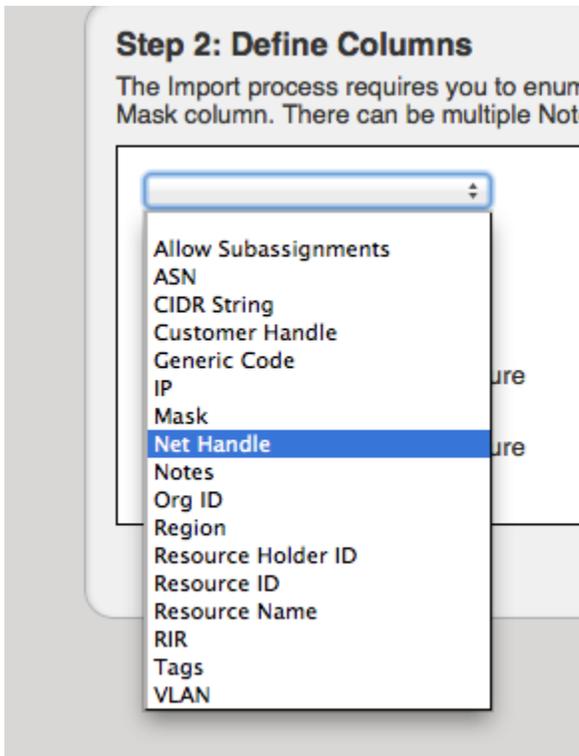
## Step 2 - Prep your Data

You can download [Data Import](#) templates from the [Dashboard Tab](#) or [Data Import Tab](#). We recommend that you open the CSV import templates and get familiar with the data fields that you can import into the platform.

[For Company information](#) you can import relevant data including mailing/billing address information as well as ARIN specific SWIP fields, and specific DNS servers.

[For Contact information](#) you can import contact records assigned to a given **Company**. We support typical fields for this data including Name, multiple email fields, phone numbers as well as Timezone and Role (Roles can be customized from the [IPAM Admin Tab](#)).

[For IPv4 Block information](#) you can import the following fields:



## Step 3 - Import your Data

Get to the [Data Import Tab](#) from the [Admin button](#) to import your data. For larger data import runs, feel free to [contact 6connect](#) at any time for assistance at [support@6connect.com](mailto:support@6connect.com).

### Site Import

[Upload/Import from CSV](#)

[Import from Sales Force](#)

### DNS Import

[BIND Zone Upload/Import](#)

### IP Import

[Upload/Import from CSV](#)

[Import from RIR](#)

### List Management

[List Management](#)

[Import Templates](#)

## BETA FEATURE - Salesforce integration

### Salesforce Import Parameters

Salesforce API Username:

Salesforce Password:

Salesforce Security Token:

Salesforce Custom Field Name:

Account table custom field name matching unique record ID. Example: CUSTID\_\_c

[Start Import](#)

For Salesforce integration, we have also provided a Beta feature for testing. This Import feature allows you to import Account data from Salesforce by matching to your relevant unique identifier field name.

## Import Aggregate Blocks

### Import Aggregates

The Welcome tab allows you to lookup and import your aggregate blocks.

#### Step 1 - Lookup from Source IP

We automatically lookup your ARIN or RIPE information based on the IP address you are connected to:

view
6connect Service Provider Edition - Copyright © 2012 v3.3.x - (6connect)
SUPPORT ADMIN LOGOUT

Dashboard
Customer
DNS
IPAM
Peering
Objects
Log
Reporting

Unsure what to do? Type 'help'

### Welcome to 6connect's Network Automation Platform!

I believe your organization name is: **Comcast Business Communications, LLC**

I believe your ARIN ORGID is: **CBCI**

[Confirm, this is my organization](#)

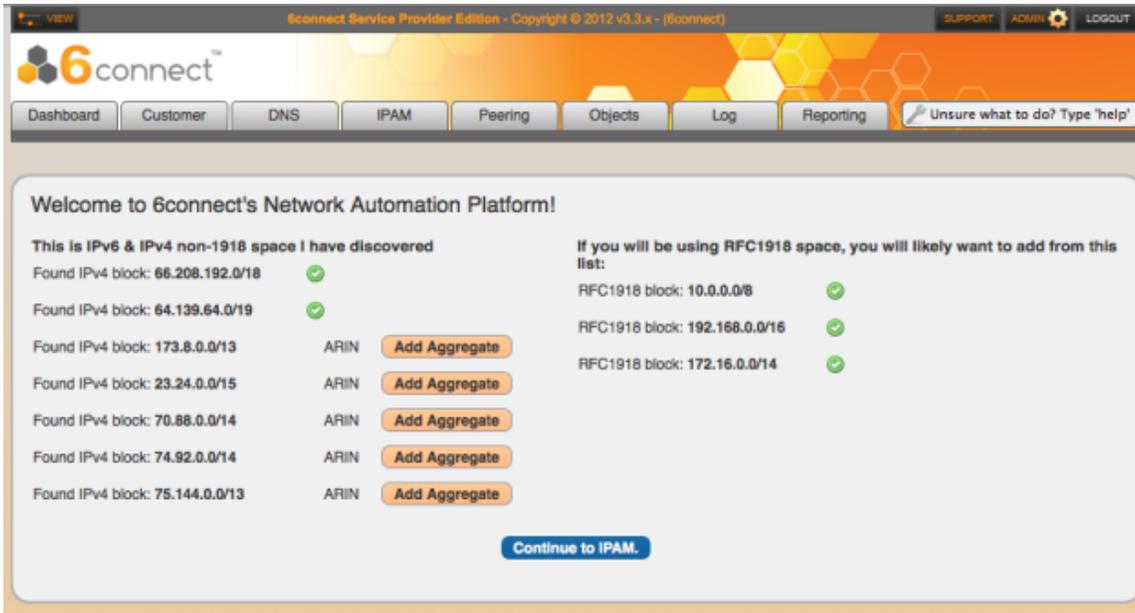
If this is incorrect, please enter an IP address from your network and I will attempt to figure out your aggregates:

[Inquire again](#)

If you have another source IP that you would like to use for the lookup function, you can edit the IP and click on the "[Inquire Again](#)" button. If the organization name and ORGID are correct, then click on the "[Confirm](#)" button to go to the next screen.

## Step 2 - Import your aggregate blocks

Once we have identified the blocks assigned to your company, you can import the aggregates by pressing the "Add Aggregate" buttons. This page allows you to add both 1918 aggregates as well as public IP space from ARIN and RIPE.



## Step 3 - Customizing the Tool

With your aggregates added, you are now ready to customize the tool and import your data! Go to the [Customizing](#) section for details.

## Import DNS Zones

When it comes to importing your DNS zones, the simplest way is using the BIND zone import function built into ProVision. Below, you can also download "sample" files if you wish for examples.

### Preparing your DNS Zones for Import

If your zone data is currently in BIND format - this is very straightforward.

There are three components to the upload process:

#### 1) *The named.conf configuration file tied to the zones you are uploading (required)*

This tells the importer the Zone Name and where the zone file is written. It could be as simple as a multi-line file:

#### Simple DNS Config File

```
zone "my-zone.com" { type master; file "my-zone.com.zone"; };
zone "my-other-zone.com" { type master; file "my-other-zone.com.zone"; };
zone "my-third-zone.com" { type master; file "my-third-zone.com.zone"; };
```

or could be more complex like this file structure directory:

#### Complex DNS Config File

```
zone "my-zone.com" { type master; file "/usr/local/zones/my-zone.com.zone"; };
zone "my-other-zone.com" { type master; file
"/usr/local/zones/more/my-other-zone.com.zone"; };
zone "my-third-zone.com" { type master; file "/usr/local/zones/more/even
more/my-third-zone.com.zone"; };
```

For a sample Simple Config [conf.conf](#)

## 2) A ZIP or TAR file of the DNS zones themselves (required)

This is as it sounds - a file archive where we can find the zones and it should match the configuration file uploaded in Step 1.



### Zone Order

These zone files can be in any order, or in sub-directories, so long as the configuration file (Step 1) correctly points to them

For a sample ZIP [zones.zip](#)

## 3) Match CSV for assigning DNS Zones to Resources (optional)

This file allows the administrator to "assign" zone files to a given Resource. If you have Imported a group of Resources, they have Resource IDs associated with them. You can then import DNS zones and assign them to those Resource IDs. When complete, you will be able to pull up the Resource Record and see the DNS Zones associated to that Resource ID.

### Sample CSV File

```
my-zone.com,test-01,fun stuff
my-otherzone.com,test-02,great stuff
even-reverse-zones.arpa,test-03,amazing stuff
```

Note the columns are the "Zone Name", the "Resource ID", and "Notes".

For a sample CSV [config.csv](#)

## Feedback and Feature Requests

For information on future releases, click on the "Coming Soon" link on the Dashboard.

Status	
Backup	
User Accounts	25
Version	3.9.x
Coming Soon	

You can also submit product feedback and feature requests to [support@6connect.com](mailto:support@6connect.com)

## Tutorials

### Tutorials



Here we have grouped together video tutorials for various tasks and UI components. We link to these in the Getting Started area in the documentation, but you can also browse them individually depending on your needs. If you have suggestions for content - please send them to [support@6connect.com](mailto:support@6connect.com).

## Common Tasks

## UI Tours

- [Common Tasks](#)
- [UI Tours](#)

## Common Tasks

[Adding/Editing blocks](#)

[Aggregating/Splitting blocks](#)

[SWIP configuration and use](#)

[RPSL configuration and use](#)

## UI Tours

### Administration

#### Managing Group and User Permissions



### DNS

#### PowerDNS w/ MySQL Support



## FAQ



▼ [How can I manage overlapping/duplicate IP blocks?](#)

When breaking apart blocks - use the LIR functions (Admin->IPAM Admin->RIR/LIR Manager) to differentiate blocks for 1918 space.

▼ [On the dashboard, I see "n+1" users - why?](#)

The users list includes a "system user" that is only used by ProVision internally in the application.

▼ [I have already SWIPed subnets to ARIN. What happens if I try to SWIP from ProVision, but the block is already SWIPed?](#)

In the case when a user already has SWIPped blocks to ARIN, 6connect checks prior to actually performing a SWIP. In the process, if the IP block is already SWIPped, it will check for existing ARIN customer data and update the 6connect data to reflect what ARIN has on file. Once that is complete, the user can then perform a de-SWIP function using ProVision.

▼ [How does 6connect avoid duplicate assignments or resolve conflicts?](#)

When you make an API request to assign a block, if the block is already assigned to another resource, you will receive an error. If your process is to search for and then assign blocks, the Smart Assign API call may be very helpful. That call combines the search and assignment into one action.

✓ [My VM works, but I am getting a "URL Not Found" error when using ProVision](#)

Please make sure that URL rewriting is enabled in your instance (apache mod\_rewrite)

✓ [My DNS zone views aren't working as they should!](#)

In some legacy instances we have seen zone record-view linkages come out of alignment and result in unexpected behavior.



**BACKUP YOUR DATABASE**

Please note that the following mysql commands modify your database! Please take a backup copy of your database before performance any database modifications.

First, verify the error with the following mysql commands:

```
SELECT count(*) FROM `zone_server_linkage` as t1
INNER JOIN `records` as t2 ON t1.`zoneid` = t2.`zone_id`
INNER JOIN `dns_views` as t3 ON t1.`serverid` = t3.`server_id` AND
`name` = '_6connectDefault'
LEFT JOIN `dns_view_record_linkage` as t4 ON t2.`id` = t4.`record_id`
AND t3.`id` = t4.`view_id`
WHERE t4.`id` IS NULL;
```

If the reply comes back non-zero, then your database is most likely exhibiting unexpected behavior.

The following mysql commands will re-align all the record-view linkages:

```
INSERT INTO `dns_view_record_linkage` SELECT '', t2.`id` as `record_id`,
t3.`id` as `view_id` FROM `zone_server_linkage` as t1
INNER JOIN `records` as t2 ON t1.`zoneid` = t2.`zone_id`
INNER JOIN `dns_views` as t3 ON t1.`serverid` = t3.`server_id` AND
`name` = '_6connectDefault'
LEFT JOIN `dns_view_record_linkage` as t4 ON t2.`id` = t4.`record_id`
AND t3.`id` = t4.`view_id`
WHERE t4.`id` IS NULL;
```

Contact support([support@6connect.com](mailto:support@6connect.com)) if you have any additional questions or this does not resolve the issue.

## Previous Versions

Documentation for Previous Versions of 6connect software:

[6connect-Service\\_Provider\\_Edition\\_3.9.pdf](#)

[6C-v3.0-Manual.pdf](#)

[6C-v2.5.13-Manual.pdf](#)

## CLI (Alpha)

Command Line Interface - ALPHA

- [Overview](#)
- [CLI Commands \(ALPHA\)](#)

## Overview

The command line interface for ProVision is a beta feature that has been release for feedback.



### How to Access the CLI from your browser

When logged into ProVision via a web browser, use the key combination "**Control+Shift+S**" or "**Control+Shift+~**" to access/close the CLI

## CLI Commands (ALPHA)



### CLI Help

When in the CLI, type:

```
ipam man
```

for sample commands and syntax

Currently, the CLI supports the following commands:

```
ipam <command> [-t] [<cidr>] [<resource name>] [<args>]

    show:      show details for a block. Examples:

                - "ipam show 10.0.0.0/8" will show details for the block 10.0.0.0/8

                - "ipam show holding" will show details for all blocks in the
Holding Tank

                - "ipam show "<resource name>" will show details for all blocks
assigned to <resource name>

    add:      add a block. ex: "ipam add 192.168.0.0/24"

    update:   update attributes for a block. ex: ipam update 192.168.0.0/24 --vlan=100
tags=VM,Dev

    assign:   assign a block to a resource. ex: ipam assign 192.168.0.0/24 "<resource
name>"

    assign:   smart assign a block to a resource. ex: ipam assign --mask=24 --rir=ARIN
--type=ipv4 "<resource name>"

    unassign: reclaims a block from a resource and places it in the Holding Tank. If
the block is already in the holding tank, reclaims it and makes it available.
```

## API v1

- [1 - Overview](#)
- [2 - Making API Requests](#)
- [3 - SDK - PHP](#)
- [API Module - Admin and Audit](#)
- [API Module - DHCP](#)
- [API Module - DNS](#)
- [API Module - IPAM](#)
- [API Module - LIR](#)
- [API Module - Resource](#)

### 1 - Overview

## 6connect API - Overview

The 6Connect API is a RESTful API to access your data in the 6Connect tools. ReST relies on stateless, client-server communication, and is usually always implemented using the HTTP protocol (the 6Connect API uses HTTPS). It is a simple and lightweight alternative to Web Services and can be implemented in nearly any language. The 6Connect API operates similarly to other popular ReST APIs you may have worked with, such as Facebook or Twitter. You simply create an HTTP GET or POST request according to our standard, send it to the server, and receive data back.

To learn more about request formatting, making requests, and the tools available, visit [Making API Requests](#).

You can also get the [PHP SDK](#) for PHP libraries and sample code.

Here are some important details about our ReST implementation:

- The API only comes with the full 6Connect IPAM product. If you would like to upgrade to the full version, contact [sales@6Connect.com](mailto:sales@6Connect.com).
- All transactions are over HTTPS (SSL - port 443) only. Any transaction not using SSL will be rejected, and you will have potentially exposed sensitive data.
- All API results are formatted in JSON. XML support is coming soon.
- All requests are either HTTP GET or POST requests. We suggest using POST if the length of data in the request is over 8KB.
- You can use any language you would like to query the API. We currently have an [SDK for PHP](#). Looking at the sample code would probably help you implement it in any language though.

## 2 - Making API Requests

### 6connect API - Making API Requests

API requests can be generated within the web UI by the API Request Generator, or generated programmatically in any language.

**API request looks like this:**

```
https://cloud.6connect.com/ex/api/v1/api.php?target=ipam&action=get&apiKey=38-3NMTNS71JTFX&type=IP&mask=24&time=1335316083&has  
h=VR/owFQCik9HkbsM/oBcfNiLzHSrw=
```

**An API response looks like this:**

```
{"success":1, "message":"1 blocks found",  
"data":[{"id":"7539", "oct1":"1", "oct2":"2", "oct3":"3", "oct4":"0", "mask":"24", "child1":null, "child2":null, "is_assigned":"0", "is_swipped":"0",  
"is_aggregate":"1", "custid":"holding", "last_updated_time":"2012-03-20 09:49:00", "description":null, "parent":null, "rir":"ARIN", "notes":"2012-03-20  
09:49:00", "generic_code":null, "region":null, "vlan":null, "arin_net_id":null, "arin_cust_id":null, "arin_swip_time":"0000-00-00  
00:00:00", "assigned_time":"2012-03-20 09:45:12"}]}
```

#### Using API keys

When using the API without an established authentication to ProVision, you must include both the 'apiKey' and 'hash' parameters.

**apiKey:** Found at AdminAPI

**hash:** The hash is generated from your secret key and the query string. The exact algorithm is `base64_encode(hash_hmac('sha256', preg_replace('/&hash=.*$/ ', $_SERVER['QUERY_STRING']), $secretKey, TRUE))`; but the [PHP-SDK](#) is the supported method of accessing the API.

An easy way to test the API and ensure that keys for a user are setup correctly, is to use the API Request Generator, located in the web interface at AdminAPI. Fill out the fields specifying the type of request, and any additional data. Click "Generate Request URL", and copy and paste the resulting URL into a web browser. The request URL and response will look like those listed above.

The 6Connect API can be used in just about any scripting or programming language. We have a [PHP SDK](#) that provides example code, and several useful functions for interacting with the API. Even if you don't want to use PHP, the samples will help you create code in other languages.

## 3 - SDK - PHP

### 6connect API - Getting Started with the SDK for PHP

The 6Connect API allows you to access to data and functions of the 6Connect web tools. The SDK for PHP will help you get this setup quickly by outlining the requirements, prerequisites and provide sample code.

#### Prerequisites

The API only comes with the full 6Connect IPAM product. If you would like to upgrade to the full version, please contact [sales@6Connect.com](mailto:sales@6Connect.com).

## Create Your API Credentials

To use the 6Connect SDK for PHP, you will need a 6Connect API Key and Secret Key.

### To create your API Key and Secret Key:

- Log into your 6Connect instance (hosted or local)
- Click on the Admin icon, and go into the Administration section.
- Click on the "Manage Users" tab, and go to the "API Keys" sub menu.
- Select the user from the drop down you want to enable API access for, and click "Generate Keys".
- The API Key and the Secret Key will now appear directly below that.

\*Note that generating a new API will automatically revoke an older API Key.

6Connect recommends that each user accessing the API have their own API key configured. However, you can alternatively setup API users by functionality or roles. While the platform is flexible, you should follow your organizations security policies.

### Important!

Your Secret Key is a secret! Only you and 6Connect should ever know this information. It is important to keep it confidential to protect the privacy of your data. Store it securely and never share this key with other users or place it on other systems. Never include the secret key in requests to 6Connect, support requests to 6Connect, and never e-mail it to anyone. Do not share it outside your organization. No one who legitimately represents 6Connect will ever ask you for your Secret Key.

## Requirements

Aside from following the prerequisites, you will need a basic understanding of object oriented programming in PHP and the right tools installed on your system to use the API.

### Minimum Requirements

- PHP 5.3 or newer.
- PHP JSON and PCRE extensions (XML will be coming soon).
- Curl PHP extension compiled with OpenSSL libraries. [Click here for more information on curl.](#)

If you aren't sure what is running on your system, you can create a php page on your system and call `phpinfo()` and view this page in a browser, or run `php -i` on the command line.

## Install the SDK

Download the file [6connectAPI-PHP-SDK.tgz](#).

### Configure the SDK Security Credentials

- Extract the zipped tar file to a directory.
- Open the `api-config.php` located in the downloaded SDK files.
- Read through the file and place in your instance name (or path for local installs), API Key and Secret Key information as specified.
- Make sure all files are in the same directory (the core class looks for a config file in the same directory by default).
- Run the sample code `api-example.php`!

### Important!

You must setup user API access before running the sample. See the previous section "Create Your API Credentials" for more information.

## Need More Information?

If you need more general information on the API, try the [API Overview](#).

If you need information on methods available via the API, look at the [API Reference](#).

The SDK also contains a README file with other useful information particular to php.

## API Module - Admin and Audit

This section covers the functions found under the Admin section of ProVision.

# Authentication Testing

## Authentication Testing

testSSH																					
URL	/api/v1/api.php?target=auth&action=testSSH																				
Description	Returns success or failure of a connection to an external server via SSH.																				
Returns	<p><b>Examples:</b></p> <table border="1"> <tbody> <tr> <td>SUCCESSFUL</td> <td>{"success":1,"message":"Success!"}</td> </tr> <tr> <td>ERROR</td> <td>{'success':0, 'message':'error message'}</td> </tr> </tbody> </table>	SUCCESSFUL	{"success":1,"message":"Success!"}	ERROR	{'success':0, 'message':'error message'}																
SUCCESSFUL	{"success":1,"message":"Success!"}																				
ERROR	{'success':0, 'message':'error message'}																				
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>SSHServer</td> <td>STRING</td> <td>totally.awesome.com</td> <td>IP or FQDN of server.</td> </tr> <tr> <td>SSHPort</td> <td>NUMBER</td> <td>22</td> <td>Port ssh is running on.</td> </tr> </tbody> </table>	Name	Type	Example	Description	SSHServer	STRING	totally.awesome.com	IP or FQDN of server.	SSHPort	NUMBER	22	Port ssh is running on.								
Name	Type	Example	Description																		
SSHServer	STRING	totally.awesome.com	IP or FQDN of server.																		
SSHPort	NUMBER	22	Port ssh is running on.																		
Optional Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>username</td> <td>STRING</td> <td>jsmith</td> <td>Username on target server.</td> </tr> <tr> <td>password</td> <td>STRING</td> <td>password123</td> <td>Password for user.</td> </tr> <tr> <td>directory</td> <td>STRING</td> <td>/tmp</td> <td>Directory to attempt to access after successful login.</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Example	Description	username	STRING	jsmith	Username on target server.	password	STRING	password123	Password for user.	directory	STRING	/tmp	Directory to attempt to access after successful login.				
Name	Type	Example	Description																		
username	STRING	jsmith	Username on target server.																		
password	STRING	password123	Password for user.																		
directory	STRING	/tmp	Directory to attempt to access after successful login.																		
Example URL	/api/v1/api.php?target=auth&action=testSSH&username=jsmith&password=password123&directory=%2Fvar%2Fnamed%2F6connect%2Fq4&SSHPort=22																				

testLDAP	
URL	/api/v1/api.php?target=auth&action=testSSH
Description	Test basic connectivity to an LDAP server. Does not test actual authentication against server.

Returns	<b>Examples:</b> SUCCESSFUL: <code>{'success':1, 'id':'12345'}</code> ERROR: <code>{'success':0, 'message':'unable to add block'}</code> >																
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>ldapServer</td> <td>STRING</td> <td>ldap.awesome.com</td> <td>IP or FQDN of the LDAP server.</td> </tr> <tr> <td>ldapPort</td> <td>NUMBER</td> <td>389</td> <td>User-defined block code as defined in Admin-IPAM settings: Generic Code Per Block Name</td> </tr> <tr> <td>ldapMode</td> <td>STRING</td> <td>SSL</td> <td>Options are: SSL, TLS, or None.</td> </tr> </tbody> </table>	Name	Type	Example	Description	ldapServer	STRING	ldap.awesome.com	IP or FQDN of the LDAP server.	ldapPort	NUMBER	389	User-defined block code as defined in Admin-IPAM settings: Generic Code Per Block Name	ldapMode	STRING	SSL	Options are: SSL, TLS, or None.
Name	Type	Example	Description														
ldapServer	STRING	ldap.awesome.com	IP or FQDN of the LDAP server.														
ldapPort	NUMBER	389	User-defined block code as defined in Admin-IPAM settings: Generic Code Per Block Name														
ldapMode	STRING	SSL	Options are: SSL, TLS, or None.														
Optional Parameters	None																
Example URL	<code>/api/v1/api.php?target=auth&amp;action=testLDAP&amp;ldapPort=389&amp;ldapServer=ldap.awesome.com&amp;ldapMode=None</code>																

<b>testSecure64</b>													
URL	<code>/api/v1/api.php?target=auth&amp;action=testSecure64</code>												
Description	Returns success or failure of a connection to an Secure64 DNS appliance.												
Returns	<b>Examples:</b> <table border="1"> <tbody> <tr> <td>SUCCESSFUL</td> <td><code>{"success":1,"message":"Success!"}</code></td> </tr> <tr> <td>ERROR</td> <td><code>{'success':0, 'message':'error message'}</code></td> </tr> </tbody> </table>	SUCCESSFUL	<code>{"success":1,"message":"Success!"}</code>	ERROR	<code>{'success':0, 'message':'error message'}</code>								
SUCCESSFUL	<code>{"success":1,"message":"Success!"}</code>												
ERROR	<code>{'success':0, 'message':'error message'}</code>												
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>SSHServer</td> <td>STRING</td> <td>totally.awesome.com</td> <td>IP or FQDN of server.</td> </tr> <tr> <td>SSHPort</td> <td>NUMBER</td> <td>22</td> <td>Port ssh is running on.</td> </tr> </tbody> </table>	Name	Type	Example	Description	SSHServer	STRING	totally.awesome.com	IP or FQDN of server.	SSHPort	NUMBER	22	Port ssh is running on.
Name	Type	Example	Description										
SSHServer	STRING	totally.awesome.com	IP or FQDN of server.										
SSHPort	NUMBER	22	Port ssh is running on.										

Optional Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>username</td> <td>STRING</td> <td>jsmith</td> <td>Username on target server.</td> </tr> <tr> <td>password</td> <td>STRING</td> <td>password123</td> <td>Password for user.</td> </tr> <tr> <td>directory</td> <td>STRING</td> <td>/tmp</td> <td>Directory to attempt to access after successful login.</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Example	Description	username	STRING	jsmith	Username on target server.	password	STRING	password123	Password for user.	directory	STRING	/tmp	Directory to attempt to access after successful login.				
	Name	Type	Example	Description																	
	username	STRING	jsmith	Username on target server.																	
	password	STRING	password123	Password for user.																	
	directory	STRING	/tmp	Directory to attempt to access after successful login.																	
Example URL	/api/v1/api.php?target=auth&action=testSecure64&username=jsmith&password=password123&directory=%2Fvar%2Fnamed%2F6connect%2Fqa4&SSHPort=22																				

## Log Management

### Log Management



Get					
URL	/api/v1/api.php?target=log&action=get				
Description	Returns a list of log entries. Use optional parameters to filter the list.				
Returns	<p><b>Examples:</b></p> <table border="1"> <tbody> <tr> <td>SUCCESSFUL</td> <td><pre>{ "success": 1, "message": "Search Successful.", "data": { "logId": "31568", "time": "2012-05-07 17:44:43", "logLevel": "INFO", "userId": "39", "userName": "anna@6connect.com", "logCategory": "User", "message": "Anna Claiborne (anna@6connect.com) logged in via local authentication", "ip": "107.111.0.228" } }</pre></td> </tr> <tr> <td>ERROR</td> <td><pre>{ 'success': 0, 'message': 'error message' }</pre></td> </tr> </tbody> </table> <p><b>Data Detail</b></p>	SUCCESSFUL	<pre>{ "success": 1, "message": "Search Successful.", "data": { "logId": "31568", "time": "2012-05-07 17:44:43", "logLevel": "INFO", "userId": "39", "userName": "anna@6connect.com", "logCategory": "User", "message": "Anna Claiborne (anna@6connect.com) logged in via local authentication", "ip": "107.111.0.228" } }</pre>	ERROR	<pre>{ 'success': 0, 'message': 'error message' }</pre>
SUCCESSFUL	<pre>{ "success": 1, "message": "Search Successful.", "data": { "logId": "31568", "time": "2012-05-07 17:44:43", "logLevel": "INFO", "userId": "39", "userName": "anna@6connect.com", "logCategory": "User", "message": "Anna Claiborne (anna@6connect.com) logged in via local authentication", "ip": "107.111.0.228" } }</pre>				
ERROR	<pre>{ 'success': 0, 'message': 'error message' }</pre>				

Name	Type	Example	Description
logId	INTEGER	24	Unique log entry id.
time	DATETIME	2012-05-07 22:10:07	Date and time year to second.
logLevel	STRING	NOTICE	Standard syslog log levels in verbose format (EMERG, ALERT, CRIT, ERR, WARNING, NOTICE, INFO, DEBUG).
userId	Integer	11	The unique user id associated with the log entry.
userName	STRING	anna@6connect.com	The unique user name associated with the log entry.
logCategory	STRING	IPAM	The 6connect category for the log entry (User, IPAM, Resource Holder, DNS, Peering, Assistant, NTP, Reporting).
message	STRING	Created new children from 1.0.0.0/24	The detailed log message.
ip	STRING	107.111.0.228	The remote IP address of the user who took the action being logged.

Required Parameters

None

Optional Parameters

Name	Type	Example	Description
likeFlag	BOOL	1	When 1, string searches are done via LIKE with wildcards at both ends. When 0, strict comparison is used.
logId	INTEGER	24	Unique log entry id.
timeStart	DATETIME	2012-05-07 [21:00:00]	Retrieve logs starting at this Date and optional time year to second.
timeEnd	DATETIME	2012-05-07 [22:00:00]	Retrieve logs ending at this Date and optional time year to second.
limit	INTEGER	100	Total log entries to retrieve. Default limit is 1000 records.
offset	INTEGER	50	Offset from 0 to retrieve log entries
userName	STRING	<a href="mailto:anna@6connect.com">anna@6connect.com</a>	The unique user name associated with the log entry.
logCategory	STRING	IPAM	The 6connect category for the log entry (User, IPAM, Resource Holder, DNS, Peering, Assistant, NTP, Reporting).

	logLevel	STRING	NOTICE	Standard syslog log levels in verbose format (EMERG, ALERT, CRIT, ERR, WARNING, NOTICE, INFO, DEBUG).
	ip	STRING	1.2.3.4	The remote IP address of the user whose action was logged
	block	STRING	1.2.3.4/8	Used to return any actions performed on the specified block.
Example URL	/api/v1/api.php?target=log&action=get			

## Zone Templates

### Zone Templates



<b>Get</b>	
URL	/api/v1/api.php?target=zoneTemplate&action=get
Description	Returns success or failure of a connection to an external server via SSH.

Returns	<p><b>Examples:</b></p> <table border="1"> <tr> <td>SUCCESSFUL</td> <td> <pre>{   "success":1,   "message":"Found 1 records for template \\Awesome Template\\.",   "data":{     "templateId":"1011",     "name":"Awesome Template",     "created":"2013-07-31 14:01:24",     "modified":"2013-07-31 14:01:24",     "userId":"112",     "soa":null,     "refresh":"14400",     "retry":"3600",     "expire":"604800",     "minimum":null,     "ttl":null,     "userName":"joe@smith.com",     "records":[{"templateRecordId":"4", "templateId":"1011", "host":"www", "type":"A", "ttl":"3600", "value":"1.2.3.4", "ordering":"0"}]} }</pre> </td> </tr> <tr> <td>ERROR</td> <td> <pre>{'success':0, 'message':'error message'}</pre> </td> </tr> </table>	SUCCESSFUL	<pre>{   "success":1,   "message":"Found 1 records for template \\Awesome Template\\.",   "data":{     "templateId":"1011",     "name":"Awesome Template",     "created":"2013-07-31 14:01:24",     "modified":"2013-07-31 14:01:24",     "userId":"112",     "soa":null,     "refresh":"14400",     "retry":"3600",     "expire":"604800",     "minimum":null,     "ttl":null,     "userName":"joe@smith.com",     "records":[{"templateRecordId":"4", "templateId":"1011", "host":"www", "type":"A", "ttl":"3600", "value":"1.2.3.4", "ordering":"0"}]} }</pre>	ERROR	<pre>{'success':0, 'message':'error message'}</pre>								
SUCCESSFUL	<pre>{   "success":1,   "message":"Found 1 records for template \\Awesome Template\\.",   "data":{     "templateId":"1011",     "name":"Awesome Template",     "created":"2013-07-31 14:01:24",     "modified":"2013-07-31 14:01:24",     "userId":"112",     "soa":null,     "refresh":"14400",     "retry":"3600",     "expire":"604800",     "minimum":null,     "ttl":null,     "userName":"joe@smith.com",     "records":[{"templateRecordId":"4", "templateId":"1011", "host":"www", "type":"A", "ttl":"3600", "value":"1.2.3.4", "ordering":"0"}]} }</pre>												
ERROR	<pre>{'success':0, 'message':'error message'}</pre>												
Required Parameters	None												
Optional Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>templateId</td> <td>NUMBER</td> <td>3</td> <td>ID of the specific template to get.</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Example	Description	templateId	NUMBER	3	ID of the specific template to get.				
Name	Type	Example	Description										
templateId	NUMBER	3	ID of the specific template to get.										
Example URL	/api/v1/api.php?target=zoneTemplate&action=get												

Update	
URL	/api/v1/api.php?target=zoneTemplate&action=update
Description	Create a new template or update an existing template.
Returns	<p><b>Examples:</b></p> <p>SUCCESSFUL: <pre>{   "success":1,   "message":"Template updated",   "data":{     "templateId":"1011",     "name":"Awesome Template",     "created":"2013-08-05 23:15:52",     "modified":"2013-08-05 23:15:52",     "userId":"112",     "soa":"ns1.test.net hostmaster.ns1.test.net",     "refresh":"14400",     "retry":"3600",     "expire":"604800",     "minimum":null,     "ttl":false,     "userName":"anna@6connect.com",     "records":"1"} }</pre></p> <p>ERROR: <pre>{'success':0, 'message':'Error updating template: error details'}&gt;</pre></p>

Required Parameters

Name	Type	Example	Description
name	STRING	Test Template	The name of the template to be created or updated.

Optional Parameters

Name	Type	Example	Description
soa	STRING	ns1.test.net hostmaster.ns 1.test.net	A valid SOA for the template in for format
ttl	INTEGER	86400	The TTL for the zone template, which is the default expiration time for all records without their own TTL.
refresh	INTEGER	14400	The time period for slaves to refresh the zone.
retry	INTEGER	3600	Time that a slave should retry refreshing the zone in case of incident.
expire	INTEGER	604800	Time for a slave to expire a zone.
mininum	INTEGER	3600	The maximum caching time in the event of failed lookups.

count_records	INTEGER	5	Number of host records submitted with the update. All the following parameters names should be followed with their position in the count. In this example, the first record would have all the parameters for the first record followed by _1, the second record _2, and so on. This will be the order all records in the template follow.
host_1	STRING		The DNS record value.
t1_1	INTEGER	3600	TTL of the specific host record.
type_1	STRING	A	A valid DNS record type.
value_1	IP	1.2.3.4	A valid IPv4 or IPv6 address.

Example URL

api/v1/api.php?target=zoneTemplate&action=update&templateId=1011&count\_records=1&name=Awesome+Template&soa=ns1.test.net+hostmaster.ns1.test.net  
&refresh=14400&retry=3600&expire=604800&minimum=3600&value\_0=undefined+undefined&host\_1=www&t1\_1=3600&type\_1=A&value\_1=1.2.3.4

#### Delete

URL

/api/v1/api.php?target=zoneTemplate&action=delete

Description

Deletes a DNS template.

Returns	<p><b>Examples:</b></p> <table border="1"> <tr> <td>SUCCESSFUL</td> <td><code>{"success":1,"message":"Template \"Test Template\" delete."}</code></td> </tr> <tr> <td>ERROR</td> <td><code>{"success":0,"message":"No template found for templateId \"1005\"."}</code></td> </tr> </table>	SUCCESSFUL	<code>{"success":1,"message":"Template \"Test Template\" delete."}</code>	ERROR	<code>{"success":0,"message":"No template found for templateId \"1005\"."}</code>				
SUCCESSFUL	<code>{"success":1,"message":"Template \"Test Template\" delete."}</code>								
ERROR	<code>{"success":0,"message":"No template found for templateId \"1005\"."}</code>								
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>templateId</td> <td>INTEGER</td> <td>3</td> <td>ID of the template to delete.</td> </tr> </tbody> </table>	Name	Type	Example	Description	templateId	INTEGER	3	ID of the template to delete.
Name	Type	Example	Description						
templateId	INTEGER	3	ID of the template to delete.						
Optional Parameters	None.								
Example URL	<code>/api/v1/api.php?target=zoneTemplate&amp;action=delete&amp;templateId=1005</code>								

## API Module - DHCP

- [DHCP Server Control](#)
- [DHCP Entry Control](#)

### DHCP Server Control

<b>get</b>	
URL	<code>/api/v1/api.php?target=DHCPServer&amp;action=get</code>
Description	Accepts search criteria to retrieve a list of all matching DHCP Servers.
Returns	<p><b>Examples:</b></p>

SUCCESSFUL:	<pre>{"success":1,"message":"Search Successful.", "data":{"DHCPId": "1", "DHCPServer": "trace.foo.com", "DHCPPort": "22", "DHCPUsername": "benner", "DHCPPassword": "h}k*c))jwqhg*d", "DHCPType": "ISC", "DHCPConfigPath": "VusrVlocalVdhcpVetcVdhcpd.conf", "DHCPServerStop": "sudo kill -9 `cat VvarVrunVdhcpd.pid`", "DHCPServerStart": "sudo VusrVlocalVdhcpVsbinVdhcpd -p 75", "DHCPDefaultLease": null, "DHCPMaxLease": null, "DHCPAuthoritative": "1", "DHCPLogFacility": "local7", "DHCPDomainName": null, "DHCPNameServers": null, "DHCPUseText": "0", "DHCPConfigText": null}}}</pre>
ERROR:	<pre>{"success":0, "message":"error message"}</pre>

**Data Detail:**

Name	Type	Description
DHCPId	INT	The ID of the DHCP Server entry.
DHCPServer	STRING	The address of the DHCP Server
DHCPPort	INT	The port the DHCP Server can be reached on.
DHCPUsername	STRING	The user name required to access the DHCP Server
DHCPPassword	STRING	The password required to access the DHCP Server
DHCPType	STRING	The type of DHCP Server
DHCPConfigPath	STRING	Path to DHCP Configuration file
DHCPServerStop	STRING	Command to stop the DHCP Server
DHCPServerStart	STRING	Command to start the DHCP Server
DHCPDefaultLease	STRING	Default lease time for this server
DHCPMaxLease	STRING	Maximum lease time for this server
DHCPAuthoritative	BOOL	Whether or not this server is authoritative
DHCPLogFacility	STRING	Logging facility for this server
DHCPDomainName	STRING	Domain names servers used by this server
DHCPNameServers	STRING	Name servers used by this server
DHCPUseText	BOOL	Whether or not to use the entry builder or a config text file
DHCPConfigText	STRING	The text of the config text file

Optional Parameters

Name	Type	Example	Description
likeFlag	BOOL	1	When 1, string searches are done via LIKE with wildcards at both ends. When 0, strict comparison is used.
generalFlag	BOOL	1	When 1, searches over the provided parameters using OR. If 0 or omitted, uses AND.

Name	Type	Example	Description
DHCPId	INT	123	The DHCP Server ID to search for.
DHCPServer	STRING	IP/domain	The Server Name to search for.
DHCPPort	STRING	43	The Port to search for.
DHCPUsername	STRING	kjennings	The Username to search for.
DHCPType	STRING	MSDHCP	The DHCP Server Type to search for.
DHCPConfig Path	STRING	/where/is/it/	The Config Path to search for.
DHCPServer Stop	STRING	/path/to/server/stop	Search by server stop command.
DHCPServer Start	STRING	/path/to/server/start	Search by server start command.
DHCPDefault Lease	INT	64000	Search by default lease.
DHCPMaxLease	INT	128000	Search by max lease.
DHCPAuthoritative	BOOL	1	Search by whether the server is authoritative.
DHCPLogFacility	STRING	local7	Search by logging facility.
DHCPDomain Name	STRING	domain.name.server	Search by domain name servers.
DHCPNameServers	STRING	ns.domain.com	Search by name servers.
DHCPUseText	BOOL	1	Search by using text configs or not.
DHCPConfig Text	STRING	Text File	Search by text file contents.

**update**

URL `/api/v1/api.php?target=DHCPServer&action=update`

Description First performs a search based on the submitted DHCP Server criteria, then performs an Update across those entries based on new values.

Returns

**Examples:**

SUCCESSFUL:	<code>{"success":1,"message":"Update Successful."}</code>
ERROR:	<code>{"success":0, "message":"error message"}</code>

Optional Parameters

Name	Type	Example	Description
likeFlag	BOOL	1	When 1, string searches are done via LIKE with wildcards at both ends. When 0, strict comparison is used.

Name	Type	Example	Description
SearchId	INT	123	The DHCP Server ID to search for.
SearchServer	STRING	IP/domain	The Server Name to search for.
SearchPort	STRING	43	The Port to search for.
SearchUsername	STRING	kjennings	The Username to search for.
SearchType	STRING	MSDHCP	The DHCP Server Type to search for.
SearchConfigPath	STRING	/where/is/it/	The Config Path to search for.
SearchServerStop	STRING	/path/to/server/stop	Search by server stop command.
SearchServerStart	STRING	/path/to/server/start	Search by server start command.
SearchDefaultLease	INT	64000	Search by default lease.
SearchMaxLease	INT	128000	Search by max lease.
SearchAuthoritative	BOOL	1	Search by whether the server is authoritative.
SearchLogFacility	STRING	local7	Search by logging facility.
SearchDomainName	STRING	domain.name.server	Search by domain name servers.
SearchNameServers	STRING	ns.domain.com	Search by name servers.
SearchUseText	BOOL	1	Search by using text configs or not.
SearchConfigText	STRING	Text File	Search by text file contents.

Name	Type	Example	Description
UpdateServer	STRING	IP/domain	The new server address.
UpdatePort	STRING	43	The new port.
UpdateUsername	STRING	kjennings	The new username.
UpdatePassword	STRING	*****	The new password.
UpdateType	STRING	ISC	The new server type.
UpdateConfigPath	STRING	/where/is/it/	The new config path.
UpdateServerStop	STRING	/path/to/server/stop	The new server stop command.
UpdateServerStart	STRING	/path/to/server/start	The new server start command.
UpdateDefaultLease	INT	64000	The new default lease.
UpdateMaxLease	INT	128000	The new max lease.
UpdateAuthoritative	BOOL	1	The new Authoritative status.
UpdateLogFacility	STRING	local7	The new logging facility.
UpdateDomainName	STRING	domain.name.server	The new domain name servers.
UpdateNameServers	STRING	ns.domain.com	The new name servers.
UpdateUseText	BOOL	1	The new use text file setting.
UpdateConfigText	STRING	Text File	The new use text file content.

#### add

URL </api/v1/api.php?target=DHCPserver&action=add>

Description Adds a new DHCP Server.

Returns

**Examples:**

SUCCESSFUL:	<code>{"success":1,"message":"Add Successful. ","data":123}</code>
ERROR:	<code>{"success":0, "message":"error message"}</code>

**Data Detail:**

Name	Type	Description
data	INT	The ID of the new DHCP Server.

Required Parameters

Name	Type	Example	Description
DHCPServer	STRING	IP/domain	The new server address.
DHCPPort	STRING	43	The new port.
DHCPUsername	STRING	kjennings	The new username.
DHCPPassword	STRING	*****	The new password.
DHCPType	STRING	ISC	The new server type.

## Optional Parameters

Name	Type	Example	Description
DHCPConfig Path	STRING	/where/is/it/	The new config path.
DHCPServer Stop	STRING	/path/to/server/stop	The new server stop command.
DHCPServer Start	STRING	/path/to/server/start	The new server start command.
DHCPDefault Lease	INT	64000	The new default lease.
DHCPMaxLease	INT	128000	The new max lease.
DHCPAuthoritative	BOOL	1	The new Authoritative status.
DHCPLogFacility	STRING	local7	The new logging facility.
DHCPDomain Name	STRING	domain.name.server	The new domain name servers.
DHCPNameServers	STRING	ns.domain.com	The new name servers.
DHCPUseText	BOOL	1	The new use text file setting.
DHCPConfig Text	STRING	Text File	The new use text file content.

**delete**

URL

/api/v1/api.php?target=DHCPServer&amp;action=delete

Description

Performs a search over the DHCP Servers dataset and deletes all found matches.

Returns

**Examples:**

SUCCESSFUL:

```
{"success":1,"message":"DHCP Server(s) Deleted."}
```

ERROR:

```
{"success":0,"message":"error message"}
```

Optional Parameters

Name	Type	Example	Description
likeFlag	BOOL	1	When 1, string searches are done via LIKE with wildcards at both ends. When 0, strict comparison is used.
generalFlag	BOOL	1	When 1, searches over the provided parameters using OR. If 0 or omitted, uses AND.

Name	Type	Example	Description
DHCPId	INT	123	The DHCP Server ID to search for.
DHCPServer	STRING	IP/domain	The Server Name to search for.
DHCPPort	STRING	43	The Port to search for.
DHCPUsername	STRING	kjennings	The Username to search for.
DHCPType	STRING	MSDHCP	The DHCP Server Type to search for.
DHCPConfig Path	STRING	/where/is/it/	The Config Path to search for.
DHCPServer Stop	STRING	/path/to/server/stop	Search by server stop command.
DHCPServer Start	STRING	/path/to/server/start	Search by server start command.
DHCPDefault Lease	INT	64000	Search by default lease.
DHCPMaxLease	INT	128000	Search by max lease.
DHCPAuthoritative	BOOL	1	Search by whether the server is authoritative.
DHCPLogFacility	STRING	local7	Search by logging facility.
DHCPDomain Name	STRING	domain.name.server	Search by domain name servers.
DHCPNameServers	STRING	ns.domain.com	Search by name servers.
DHCPUseText	BOOL	1	Search by using text configs or not.
DHCPConfig Text	STRING	Text File	Search by text file contents.

<b>testConnection</b>													
URL	/api/v1/api.php?target=DHCPServer&action=testConnection												
Description	Performs a search over the DHCP Servers dataset and tests the login/password combo for each one returned. Returns a status array with three elements: a 1 or 0 for success/failure, the server in question, and the failure/success message.												
Returns	<p><b>Examples:</b></p> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td><pre>{"success":1,"message":"Pushes Attempted.", "data":[[1, "foo.fun.com", "Successfully authenticated on DHCP Server 'foo.fun.com'.",[0,"foo.fun.com", "Could not authenticate on server 'foo.fun.com'. Connection refused."],[0,"28.39.106.129", "Could not connect to server '28.39.106.129'. Connection refused."]]}</pre></td> </tr> <tr> <td>ERROR:</td> <td><pre>{"success":0, "message":"error message"}</pre></td> </tr> </table>	SUCCESSFUL:	<pre>{"success":1,"message":"Pushes Attempted.", "data":[[1, "foo.fun.com", "Successfully authenticated on DHCP Server 'foo.fun.com'.",[0,"foo.fun.com", "Could not authenticate on server 'foo.fun.com'. Connection refused."],[0,"28.39.106.129", "Could not connect to server '28.39.106.129'. Connection refused."]]}</pre>	ERROR:	<pre>{"success":0, "message":"error message"}</pre>								
SUCCESSFUL:	<pre>{"success":1,"message":"Pushes Attempted.", "data":[[1, "foo.fun.com", "Successfully authenticated on DHCP Server 'foo.fun.com'.",[0,"foo.fun.com", "Could not authenticate on server 'foo.fun.com'. Connection refused."],[0,"28.39.106.129", "Could not connect to server '28.39.106.129'. Connection refused."]]}</pre>												
ERROR:	<pre>{"success":0, "message":"error message"}</pre>												
Optional Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>likeFlag</td> <td>BOOL</td> <td>1</td> <td>When 1, string searches are done via LIKE with wildcards at both ends. When 0, strict comparison is used.</td> </tr> <tr> <td>generalFlag</td> <td>BOOL</td> <td>1</td> <td>When 1, searches over the provided parameters using OR. If 0 or omitted, uses AND.</td> </tr> </tbody> </table>	Name	Type	Example	Description	likeFlag	BOOL	1	When 1, string searches are done via LIKE with wildcards at both ends. When 0, strict comparison is used.	generalFlag	BOOL	1	When 1, searches over the provided parameters using OR. If 0 or omitted, uses AND.
Name	Type	Example	Description										
likeFlag	BOOL	1	When 1, string searches are done via LIKE with wildcards at both ends. When 0, strict comparison is used.										
generalFlag	BOOL	1	When 1, searches over the provided parameters using OR. If 0 or omitted, uses AND.										

Name	Type	Example	Description
DHCPId	INT	123	The DHCP Server ID to search for.
DHCPServer	STRING	IP/domain	The Server Name to search for.
DHCPPort	STRING	43	The Port to search for.
DHCPUsername	STRING	kjennings	The Username to search for.
DHCPType	STRING	MSDHCP	The DHCP Server Type to search for.
DHCPConfig Path	STRING	/where/is/it/	The Config Path to search for.
DHCPServer Stop	STRING	/path/to/server/stop	Search by server stop command.
DHCPServer Start	STRING	/path/to/server/start	Search by server start command.
DHCPDefault Lease	INT	64000	Search by default lease.
DHCPMaxLease	INT	128000	Search by max lease.
DHCPAuthoritative	BOOL	1	Search by whether the server is authoritative.
DHCPLogFacility	STRING	local7	Search by logging facility.
DHCPDomain Name	STRING	domain.name.server	Search by domain name servers.
DHCPNameServers	STRING	ns.domain.com	Search by name servers.
DHCPUseText	BOOL	1	Search by using text configs or not.
DHCPConfig Text	STRING	Text File	Search by text file contents.

<b>push</b>													
URL	/api/v1/api.php?target=DHCPServer&action=push												
Description	Performs a search over the DHCP Servers dataset and pushes the current config file before restarting the servers. Returns a status array with three elements: a 1 or 0 for success/failure, the server in question, and the failure/success message. A response code of '2' indicates that the push went smoothly, but the configuration file itself contains errors. In this case the error return will be the actual error output from the DHCP server.												
Returns	<p><b>Examples:</b></p> <table border="1"> <tbody> <tr> <td>SUCCESSFUL:</td> <td><pre>{"success":1,"message":"Pushes Attempted.,"data":[[1,"trace.bind.com","Successfully pushed DHCP Config to server 'trace.bind.com'. Server Restarted."],[0,"trace.bind.com","Could not authenticate on server 'trace.bind.com'. Connection refused."],[0,"208.39.106.169","Could not connect to server '208.39.106.169'. Connection refused."]]}</pre></td> </tr> <tr> <td>ERROR:</td> <td><pre>{"success":0,"message":"error message"}</pre></td> </tr> </tbody> </table>	SUCCESSFUL:	<pre>{"success":1,"message":"Pushes Attempted.,"data":[[1,"trace.bind.com","Successfully pushed DHCP Config to server 'trace.bind.com'. Server Restarted."],[0,"trace.bind.com","Could not authenticate on server 'trace.bind.com'. Connection refused."],[0,"208.39.106.169","Could not connect to server '208.39.106.169'. Connection refused."]]}</pre>	ERROR:	<pre>{"success":0,"message":"error message"}</pre>								
SUCCESSFUL:	<pre>{"success":1,"message":"Pushes Attempted.,"data":[[1,"trace.bind.com","Successfully pushed DHCP Config to server 'trace.bind.com'. Server Restarted."],[0,"trace.bind.com","Could not authenticate on server 'trace.bind.com'. Connection refused."],[0,"208.39.106.169","Could not connect to server '208.39.106.169'. Connection refused."]]}</pre>												
ERROR:	<pre>{"success":0,"message":"error message"}</pre>												
Optional Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>likeFlag</td> <td>BOOL</td> <td>1</td> <td>When 1, string searches are done via LIKE with wildcards at both ends. When 0, strict comparison is used.</td> </tr> <tr> <td>generalFlag</td> <td>BOOL</td> <td>1</td> <td>When 1, searches over the provided parameters using OR. If 0 or omitted, uses AND.</td> </tr> </tbody> </table>	Name	Type	Example	Description	likeFlag	BOOL	1	When 1, string searches are done via LIKE with wildcards at both ends. When 0, strict comparison is used.	generalFlag	BOOL	1	When 1, searches over the provided parameters using OR. If 0 or omitted, uses AND.
Name	Type	Example	Description										
likeFlag	BOOL	1	When 1, string searches are done via LIKE with wildcards at both ends. When 0, strict comparison is used.										
generalFlag	BOOL	1	When 1, searches over the provided parameters using OR. If 0 or omitted, uses AND.										

Name	Type	Example	Description
DHCPId	INT	123	The DHCP Server ID to search for.
DHCPServer	STRING	IP/domain	The Server Name to search for.
DHCPPort	STRING	43	The Port to search for.
DHCPUsername	STRING	kjennings	The Username to search for.
DHCPType	STRING	MSDHCP	The DHCP Server Type to search for.
DHCPConfig Path	STRING	/where/is/it/	The Config Path to search for.
DHCPServer Stop	STRING	/path/to/server/stop	Search by server stop command.
DHCPServer Start	STRING	/path/to/server/start	Search by server start command.
DHCPDefault Lease	INT	64000	Search by default lease.
DHCPMaxLease	INT	128000	Search by max lease.
DHCPAuthoritative	BOOL	1	Search by whether the server is authoritative.
DHCPLogFacility	STRING	local7	Search by logging facility.
DHCPDomain Name	STRING	domain.name.server	Search by domain name servers.
DHCPNameServers	STRING	ns.domain.com	Search by name servers.
DHCPUseText	BOOL	1	Search by using text configs or not.
DHCPConfig Text	STRING	Text File	Search by text file contents.

## DHCP Entry Control

get					
URL	/api/v1/api.php?target=DHCPEntry&action=get				
Description	Accepts search criteria to retrieve a list of all matching DHCP Entries along with their associated Options.				
Returns	<p><b>Examples:</b></p> <table border="1"><tbody><tr><td>SUCCESSFUL:</td><td><pre>{"EntryId": "27", "EntryParent": null, "EntryServerId": "1", "EntryName": "mike", "EntryType": "host", "EntryNetmask": "255.255.255.0", "EntryIPCount": "1", "EntryPercent": "1", "Options": [{"OptionId": "46", "OptionSubnetId": "27", "OptionKey": "hardware ethernet", "OptionValue": "11:23:45:67:89:ab"}, {"OptionId": "47", "OptionSubnetId": "27", "OptionKey": "fixed-address", "OptionValue": "10.20.30.158"}]}</pre></td></tr><tr><td>ERROR:</td><td><pre>{"success": 0, "message": "error message"}</pre></td></tr></tbody></table> <p><b>Data Detail:</b></p>	SUCCESSFUL:	<pre>{"EntryId": "27", "EntryParent": null, "EntryServerId": "1", "EntryName": "mike", "EntryType": "host", "EntryNetmask": "255.255.255.0", "EntryIPCount": "1", "EntryPercent": "1", "Options": [{"OptionId": "46", "OptionSubnetId": "27", "OptionKey": "hardware ethernet", "OptionValue": "11:23:45:67:89:ab"}, {"OptionId": "47", "OptionSubnetId": "27", "OptionKey": "fixed-address", "OptionValue": "10.20.30.158"}]}</pre>	ERROR:	<pre>{"success": 0, "message": "error message"}</pre>
SUCCESSFUL:	<pre>{"EntryId": "27", "EntryParent": null, "EntryServerId": "1", "EntryName": "mike", "EntryType": "host", "EntryNetmask": "255.255.255.0", "EntryIPCount": "1", "EntryPercent": "1", "Options": [{"OptionId": "46", "OptionSubnetId": "27", "OptionKey": "hardware ethernet", "OptionValue": "11:23:45:67:89:ab"}, {"OptionId": "47", "OptionSubnetId": "27", "OptionKey": "fixed-address", "OptionValue": "10.20.30.158"}]}</pre>				
ERROR:	<pre>{"success": 0, "message": "error message"}</pre>				

Name	Type	Description
EntryId	INT	The ID of the DHCP Entry.
EntryParent	INT	The parent Entry of this one
EntryServerId	INT	The DHCP Server to which this entry belongs.
EntryType	STRING	The Entry type. Either 'host' or 'subnet'.
EntryName	STRING	The name of this entry. In the case of a Host, it is the hostname. In the case of a subnet, it is the subnet address.
EntryNetmask	STRING	The subnet mask. Empty on type 'host'
EntryIPCount	INT	The number of IPs in this Entry.
EntryPercent	INT	Percentage of this Entry currently assigned.
Options	STRING	If present, this array contains objects enumerating each option and its type.
OptionId	STRING	The ID of this Option
OptionSubnetId	INT	The ID of the parent. Identical to EntryId.
OptionKey	STRING	The key portion of the option key-value pairing.
OptionValue	STRING	The value portion of the option key-value pairing.

Optional Parameters

Name	Type	Example	Description
likeFlag	BOOL	1	When 1, string searches are done via LIKE with wildcards at both ends. When 0, strict comparison is used.
generalFlag	BOOL	1	When 1, searches over the provided parameters using OR. If 0 or omitted, uses AND.

Name	Type	Example	Description
EntryId	INT	123	The ID of the DHCP Entry to search for.
EntryParent	INT	123	The parent Entry to search for.
EntryServerId	INT	123	The DHCP Server to search for.
EntryType	STRING	subnet	The Entry type to search for.
EntryName	STRING	30.20.10.1	The name to search for.
EntryNetmask	STRING	255.255.255.0	The subnet mask to search for.
OptionId	STRING	123	The Option ID to search for.
OptionKey	STRING	range	The key portion of the option key-value pairing to search for.
OptionValue	STRING	30.20.10.10 30.20.10.40	The value portion of the option key-value pairing to search for.

**update**

URL

`/api/v1/api.php?target=DHCPEntry&action=update`

Description

First performs a search based on the submitted DHCP Entry criteria, then performs an Update across those entries and all found Options based on new values.

Returns

**Examples:**

SUCCESSFUL:

```
{"success":1,"message":"Update Successful."}
```

ERROR:

```
{"success":0, "message":"error message"}
```

Optional Parameters

Name	Type	Example	Description
likeFlag	BOOL	1	When 1, string searches are done via LIKE with wildcards at both ends. When 0, strict comparison is used.

Name	Type	Example	Description
SearchId	INT	123	The ID of the DHCP Entry to search for.
SearchParent	INT	123	The parent Entry to search for.
SearchServer Id	INT	123	The DHCP Server to search for.
SearchType	STRING	subnet	The Entry type to search for.
SearchName	STRING	30.20.10.1	The name to search for.
SearchNetmask	STRING	255.255.255.0	The subnet mask to search for.
SearchId	STRING	123	The Option ID to search for.
SearchKey	STRING	range	The key portion of the option key-value pairing to search for.
SearchValue	STRING	30.20.10.10 30.20.10.40	The value portion of the option key-value pairing to search for.

Name	Type	Example	Description
UpdateParent	INT	123	The new parent data.
UpdateServerId	INT	123	The new DHCP Server ID.
UpdateType	STRING	subnet	The new Entry type.
UpdateName	STRING	30.20.10.1	The new name.
UpdateNetmask	STRING	255.255.255.0	The new subnet mask.
UpdateKey	STRING	range	The new key portion of the option key-value pairing.
UpdateValue	STRING	30.20.10.10 30.20.10.40	The new value portion of the option key-value pairing.

#### updateOption

URL

/api/v1/api.php?target=DHCPEntry&action=updateOption

Description

First performs a search based on the submitted DHCP Entry criteria, then performs an Update across all found Options, without altering found Entries.

Returns

**Examples:**

SUCCESSFUL:

```
{"success":1,"message":"Update Successful."}
```

ERROR:

```
{"success":0,"message":"error message"}
```

Optional Parameters

Name	Type	Example	Description
likeFlag	BOOL	1	When 1, string searches are done via LIKE with wildcards at both ends. When 0, strict comparison is used.

Name	Type	Example	Description
SearchId	INT	123	The ID of the DHCP Entry to search for.
SearchParent	INT	123	The parent Entry to search for.
SearchServer Id	INT	123	The DHCP Server to search for.
SearchType	STRING	subnet	The Entry type to search for.
SearchName	STRING	30.20.10.1	The name to search for.
SearchNetmask	STRING	255.255.255.0	The subnet mask to search for.
SearchId	STRING	123	The Option ID to search for.
SearchKey	STRING	range	The key portion of the option key-value pairing to search for.
SearchValue	STRING	30.20.10.10 30.20.10.40	The value portion of the option key-value pairing to search for.

Name	Type	Example	Description
UpdateKey	STRING	range	The new key portion of the option key-value pairing.
UpdateValue	STRING	30.20.10.10 30.20.10.40	The new value portion of the option key-value pairing.

#### add

URL

/api/v1/api.php?target=DHCPEntry&action=add

Description

Adds a new DHCP Entry and returns the new ID.

Returns	<p><b>Examples:</b></p> <table border="1"> <tr> <td data-bbox="824 184 1154 268">SUCCESSFUL:</td> <td data-bbox="1154 184 1481 268"><i>{"success":1,"message":"Add Successful.", "data":123}</i></td> </tr> <tr> <td data-bbox="824 268 1154 352">ERROR:</td> <td data-bbox="1154 268 1481 352"><i>{"success":0, "message":"error message"}</i></td> </tr> </table> <p><b>Data Detail:</b></p> <table border="1"> <thead> <tr> <th data-bbox="824 453 1042 501">Name</th> <th data-bbox="1042 453 1261 501">Type</th> <th data-bbox="1261 453 1481 501">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="824 501 1042 585">data</td> <td data-bbox="1042 501 1261 585">INT</td> <td data-bbox="1261 501 1481 585">The ID of the new DHCP Entry.</td> </tr> </tbody> </table>	SUCCESSFUL:	<i>{"success":1,"message":"Add Successful.", "data":123}</i>	ERROR:	<i>{"success":0, "message":"error message"}</i>	Name	Type	Description	data	INT	The ID of the new DHCP Entry.						
SUCCESSFUL:	<i>{"success":1,"message":"Add Successful.", "data":123}</i>																
ERROR:	<i>{"success":0, "message":"error message"}</i>																
Name	Type	Description															
data	INT	The ID of the new DHCP Entry.															
Required Parameters	<table border="1"> <thead> <tr> <th data-bbox="824 646 992 695">Name</th> <th data-bbox="992 646 1154 695">Type</th> <th data-bbox="1154 646 1317 695">Example</th> <th data-bbox="1317 646 1481 695">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="824 695 992 842">EntryServerId</td> <td data-bbox="992 695 1154 842">INT</td> <td data-bbox="1154 695 1317 842">123</td> <td data-bbox="1317 695 1481 842">The DHCP Server this new Entry belongs to.</td> </tr> <tr> <td data-bbox="824 842 992 957">EntryType</td> <td data-bbox="992 842 1154 957">STRING</td> <td data-bbox="1154 842 1317 957">subnet</td> <td data-bbox="1317 842 1481 957">The Entry type of this new Entry.</td> </tr> <tr> <td data-bbox="824 957 992 1066">EntryName</td> <td data-bbox="992 957 1154 1066">STRING</td> <td data-bbox="1154 957 1317 1066">30.20.10.1</td> <td data-bbox="1317 957 1481 1066">The name of this new Entry.</td> </tr> </tbody> </table>	Name	Type	Example	Description	EntryServerId	INT	123	The DHCP Server this new Entry belongs to.	EntryType	STRING	subnet	The Entry type of this new Entry.	EntryName	STRING	30.20.10.1	The name of this new Entry.
Name	Type	Example	Description														
EntryServerId	INT	123	The DHCP Server this new Entry belongs to.														
EntryType	STRING	subnet	The Entry type of this new Entry.														
EntryName	STRING	30.20.10.1	The name of this new Entry.														
Optional Parameters	<table border="1"> <thead> <tr> <th data-bbox="824 1129 992 1178">Name</th> <th data-bbox="992 1129 1154 1178">Type</th> <th data-bbox="1154 1129 1317 1178">Example</th> <th data-bbox="1317 1129 1481 1178">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="824 1178 992 1293">EntryParent</td> <td data-bbox="992 1178 1154 1293">INT</td> <td data-bbox="1154 1178 1317 1293">123</td> <td data-bbox="1317 1178 1481 1293">The parent Entry to search for.</td> </tr> <tr> <td data-bbox="824 1293 992 1402">EntryNetmask</td> <td data-bbox="992 1293 1154 1402">STRING</td> <td data-bbox="1154 1293 1317 1402">255.255.255.0</td> <td data-bbox="1317 1293 1481 1402">The subnet mask of this new Entry.</td> </tr> </tbody> </table>	Name	Type	Example	Description	EntryParent	INT	123	The parent Entry to search for.	EntryNetmask	STRING	255.255.255.0	The subnet mask of this new Entry.				
Name	Type	Example	Description														
EntryParent	INT	123	The parent Entry to search for.														
EntryNetmask	STRING	255.255.255.0	The subnet mask of this new Entry.														

<b>addOption</b>	
URL	/api/v1/api.php?target=DHCPEntry&action=addOption
Description	Creates a new DHCP Option and returns the new ID.

Returns	<p><b>Examples:</b></p> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td><code>{"success":1,"message":"Add Successful.", "data":123}</code></td> </tr> <tr> <td>ERROR:</td> <td><code>{"success":0, "message":"error message"}</code></td> </tr> </table> <p><b>Data Detail:</b></p> <table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>data</td> <td>INT</td> <td>The ID of the new DHCP Option.</td> </tr> </tbody> </table>	SUCCESSFUL:	<code>{"success":1,"message":"Add Successful.", "data":123}</code>	ERROR:	<code>{"success":0, "message":"error message"}</code>	Name	Type	Description	data	INT	The ID of the new DHCP Option.
SUCCESSFUL:	<code>{"success":1,"message":"Add Successful.", "data":123}</code>										
ERROR:	<code>{"success":0, "message":"error message"}</code>										
Name	Type	Description									
data	INT	The ID of the new DHCP Option.									

Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>OptionSubnet Id</td> <td>INT</td> <td>123</td> <td>The DHCP Entry this Option belongs to.</td> </tr> <tr> <td>OptionKey</td> <td>STRING</td> <td>range</td> <td>The key portion of the option key-value pairing to search for.</td> </tr> <tr> <td>OptionValue</td> <td>STRING</td> <td>30.20.10.10 30.20.10.40</td> <td>The value portion of the option key-value pairing to search for.</td> </tr> </tbody> </table>	Name	Type	Example	Description	OptionSubnet Id	INT	123	The DHCP Entry this Option belongs to.	OptionKey	STRING	range	The key portion of the option key-value pairing to search for.	OptionValue	STRING	30.20.10.10 30.20.10.40	The value portion of the option key-value pairing to search for.
Name	Type	Example	Description														
OptionSubnet Id	INT	123	The DHCP Entry this Option belongs to.														
OptionKey	STRING	range	The key portion of the option key-value pairing to search for.														
OptionValue	STRING	30.20.10.10 30.20.10.40	The value portion of the option key-value pairing to search for.														

delete					
URL	/api/v1/api.php?target=DHCPEntry&action=delete				
Description	Performs a search over the DHCP Entry dataset and deletes all found matches, along with their associated Options.				
Returns	<p><b>Examples:</b></p> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td><code>{"success":1,"message":"DHCPEntries(s) Deleted."}</code></td> </tr> <tr> <td>ERROR:</td> <td><code>{"success":0, "message":"error message"}</code></td> </tr> </table>	SUCCESSFUL:	<code>{"success":1,"message":"DHCPEntries(s) Deleted."}</code>	ERROR:	<code>{"success":0, "message":"error message"}</code>
SUCCESSFUL:	<code>{"success":1,"message":"DHCPEntries(s) Deleted."}</code>				
ERROR:	<code>{"success":0, "message":"error message"}</code>				

Optional Parameters

Name	Type	Example	Description
likeFlag	BOOL	1	When 1, string searches are done via LIKE with wildcards at both ends. When 0, strict comparison is used.

Name	Type	Example	Description
EntryId	INT	123	The ID of the DHCP Entry to search for.
EntryParent	INT	123	The parent Entry to search for.
EntryServerId	INT	123	The DHCP Server to search for.
EntryType	STRING	subnet	The Entry type to search for.
EntryName	STRING	30.20.10.1	The name to search for.
EntryNetmask	STRING	255.255.255.0	The subnet mask to search for.
OptionId	STRING	123	The Option ID to search for.
OptionKey	STRING	range	The key portion of the option key-value pairing to search for.
OptionValue	STRING	30.20.10.10 30.20.10.40	The value portion of the option key-value pairing to search for.

**deleteOption**

URL

/api/v1/api.php?target=DHCPEntry&action=deleteOption

Description

Performs a search over the DHCP Entry dataset and deletes all found Options while leaving the Entries intact.

Returns

**Examples:**

SUCCESSFUL:

```
{"success":1,"message":"DHCP  
Option(s) Deleted."}
```

ERROR:

```
{"success":0, "message":"error  
message"}
```

## Optional Parameters

Name	Type	Example	Description
likeFlag	BOOL	1	When 1, string searches are done via LIKE with wildcards at both ends. When 0, strict comparison is used.

Name	Type	Example	Description
EntryId	INT	123	The ID of the DHCP Entry to search for.
EntryParent	INT	123	The parent Entry to search for.
EntryServerId	INT	123	The DHCP Server to search for.
EntryType	STRING	subnet	The Entry type to search for.
EntryName	STRING	30.20.10.1	The name to search for.
EntryNetmask	STRING	255.255.255.0	The subnet mask to search for.
OptionId	STRING	123	The Option ID to search for.
OptionKey	STRING	range	The key portion of the option key-value pairing to search for.
OptionValue	STRING	30.20.10.10 30.20.10.40	The value portion of the option key-value pairing to search for.

## API Module - DNS

- [DNS Server Control](#)
- [DNS Zone Control](#)
- [DNS Record Control](#)
- [Server-Zone Linkage](#)

- Name Server Control

## DNS Server Control

get					
URL	/api/v1/api.php?target=dnsServer&action=get				
Description	If provided with an id, fetches that DNS Server from the database. If not, fetches a list of all stored DNS Servers				
Returns	<p><b>Examples:</b></p> <table border="1"> <tbody> <tr> <td>SUCCESSFUL:</td> <td> <pre>{   "success": 1,   "message": "Fetch Successful.",   "data": [     {       "id": "10",       "server": "mrbomasm-dns-4.onnet.net",       "username": "user",       "password": "vwvddp",       "port": "2600",       "customer_name": null,       "transfer_type": "SCP",       "remote_directory": "zones",       "named_conf_path": "Vetc/zones",       "active": "1",       "post_command": null,       "pre_command": null,       "dyn_DNSSEC_contact": null,       "powerdns_backend": "Bind",       "db_username": null,       "db_password": "",       "db_port": null,       "db_name": null,       "server_type": "slave",       "SOA": null,       "master_id": null,       "options": {         "\customer_name": "\",         "server_type": "slave",         "SOA": "\",         "remote_directory": "zones",         "named_conf_path": "\Vetc\Vzones",         "dyn_DNSSEC_contact": "\",         "post_command": "\",         "pre_command": "\",         "powerdns_backend": "Bind",         "db_username": "\",         "db_password": "\",         "db_port": "\",         "db_name": "\",         "enable_views": "1"       }     },     {       "testID": "963",       "zoneCount": "8",       "views": [         {           "id": "1",           "server_id": "10",           "name": "_6connectDefault",           "extras": [             "\description": "\",             "timestamp": "1371789181",             {               "id": "3",               "server_id": "10",               "name": "internal",               "extras": [                 "\description": null,                 "timestamp": "1374686650"               ]             }           ]         }       ]     }   ] }</pre> </td> </tr> <tr> <td>ERROR:</td> <td> <pre>{   "success": 0,   "message": "error message" }</pre> </td> </tr> </tbody> </table> <p><b>Data Detail:</b></p>	SUCCESSFUL:	<pre>{   "success": 1,   "message": "Fetch Successful.",   "data": [     {       "id": "10",       "server": "mrbomasm-dns-4.onnet.net",       "username": "user",       "password": "vwvddp",       "port": "2600",       "customer_name": null,       "transfer_type": "SCP",       "remote_directory": "zones",       "named_conf_path": "Vetc/zones",       "active": "1",       "post_command": null,       "pre_command": null,       "dyn_DNSSEC_contact": null,       "powerdns_backend": "Bind",       "db_username": null,       "db_password": "",       "db_port": null,       "db_name": null,       "server_type": "slave",       "SOA": null,       "master_id": null,       "options": {         "\customer_name": "\",         "server_type": "slave",         "SOA": "\",         "remote_directory": "zones",         "named_conf_path": "\Vetc\Vzones",         "dyn_DNSSEC_contact": "\",         "post_command": "\",         "pre_command": "\",         "powerdns_backend": "Bind",         "db_username": "\",         "db_password": "\",         "db_port": "\",         "db_name": "\",         "enable_views": "1"       }     },     {       "testID": "963",       "zoneCount": "8",       "views": [         {           "id": "1",           "server_id": "10",           "name": "_6connectDefault",           "extras": [             "\description": "\",             "timestamp": "1371789181",             {               "id": "3",               "server_id": "10",               "name": "internal",               "extras": [                 "\description": null,                 "timestamp": "1374686650"               ]             }           ]         }       ]     }   ] }</pre>	ERROR:	<pre>{   "success": 0,   "message": "error message" }</pre>
SUCCESSFUL:	<pre>{   "success": 1,   "message": "Fetch Successful.",   "data": [     {       "id": "10",       "server": "mrbomasm-dns-4.onnet.net",       "username": "user",       "password": "vwvddp",       "port": "2600",       "customer_name": null,       "transfer_type": "SCP",       "remote_directory": "zones",       "named_conf_path": "Vetc/zones",       "active": "1",       "post_command": null,       "pre_command": null,       "dyn_DNSSEC_contact": null,       "powerdns_backend": "Bind",       "db_username": null,       "db_password": "",       "db_port": null,       "db_name": null,       "server_type": "slave",       "SOA": null,       "master_id": null,       "options": {         "\customer_name": "\",         "server_type": "slave",         "SOA": "\",         "remote_directory": "zones",         "named_conf_path": "\Vetc\Vzones",         "dyn_DNSSEC_contact": "\",         "post_command": "\",         "pre_command": "\",         "powerdns_backend": "Bind",         "db_username": "\",         "db_password": "\",         "db_port": "\",         "db_name": "\",         "enable_views": "1"       }     },     {       "testID": "963",       "zoneCount": "8",       "views": [         {           "id": "1",           "server_id": "10",           "name": "_6connectDefault",           "extras": [             "\description": "\",             "timestamp": "1371789181",             {               "id": "3",               "server_id": "10",               "name": "internal",               "extras": [                 "\description": null,                 "timestamp": "1374686650"               ]             }           ]         }       ]     }   ] }</pre>				
ERROR:	<pre>{   "success": 0,   "message": "error message" }</pre>				

Name	Type	Description
id	INT	Server ID
server	STRING	Server Name
username	STRING	Login Name
password	CRYPT	Login Password
port	INT	Port the Server listens on
zoneCount	INT	The number of zones attached to this server.
options	JSON	<p>The options entry is a JSON-encoded string containing a variety of server-specific configuration options.</p> <p>This string will vary widely by server type and configuration. The following are a selection of common settings.</p>
transfer_type	STRING	The type of server this is. Valid settings include SCP, PowerDNS, Secure64, Secure64Signer
server_type	STRING	Whether this server is a master or a slave server
SOA	STRING	The SOA entry to be used for zones on this server
remote_directory	STRING	The directory where SCP will place the zone files.
named_conf_path	STRING	The zone file location used when writing the server configuration file

pre_command	STRING	The command executed on the server before the zones are transferred
post_command	STRING	The command executed on the server after the transfer is complete
enable_views	INT	Whether or not Views are enabled
views	JSON	The views entry is a JSON-encoded string containing all the information about the Views attached to this server, if any exist.
id	INT	The View ID
server_id	INT	The ID of the server the View is attached to
name	STRING	The name of the View
description	STRING	A description of the View
timestamp	INT	The UNIX timestamp of when the view was created.
extras	JSON	A JSON-encoded array of the extra attributes printed out in the view definition in the config file.

Optional Parameters

Name	Type	Example	Description
id	INTEGER	15	The server id to fetch.

<b>add</b>	
URL	/api/v1/api.php?target=dnsServer&action=add
Description	Adds a new DNS Server

Returns	<p><b>Examples:</b></p> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td>{"success":1,"message":"Add Successful."}</td> </tr> <tr> <td>ERROR:</td> <td>{"success":0, "message":"error message"}</td> </tr> </table>	SUCCESSFUL:	{"success":1,"message":"Add Successful."}	ERROR:	{"success":0, "message":"error message"}																				
SUCCESSFUL:	{"success":1,"message":"Add Successful."}																								
ERROR:	{"success":0, "message":"error message"}																								
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>server</td> <td>STRING</td> <td>dns.yourdomain.com</td> <td>Address of the DNS Server</td> </tr> <tr> <td>username</td> <td>STRING</td> <td>bobuser</td> <td>Login name for Server</td> </tr> <tr> <td>password</td> <td>STRING</td> <td>password1</td> <td>Login password for Server</td> </tr> <tr> <td>transferType</td> <td>STRING</td> <td>SCP</td> <td>Protocol</td> </tr> </tbody> </table>	Name	Type	Example	Description	server	STRING	dns.yourdomain.com	Address of the DNS Server	username	STRING	bobuser	Login name for Server	password	STRING	password1	Login password for Server	transferType	STRING	SCP	Protocol				
Name	Type	Example	Description																						
server	STRING	dns.yourdomain.com	Address of the DNS Server																						
username	STRING	bobuser	Login name for Server																						
password	STRING	password1	Login password for Server																						
transferType	STRING	SCP	Protocol																						
Optional Parameters	<p>There optional parameters vary according to what type of server is being configured.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>customerName</td> <td>STRING</td> <td>/tmp/zones</td> <td>Customer Name</td> </tr> <tr> <td>remoteDirectory</td> <td>STRING</td> <td>/tmp/zones</td> <td>Zone Directory on Server</td> </tr> <tr> <td>namedConfPath</td> <td>STRING</td> <td>/tmp</td> <td>Zone Path</td> </tr> <tr> <td>preCommand</td> <td>STRING</td> <td>/path/to/stuff/precommand</td> <td></td> </tr> <tr> <td>postCommand</td> <td>STRING</td> <td>/path/to/stuff/postcommand</td> <td></td> </tr> </tbody> </table>	Name	Type	Example	Description	customerName	STRING	/tmp/zones	Customer Name	remoteDirectory	STRING	/tmp/zones	Zone Directory on Server	namedConfPath	STRING	/tmp	Zone Path	preCommand	STRING	/path/to/stuff/precommand		postCommand	STRING	/path/to/stuff/postcommand	
Name	Type	Example	Description																						
customerName	STRING	/tmp/zones	Customer Name																						
remoteDirectory	STRING	/tmp/zones	Zone Directory on Server																						
namedConfPath	STRING	/tmp	Zone Path																						
preCommand	STRING	/path/to/stuff/precommand																							
postCommand	STRING	/path/to/stuff/postcommand																							

<b>delete</b>					
URL	/api/v1/api.php?target=dnsServer&action=delete				
Description	Deletes a DNS Server				
Returns	<p><b>Examples:</b></p> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td>{"success":1,"message":"Delete Successful."}</td> </tr> <tr> <td>ERROR:</td> <td>{"success":0, "message":"error message"}</td> </tr> </table>	SUCCESSFUL:	{"success":1,"message":"Delete Successful."}	ERROR:	{"success":0, "message":"error message"}
SUCCESSFUL:	{"success":1,"message":"Delete Successful."}				
ERROR:	{"success":0, "message":"error message"}				

Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>NUMERIC</td> <td>5</td> <td>ID of server to delete.</td> </tr> </tbody> </table>	Name	Type	Example	Description	id	NUMERIC	5	ID of server to delete.
	Name	Type	Example	Description					
id	NUMERIC	5	ID of server to delete.						
Example URL	/api/v1/api.php?target=dnsServer&action=delete&id=5								

### update

URL	/api/v1/api.php?target=dnsServer&action=update				
Description	Updates an existing DNS Server with new information.				
Returns	<p><b>Examples:</b></p> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td>{"success":1, "message":"Update Successful."}</td> </tr> <tr> <td>ERROR:</td> <td>{"success":0, "message":"error message"}</td> </tr> </table>	SUCCESSFUL:	{"success":1, "message":"Update Successful."}	ERROR:	{"success":0, "message":"error message"}
SUCCESSFUL:	{"success":1, "message":"Update Successful."}				
ERROR:	{"success":0, "message":"error message"}				

Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>username</td> <td>STRING</td> <td>bobuser</td> <td>Login name for Server</td> </tr> <tr> <td>password</td> <td>STRING</td> <td>password1</td> <td>Login password for Server</td> </tr> <tr> <td>transferType</td> <td>STRING</td> <td>SCP</td> <td>Protocol</td> </tr> </tbody> </table>	Name	Type	Example	Description	username	STRING	bobuser	Login name for Server	password	STRING	password1	Login password for Server	transferType	STRING	SCP	Protocol
	Name	Type	Example	Description													
	username	STRING	bobuser	Login name for Server													
	password	STRING	password1	Login password for Server													
transferType	STRING	SCP	Protocol														

Optional Parameters	There optional parameters vary according to what type of server is being configured.																								
	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>customerName</td> <td>STRING</td> <td>/tmp/zones</td> <td>Customer Name</td> </tr> <tr> <td>remoteDirectory</td> <td>STRING</td> <td>/tmp/zones</td> <td>Zone Directory on Server</td> </tr> <tr> <td>namedConfPath</td> <td>STRING</td> <td>/tmp</td> <td>Zone Path</td> </tr> <tr> <td>preCommand</td> <td>STRING</td> <td>/path/to/stuff/precommand</td> <td></td> </tr> <tr> <td>postCommand</td> <td>STRING</td> <td>/path/to/stuff/postcommand</td> <td></td> </tr> </tbody> </table>	Name	Type	Example	Description	customerName	STRING	/tmp/zones	Customer Name	remoteDirectory	STRING	/tmp/zones	Zone Directory on Server	namedConfPath	STRING	/tmp	Zone Path	preCommand	STRING	/path/to/stuff/precommand		postCommand	STRING	/path/to/stuff/postcommand	
	Name	Type	Example	Description																					
	customerName	STRING	/tmp/zones	Customer Name																					
	remoteDirectory	STRING	/tmp/zones	Zone Directory on Server																					
	namedConfPath	STRING	/tmp	Zone Path																					
	preCommand	STRING	/path/to/stuff/precommand																						
postCommand	STRING	/path/to/stuff/postcommand																							

### transferAll

URL	/api/v1/api.php?target=dnsServer&action=transferServer
-----	--

Description	Performs a full zone push on a DNS Server, executing pre and post commands, transferring files, and restarting services.										
Returns	<b>Examples:</b> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td colspan="2">{"success":1,"message":"Transfer Successful."}</td> </tr> <tr> <td>ERROR:</td> <td colspan="2">{"success":0,"message":"error message"}</td> </tr> </table>			SUCCESSFUL:	{"success":1,"message":"Transfer Successful."}		ERROR:	{"success":0,"message":"error message"}			
SUCCESSFUL:	{"success":1,"message":"Transfer Successful."}										
ERROR:	{"success":0,"message":"error message"}										
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>push</td> <td>INT</td> <td>1</td> <td>The ID of the server to push zones to</td> </tr> </tbody> </table>			Name	Type	Example	Description	push	INT	1	The ID of the server to push zones to
Name	Type	Example	Description								
push	INT	1	The ID of the server to push zones to								

### transferSingle

URL	/api/v1/api.php?target=dnsServer&action=transferSingle										
Description	<p>Transfers a single Zone file to all its associated DNS Servers, along with updated server configurations.</p> <p>Performs pre and post commands on the target servers, transfers the zone file(s), and restarts services.</p>										
Returns	<b>Examples:</b> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td colspan="2">{"success":1,"message":"Updated Zone: \$name.zone on \$server via SCP"}</td> </tr> <tr> <td>ERROR:</td> <td colspan="2">{"success":0,"message":"error message"}</td> </tr> </table>			SUCCESSFUL:	{"success":1,"message":"Updated Zone: \$name.zone on \$server via SCP"}		ERROR:	{"success":0,"message":"error message"}			
SUCCESSFUL:	{"success":1,"message":"Updated Zone: \$name.zone on \$server via SCP"}										
ERROR:	{"success":0,"message":"error message"}										
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>zoneID</td> <td>NUMERIC</td> <td>35</td> <td>The ID of the zone to push</td> </tr> </tbody> </table>			Name	Type	Example	Description	zoneID	NUMERIC	35	The ID of the zone to push
Name	Type	Example	Description								
zoneID	NUMERIC	35	The ID of the zone to push								

## DNS Zone Control

### get

URL	/api/v1/api.php?target=zone&action=get		
Description	<p>Accepts search criteria to retrieve a list of all matching DNS Zones and associated Records.</p> <p>Search can be performed on any combination of Zone and Record attributes.</p>		
Returns	<b>Examples:</b>		

SUCCESSFUL:	{       "success":1,"message":"Search Successful.",       "data":{"zoneId":"932","zoneName":"185.160.209.in-addr.arpa",       "zoneResourceId":"81",       "zoneSerial":"2013040302","zoneRefresh":"28800",       "zoneRetry":"7200","zoneExpire":"604800",       "zoneMinimum":"86400","zoneSOA":null,       "zoneTags":null,"zoneTTL":"28800",       "zoneAutoCheck":"1","zoneEnableDNSSEC":null,       "recordId":"154110","recordZoneId":"932",       "recordHost":"185.160.209.inaddr.arpa.",       "recordType":"NS","recordValue":"auth01.verosity.net.",       "recordDescription":null,"recordTTL":"28800",       "recordOrdering":"1",       "recordErrors":null,"assetId":"0",       "userCanCreate":0,"userCanDelete":1,"userCanUpdate":1}}
ERROR:	{       "success":0,"message":"error message"}

**Data Detail:**

Name	Type	Description
zoneId	INT	The Id of the Zone entry. A single Zone entry might have multiple Records.
zoneName	STRING	The Zone name.
zoneResourceId	INT	The resource Id associated with this Zone.
zoneSerial	INT	Zone Serial.
zoneRefresh	INT	Zone Refresh.
zoneRetry	INT	Zone Retry.
zoneExpire	INT	Zone Expire.
zoneMinimum	INT	Zone Minimum.
zoneSOA	STRING	Zone SOA.
zoneTags	STRING	All the tags associated with this Zone.
zoneTTL	STRING	Zone TTL.

zoneEnableDNSSEC	BOOL	Whether or not DNSSEC is enabled for this Zone.
zoneAutoCheck	BOOL	Whether or not this zone is configured to be automatically validated on load/edit.
recordId	INT	The Id of this Record Entry. It is always included with its parent Zone.
recordHost	STRING	The Hostname of this Record.
recordType	STRING	The Record Type (MX,NS,A,PTR,etc)
recordValue	STRING	The Value of this Record.
recordDescription	STRING	A short description of this Record.
recordTTL	STRING	The TTL of this Record.
recordOrdering	INT	The numerical order in which the record appears in the zone.
recordErrors	STRING	A string containing any detected problems with this record
userCanCreate	BOOL	Whether or not the user has DNS CREATE permissions on this zone's resource
userCanUpdate	BOOL	Whether or not the user has DNS UPDATE permissions on this zone's resource
userCanDelete	BOOI	Whether or not the user has DNS DELETE permissions on this zone's resource

unpagedRows	INT	If pagination is used, this value will contain a total count of records had the pagination not been used.
-------------	-----	---

Optional Parameters

Name	Type	Example	Description
likeFlag	BOOL	1	When 1, string searches are done via LIKE with wildcards at both ends. When 0, strict comparison is used.
generalFlag	BOOL	1	When 1, searches over the provided parameters using OR. If 0 or omitted, uses AND.
selectCount	INT	30	When supplied only returns the first X entries
selectOffset	INT	10	When supplied, only returns entries after record X
sortBy	JSON	<code>{"zoneName": "desc", "zoneMask": "asc"}</code>	A JSON-encoded object containing a list of columns to sort on and the direction in which to sort. Any API variable may be used for sorting. Valid sort directions are ASC and DESC.

Name	Type	Example	Description
------	------	---------	-------------

zoneId	INT	123	The Zone Id to search for.
zoneName	STRING	foo	The Zone Name to search for.
zoneResourceId	INT	5	The Resource Id to search for.
zoneSerial	INT	2012033001	The Zone Serial to search for.
zoneRefresh	INT	36000	The Zone Refresh to search for.
zoneRetry	INT	800	The Zone Retry to search for.
zoneExpire	INT	6090000	The Zone Expire to search for.
zoneMinimum	INT	10	The Zone Minimum to search for.
zoneSOA	STRING	200	The Zone SOA to search for.
zoneTags	STRING	client,production	Zone Tags to search for.
zoneTTL	INT	3600	The Zone TTL to search for.
zoneEnabledDNSSEC	INT	1	Search based on DNSSEC settings.
recordId	INT	123	The Record Id to search for.
recordZoneId	INT	123	The parent Zone to search for.
recordHost	STRING	@	The Record Host to search for.
recordType	STRING	NS	The Record Type to search for.

recordValue	STRING	ns1.dns.6connect.com.	The Record Value to search for.
recordDescription	STRING	Description	Search based on Record Description.
recordTTL	STRING	3600	The Record TTL to search for.

## search

### URL

/api/v1/api.php?target=zone&action=search

### Description

Accepts search criteria to retrieve a list of all matching DNS Zones but NO associated Records. Search can be performed on any combination of Zone and Record attributes.

### Returns

#### Examples:

SUCCESSFUL:	{ "success":1, "message":"Search Successful.", "data":[{"zoneId":"123", "zoneName":"foobs.net", "zoneResourceId":"483", "zoneIpVer":null, "zoneMask":null, "zoneSerial":"2012121803", "zoneRefresh":null, "zoneRetry":null, "zoneExpire":null, "zoneMinimum":null, "zoneSOA":null, "zoneTags":["Aaron,Personal"], "zoneTTL":"3600", "zoneEnableDNSSEC":"1", "zoneLocalSigning":"1", "assetId":"0", "recordCount":"1", "unpaginatedRows":"215"}]}
ERROR:	{ "success":0, "message":"error message" }

#### Data Detail:

Name	Type	Description
zoneId	INT	The Id of the Zone entry. A single Zone entry might have multiple Records.
zoneName	STRING	The Zone name.
zoneResourceId	INT	The resource Id associated with this Zone.
zoneSerial	INT	Zone Serial.

zoneRefresh	INT	Zone Refresh.
zoneRetry	INT	Zone Retry.
zoneExpire	INT	Zone Expire.
zoneMinimum	INT	Zone Minimum.
zoneSOA	STRING	Zone SOA.
zoneTags	STRING	All the tags associated with this Zone.
zoneTTL	STRING	Zone TTL.
zoneEnableDNSSEC	BOOL	Whether or not DNSSEC is enabled for this Zone.
zoneAutoCheck	BOOL	Whether or not this zone is configured to be automatically validated on load/edit.
recordCount	int	How many records are associated with this zone.
userCanCreate	BOOL	Whether or not the user has DNS CREATE permissions on this zone's resource
userCanUpdate	BOOL	Whether or not the user has DNS UPDATE permissions on this zone's resource
userCanDelete	BOOL	Whether or not the user has DNS DELETE permissions on this zone's resource
unpagedRows	INT	If pagination is used, this value will contain a total count of records had the pagination not been used.

Optional Parameters

Name	Type	Example	Description
likeFlag	BOOL	1	When 1, string searches are done via LIKE with wildcards at both ends. When 0, strict comparison is used.
generalFlag	BOOL	1	When 1, searches over the provided parameters using OR. If 0 or omitted, uses AND.
selectCount	INT	30	When supplied only returns the first X entries
selectOffset	INT	10	When supplied, only returns entries after record X
sortBy	JSON	<pre>{"zoneName": "desc", "zoneMask": "asc"}</pre>	A JSON-encoded object containing a list of columns to sort on and the direction in which to sort. Any API variable may be used for sorting. Valid sort directions are ASC and DESC.

Name	Type	Example	Description
zoneId	INT	123	The Zone Id to search for.
zoneName	STRING	foo	The Zone Name to search for.
zoneResourceId	INT	5	The Resource Id to search for.

zoneSerial	INT	2012033001	The Zone Serial to search for.
zoneRefresh	INT	36000	The Zone Refresh to search for.
zoneRetry	INT	800	The Zone Retry to search for.
zoneExpire	INT	6090000	The Zone Expire to search for.
zoneMinimum	INT	10	The Zone Minimum to search for.
zoneSOA	STRING	200	The Zone SOA to search for.
zoneTags	STRING	client,production	Zone Tags to search for.
zoneTTL	INT	3600	The Zone TTL to search for.
zoneEnabledDNSSEC	INT	1	Search based on DNSSEC settings.
recordId	INT	123	The Record Id to search for.
recordZoneId	INT	123	The parent Zone to search for.
recordHost	STRING	@	The Record Host to search for.
recordType	STRING	NS	The Record Type to search for.
recordValue	STRING	ns1.dns.6connect.com.	The Record Value to search for.
recordDescription	STRING	Description	Search based on Record Description.
recordTTL	STRING	3600	The Record TTL to search for.

**update**

URL

/api/v1/api.php?target=zone&amp;action=update

Description

First performs a search based on the submitted Zone and Record criteria, then performs an Update across those entries based on new values.

Returns

**Examples:**

SUCCESSFUL:

{"success":1,"message":"Update Successful."}

ERROR:

{"success":0,"message":"error message"}

Optional Parameters

Name	Type	Example	Description
likeFlag	BOOL	1	When 1, string searches are done via LIKE with wildcards at both ends. When 0, strict comparison is used.
generalFlag	BOOL	1	When 1, searches over the provided parameters using OR. If 0 or omitted, uses AND.

Name	Type	Example	Description
searchZoneId	INT	123	The Zone Id to search for.
searchZoneName	STRING	foo	The Zone Name to search for.
searchZoneResourceId	INT	5	The Resource Id to search for.
searchZoneSerial	INT	2012033001	The Zone Serial to search for.
searchZoneRefresh	INT	36000	The Zone Refresh to search for.
searchZoneRetry	INT	800	The Zone Retry to search for.

searchZoneE xpire	INT	6090000	The Zone Expire to search for.
searchZoneM inimum	INT	10	The Zone Minimum to search for.
searchZoneS OA	STRING	200	The Zone SOA to search for.
searchZoneT ags	STRING	client,producti on	Zone Tags to search for.
searchZoneT TL	INT	3600	The Zone TTL to search for.
searchZoneE nableDNSSE C	INT	1	Search based on DNSSEC settings.
searchRecord Id	INT	123	The Record Id to search for.
searchRecord Host	STRING	@	The Record Host to search for.
searchRecord Type	STRING	NS	The Record Type to search for.
searchRecord Value	STRING	ns1.dns.6con nect.com.	The Record Value to search for.
searchRecord Description	STRING	Description	Search based on Record Description.
searchRecord TTL	STRING	3600	The Record TTL to search for.

Name	Type	Example	Description
updateZoneN ame	STRING	foo	The Zone name to replace into the searched rows.
updateZoneR esourceId	INT	5	The Resource Id to replace into the searched rows.

updateZoneSerial	INT	2012033001	The Zone Serial to replace into the searched rows.
updateZoneRefresh	INT	36000	The Zone Refresh to replace into the searched rows.
updateZoneRetry	INT	800	The Zone Retry to replace into the searched rows..
updateZoneExpire	INT	6090000	The Zone Expire to replace into the searched rows.
updateZoneMinimum	INT	10	The Zone Minimum to replace into the searched rows.
updateZoneSOA	STRING	200	The Zone SOA to replace into the searched rows.
updateZoneTags	STRING	client,production	Zone Tags to replace into the searched rows.
updateZoneTTL	INT	3600	The Zone TTL to replace into the searched rows.
updateZoneEnableDNSSEC	INT	1	Update DNSSEC Settings.
updateRecordHost	STRING	@	The Record Host to replace into the searched rows.
updateRecordType	STRING	NS	The Record Type to replace into the searched rows.

updateRecord Value	STRING	ns1.dns.6connect.com.	The Record Value to replace into the searched rows.
updateRecord Description	STRING	Description	Update Record Descriptions.
updateRecord TTL	STRING	3600	The Record TTL to replace into the searched rows.

<b>add</b>											
URL	/api/v1/api.php?target=zone&action=add										
Description	Adds a new DNS Zone.										
Returns	<p><b>Examples:</b></p> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td>{"success":1,"message":"Add Successful. ","data":123}</td> </tr> <tr> <td>ERROR:</td> <td>{"success":0, "message":"error message"}</td> </tr> </table> <p><b>Data Detail:</b></p> <table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>data</td> <td>INT</td> <td>The Id of the new Zone entry.</td> </tr> </tbody> </table>	SUCCESSFUL:	{"success":1,"message":"Add Successful. ","data":123}	ERROR:	{"success":0, "message":"error message"}	Name	Type	Description	data	INT	The Id of the new Zone entry.
SUCCESSFUL:	{"success":1,"message":"Add Successful. ","data":123}										
ERROR:	{"success":0, "message":"error message"}										
Name	Type	Description									
data	INT	The Id of the new Zone entry.									
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>zoneName</td> <td>STRING</td> <td>254.221.67.in-addr.arpa</td> <td>The name for the new Zone.</td> </tr> </tbody> </table>	Name	Type	Example	Description	zoneName	STRING	254.221.67.in-addr.arpa	The name for the new Zone.		
Name	Type	Example	Description								
zoneName	STRING	254.221.67.in-addr.arpa	The name for the new Zone.								

Optional Parameters

Name	Type	Example	Description
zoneResourceId	STRING	123	Resource Id for the new Zone.
zoneSerial	INT	2012033001	Serial for the new Zone.
zoneRefresh	INT	36000	Refresh for the new Zone.
zoneRetry	INT	800	Retry for the new Zone.
zoneExpire	INT	6090000	Expire for the new Zone.
zoneMinimum	INT	10	Minimum for the new Zone.
zoneSOA	STRING	200	SOA for the new Zone.
zoneTags	STRING	client,producti on	Tags for the new Zone.
zoneTTL	STRING	3600	TTL for the new Zone.
zoneEnabledDNSSEC	INT	1	Whether or not this new zone uses DNSSEC.
zoneAutoCheck	BOOL	1	Whether or not this zone is configured to be automatically validated on load/edit.

**delete**

URL

/api/v1/api.php?target=zone&action=delete

Description

Performs a search over the Zones and Records dataset and deletes all found Zones, plus all associated Records of those Zones.

Returns

**Examples:**

SUCCESSFUL:	{"success":1,"message":"Zones and Associated Records Deleted."}
ERROR:	{"success":0,"message":"error message"}

## Optional Parameters

Name	Type	Example	Description
deleteZoneId	INT	123	The Zone Id to search for.
deleteZoneName	STRING	foo	The Zone Name to search for.
deleteZoneResourceId	INT	5	The Resource Id to search for.
deleteZoneSerial	INT	2012033001	The Zone Serial to search for.
deleteZoneRefresh	INT	36000	The Zone Refresh to search for.
deleteZoneRetry	INT	800	The Zone Retry to search for.
deleteZoneExpire	INT	6090000	The Zone Expire to search for.
deleteZoneMinimum	INT	10	The Zone Minimum to search for.
deleteZoneSOA	STRING	200	The Zone SOA to search for.
deleteZoneTags	STRING	client,production	Zone Tags to search for.
deleteZoneTTL	INT	3600	The Zone TTL to search for.
deleteZoneEnableDNSSEC	INT	1	Search based on DNSSEC settings.
deleteRecordId	INT	123	The Record Id to search for.
deleteRecordHost	STRING	@	The Record Host to search for.
deleteRecordType	STRING	NS	The Record Type to search for.
deleteRecordValue	STRING	ns1.dns.6connect.com.	The Record Value to search for.

deleteRecord Description	STRING	Description	Search based on Record Description.
deleteRecord TTL	STRING	3600	The Record TTL to search for.

### getRecordTypes

URL	/api/v1/api.php?target=zone&action=getRecordTypes										
Description	Returns a list of all Record Types allowed by the system.										
Returns	<p><b>Examples:</b></p> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td><pre>{"success":1,"message":"Search Successful.,"data":{"recordType":"A"},{"recordType":"AAAA"}, {"recordType":"MX"}, {"recordType":"CNAME"}, {"recordType":"PTRG"}, {"recordType":"NS"}, {"recordType":"TXT"}, {"recordType":"DNSKEY"}, {"recordType":"SRV"}, {"recordType":"DS"}, {"recordType":"TEST"}]}</pre></td> </tr> <tr> <td>ERROR:</td> <td><pre>{"success":0, "message":"error message"}</pre></td> </tr> </table> <p><b>Data Detail:</b></p> <table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>recordType</td> <td>STRING</td> <td>A Record Type</td> </tr> </tbody> </table>	SUCCESSFUL:	<pre>{"success":1,"message":"Search Successful.,"data":{"recordType":"A"},{"recordType":"AAAA"}, {"recordType":"MX"}, {"recordType":"CNAME"}, {"recordType":"PTRG"}, {"recordType":"NS"}, {"recordType":"TXT"}, {"recordType":"DNSKEY"}, {"recordType":"SRV"}, {"recordType":"DS"}, {"recordType":"TEST"}]}</pre>	ERROR:	<pre>{"success":0, "message":"error message"}</pre>	Name	Type	Description	recordType	STRING	A Record Type
SUCCESSFUL:	<pre>{"success":1,"message":"Search Successful.,"data":{"recordType":"A"},{"recordType":"AAAA"}, {"recordType":"MX"}, {"recordType":"CNAME"}, {"recordType":"PTRG"}, {"recordType":"NS"}, {"recordType":"TXT"}, {"recordType":"DNSKEY"}, {"recordType":"SRV"}, {"recordType":"DS"}, {"recordType":"TEST"}]}</pre>										
ERROR:	<pre>{"success":0, "message":"error message"}</pre>										
Name	Type	Description									
recordType	STRING	A Record Type									

### getFile

URL	/api/v1/api.php?target=zone&action=getFile&zoneId=50								
Description	Returns a fully written zone file. If one does not exist, returns false.								
Returns	A Zone File								
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>zoneId</td> <td>INT</td> <td>50</td> <td>The Id of the zone to retrieve.</td> </tr> </tbody> </table>	Name	Type	Example	Description	zoneId	INT	50	The Id of the zone to retrieve.
Name	Type	Example	Description						
zoneId	INT	50	The Id of the zone to retrieve.						

### getDSFile

URL	/api/v1/api.php?target=zone&action=getDSFile&zoneId=50
-----	--

Description	Returns a fully written zone DS key file. If one does not exist, returns false.								
Returns	A Zone DS Key File								
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>zoneld</td> <td>INT</td> <td>50</td> <td>The Id of the zone whose DS keys wer are to retrieve.</td> </tr> </tbody> </table>	Name	Type	Example	Description	zoneld	INT	50	The Id of the zone whose DS keys wer are to retrieve.
Name	Type	Example	Description						
zoneld	INT	50	The Id of the zone whose DS keys wer are to retrieve.						

### checkZone

URL	/api/v1/api.php?target=zone&action=checkZone&zoneld=50								
Description	Runs a zone file through Named checkzone								
Returns	<p><b>Examples:</b></p> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td>{"success":1,"message":"No errors found."}</td> </tr> <tr> <td>ERROR:</td> <td>{"success":0,"message":"21: ignoring out-of-zone data (veggie.com) 22: ignoring out-of-zone data (veggie.com) dns_rdata_fromtext: 23: near '2001::db8:': bad IPv6 address dns_rdata_fromtext: 24: near '1.2.3.': bad dotted quad dns_rdata_fromtext: 25: near '2001::db8::\32': bad IPv6 address "}</td> </tr> </table>	SUCCESSFUL:	{"success":1,"message":"No errors found."}	ERROR:	{"success":0,"message":"21: ignoring out-of-zone data (veggie.com) 22: ignoring out-of-zone data (veggie.com) dns_rdata_fromtext: 23: near '2001::db8:': bad IPv6 address dns_rdata_fromtext: 24: near '1.2.3.': bad dotted quad dns_rdata_fromtext: 25: near '2001::db8::\32': bad IPv6 address "}				
SUCCESSFUL:	{"success":1,"message":"No errors found."}								
ERROR:	{"success":0,"message":"21: ignoring out-of-zone data (veggie.com) 22: ignoring out-of-zone data (veggie.com) dns_rdata_fromtext: 23: near '2001::db8:': bad IPv6 address dns_rdata_fromtext: 24: near '1.2.3.': bad dotted quad dns_rdata_fromtext: 25: near '2001::db8::\32': bad IPv6 address "}								
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>zoneld</td> <td>INT</td> <td>50</td> <td>The Id of the zone to check.</td> </tr> </tbody> </table>	Name	Type	Example	Description	zoneld	INT	50	The Id of the zone to check.
Name	Type	Example	Description						
zoneld	INT	50	The Id of the zone to check.						

### getArchivedZone

URL	/api/v1/api.php?target=zone&action=getArchivedZone
Description	Searches for all archived versions of the a zone. Zones are archived every time changes are pushed to their DNS Server.
Returns	<p><b>Examples:</b></p>

SUCCESSFUL:	<pre>{   "success": 1,   "message": "Search Successful.",   "data": {     "zoneArchived": "2768",     "zoneId": "1227",     "zoneArchiveTimestamp": "1375298692",     "zoneArchiveFingerprint": "d060e59d69606326d80b2e55b50f0bc9",     "zoneName": "6connect.com",     "zoneIpv": null,     "zoneMask": null,     "zoneSerial": "2013073105",     "zoneRefresh": "14400",     "zoneRetry": "2000",     "zoneExpire": "604800",     "zoneMinimum": "3600",     "zoneSOA": null,     "zoneTags": null,     "zoneTTL": "3600",     "zoneEnableDNSSEC": "1",     "zoneResourceId": "1013",     "zonePreviousViewLinkage": []   } }</pre>
ERROR:	<pre>{   "success": 0,   "message": "error message" }</pre>

**Data Detail:**

Name	Type	Description
zoneId	INT	The Id of the Zone entry to find archived versions of.
zoneArchiveId	INT	The ID of the Archive Entry
zoneArchiveTimestamp	INT	A timestamp marking when this zone was archived.
zoneArchiveFingerprint	STRING	A hash value identifying this zone. Used for comparing versions.
zoneName	INT	Zone Name.
zoneMask	INT	Zone Mask.
zoneSerial	INT	Zone Serial.
zoneRefresh	INT	Zone Refresh.
zoneRetry	INT	Zone Retry.
zoneExpire	INT	Zone Expire.
zoneMinimum	INT	Zone Minimum.
zoneSOA	STRING	Zone SOA.
zoneTags	STRING	Zone Tags.
zoneTTL	INT	Zone TTL.
zoneEnableDNSSEC	STRING	Whether or not this version had DNSSEC enabled.
zoneResourceId	STRING	Zone Resource ID
zonePreviousViewLinkage	JSON	A JSON-encoded array of views this zone was linked to.

## Optional Parameters

Name	Type	Example	Description
zoneId	INT	123	The Zone Id to search for.
zoneArchiveId	INT	123	The Zone Archive Id
zoneArchiveTimestamp	INT	2012033001	The Zone Archive Timestamp
fetchArchiveFile	BOOL	1	Whether or not to return the full Zone file with the result set..

## DNS Record Control

### get

URL

/api/v1/api.php?target=record&action=get

Description

Accepts search criteria to retrieve a list of all matching DNS Records. Search can be performed on any combination of Zone and Record attributes.

Returns

**Examples:**

SUCCESSFUL:	<pre>{"success":1,"message":"Search Successful.", "data":{"recordId": "30894", "recordZoneId": "229", "recordHost": "@", "recordType": "NS", "recordValue": "ns1.domain.com.", "recordDescription": "", "recordTTL": ""}}</pre>
ERROR:	<pre>{"success":0, "message":"error message"}</pre>

**Data Detail:**

Name	Type	Description
recordId	INT	The ID of this Record Entry. It is always included with its parent Zone.
recordZoneId	INT	The ID of this Record's parent Zone.
recordHost	STRING	The Hostname of this Record.
recordType	STRING	The Record Type (MX,NS,A,PTR,etc)
recordValue	STRING	The Value of this Record.
recordDescription	STRING	A short description of this Record.
recordTTL	STRING	The TTL of this Record.

Optional Parameters

Name	Type	Example	Description
likeFlag	BOOL	1	When 1, string searches are done via LIKE with wildcards at both ends. When 0, strict comparison is used.
selectCount	INT	30	When supplied only returns the first X entries
selectOffset	INT	10	When supplied, only returns entries after record X
sortBy	JSON	{"zoneName": "desc", "zoneMask": "asc"}	A JSON-encoded object containing a list of columns to sort on and the direction in which to sort. Any API variable may be used for sorting. Valid sort directions are ASC and DESC.

Name	Type	Example	Description
recordId	INT	123	The Record ID to search for.
recordZoneId	INT	123	The parent Zone to search for.
recordHost	STRING	@	The Record Host to search for.
recordType	STRING	NS	The Record Type to search for.
recordValue	STRING	ns1.dns.6connect.com.	The Record Value to search for.
recordDescription	STRING	Description	Search based on Record Description.
recordTTL	STRING	3600	The Record TTL to search for.

#### update

URL

/api/v1/api.php?target=record&action=update

Description

First performs a search based on the submitted Zone and Record criteria, then performs an Update across those entries based on new values.

Returns

**Examples:**

SUCCESSFUL:

```
{"success":1,"message":"Update Successful."}
```

ERROR:

```
{"success":0,"message":"error message"}
```

Optional Parameters

Name	Type	Example	Description
likeFlag	BOOL	1	When 1, string searches are done via LIKE with wildcards at both ends. When 0, strict comparison is used.
generalFlag	BOOL	1	When 1, searches over the provided parameters using OR. If 0 or omitted, uses AND.

Name	Type	Example	Description
searchZoneId	INT	123	The Zone ID to search for.
searchZoneName	STRING	foo	The Zone Name to search for.
searchZoneCustomerId	INT	5	The Customer ID to search for.
searchZoneIpVersion	STRING	IPv6	The IP Version to search for.
searchZoneMask	STRING		The Zone Mask to search for.
searchZoneSerial	INT	2012033001	The Zone Serial to search for.
searchZoneRefresh	INT	36000	The Zone Refresh to search for.
searchZoneRetry	INT	800	The Zone Retry to search for.
searchZoneExpire	INT	6090000	The Zone Expire to search for.
searchZoneMinimum	INT	10	The Zone Minimum to search for.

searchZoneSOA	STRING	200	The Zone SOA to search for.
searchZoneTags	STRING	client,production	Zone Tags to search for.
searchZoneTTL	INT	3600	The Zone TTL to search for.
searchZoneEnableDNSSEC	INT	1	Search based on DNSSEC settings.
searchRecordId	INT	123	The Record ID to search for.
searchRecordHost	STRING	@	The Record Host to search for.
searchRecordType	STRING	NS	The Record Type to search for.
searchRecordValue	STRING	ns1.dns.6connect.com.	The Record Value to search for.
searchRecordDescription	STRING	Description	Search based on Record Description.
searchRecordTTL	STRING	3600	The Record TTL to search for.

Name	Type	Example	Description
updateZoneName	STRING	foo	The Zone name to replace into the searched rows.
updateZoneCustId	INT	5	The Customer ID to replace into the searched rows.
updateZoneIpver	STRING	IPv6	The IP Version to replace into the searched rows.

updateZoneMask	STRING		The Zone Mask to replace into the searched rows.
updateZoneSerial	INT	2012033001	The Zone Serial to replace into the searched rows.
updateZoneRefresh	INT	36000	The Zone Refresh to replace into the searched rows.
updateZoneRetry	INT	800	The Zone Retry to replace into the searched rows..
updateZoneExpire	INT	6090000	The Zone Expire to replace into the searched rows.
updateZoneMinimum	INT	10	The Zone Minimum to replace into the searched rows.
updateZoneSOA	STRING	200	The Zone SOA to replace into the searched rows.
updateZoneTags	STRING	client,production	Zone Tags to replace into the searched rows.
updateZoneTTL	INT	3600	The Zone TTL to replace into the searched rows.
updateZoneEnableDNSSEC	INT	1	Update DNSSEC Settings.
updateRecordHost	STRING	@	The Record Host to replace into the searched rows.

updateRecord Type	STRING	NS	The Record Type to replace into the searched rows.
updateRecord Value	STRING	ns1.dns.6connect.com.	The Record Value to replace into the searched rows.
updateRecord Description	STRING	Description	Update Record Descriptions.
updateRecord TTL	STRING	3600	The Record TTL to replace into the searched rows.

## add

URL `/api/v1/api.php?target=record&action=add`

Description Adds a new Record to a supplied Zone.

Returns

### Examples:

SUCCESSFUL:	<code>{"success":1,"message":"Add Successful.","data":123}</code>
ERROR:	<code>{"success":0,"message":"error message"}</code>

### Data Detail:

Name	Type	Description
data	INT	The ID of the new Record entry.

Required Parameters

Name	Type	Example	Description
newRecordZoneId	INT	123	The Zone ID of the new Record.
newRecordHost	STRING	@	New Host Name.
newRecordType	STRING	PTR	New Record Type.
newRecordValue	STRING	123	New Record Value.

Optional Parameters	Name	Type	Example	Description
	newRecordDescription	STRING	Description.	Notes for the Record.
	newRecordTTL	INT	foo	Record TTL.

delete					
URL	/api/v1/api.php?target=record&action=delete				
Description	Performs a search over the Zones and Records dataset and deletes all found Records, but leaves their parent Zones intact.				
Returns	<p><b>Examples:</b></p> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td>{"success":1,"message":"Deletion Successful."}</td> </tr> <tr> <td>ERROR:</td> <td>{"success":0,"message":"error message"}</td> </tr> </table>	SUCCESSFUL:	{"success":1,"message":"Deletion Successful."}	ERROR:	{"success":0,"message":"error message"}
SUCCESSFUL:	{"success":1,"message":"Deletion Successful."}				
ERROR:	{"success":0,"message":"error message"}				

Optional Parameters	Name	Type	Example	Description
	deleteZoneId	INT	123	The Zone ID to search for.
	deleteZoneName	STRING	foo	The Zone Name to search for.
	deleteZoneCustomerId	INT	5	The Customer ID to search for.
	deleteZoneIpVersion	STRING	IPv6	The IP Version to search for.
	deleteZoneMask	STRING		The Zone Mask to search for.
	deleteZoneSerial	INT	2012033001	The Zone Serial to search for.
	deleteZoneRefresh	INT	36000	The Zone Refresh to search for.
	deleteZoneRetry	INT	800	The Zone Retry to search for.
	deleteZoneExpire	INT	6090000	The Zone Expire to search for.

deleteZoneMinimum	INT	10	The Zone Minimum to search for.
deleteZoneSOA	STRING	200	The Zone SOA to search for.
deleteZoneTags	STRING	client,production	Zone Tags to search for.
deleteZoneTTL	INT	3600	The Zone TTL to search for.
deleteZoneEnableDNSSEC	INT	1	Search based on DNSSEC settings.
deleteRecordID	INT	123	The Record ID to search for.
deleteRecordHost	STRING	@	The Record Host to search for.
deleteRecordType	STRING	NS	The Record Type to search for.
deleteRecordValue	STRING	ns1.dns.6connect.com.	The Record Value to search for.
deleteRecordDescription	STRING	Description	Search based on Record Description.
deleteRecordTTL	STRING	3600	The Record TTL to search for.

### switch

URL /api/v1/api.php?target=record&action=switch

Description Switches the order of two record entries.

Returns

**Examples:**

SUCCESSFUL:	{"success":1,"message":"Record Moved."}
ERROR:	{"success":0,"message":"error message"}

#### Required Parameters

Name	Type	Example	Description
moveWhichId	INT	123	The Record Id to be moved.
moveAfterId	INT	@	The Id of the Record the first Record is to be moved after.

## Server-Zone Linkage

### get

URL

/api/v1/api.php?target=zoneLinkage&action=get

Description

Searches for Server-Zone Linkages. If no search parameters are supplied, all linkages are returned.

Returns

**Examples:**

SUCCESSFUL:	<pre>{"success":1,"message":"2 rows retrieved.", "data":[{"id":"285", "zoneld":"64", "serverId":"1", "serverName":"173.164.182.169", "serverType":"SCP", "serverMasterType":"master", "zoneName":"bind.com", "resourceId":"483"}, {"id":"287", "zoneld":"371", "serverId":"1", "serverName":"173.164.182.169", "serverType":"SCP", "serverMasterType":"master", "zoneName":"132.235.198.in-addr.arpa", "resourceId":"577"}]}</pre>
ERROR:	<pre>{"success":0, "message":"error message"}</pre>

**Data Detail:**

Name	Type	Description
id	INT	The Linkage Id.
zoneld	INT	The Zoneld involved in this link.
serverId	INT	The ServerId involved in this link.
serverName	STRING	The server name
serverType	STRING	The server transfer type
serverMasterType	STRING	Whether this server is a master or a slave.
zoneName	STRING	The zone name
resourceId	INT	The Resource Id the Zone is attached to.

Optional Parameters

Name	Type	Example	Description
id	INTEGER	15	Fetches the linkage with the matching id.
serverId	INTEGER	15	Fetches all linkages with the matching serverId.
zoneId	INTEGER	15	Fetches all linkages with the matching zoneId.

**add**

URL

/api/v1/api.php?target=zoneLinkage&action=add

Description

Adds a new link between a DNS Server and a Zone

Returns

**Examples:**

SUCCESSFUL:

```
{"success":1,"message":"Link Added."}
```

ERROR:

```
{"success":0, "message":"error message"}
```

Required Parameters

Name	Type	Example	Description
serverId	INT	16	The DNS Server Id.
zoneId	INT	105	The Zone Id.

**delete**

URL

/api/v1/api.php?target=zoneLinkage&action=delete

Description

Deletes a link between a DNS Server and a Zone

Returns

**Examples:**

SUCCESSFUL:

```
{"success":1,"message":"Link Deleted."}
```

ERROR:

```
{"success":0, "message":"error message"}
```

### Optional Parameters

Name	Type	Example	Description
id	INTEGER	15	Fetches the linkage with the matching id.
serverId	INTEGER	15	Fetches all linkages with the matching serverId.
zoneId	INTEGER	15	Fetches all linkages with the matching zoneId.

## Name Server Control

### get

URL

/api/v1/api.php?target=nameServer&action=get

Description

Fetches a list of all stored Name Servers

Returns

**Examples:**

SUCCESSFUL:	<pre>{ "success": 1, "message": "Fetch Successful.", "data": [{"id": "1", "nameserver": "ns1.dns.6connect.net", "add_to_zones_default": "1", "ordering": "10", "uses": "34"}, {"id": "10", "nameserver": "ns2.dns.6connect.net", "add_to_zones_default": "1", "ordering": "11", "uses": "46"}, {"id": "9", "nameserver": "ns4.dns.6connect.net", "add_to_zones_default": "1", "ordering": "14", "uses": "12"}, {"id": "3", "nameserver": "ns3.dns.6connect.net", "add_to_zones_default": "1", "ordering": "15", "uses": "46"}]}</pre>
ERROR:	<pre>{ "success": 0, "message": "error message" }</pre>

**Data Detail:**

Name	Type	Description
id	INT	Server ID
nameserver	STRING	Server Name
add_to_zones_default	BOOL	Whether or not this is a default server.
ordering	INT	Display order
uses	INT	How many zones have been assigned to this NameServer

**add**

URL

/api/v1/api.php?target=nameServer&action=add

Description

Adds a new DNS Server

Returns

**Examples:**

SUCCESSFUL:	<pre>{ "success": 1, "message": "Add Successful." }</pre>
ERROR:	<pre>{ "success": 0, "message": "error message" }</pre>

Required Parameters

Name	Type	Example	Description
newServer	STRING	ns.yourdomain.com	Name of the NameServer

<b>delete</b>									
URL	/api/v1/api.php?target=nameServer&action=delete								
Description	Deletes a NameServer								
Returns	<p><b>Examples:</b></p> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td><i>{"success":1,"message":"Server Deleted."}</i></td> </tr> <tr> <td>ERROR:</td> <td><i>{"success":0, "message":"error message"}</i></td> </tr> </table>	SUCCESSFUL:	<i>{"success":1,"message":"Server Deleted."}</i>	ERROR:	<i>{"success":0, "message":"error message"}</i>				
SUCCESSFUL:	<i>{"success":1,"message":"Server Deleted."}</i>								
ERROR:	<i>{"success":0, "message":"error message"}</i>								
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>NUMERIC</td> <td>5</td> <td>ID of server to delete.</td> </tr> </tbody> </table>	Name	Type	Example	Description	id	NUMERIC	5	ID of server to delete.
Name	Type	Example	Description						
id	NUMERIC	5	ID of server to delete.						
Example URL	/api/v1/api.php?target=nameServer&action=delete&id=5								

<b>setDefault</b>													
URL	/api/v1/api.php?target=nameServer&action=setDefault												
Description	Default NameServers have all new zones added to them as they are created. Multiple NameServers can be classified as Default.												
Returns	<p><b>Examples:</b></p> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td><i>{"success":1, "message":"Success."}</i></td> </tr> <tr> <td>ERROR:</td> <td><i>{"success":0, "message":"error message"}</i></td> </tr> </table>	SUCCESSFUL:	<i>{"success":1, "message":"Success."}</i>	ERROR:	<i>{"success":0, "message":"error message"}</i>								
SUCCESSFUL:	<i>{"success":1, "message":"Success."}</i>												
ERROR:	<i>{"success":0, "message":"error message"}</i>												
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>NUMERIC</td> <td>5</td> <td>ID of server to modify.</td> </tr> <tr> <td>value</td> <td>1 or 0</td> <td>1</td> <td>1 = Default, 0 = Normal</td> </tr> </tbody> </table>	Name	Type	Example	Description	id	NUMERIC	5	ID of server to modify.	value	1 or 0	1	1 = Default, 0 = Normal
Name	Type	Example	Description										
id	NUMERIC	5	ID of server to modify.										
value	1 or 0	1	1 = Default, 0 = Normal										
Example URL	/api/v1/api.php?target=nameServer&action=setDefault&id=3&value=1												

<b>orderUp</b>	
URL	/api/v1/api.php?target=nameServer&action=orderUp
Description	Swaps the index order of the targeted NameServer with that of the one above it.

Returns	<b>Examples:</b> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td><code>{"success":1,"message":"Reordering Successful."}</code></td> </tr> <tr> <td>ERROR:</td> <td><code>{"success":0,"message":"error message"}</code></td> </tr> </table>	SUCCESSFUL:	<code>{"success":1,"message":"Reordering Successful."}</code>	ERROR:	<code>{"success":0,"message":"error message"}</code>
SUCCESSFUL:	<code>{"success":1,"message":"Reordering Successful."}</code>				
ERROR:	<code>{"success":0,"message":"error message"}</code>				
Example URL	/api/v1/api.php?target=nameServer&action=orderUp&id=3				

<b>orderDown</b>									
URL	/api/v1/api.php?target=nameServer&action=orderDown								
Description	Swaps the index order of the targetted NameServer with that of the one below it.								
Returns	<b>Examples:</b> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td><code>{"success":1,"message":"Reordering Successful."}</code></td> </tr> <tr> <td>ERROR:</td> <td><code>{"success":0,"message":"error message"}</code></td> </tr> </table>	SUCCESSFUL:	<code>{"success":1,"message":"Reordering Successful."}</code>	ERROR:	<code>{"success":0,"message":"error message"}</code>				
SUCCESSFUL:	<code>{"success":1,"message":"Reordering Successful."}</code>								
ERROR:	<code>{"success":0,"message":"error message"}</code>								
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>NUMERIC</td> <td>5</td> <td>ID of server to activate.</td> </tr> </tbody> </table>	Name	Type	Example	Description	id	NUMERIC	5	ID of server to activate.
Name	Type	Example	Description						
id	NUMERIC	5	ID of server to activate.						
Example URL	/api/v1/api.php?target=nameServer&action=orderDown&id=5								

## API Module - IPAM

### IP Address Management (IPv4 and IPv6)

- Get
- Add
- Update
- Add Tag
- Delete Tag
- Smart Assign
- Direct Assign
- Unassign
- getTagList
- getRIRList
- getRegionList
- utilization
- aggregate
- split

<b>Get</b>	
URL	/api/v1/api.php?target=ipam&action=get

**Description**

Returns a list of IP blocks. Use optional parameters to filter the list. If multiple parameters are specified, only blocks matching all parameters will be returned.

**Returns**

**Examples:**

SUCCESSFUL	<pre>{ "success": 1, "message": "1 blocks found. ", "data": [ { "id": 5890, "type": "ipv4", "top_aggregate": null, "cidr": "192.168.0.0V24", "formatted_ip": "192.168.0.0V24", "address": "3232235520", "end_address": "3232235775", "mask": 24, "child1": null, "child2": null, "is_assigned": 0, "is_swipped": 0, "is_aggregate": 1, "custid": 81, "resource_id": 81, "resource_name": "Available", "last_updated_time": null, "description": null, "parent": null, "rir": "1918", "lir_id": null, "notes": null, "generic_code": null, "code": null, "region": "SFO", "vlan": 100, "arin_net_id": null, "arin_cust_id": null, "org_id": null, "arin_swip_time": null, "assigned_time": null, "asn": null, "allowSubAssignments": false, "permissions": { "permissionIPAMRead": "1", "permissionIPAMUpdate": "1", "permissionIPAMCreate": "1", "permissionSWIP": "1", "permissionAdmin": "1" }, "range": "192.168.0.0 - 192.168.0.255", "tags": [ "Customer", "PTP" ] } } }</pre>
ERROR	<pre>{'success':0, 'message':'error message'}</pre>

**Optional Parameters**

Name	Type	Example	Description
asn	INTEGER	1000	Filters blocks based on their ASN

allowSubAssignments	BOOL	true	Filters blocks based on whether they allow sub-assignments or not. Acceptable values: "true" or "false"
block	STRING	213.37.29.0/24	CIDR block description
code	STRING		User-defined block code as defined in Admin-IPAM settings: Generic Code Per Block Name
id	INTEGER	1234	The ID of the block
isAggregate	BOOL	true	Indicates if the block has been split into children or not. A value of 'true' will return blocks with no children.
isAssigned	BOOL	true	Acceptable values: "true" or "false"
isSwipped	BOOL	true	Acceptable values: "true" or "false"
mask	INTEGER	24	Integer bitmask
region	STRING	SFO	The value from the list of name/value pairs which make up the list of available regions

resourceHolderId	STRING	6c-1234	The 20 character resource holder ID. This is an internal short identifier for the customer, and should be used to link resource holder details in the 6Connect database back to your organization. Consider using internal customer numbers or department numbers for this field depending on your use case. Example: Customer-001 or 000213.
resourceId	INTEGER		The ID of the resource the block is assigned to
rir	STRING	ARIN	Acceptable values: ARIN, RIPE, APNIC, AfrinIC
search	STRING	192.168	If a search term is provided, all IPAM fields including assigned Resource Holder name will be checked with a LIKE comparison to find matching blocks
selectCount	INTEGER	50	# of blocks to get

selectOffset	INTEGER	25	Offset for results set; useful for paging (e.g. selectCount = 50, selectOffset = 100 would return the 3rd page of 50 results)
sortField	STRING	cidr	Attribute to sort blocks by. Acceptable values: cidr, mask, rir, vlan, code, updateTime
sortOrder	INTEGER	ASC	ASC or DESC
tags	STRING	customer,vpn	Comma-separated list of tags
topAggregateId	INTEGER	1234	The ID of the aggregate block to which the block belongs
vlan	INTEGER	123	VLAN for the block

Example URL

/api/v1/api.php?target=ipam&action=get&rir=ARIN&tags=customer,vpn

## Add

URL

/api/v1/api.php?target=ipam&action=add

Description

Adds an IPv4 or IPv6 block

Returns

**Examples:**  
 SUCCESSFUL: `{'success':1, 'id':'12345'}`  
 ERROR: `{'success':0, 'message':'unable to add block'}`>

Required Parameters

Name	Type	Example	Description
block	STRING	213.37.29.0/24	CIDR block description
rir	STRING	ARIN	Acceptable values: ARIN, RIPE, APNIC, AfriNIC

## Optional Parameters

Name	Type	Example	Description
allowSubAssignments	BOOL	true	Does the block allow sub-assignments? If the block is assigned and allowSubAssignments is "true", children split from this block will be able to be assigned to different resources. Acceptable values: "true" or "false"
asn	INTEGER	1000	ASN for the block
code	STRING		User-defined block code as defined in Admin-IPAM settings: Generic Code Per Block Name
notes	STRING		Notes or description for the block
region	STRING	SFO	The value from the list of name/value pairs which make up the list of available regions
tags	STRING	customer,vpn	Comma-separated list of tags
vlan	INTEGER	123	VLAN for the block

Example URL

</api/v1/api.php?target=ipam&action=add&rir=ARIN>

Update

URL	/api/v1/api.php?target=ipam&action=update&type=IP&ipid=13420&rir=RIPE											
Description	Updates detail data about an IP block.											
Returns	<b>Examples:</b> <table border="1" data-bbox="824 321 1479 485"> <tr> <td>SUCCESSFUL</td> <td colspan="3"><code>{"success":1,"message":"Update Successful."}</code></td> </tr> <tr> <td>ERROR</td> <td colspan="3"><code>{'success':0, 'message':'error message'}</code></td> </tr> </table>				SUCCESSFUL	<code>{"success":1,"message":"Update Successful."}</code>			ERROR	<code>{'success':0, 'message':'error message'}</code>		
SUCCESSFUL	<code>{"success":1,"message":"Update Successful."}</code>											
ERROR	<code>{'success':0, 'message':'error message'}</code>											
Required Parameters	<table border="1" data-bbox="824 548 1479 680"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>int</td> <td>125</td> <td>ID of the IP block</td> </tr> </tbody> </table>				Name	Type	Example	Description	id	int	125	ID of the IP block
Name	Type	Example	Description									
id	int	125	ID of the IP block									

Optional Parameters

Name	Type	Example	Description
allowSubAssignments	BOOL	true	Does the block allow sub-assignments? If the block is assigned and allowSubAssignments is "true", children split from this block will be able to be assigned to different resources. Acceptable values: "true" or "false"
asn	INTEGER	1000	ASN for the block
code	STRING		Arbitrary user-defined block code
lirId	INTEGER		The numeric ID of an LIR resource the block should be linked to
notes	STRING	Words	Misc. Notes
region	STRING	Chicago, IL	The region this IP block is assigned to.
rir	STRING	ARIN	The RIR
vlan	NUMERIC	50000	

**Add Tag**

URL	/api/v1/api.php?target=ipam&action=addTag&id=13420&tag=Infra				
Description	Adds a tag to an IP block.				
Returns	<p><b>Examples:</b></p> <table border="1"> <tbody> <tr> <td>SUCCESSFUL</td> <td><i>{"success":1,"message":"Tag Added."}</i></td> </tr> <tr> <td>ERROR</td> <td><i>{'success':0, 'message':'error message'}</i></td> </tr> </tbody> </table>	SUCCESSFUL	<i>{"success":1,"message":"Tag Added."}</i>	ERROR	<i>{'success':0, 'message':'error message'}</i>
SUCCESSFUL	<i>{"success":1,"message":"Tag Added."}</i>				
ERROR	<i>{'success':0, 'message':'error message'}</i>				

Required Parameters	Name	Type	Example	Description
	id	int	125	ID of the block
	tag	STRING	Customer	The tag to add

Delete Tag													
URL	/api/v1/api.php?target=ipam&action=deleteTag&id=13420&tag=Prod												
Description	Removes a tag from an IP block.												
Returns	<p><b>Examples:</b></p> <table border="1"> <tr> <td>SUCCESSFUL</td> <td><i>{"success":1,"message":"Tag Removed."}</i></td> </tr> <tr> <td>ERROR</td> <td><i>{'success':0, 'message':'error message'}</i></td> </tr> </table>	SUCCESSFUL	<i>{"success":1,"message":"Tag Removed."}</i>	ERROR	<i>{'success':0, 'message':'error message'}</i>								
SUCCESSFUL	<i>{"success":1,"message":"Tag Removed."}</i>												
ERROR	<i>{'success':0, 'message':'error message'}</i>												
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>int</td> <td>125</td> <td>ID of the block</td> </tr> <tr> <td>tag</td> <td>STRING</td> <td>Customer</td> <td>The tag to delete</td> </tr> </tbody> </table>	Name	Type	Example	Description	id	int	125	ID of the block	tag	STRING	Customer	The tag to delete
Name	Type	Example	Description										
id	int	125	ID of the block										
tag	STRING	Customer	The tag to delete										

Smart Assign	
URL	/api/v1/api.php?target=ipam&action=smartAssign
Description	Selects a block based on supplied parameters (rir, tags, etc.) and assigns to an Resource Holder.
Returns	<p><b>Examples:</b></p> <p>SUCCESSFUL: <i>{'success':1, 'id':'12345'}</i></p> <p>ERROR: <i>{'success':0, 'message':'Unable to assign block'}</i></p>

Required Parameters

24

Name	Type	Example	Description
code	STRING	Code X	Arbitrary user-defined block code
mask	INTEGER		The size of the block to be assigned
rir	STRING	ARIN	Acceptable values: ARIN, RIPE, APNIC, AfriNIC
resourceHolderId*	STRING		The 20 character resource holder ID. This is an internal short identifier for the customer, and should be used to link resource holder details in the 6Connect database back to your organization. Consider using internal customer numbers or department numbers for this field depending on your use case. Example: Customer-001 or 000213.
resourceId*	INTEGER		Integer ID of the resource to assign the block to
type	STRING	"IPv4" or "IPv6"	The type of block to assign

\*Either resourceHolderId or resourceId can be used, but only one must be provided

Optional Parameters	<b>Name</b>	<b>Type</b>	<b>Example</b>	<b>Description</b>
	tags	STRING	customer,vpn	Comma separated string of tags
	region	STRING	Ashburn	Region to assign from
Example URL	/api/v1/api.php?target=ipam&action=smartAssign&mask=24&resourceHolderId=SJS-0031&rir=ARIN&tags=customer,vpn			

Direct Assign	
URL	/api/v1/api.php?target=ipam&action=directAssign
Description	Assigns a block to an Resource Holder
Returns	<b>Examples:</b> SUCCESSFUL: <code>{'success':1, 'message':'213.37.29.0/24 assigned to SJS-0031', 'id':'12345'}</code> ERROR: <code>{'success':0, 'message':'Unable to assign block'}</code>

Required Parameters

Name	Type	Example	Description
block*	STRING	213.37.29.0/24	CIDR block description
id*	int	125	ID of the IP block
*Either block or id can be used, but only one must be provided			
resourceHolderId*	STRING		The 20 character resource holder ID. This is an internal short identifier for the customer, and should be used to link resource holder details in the 6Connect database back to your organization. Consider using internal customer numbers or department numbers for this field depending on your use case. Example: Customer-001 or 000213.
resourceId*	INTEGER		Integer ID of the resource to assign the block to
*Either resourceHolderId or resourceId can be used, but only one must be provided			

Example URL

/api/v1/api.php?target=ipam&action=directAssign&block=213.37.29.0/24&resourceHolderId=SJS-0031

**Unassign**

URL

/api/v1/api.php?target=ipam&action=unassign

Description

Reclaims the specified block to be reassigned in the future

Returns	<b>Examples:</b> SUCCESSFUL: <code>{'success':1, 'message':'213.37.29.0/24 unassigned', 'id':'12345'}</code> ERROR: <code>{'success':0, 'message':'213.37.29.0/24 was not found'}</code>												
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>block*</td> <td>STRING</td> <td>213.37.29.0/24</td> <td>CIDR block description</td> </tr> <tr> <td>id*</td> <td>int</td> <td>125</td> <td>ID of the IP block</td> </tr> </tbody> </table> <p>*Either block or id can be used, but only one must be provided</p>	Name	Type	Example	Description	block*	STRING	213.37.29.0/24	CIDR block description	id*	int	125	ID of the IP block
Name	Type	Example	Description										
block*	STRING	213.37.29.0/24	CIDR block description										
id*	int	125	ID of the IP block										
Example URL	/api/v1/api.php?target=ipam&action=unassign&block=213.37.29.0/24												

getTagList					
URL	/api/v1/api.php?target=ipam&action=getTagList				
Description	Returns a list of all valid IP Tags in the database.				
Returns	<b>Examples:</b> <table border="1"> <tr> <td>SUCCESSFUL</td> <td><code>{"success":1,"message":"Tags Retrieved.","data":[{"value":"IT","name":"IT"}, {"value":"LTE","name":"LTE Mobile"}, {"value":"PTP","name":"Point to Point"}, {"value":"Prod","name":"Production"}, {"value":"VM","name":"Virtual Machines"}, {"value":"VOIP","name":"VOIP"}, {"value":"ANY","name":"ANY"}]}</code></td> </tr> <tr> <td>ERROR</td> <td><code>{'success':0, 'message':'error message'}</code></td> </tr> </table>	SUCCESSFUL	<code>{"success":1,"message":"Tags Retrieved.","data":[{"value":"IT","name":"IT"}, {"value":"LTE","name":"LTE Mobile"}, {"value":"PTP","name":"Point to Point"}, {"value":"Prod","name":"Production"}, {"value":"VM","name":"Virtual Machines"}, {"value":"VOIP","name":"VOIP"}, {"value":"ANY","name":"ANY"}]}</code>	ERROR	<code>{'success':0, 'message':'error message'}</code>
SUCCESSFUL	<code>{"success":1,"message":"Tags Retrieved.","data":[{"value":"IT","name":"IT"}, {"value":"LTE","name":"LTE Mobile"}, {"value":"PTP","name":"Point to Point"}, {"value":"Prod","name":"Production"}, {"value":"VM","name":"Virtual Machines"}, {"value":"VOIP","name":"VOIP"}, {"value":"ANY","name":"ANY"}]}</code>				
ERROR	<code>{'success':0, 'message':'error message'}</code>				

getRIRList	
URL	/api/v1/api.php?target=ipam&action=getRIRList
Description	Returns a list of all valid RIRs in the database.

Returns	<p><b>Examples:</b></p> <table border="1"> <tr> <td>SUCCESSFUL</td> <td><pre>{ "success": 1, "message": "RIRs Retrieved.", "data": [{"value": "ARIN", "name": "ARIN"}, {"value": "1918", "name": "1918"}, {"value": "AfriNIC", "name": "AfriNIC"}, {"value": "APNIC", "name": "APNIC"}, {"value": "LACNIC", "name": "LACNIC"}, {"value": "RIPE", "name": "RIPE"}, {"value": "1918-SJC", "name": "1918-SJC"}]}</pre></td> </tr> <tr> <td>ERROR</td> <td><pre>{ 'success': 0, 'message': 'error message' }</pre></td> </tr> </table>	SUCCESSFUL	<pre>{ "success": 1, "message": "RIRs Retrieved.", "data": [{"value": "ARIN", "name": "ARIN"}, {"value": "1918", "name": "1918"}, {"value": "AfriNIC", "name": "AfriNIC"}, {"value": "APNIC", "name": "APNIC"}, {"value": "LACNIC", "name": "LACNIC"}, {"value": "RIPE", "name": "RIPE"}, {"value": "1918-SJC", "name": "1918-SJC"}]}</pre>	ERROR	<pre>{ 'success': 0, 'message': 'error message' }</pre>
SUCCESSFUL	<pre>{ "success": 1, "message": "RIRs Retrieved.", "data": [{"value": "ARIN", "name": "ARIN"}, {"value": "1918", "name": "1918"}, {"value": "AfriNIC", "name": "AfriNIC"}, {"value": "APNIC", "name": "APNIC"}, {"value": "LACNIC", "name": "LACNIC"}, {"value": "RIPE", "name": "RIPE"}, {"value": "1918-SJC", "name": "1918-SJC"}]}</pre>				
ERROR	<pre>{ 'success': 0, 'message': 'error message' }</pre>				

getRegionList					
URL	/api/v1/api.php?target=ipam&action=getRegionList				
Description	Returns a list of all valid Regions in the database.				
Returns	<p><b>Examples:</b></p> <table border="1"> <tr> <td>SUCCESSFUL</td> <td><pre>{ "success": 1, "message": "Regions Retrieved.", "data": [{"value": "ANY", "name": "Any Region"}, {"value": "ASH1", "name": "Ashburn, VA"}, {"value": "BOS", "name": "Boston, MA"}, {"value": "CHI", "name": "Chicago, IL"}, {"value": "DAL", "name": "Dallas, TX"}, {"value": "DEN", "name": "Denver, CO"}, {"value": "FRKT", "name": "Frankfurt, DE"}, {"value": "LON1", "name": "London, UK"}, {"value": "MIA", "name": "Miami, FL"}, {"value": "PAR", "name": "Paris, FR"}, {"value": "SFO", "name": "San Francisco, CA"}, {"value": "SEA", "name": "Seattle, WA"}, {"value": "Tokyo", "name": "Tokyo"}, {"value": "Singapore", "name": "Singapore"}, {"value": "Jakarta", "name": "Jakarta"}]}</pre></td> </tr> <tr> <td>ERROR</td> <td><pre>{ 'success': 0, 'message': 'error message' }</pre></td> </tr> </table>	SUCCESSFUL	<pre>{ "success": 1, "message": "Regions Retrieved.", "data": [{"value": "ANY", "name": "Any Region"}, {"value": "ASH1", "name": "Ashburn, VA"}, {"value": "BOS", "name": "Boston, MA"}, {"value": "CHI", "name": "Chicago, IL"}, {"value": "DAL", "name": "Dallas, TX"}, {"value": "DEN", "name": "Denver, CO"}, {"value": "FRKT", "name": "Frankfurt, DE"}, {"value": "LON1", "name": "London, UK"}, {"value": "MIA", "name": "Miami, FL"}, {"value": "PAR", "name": "Paris, FR"}, {"value": "SFO", "name": "San Francisco, CA"}, {"value": "SEA", "name": "Seattle, WA"}, {"value": "Tokyo", "name": "Tokyo"}, {"value": "Singapore", "name": "Singapore"}, {"value": "Jakarta", "name": "Jakarta"}]}</pre>	ERROR	<pre>{ 'success': 0, 'message': 'error message' }</pre>
SUCCESSFUL	<pre>{ "success": 1, "message": "Regions Retrieved.", "data": [{"value": "ANY", "name": "Any Region"}, {"value": "ASH1", "name": "Ashburn, VA"}, {"value": "BOS", "name": "Boston, MA"}, {"value": "CHI", "name": "Chicago, IL"}, {"value": "DAL", "name": "Dallas, TX"}, {"value": "DEN", "name": "Denver, CO"}, {"value": "FRKT", "name": "Frankfurt, DE"}, {"value": "LON1", "name": "London, UK"}, {"value": "MIA", "name": "Miami, FL"}, {"value": "PAR", "name": "Paris, FR"}, {"value": "SFO", "name": "San Francisco, CA"}, {"value": "SEA", "name": "Seattle, WA"}, {"value": "Tokyo", "name": "Tokyo"}, {"value": "Singapore", "name": "Singapore"}, {"value": "Jakarta", "name": "Jakarta"}]}</pre>				
ERROR	<pre>{ 'success': 0, 'message': 'error message' }</pre>				

utilization													
URL	/api/v1/api.php?target=ipam&action=utilization												
Description	Gets the utilization percentages for a specific ip block or ip block and mask combination.												
Example URLs	<a href="https://ops.6connect.com/qa-4.1/api/v1/api.php?target=ipam&amp;action=aggregate&amp;id=23131">https://ops.6connect.com/qa-4.1/api/v1/api.php?target=ipam&amp;action=aggregate&amp;id=23131</a>												
Returns	<p><b>Examples:</b></p> <table border="1"> <tbody> <tr> <td>SUCCESSFUL</td> <td><pre>{ "success":1, "message": "10.2.0.128\25 aggregated into 10.2.0.0\24", "id":16326 }</pre></td> </tr> <tr> <td>ERROR</td> <td><pre>{'success':0, 'message':'error message'}</pre></td> </tr> </tbody> </table>	SUCCESSFUL	<pre>{ "success":1, "message": "10.2.0.128\25 aggregated into 10.2.0.0\24", "id":16326 }</pre>	ERROR	<pre>{'success':0, 'message':'error message'}</pre>								
SUCCESSFUL	<pre>{ "success":1, "message": "10.2.0.128\25 aggregated into 10.2.0.0\24", "id":16326 }</pre>												
ERROR	<pre>{'success':0, 'message':'error message'}</pre>												
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>int</td> <td>125</td> <td>ID of the IP block.</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Example	Description	id	int	125	ID of the IP block.				
Name	Type	Example	Description										
id	int	125	ID of the IP block.										
Optional Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>mask</td> <td>int</td> <td>24</td> <td>The specific mask size to retrieve utilization for. If using this parameter, the id parameter should be the id of the aggregate.</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Example	Description	mask	int	24	The specific mask size to retrieve utilization for. If using this parameter, the id parameter should be the id of the aggregate.				
Name	Type	Example	Description										
mask	int	24	The specific mask size to retrieve utilization for. If using this parameter, the id parameter should be the id of the aggregate.										

aggregate	
URL	/api/v1/api.php?target=ipam&action=aggregate
Description	Aggregates a selected block to the mask specified. If no mask specified, re-aggregates blocks to next parent. IE. calling aggregate on a /25 will aggregate both children back to the parent /24. All child blocks must be Available for aggregation to succeed.

Returns	<b>Examples:</b> <table border="1"> <tr> <td>SUCCESSFUL</td> <td><pre>{ "success":1, "message": "10.2.0.128\25 aggregated into 10.2.0.0\24", "id":16326 }</pre></td> </tr> <tr> <td>ERROR</td> <td><pre>{'success':0, 'message':'error message'}</pre></td> </tr> </table>	SUCCESSFUL	<pre>{ "success":1, "message": "10.2.0.128\25 aggregated into 10.2.0.0\24", "id":16326 }</pre>	ERROR	<pre>{'success':0, 'message':'error message'}</pre>
SUCCESSFUL	<pre>{ "success":1, "message": "10.2.0.128\25 aggregated into 10.2.0.0\24", "id":16326 }</pre>				
ERROR	<pre>{'success':0, 'message':'error message'}</pre>				

Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>int</td> <td>125</td> <td>ID of the IP block.</td> </tr> </tbody> </table>	Name	Type	Example	Description	id	int	125	ID of the IP block.
Name	Type	Example	Description						
id	int	125	ID of the IP block.						

Optional Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>mask</td> <td>int</td> <td>24</td> <td>Auto aggregate the block back to this mask size. Note all blocks up this mask size must be Available or call will fail.</td> </tr> <tr> <td>block*</td> <td>STRING</td> <td>213.37.29.0/24</td> <td>CIDR block. Only the block id OR the block CIDR can be specified. Not both.</td> </tr> </tbody> </table>	Name	Type	Example	Description	mask	int	24	Auto aggregate the block back to this mask size. Note all blocks up this mask size must be Available or call will fail.	block*	STRING	213.37.29.0/24	CIDR block. Only the block id OR the block CIDR can be specified. Not both.
Name	Type	Example	Description										
mask	int	24	Auto aggregate the block back to this mask size. Note all blocks up this mask size must be Available or call will fail.										
block*	STRING	213.37.29.0/24	CIDR block. Only the block id OR the block CIDR can be specified. Not both.										

split	
URL	/api/v1/api.php?target=ipam&action=split
Description	Splits a selected block to the mask specified. If no mask specified, it split blocks to next child. IE. calling aggregate on a /24 will split both parent to the child /25s. All parent blocks must be Available, or have Allow Sub Assignments on for a split to succeed.

Returns	<p><b>Examples:</b></p> <table border="1"> <tr> <td>SUCCESSFUL</td> <td> <pre>{ "success":1, "message": "10.1.0.0\24 split into 10.1.0.0\25 and 10.1.0.128\25", "data": {"child1":23441 , "child2":23451}}</pre> </td> </tr> <tr> <td>ERROR</td> <td> <pre>{'success':0, 'message':'error message'}</pre> </td> </tr> </table>	SUCCESSFUL	<pre>{ "success":1, "message": "10.1.0.0\24 split into 10.1.0.0\25 and 10.1.0.128\25", "data": {"child1":23441 , "child2":23451}}</pre>	ERROR	<pre>{'success':0, 'message':'error message'}</pre>												
SUCCESSFUL	<pre>{ "success":1, "message": "10.1.0.0\24 split into 10.1.0.0\25 and 10.1.0.128\25", "data": {"child1":23441 , "child2":23451}}</pre>																
ERROR	<pre>{'success':0, 'message':'error message'}</pre>																
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>int</td> <td>125</td> <td>ID of the IP block.</td> </tr> </tbody> </table>	Name	Type	Example	Description	id	int	125	ID of the IP block.								
Name	Type	Example	Description														
id	int	125	ID of the IP block.														
Optional Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>mask</td> <td>int</td> <td>24</td> <td>Auto aggregate the block back to this mask size. Note all blocks up this mask size must be Available or call will fail.</td> </tr> <tr> <td>block*</td> <td>STRING</td> <td>213.37.29.0/24</td> <td>CIDR block. Only the block id OR the block CIDR can be specified. Not both.</td> </tr> <tr> <td>autoSplitLimit</td> <td>int</td> <td>4</td> <td>A number the power of 2 (^2).</td> </tr> </tbody> </table>	Name	Type	Example	Description	mask	int	24	Auto aggregate the block back to this mask size. Note all blocks up this mask size must be Available or call will fail.	block*	STRING	213.37.29.0/24	CIDR block. Only the block id OR the block CIDR can be specified. Not both.	autoSplitLimit	int	4	A number the power of 2 (^2).
Name	Type	Example	Description														
mask	int	24	Auto aggregate the block back to this mask size. Note all blocks up this mask size must be Available or call will fail.														
block*	STRING	213.37.29.0/24	CIDR block. Only the block id OR the block CIDR can be specified. Not both.														
autoSplitLimit	int	4	A number the power of 2 (^2).														

## API Module - LIR

### LIR Management

- [Get](#)
- [Delete](#)

<b>Get</b>	
URL	/api/v1/api.php?target=lir&action=get

Description	Returns a list of LIRs
Returns	<p><b>Examples:</b></p> <pre> SUCCESSFUL {   "success": 1,   "message": "2 objects found",   "data": [     {       "id": "100",       "name": "RIPE Test LIR",       "slug": "ripe-test-lir",       "entities": [         {           "mnt_by": "mntner@email.com"            "mnt_by_password": "password",            "admin_c": "test-admin-c",            "tech_c": "test-tech-c",            "api_key": null         }       ],       "rir": "RIPE"     },     {       "id": "101",       "name": "ARIN Test LIR",       "slug": "arin-test-lir",       "entities": [         {           "org_handle": "TEST-10",            "admin_poc": "TEST-ARIN",            "net_poc": "TEST-ARIN",            "abuse_poc": "",            "net_name_prefix": </pre>

```
"PRFX",  
  
"api_key":  
"API-XXXX-YYYY-ZZZZ-123  
4"  
    },  
    "rir":  
"ARIN",  
    "asn":  
"1000"  
    }
```

	<pre> ] } </pre>
ERROR	<pre> {   "success": 0,   "message": "error message" } </pre>
Example URL	/api/v1/api.php?target=lir&action=get

Delete					
URL	/api/v1/api.php?target=lir&action=delete&id=<ID>				
Description	Deletes and LIR				
Returns	<p><b>Examples:</b></p> <table border="1"> <tr> <td>SUCCESSFUL</td> <td> <pre> {   "success": 1,   "message": "LIR deleted." } </pre> </td> </tr> <tr> <td>ERROR</td> <td> <pre> {   "success": 0,   "message": "error message" } </pre> </td> </tr> </table>	SUCCESSFUL	<pre> {   "success": 1,   "message": "LIR deleted." } </pre>	ERROR	<pre> {   "success": 0,   "message": "error message" } </pre>
SUCCESSFUL	<pre> {   "success": 1,   "message": "LIR deleted." } </pre>				
ERROR	<pre> {   "success": 0,   "message": "error message" } </pre>				
Example URL	/api/v1/api.php?target=lir&action=delete&id=100				

## API Module - Resource

### Resources

- [get](#)
- [add](#)
- [update](#)
- [delete](#)

get	
URL	/api/v1/api.php?target=resource&action=get
Description	Get a resource or resources

Returns

**Examples:**

```
SUCCESSFUL: {"success":1,"message":"Search successful","data":[{"id":"57","name":"2nd Email","slug":"6c-contact-email2","type":"field","parent_id":"1","category_id":null,"attr":[]}]}
```

```
ERROR: {"success":0,"message":"Search failed"}
```

Optional Parameters

Name	Type	Notes/Example
name	STRING	Name of the resource. Example: 6Connect, Inc.
slug	STRING	The unique URL friendly name of the resource. Example: 6connect-inc
type	STRING	Type of resource (eg. <i>entry</i> , <i>field</i> , <i>category</i> )

At most, one of the following:

Name	Type	Notes/Example
id	INTEGER	Get the resource which has this ID
resource_in	ARRAY	Get any resource which has any of these IDs
resource_not_in	ARRAY	Get all the resources which do not have any of these IDs

At most, one of the following:

Name	Type	Notes/Example
parent_id	INTEGER	Get the resources whose parent has this ID
parent__in	ARRAY	Get any resource whose parents have any of these IDs
parent__not_in	ARRAY	Get all resources whose parents do not have any of these IDs

At most, one of the following:

Name	Type	Notes/Example
category_id	INTEGER	Get the resources of the category that has this ID
category__in	ARRAY	Get the resources of the categories that have any of these IDs
category__not_in	ARRAY	Get the resources of all the categories that do not have any of these IDs

You can set the order of the results by setting the STRING value of the parameter `orderby` to one of the following :

- none
- id
- name (*default*)
- slug
- type
- parent\_id
- date
- resource\_\_in (*preserve order given in the resource\_\_in array*)

You can set the direction of the ordering of the results by setting the STRING value of the parameter **order** to one of the following :

- ASC (*default*)
- DESC

You can further limit the results based on attributes the resources may have:

Name	Type	Notes/Example
attr_key	STRING	The name of the attribute. Example: network-fqdn
attr_value	STRING	The value of any attribute, or if attr_key is specified, the value of the attribute defined in attr_key.

attr_compare	STRING	<p>If both attr_key and attr_value are given, the results are by default compared based on the value given as attr_value being equal to the value stored in the database. You can optionally change this by setting the STRING value of attr_compare to one of the following:</p> <ul style="list-style-type: none"> <li>• = (default)</li> <li>• !=</li> <li>• &gt;</li> <li>• &gt;=</li> <li>• &lt;</li> <li>• &lt;=</li> <li>• LIKE</li> <li>• NOT LIKE</li> <li>• IN</li> <li>• NOT IN</li> <li>• BETWEEN</li> <li>• NOT BETWEEN</li> <li>• EXISTS</li> </ul>
--------------	--------	---

You can restrict the range of the resources returned.

Name	Type	Notes/Example
resources_per_page	INTEGER	How many resources to return.
offset	INTEGER	How many resources to offset (the initial resource is 0, not 1).
paged	INTEGER	The page to return (starts at 1, not 0). This parameter is provided for convenience and is used to calculate the offset where: offset=(paged-1)*resources_per_page

Example URL

/api/v1/api.php?target=resource&action=get&id=7

## add

URL

/api/v1/api.php?target=resource&action=add

Description

Add a resource.

Returns

### Examples:

*/api/v1/api.php?target=resource&action=add&meta[name]=apitest&meta[type]=entry&meta[section]=firewall&fields[network-fqdn]=www.example.com*

SUCCESSFUL: {"success":1,"message":"Resource added","data":{"id":1077,"name":"apitest","slug":"apitest","type":"entry","parent\_id":1,"category\_id":"NULL","attr":{"\_section":"70","network-fqdn":"www.example.com"},"section":{"id":"70","name":"Firewall","slug":"firewall","type":"section","parent\_id":"1","category\_id":null,"attr":{}}}

*/api/v1/api.php?target=resource&action=add&meta[name]=apitest&meta[type]=entry&fields[network-fqdn]=www.example.com*

ERROR:{"success":0,"message":"Entries must be assigned to a section"}

Required Parameters

Name	Type	Notes/Example
meta[name]	STRING	Name of the resource
meta[type]	STRING	Type of resource (entry, section, field, ect)

Optional Parameters

Name	Type	Notes/Example
meta[parent_id]	INTEGER	ID of the parent resource
meta[category_id]	INTEGER	ID of the category

Required Parameters

(meta[type] = entry)

One of the following:

Name	Type	Notes/Example
meta[section_id]	INTEGER	ID of the section that the entry will be assigned to
meta[section]	STRING	Slug of the section that the entry will be assigned to

Optional Parameters

(meta[type] = entry)

Name	Type	Notes/Example
------	------	---------------

fields[]	ARRAY	<p>Entry field values (for fields that have already been assigned to the section) can be populated when the entry is created.</p> <p>The format is field[field-slug][field-instance]. If the field instance is left blank, it will simply be the next value in the instance array. For example:</p> <p><i>fields[network-fqdn][]=example.com&amp;fields[network-fqdn][]=test.com</i></p> <p>would be written in JSON as</p> <pre>var fields = {   "network-fqdn"   : [     "example.com",     "test.com"   ] }</pre> <p>A field can be added to a section multiple times. The field instance is used to keep track of which field occurrence we are referring. In this example, the network-fqdn field had been added twice to the section so we were able to store two values for it.</p>
----------	-------	--

meta[custom_id]	STRING	A custom ID for the entry. In the past this has been called the Resource Holder ID or Customer ID. Most recently it was implemented as a text field with the slug "6c-resourceholder-id." Now it is a fundamental part the entry type resources.
-----------------	--------	--

Required Parameters

(meta[type] = field)

Name	Type	Notes/Example
meta[field_type]	STRING	Type of field <ul style="list-style-type: none"> <li>• text</li> <li>• textarea</li> <li>• radios</li> <li>• checkboxes</li> <li>• choicebox</li> </ul>

Optional Parameters

(meta[type] = field)

Name	Type	Notes/Example
meta[help_block]	STRING	Fields can have a line of text under them with instructions
meta[options]	ARRAY	Fields of type radios, checkboxes, or choicebox can have multiple options. This could be multiple radio buttons or a choicebox (dropdown) with several options. For example:  meta[type]=field&meta[name]=Colors&meta[field_type]=choicebox&meta[options][]=Blue&meta[options][]=Green  Will create a choicebox with dropdown options of Blue and Green.

update										
URL	/api/v1/api.php?target=resource&action=update									
Description	Update a resource.									
Returns	<p><b>Examples:</b></p> <p>SUCCESSFUL: <code>{"success":1,"message":"Resource Updated","data":{"id":"1055","name":"87-child-1","slug":"87-child-1","type":"entry","parent_id":"87","category_id":"65","attr":{"_section":"70"},"section":{"id":"70","name":"Firewall","slug":"firewall","type":"section","parent_id":"1","category_id":null,"attr":{}}}}</code></p> <p>ERROR: <code>{"success":0,"message":"No resource found with ID: 1079"}</code></p>									
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Notes/Example</th> </tr> </thead> <tbody> <tr> <td>meta[id]</td> <td>INTEGER</td> <td>ID of resource</td> </tr> <tr> <td>meta[type]</td> <td>STRING</td> <td>Type of resource (entry, section, field, ect)</td> </tr> </tbody> </table>	Name	Type	Notes/Example	meta[id]	INTEGER	ID of resource	meta[type]	STRING	Type of resource (entry, section, field, ect)
Name	Type	Notes/Example								
meta[id]	INTEGER	ID of resource								
meta[type]	STRING	Type of resource (entry, section, field, ect)								
Optional Parameters (meta[type] = entry)	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Notes/Example</th> </tr> </thead> <tbody> <tr> <td>fields[]</td> <td>ARRAY</td> <td>See "add" documentation</td> </tr> </tbody> </table>	Name	Type	Notes/Example	fields[]	ARRAY	See "add" documentation			
Name	Type	Notes/Example								
fields[]	ARRAY	See "add" documentation								

## Optional Parameters

(meta[type] = section)

Name	Type	Notes/Example
fields[]	ARRAY	<p>The fields value should be all the fields that are assigned to the section. Giving an empty array as the fields value will remove all fields from the section.</p> <p>The format is:</p> <pre>fields[position][key]</pre> <p>The position value is the position that the field will appear in (0 is first). The position value must always be included. An example field format for an existing field could be:</p> <pre>fields[0][id]=2 fields[0][slug]=asset-serial-number fields[0][help_block]=something fields[0][new]=false</pre> <ul style="list-style-type: none"><li>▪ Either the id or the slug is required, not both.</li><li>▪ When the "new" parameter is not included, FALSE is assumed</li></ul> <p>If you want to create a new field and assign it to the section, use a format like this:</p> <pre>fields[10][name]=TextArea fields[10][field_type]=textarea fields[10][new]=true</pre>

delete

URL	/api/v1/api.php?target=resource&action=delete								
Description	Delete a resource.								
Returns	<b>Examples:</b> SUCCESSFUL: {"success":1,"message":"Resource deleted."} ERROR: {"success":0,"message":"No resource found with ID: 57"}								
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Notes/Example</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>INTEGER</td> <td>ID of the resource</td> </tr> </tbody> </table>			Name	Type	Notes/Example	id	INTEGER	ID of the resource
Name	Type	Notes/Example							
id	INTEGER	ID of the resource							
Example URL	/api/v1/api.php?target=resource&action=delete&id=57								

# Release Notes

## Release Notes



- ProVision 4.1.0
- ProVision 4.0.0
- ProVision 3.9.3
- ProVision 3.9.2
- ProVision 3.9.1
- ProVision 3.9.0
- ProVision 3.8.0
- ProVision 3.7.0
- ProVision 3.6.0
- ProVision 3.5.0
- ProVision 3.4.0
- ProVision 3.3.0
- ProVision 3.2.0
- ProVision 3.0.0
- ProVision 2.5

## ProVision 4.1.0

The 4.1.0 release is a minor release that packs in some big updates and customer feature requests.

Contact 6connect at [info@6connect.com](mailto:info@6connect.com) to schedule a demo or get more information.

▼ [Click here to see 4.1.x Release Notes...](#)

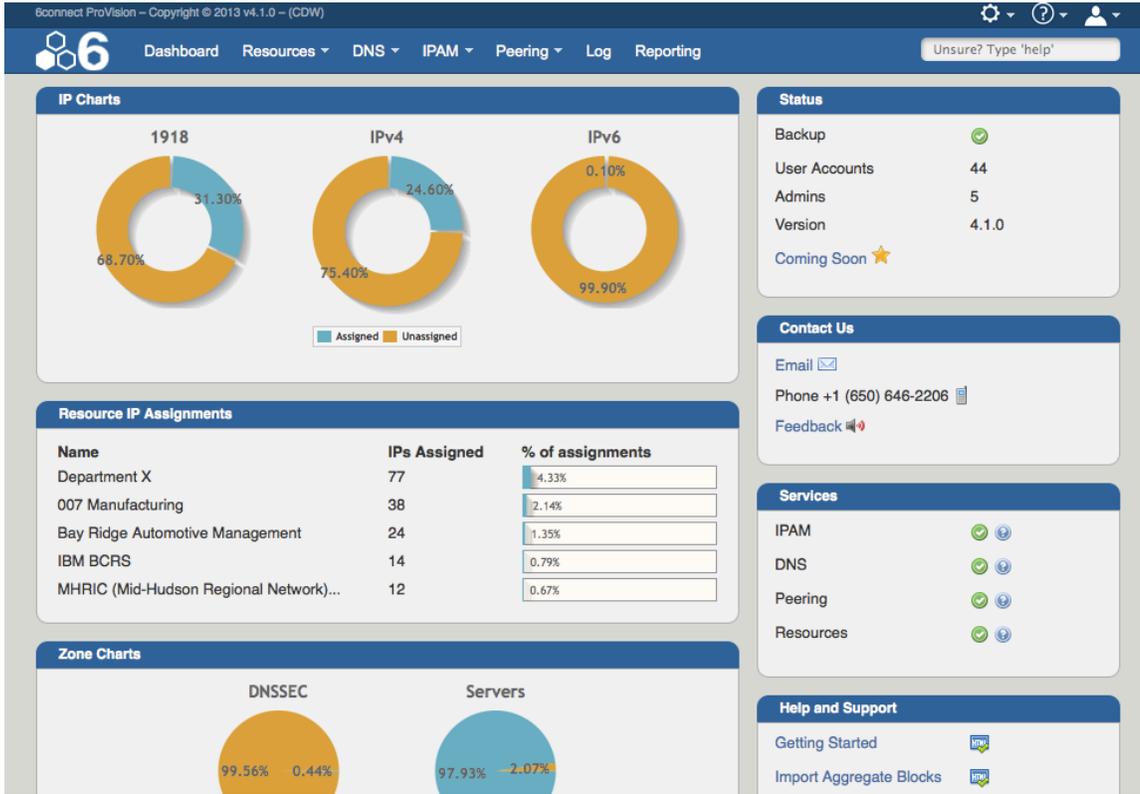
- ProVision 4.1.1
- ProVision 4.1.2
- ProVision 4.1.3
- ProVision 4.1.4
- ProVision 4.1.5
- ProVision 4.1.6
- ProVision 4.1.7
- ProVision 4.1.8
- ProVision 4.1.9
- ProVision 4.1.10
- ProVision 4.1.11

- ProVision 4.1.12
- ProVision 4.1.13
- ProVision 4.1.14
- ProVision 4.1.15

## New Features

### Dashboard Redesign

The main dashboard is completely redesigned with pie and bar graphs for a variety of IPAM, DNS, and resource data.



### IP Detail View Options

View your IP aggregates in the way you want to view them. Set filters for each aggregate based on Mask, LIR, ASN, Tags, Region, Code, VLAN, or Resource Assignment.

198.11.8.0/21 (198.11.8.0 - 198.11.15.255) Tags Used: Customer, DO NOT USE, PTP Regions Used: E

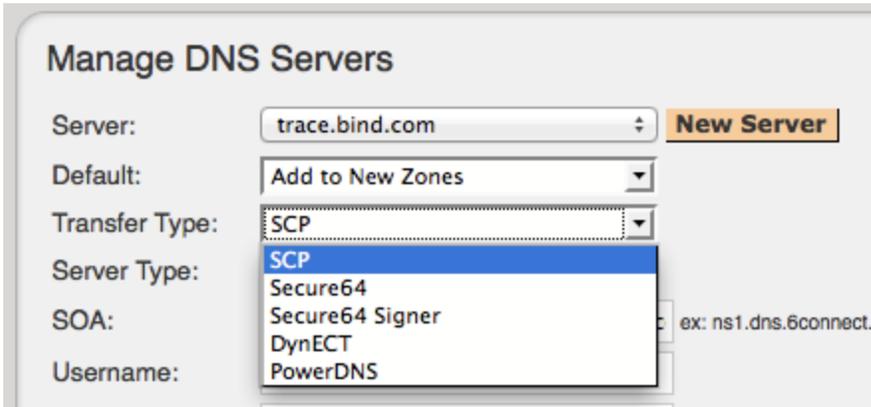
Filtered by: Mask: 24

Filter By: Mask | LIR | ASN | Tags | Region | Code | VLAN | Assigned To | Apply | Clear | Make Default

<input type="checkbox"/>	Address	Hosts	LIR	Region	Notes	Tags	Block Code	Assigned To
<input type="checkbox"/>	198.11.8.0/24	256		BOS		Customer	LIGHTOWER-ISP-F	Has Children
<input type="checkbox"/>	198.11.9.0/24	256		BOS	13601-IP-BSTPMAME	Customer	LIGHTOWER-ISP-F	Potamus Trading LLC
<input type="checkbox"/>	198.11.10.0/24	256		NYC		Customer	LIGHTOWER-ISP-F	Has Children
<input type="checkbox"/>	198.11.11.0/24	256		NYC		Customer	LIGHTOWER-ISP-F	Has Children

### PowerDNS - MySQL Support

Full support for the MySQL PowerDNS backend. Zone updates are pushed out in easy SQL updates to the target server.



### PowerDNS MySQL Importer

Just enter the target server information, click import, and watch all your zone data populate the 6connect database.



### Global Tag Update/Delete Functionality

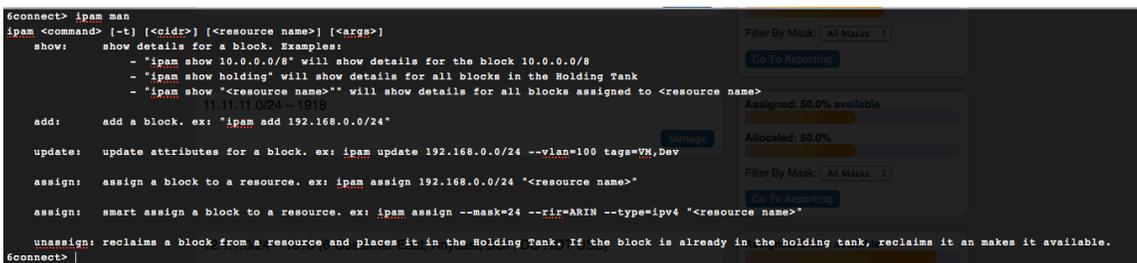
Update and delete tags globally from the primary tags list, just as nature intended.

### DNS Audit Tools (Beta)

View both forward and reverse DNS for entire /24 to get a clear view for how the block is used. Audit against public DNS to see if records are mis-matched or out of sync.

### CLI (Alpha)

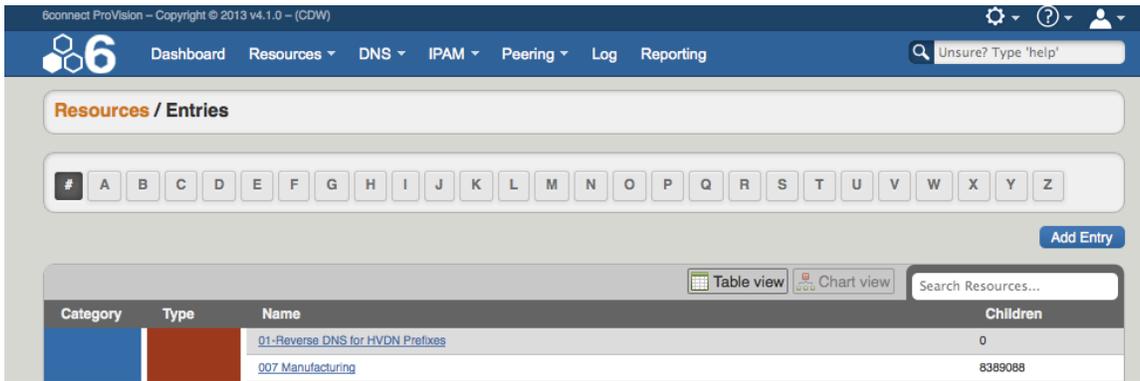
Hate clicking and have a general disdain for UI? This feature is for you! Just press ctrl+shift+s to try 6connect on the CLI with simple commands for quick no-frills interaction.



## Feature Improvements

### Resource UI Functionality

Easier navigation and button placement, alphabetical browse, data usage charts and more!

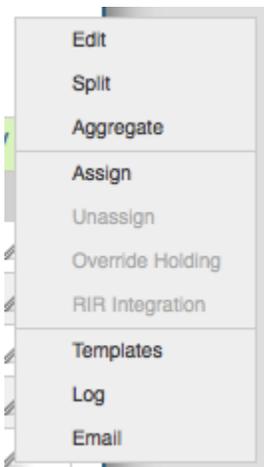


### Header Redesign

The header and menu bars have been streamlined to take up less space, and present menu/options in a clean format.

### Option Menu Redesign

Say goodbye to icons. All option menus have been switched to simple text drop down for easier and more intuitive interaction.



### IP Import Option Additions

You can now import all data relevant to an IP including Resource Holder ID, Resource ID, Org ID, Net Handle, Customer Handle, Generic Code, ASN, Allow Sub-Assignments, and of course all the basic information.

**Step 1: New Import**  
 The IP Import accepts CSV files in a variety of configurations and formats. For an example file, [click here](#).

Job Name:  Description:   No file selected.

File must be in CSV Format.

---

**Step 2: Define Columns**  
 The Import process requires you to enumerate the function of the columns in the provided CSV. There must be either a CIDR column or both an IP and a Mask column. There can be multiple Notes, Tags, and Regions fields. Either a defined RIR column or a Default RIR is required.

Resource Holder ID	IP	Mask	Notes	RIR
Allow Subassignments	IP Block	subnet	Notes	RIR
ASN	10.2.3.0	/24	Internal space	1918
CIDR String	10.5.3.2	/32	Customer assign	ARIN
Generic Code	10.10.10.0	/30	PTP	1918
IP	10.1.0.0	/24		ARIN
Mask	10.2.0.0	/24	Development	1918
Net Handle	10.4.0.0	/24	Ticket 1515	1918
Notes				
Org ID				
Region				

Default RIR:

---

**Step 3: Summary**  
 The importer has detected references which do not currently exist in the system. Rows referencing non-existent data cannot be imported. Please select the

## Bug Fixes

- IM-919 - IPAM menu UI issues in IE9
- IM-1054 - DNS Add Zone allowing spaces in domain name
- IM-1057 - DNS edit - hitting enter key to save record creates strange pop up in Safari
- IM-1119 - Odd session behavior on API calls
- IM-1162 - Read Only user can edit IPAM blocks from IPAM gadget
- IM-1163 - Read Only user can't see time machine data from IPAM gadget

## ProVision 4.1.1

The 4.1.1 is bug fix release only.

### Bug Fixes

- IM-1193 - Path to zones incorrect in 6connect conf file

## ProVision 4.1.2

The 4.1.2 release is a bug fix release only.

### Bug Fixes

- IM-1196 - Error on IP Imports (php 5.4 and up)

## ProVision 4.1.3

The 4.1.3 is bug fix release only.

### Bug Fixes

- IM-1159 - Can't generate Resource report .csv download
- IM-1170 - Update copyright text

IM-1183 - Memory allocation exhaustion on large PowerDNS import  
IM-1197 - IPAM view aggregate tree not restricting to top block  
IM-1201 - IPAM gadget not displaying addresses  
IM-1204 - Javascript error when splitting blocks  
IM-1205 - IPAM Gadget browse assign does not have correct IP type set  
IM-1207 - IPAM Gadget RIR integration broken

## **ProVision 4.1.4**

The 4.1.4 is bug fix release only.

### **Bug Fixes**

IM-1270 - Fix table name in 4.1.0 upgrade script

## **ProVision 4.1.5**

The 4.1.5 is bug fix release only.

### **Bug Fixes**

IM-1187 - Reporting Resource count incorrect  
IM-1194 - Javascript errors on dashboard (text not properly escaped)  
IM-1211 - Update EULA  
IM-1212 - Update text on Welcome  
IM-1215 - Assigned IP count on dashboard inaccurate  
IM-1216 - IPAM - can aggregate assigned block  
IM-1217 - IPAM - Sub-assigning block to same resource results in being unable to reclaim block and other weirdness  
IM-1222 - Move HOSTED\_COMPANY define from constants to globals

## **ProVision 4.1.6**

The 4.1.6 is bug fix release only.

### **Bug Fixes**

IM-990 - ResourceQuery returning incorrect results from empty arrays  
IM-1060 - Edit DNS zone missing mouse over text  
IM-1155 - IPAM Manage Email function does nothing  
IM-1199 - DNS Import Error  
IM-1202 - IPAM utilization incorrect or unsplit aggregates  
IM-1213 - FireFox on Windows - Admin text areas too wide  
IM-1220 - IPAM edit multiple not working as expected  
IM-1221 - IPAM manage loading gif not going away  
IM-1226 - DNS import does not correctly handle multi-line zone declarations  
IM-1228 - Warning on dashboard when no data present

IM-1232 - IP assigned totals incorrect on dashboard

IM-1233 - File swipNetBlock spelled incorrectly - problems on case sensitive systems

IM-1235 - SQL templates contain log data

IM-1238 - SQL template contains 6connect default SOA, should be generic

IM-1241 - SQL template should have more generic default IP list

IM-1242 - SQL template contains /67 in default v4 masks

IM-1243 - SQL template contains bad assortment of default v6 masks

IM-1244 - IPAM admin contains lists for v4/v6 exact matches - no longer used

IM-1245 - SQL template default LIR should contain more generic data

IM-1246 - Process holding tank completed text should contain resource name, not number

IM-1250 - IPAM delete throwing error, instead of returning error on bad delete action

## **ProVision 4.1.7**

The 4.1.7 release is a bug fix release that contains a few feature improvements.

### **Feature Improvements**

IM-1300 - Add all RIRs to LIR manager - All RIR's are now standardized as 1918, ARIN, RIPE, LacNIC, APNIC, and AfriNIC). All customer RIR's should be moved to LIRs and the next update will provide automatic migration.

IM-1218 - Move Resource Holder ID to fundamental attributes for type Resource Holder - The Resource Holder ID is now the Customer ID under the fundamental resource properties.

IM-1268 - Handle multiple ORGs per LIR for ARIN - Added the ability to handle multiple ORGs per LIR for the ARIN RIR.

IM-1156 - Link Resources on Dashboard to Resource records.

IM-1269 - Move DNS and IPAM admin sections to their correct locations - IPAM and DNS admin options are now under the IPAM Admin and DNS Admin sections.

### **Bug Fixes**

IM-1166 - Sorting functions inconsistent on zone list

IM-1195 - Remove old help URLs from admin

IM-1225 - Zone import giving incorrect error when conf file omitted

IM-1249 - Available/Reverse/Holding resources should be protected

IM-1251 - DNS server add by default behavior is not correct

IM-1271 - Field mis-alignment on IPAM gadget

IM-1275 - EULA has escaped quotes

IM-1298 - Default SOA for zones being used when over ridden

IM-1299 - IP Import not working for Radius/LDAP users

## **ProVision 4.1.8**

The 4.1.8 release is a bug fix release that contains a new feature improvement.

### **Feature Improvements**

IM-1255 - User customizable header and support links. Under Admin->Application Settings, you can now set your own custom header image (note the image size), as well as email and phone number to contact for support.

**Application Settings**

Time Zone	America/Los_Angeles
Company Name	6connect
Generic name	Customer
Header Image	/images/topbar-logo.png 64 x 38px Browse... No file selected.
Support Email	support@6connect.com
Support Phone	+1 (650) 646-2206

**Update**

## Bug Fixes

- IM-1317 - IPAM Gadget - Region dropdown is blank when editing
- IM-1313 - \$GENERATE on DNS not producing appropriate PTRs
- IM-1310 - Admin-Users - Check User Permissions field no longer working
- IM-1306 - Unable to add maintainers to new LIRs
- IM-1302 - Can't Smart Assign when using the LIR parameter
- IM-1237 - Admin->API page loading very slow even with no data
- IM-1229 - RIR Integration missing link to LIR Manager if no LIRs exist
- IM-1182 - API - fix request generator to updated resource parameters
- IM-1121 - Logging message missing data for outbound contact email

## ProVision 4.1.9

The 4.1.9 release is a bug fix release that contains a few new feature improvements.

### Feature Improvements

**IM-1258** - Add separate name field to DNS servers, so there is both an FQDN and name. There are now a separate fields for a server Nick Name and it's FQDN/IP. Now DNS servers can be easily recognized by their common/nick name or their FQDN/IP address, as both are displayed representing the server.

## Manage DNS Servers

Server:

Nick Name:

FQDN or IP:  ex: ns1.dns.6connect.net or 216.239.32.10

Default:

Transfer Type:

Server Type:

SOA:  ex: ns1.dns.6connect.net. hostmaster.6connect.net.

**IM-1329** - Add field override for Net Name on swip.

**IM-1312** - Add override field for Customer Name on swip

Both the ARIN Customer Name and Net Name can be over ridden from the automatically generated defaults at swip time using the fields provided on the RIR Integration Panel.

ARIN Integration: 67.221.244.0/28 (67.221.244.0 - 67.221.244.15)

Org Handle	Admin POC	Net POC	Abuse POC	Net Name Prefix	API Key
<input checked="" type="radio"/> CONNE-81	admin-c	tech-c	abuse-c	NET	API-B7BF-F4AD-4695-8508

Net Name:

Registrar Public Name (Simple Reassign only):

By default, when ARIN blocks are SWIPed the customer name in the WHOIS database will be set to the assigned resource name. To override this, enter a public name to use in this field.

## Bug Fixes

**IM-1330** - API - IPAM get parameter block not working as expected

**IM-1324** - Missing images on admin-dhcp.php

**IM-1323** - API - Inconsistent results between Resource Get and Resource Add

**IM-1322** - IPAM Gadget - Action menu display issue sub assignments allowed

**IM-1321** - IPAM Gadget - All tags should display during edit, not just in use

**IM-1319** - IPAM Admin - Bad URL "Back to IPAM Admin" in tag editor

**IM-1259** - configTest - Add test items for expect, namp, and php hash extension

**IM-1142** - Center command results given by assistant

## ProVision 4.1.10

### Bug Fixes

**IM-1290** - Resource: list view does not have default when Resource link clicked - defaults to empty list.

IM-1320 - Resource: allowed to become it's own parent.

IM-1327 - Dashboard: displaying information for resources above current sign in permission level.

IM-1331 - Import : progress bar flashes/drops to 0 at times while upload in progress.

IM-1334 - Import: new resources not adding correctly from IP import.

IM-1337 - API: Valid requests are being rejected.

IM-1338 - License check interval too long

## ProVision 4.1.11

### New Features

#### DNS Deletion Queue

Addition of a deletion queue for zones for flexible removal of zones from DNS Servers.

6connect ProVision - © 2014 v4.1.1 - (Awesome Co.)

Dashboard Resources DNS IPAM Peering Log Reporting Search or type help

Create a DNS Zone

Enter Zone Name Select Awesomers...

Page: 1 (14 total)

Zone	Customer	Tags	DNSSEC	DS	Records	Actions
<a href="#">1800druidia.com</a>	<a href="#">Apple Computer</a>				2	
<a href="#">blimey.com</a>	<a href="#">Acer Worldwide</a>				2	
<a href="#">cnn.com</a>	<a href="#">Acer Worldwide</a>				0	
<a href="#">expect.com</a>	<a href="#">Apple Computer</a>				2	
<a href="#">fabulous.com</a>	<a href="#">APC Schneider</a>				2	
<a href="#">fancy.com</a>	<a href="#">Acer Worldwide</a>				2	
<a href="#">happybirthday.com</a>	<a href="#">Joe's Bait and Tackle and ISP</a>	super	DNSSEC	X	2	
<a href="#">happyfestivus.com</a>	<a href="#">Joe's Bait and Tackle and ISP</a>	less super			4	
<a href="#">jd-sweatshop.com</a>	<a href="#">Jack Daniel's Sweat Shop</a>				2	
<a href="#">memychristmas.com</a>	<a href="#">Joe's Bait and Tackle and ISP</a>	most super			0	
<a href="#">tuco.com</a>	<a href="#">Acer Worldwide</a>				2	
<a href="#">zone1</a>	<a href="#">Anna's Test Site</a>				3	
<a href="#">1.1.62.in-addr.arpa</a>	<a href="#">Jack Daniel's Sweat Shop</a>				2	
<a href="#">0.0.0.5.b.d.0.1.0.0.2.ip6.arpa</a>	<a href="#">APC Schneider</a>				1	

### Bug Fixes

IM-1344 to IM-1351 - All bug fixes related to a bad version of bootstrap.

## ProVision 4.1.12

### Bug Fixes

The 4.1.12 release is bug fix only release.

IM-1372 - Search - Search box breaks on Admin navigation when large custom header image set

IM-1371 - Dashboard - Reverse DNS resource appearing on dashboard stats

IM-1369 - DNS - Large DNS pushes appear to be hanging

IM-1368, IM-1343 - Reporting - CSV and HTML column headers do not match values

- IM-1366 - DNS - Menus not appearing on pages after 1
- IM-1361 - DNS - Admin DNS "Edit DNS Records" links to nonexistent page (404 error)
- IM-1360 - IPAM - Admin IPAM "Process Holding Tank" Back to IPAM links to nonexistent page (404 error)
- IM-1358 - DNS Import - Can't edit resource assignment during import
- IM-1357 - DNS Import - Hanging on single zone import
- IM-1355 - IPAM - Aggregate details log menu not pulling log entries
- IM-1354 - Reporting - Error on user reporting
- IM-1352 - Reporting - All users showing in user list when permissions should restrict
- IM-1335 - IPAM Import - Data not revalidated after editing import row
- IM-1305 - DNS - Download of all zip files permissions issue
- IM-1301 - Resource - Customer field type box disappearing on add
- IM-1296 - Admin - Create backup now option updates general settings
- IM-1224 - Admin - API request generator slow to load
- IM-1223 - IPAM - LIR manager not saving Org info
- IM-1039 - Resource - ZIP/Postal code field missing in edit dialog

## ProVision 4.1.13

### New Features

#### Add User CLI (IM-1387)

Local installs can now add a user from the CLI with the add-user tool found in the ./tools directory.

```

/tools> php add-user.php
Add New User - 6connect ProVision
-----
62 - Remove edit/delete options from ipam tags list, options no longer needed
Enter email (username):
gary@6connect.com
First Name:
Gary
Last Name:
Canty
Password:
changemefast!
Add to Global Admins? (y/n):
y
5 - DNS - Delete and Push options shows in action menu when user does not have Delete per
User `gary@6connect.com` was added.

```

### Updates

- IM-1385 - IPAM - RIPE API updates/invalid RIPE API URIs
- IM-1382 - Remove edit/delete options from ipam tags list, options no longer needed

### Bug Fixes

- IM-1389 - Reporting - Utilization numbers incorrect for IPAM reporting
- IM-1386 - IPAM API - Direct Assign - resourceHolderId parameter not working as expected
- IM-1383 - Session timeout logs do not set username

IM-1375 - DNS - Delete and Push options shows in action menu when user does not have Delete permissions

IM-1374 - Resource - Unable to save resources on edit (javascript error)

IM-1359 - Dashboard - IPAM/DNS graphs inaccurate for some users, counting data out of permissions scope

IM-1041 - IPAM Admin - Edit Contact Roles link depreciated, should be removed

## ProVision 4.1.14

### New Features

#### Rebuild Permissions CLI Tool (IM-1405)

Command line tool to rebuild the user group permissions for a specific group or resource, or group/resource combination.

### Bug Fixes

IM-1404 - User Admin - Group Management page error 'Circular Reference

IM-1403 - Dashboard - Warnings on dashboard when user has no IPAM permissions

IM-1402 - Reporting - IPAM reporting assigned numbers still incorrect

IM-1401 - DNS - Text update for Delete and Push confirmation window

IM-1400 - DNS - Default servers not added to zones as expected

IM-1399 - IPAM API - IPAM get call ignoring valid tags

IM-1392 - DNS - Action menu disappears when using filter function

IM-1390 - Admin DNS - Push to Secure64 errors (short timeout)

## ProVision 4.1.15

The 4.1.15 release is a bug fix release that contains a few feature improvements.

### Feature Improvements

CFR-76 - Add Configuration setting for submitting changes to RIPE test v. live database

IM-140 (part) - Password management improvements - stored passwords are now stored outside of primary database structure.

### Bug Fixes

IM-1395 - DNS - Assign DNS zone drop down displaying incorrect information

IM-1396 - IPAM - SWIP check mark displayed when IP is only assigned

IM-1421 - IPAM - Smart assign producing two log messages (should be one)

IM-1423 - API - Hashed API requests fail if first character of key is 0

IM-1424 - IPAM - Scan loading indicator never goes away