



# Platform Documentation

## ProVision

Application Version 5.0.3

### Covering:

- Resource Manager
- IP Address Manager
- DNS Manager
- DHCP Manager
- Peering Manager

For additional information, please visit <http://docs.6connect.com> or contact 6connect at [support@6connect.com](mailto:support@6connect.com)

1. Home	4
1.1 ProVision Installation Guide	5
1.1.1 Hosted Instances Guide	6
1.1.2 Local / VM Installation Guide	7
1.1.2.1 System Requirements	8
1.1.2.2 Backup and Redundancy: Local / VM	10
1.1.2.3 CentOS Configuration Guide	11
1.1.2.4 6connect Local Software Installation	15
1.1.2.5 6connect Local Software Upgrades	18
1.2 ProVision Getting Started	20
1.2.1 Resource Concepts	21
1.2.1.1 Classes	22
1.2.1.2 Database Layout	25
1.2.1.3 The Asset System	28
1.2.2 Workflow Concepts	30
1.2.3 UI Element Legend	32
1.3 ProVision User Guide	34
1.3.1 The Dashboard	35
1.3.2 Working with Resources	37
1.3.2.1 Customizing Sections	39
1.3.2.2 Customizing Fields	41
1.3.2.3 Gadgets	42
1.3.2.3.1 XML Specifications	47
1.3.3 DNS Tab	48
1.3.4 DHCP Tab	50
1.3.5 IPAM Tab	54
1.3.5.1 Working with IP Blocks	56
1.3.6 Peering v2	63
1.3.6.1 Peering Workflow	64
1.3.6.2 Peering - Common Tasks	65
1.3.6.2.1 Add Routers	66
1.3.6.2.2 Add Sessions	67
1.3.6.2.3 Import Sessions	68
1.3.6.3 Managing Peer Communications	69
1.3.6.4 Managing Peer Sessions	71
1.3.7 Log	73
1.3.8 Reporting	75
1.4 ProVision Admin Guide	77
1.4.1 Admin Preferences	78
1.4.2 Authentication Options	80
1.4.2.1 LDAP Authentication	81
1.4.2.2 LDAP Authentication on Windows Server	83
1.4.2.3 RADIUS Authentication	86
1.4.3 IPAM Administration	90
1.4.3.1 IPAM Parameters	92
1.4.3.2 Holding Tank Management	94
1.4.3.3 LIR Management and Use	95
1.4.3.3.1 ARIN LIR Setup and Use	98
1.4.3.3.2 RIPE LIR Setup and Use	100
1.4.4 DNS Administration	102
1.4.4.1 Working with DNS Zones	106
1.4.4.1.1 Editing DNS Zones	108
1.4.4.2 Configuring ISC BIND Support	111
1.4.4.3 Configuring DynECT Support	112
1.4.4.4 Configuring PowerDNS Support	113
1.4.4.5 Configuring Secure64 Support	116
1.4.4.6 Configuring DNSSEC	123
1.4.4.7 Configuring Split Horizon/Views	124
1.4.4.8 Configuring DNS Templates	127
1.4.5 Importing Your Data	129
1.4.5.1 Import Aggregate Blocks	131

1.4.5.2 Import DNS Zones	133
1.4.5.3 Resource Import Tool	138
1.4.5.3.1 Resource Importer Walkthrough - Step 1	140
1.4.5.3.2 Resource Importer Walkthrough - Step 2	142
1.4.5.3.3 Resource Importer Walkthrough - Step 3	144
1.4.5.3.4 Resource Importer Walkthrough - Step 4	145
1.4.5.3.5 Resource Importer Walkthrough - Step 5	149
1.4.5.3.6 Resource Importer Walkthrough - Step 6	150
1.4.6 Users & Permissions	151
1.4.6.1 Global Permissions	152
1.4.6.2 Resource Permissions	153
1.4.6.3 Users and Groups	154
1.4.6.4 Verifying Permissions	157
1.4.7 API Tab	158
1.4.8 Templates	159
1.5 ProVision Developer Tools	161
1.5.1 API v1	162
1.5.1.1 1 - Overview	163
1.5.1.2 2 - Making API Requests	164
1.5.1.3 3 - SDK - PHP	166
1.5.1.4 API Module - Admin and Audit	168
1.5.1.4.1 Authentication Testing	169
1.5.1.4.2 Log Management	172
1.5.1.4.3 Zone Templates	176
1.5.1.5 API Module - DHCP	180
1.5.1.6 API Module - DNS	207
1.5.1.7 API Module - IPAM	254
1.5.1.8 API Module - LIR	286
1.5.1.9 API Module - Peering	289
1.5.1.10 API Module - Resource	302
1.5.1.11 How Do I...	314
1.5.2 CLI (Alpha)	322
1.6 Help & Support	324
1.6.1 Tutorials	325
1.6.1.1 Common Tasks	326
1.6.1.2 UI Tours	327
1.6.2 FAQ	328
1.6.3 Additional Resources	330
1.6.3.1 Import Templates	331
1.6.3.2 List of Abbreviations	332
1.6.3.3 Previous Documentation Versions	333
1.6.4 Feedback and Feature Requests	334
1.7 6connect Scanlet (Beta)	335
1.8 Release Notes	339
1.8.1 ProVision 5.0.0	340
1.8.1.1 ProVision 5.0.4	347
1.8.1.2 ProVision 5.0.3	352
1.8.1.3 ProVision 5.0.2	353
1.8.1.4 ProVision 5.0.1	355
1.8.2 ProVision 4.2.0	356
1.8.2.1 ProVision 4.2.1	357
1.8.3 ProVision 4.1.0	360
1.8.3.1 ProVision 4.1.23	364
1.8.3.2 ProVision 4.1.22	365
1.8.3.3 ProVision 4.1.21	366
1.8.3.4 ProVision 4.1.20	368
1.8.3.5 ProVision 4.1.19	369
1.8.3.6 ProVision 4.1.18	372
1.8.3.7 ProVision 4.1.17	373
1.8.3.8 ProVision 4.1.16	374
1.8.3.9 ProVision 4.1.15	375

1.8.3.10 ProVision 4.1.14	376
1.8.3.11 ProVision 4.1.13	377
1.8.3.12 ProVision 4.1.12	378
1.8.3.13 ProVision 4.1.11	379
1.8.3.14 ProVision 4.1.10	380
1.8.3.15 ProVision 4.1.9	381
1.8.3.16 ProVision 4.1.8	383
1.8.3.17 ProVision 4.1.7	384
1.8.3.18 ProVision 4.1.6	385
1.8.3.19 ProVision 4.1.5	386
1.8.3.20 ProVision 4.1.4	387
1.8.3.21 ProVision 4.1.3	388
1.8.3.22 ProVision 4.1.2	389
1.8.3.23 ProVision 4.1.1	390
1.8.4 ProVision 4.0.0	391
1.8.5 ProVision 3.9.0	394
1.8.5.1 ProVision 3.9.3	395
1.8.5.2 ProVision 3.9.2	396
1.8.5.3 ProVision 3.9.1	397
1.8.6 ProVision 3.8.0	398
1.8.7 ProVision 3.7.0	399
1.8.8 ProVision 3.6.0	400
1.8.9 ProVision 3.5.0	401
1.8.10 ProVision 3.4.0	402
1.8.11 ProVision 3.3.0	403
1.8.12 ProVision 3.2.0	404
1.8.13 ProVision 3.0.0	405
1.8.14 ProVision 2.5	406
1.9 Coming Soon	407



# **ProVision Installation Guide**

## **Installing ProVision**

You have 6connect ProVision and now it's time to set it up! 6connect offers both cloud hosted instances and local installations of ProVision. Follow the links below for specific instructions on each instance type.

For setup assistance or additional information, you can contact our [support](#) team at [support@6connect.com](mailto:support@6connect.com).

### **Table of contents**

- [Hosted Instances Guide](#)
- [Local / VM Installation Guide](#)

# Hosted Instances Guide

## Hosted Instances Guide

With a cloud hosted instance of ProVision, all you need is one of the following web browsers with an internet connection and login credentials!

Once you have confirmed that you have a supported browser and valid login, you can proceed to [ProVision Getting Started](#), the [ProVision User Guide](#), or the [ProVision Admin Guide](#) to learn more about ProVision.

### 6connect Cloud Hosted Instance: Browser Requirements

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)

### Backup and Redundancy

#### 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 Installation Guide

## Installing ProVision

Local and VM installs of ProVision have specific requirements and configuration settings. Please follow the links below for detailed instructions on how to set up your local installation of ProVision.

For setup assistance or additional information, you can contact our [support](#) team at [support@6connect.com](mailto:support@6connect.com).

### Table of contents

- [System Requirements](#)
- [Backup and Redundancy: Local / VM](#)
- [CentOS Configuration Guide](#)
- [6connect Local Software Installation](#)
- [6connect Local Software Upgrades](#)

## System Requirements

- ProVision System Requirements
  - 6connect Virtual Machine
  - 6connect Locally Hosted Instance

### ProVision System Requirements

## 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 5.5+: <http://www.mysql.com/downloads/>

**REQUIREMENT: MySQL master/master replication**

Please note that MySQL 5.5.38+ is required for master/master replication functions to work correctly

**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: Local / VM

### Backup and Redundancy

## 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

## CentOS Configuration Guide

- CentOS Configuration Guide
  - Before You Begin
  - Install and Configure MySQL
  - Install and Configure PHP
  - Install PCNTL
  - Install and Configure Apache and SSL
  - Install and Configure Source Guardian
  - Configure SELinux
  - Configure IPTables
  - Install Radius
  - Install 6connect

### Before You Begin

Ensure that [System Requirements](#) have been met prior to proceeding with the 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 'somesongpassword';
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
```



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

When ready, proceed to [6connect Local Software Installation](#) for detailed installation instructions.

## 6connect Local Software Installation

### Local Software Installation & Specific Configuration Instructions

- Local Software Installation & Specific Configuration Instructions
  - Before You Begin
  - Apache Configuration Requirements
  - MySQL Configuration
  - PHP Configuration
  - Source Guardian
  - Additional PHP Extensions
  - Additional System Packages
  - DNS Tools and Packages
  - Install 6connect ProVision Software

#### Before You Begin

Ensure that [System Requirements](#) have been met and [CentOS configuration](#) followed (if applicable) prior to proceeding with the configuration and installation instructions on this page.

### Apache Configuration Requirements



**mod\_rewrite, mod\_ssl, mod\_deflate, and mod\_headers 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:

#### **Apache 2.2:**

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

#### **Apache 2.4:**

```
<Directory /<ProVision webroot>>
    Options FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
```



#### **AllowOverride**

Please note that if the AllowOverride is not enabled on the doc root - there will be multiple issues in the ProVision UI!

### 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 version 5.0.0 of ProVision, PHP versions >5.5 are required

```
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

## Install 6connect ProVision Software

The local installation process is as follows:



Scripts must be run in the exact order listed.

1. Install all the packages, extensions, and perform configuration listed above and the Source Guardian extension.
  - a. 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.
  - a. `tar -xof productionBuild-4.0.3.tar`  
This will place all the new files into your web root directory.
3. Set the directory permissions for all use data directories:
  - a. Run script from the command line as root: "configDir.sh"
4. Check for all installed modules and license info:
  - a. Run script from web browser: "configTest.php"
5. Configure the database info in globals.php, install the default database and initial user:
  - a. Run script from web browser: "configBootstrap.php"
6. Carefully note the login credentials provided before continuing.
7. Set the secure directory, create keys, and write this info to globals.php:
  - a. Run script from the command line as root: "configSecureKeys.sh"
8. Log in and use!

## 6connect Local Software Upgrades

- Upgrading Local Installs of 6connect Provision

### Upgrading Local Installs of 6connect Provision

#### Upgrades after 4.1.20 and up

You must be running at least 4.1.20 to follow any of the upgrade methods listed below. If you are not yet at this version, upgrade to 4.1.20 using the old upgrade process first, and then continue using the new upgrade process or contact [support@6connect.com](mailto:support@6connect.com) or any questions or to schedule an upgrade to the latest version.

There are now 3 different methods to run upgrades.

**Old Method** (See Upgrades prior to 4.1.19 for detailed instructions)

Download the latest 6connect tar file from <https://cloud.6connect.com/Download/Latest/>

Extract in web root.

Run the upgrade scripts located in upgrade/scripts in order of version number via `php <upgrade-script.php> -v`

#### Command Line

In upgrade/scripts run 'php upgrade.php -h' to get the help and full usage of upgrade.php. This script will automatically get the latest tar file, create a backup, and run all the necessary upgrades between the current and latest version. The most common usage of upgrade will look like this 'php upgrade.php -v -b </path/to/store/backup>'

#### GUI

In the 6connect tool, navigate to Admin. If there is a new version available, an Upgrade button will be available. Click on the Upgrade Now button to go to the upgrade page. It will automatically download the latest version available, run all upgrade scripts, and create a log of the upgrade process.

#### Upgrades prior and up to 4.1.19

IMPORTANT NOTE FOR 4.1.15 - The configDir.sh script must be run as root after the 4.1.15 tar file is unpacked and before running upgrade-4.1.15.php.

1. Create a database backup.

```
mysqldump -u <user> -p<pass> <6connect database name> > /tmp/6connectDBBackup.<date>.sql
```

2. Create a directory backup. Even if you have offsite backup's with 6connect enabled, perform this step to ensure the most current data is saved.

```
tar -cvf 6connectFileBackup.<date>.tar /path/to/webroot
```

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

```
tar -xof productionBuild-4.1.4.tar
```

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

4. Run database upgrades, located in ./dev.

The simple rule of thumb is to run every database upgrade from the version after yours, to the version you want to get to. Here is the short cut list:

If upgrading from 4.1.0 or higher:

```
php upgrade-4.1.3.php -v
```

```
php upgrade-4.1.4.php -v
```

```
php upgrade-4.1.5.php -v
```

```
php upgrade-4.1.6.php -v
```

```
php upgrade-4.1.7.php -v
```

```
php upgrade-4.1.8.php -v
```

```
php upgrade-4.1.9.php -v
```

```
php upgrade-4.1.10.php -v
```

```
php upgrade-4.1.11.php -v
```

```
php upgrade-4.1.12.php -v
```

```
php upgrade-4.1.13.php -v
```

```
php upgrade-4.1.14.php -v
```

```
configDir.sh <web user> (after tar 4.1.15 tar file unpacked)
```

```
php upgrade-4.1.15.php -v
```

```
php upgrade-4.1.16.php -v
php upgrade-4.1.17.php -v
php upgrade-4.1.18.php -v
php upgrade-4.1.19.php -v
```

If upgrading from 3.9.3:

Contact 6connect Support - [support@6connect.com](mailto:support@6connect.com)

5. Check directory/file permissions for the following and make sure they read/write for the web user:

archive

keys

scans

zones

data/globals.php

images/custom

configDir.sh can be run to correct any permissions issues.

Check the imports directory for read/write permission in the configured php session dir.

6. Go to <http://<web root>/configTest.php>. If there are any configuration errors listed, they must be corrected.

7. Login and use!

# **ProVision Getting Started**

## **Welcome to ProVision!**

Our Getting Started documents provide an overview of concepts to orient you to working in ProVision. Below are some of the resources available. If you need setup assistance or additional information, you can contact our support team at [support@6connect.com](mailto:support@6connect.com).

## **ProVision Getting Started**

**Resource Concepts** - The Resource Management System is a key component of ProVision. This system supports a variety of hierarchies and metadata - understanding how these pieces can be used is important prior to importing data or setting up the application.

**Workflow Concepts** - ProVision has two distinct interfaces depending on the user level and task. It is important to understand how these interfaces work together from the centralized data. This is important for user on-boarding and training of internal operations staff, developers or engineering teams.

**UI Element Legend** - ProVision has some UI elements that you should be familiar with for easy day to day operation.

## **ProVision User Guide**

The user guide gives you an overview on the standard UI functions of ProVision and installed Modules.

## **ProVision Admin Guide**

The Admin Guide provides an overview of administrative functions of the different functional areas of ProVision.

## **ProVision Developer Tools**

The Developer Tools section has details on our [API](#) and related information - including [code samples](#).

## **Additional Resources**

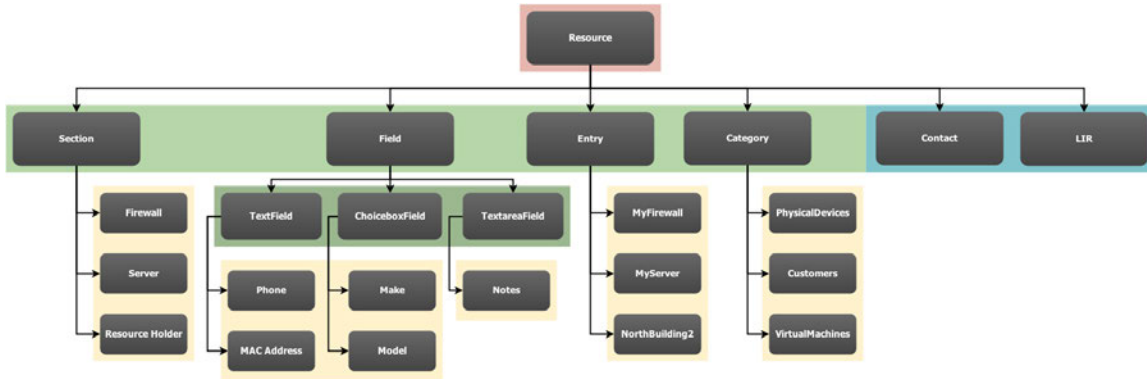
You can also browse the [Tutorials](#) and [FAQ](#), if you have any questions, please contact our support team at [support@6connect.com](mailto:support@6connect.com).



# Resource Concepts

## Overview

In Provision, the Resource System (RS) is an expression of object-oriented programming. In this context, the term “resource” is equivalent to the term “object”, where an object is an instance of a class. Traditionally in OOP, there is an Object class that is the root of the class hierarchy. In the RS, the Resource class is the root class. Every class in the system has Resource as a superclass and all resource objects implement the methods of that class.



The diagram above shows examples of resource sub-types. The items on a green or blue background are types of resources; they each have their own corresponding Class. An item on a yellow background is an example of an object that could have been instantiated from the class (resource type) that it's part of.

### Additional Information:

- [Classes](#)
- [Database Layout](#)
- [The Asset System](#)

# Classes

## Classes

"A class--the basic building block of an object-oriented language such as Java--is a template that describes the data and behavior associated with instances of that class. When you instantiate a class you create an object that looks and feels like other instances of the same class."

*Mary Campione and Kathy Walrath, The Java Tutorial: Object-Oriented Programming for the Internet, The Java Series (Reading, Mass.: Addison Wesley, 1996)*

- [Classes](#)
  - [Class Resource](#)
    - [Properties](#)
  - [Examples](#)
    - [1 - PHP](#)
    - [2 - API request](#)

## Class Resource

```
class Resource {
    public int      $id;
    public string  $name;
    public string  $slug;
    public string  $type;
    public int      $parent_id;
    public int      $category_id;

    protected array $attr    = array();
    protected bool  $loaded = FALSE;

    public object get_attr( string $key );
    public void   set_attr( string $key, object $value );
    public bool   loaded();
}
```

## Properties

As you can see from the [database layout](#), the public properties of the Resource class are all part of the main **resource** table. The two protected properties **attr** and **loaded** are created at runtime. There are many situations where only the core information is required. To improve performance, attribute data is ignored when it is not required. Attributes are stored in the database as longtext; non-primitive types (such as arrays) are serialized and stored as a string.

```
$attr
A key-value store of the attributes that exist in the resource_attr table.

$loaded
A boolean value which is used to indicate whether or not the attributes have been loaded.
```

### Why do some attributes have names that start with an underscore?

This is the convention for storing metadata. Most attributes are for storing data that is created by the user and is available to be directly edited by the user. When we want to store system data, configuration options, or just data that isn't meant for human consumption - we store it as metadata. An attribute is identified as being metadata by the convention of starting the name/key of the attribute with an underscore character (e.g. `_meta`). If you are interfacing with the API, you will frequently come across metadata. You're welcome to modify the metadata of a resource (if you know what you're doing) or add metadata attributes for known metadata keys, but you shouldn't create your own attributes with keys that begin with an underscore. Future versions of ProVision will use new metadata keys without warning, and if there is a naming conflict, your data could be lost.

## Examples

These examples show the different methods that can be used to find and load a Resource object. They also show different data structures that are used to represent the object.

### 1 - PHP



#### Internal code example

To help users better understand how ProVision works, some of the examples in this documentation are of internal processes. They can contain code that only works when used as part of the core system and thus is not applicable to 3rd party development. The API is currently the only way for external tools to integrate with ProVision. Any example that contains internal code should be clearly labeled. Some common characteristics of these examples are code that doesn't use the API and code written in PHP (most example code will be in JavaScript).

This example uses the ResourceQuery class to find a resource object and then prints the result. It is included to show the similarity between finding a resource via the API and what happens under the hood.

```
$params = array(
    'slug' => 'tlr'
);
$resourceQuery = new ResourceQuery();
$resource = $resourceQuery->query($params);

var_dump($resource);
/*
array (size=1)
  0 =>
    object(Resource)[27]
      protected 'id' => string '1' (length=1)
      protected 'name' => string 'TLR' (length=3)
      protected 'slug' => string 'tlr' (length=3)
      protected 'type' => string 'resource' (length=8)
      protected 'parent_id' => null
      protected 'category_id' => null
      protected 'attr' =>
        array (size=0)
          empty
      protected 'loaded' => boolean true
*/
```

### 2 - API request

This is a standard API request, the request data is urlencoded and the result is JSON

**/api/v1/api.php?target=resource&action=get&slug=TLR**

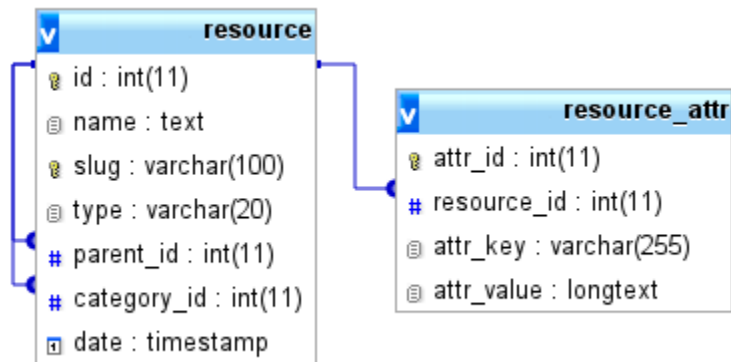
```
{
  "success": 1,
  "message": "Search successful",
  "data": [
    {
      "id": "1",
      "name": "TLR",
      "slug": "tlr",
      "type": "resource",
      "parent_id": null,
      "category_id": null,
      "attr": {}
    }
  ]
}
```

## Database Layout

### Database Layout

Details of the database and tables used by the RS are not necessary and should have no bearing on usage or API based development. However, a visualization of these tables may help some users better understand how the RS works, so they are provided below.

Figure



### Relations

`resource`.`category\_id` -> `resource`.`id`

`resource`.`parent\_id` -> `resource`.`id`

`resource\_attr`.`resource\_id` -> `resource`.`id`

### Structure in SQL

**resource**[Expand source](#)

```
--
-- Table structure for table `resource`
--
CREATE TABLE IF NOT EXISTS `resource` (
  `id` int(11) NOT NULL,
  `name` text NOT NULL,
  `slug` varchar(100) NOT NULL,
  `type` varchar(20) NOT NULL,
  `parent_id` int(11) DEFAULT NULL,
  `category_id` int(11) DEFAULT NULL,
  `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1115 ;
--
-- RELATIONS FOR TABLE `resource`:
--   `category_id`
--     `resource` -> `id`
--   `parent_id`
--     `resource` -> `id`
--
--
-- Indexes for dumped tables
--
--
-- Indexes for table `resource`
--
ALTER TABLE `resource`
  ADD PRIMARY KEY (`id`), ADD UNIQUE KEY `slug` (`slug`), ADD KEY `category_id`
(`category_id`), ADD KEY `parent_id` (`parent_id`);
--
-- AUTO_INCREMENT for dumped tables
--
--
-- AUTO_INCREMENT for table `resource`
--
ALTER TABLE `resource`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=1115;
--
-- Constraints for dumped tables
--
--
-- Constraints for table `resource`
--
ALTER TABLE `resource`
ADD CONSTRAINT `resource_ibfk_1` FOREIGN KEY (`category_id`) REFERENCES `resource` (`id`)
ON DELETE SET NULL ON UPDATE CASCADE,
ADD CONSTRAINT `resource_ibfk_2` FOREIGN KEY (`parent_id`) REFERENCES `resource` (`id`)
ON DELETE SET NULL ON UPDATE CASCADE;
```

## resource\_attr

[Expand source](#)

```
--
-- Table structure for table `resource_attr`
--
CREATE TABLE IF NOT EXISTS `resource_attr` (
  `attr_id` int(11) NOT NULL,
  `resource_id` int(11) NOT NULL,
  `attr_key` varchar(255) NOT NULL,
  `attr_value` longtext NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=6744 ;
--
-- RELATIONS FOR TABLE `resource_attr`:
--   `resource_id`
--     `resource` -> `id`
--
--
-- Indexes for dumped tables
--
--
-- Indexes for table `resource_attr`
--
ALTER TABLE `resource_attr`
  ADD PRIMARY KEY (`attr_id`), ADD KEY `item_id` (`resource_id`);
--
-- AUTO_INCREMENT for dumped tables
--
--
-- AUTO_INCREMENT for table `resource_attr`
--
ALTER TABLE `resource_attr`
MODIFY `attr_id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=6744;
--
-- Constraints for dumped tables
--
--
-- Constraints for table `resource_attr`
--
ALTER TABLE `resource_attr`
ADD CONSTRAINT `resource_attr_ibfk_1` FOREIGN KEY (`resource_id`) REFERENCES `resource`
(`id`) ON DELETE CASCADE ON UPDATE CASCADE;
```

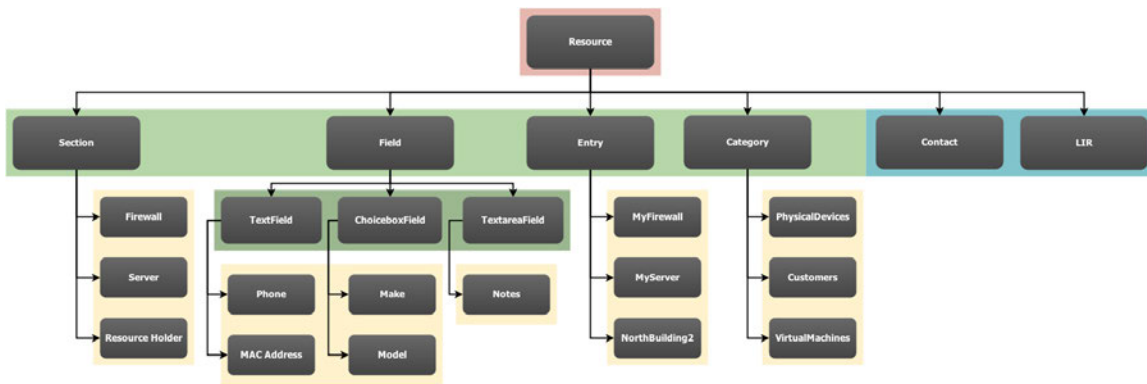
# The Asset System

## Prerequisites

Some knowledge of object orientated programming (OOP) is recommend to understand the following description of the Asset System. If you are unfamiliar with OOP concepts, I would recommend reading a tutorial such as this one (<http://docs.oracle.com/javase/tutorial/java/concepts/index.html>) provided by Oracle or this one ([http://msdn.microsoft.com/en-us/library/ca22fyhc\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/ca22fyhc(v=vs.90).aspx)) provided by Microsoft, to help you understand terms like class, object, instantiate, property, method, and others.

## Overview

The asset system is a content management system (CMS) that is built as an extension to the resource system. It's the main use of the resource system, and to many, the terms "asset system" and "resource system" can seem synonymous. In the diagram below, the Resource class is at the top in red. The child-classes that make up the asset system are in green. Yellow is used for examples of objects (not classes) that could/would have been instantiated from their Class. And the items in blue are examples of resource child-classes (resource types) that exist outside of the asset system.



## Introduction

When writing software, the developer creates classes. A class is like a blueprint for objects. The class defines the properties and methods that the future objects will have, and like blueprints, multiple objects can be created from a single class. The Resource Class is a class, and each resource "type" (e.g. Section, Field, Contact, ect.) has a class, something which has been written in core code and cannot be changed by the user. The purpose of the asset system is to reproduce this fundamental low-level class-object system in such a way that the user can create their own classes, properties, methods, and objects without needing to dive into the code.

## Components

### Section

Sections are like classes, they are the templates/blueprints of the asset system. To create the structure of the blueprint, the user assigns fields (i.e. properties) and sometimes gadgets (i.e. methods) to the section.

### Entry

Entries are the objects of the asset system. An entry cannot be created without a section to use as its blueprint. Creating an entry from a section is like instantiating an object from a class.

### Field

Fields are the properties of the class. At time of writing, Field is the only asset-resource class that has it's own child-classes; this is to accommodate the different types of fields. For example, when creating a class *Car*, the developer might give the *Car* class the property *String color*. In a similar fashion, a user of the Asset System could create a Section called *Truck*, a TextField called *color*, and then assign that textfield to the section. When the user goes to create an entry from the section *Truck*, they'll be given the option to include a text value for the field *color*.

Fields also have a use beyond acting as properties for classes. The field object (in this case *color*) is a resource object in it's own right. This



means it can be modified independently of the sections that have assigned it and the entries that are using it. For example, a field which shows a dropdown box of several options could be modified to include more options; any entry which is using that field would automatically receive those new changes. Or consider a simple textfield object called "MAC Address" that is used by several sections and entries. If that field was modified to include a filter that checks the input for a valid MAC string, any entry using that field would get those improved validation checks.

Also, because the same field object can be assigned to multiple sections, it's easier to find entries by their values because they're all using the exact same field object. The alternative would have to be a blind text search to try and find different objects but with contextually similar values, and that method is notoriously unreliable. **This is why it's encouraged to assign the same field object to different sections as opposed to just making new fields each time.**

Fields are like what you might call class properties or class variables, but they've also got a lot more functionality available for when you need it.

## Category

Categories are just an organisational tool. There is a clearly defined relationship between Sections, Entries, and Fields, but Categories exist on their own. If you look on the [Classes page](#), you'll see that every Resource has the same 6 fundamental properties and 3 of them are ID values. The first is the ID that belongs to the resource itself, the second is the ID of the resource's parent, and the third is the ID of the Category that the resource belongs to (if any). There isn't a strict hierarchy here, how you use categories is entirely up to you. You can create categories, child categories, and carefully plan exactly how you want the resources in your system to be organised. Or you can ignore the whole thing completely and just let every resource have the default category of "uncategorized." Many users find that the ability to create hierarchical parent-child relationships with entries, and then filter down results even further by Section, leaves the use of Categories unnecessary. But if you want to use them, it's there.

## Gadgets

Gadgets are not resources, which is why they're not included in the chart at the top of the page. Gadgets are self-contained applications and are limited to only using HTML, CSS, and JavaScript. All they know about the page that they're loaded on is the ID of the resource. However, because gadgets can interact with the API via JavaScript/AJAX, they're the perfect way to add new features to the asset system in a maintainable and modular way. At its core, the asset system just allows users to create entries and then modify their text-based attributes through a simple form. The ability for gadgets (such as the IPAM-Gadget) to interact with the API, is what makes the asset system so powerful.

Currently, the only gadgets that can be assigned to sections are gadgets that have been created by 6Connect. However, our API is robust enough that almost anything you can do through ProVision could be recreated in the form of an isolated gadget. And because they're just made from HTML and JavaScript, it shouldn't be too strenuous for anyone to write a gadget of their own. If you want to create your own gadgets, I would recommend emailing us first with an outline of what you're trying to do. Then my recommended procedure would be to first create it as a standalone HTML/JavaScript webpage that connects to our API (you may need to disable cross-domain request security in your browser to make the AJAX connections work). Once you have your standalone page working, the process to turn that into an embeddable gadget is trivial.

Note: Gadgets are initialized as AngularJS applications. Both the AngularJS and jQuery libraries will be loaded on the page and available to use, but it is highly recommended to make the entire gadget in the form of an AngularJS app. But as noted above, it's best to contact us first so we can help you in the right direction.

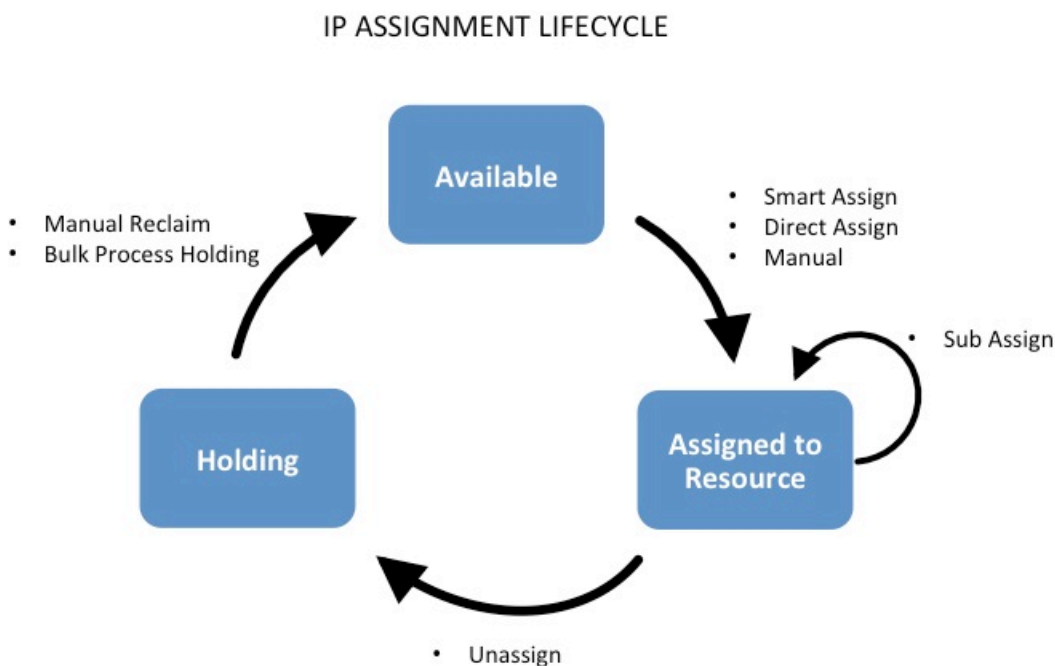
## Workflow Concepts

### Workflow Concepts

- Workflow Concepts
  - IP Assignment Lifecycle
  - IP Management

### IP Assignment Lifecycle

In ProVision, the IP assignment lifecycle starts with an available block. When assigning the block to a resource, there are multiple methods that may be used: Smart Assign, Direct Assign, or Manual Assign. Once assigned to a Resource, blocks can be further subassigned via the same methods. When a block is unassigned, it proceeds into the Holding Tank, where blocks are held until either a set time has elapsed, or until they are manually reclaimed to 'available' status.



For more information on performing tasks in this IP Assignment Lifecycle, see the following documentation sections:

[Working with IP Blocks](#)

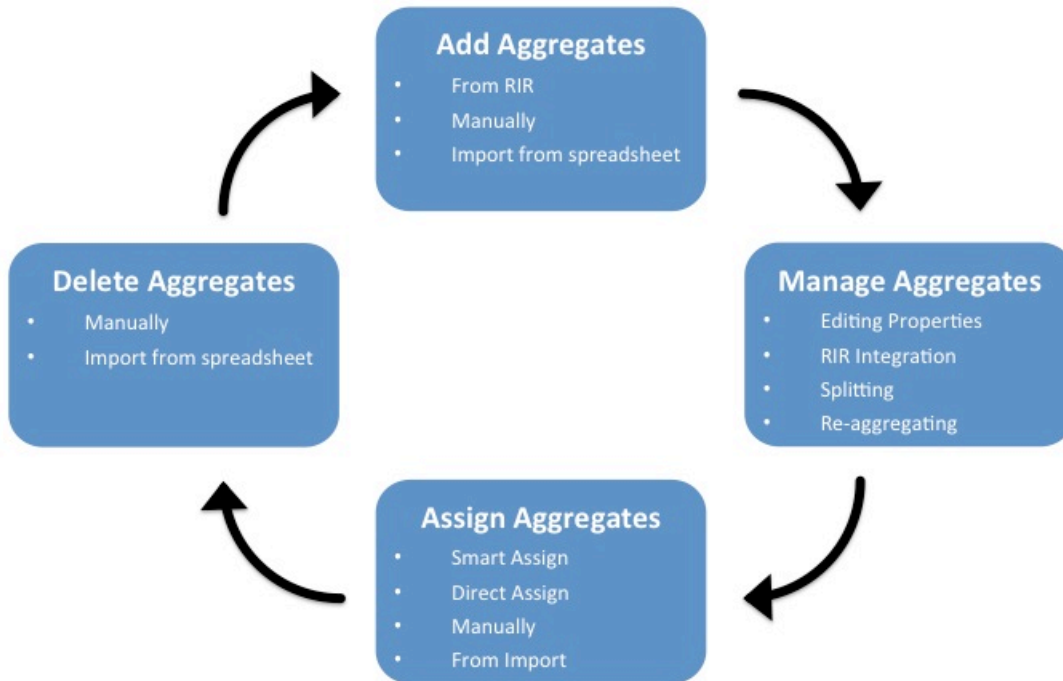
[IPAM Administration](#)

### IP Management

IP Management is comprised of four basic functions: adding aggregates into ProVision, managing those aggregate blocks, assigning them to a resource, and deleting the aggregates.

ProVision provides multiple ways for you to achieve each step, depending on your needs. For example, if your organization currently uses spreadsheet data to track aggregates, ProVision provides tools that can import your existing spreadsheets for bulk updates, saving you time. Need to just quickly assign a single IP? Direct Assign will allow you to do so with just a few clicks.

## IP MANAGEMENT FLOW



For more information on performing tasks in this IP Management Flow, see the following documentation sections:

[Working with IP Blocks](#)

[IPAM Administration](#)

[Importing Your Data](#)

[Import Aggregate Blocks](#)

# UI Element Legend

## Common Icons

While working in ProVision, you will come across a number of icons regularly used to denote status, or with which you can interact to perform tasks. Here is a brief legend to help orient you to the most common icons you'll encounter.

### Interactive Icons:



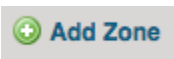
#### Action Menu (Wrench Icon):

The Action Menu is used throughout ProVision to perform actions on individual items. Clicking on the wrench will bring up a menu of tasks specific to that item, such as "Edit", "View", "Delete", "Reassign", and so on.



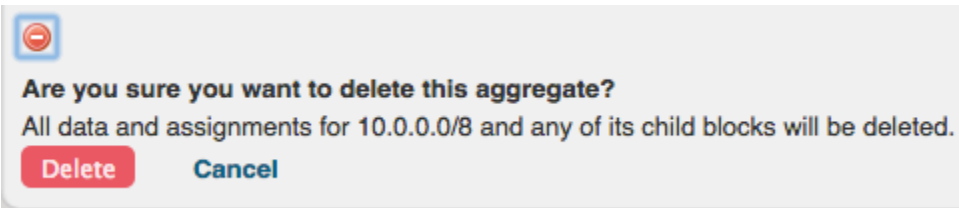
#### "Add" Button

Clicking on the Add button will open a menu to add a new entry to the page, such as adding an aggregate or adding a zone.



#### Red "No Entry" Button

In its interactive state, the red "No Entry" button may be used in ProVision to delete an entry. Clicking on the button will expand a menu with delete confirmation options.



### Status Icons:



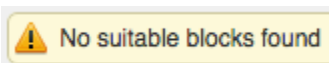
#### Green Check:

The green check indicates a successful result or enabled option. In Peering, it indicates that the entry is a current Peer.



#### Yellow "Warning" Exclamation:

Indicates an unsuccessful result followed by a description.





**Red "No Entry" Button (Status):**

In Peering, the red "No Entry" status indicates a peer that has been marked "Not Qualified".

# ProVision User Guide

## User Guide

The ProVision User Guide provides information on features accessible in the standard user tabs within ProVision. For more detailed information on features accessible with Admin permissions, see the [ProVision Admin Guide](#).

### Table of contents

- The Dashboard
- Working with Resources
- DNS Tab
- DHCP Tab
- IPAM Tab
- Peering v2
- Log
- Reporting

# The Dashboard

## The Dashboard

The Dashboard is your first stop when logging into 6connect Provision, giving you a quick graphical status overview as well as convenient links for reference and support.

### Overview:

#### *IP Charts:*

Illustrates the percentage of assigned vs unassigned hosts for 1918 / IPv4 / IPv6 space out of the total available hosts in ProVision viewable by the user.

#### *Status:*

General status information on whether backup is enabled, number of user / admin accounts, ProVision version number, and a 'Coming Soon' link to the future releases roadmap in the the documentation.

#### *Contact Us:*

Need help, have a question, or would like to provide suggestions on improvements? Contact us through the provided links.

#### *Resource IP Assignments:*

Bar charts illustrating the top 5 Resources that have the most assignments, with the % being the number of assignments for that resource over the total available IPs available in ProVision (that are viewable by the user).

#### *Help and Support:*

Handy links to commonly referenced documentation sections.

#### *Zone Charts:*

Pie charts of Zones with/without DNSSEC, and Zones with/without servers.

#### *Resource Zone Assignments:*

Bar charts illustrating the top 5 Zones that have the most assignments, with the % being the number of assignments for that Zone over the total available in ProVision (that are viewable by the user).





## Working with Resources

### 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.

- **Resource Sections:** 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 Section, you have flexibility in organizing the data. Check out [Customizing Sections](#) and

[Customizing Fields](#) for more details on how to fit these elements to your business.

Resources / Sections			Add Section
			Search Resources...
Name	Entries	Category	
<a href="#">Circuit - LAN</a>	2	<a href="#">Uncategorized</a>	
<a href="#">Circuit - WAN</a>	7	<a href="#">Uncategorized</a>	
<a href="#">Contact</a>	11	<a href="#">Uncategorized</a>	
<a href="#">Customer</a>	3	<a href="#">Uncategorized</a>	
<a href="#">Data Center</a>	1	<a href="#">Uncategorized</a>	
<a href="#">Desktop Server</a>	1	<a href="#">Uncategorized</a>	
<a href="#">Device</a>	60	<a href="#">Uncategorized</a>	
<a href="#">DHCP Servers</a>	1	<a href="#">Uncategorized</a>	
<a href="#">Firewall</a>	6	<a href="#">Uncategorized</a>	
<a href="#">Host</a>	0	<a href="#">Uncategorized</a>	
<a href="#">LIR</a>	2	<a href="#">Uncategorized</a>	
<a href="#">Load Balancer</a>	1	<a href="#">Uncategorized</a>	
<a href="#">Location</a>	44	<a href="#">Uncategorized</a>	
<a href="#">Physical Interface</a>	13	<a href="#">Uncategorized</a>	
<a href="#">Rack</a>	25	<a href="#">Uncategorized</a>	
<a href="#">Resource Holder</a>	138	<a href="#">Customer</a>	

- **Resource Categories:** Categories can be used to create some filtered views for given Resources and Sections. For example, you can create a Section called "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 Section called "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 to customize Sections?** Check out [Customizing Sections](#) and [Customizing Fields](#) for more details!

### Some examples:

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

Learn more about working with Resources through the following links:

- Customizing Sections
- Customizing Fields
- Gadgets
  - XML Specifications

## Customizing Sections

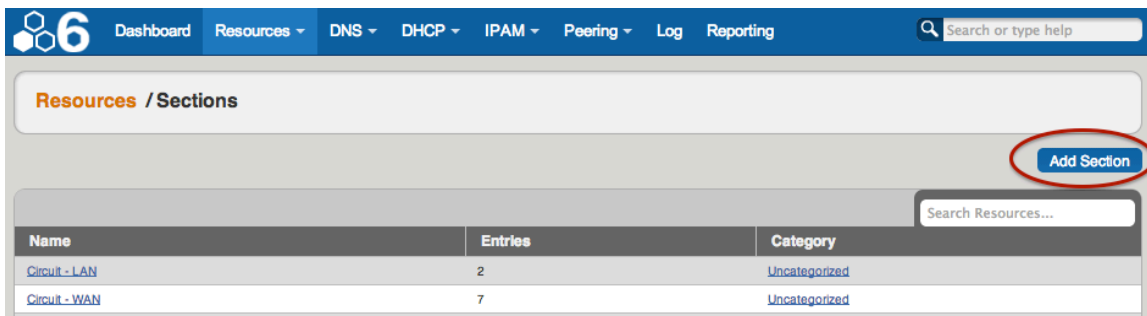
### Customizing Sections

You can create as many Sections as you wish (Firewall, Server, VM, Virtual Interface, etc.) and customize the fields that you care about for each Section. 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" Section. This way you can still track the console port for your physical firewalls like normal.

- Customizing Sections
  - Step 1: Create a New Section
  - Step 2: Add a Custom Field to a Section
  - Step 3: Edit Custom Field Data
  - Step 4: Add Gadgets to your Section

#### Step 1: Create a New Section

Click "Add Section" from the **Sections** sub-tab under the **Resources** Tab

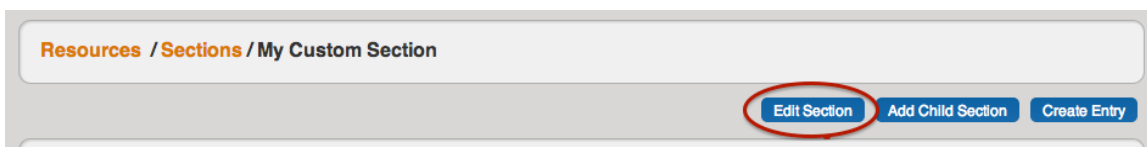


Create a new Section by specifying a Name, Parent, and Category

The screenshot shows the 'Create Section' form. The breadcrumb is 'Resources / Sections / Create Section'. Under the 'Fundamentals' section, there are three fields: 'Name (required)' with the value 'My Custom Section', 'Parent' with a dropdown menu showing 'TLR', and 'Category' with a dropdown menu showing 'Uncategorized'. A 'Create' button is located at the bottom right of the form.

#### Step 2: Add a Custom Field to a Section

Manage existing fields and add custom fields for the selected Section by clicking "Edit Section"



Add existing or [Customizing Fields](#) for your Section. You can add new [Customizing Fields](#) of different types (text, dropdown, text area) by dragging and dropping the fields as well as use any existing fields that are available. See the [Customizing Fields](#) page for more details.

### Step 3: Edit Customize Field Data

Select the field name and you will get an editing window to modify the parameters of the field. Custom fields may be renamed and have other attributes updated, whereas protected system fields may have noted restrictions.

### Step 4: Add Gadgets to your Section

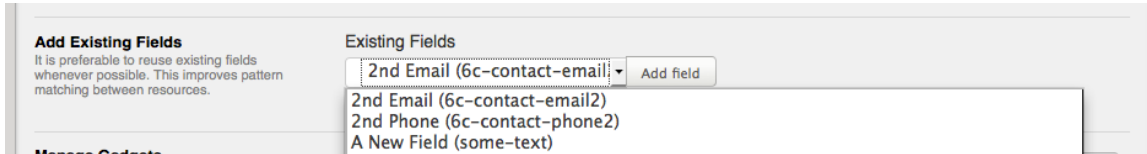
You will notice on this customization screen, you also have an area for [Gadgets](#). Gadgets are areas of additional functionality that can be added to the UI of a given Resource. Simply select the Gadget(s) you want to show for that section, hit "Add", then organize in the order you wish to view. Once added to the Section, Gadgets will be visible for all Resources of that Section.

## Customizing Fields

### Working with Fields

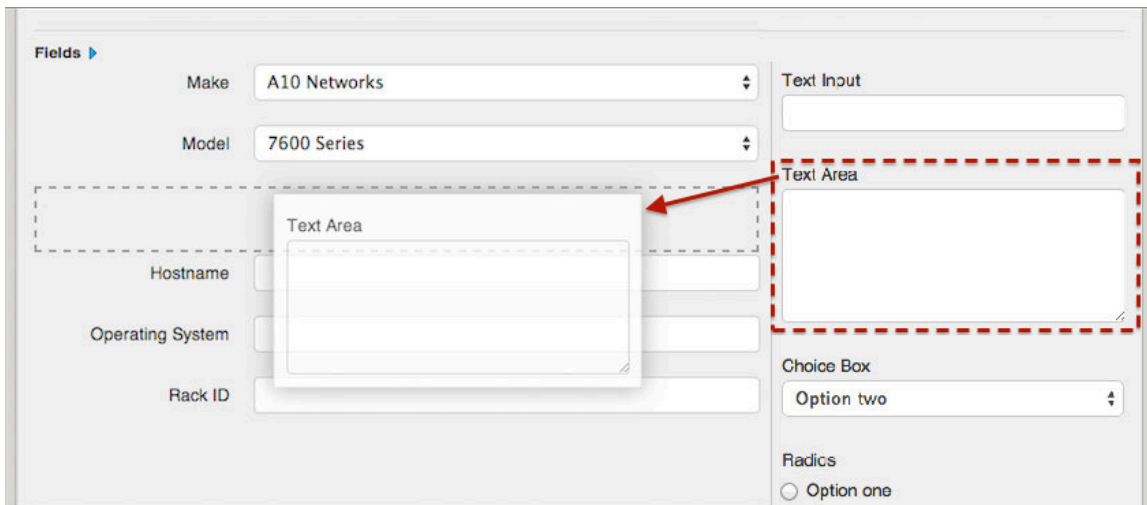
#### Creating Fields

To add an existing field to a Section, select the field name from the dropdown menu and click on the "Add Field" button.



The screenshot shows a dialog box titled "Add Existing Fields" with a note: "It is preferable to reuse existing fields whenever possible. This improves pattern matching between resources." Below the note is a section labeled "Existing Fields" containing a dropdown menu with the following options: "2nd Email (6c-contact-email)", "2nd Email (6c-contact-email2)", "2nd Phone (6c-contact-phone2)", and "A New Field (some-text)". To the right of the dropdown is an "Add field" button.

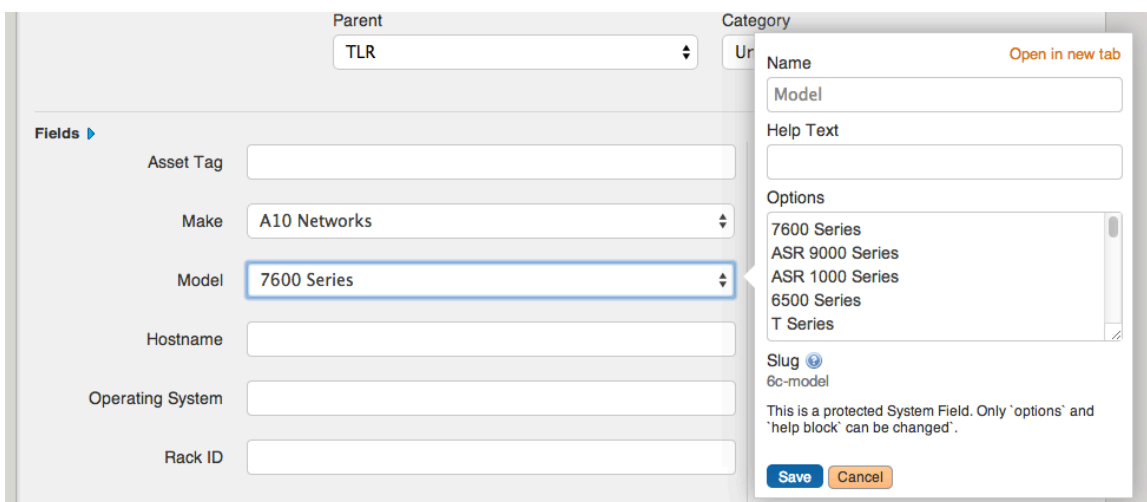
To add a new custom field to a Section, simply click on the custom field type name (Text Input, Text Area, Choice Box, etc), then drag the field over to the field list and release in the desired location. Edit the field name and options as described in Editing / Removing Fields.



The screenshot shows the "Fields" section of the interface. On the left, there is a list of fields: "Make" (A10 Networks), "Model" (7600 Series), "Hostname", "Operating System", and "Rack ID". A dashed box highlights the "Text Area" field type, which is being dragged over the "Model" field. On the right, there is a panel with various field types: "Text Input", "Text Area", "Choice Box" (with "Option two" selected), and "Radios" (with "Option one" selected).

#### Editing/Removing Fields

Once fields are added to a Section, you can click on the field name to make additional changes to the fields. Custom fields may be renamed and have other attributes updated, whereas protected System Fields may have noted restrictions.



The screenshot shows the "Fields" section with a modal open for editing the "Model" field. The modal has a "Parent" dropdown set to "TLR" and a "Category" dropdown set to "Un". The modal contains the following fields: "Name" (Model), "Help Text", "Options" (a list of options: "7600 Series", "ASR 9000 Series", "ASR 1000 Series", "6500 Series", "T Series"), "Slug" (6c-model), and a note: "This is a protected System Field. Only 'options' and 'help block' can be changed". At the bottom of the modal are "Save" and "Cancel" buttons.

To rearrange the field list order, click and hold on the field name, then drag and drop into the preferred order.

To remove a field, click and hold on the field name, then simply drag and drop the field to the right side of the screen to where the "Remove Field" prompt is visible.

## Gadgets

### Gadgets

- Gadgets
  - What are Gadgets?
  - Available Gadgets
    - Resource View
    - Contact Info
    - Tech Info
    - IPAM
    - DNS
    - DHCP
    - Peering Session
    - Peer Groups
    - Peering VRF
  - Creating your own Gadgets

### What are Gadgets?

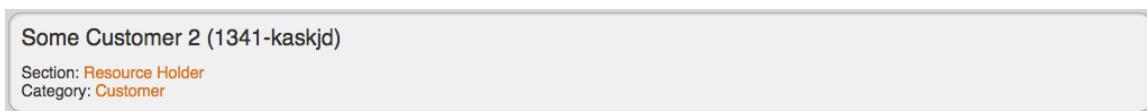
Our gadget system is similar to the Atlassian Gadget system (and Google Gadgets). When creating or editing a Section, gadgets can be added in a way similar to how you would add or remove a field. Gadgets are best described as self contained webapps; widgets but with more power.

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.

### Available Gadgets

#### *Resource View*

This visual element is used on the Resource Holder Section type. The Resource view displays and provides links for the Section and Category for the Resource.



#### *Contact Info*

This visual element is used on the Resource Holder Section type. In the Contact Info Gadget, you can track information such as mailing / billing addresses, phone number, and fax number for that Resource.



#### *Tech Info*

This visual element is used on the Resource Holder Section type. This Gadget allows you to list DNS servers, ARIN information, and enable/disable customer privacy.

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

Org ID: ARIN-ORGNAME

Org POC: ARIN-POC1

Net POC: ARIN-POC2

Abuse POC:

Origin AS: 23456

Residential Customer Privacy: Disabled

## IPAM

This visual element is used on the Resource Holder Section type. IPAM Gadget allows you to view, assign, and manage blocks for that resource. For more information on assigning and managing blocks, see [Working with IP Blocks - Assigning IP Space](#).

IPAM

Assign Block:

Browse To Assign

List available blocks

Direct Assign

x.x.x.x/yy or x:x:x:x:x:x/yyy

Assign

Smart Assign

IPv4

Size

RIR

Region

Tags...

Tag selection mode:

☒ Standard – match all selected tags
 ☐ Strict – match exactly the selected tags
 ☐ Exclude – match blocks not tagged with any selected tags

Show advanced options

Smart Assign

Filter:

Notes/CIDR...

RIR

Region

All Masks

6connect Labz

Tags...

Filter

Clear

Address	Hosts	LIR	Region	Notes	Tags	Assigned	Updated
10.0.0.2/32	1		Los Angeles, CA			2014-07-21	2014-11-24
10.0.0.3/32	1		Los Angeles, CA		Anycast,PTP	2014-07-21	2014-07-21

## DNS

This visual element is used on the Resource Holder Section type. The DNS Gadget allows you to add new Zones as well as view and manage existing zones. For more information on DNS functions and managing zones, refer to the documentation for the [DNS Tab](#).

DNS

New DNS Zone

Create Zone

Zone Delegation

Delegated Zone

Slave IP

Customer

Zone name

IPv4 or IPv6

84

Add Slave

Zone Records

Tags

Entries

test-domain.com

5

## DHCP

This visual element is used on the Server Section type.

The DHCP Management Gadget in the "Off" configuration:

DHCP Management

DHCP Services:
☒ Off
☐ On

Update

The DHCP Management Gadget in the "On" configuration:

DHCP Management

+ Connection Configuration

+ Server Details

+ DHCP Pools

+ Create a New DHCP Pool

Save Configuration

Push Configuration

Peering Session

This visual element is used on the Router Section type. In Peering Sessions Gadget, by clicking on the Action Menu (wrench icon) you can perform basic session edit functions such as Edit, Config Manager, Email, Admin Up/ Down, and Delete. For additional information on Peering, see [Peering v2](#).

Peering Sessions

Exchange	Source	Peer	Destination	Type	Max Prefixes	Prefixes Rcvd	State	Notes
Equinix Palo Alto	AS8038 – 50.240.195.137	Amazon.com	AS16509 – 198.32.176.36	Peer	0		inactive	
Equinix Palo Alto	AS8038 – 50.240.195.137	AwesomeCo	AS2137 – 1.2.3.1	Unknown			Idle	

Peer Groups

The Peer Group Gadget allows you to add peer groups for IPv4 and IPv6 for a selected exchange from a router's Resource Entry page.

Peer Groups

Add Peer Group:

Select Exchange...

☒ IPv4
☐ IPv6

Add Group

Exchange

Peer Group

Type

To do this, simply select the exchange, type in a Peer Group name in the text box, select IPv4 or IPv6, the click "Add Group".

Peer Groups added from this gadget will be then be available to select in the "Add Session" dialog box in the [Peering](#) tab.

Peer Groups

Peer group added: Peer Group 2 - Equinix Palo Alto (ipv4)

×

Add Peer Group:

Equinix Internet Exchange Palo Alto

☒ IPv4
☐ IPv6

Add Group

Exchange

Peer Group

Type

Equinix Palo Alto

Peer Group 2

ipv4

delete

Note

Peer groups listed in the Gadget are for ProVision only and should reflect groups that exist on the router.

Adding or deleting peer groups from the Gadget will not add or delete them on the router.

For additional information on Peering, see [Peering v2](#).

Peering VRF

The Peering VRF Gadget allows you to add VRFs from a router's Resource Entry page.

Enabling "VRF Support" in the Admin home page under "Peering Settings" will automatically add the VRF gadget to the router Section.

44



**Peering Settings**

ASN:   
Numbers only. For multiple ASNs, use a comma-separated list, e.g. 1234,5678

VRF Support: ☒

[Update](#)

The VRF gadget will then be accessible in a router's Resource Entry page.

**VRFs**

[Add VRF](#)

VRF Name	ASN
VRF1	2345

NOTE: Peer VRFs listed here are for ProVision only and should reflect VRFs that exist on the router. Adding or deleting VRFs here will not add or delete them on the router.

To add a VRF, type the VRF name and ASN, then hit "Add VRF".

**VRFs**

✓ VRF added: VRF1 - AS2345

[Add VRF](#)

VRF Name	ASN	
VRF1	2345	<a href="#">delete</a>

NOTE: Peer VRFs listed here are for ProVision only and should reflect VRFs that exist on the router. Adding or deleting VRFs here will not add or delete them on the router.

To delete a VRF, click on "delete" next to the VRF entry in the gadget.

Once VRFs are set up for a router, the source ASNs for the associated VRFs will appear in the Source ASN dropdown when adding or editing a session for that router from the [Peering](#) tab.

**Add Session**

Type:  Peer Group:

Exchange:  MD5:

Note:

Max Prefixes:

**Source**

Router:

IP Address:

ASN:  (Dropdown menu shows: AS8038, AS20940, AS2345 - VRF1, AS5678 - VRF2)

**Destination**

Select peer and public IP data PeeringDB or specify custom data for the session.

Peer:

Public IP:

Peer:

ASN:

IP Address:

☐ Configure router after saving? [Save](#)

Peering VRF currently only supports Cisco routers.

## Creating your own Gadgets

6connect provides XML specifications for users interested in creating their own gadgets for ProVision. See the XML Specifications section linked below for more information.



User created gadgets are not supported at this time and the specification below could change without notice. If you want to make your own gadget, please get in touch so we can help you

- [XML Specifications](#)

## XML Specifications

### XML Specifications

#### THIS IS AN EXPERIMENTAL FEATURE

User created gadgets are not supported at this time and the specification below could change without notice. If you want to make your own gadget, please get in touch so we can help you.

- [XML Specifications](#)
  - [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.

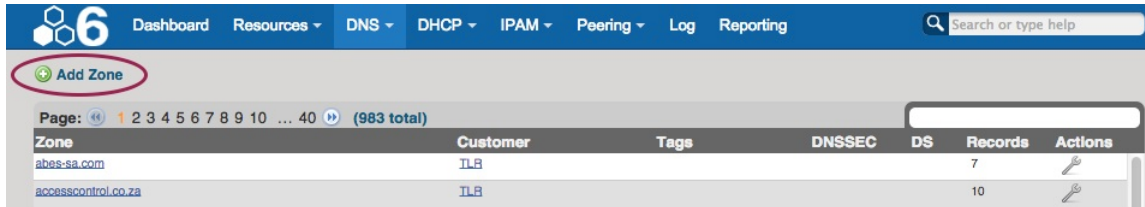
## DNS Tab

### DNS

The DNS tab allows you to add new Zones as well as view and manage existing zones.

### Navigating to the DNS Tab

Clicking on the main DNS Tab, then on "Add Zone" 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.

The form is titled 'Create a DNS Zone'. It has two input fields: 'Enter Zone Name' and 'Select Site...'. To the right of the 'Select Site...' field is a green plus icon.

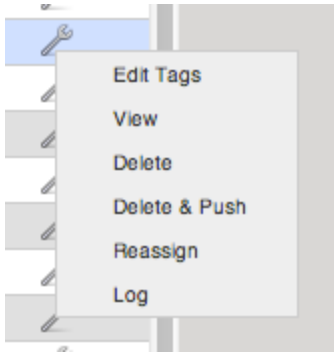
### DNS Tab User Interface

The screenshot shows the DNS Tab interface with numbered callouts 1 through 7. Callout 1 points to the pagination bar. Callout 2 points to the search bar. Callout 3 points to the 'Zone' column. Callout 4 points to the 'Customer' column. Callout 5 points to the 'Tags' column. Callout 6 points to the 'DNSSEC' column. Callout 7 points to the 'Records' column.

Zone	Customer	Tags	DNSSEC	DS	Records	Actions
6clabs.com	6connect Labs				12	
6connect.com	6connect Available				7	
aaron.com	123 Department LAB		DNSSEC	X	11	
anna.com	123 Department LAB				12	
awesome.com	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) The **Customer** 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 is 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

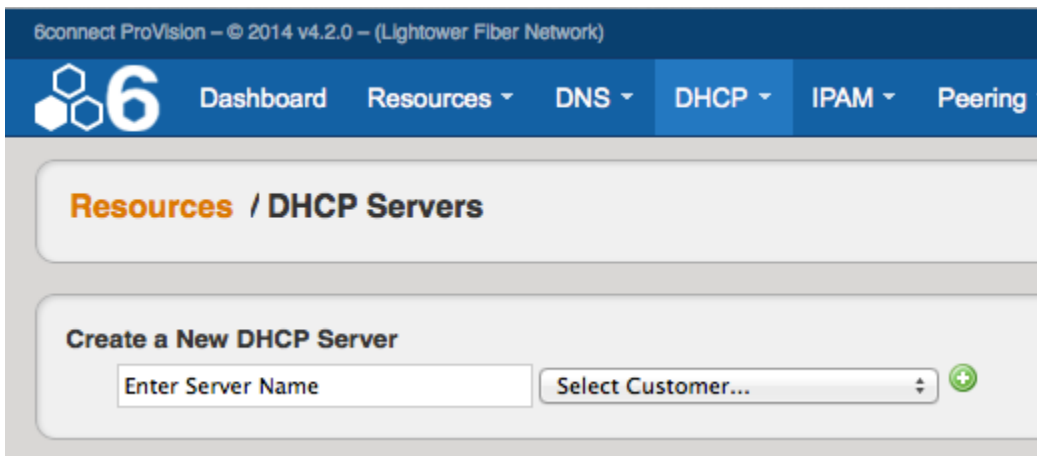
## DHCP Tab

### DHCP

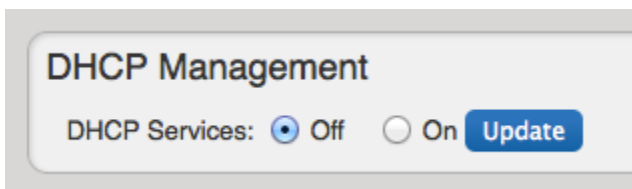
- DHCP
  - Adding DHCP Servers
  - Defining DHCP Scopes
  - Managing DHCP Server Configurations
    - Connection Configuration
    - Server Details
    - DHCP Pools
    - Create a New DHCP Pool - Subnets
    - Create a New DHCP Pool - Host
    - Saving/Pushing DHCP Server Configurations

## Adding DHCP Servers

DHCP Server Configuration is tied into the Resource Manager. To add a DHCP Server to ProVision, you can either use the "Create a New DHCP Server" dialog area from the DHCP Tab.



If DHCP is enabled for a Resource Section, you can also use the DHCP toggle function to enable this functionality.



## Defining DHCP Scopes

In order to use DHCP functions and add DHCP Pools, the IP blocks need to be defined in the IPAM section.

For a new aggregate, simply add it using the correct checkbox option as outlined below (Use the Add Aggregate function on the IPAM Tab). This will ensure the block is automatically added to the DHCP Available Resource and thus usable when building DHCP Server Configurations and defining DHCP Pools.

6 Dashboard Resources DNS DHCP IPAM Peering Log Reporting Search or type help

Advanced Search IPAM...

Subnet: x.x.x.x/yw OR xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx  
 RIR: RIR  
 VLAN:   
 Tags:   
 Region: Region  
 Resource: Available

☒ DHCP Aggregate ☐ Allow Sub-Assignment

Add Aggregate

If you would like to use an existing aggregate or part of an existing aggregate, you simply need to "assign" the block (Action Menu) to the Resource Holder "DHCP Available" as shown below.

Export Current List To

Assign block Filter by type: Resource Holder

192.64.0	07-10
192.65.0	07-10
192.65.4	05-13
192.65.8	07-10
192.65.1	07-10
192.65.1	07-10
192.65.1	07-10
192.65.2	07-10
192.65.24/30	2014-08-06

DHCP Available (Resource Holder)

Assign Block Cancel

Once the IP block is assigned to DHCP Available, it will be available to assign to a DHCP Pool via the DHCP Gadget.

## Managing DHCP Server Configurations

Once DHCP functions are enabled for a Resource Section/Type, you will be able to manage configurations per Resource as expected by expanding the relevant areas.

DHCP Management

- + Connection Configuration
- + Server Details
- + DHCP Pools
- + Create a New DHCP Pool

Save Configuration Push Configuration

### Connection Configuration

In this gadget area, this is where you enter in the information that will be used for ProVision to communicate to the DHCP Server.

+ Connection Configuration

Manual IP: 192.34.13.141

Notes: some server - testing

SSH

Username: 6connect

Password: .....

Port: 22

## Server Details

**+ Server Details**

DHCP Vendor:

DHCP Config File Path:

**Server Options**

Routers:

Domain Name Servers:

Domain Name:

**Free Lines (appended to DHCP Server Config):**

No lines saved.

Add a new Line:

**Server Commands**

Config Test:

Server Stop:

Server Start:

**Advanced Options**

Authoritative:

Default Lease Time:

Max Lease Time:

Local Port:

Log Facility:

## DHCP Pools

In this areas, the admin can specify what DHCP Pools are linked to the DHCP server. This includes any host reservations as well as DHCP Pools as defined in the next section.

**+ DHCP Pools**

**Linked Pools**

Internal Lab - VM [Subnet]

**Existing Pools**

Internal Lab - VM [Subnet]

LargerTest [Subnet]

TsetNet [Subnet]

Use the Action menu to make changes to Linked or Existing Pools.

Link to Server

Delete Pool

## Create a New DHCP Pool - Subnets

When Assigning a Subnet (via dropdown) the IP Assignment selection will pull the data from the DHCP Available blocks that you defined earlier. You can use either a Smart or Direct assignment depending on your preference.

**+ Create a New DHCP Pool**

Create a new

Subnet Name:

New IP Assignment:

Free Lines:

No lines saved.

Add a New Line:

## Create a New DHCP Pool - Host

When reserving Hostname/MAC data, change the Dropdown to "Host". This will also give you an option to assign from an existing DHCP block, or a specific IP address.



### + Create a New DHCP Pool

Create a new

Host

Hostname:

(ex: 6connect.com)

MAC Address:

(ex: 00:11:22:33:44:55:66:77)

New IP Assignmnet:

Smart

IPv4

RIR

Region

Free Lines:

No lines saved.

Add a New Line:

Add

Add Pool

### Saving/Pushing DHCP Server Configurations

It is recommended that you Save your Configuration. When you Push a Configuration the configuration is automatically saved.

Save Configuration

Push Configuration

# IPAM Tab

## IPAM

The **IPAM** tab provides a listing of aggregate blocks and tools to add and manage aggregates.

**Aggregate Blocks**

All - IPv4 - IPv6 - DHCP

Type to filter aggregate blocks

- 10.0.0.0/8
- 10.0.0.0/8
- 10.1.0.0/24
- 10.2.3.0/24
- 10.4.0.0/24
- 10.5.3.2/32
- 11.1.1.0/24
- 13.0.0.0/24

**10.0.0.0/8 – 1918** Manage

Los Angeles, CA – Miami, FL – – – NYC2

Hosts	16777216	Blocks	49	Resources	0.00% assigned		
Available	16776430	99.99%	Available	43	87.75%	6connect Available	99.99%
Assigned	785	0.00%	Assigned	5	10.20%	BitBandits, Inc.	0.00%
Holding	1	0.00%	Holding	1	2.04%	Glenlivet	0.00%
Allocated	791	0.00%	Allocated	11	22.44%	7connect	0.00%
						6connect Labz	0.00%

Filter By Mask: All Masks

Go To Reporting

## UI Elements:

### Add Aggregate:

Opens a menu to add an aggregate block with options for RIR, VLAN, Tags, Region, Resource, and enabling Sub-Assignments / DHCP Aggregate.

**Subnet** x.x.x.x/yy OR xxxx:xxxx:xxxx:xxxx:

**RIR** RIR

**VLAN**

**Tags**

**Region** Region

**Resource** 123 Department

☐ DHCP Aggregate ☐ Allow Sub-Assignment

**Add Aggregate**

### "Advanced" Button:

Opens the IPAM Manage screen for all blocks. See [Working with IP Blocks - Architecting IP Address Blocks](#) for more information on working in IPAM Manage.

### Aggregate Blocks List:

Provides a searchable listing of all aggregate blocks in the left sidebar. Selecting "All / IPv4 / IPv6 / DHCP" will filter the visible aggregates in the center of the page.

### Top Level Aggregate Box:

Provides detailed information on that aggregate, including percentage breakdowns and the top 5 Resources assigned under that aggregate.

"Manage" opens the IPAM Manage screen for blocks under that aggregate. See [Working with IP Blocks - Architecting IP Address Blocks](#) for more information on working in IPAM Manage.

"Filter by Mask" provides a way to further filter percent assigned and percent allocated by mask.

"Go To Reporting" provides a shortcut to the Reporting tab.

The red icon provides the option to delete the aggregate.

Are you sure you want to delete this aggregate?

All data and assignments for 10.0.0.0/8 and any of its child blocks will be deleted.

**Delete** **Cancel**

## Working with IP Blocks

For additional information on performing IPAM tasks and working with blocks, see the following section:

- [Working with IP Blocks](#)

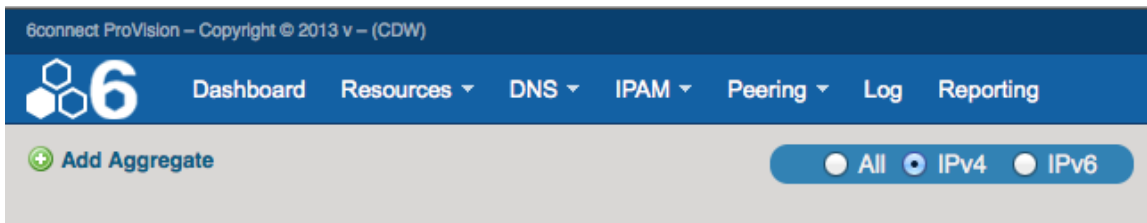
## Working with IP Blocks

### Working with IP Blocks

- Working with IP Blocks
  - Adding/Deleting IP Address Aggregates
  - Architecting IP Address Blocks
    - Splitting/Aggregating blocks manually
    - Splitting/Aggregating blocks with Templates
    - IP Block parameters and Editing Attributes
      - Edit Attributes Overview:
  - Assigning IP Space
    - Assigning Space from the IPAM Gadget
      - Browse to Assign
      - Direct Assign
      - Smart Assign
    - Manually Assigning Space from the IPAM Manager
  - Sub Assigning IP Space
  - Unassigning IP Space

## Adding/Deleting IP Address Aggregates

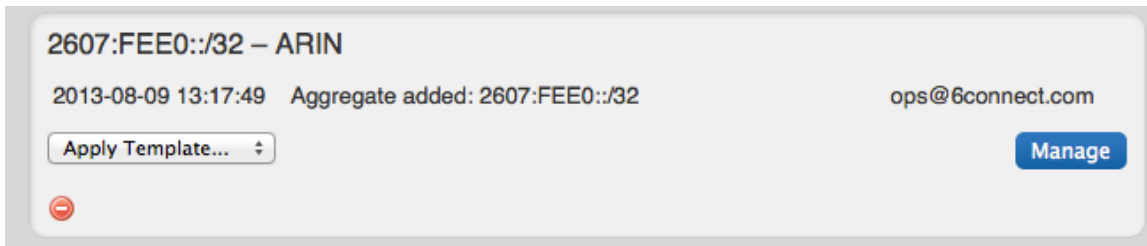
On the standard IPAM page there is an option to "Add Aggregate". Click on the green "Add" icon.



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

The screenshot shows a form for adding a new IP aggregate. It has several input fields: 'Subnet' with a placeholder 'x.x.x.x/yy OR xxxx:xxxx:xxxx:xxxx:', 'RIR' with a dropdown menu, 'VLAN' with an empty text box, 'Tags' with an empty text box, 'Region' with a dropdown menu, and 'Resource' with a dropdown menu showing '123 Department'. There are two checkboxes: 'DHCP Aggregate' and 'Allow Sub-Assignment'. At the bottom left of the form is an orange 'Add Aggregate' button.

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



To delete the aggregate - press the red icon and you will have the option delete the aggregate.



#### Requirements to Delete an IP Aggregate

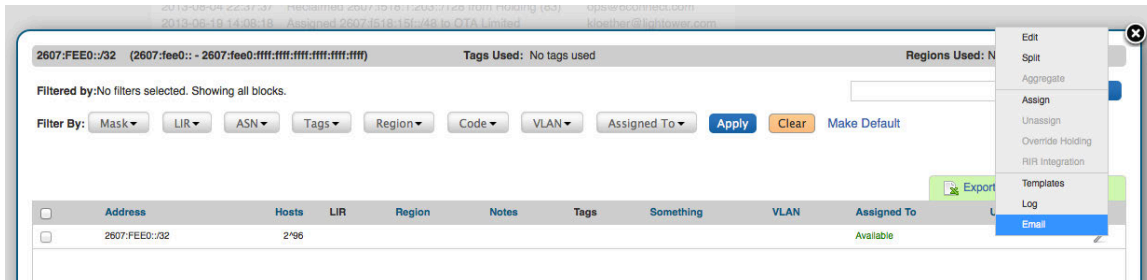
In order to delete an IP Aggregate, all resources need to be "unassigned". Once they are unassigned from their respective resources, the "Apply Template" drop down will permit the function "Aggregate" which will bring the IP block back to it's original size.

Once the block is back to it's original size and there are no subnets assigned, the IP Aggregate can be deleted.

## Architecting IP Address Blocks

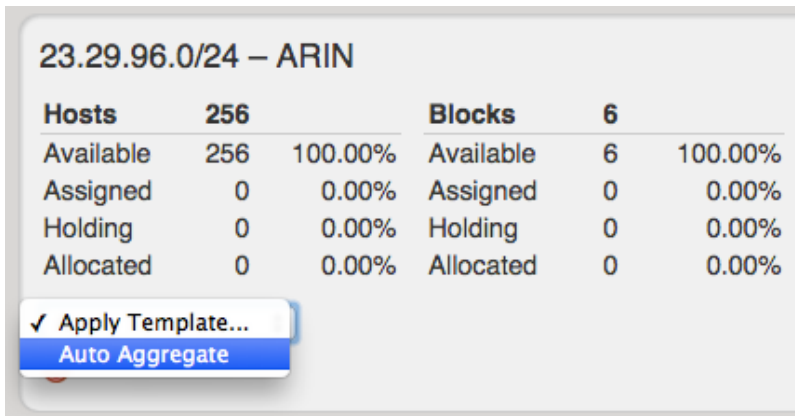
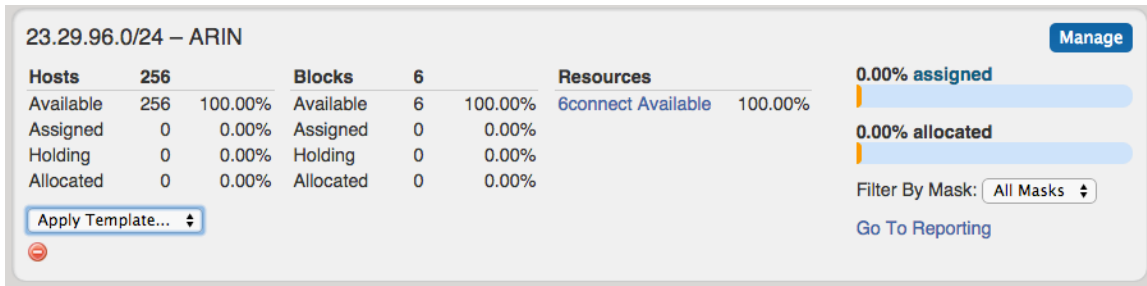
### Splitting/Aggregating blocks manually

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

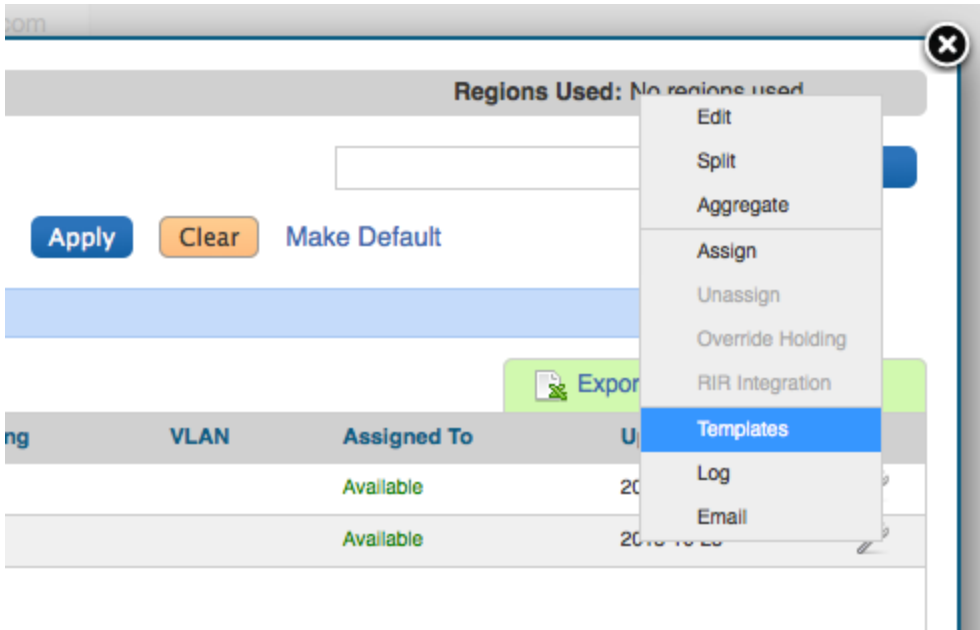


### Splitting/Aggregating blocks with Templates

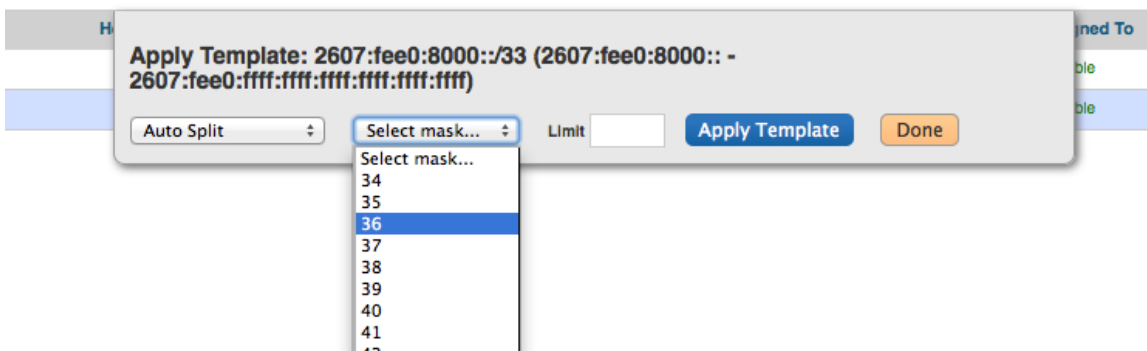
When you first import a block, you can select the template to use from the main IPAM page.



You can also use the "Templates" option from the Action Menu on the IPAM Manage screen for the specific block.



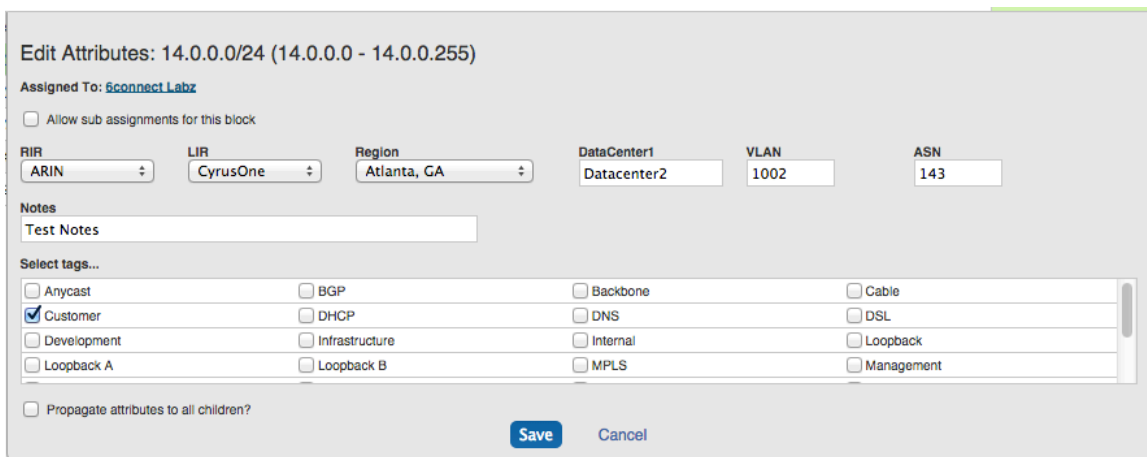
Then, select the auto split parameters from there, and hit "Apply Template".



### IP Block parameters and Editing Attributes

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

From here you can set a variety of attributes for a given block. These values are also customizable from the Admin screen - [IPAM Admin](#). For more information on IPAM management, see [IPAM Administration](#) and [IPAM Parameters](#).



### Edit Attributes Overview:

**Allow Subassignments:** When editing a block that has been assigned, checking this box allows for further subassignments, indicated by a blue arrow next to the assignment in the Manage screen. Note: Subassign status cannot be changed if a block has children.

**RIR / LIR / Region:** Select the information from the drop down menus. LIR and Regions can be customized in the IPAM Admin section of ProVision - see [IPAM Administration](#) and [IPAM Parameters](#).

**Generic Code (Here, DataCenter1):** This is a customizable text field that can be used to track information specific to your needs. It can be filtered in the IPAM Manage screen. The header, display, and enabling settings for this field are set under IPAM Configuration in the [IPAM Administration](#) section.

**VLAN:** Numerical VLAN information for the block. Settings to enable this field are set under IPAM Configuration in the [IPAM Administration](#) section.

**ASN:** ASN information for the block.

**Notes:** Freeform text field for additional information you wish to capture.

**Tags:** Tags can be set under Edit Tags in the [IPAM Administration](#) section.

**Propagate Attributes to Children:** Select this box when editing a parent block to carry through attribute changes to all children of that block. To view parent blocks, simply ensure that top level or all masks are selected in the Filter menu in the IPAM Manage screen.

Note: The VLAN of a child cannot be different from that of its parent, so for multi-level situations (Parent -> Child -> Grandchild), VLAN should be updated at the top tier parent level.

After editing the desired attributes for the block, simply hit "Save".

### Assigning IP Space

There are two areas where you can assign IP Space: in the IPAM Gadget for the particular Resource, or through IPAM Manage for manually assigning a block to a resource. The IPAM Gadget allows for more detailed assignment options including Browse to Assign, Direct Assign, and Smart Assign with advanced options, and is the primary tool for space assignment.

#### Assigning Space from the IPAM Gadget

The IPAM Gadget is accessed from a Resource Entry page, once enabled for the Section (to add Gadgets, see [Customizing Sections](#) and [Add Gadgets to your Section](#)).

The screenshot shows the IPAM gadget interface. At the top, there's a section for "Assign Block:" with two options: "Browse To Assign" (with a link "List available blocks") and "Direct Assign" (with a text input field for "x.x.x.x/yy or x:x:x:x:x:x/yyy" and an "Assign" button). Below this is the "Smart Assign" section, which includes dropdowns for "IPv4", "Size", "RIR", and "Region", a "Tags..." input field, and a "Tag selection mode:" section with three radio buttons: "Standard - match all selected tags" (selected), "Strict - match exactly the selected tags", and "Exclude - match blocks not tagged with any selected tags". There's a "Show advanced options" link and a "Smart Assign" button. At the bottom, there's a "Filter:" section with a "Notes/CIDR..." input field, dropdowns for "RIR", "Region", and "All Masks", a "6connect Labz" dropdown, a "Tags..." input field, and "Filter" and "Clear" buttons. Below the filter section is a table with the following data:

Address	Hosts	LIR	Region	Notes	Tags	Assigned	Updated
10.0.0.2/32	1		Los Angeles, CA			2014-07-21	2014-11-24
10.0.0.3/32	1		Los Angeles, CA		Anycast,PTP	2014-07-21	2014-07-21

You have three options for assigning IP space using the IPAM Gadget:

#### Browse to Assign

This brings up a list of IP aggregates where you can select the block to assign.

## Direct Assign

This field allows you to manually enter an IP block to assign. Enter an IPv4 or IPv6 block, and then click "Assign".

## Smart Assign

This series of dropdowns allows you to specify the parameters for the type of IP block you want to assign, as well as tag selection modes. Then it will look at the IPAM blocks that match your criteria to find the correct IP assignment based on availability and relevant parameters.

Additional advanced Smart Assign options are available under "Advanced Options", including VLAN and LIR.

Hide advanced options

DataCenter1 ▾ VLAN  LIR ▾ Assigned Resource ▾

**Note:** The Assigned Resource option will filter by blocks that are already assigned to the selected resource and are set to Allow Subassignments. If no resource is selected, the filter will default to Available blocks.

**Smart Assign**

Once your criteria has been set, click the "Smart Assign" button.

## Manually Assigning Space from the IPAM Manager

You can also assign blocks manually using the "Assign" function from the IPAM Manager screen (accessible from the IPAM Tab). Click the Action Menu (wrench icon), then select "Assign".

10.0.0.0/8 (10.0.0.0 - 10.255.255.255) Tags Used: Anycast, Cable, Customer, PTP Regions Used: Boston, MA, LAX, MIA, NYC

Filtered by: No filters selected. Showing all blocks.

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	DataCenter1	VLAN	Assigned To	
<input type="checkbox"/>	10.0.0.0/32	1		Los Angeles, CA		PTP			Amazon Server	
<input type="checkbox"/>	10.0.0.1/32	1		Miami, FL		PTP			8connect Labz	
<input type="checkbox"/>	10.0.0.2/32	1		Los Angeles, CA					8connect Labz	
<input type="checkbox"/>	10.0.0.3/32	1		Los Angeles, CA		Anycast, PTP			8connect Labz	
<input type="checkbox"/>	10.0.0.4/32	1		Los Angeles, CA					8connect Labz	
<input type="checkbox"/>	10.0.0.5/32	1		Los Angeles, CA		Anycast, PTP			8connect Labz	
<input type="checkbox"/>	10.0.0.6/32	1		Miami, FL		Anycast, Customer, PTP			8connect Labz	
<input type="checkbox"/>	10.0.0.7/32	1		Miami, FL		Anycast, Customer, PTP			8connect Labz	
<input type="checkbox"/>	10.0.0.8/29	8		Los Angeles, CA		Anycast, PTP			Available	
<input type="checkbox"/>	10.0.0.16/28	16		Los Angeles, CA		Anycast, PTP			Holding	

Then, select the Resource to assign the block. A filter tool is provided to narrow the list to a particular Section type.

**Assign block** Filter by type: All Types ▾

**Export Current List To CSV**

007 Manufacturing (Resource Holder)	Firewall	Reverse DNS for HVDN Prefixes (Resource Holder)	123 Enterprises (Resource Holder)	33rd St. Bistro (Resource Holder)
44 Magnum Beats (Resource Holder)	LIR	(Resource Holder)	6c-009 (Resource Holder)	6c-026 (Resource Holder)
6c-033 (Resource Holder)	Rack	(Resource Holder)	9-All Nine's (Resource Holder)	32ND STREET YOUNG MENS AND YOUNG
A + Technology Solutions	Router	of New York (Resource Holder)	ABCAM (Resource Holder)	Acadian Asset Management LLC (Resource Holder)
Acceleron Pharma, Inc.	Server	Plus Communications Inc (Resource Holder)	Acronis, Inc. (Resource Holder)	ACS Consultant Company (Resource Holder)
ADA Investment Management	Storage Array	Inc. (Resource Holder)	Adelphi University (Resource Holder)	ADTRAN, Inc. (Resource Holder)
Advanced Instruments (Resource Holder)	Storage Controller	Aer Lingus (Resource Holder)	Affiliated Pathology Service (Resource Holder)	
Advanced Physician Services, PC (Resource Holder)	Switch			
	Virtual Machine			

**Assign Block** **Cancel**

After assigning, you can further edit the block attributes or subassign space.

## 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.

To allow sub assignments for multiple blocks at once, open the Manage screen for the aggregate. Then, select the desired blocks and click "Edit Selected Blocks". The Multi-block edit interface will open. In that interface, select the check boxes next to "Allow sub assignments for this block" and the "Update field" below it. Lastly, save your changes.

Edit selected blocksAssign selected blocks

☐

64.20.190.

☐

64.20.190.

☐

64.20.190.

☒

64.20.190.☐☒☐

Updating information for 2 blocks

☒ Allow sub assignments for this block

☒ Update field

RIR

LIR

Block Code

ARIN

☐ Update field

☐ Update field

☐ Update field

Notes

☐ Update field

Select tags...

☒ Ignore – do not update tags information

☐ Replace tags – replace tags for each block with the selected tags below

Unassigning IP Space

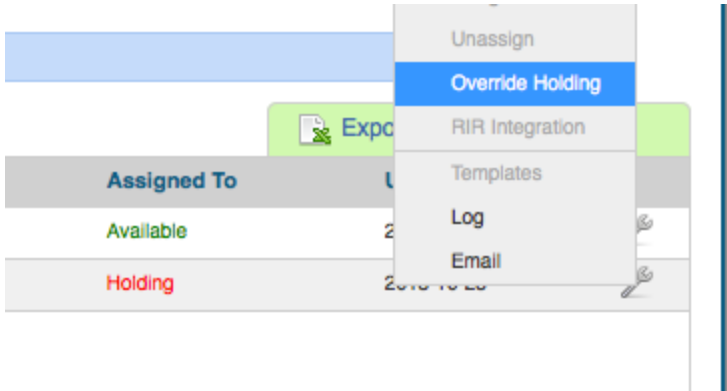
When a block is assigned, you will then have the option of unassigning the block from the resource and returning it to the Holding Tank.

To unassign the block, simply click on the Action Menu (wrench icon) for the block and select "Unassign".

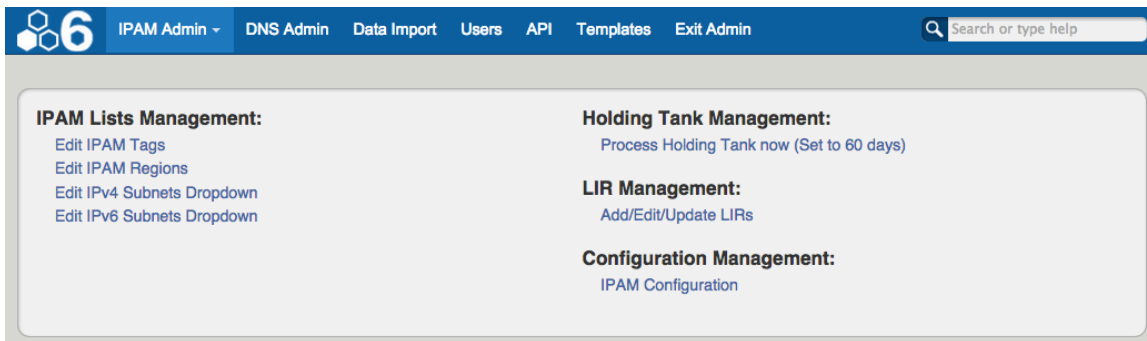
Assigned To	Updated
Amazon Server 1	<div>Edit</div>
6connect Labz	<div>Split</div>
6connect Labz	<div>Aggregate</div>
6connect Labz	<div>Assign</div>
6connect Labz	<div>Unassign</div>
6connect Labz	<div>Override Holding</div>
6connect Labz	<div>RIR Integration</div>
6connect Labz	<div>Templates</div>
Available	<div>Log</div>
7connect	<div>Email</div>
	2014-09-05

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 under [IPAM Admin](#) (this will only process blocks that were present for the specified number of days).



For more information on the Holding Tank, see [Holding Tank Management](#).

## Peering v2

### 6connect Peering

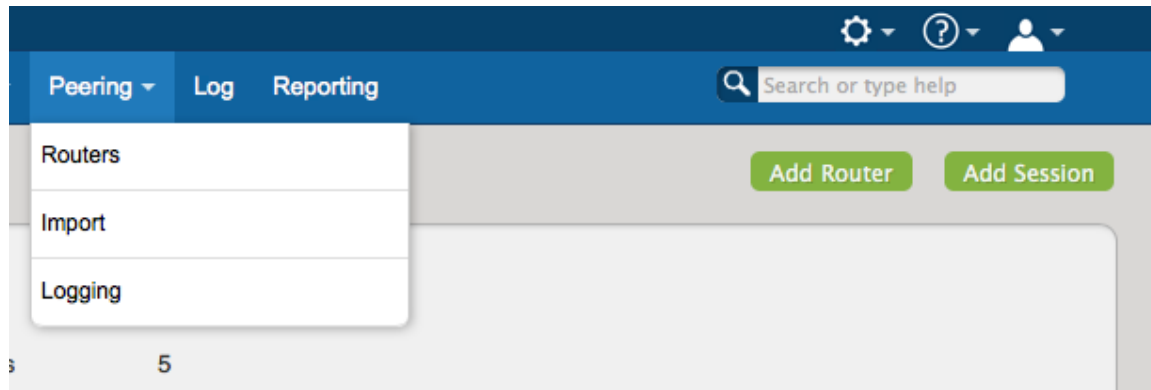
The **Peering** tab displays peering stats, allows you to add routers and sessions, and to manage communications and sessions for each exchange.

Three other sections are available via the drop down menu:

**Routers** - Links to the resource list of routers

**Import** - Import Sessions by exchange and router

**Logging** - View peering related logs



#### Table of contents

- Peering Workflow
- Peering - Common Tasks
  - Add Routers
  - Add Sessions
  - Import Sessions
- Managing Peer Communications
- Managing Peer Sessions

## Peering Workflow

### Peering Workflow

(Typical Workflow order in progress)

## Peering - Common Tasks

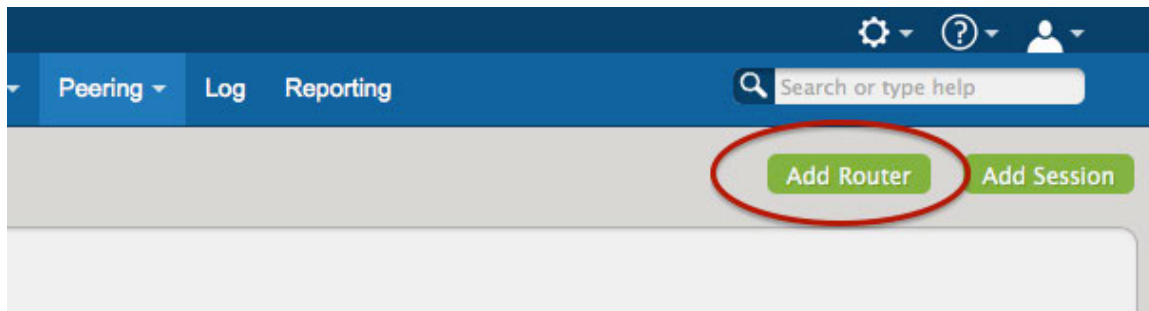
### Peering Common Tasks

- Add Routers
- Add Sessions
- Import Sessions

## Add Routers

### Adding Routers

Navigate to the **Peering** tab. Select "Add Router".



Enter the router information for Parent, Name, Make, Model, Addresses, Username, Password, and Exchange.

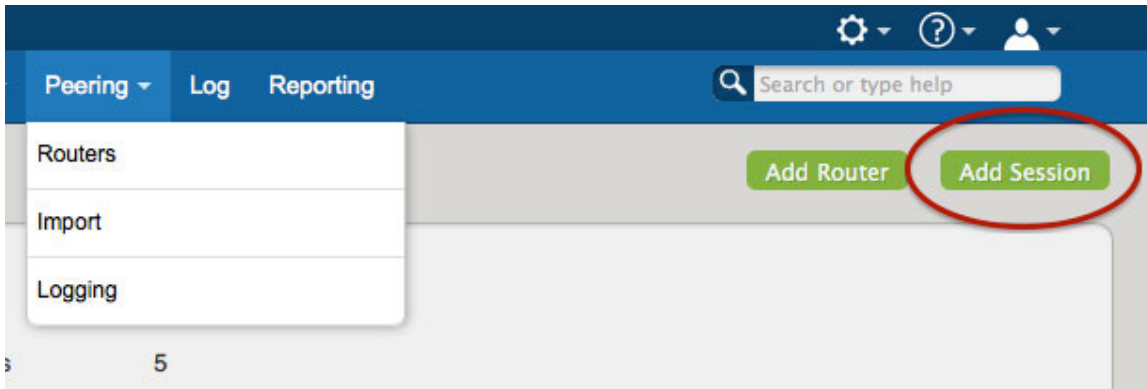
For Peer Group, type in the name of the desired Peer Group name, select whether it is IPv4 / IPv6, and click "Add Group". Lastly, click "Add Router".

A screenshot of a web form titled 'Add Router'. The form is enclosed in a light gray border with a close button (X) in the top right corner. It contains several input fields and dropdown menus. On the left side, there are fields for 'Parent Resource' (dropdown with 'TLR' selected), 'Name' (text input), 'Make' (dropdown with 'A10 Networks' selected), 'Model' (dropdown with '7600 Series' selected), 'Hostname' (text input), 'IPv4 Address' (text input), 'IPv6 Address' (text input), 'Username' (text input), and 'Password' (text input). On the right side, there are fields for 'Exchange' (dropdown with 'Equinix Internet Exchange' selected), 'Peer Group' (text input), and 'Type' (radio buttons for 'IPv4' and 'IPv6', with 'IPv4' selected). Below these fields, there is a table with three columns: 'Exchange', 'Peer Group', and 'Type'. The table is currently empty, with the text 'No groups specified' below the headers. A green 'Add Router' button is located at the bottom right of the form.

## Add Sessions

### Adding a Peering Session

From the **Peering** tab, Select "Add Session".



In the Add Session form, fill out the session information including the session Type and Exchange, the Source information, Peer Group, and the Destination. Destination IP can be pulled from the public PeeringDB information, or custom data may be specified.

If you have **enabled** and **added VRFs** to a router, the source ASNs for the associated VRFs will appear in the source ASN dropdown when adding or editing a session for that router.

A screenshot of the 'Add Session' form in the Peering Manager interface. The form is titled 'Add Session' and contains several sections for inputting session details. The 'Source' section includes fields for 'Type' (set to 'Peer'), 'Exchange' (a dropdown menu), 'Note' (a text area), 'Router' (a dropdown menu), 'IP Address' (a text field), and 'ASN' (a dropdown menu showing 'AS20940'). The 'Destination' section includes a dropdown for 'Peer Group' (set to 'Select Peer Group...'), 'MD5' (a text field), 'Max Prefixes' (a text field), and a sub-section for selecting peer and public IP data from PeeringDB or specifying custom data. This sub-section has a 'Peer' dropdown (set to 'Peer Name') and a 'Public IP' dropdown (set to 'Public IP (from PeeringDB)'). Below these are text fields for 'Peer', 'ASN', and 'IP Address'. At the bottom left, there is a checkbox labeled 'Configure router after saving?'. At the bottom right, there is a green 'Save' button.

If you would like for the router to be automatically configured when adding your session, check the "Configure Router After Saving" box, then hit "Save". If left unchecked, the session can always be configured later in the Peering Manager.

## Import Sessions

### Importing Sessions

Navigate to the [Peering Tab](#). In the dropdown menu, select **Import**.

First, select the desired exchange and router. Available Sessions will automatically populate the list.

The screenshot shows the 'Import Sessions' interface. At the top, there are two dropdown menus: 'Exchange' (labeled 'Select Exchange...') and 'Router' (labeled 'Select Router...'). Below these is a table with the following headers: Type, Source ASN, Source IP Address, Peer, Peer ASN, Peer IP Address, and Group. The table content shows 'No router selected'. At the bottom left is a blue button labeled 'Import Selected Sessions', and at the bottom right is a blue link labeled 'Clear'.

If edits need to be made to the session prior to import, simply click on the wrench icon to edit fields, then click "Done".

Lastly, select the check box next to each Session to import and click "Import Selected Sessions". Successful imports will then display with a green check mark at the beginning of the row.

The screenshot shows the 'Import Sessions' interface with the 'Exchange' dropdown set to 'Equinix Internet Exchange Palo' and the 'Router' dropdown set to 'lab2-cisco'. Below the dropdowns, it says 'Importing sessions from lab2-cisco (50.240.195.132) at Equinix Palo Alto.' and '13 sessions found. 0 already imported or added.' There is a blue button labeled 'Import Selected Sessions' and a blue link labeled 'Clear'. Below this is a table with the following headers: Type, Source ASN, Router, Peer, Peer ASN, Peer IP Address, Group, State, and an edit icon (wrench). The table contains three rows of data:

<input type="checkbox"/>	Type	Source ASN	Router	Peer	Peer ASN	Peer IP Address	Group	State	
<input checked="" type="checkbox"/>	Unknown	8038	50.240.195.132		8038	67.221.246.2		Established, up for 4d16h	
<input checked="" type="checkbox"/>	Unknown	8038	50.240.195.132	Private Peer 1	2	192.168.0.2		Idle	
<input type="checkbox"/>	Unknown	8038	50.240.195.132		9977	192.168.0.3	group-a	Idle	

The next step is to [configure and manage your sessions](#).



## Managing Peer Communications

### Communications Manager

Navigate to the **Peering** tab. Select "Communications" for the desired exchange to bring up the peer communications manager.



**Equinix Ashburn - Ashburn, US (206.223.115.0/24)**


Current Peers: 6  
Qualified Peers: 183  
Not Qualified Peers: 0  
Most Recent Peer: Akamai Technologies - 07/25/2014

Rejected Requests: 1  
Pending Requests: 0  
Most Recent Request:

Sessions Tracked: 18  
Sessions Up:  
Peers Without Sessions: 177

**Communications** **Sessions**

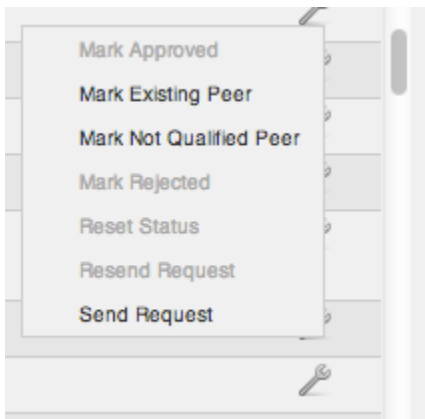
The communications manager lists the current peer communications, allowing you to mark peering status and send out peering requests from the interface. Current peers are denoted by a green check symbol under **Peer**; peers that are not qualified will show a red 'no entry' symbol. **Request** shows the peering request status, which may be: none, sent, accepted, or rejected. Updates made to the communications status will be logged under **Notes**.



Peer	ASN	Name	Request	Notes
✓	AS15145			2014-09-24 – Session updated: (AS8038/50.240.195.135) - (AS15145/1.2.3.15) 2014-09-22 – Session added: (AS8038/50.240.195.135) - (AS15145/1.2.3.15)
	AS7575	AARNet		
	AS9264	Academia Sinica Network(ASNet)		
✓	AS20940	Akamai		2014-09-22 – Session added: (AS8038/50.240.195.135) - (AS20940/206.126.236.102) 2014-09-22 – Session deleted: (AS8038/50.240.195.135) - (AS20940/206.126.236.102)
	AS20940	Akamai Technologies		

### Action Menu (Wrench Icon) Options

Select the wrench icon to manage the communication status:



**Mark Approved:** Marks the peer as approved. Available after receiving a request response.

**Mark Existing Peer:** Marks a peer as an existing one and removes the email request options.

**Mark Not Qualified Peer:** Marks a peer as "not qualified" and removes the email request options.

**Mark Rejected:** Marks the peer as rejected. Available after receiving a request response.

**Reset Status:** Resets the status of the peer, opening up the options to mark peer as existing, not qualified, or to send email requests.

**Resend Request:** Resends the peering request.

**Send Request:** Sends an initial peering request email to the peering coordinator. The email template pre-populates data based on peeringdb data (To address, Subject line and Peering exchange information). You have the chance to edit the email prior to sending.

## Managing Peer Sessions

### Managing Peering Sessions

To bring up the Peering Manager, click on "Sessions" for the desired exchange in the [Peering](#) tab.

#### Equinix Ashburn - Ashburn, US (206.223.115.0/24)

Current Peers: 6  
Qualified Peers: 183  
Not Qualified Peers: 0  
Most Recent Peer: Akamai Technologies - 07/25/2014

Rejected Requests: 1  
Pending Requests: 0  
Most Recent Request:

Sessions Tracked: 18  
Sessions Up:  
Peers Without Sessions: 177

[Communications](#)[Sessions](#)

### The Peering Manager UI:

#### BGP Sessions - Equinix Palo Alto

Filter by: Peer Source ASN Destination ASN IP Type Session Type State

Filter Clear Filters

Add Session

Source	Router	Peer	Destination	Peer Group	Type	Prfx Rcvd/Max	State	Notes
AS8038 - 50.240.195.135	lab1-cisco		AS15141 - 1.2.3.10		Peer	0/0	Idle	
AS8038 - 50.240.195.135	lab1-cisco		AS15142 - 1.2.3.11		Peer	0/0	Idle	
AS8038 - 50.240.195.135	lab1-cisco		AS15144 - 1.2.3.14		Peer	0/0	Idle	
AS8038 - 50.240.195.135	lab1-cisco		AS15145 - 1.2.3.15		Peer	0/0	configured	
AS8038 - 50.240.195.137	lab1-juniper	Amazon.com	AS16509 - 198.32.176.36	group-a	Peer	0/0	inactive	
AS8038 - 50.240.195.135	lab1-cisco	Akamai	AS20940 - 206.126.236.102	group-a	Peer	0/0	Idle	

**1) Filter Options:** The sessions list may be filtered by Peer, Source ASN, Destination ASN, IP Type, Session Type, or State. Once you've chosen the filter criteria, click on "Filter". Select "Clear Filters" to return to the full session list.

**2) Add Session:** A session can be added from the Peering Manager just like the [Add Session](#) at the top of the Peering page - the exchange field is simply automatically filled with the current exchange.

**3) Session Information:** Lists session Source, Router, Peer, Destination, Peer Group, Type, Prefixes Received / Max Prefixes, State, and Notes.

**4) Edit Session (Wrench):** Clicking on the wrench icon will bring up the following tools to manage your sessions:

configured

in

Id

Id

no

no

Id

Id

Idle

Edit

Configure

Config Manager

Email NOC

Email Policy

Email Technical

Admin Up

Admin Down

Delete

### Action Menu (Wrench Icon) Options

**Edit:** Edit session information such as Type, Exchange, Source, Peer Group, Prefixes, or Destination.

**Configure:** 1-click configure which uses default router configuration, username, and password settings.

**Config Manager:** The Config Manager allows for custom configuration commands and user-level username/ password to be entered prior to pushing the config. This is a one time use configuration.

**Email NOC:** Brings up the NOC (Network Operations Center) email template. The email template pre-populates data based on peeringdb data (To address, Subject line and Peering exchange information). You have the chance to edit the email prior to sending.

**Email Policy:** Brings up the policy email template. The email template pre-populates data based on peeringdb data (To address, Subject line and Peering exchange information). You have the chance to edit the email prior to sending.

**Email Technical:** Brings up the technical email template. The email template pre-populates data based on peeringdb data (To address, Subject line and Peering exchange information). You have the chance to edit the email prior to sending.

**Admin Up:** Ups a bgp session without removing it or adding it to the config.

**Admin Down:** Downs a bgp session without removing it or adding it to the config. On Cisco, Admin Down moves the session to Idle (Admin) state, on Juniper it deactivates the session.

**Delete:** Sessions of type "Peer" are removed from the router when deleted in ProVision. Other sessions will only be removed from the sessions list in ProVision.

## Log

## Log

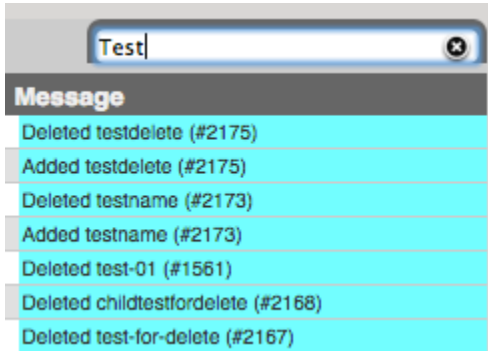
The 6connect ProVision log provides detailed information on actions performed in ProVision.

6 Dashboard Resources DNS DHCP IPAM Peering Log Reporting					
Page: 1 2 3 4 5 6 7 8 9 10 ... 2863 (143135 total) All Levels All Categories					
Time	User	IP	Level	Category	Message
2014-12-15 09:24:10	ops@6connect.com	108.198.70.50	Informational	Reporting	Generated User Activity Report for All Users
2014-12-15 09:15:48	ops@6connect.com	108.198.70.50	Informational	User	ops@6connect.com logged in via local authentication
2014-12-15 09:11:31	ops@6connect.com	108.198.70.50	Informational	User	Session timeout.
2014-12-12 15:47:11	ops@6connect.com	80.129.16.234	Informational	User	ops@6connect.com logged in via local authentication
2014-12-12 10:31:34	ops@6connect.com	108.198.70.50	Informational	Resource	Deleted testdelete (#2175)
2014-12-12 10:28:22	ops@6connect.com	108.198.70.50	Informational	Resource	Added testdelete (#2175)
2014-12-12 10:24:39	ops@6connect.com	108.198.70.50	Informational	Resource	Deleted testname (#2173)
2014-12-12 10:00:48	ops@6connect.com	108.198.70.50	Informational	Resource	Added 2173-dhcp-module (#2174)
2014-12-12 09:53:38	ops@6connect.com	108.198.70.50	Informational	Resource	Added testname (#2173)
2014-12-12 09:41:21	ops@6connect.com	74.133.203.97	Informational	User	ops@6connect.com logged in via local authentication
2014-12-12 09:41:15	ops@6connect.com	74.133.203.97	Informational	User	Session timeout.
2014-12-12 09:15:12	Unknown	64.233.172.180	Informational	User	Session timeout.
2014-12-12 09:14:43	ops@6connect.com	108.198.70.50	Informational	User	ops@6connect.com logged in via local authentication
2014-12-12 09:14:42	ops@6connect.com	108.198.70.50	Informational	User	ops@6connect.com logged in via local authentication
2014-12-12 09:14:33	ops@6connect.com	108.198.70.50	Informational	User	Session timeout.
2014-12-12 02:33:14	ops@6connect.com	217.247.77.73	Informational	User	ops@6connect.com logged in via local authentication
2014-12-12 00:11:53	ops@6connect.com	217.247.77.73	Informational	User	ops@6connect.com logged in via local authentication
2014-12-11 13:30:00	ops@6connect.com	74.133.203.97	Informational	User	ops@6connect.com logged in via local authentication
2014-12-11 13:29:54	Unknown	74.133.203.97	Informational	User	Session timeout.
2014-12-11 11:01:36	ops@6connect.com	108.198.70.50	Informational	User	ops@6connect.com logged in via local authentication
2014-12-11 11:01:35	ops@6connect.com	108.198.70.50	Informational	User	ops@6connect.com logged in via local authentication
2014-12-11 11:01:22	Unknown	108.198.70.50	Informational	User	Session timeout.
2014-12-11 09:06:29	ops@6connect.com	108.198.70.50	Informational	Resource	Deleted test-01 (#1561)
2014-12-11 09:06:10	ops@6connect.com	108.198.70.50	Informational	DNS	Deleted zone cilt.com and all records.
2014-12-11 09:04:33	daemon	localhost	Informational	DNS	Created zone cilt.com

## Log Features

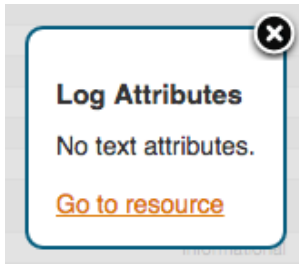
### Search:

- Enter the search text above the Message column



### Additional Details:

- Clicking on a Resource log item provides a link to the Resource's entry page

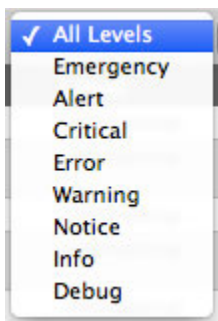


- Clicking on an API log item provides additional details about the API call

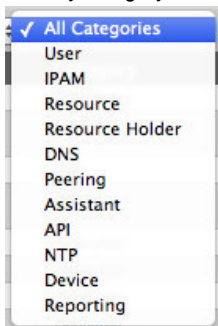


#### Filters:

- Filter by notification level



- Filter by category



## Reporting

### Reporting

The ProVision Reporting tab provides an overview of program statistics, as well as a way to view and download activity information.

#### Stats

Items of interest provided under stats include most recent login, number of Resources, DNS zone breakdowns, IPAM hosts, and estimated IP runout time.

The screenshot shows the ProVision Reporting interface. At the top is a navigation bar with links: Dashboard, Resources, DNS, DHCP, IPAM, Peering, Log, and Reporting (which is highlighted). A search bar is on the right. The main content area is divided into three sections:

- Environment:** A table showing system information.

Last Login:	12/15/2014 - 10:33:27
Last User:	ops@6connect.com
Total Resources:	435
Total Contacts:	11
- DNS Stats:** A table showing DNS zone and record statistics.

Total Zones:	987	Total A Records:	1883
Forward Zones:	664	Total AAAA Records:	1
Reverse Zones:	323	Total PTR Records:	77412
Total NS Records:	3955	Total MX Records:	745
- IPAM Stats:** A table showing IP address pool statistics, divided into IPv4 Public and IPv4 1918 (Private) sections.

IPv4 Public		IPv6	
Total IPv4 Hosts:	1,187,264	Total IPv6 Hosts:	158,456,325,028,528,675,187,087,900,672
Total Assigned IPv4 Hosts:	3,690	Total Assigned IPv6 Hosts:	1,209,019,206,256,502,329,311,232
Total Available IPv4 Hosts:	1,183,574	Total Available IPv6 Hosts:	158,455,116,009,322,418,684,758,589,440
IPv4 Assigned Date Range:	09/16/2013 – 12/10/2014 (450 days)	IPv6 Assigned Date Range:	11/27/2013 – 11/03/2014 (341 days)
IPv4 Assigned Rate :	8 hosts/day	IPv6 Assigned Rate :	3,545,510,868,787,396,608,000 hosts/day
IPv4 Projected Runout :	395 years, 163 days	IPv6 Projected Runout :	122443 years, 63 days
IPv4 1918 (Private)			
Total 1918 Hosts:	33,620,208		
Total Assigned 1918 Hosts:	3,160,418		
Total Available 1918 Hosts:	30,459,790		
1918 Assigned Date Range:	11/01/2013 – 12/10/2014 (405 days)		
1918 Assigned Rate :	7,804 hosts/day		
1918 Projected Runout :	10 years, 253 days		

### Reports

#### User Activity

To run a User Activity report, simply select the user from the drop down menu and a desired date range for the report. Clicking on "Show Data" will show the User, IP, Timestamp, and Action in a table at the bottom of the page. To export the data to .csv, simply select "Download CSV".

The form is titled "User Activity". It contains a "User:" label followed by a dropdown menu currently showing "All Users". Below this are two date input fields: "From:" with the value "12/05/2014" and "To:" with the value "12/15/2014". At the bottom are two buttons: "Show Data" (blue) and "Download CSV" (orange).

#### Customer List

The Customer List report reflects all Resources created under the Category of "Customer". Clicking on "Show Data" will show information collected from the Contact Info and Tech Info gadgets, parent information, and IP / zone assignment counts. To export the data to .csv, simply select "Download CSV".

Customer List

Show Data

Download CSV

IPAM

The IPAM report is highly customizable, allowing you to view information for all aggregates or selected blocks.

**Required Fields:** IPv4 and/or IPv6 must be selected for the report.

**Optional Fields:** Assigned, SWIP status, Assignment / Update dates, RIR, Assigned to Resource, Region, Tag, and Generic Code (in this case, "Datacenter1") are all optional parameters to narrow your results.

IPAM

All Aggregates

☒ IPv4
☒ IPv6
☐ Assigned?
☐ SWIPed/RIR Synched?

Assignment Date:

From: 12/01/2014

To: 12/15/2014

Clear

Last Updated:

From:

To:

Clear

RIR

☐ ARIN
☐ RIPE
☐ LACNI

Assigned to Resource

☐ ACER worldwide
☐ Agencia Nacional do Cine
☐ Agência Espacial Brasile
☐ Amazon

Region

☐ Quito
☒ San Jose, CA
☐ Silicon Valley, C

Tag

☐ Anycast
☐ BGP
☐ BB

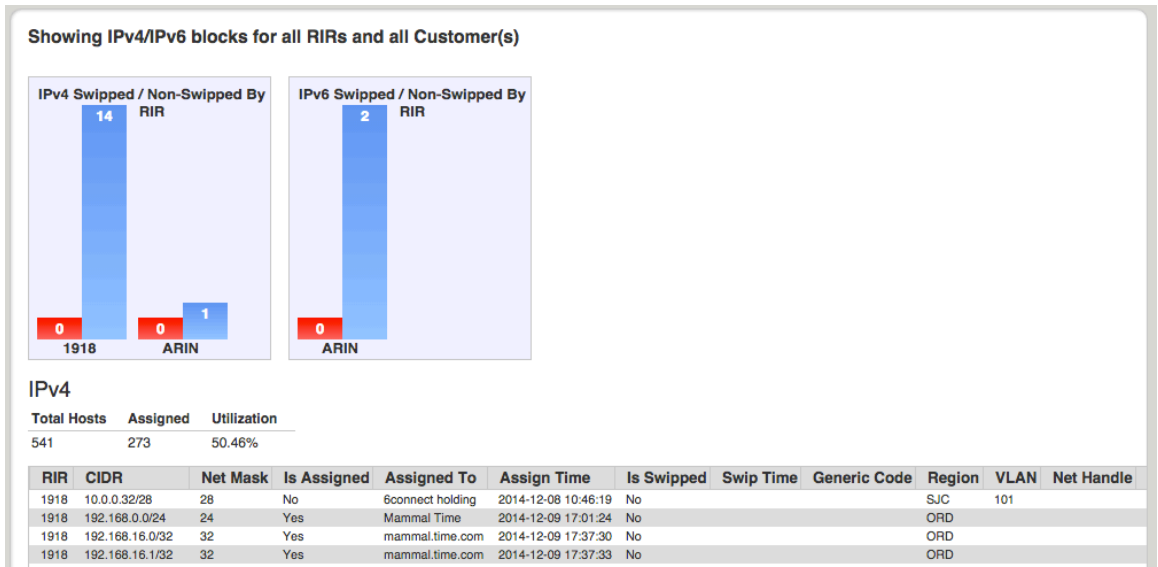
DataCenter1

☐ DS213
☐ Ethernet0.cr01.frfC
☐ Ethernet0.cr02.frfC

Show Data

Download CSV

Clicking on "Show Data" will show bar charts for Swipped/ Non-Swipped by RIR, host and utilization stats, as well as detailed block information. To export the data to .csv, simply select "Download CSV".





# **ProVision Admin Guide**

## **Admin Guide**

The ProVision Admin Guide provides information on features accessible with Admin permissions within ProVision. For more detailed information on features accessible in the standard user tabs , see the [ProVision User Guide](#).

### **Table of contents**

- Admin Preferences
- Authentication Options
- IPAM Administration
- DNS Administration
- Importing Your Data
- Users & Permissions
- API Tab
- Templates

# 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

**VRF Support:** Check to enable adding the VRF gadget to the router Section. Currently, only supports Cisco routers.

### 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

### Authentication



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.



#### Change Order of Login Menu Dropdown

The drop down menu defaults to "local" - if you are using another authentication method, you can use the following to change the default ordering and improve usability.

In the file data/globals.php, add a line:

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

Acceptable values instead of 'ldap' are 'local', 'radius' and 'ldap'.

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

- [LDAP Authentication](#)
- [LDAP Authentication on Windows Server](#)
- [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

### LDAP Schema - Example

```
attributetype (1.3.6.1.4.1.5023215.2.3.21 NAME 'sixConnGroup'
              SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

objectclass ( 1.3.6.1.4.1.5023215.2.4.2 NAME 'sixConnectPermissionsV2'
             DESC '6Connect Permissions Object v2'
             SUP top AUXILIARY
             MUST ( sixConnGroup ) )
```

### LDAP User Example

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: "IT Engineering"
sixConnGroup: "Sales"
sixConnGroup: "Customer Admin"
userPassword: testpass
```

To create a new user, make a new ldif 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".



#### Using SSL encryption

To use SSL encryption with LDAP, the ldap.conf file must be correctly configured on the ProVision server.

Typically, the LDAP configuration file is kept at "/etc/ldap/ldap.conf". Make sure the following line is present:

```
TLS_REQCERT allow
```

and restart the webserver.

## LDAP Authentication on Windows Server

### LDAP Authentication on Windows Server

Starting in 3.6, ProVision supports LDAP authentication (including Windows Server!). To setup an LDAP server for authentication, you must perform the following procedures:

- Configure the LDAP Server (Extend the Schema, Adding an Attribute/Schema Object)
- Test the LDAP Server
- Configure ProVision for LDAP Authentication

#### Configuring the LDAP functions on your Windows Server

You should confirm these steps with your LDAP admin - the purpose of this walkthrough is to provide some level of detail on how to extend LDAP functionality to support integration with an application like ProVision.

**Step 1:** Prepare to extend the Schema (<http://technet.microsoft.com/en-us/library/cc961754.aspx>)

This is not a minor operation and requires interaction with various control modification areas of Windows Server:

- If you have not modified the schema before, you will need to use the Active Directory Schema console on a DC (Domain Controller) to permit write access to the DC schema.
- Since the schema object has dedicated permissions, admins must be a member of the Schema Administrator group (Schema Admins).
- Note that the DC that is holding the Schema Master Role is the only one allowed to write to it.

**Step 2:** Decide on method for Installing/executing Schema Extensions (<http://technet.microsoft.com/en-us/library/cc961742.aspx>)

If you have already used other AD integrations, this should be straightforward. We recommend using the LDIF script method

**Step 3:** Add and Modify a Schema Object (<http://technet.microsoft.com/en-us/library/cc961575.aspx>)

To add a new attribute to the schema, you first have to create a attribute object. The you will need to complete the following steps:

- Select a name for the attribute (ProVision assumes that the name will be '**sixConnGroup**')
- Get a valid Object Identifier (OID) from an issuing authority (<http://msdn.microsoft.com/en-us/library/ms677620.aspx>)



#### Generate an Object Identifier

Microsoft has released a script that can generate an Object Identifier (OID):

<https://gallery.technet.microsoft.com/scriptcenter/56b78004-40d0-41cf-b95e-6e795b2e8a06>

- Document the attribute syntax
- Confirm that the attribute should be single-value
- Confirm the attribute indexing behavior
- Decide if the attribute needs to be distributed to the Global Catalog

#### LDAP Schema - Example

```
attributetype (1.3.6.1.4.1.5023215.2.3.21 NAME 'sixConnGroup'
              SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

objectclass ( 1.3.6.1.4.1.5023215.2.4.2 NAME 'sixConnectPermissionsV2'
             DESC '6Connect Permissions Object v2'
             SUP top AUXILIARY
             MUST ( sixConnGroup ) )
```

#### LDAP User Example

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: "IT Engineering"
sixConnGroup: "Sales"
sixConnGroup: "Customer Admin"
userPassword: testpass
```

To create a new user, make a new ldif 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".



### Using SSL encryption

To use SSL encryption with LDAP, the `ldap.conf` file must be correctly configured on the ProVision server.

Typically, the LDAP configuration file is kept at `/etc/ldap/ldap.conf`. Make sure the following line is present:

```
TLS_REQCERT allow
```

and restart the webserver.

## RADIUS Authentication

### 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:

- RADIUS Authentication
  - [Add the 6connect VSA to the Radius Installation](#)
  - [Configure Radius Accounts](#)
  - [Test Radius Accounts](#)
  - [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".

# IPAM Administration

## Overview

The screenshot shows the IPAM Administration interface with a blue header bar containing navigation links: IPAM Admin, DNS Admin, Data Import, Users, API, Templates, and Exit Admin. A search bar is on the right. The main content area is divided into four sections: IPAM Lists Management (with links to Edit IPAM Tags, Edit IPAM Regions, Edit IPv4 Subnets Dropdown, and Edit IPv6 Subnets Dropdown), Holding Tank Management (with a link to Process Holding Tank now (Set to 60 days)), LIR Management (with a link to Add/Edit/Update LIRs), and Configuration Management (with a link to IPAM Configuration).

IPAM Administration is accessed through the Admin area of ProVision. It includes sections to manage IPAM Lists, the Holding Tank, LIR, and IPAM Configuration.

## IPAM Lists Management

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

The screenshot shows the IPAM Configuration page with a title "IPAM Configuration". It contains several settings: Holding Tank Days (60), IPv4 Block Scanner Enable (checked), IPv4 Block Scanner Max Block Size (/20), Regions Enable (checked), Generic Code Per Block Enable (checked), Generic Code Per Block Display (checked), Generic Code Per Block Name (DataCenter1), Enable VLAN per Block (checked), RIPE Database (RIPE selected, TEST unselected), and Show /32 or /128 mask for statics (checked). An "Update" button is at the bottom left.

**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.

## IPAM Parameters

### Overview

The elements

**IPAM Lists Management:**

- Edit IPAM Tags
- Edit IPAM Regions
- Edit IPv4 Subnets Dropdown
- Edit IPv6 Subnets Dropdown

### 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 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' dropdown menu currently set to 'All Masks'. A 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 shows several rows of IP blocks, some with 'Available' status and others with 'Some Customer' status. A red block is highlighted in the table.

Hosts	RIR	Region	Notes	Tags	Something	Assigned To	Updated
2	1918					<a href="#">bri.awesomesauce.com</a>	2013-07-24 19:54:13
2	1918					<a href="#">bri.awesomesauce.com</a>	2013-07-24 19:54:21
1	1918					<a href="#">bri.totalityrad.com</a>	2013-07-25 16:41:40
12.11.10.5/32	1	1918				<a href="#">bri.totalityrad.com</a>	2013-07-25 16:41:48
12.11.10.6/31	2	1918				Available	2013-07-25 16:41:03
12.11.10.8/29	8	1918				Some Customer	2013-08-05 19:46:21
12.11.10.16/29	8	1918				Some Customer	2013-08-05 19:46:37
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: <input type="text"/> 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: <input type="text"/> 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.totallyrad.com 12.11.10.5/32 - Assigned to br1.totallyrad.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

2001:db7::32 (2001:db7:: - 2001:db7:ffff:ffff:ffff:ffff) Tags Used: Anycast, BB, BGP, Customer, DSL, Infrastructure, Internal, Loopback, MPLS Regions Used: LON										
Filtered by: Mask: 36, 48, 64 X           Search           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

### 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.



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



#### **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.

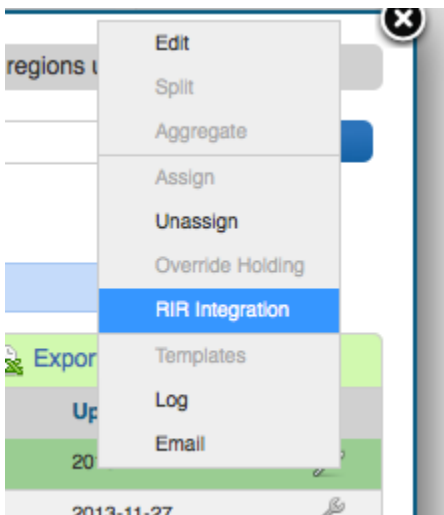
We currently support ARIN and RIPE with support for APNIC, LACNIC and AfriNIC coming in 2014.

### LIR Setup and Use

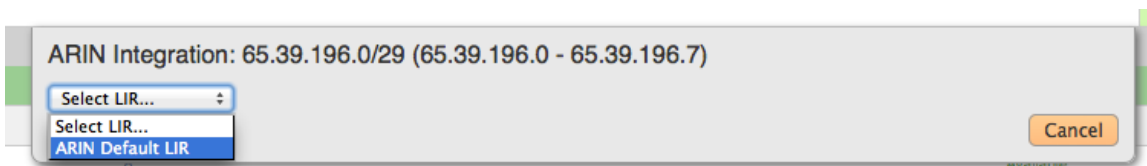
There is an LIR Manager available from the IPAM Admin page.

LIR Manager				
Name	RIR	ASN	MNT-BY/ORG-HANDLE	Other Information
ARIN Default LIR	ARIN	9498	CONN-81	Admin POC: 6CONN-ARIN , Tech POC: 6CONN-ARIN
ipspace 1234	RIPE			
<a href="#">Add LIR</a>				
<a href="#">Back to IPAM Admin</a>				

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):

ARIN Integration: 67.221.244.0/28 (67.221.244.0 - 67.221.244.15)

6connect

Org Handle	Admin POC	Net POC	Abuse POC	Net Name Prefix	API Key
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.

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	<input type="text" value="CONNE-81"/>	<input type="button" value="Delete"/>
Admin POC	<input type="text" value="6CONN-ARIN"/>	
Tech POC	<input type="text" value="6CONN-ARIN"/>	
Abuse POC	<input type="text" value="6CONN-ARIN"/>	
NET Name Prefix	<input type="text" value="6CONN"/>	
API Key	<input type="text"/>	

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

### RIPE

## Add LIR

RIR

RIPE

Name

Some Company

ASN

12345

Maintainer

Delete

Password

Admin Contact

Tech Contact

Add Maintainer

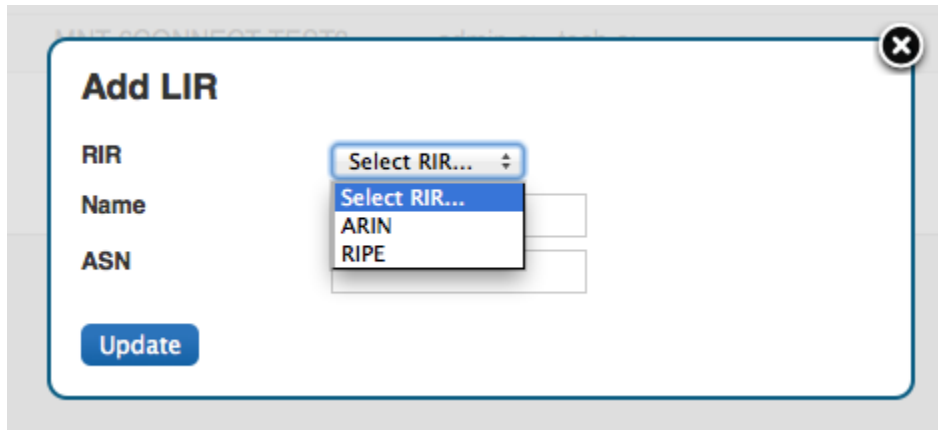
Update

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

## ARIN LIR Setup and Use

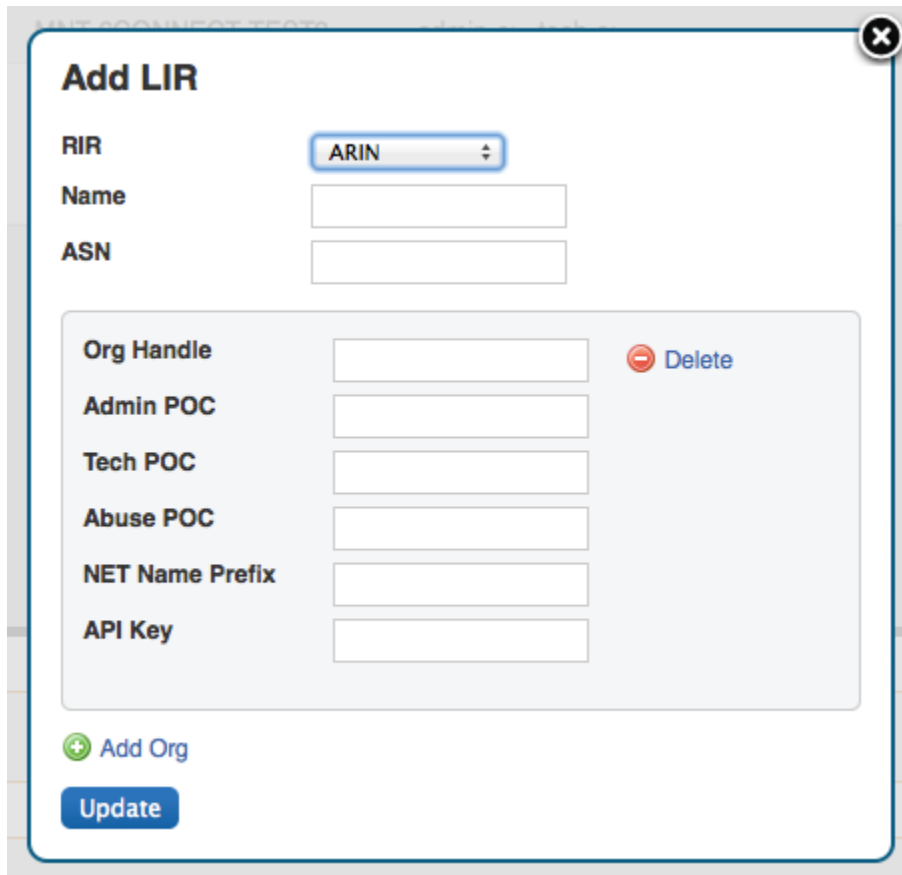
### Step 1: Setup the LIR information via the LIR Manager

You will be prompted to select the RIR



The screenshot shows a dialog box titled "Add LIR" with a close button in the top right corner. Inside the dialog, there are three input fields: "RIR", "Name", and "ASN". The "RIR" field has a dropdown menu open, showing the options "Select RIR...", "ARIN", and "RIPE". Below the input fields is a blue "Update" button.

Add in the requisite Org and POC information



The screenshot shows the "Add LIR" dialog box with the "RIR" dropdown menu now set to "ARIN". Below the "Name" and "ASN" fields, there is a section for Org and POC information. This section contains a table with the following fields: "Org Handle", "Admin POC", "Tech POC", "Abuse POC", "NET Name Prefix", and "API Key". Each field has a corresponding input box. To the right of the "Org Handle" input box is a red minus icon and the text "Delete". Below the table is a green plus icon and the text "Add Org". At the bottom of the dialog is a blue "Update" button.



#### 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](#):

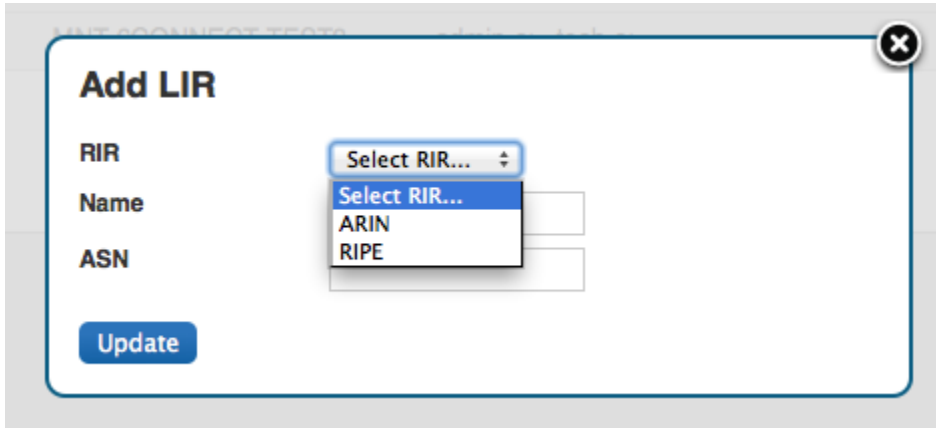
Used to subdelegate IP addresses to a downstream organization that will further subdelegate the IP addresses to their own customers. These requests must be submitted by an ARIN Online user account linked to the parent organization's Admin or Tech POC, or the Tech POC for the resource.

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

## RIPE LIR Setup and Use

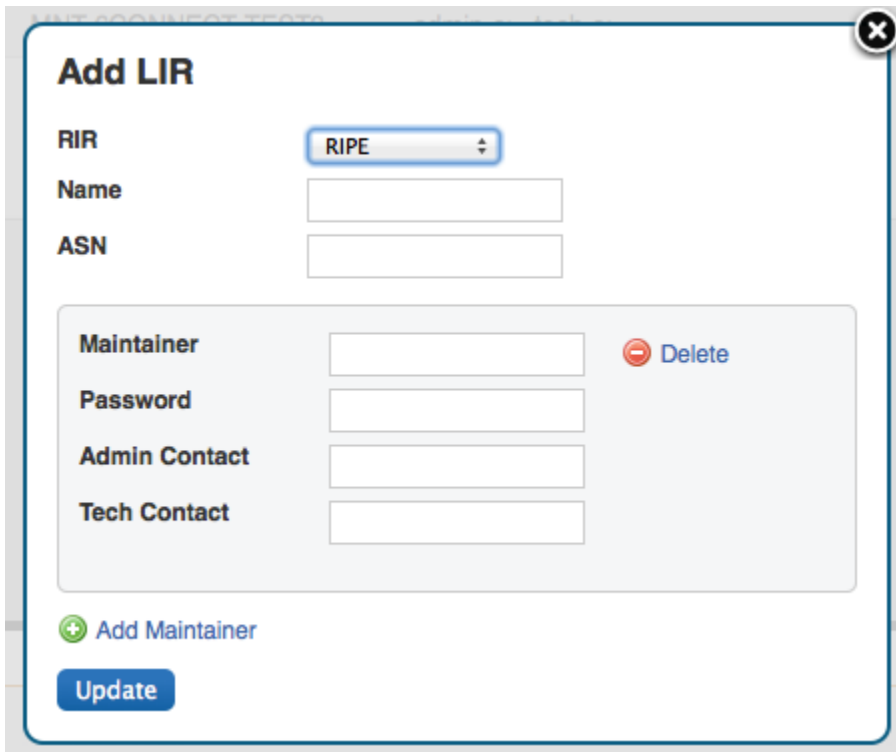
### Step 1: Setup the LIR information via the LIR Manager

You will be prompted to select the RIR:



The screenshot shows a modal dialog titled "Add LIR" with a close button in the top right corner. Inside the dialog, there are three input fields: "RIR", "Name", and "ASN". The "RIR" field has a dropdown menu open, showing the options "Select RIR...", "ARIN", and "RIPE". The "Name" and "ASN" fields are empty. At the bottom left of the dialog is a blue "Update" button.

Then add in the requisite Maintainer Object related information:



The screenshot shows the "Add LIR" dialog box with the "RIR" dropdown set to "RIPE". The "Name" and "ASN" fields are still empty. Below these fields is a section for "Maintainer" information, which includes four input fields: "Maintainer", "Password", "Admin Contact", and "Tech Contact". To the right of the "Maintainer" field is a red "Delete" button. At the bottom left of the dialog is a green "+ Add Maintainer" link, and at the bottom right is a blue "Update" button.



#### 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.



2014-03-04 16:22:52	
2014-04-23 13:33:18	
2013-10-21 08:31:29	
2013-10-21 08:31:43	
2013-11-25 08:20:53	
2014-06-03 22:44:55	
2014-01-30 12:29:05	
2013-11-27 15:36:12	
2014-06-03 22:07:00	

- Edit
- Unassign
- RIR Integration
- Log
- Email

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.

193.0.0.0 - 193.0.0.31

(NET-193-0-0-0-27)

MNT-BY: MNT-6CONNECT-TEST

Hosts	RIR
<a href="#">32</a>	RIPE
<a href="#">32</a>	RIPE

## DNS Administration

# DNS Administration

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 Administration
- DNS Functions
  - DNS Zone Transfers
  - Manage DNS Servers
  - Bulk Zone Assignments
  - Bulk DNS Changes
  - Nameserver Management
  - Notes
  - System Information for Local Installations

## 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 several input fields and dropdown menus. The "Server" field has a dropdown menu with "ns1-ns1.6clabs.com" selected and a "New Server" button next to it. The "Nick Name" field contains "ns1". The "FQDN or IP" field contains "ns1.6clabs.com" with an example "ex: ns1.dns.6connect.net or 216.239.32.10". The "Default" field is a dropdown menu with "Do Not Add to New Zones" selected. The "Transfer Type" field is a dropdown menu with "ISC BIND" selected. The "Server Type" field is a dropdown menu with "Master" selected. The "SOA" field contains "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 commands.
- 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 <link to ip list creation>IP List</link> 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.

### Additional Information:

For more information on DNS and configurations, see the following sections.

- [Working with DNS Zones](#)
- [Configuring ISC BIND Support](#)
- [Configuring DynECT Support](#)
- [Configuring PowerDNS Support](#)
- [Configuring Secure64 Support](#)
- [Configuring DNSSEC](#)
- [Configuring Split Horizon/Views](#)
- [Configuring DNS Templates](#)

## Working with DNS Zones

### Using the DNS Gadget

When you have defined a Resource, you can assign the DNS Gadget to a given Section. 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 the DNS Gadget interface. At the top, there's a 'New DNS Zone' section with a text input field for the zone name, a dropdown menu set to '-- no template --', and a 'Create Zone' button. Below this is the 'Zone Delegation' section, which includes a 'Delegated Zone' text input, a 'Slave IP' dropdown set to 'IPv4 or IPv6', a 'Customer' text input with the value '260', and an 'Add Slave' button.

### Navigating the DNS Tab

Clicking on the main DNS Tab, then on "Add Zone" 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.

The screenshot shows the 'Create a DNS Zone' form. It has a text input field labeled 'Enter Zone Name' and a dropdown menu labeled 'Select Site...'. To the right of the dropdown is a green plus sign button.

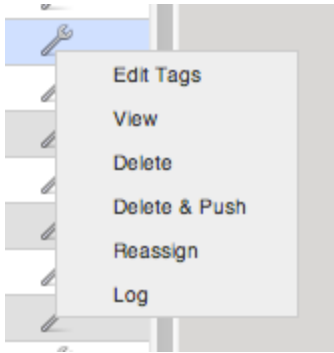
### DNS Tab User Interface

The screenshot shows the DNS Tab User Interface. At the top, there's a 'Page:' label followed by a pagination control showing '1 2 3 4' and '(92 total)'. To the right is a search bar. Below this is a table with the following columns: Zone, Customer, Tags, DNSSEC, DS, Records, and Actions. The table contains five rows of data. Numbered callouts (1-7) point to specific elements: 1 points to the pagination control, 2 points to the search bar, 3 points to the 'Zone' column header, 4 points to the 'Customer' column header, 5 points to the 'Tags' column header, 6 points to the 'DNSSEC' column header, and 7 points to the 'Records' column header.

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>	<a href="#">123 Department LAB</a>		DNSSEC	X	11	
<a href="#">anna.com</a>	<a href="#">123 Department LAB</a>				12	
<a href="#">awesome.com</a>	<a href="#">Anna's Test Site</a>		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) The **Customer** 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 is 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)**

The screenshot shows the DNS Zone Management interface for the domain **awesome.com**. The DNSSEC status is **Enabled**. The SOA record details are: TTL: 3600, Serial: 2013071001, Refresh: 14400, Retry: 3600, Expire: 604800, Minimum: 3600. The "Link Zone to Server" section shows the IP 173.164.182.169 as the Master server, with an "Add" button. The "Current Masters" list includes ns1.6clabs.com, nalinmk.com, and cache.6connect.com, each with a red error icon. The "Current Slaves" list is empty. At the bottom, there are buttons for "Edit SOA", "Disable Auto Check", "Check Zone [OK!]", and "Push Zone".

Figure 1: Normal zone with no errors

The screenshot shows the DNS Zone Management interface for the domain **awesome.com**. The DNSSEC status is **Enabled**. The SOA record details are the same as in Figure 1. The "Link Zone to Server" section shows the IP 173.164.182.169 as the Master server, with an "Add" button. The "Current Masters" list includes ns1.6clabs.com, nalinmk.com, and cache.6connect.com, each with a red error icon. The "Current Slaves" list is empty. At the bottom, there are buttons for "Edit SOA", "Disable Auto Check", and a red "ERRORS!" button. Below the buttons, there are three red error messages: "View 6connectGeneric on cache.6connect.com NS 'ns2.dns.6connect.net.awesome.com' has no address records (A or AAAA)", "View 6connectGeneric on nalinmk.com NS 'ns2.dns.6connect.net.awesome.com' has no address records (A or AAAA)", and "View 6connectGeneric on ns1.6clabs.com NS 'ns2.dns.6connect.net.awesome.com' has no address records (A or AAAA)".

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
Search
Clear

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	

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	
DNS Views: ns1.6clabs.com					
	ns1.6clabs.com				
	ns2.6clabs.com				
	ns3.6clabs.com				
	ns4.6clabs.com				
	ns5.6clabs.com				
	ns6.6clabs.com				
	ns7.6clabs.com				
	ns8.6clabs.com				
	ns9.6clabs.com				
	ns10.6clabs.com				
	ns11.6clabs.com				
	ns12.6clabs.com				
	ns13.6clabs.com				
	ns14.6clabs.com				
	ns15.6clabs.com				
	ns16.6clabs.com				
	ns17.6clabs.com				
	ns18.6clabs.com				
	ns19.6clabs.com				
	ns20.6clabs.com				
	ns21.6clabs.com				
	ns22.6clabs.com				
	ns23.6clabs.com				
	ns24.6clabs.com				
	ns25.6clabs.com				
	ns26.6clabs.com				
	ns27.6clabs.com				
	ns28.6clabs.com				
	ns29.6clabs.com				
	ns30.6clabs.com				
	ns31.6clabs.com				
	ns32.6clabs.com				
	ns33.6clabs.com				
	ns34.6clabs.com				
	ns35.6clabs.com				
	ns36.6clabs.com				
	ns37.6clabs.com				
	ns38.6clabs.com				
	ns39.6clabs.com				
	ns40.6clabs.com				
	ns41.6clabs.com				
	ns42.6clabs.com				
	ns43.6clabs.com				
	ns44.6clabs.com				
	ns45.6clabs.com				
	ns46.6clabs.com				
	ns47.6clabs.com				
	ns48.6clabs.com				
	ns49.6clabs.com				
	ns50.6clabs.com				
	ns51.6clabs.com				
	ns52.6clabs.com				
	ns53.6clabs.com				
	ns54.6clabs.com				
	ns55.6clabs.com				
	ns56.6clabs.com				
	ns57.6clabs.com				
	ns58.6clabs.com				
	ns59.6clabs.com				
	ns60.6clabs.com				
	ns61.6clabs.com				
	ns62.6clabs.com				
	ns63.6clabs.com				
	ns64.6clabs.com				
	ns65.6clabs.com				
	ns66.6clabs.com				
	ns67.6clabs.com				
	ns68.6clabs.com				
	ns69.6clabs.com				
	ns70.6clabs.com				
	ns71.6clabs.com				
	ns72.6clabs.com				
	ns73.6clabs.com				
	ns74.6clabs.com				
	ns75.6clabs.com				
	ns76.6clabs.com				
	ns77.6clabs.com				
	ns78.6clabs.com				
	ns79.6clabs.com				
	ns80.6clabs.com				
	ns81.6clabs.com				
	ns82.6clabs.com				
	ns83.6clabs.com				
	ns84.6clabs.com				
	ns85.6clabs.com				
	ns86.6clabs.com				
	ns87.6clabs.com				
	ns88.6clabs.com				
	ns89.6clabs.com				
	ns90.6clabs.com				
	ns91.6clabs.com				
	ns92.6clabs.com				
	ns93.6clabs.com				
	ns94.6clabs.com				
	ns95.6clabs.com				
	ns96.6clabs.com				
	ns97.6clabs.com				
	ns98.6clabs.com				
	ns99.6clabs.com				
	ns100.6clabs.com				

All Views  
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.

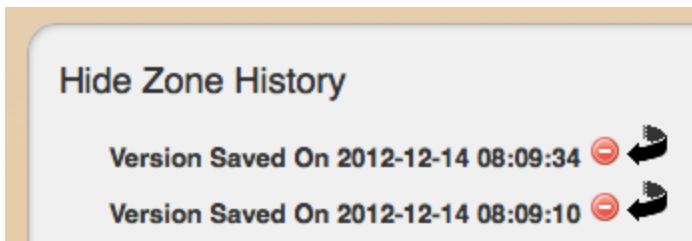
@ IN COMMENT update A record based on turnup date
@ IN NS dns2.mycloud.net.
@ IN NS dns3.mycloud.net.
amazon.com. 3600 IN NS ns1.dns.6connect.net.
amazon.com. 3600 IN NS ns2.dns.6connect.net.
amazon.com. 3600 IN NS ns3.dns.6connect.net.
amazon.com. 3600 IN NS ns1.dns.bind.com.
IN MX 10 mx.mycloud.net.
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)
- BIND or MySQL backend

### Overview



### Step 1: Setup your PowerDNS Server

### Manage DNS Servers

Server:	<input type="text" value="208.39.104.106"/>	<a href="#">New Server</a>
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"/>	
	<a href="#">Update Server</a>	<a href="#">Delete Server</a>

### 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



Only BIND and MySQL backends are supported.

## 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 Authoritative 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: 50.198.192.141

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 ProVision 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.



6connect ProVision – © 2014 v5.0.0 – (6connect) ops@6connect.com

IPAM Admin DNS Admin Data Import Users API Templates Exit Admin Search Admin User Preferences

### Manage DNS Servers

Server:  **New Server**

### Bulk Zone Assignment

Search for Zone:  **Match**

### Bulk Record Changes

**WARNING. This is a power user tool.**

Record Host  Record Type  Record Value  ☐ Strict Comparison **Search Records**

### Nameserver Management

Server	Default	Uses
▼ ns1.dns.6connect.net	<input type="checkbox"/>	0
▲ ▼ ns2.dns.6connect.net	<input type="checkbox"/>	0
▲ ▼ ns3.dns.6connect.net	<input type="checkbox"/>	0

### DNS Functions

- Manage Defaults and Options
- Edit DNS Record Types
- Generate all DS records for DNSSEC
- Generate zip file of all zones
- Increment All Serials
- DNS View ACL Management

### DNS Zone Transfers:

dns.6connect.net	0 Zones	<input type="checkbox"/>
services1.tcp0.com	0 Zones	<input type="checkbox"/>
ns1.sc2000.net	0 Zones	<input type="checkbox"/>
test.server	0 Zones	<input type="checkbox"/>
6connect Test Server	2 Zones	<input type="checkbox"/>
ns1.6clabs.com	0 Zones	<input type="checkbox"/>
ns2.6clabs.com	0 Zones	<input type="checkbox"/>
powerdns	2 Zones	<input type="checkbox"/>
Example Server	5 Zones	<input type="checkbox"/>

Push Zones to Checked Servers: **Push**

Then fill in the information for your Secure64 server (including any relevant SOA information):

6connect ProVision – © 2014 v5.0.0 – (6connect) ops@6connect.com

IPAM Admin DNS Admin Data Import Users API Templates Exit Admin Search or type help

### Manage DNS Servers

Server Name:  **Edit Server**  
 FQDN or IP:   
 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**

### Bulk Zone Assignment

Search for Zone:  **Match**

### Bulk Record Changes

**WARNING. This is a power user tool.**

Record Host  Record Type  Record Value  ☐ Strict Comparison

### DNS Functions

- Manage Defaults and Options
- Edit DNS Record Types
- Generate all DS records for DNSSEC
- Generate zip file of all zones
- Increment All Serials
- DNS View ACL Management

### DNS Zone Transfers:

dns.6connect.net	0 Zones	<input type="checkbox"/>
services1.tcp0.com	0 Zones	<input type="checkbox"/>
ns1.sc2000.net	0 Zones	<input type="checkbox"/>
test.server	0 Zones	<input type="checkbox"/>
6connect Test Server	2 Zones	<input type="checkbox"/>
ns1.6clabs.com	0 Zones	<input type="checkbox"/>
ns2.6clabs.com	0 Zones	<input type="checkbox"/>
powerdns	2 Zones	<input type="checkbox"/>
Example Server	5 Zones	<input type="checkbox"/>
test	0 Zones	<input type="checkbox"/>

Push Zones to Checked Servers: **Push**

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

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

6connect ProVision - © 2014 v5.0.0 - (6connect) ops@6connect.com

IPAM Admin DNS Admin Data Import Users API Templates Exit Admin Search or type help

### Manage DNS Servers

Server Name: Secure64 Test Server [Edit Server](#)

FQDN or IP: 50.198.192.141

Default: Add to New Zones

Transfer Type: Secure64 Authority

Server Type: Master

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

Username: SIXconnect

Password: .....

Port: 22

Remote Directory: /test12 where to put the zone files

Named Conf Path: /test12 path to zones within named.conf

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

Success!

### Bulk Zone Assignment

Search for Zone: [Match](#)

### DNS Functions

Manage Defaults and Options  
Edit DNS Record Types  
Generate all DS records for DNSSEC  
Generate zip file of all zones  
Increment All Serials  
DNS View ACL Management

### DNS Zone Transfers:

dns.6connect.net	0 Zones	<input type="checkbox"/>
services1.tcp0.com	0 Zones	<input type="checkbox"/>
ns1.sc2000.net	0 Zones	<input type="checkbox"/>
test.server	0 Zones	<input type="checkbox"/>
6connect Test Server	2 Zones	<input type="checkbox"/>
ns1.6clabs.com	0 Zones	<input type="checkbox"/>
ns2.6clabs.com	0 Zones	<input type="checkbox"/>
powerdns	2 Zones	<input type="checkbox"/>
Example Server	5 Zones	<input type="checkbox"/>
test	0 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.

6connect ProVision - © 2014 v5.0.0 - (6connect) ops@6connect.com

IPAM Admin DNS Admin Data Import Users API Templates Exit Admin Search or type help

### Manage DNS Servers

Server Name: Secure64 Test Server [Edit Server](#)

FQDN or IP: 50.198.192.141

Default: Add to New Zones

Transfer Type: Secure64 Authority

Server Type: Master

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

Username: SIXconnect

Password: .....

Port: 22

Remote Directory: /test12 where to put the zone files

Named Conf Path: /test12 path to zones within named.conf

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

Success!

### Bulk Zone Assignment

Search for Zone: [Match](#)

### DNS Functions

Manage Defaults and Options  
Edit DNS Record Types  
Generate all DS records for DNSSEC  
Generate zip file of all zones  
Increment All Serials  
DNS View ACL Management

### DNS Zone Transfers:

dns.6connect.net	0 Zones	<input type="checkbox"/>
services1.tcp0.com	0 Zones	<input type="checkbox"/>
ns1.sc2000.net	0 Zones	<input type="checkbox"/>
test.server	0 Zones	<input type="checkbox"/>
6connect Test Server	2 Zones	<input type="checkbox"/>
ns1.6clabs.com	0 Zones	<input type="checkbox"/>
ns2.6clabs.com	0 Zones	<input type="checkbox"/>
powerdns	2 Zones	<input type="checkbox"/>
Example Server	5 Zones	<input type="checkbox"/>
test	0 Zones	<input type="checkbox"/>

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

### 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 ProVision – © 2014 v5.0.0 – (6connect) ops@6connect.com

IPAM Admin ▾ DNS Admin Data Import Users API Templates Exit Admin Search or type help

### Manage DNS Servers

Server:  **New Server**

### Bulk Zone Assignment

Search for Zone:  
 **Match**

Matched Zones:  
[atestzone.com](#)  
[dane.test.dnservices.co.za](#)  
[eohns.test.dnservices.co.za](#)  
[eohns1.test.dnservices.co.za](#)  
[eoh-ns.test.dnservices.co.za](#)  
[mydomain.test.dnservices.co.za](#)  
[Testzone2.com](#)

Assign to:  as  **Assign**

### Bulk Record Changes

**WARNING. This is a power user tool.**

Record Host  Record Type  Record Value  ☐ Strict Comparison **Search Records**

### DNS Functions

Manage Defaults and Options  
 Edit DNS Record Types  
 Generate all DS records for DNSSEC  
 Generate zip file of all zones  
 Increment All Serials  
 DNS View ACL Management

### DNS Zone Transfers:

dns.6connect.net	0 Zones	<input type="checkbox"/>
services1.tcp0.com	0 Zones	<input type="checkbox"/>
ns1.sc2000.net	0 Zones	<input type="checkbox"/>
test.server	0 Zones	<input type="checkbox"/>
6connect Test Server	2 Zones	<input type="checkbox"/>
ns1.6clabs.com	0 Zones	<input type="checkbox"/>
ns2.6clabs.com	0 Zones	<input type="checkbox"/>
powerdns	2 Zones	<input type="checkbox"/>
Example Server	5 Zones	<input type="checkbox"/>
test	0 Zones	<input type="checkbox"/>
Secure64 Test Server	0 Zones	<input 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

6connect ProVision – © 2014 v5.0.0 – (6connect) ops@6connect.com

IPAM Admin ▾ DNS Admin Data Import Users API Templates Exit Admin Search or type help

### Manage DNS Servers

Server:  **New Server**

### Bulk Zone Assignment

Search for Zone:  
 **Match**

Matched Zones:  
[atestzone.com](#)  
[Testzone2.com](#)

Assign to:  as  **Assign**

### Bulk Record Changes

**WARNING. This is a power user tool.**

Record Host  Record Type  Record Value  ☐ Strict Comparison **Search Records**

### DNS Functions

Manage Defaults and Options  
 Edit DNS Record Types  
 Generate all DS records for DNSSEC  
 Generate zip file of all zones  
 Increment All Serials  
 DNS View ACL Management

### DNS Zone Transfers:

dns.6connect.net	0 Zones	<input type="checkbox"/>
services1.tcp0.com	0 Zones	<input type="checkbox"/>
ns1.sc2000.net	0 Zones	<input type="checkbox"/>
test.server	0 Zones	<input type="checkbox"/>
6connect Test Server	2 Zones	<input type="checkbox"/>
ns1.6clabs.com	0 Zones	<input type="checkbox"/>
ns2.6clabs.com	0 Zones	<input type="checkbox"/>
powerdns	2 Zones	<input type="checkbox"/>
Example Server	5 Zones	<input type="checkbox"/>
test	0 Zones	<input type="checkbox"/>
Secure64 Test Server	0 Zones	<input type="checkbox"/>

Push Zones to Checked Servers: **Push**

### Nameserver Management

Select the **Select Server** and as **Master** dropdowns and **Assign** the above zones to this server.

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

Under DNS Zone Transfers, verify the server and the zones to transfer. To view the zone names, click on the # Zones link next to the server.

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

The screenshot shows the 6connect ProVision DNS Admin interface. The top navigation bar includes links for IPAM Admin, DNS Admin, Data Import, Users, API, Templates, and Exit Admin. The main content area is divided into several sections:

- Manage DNS Servers:** Includes a 'Select Server' dropdown and a 'New Server' button.
- Bulk Zone Assignment:** Includes a 'Search for Zone' field with 'Testzone' entered, a 'Match' button, a list of 'Matched Zones' (atestzone.com, Testzone2.com), an 'Assign to' dropdown, a 'Master' checkbox, and an 'Assign' button. A green 'Success!' message is displayed below.
- Bulk Record Changes:** Includes a warning message, fields for 'Record Host', 'Record Type', and 'Record Value', a 'Strict Comparison' checkbox, and a 'Search Records' button.
- Nameserver Management:** A section at the bottom.
- DNS Functions:** A sidebar with links for 'Manage Defaults and Options', 'Edit DNS Record Types', 'Generate all DS records for DNSSEC', 'Generate zip file of all zones', 'Increment All Serials', and 'DNS View ACL Management'.
- DNS Zone Transfers:** A table listing servers and their associated zones. The 'Secure64 Test Server' is highlighted with a red circle around its '2 Zones' and a checked checkbox. Below the table is a 'Push Zones to Checked Servers' button, also circled in red.

The system will present the following live progress bar.

The screenshot shows a live progress bar window. The title bar reads 'Pushing Server 1 of 1: 50.198.192.141'. Below the title bar is a blue progress bar. The main content area displays the text 'Transferring Zone File Testzone2.com.zone...'. The window has a close button in the top right corner.

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 50.198.192.141
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: atestzone.com

zonefile: /test12/6connectGeneric/m/atestzone.com.zone

zone:

name: Testzone2.com

zonefile: /test12/6connectGeneric/m/Testzone2.com.zone
```

In the example above, two 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 atestzone.com.zone
6758 2013-08-21 15:35:02 Testzone2.com.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.



### Using dig to validate your Secure64 Server installation

```
[authdnsadmin@eval138.secure64.com]# dig @50.198.192.141 atestzone.com
; <<>> DiG SourceT 3.x <<>> @50.198.192.141 atestzone.com
;; Got answer:
;; >>HEADER<< opcode: QUERY, status: NOERROR, id: 59591
;; flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0
;; QUESTION SECTION:
;atestzone.com. IN A
;; AUTHORITY SECTION:
atestzone.com. 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>		
Delimiter: <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:

**Views:**

**Add a New View**

View Name:

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:

Description:

**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:

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
T

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.

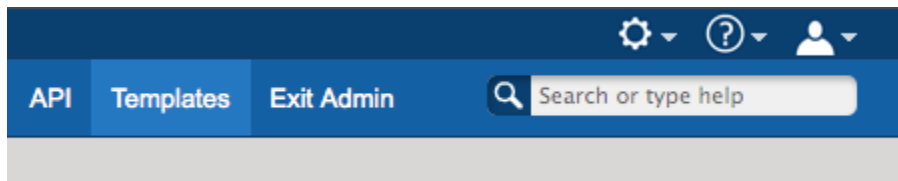
## Configuring DNS Templates

### 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:

DNS Templates				
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**

**SOA Record**

Serial	Refresh	Retry	Expiry	Minimum
<input type="text"/>	<input type="text" value="14400"/>	<input type="text" value="3600"/>	<input type="text" value="604800"/>	<input type="text" value="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	
<input type="text" value="1.2.3.4"/>	<input type="text"/>	<input type="text" value="A"/>	<input type="text"/>	<input type="text" value="cnn.com."/>	  
<input type="text" value="8.8.8.8"/>	<input type="text"/>	<input type="text" value="A"/>	<input type="text"/>	<input type="text" value="www"/>	  
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

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.

## DNS

New DNS Zone



-- no template --

Create Zone

### Zone Delegation

Delegated Zone

Slave IP

Zone name



IPv4 or IPv6

-- no template --

Anna's Template

Demo Template

Equinix

testing

VM Turnup

Add Slave

# 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!

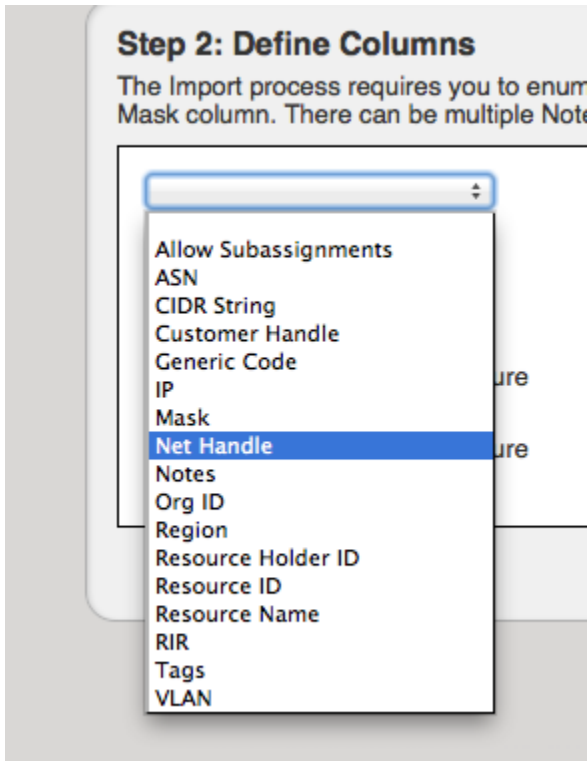
## Step 2: Prep your Data

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:




### Allow Subassignments option

When importing the field "Allow Subassignments" - the parameters accepted are "TRUE", "1", "Y", "yes"

## 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).

 IPAM Admin ▾ DNS Admin Data Import Users API Templates Exit Admin

Search or type help

**Resource Import:**

- Simple Upload/Import from CSV
- Resource Import Tool \*Beta\*

**Import Templates:**

- All Import Samples
- IP Import Sample File
- Customer Import Sample File

**Peering Import**

- Import BGP Sessions

**IP Import:**

- Upload/Import from CSV
- Import from RIR

**DNS Import:**

- BIND Zone Upload/Import
- PowerDNS Zone Import
- DynECT Zone Import

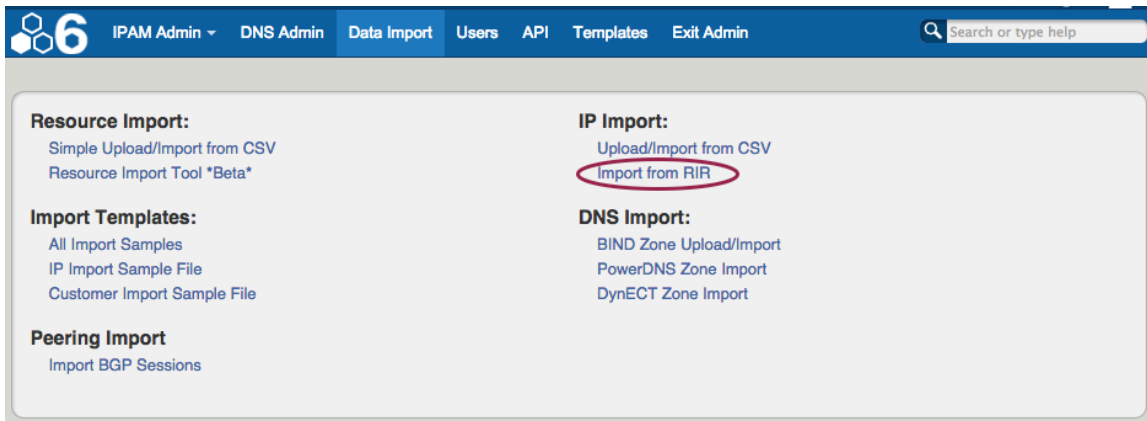
For more details, see:

- [Import Aggregate Blocks](#)
- [Import DNS Zones](#)
- [Resource Import Tool](#)

# Import Aggregate Blocks

## Import Aggregates

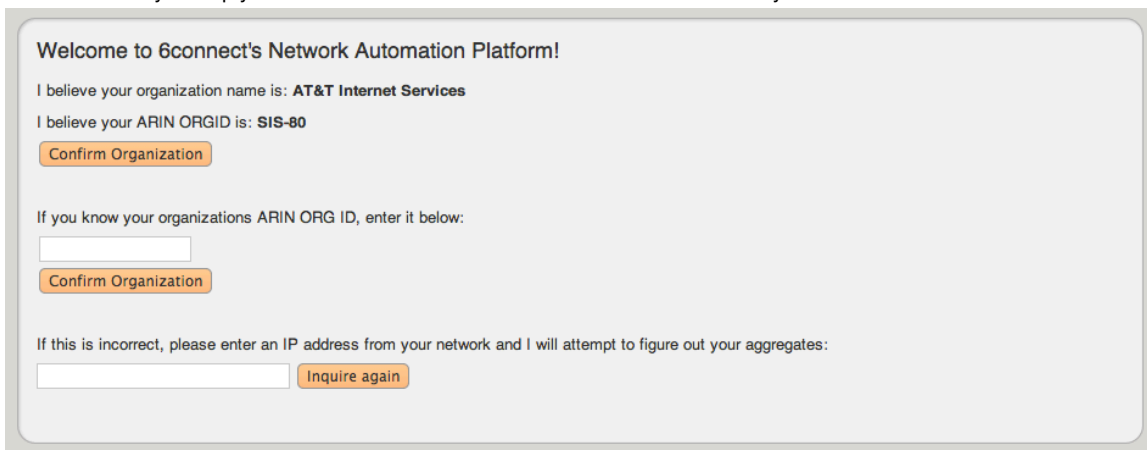
Navigate to the **Data Import Tab** from the **Admin button** to import your aggregate blocks. Select "Import from RIR" under "IP Import".



The screenshot shows the 6connect Admin interface with the 'Data Import' tab selected. The interface is divided into four main sections: Resource Import, IP Import, Import Templates, and DNS Import. The 'IP Import' section is highlighted, and the 'Import from RIR' option is circled in red. The 'Resource Import' section includes 'Simple Upload/Import from CSV' and 'Resource Import Tool \*Beta\*'. The 'Import Templates' section includes 'All Import Samples', 'IP Import Sample File', and 'Customer Import Sample File'. The 'DNS Import' section includes 'BIND Zone Upload/Import', 'PowerDNS Zone Import', and 'DynECT Zone Import'. The 'Peering Import' section includes 'Import BGP Sessions'.

### Step 1: Lookup from Source IP

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





The screenshot shows the 6connect Network Automation Platform welcome screen. It displays the following information: 'Welcome to 6connect's Network Automation Platform!', 'I believe your organization name is: AT&T Internet Services', 'I believe your ARIN ORGID is: SIS-80', and a 'Confirm Organization' button. Below this, it asks 'If you know your organizations ARIN ORG ID, enter it below:' and provides a text input field and a 'Confirm Organization' button. At the bottom, it says 'If this is incorrect, please enter an IP address from your network and I will attempt to figure out your aggregates:' and provides a text input field and an 'Inquire again' button.

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.


Dashboard Resources DNS DHCP IPAM Peering Log Reporting

 Search or type help

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

**This is IPv6 & IPv4 non-1918 space I have discovered**

Found IPv4 block: 104.48.0.0/12	ARIN	✓
Found IPv4 block: 208.188.0.0/14	ARIN	Add Aggregate
Found IPv4 block: 207.193.0.0/16	ARIN	Add Aggregate
Found IPv4 block: 209.184.0.0/16	ARIN	Add Aggregate
Found IPv4 block: 216.60.0.0/14	ARIN	Add Aggregate
Found IPv4 block: 63.170.248.0/25	ARIN	Add Aggregate
Found IPv4 block: 64.216.0.0/14	ARIN	Add Aggregate

**If you will be using RFC1918 space, you will likely want to add from this list:**

RFC1918 block: 10.0.0.0/8	1918	10.0.0.0/8	✓
RFC1918 block: 192.168.0.0/16	1918	192.168.0.0/16	✓
RFC1918 block: 172.16.0.0/12	1918	Add Aggregate	

**If you will be using Shared Transition Space, add:**

RFC6598 block: 100.64.0.0/10	6598	Add Aggregate
------------------------------	------	---------------

### Step 3: Customizing

With your aggregates added, you are now ready to customize the tool and import additional data! From here, you can manage your aggregates under the IPAM tab, edit administration functions under **IPAM Admin**, or import resources using the [Resource Import Tool](#).



## Import DNS Zones

### Importing DNS Zones

ProVision offers three DNS zone import options, available under the [Data Import](#) tab in the [Admin](#) section:

#### BIND Zone Import

- Imports using the named.conf configuration file tied to the zones you are uploading, a .zip or .tar file of the zones themselves, and an optional .csv file mapping zones to customers and DNS Servers.

#### DynECT Zone Import

- Imports and syncs ALL zones on the system with those in your DynECT 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.

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.

- Importing DNS Zones
  - Preparing your DNS Zones for Import
  - Importing your DNS Zones (BIND)
    - Step 1: Create a new DNS Import Job
    - Step 2: Map Data Columns (Optional)
    - Step 3: Reviewing Data
  - Importing your DNS Zones (PowerDNS)

### 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"; };
```

This configuration file can be taken directly from the DNS server, and can be in either ISC BIND or NSD format. The system auto-detects which one is being supplied.

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 simple 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, 174.23.14.4, 174.23.14.9
my-otherzone.com,test-02,great stuff, dns1.dns.net, dns2.dns.net
even-reverse-zones.arpa,test-03,amazing stuff
```

Note the columns are the "Zone Name", the "Resource ID", "Notes", "Master Server", "Slave Server"



### Importing DNS Server Linkages

When importing zones, you can use the "Master Server" and "Slave Server" columns to assign zones to specified DNS Servers. Please note that the IP address or FQDN of the DNS Server is supported in this field.



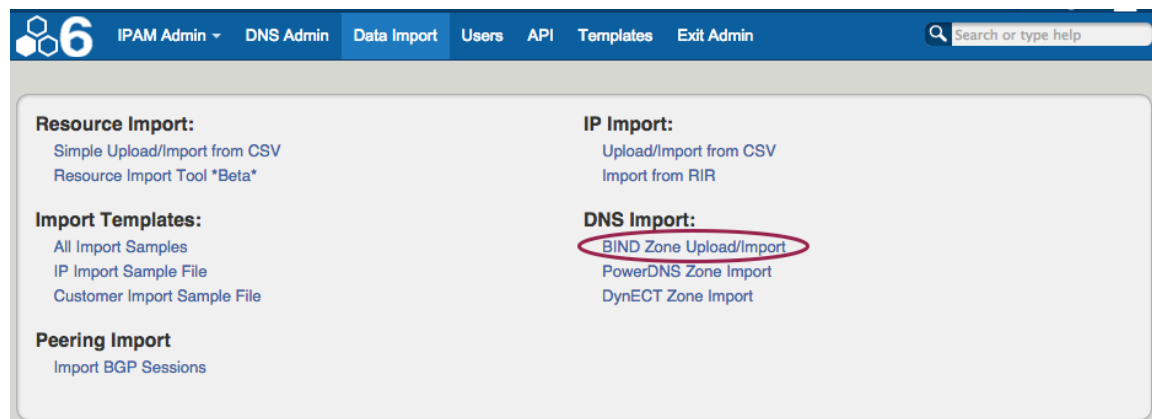
To successfully map to a DNS server, that server must already exist within Provision.

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

## Importing your DNS Zones (BIND)

### Step 1: Create a new DNS Import Job

Navigate to the [Data Import](#) Tab from the [Admin](#) button to import your data. Select "BIND Zone Upload/Import" under "DNS Import".



Create a Job Name and Description for the import. This is especially useful to keep track of progress in cases the data arrives from multiple sources, or will require multiple stages of manual review. Select the appropriate Configuration File (required), Archive File (required), and CSV File (optional) that you prepared above by selecting the "Choose File" button(s) under each section, and browsing to the correct file location. Then hit "Start Import".

**DNS Zone Import**

**New Import**

The DNS Import accepts an archive file of zones (ZIP or TAR) in both flat and hierarchical formats. You may also submit a CSV file mapping zone names to customer ids and DNS servers. Please make sure the archive file has an appropriate file extension, and that all files are encoded in UTF-8.

**Job Name:** 
**Description:**

**Configuration File:**  
**Archive File:**  
**CSV File:**

Required: a configuration file in BIND or NSD format.

Required: a ZIP or TAR of your zones.

Optional: a CSV file mapping zones to customers and DNS Servers.



### Working with Large or Multiple Data Sets



Although you cannot add new files to an existing job, for jobs with multiple sources for data (which may have different formatting), you can simply create separate jobs and descriptions for each source - no need to manually combine the data into one file before importing. The Import tool's mapping and editing functions will allow for the data to be reconciled in ProVision.

For large data sets where multiple stages of manual review might be needed, you can create a new job using the same set of data files in order to work in parallel on a different portion of the data.

After importing, the new job will appear under the "Existing Jobs" section. To continue working with this job, select it from the list and the next step will appear on the page.

**DNS Zone Import**

**Existing Jobs**

Sample DNS Import 1 last modified 12-09-2014 11:18 AM  

**New Import**

The DNS Import accepts an archive file of zones (ZIP or TAR) in both flat and hierarchical formats. You may also submit a CSV file mapping zone names to customer ids and DNS servers. Please make sure the archive file has an appropriate file extension, and that all files are encoded in UTF-8.

**Job Name:** 
**Description:**

## Step 2: Map Data Columns (Optional)

If you chose to load an optional match CSV file to assign DNS Zones to Resource, a mapping step will be available. Otherwise, proceed to Step 3: Reviewing Data.

For DNS imports, four column definitions are available: **Zone**, **Resource ID**, **Server Master IP**, and **Server Slave IP**. Using the dropdown menu, select the appropriate definition for each of the imported columns. **Zone** and **Resource Holder ID** should each only have a single column selected, however, any number of columns may be defined as **Server Master IP** or **Server Slave IP**. Other columns which do not apply under the available definitions should be left as blank, and will be skipped during the upload process.

When completed, hit "Next".

### Define Columns

The Import process requires you to enumerate the function of the columns in the provided CSV.

Zone	Resource Holder ID		Server Master IP	Server Slave IP
Zone Name	Resource Id	Notes	Master Server	Slave Server
citi.com	test-01	fun stuff	208.39.106.184	
citibank.com	test-02	great stuff	208.39.106.99	208.39.106.184
citigroup.com	test-03	amazing stuff	208.39.106.184	208.39.106.82

Next

### Step 3: Reviewing Data

After supplying the file set and defining columns (if applicable), a review step is provided. The configuration file is broken into individual jobs, scanned for errors, and shown by row (in batches of 100) to be reviewed. Zones with errors will show as color coded. and can be filtered to be viewed by All, Valid, Warnings, Invalid, or Ignored. From here, the zone can be edited or ignored.

### Review Data

Please review the data for correctness. Invalid and ignored rows will be skipped.

View: All Valid Warnings Invalid Ignored Hide

**Zone:** citi.com
 **Resource Holder:** test-01
 Edit Ignore

**Zone:** citibank.com
 **Resource Holder:** test-02
 A specified DNS Server does not exist.
 Edit Ignore

**Zone:** citigroup.com
 **Resource Holder:** test-03
 A specified DNS Server does not exist.
 Edit Ignore

#### Import Data

When you have reviewed the data import job for accuracy, hit the Execute Import button. All rows which are disabled, invalid, have warnings, or were previously successful will be passed over. Successful import rows will be marked as such.

Execute Import

Editing the zone provides options to alter the Resource Holder, enable DNS servers, and redefine Master and Slaves.

After editing, hit "Save", and continue reviewing / editing data as desired.

View: All Valid Warnings Invalid Ignored Hide

**Zone Name:** citibank.com
 **Resource Holder:** test-02
 View Save

DNS Servers:	Enabled	Server Name	Master	Slave
<input type="checkbox"/>	dns.6connect.net (dns.6connect.net)	<input type="radio"/>	<input type="radio"/>	
<input type="checkbox"/>	services1.tcp0.com (services1.tcp0.com)	<input type="radio"/>	<input type="radio"/>	
<input type="checkbox"/>	ns1.sc2000.net (ns1.sc2000.net)	<input type="radio"/>	<input type="radio"/>	
<input type="checkbox"/>	test.server (192.168.1.234)	<input type="radio"/>	<input type="radio"/>	
<input checked="" type="checkbox"/>	6connect Test Server (208.39.106.184)	<input type="radio"/>	<input checked="" type="radio"/>	
<input type="checkbox"/>	ns1.6clabs.com (ns1.6clabs.com)	<input type="radio"/>	<input type="radio"/>	
<input type="checkbox"/>	ns2.6clabs.com (ns2.6clabs.com)	<input type="radio"/>	<input type="radio"/>	

When the review step is completed, hit the "Execute Import" button. A progress bar will appear to show progress and note errors if they occur.

When the bar reaches 100%, the import is complete.

**Import Data**

When you have reviewed the data import job for accuracy, hit the Execute Import button. All rows which are disabled, invalid, have warnings, or were previously successful will be passed over. Successful import rows will be marked as such.

Execute Import

Current Block: Finished!

## Importing your DNS Zones (PowerDNS)

### Step 2: Import your PowerDNS zones

To import PowerDNS zones, first ensure the PowerDNS server has been set up under DNS Admin - [Manage DNS Servers](#).

Once server setup has been verified, navigate to the [Data Import](#) Tab in the [Admin](#) section. Select the "Power DNS Zone Import" link.

To import your data, simply choose your PowerDNS server and click "Import".

This operation will pull all zones on the target server.

This operation may take quite some time.

Choose a server: 208.39.104.106

Import

# Resource Import Tool

## Importing Resources

### The Resource Import Tool

The Resource Import Tool (in beta) allows you to import resource data from a .csv file into ProVision. In the Resource Import Tool, you can open one or more user-created .csv spreadsheets, perform basic editing functions if needed, associate the data to a specific Section, and correlate the data columns to specific Section Fields.

In ProVision, since Resources can be any desired entity, and Sections can be anything from "customers" to "firewalls" to "racks", you have total flexibility in what type of data to import with the Resource Importer to meet your specific company needs. Check out [Working With Resources](#), [Customizing Sections](#), and [Customizing Fields](#) for more details on how to fit these elements to your business.

### Step 0: Before You Begin

There are a few items that you will need have set up prior to using the Resource Importer Tool. Ensure that you have:

- The .csv document you wish to import saved with UTF-8 encoding. Windows, Mac, and Linux type .csv files are supported.
- A header row for the data in the .csv.
- The .csv file should be "clean", that is, only contain the data to be imported and a header row for that data.
- A Section created in ProVision with fields that correlate to the import data. For example, if you wish to import a list of contact information, there will need to be a Section in ProVision created for "Contacts", with fields such as "First Name", "Last Name", "email address", "Phone number", and so on. To create a new Section, or edit an existing Section, refer to [Working With Resources](#), [Customizing Sections](#), and [Customizing Fields](#).



If the above preconditions are not met, the Resource Importer Tool may not be able to correctly read the .csv file or complete the import. Verify UTF-8 .csv encoding, a clean dataset with a header row, and that an appropriate Section exists in ProVision prior to import.

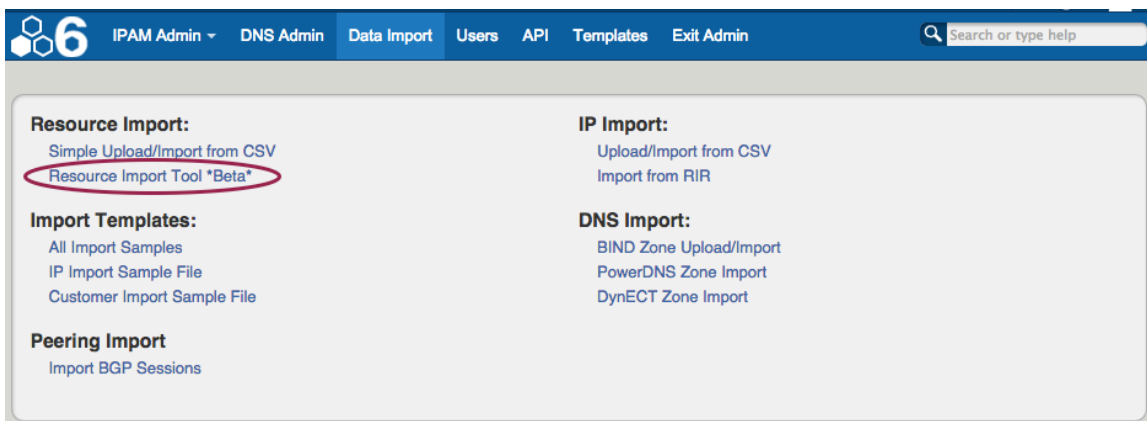


#### Best Practice

To ensure a fast and straightforward resource import, best practice is to verify ahead of time that your .csv data is correct and contains all the necessary column information for the Section. This includes a top-level Name and Unique ID, as well as a column per Section field. Data edits and column adjustments can be performed inside the Resource Importer Tool if necessary, but will require additional time and steps.

### Opening the Resource Import Tool

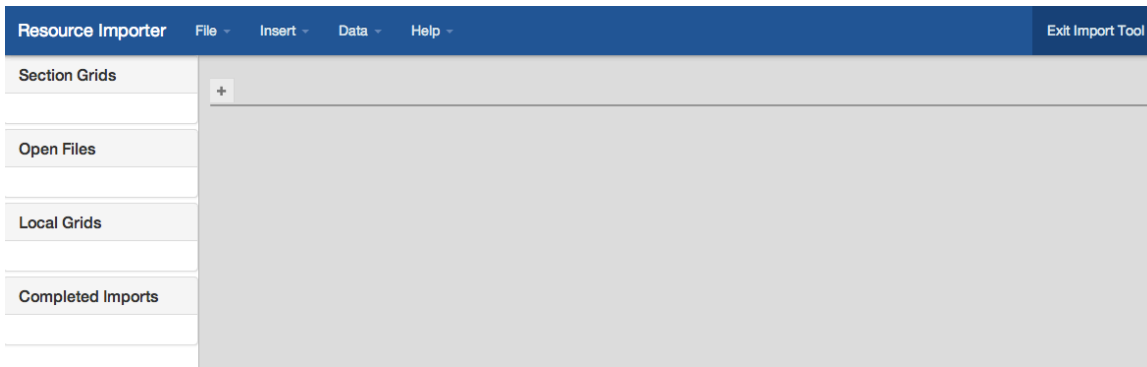
To open the Resource Import Tool, navigate to the [Data Import Tab](#) from the [Admin button](#) to import your aggregate blocks. Select "Resource Import Tool" under "Resource Import".



## The Resource Import Tool UI

When you first open the Resource Importer, you will be given the option to view a short on-screen guide to using the tool. After stepping through

the guide and/or exiting out of it, the tool will look like this:



On the top are standard menu options of "File", "Insert", and "Data" and "Help". Under those menus, you may see greyed-out functions listed. Those functions are items under development, or not available to use at the current Importer step.

On the left side of the screen is a listing of currently opened files:

**Sections Grids** lists grids currently open that were created from a ProVision Section

**Open Files** lists the current user created .csv spreadsheets that are open

**Local Grids** lists any grids that were created in the tool itself, instead of opened from an external file

**Completed Imports** show imports which have been completed and imported into ProVision

If, at any time, you need to leave the Resource Importer Tool, select the "Exit Import Tool" in the top right corner of the screen, and you will be taken back to the ProVision Dashboard.



Exiting the Resource Importer Tool prior to completing the import process will result in the current open grids being discarded.

### ***Resource Importer Walkthrough***

For a step by step walkthrough of the Resource importer, continue on to the Resource Importer Walkthrough , which shows how to import a sample contact list and perform minor editing tasks.

[Resource Importer Walkthrough - Step 1 Upload your .csv data file](#)

[Resource Importer Walkthrough - Step 2 Open a Template Grid from an existing Section](#)

[Resource Importer Walkthrough - Step 3 Reorder .csv columns to match the Section Grid column order](#)

[Resource Importer Walkthrough - Step 4 Edit Data as Needed](#)

[Resource Importer Walkthrough - Step 5 Drag rows from the .csv Grid to the Section Grid](#)

[Resource Importer Walkthrough - Step 6 Import into ProVision](#)

## Resource Importer Walkthrough - Step 1

### Importing Resources

#### Before You Begin

Ensure that you are familiar with the overview and "Before you Begin" requirements listed on the [Resource Import Tool](#) page.

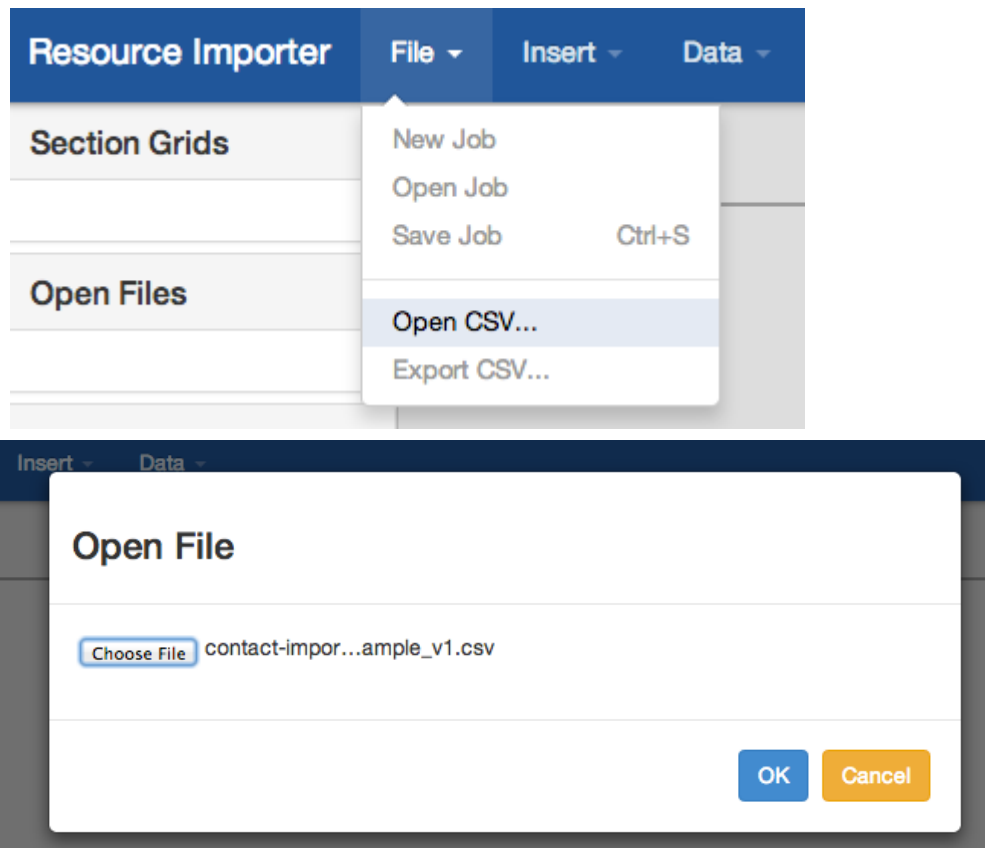
For this tutorial, we will be using the Contact Import Sample .csv available on the [Import Templates](#) page as an example, and associating it to an existing Section called "Contact" having the fields: First Name, Last Name, Email, 2nd Email, Phone, 2nd Phone, Mobile Phone, Role, and Time Zone. To create this Section, or edit an existing Section, refer to [Working With Resources](#), [Customizing Sections](#), and [Customizing Fields](#).

In order to illustrate the abilities of the Resource Importer to edit data and adjust for formatting issues, the Contact Import Sample .csv is used intentionally leaving a few less-than-ideal conditions (much like you may encounter in real life) such as leaving typos, having an extra data column, and missing a needed column. If you follow the "Before you Begin" requirements and "Best Practice" notes, however, you may be able to skip any editing or column adjustment steps.

When you are ready to begin, open the Resource Importer and proceed to Step 1.

### Step 1: Upload your .csv data file

Under the "File" Menu, select "Open .csv". Browse to and select your UTF-8 encoded data file.



After hitting "OK", your file should be visible in the workspace, as well as listed under "Open Files" like this:



Resource Importer

FileInsertDataHelp

Exit Import Tool

Section Grids

Open Files

[contact-import-sample\\_v1.csv](#)

Local Grids

Completed Imports

contact-import-sample\_v...

#	<input type="checkbox"/>	Unique ID	First Name	Last Name	Title	email	email2	Phone	Phone
0	<input type="checkbox"/>	6c-004	Aaron	Hughes	CTO	aaron@connect...	support@6conne...	1-408-555-1212	1-408-555-1212
1	<input type="checkbox"/>	6c-004	John	Parker	Sales	john@gmail.com		234.634.1234	888-cal
2	<input type="checkbox"/>	6c-004	Tom	Taylor	Janitor	ttaylor@toms.com		503-555-1256	866-55
3	<input type="checkbox"/>	6c-007	Bob	Smith	VP Ops	bsmith@apple.com		888-call-now	703-55
4	<input type="checkbox"/>	6c-008	Maurice	Carmichael	Marketing	mc@mail.com		866-555-1134	888-nic
5	<input type="checkbox"/>	6c-009	Vince	Bunch	Marketing	vbunch@happyp...	ops@happyplace...	703-555-1111	234-55
6	<input type="checkbox"/>	6c-010	Mark	Tompson	Product Manager	tompson@tlnet		888-nice-wor	354-55
7	<input type="checkbox"/>	6c-011	Herold	Waters	Engineer	hwaters@is.co.uk		234-555-6678	17 145
8	<input type="checkbox"/>	6c-012	Michael	Sanders	Project Manager	pm@mybusiness...		354-555-1235	234-23
9	<input type="checkbox"/>	6c-013	Jill	Keller	Operations	jill.keller@anothe...		17 145 125124	44 123
10	<input type="checkbox"/>	6c-014	Sarah	Campbell	Account Executive	sa.camp@intel.net		234-234 1234	888-cal
11	<input type="checkbox"/>	6c-015	Amanda	Kingston	Sales	akingston@sellin...		44 123 555 12	866-55

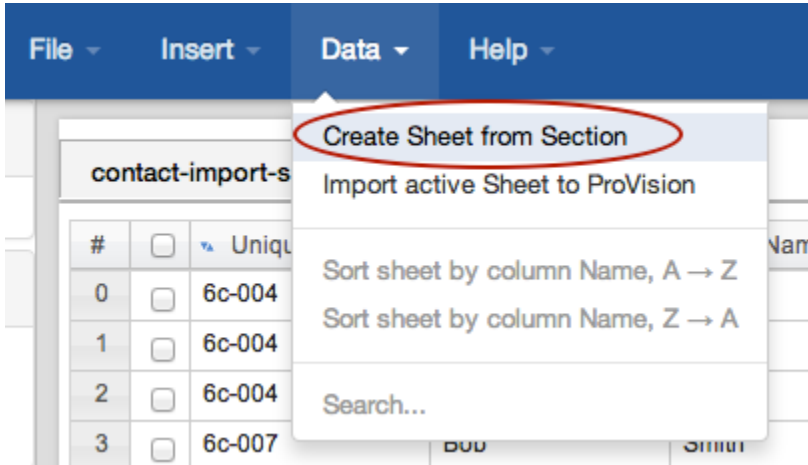
After opening your .csv grid, proceed to [Step 2 - Open a template grid from an existing Section](#)

## Resource Importer Walkthrough - Step 2

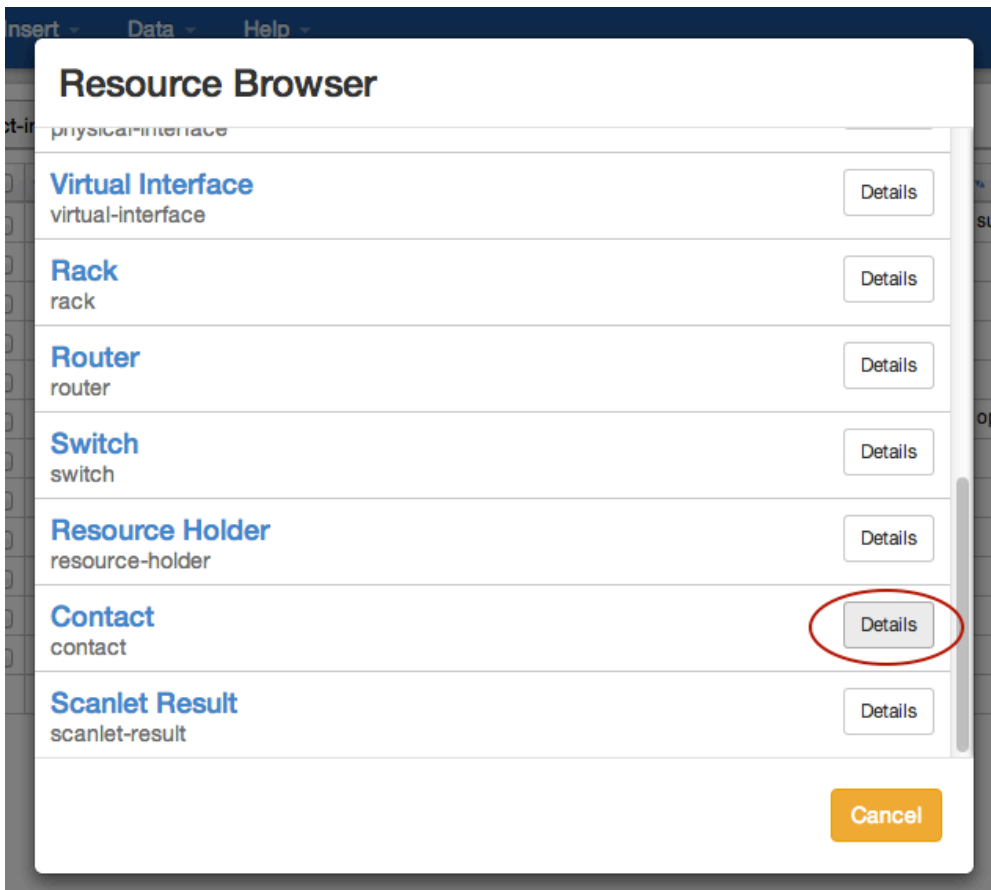
### Importing Resources

#### Step 2: Open a Template Grid from an existing Section

Under the "Data" menu, select "Create Sheet from Section".

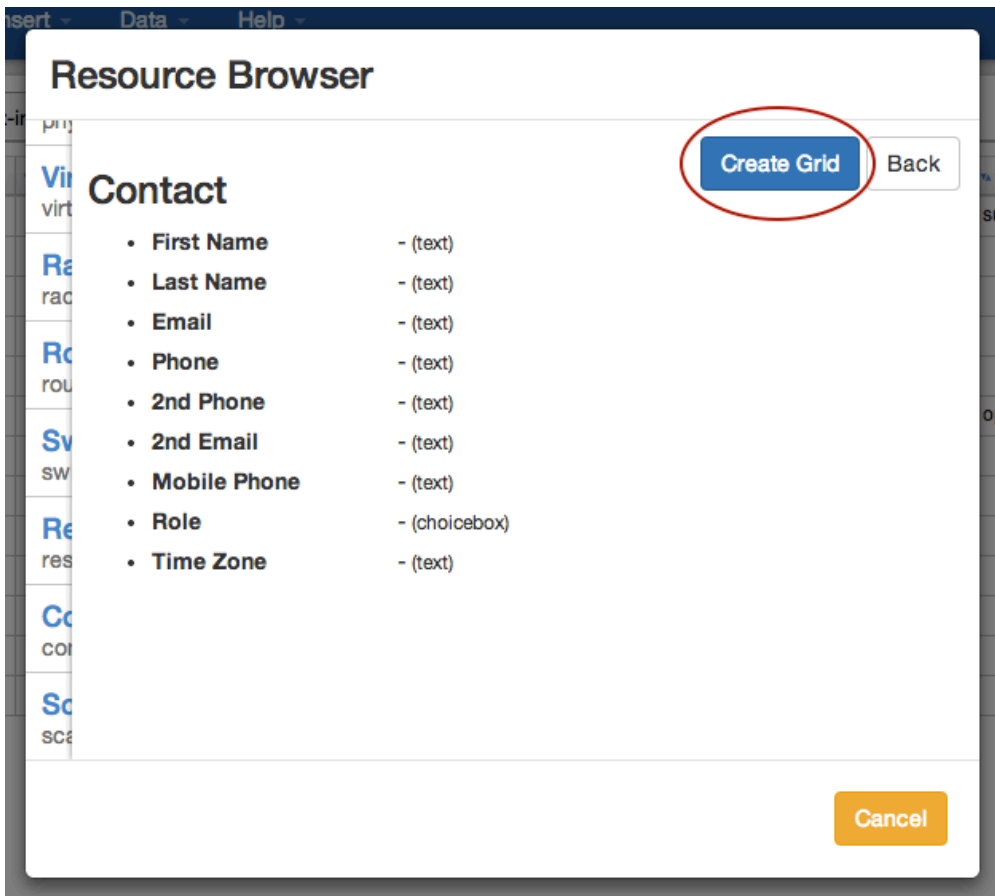


The Resource Browser will pop up, showing the list of Sections currently available in Provision. Clicking on the "Details" button will show the fields for that Section.

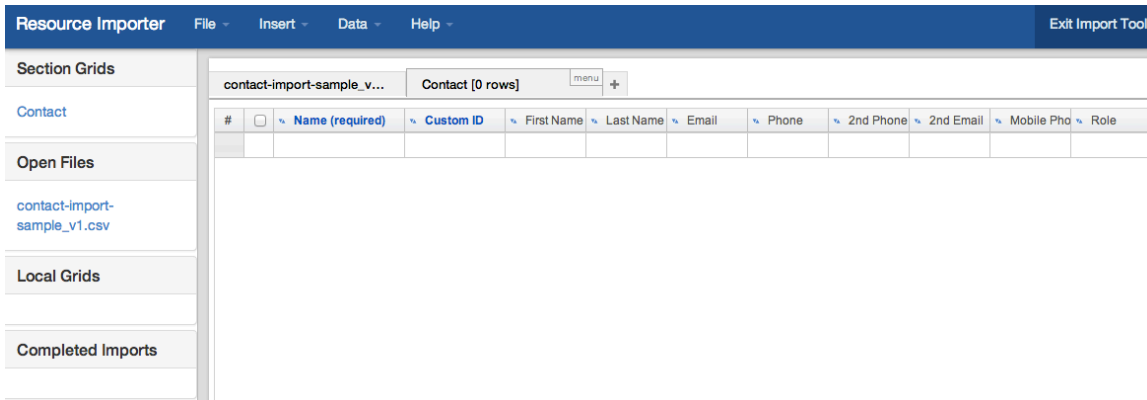


Verify that the Section and available fields match the type of data you are trying to import. In this case, the Section "Contact" has the fields that correlate to our spreadsheet data.

Select "Create Grid" to create a grid based off this Section.



When the Section Grid has been created, required fields will show in blue font with (required) after the header, in this case, "Name" is a required field. The "Custom ID" field is metadata allowing for a unique ID to be associated with each entry, but is not necessary for a successful import. The remainder of the headers directly match the Section's fields.



After you have opened your Section Grid, proceed to [Step 3: Reorder .csv columns to match the Section Grid](#)

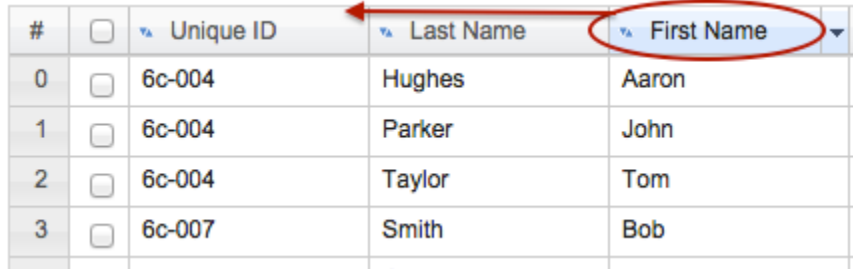
## Resource Importer Walkthrough - Step 3

### Importing Resources

#### Step 3: Reorder .csv columns to match the Section Grid column order

One of the most important steps is to reorder the columns from the .csv data to match the order of the Section Grid headers - think of the importer as copying and pasting the csv data into the "Contact" Section grid- we want to ensure that the data is under the correct headers!

*Click on the column header to Drag and Drop to the desired location:*



#	<input type="checkbox"/>	Unique ID	Last Name	First Name
0	<input type="checkbox"/>	6c-004	Hughes	Aaron
1	<input type="checkbox"/>	6c-004	Parker	John
2	<input type="checkbox"/>	6c-004	Taylor	Tom
3	<input type="checkbox"/>	6c-007	Smith	Bob

Click back and forth between the tabs to verify the column order, then click on a header and drag and drop into the desired order. This moves not only the header, but also the data below it.

#### Common Column Editing Questions

What if just my column headers are in the wrong place?

What if I have too many / too few columns in my .csv to match the Section Grid?

If you see any of these issues, proceed to [Step 4 - Edit data as needed](#).

Otherwise, if your columns match up perfectly and none of the data needs editing, skip to [Step 5 - Drag rows to the Section Grid](#)

## Resource Importer Walkthrough - Step 4

### Importing Resources

#### Step 4: Edit data as needed

As you may have noticed in Step 3, with this example we have a couple of columns that don't quite match up to the Section Grid. The "Title" column in the .csv data is an additional column we are not tracking in our Section. Also, although we have a "First Name" and "Last Name", we are missing a data column for the top-level "Name" required in the Section Grid.

#### Common Editing Questions:

- What if I have too many / too few columns in my .csv to match the Section Grid?
- What if I see a typo in the .csv data?
- What if just my column headers are in the wrong place?

#### To hide extraneous column information:

Right click on a header and deselect the check box for the column you wish to hide. In this case, we want to hide "Title".

contact-import-sample_v...		Contact [0 rows]			
#	Unique ID	First Name	Last Name	email	Phone
0	6c-004	Aaron	Hughes	aaron@6connec	1-408-555-1212
1	6c-004	John	Parker	john@gmail.com	234.634.1234
2	6c-004	Tom	Taylor	ttaylor@toms.co	503-555-1256
3	6c-007	Bob	Smith	bsmith@apple.c	888-call-now
4	6c-008	Maurice	Carmichael	mc@mail.com	866-555-1134
5	6c-009	Vince	Bunch	vbunch@happyp	703-555-1111
6	6c-010	Mark	Tompson	tompson@tt.net	888-nice-wor
7	6c-011	Herold	Waters	hwaters@is.co.u	234-555-6678
8	6c-012	Michael	Sanders	pm@mybusiness	354-555-1235
9	6c-013	Jill	Keller	jill.keller@anoth	17 145 125124
10	6c-014	Sarah	Campbell	sa.camp@intel.n	234-234 1234
11	6c-015	Amanda	Kingson	akingston@sellin...	44 123 555 12

#### To Edit Data in the Resource Importer

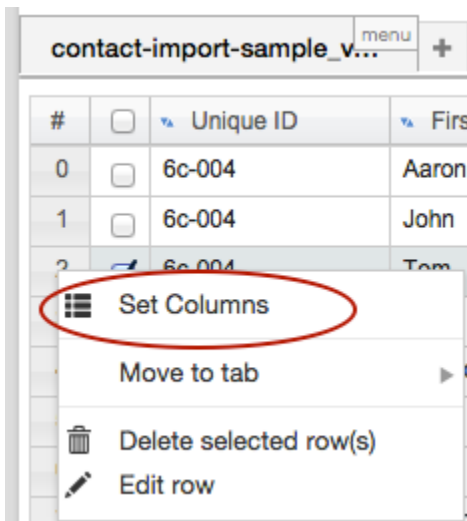
Data in the grids can be edited directly by clicking on the cell(s). In our example, we can see that "Amanda Kingson" should really be "Amanda Kingston". Let's fix that! Click in the cell, type in the edit you wish to make, and then click outside of the cell to exit edit mode. To edit a full row of data, you can right click on the row, select "Edit" row, and make multiple changes in the form box.

contact-import-sample_v...		Contact [0 rows]					
#	<input type="checkbox"/>	Unique ID	First Name	Last Name	email	email2	Phone
0	<input type="checkbox"/>	6c-004	Aaron	Hughes	aaron@6connect...	support@6conne...	1-408-555-1212
1	<input type="checkbox"/>	6c-004	John	Parker	john@gmail.com		234.634.1234
2	<input type="checkbox"/>	6c-004	Tom	Taylor	ttaylor@toms.com		503-555-1256
3	<input type="checkbox"/>	6c-007	Bob	Smith	bsmith@apple.com		888-call-now
4	<input type="checkbox"/>	6c-008	Maurice	Carmichael	mc@mail.com		866-555-1134
5	<input type="checkbox"/>	6c-009	Vince	Bunch	vbunch@happypa...	ops@happyplace...	703-555-1111
6	<input type="checkbox"/>	6c-010	Mark	Tompson	tompson@tt.net		888-nice-wor
7	<input type="checkbox"/>	6c-011	Herold	Waters	hwaters@is.co.uk		234-555-6678
8	<input type="checkbox"/>	6c-012	Michael	Sanders	pm@mybusiness....		354-555-1235
9	<input type="checkbox"/>	6c-013	Jill	Keller	jill.keller@another...		17 145 125124
10	<input type="checkbox"/>	6c-014	Sarah	Campbell	sa.camp@intel.net		234-234 1234
11	<input checked="" type="checkbox"/>	6c-015	Amanda	Kingston	akingston@sellin...		44 123 555 12

*If a column header is over the wrong data:*

If just the header is in the wrong spot (doesn't match the data below it), you can move just the column header in the Resource Importer, without moving the data below it.

1) Right click on a row of the grid to edit and select "Set Columns":



2) In the "Change Column Header" dialog box, drag and drop the column header(s) into the desired order. Remember, this only moves the headers, not the data below them! Then, hit "OK".

## Change Column Header

#
<input type='checkbox'>
Unique ID
First Name
Last Name
Title
email
email2
Phone
Phone Cell
Phone 2
TimeZone
Role

OK Cancel

### *If your .csv data is missing a data column needed for the Section grid:*

In our case, the .csv data is missing the required "Name" column for the Section grid. Think of the "Name" as the information you would want to search for in Provision. We wouldn't want to search just for "Bob" or "Smith" when looking down a list of names, so under the "Name" column, we need to see the full first and last names, like "Bob Smith".

Currently, our options to fix this are:

1) Edit the .csv directly in your spreadsheet program: (Recommended) Simply revise the .csv to include another column for "Name", and re-open the .csv in the importer. The benefit to this method is your .csv file will be set up as a template for future imports.

Or:

2) In the Resource Importer, temporarily hide the extra column in the Section Grid: Make the columns between the .csv and the Section Grid match exactly by temporarily [hiding the column](#) (in this case, "Name") in the Section Grid, proceed to move the data into the Section grid (see [Step 5](#)), then unhide the "Name" column and manually add the data as needed prior to completing the import.

contact-import-sample_v...								
Contact [12 rows]								
menu +								
#	Name (required)	Custom ID	First Name	Last Name	Email	2nd Email	Phone	
0	Aaron Hughes	6c-004	Aaron	Hughes	aaron@6c...	support@...	1-408-555...	1-
1	Amanda Kingston	6c-015	Amanda	Kingston	akingston...		44 123 55...	
2	Bob Smith	6c-007	Bob	Smith	bsmith@a...		888-call-now	
3	Herold Waters	6c-011	Herold	Waters	hwaters@i...		234-555-6...	
4	Jill Keller	6c-013	Jill	Keller	jill.keller@...		17 145 12...	
5	John Parker	6c-004	John	Parker	john@gm...		234.634.1...	
6	Mark Tompson	6c-010	Mark	Tompson	tompson...		888-nice-...	
7		6c-008	Maurice	Carmichael	mc@mail....		866-555-1...	
8		6c-012	Michael	Sanders	pm@myb...		354-555-1...	
9		6c-014	Sarah	Campbell	sa.camp...		234-234 1...	
10		6c-004	Tom	Taylor	ttaylor@to...		503-555-1...	
11		6c-009	Vince	Bunch	vbunch@...	ops@hap...	703-555-1...	

When edits and adjustments are complete, move to Step 5 - Drag rows to the Section Grid



## Resource Importer Walkthrough - Step 5

### Importing Resources

#### Step 5: Drag rows from the .csv Grid to the Section Grid

Once you have set the columns to match exactly between the .csv Grid and the Section grid, it's time to pull in the data from one to the other.

Simply click the checkboxes for the rows you wish to import (or use the "Select all" checkbox at the top), click anywhere on the row, and drag & drop onto the Section Grid tab ("Contact"). The tool will tell you how many rows you are moving as you drag them.

The screenshot shows the Resource Importer interface. On the left, the 'Section Grids' pane has the 'Contact' tab selected. The main area displays the 'Contact' section grid with columns: #, Unique ID, First Name, Last Name, email, email2, Phone, Phone 2, and Phone Cell. Below this, the 'Open Files' pane shows 'contact-import-sample\_v1.csv'. The 'Local Grids' pane is empty. The 'Completed Imports' pane is empty. The 'Contact' section grid shows 12 rows of data. A red arrow points from the 'Contact' tab to the 'Contact [0 rows]' header, and another red arrow points from the 'Contact [0 rows]' header to the 'Dragging 12 row(s)' text above the .csv grid.

#	Unique ID	First Name	Last Name	email	email2	Phone	Phone 2	Phone Cell
0	6c-004	Aaron	Hughes	aaron@connect...	support@6conne...	1-408-555-1212	1-408-555-1212	1-408-555-1212
1	6c-004	John	Parker	john@gmail.com		234.634.1234		888-call-now
2	6c-004	Tom	Taylor	ttaylor@toms.com		503-555-1256		866-555-1134
3	6c-007	Bob	Smith	bsmith@apple.com		888-call-now		703-555-1111
4	6c-008	Maurice	Carmichael	mc@mail.com		866-555-1134		888-nice-wor
5	6c-009	Vince	Bunch	vbunch@happyplace...	ops@happyplace...	703-555-1111		234-555-6678
6	6c-010	Mark	Tompson	tompson@t.net		888-nice-wor		354-555-1235
7	6c-011	Herold	Waters	hwaters@is.co.uk		234-555-6678		17 145 125124
8	6c-012	Michael	Sanders	pm@mybusiness...		354-555-1235		234-234 1234
9	6c-013	Jill	Keller	jill.keller@another...		17 145 125124		44 123 555 12
10	6c-014	Sarah	Campbell	sa.camp@intel.net		234-234 1234		888-call-now
11	6c-015	Amanda	Kingston	akingston@sellin...		44 123 555 12		866-555-1134

Click on the "Contact" tab when you are done, and you will now see your data in there, instead of the original .csv.

If you had to hide columns in the Section Grid prior to moving the .csv data, verify that all columns are visible and the required data filled in. In this case, we filled in the "Name" Column that was missing in the original .csv.

The screenshot shows the Resource Importer interface. On the left, the 'Section Grids' pane has the 'Contact' tab selected. The main area displays the 'Contact' section grid with columns: #, Name (required), Custom ID, First Name, Last Name, Email, 2nd Email, Phone, 2nd Phone, Mobile Phc, Role, and Time Zone. Below this, the 'Open Files' pane shows 'contact-import-sample\_v1.csv'. The 'Local Grids' pane is empty. The 'Completed Imports' pane is empty. The 'Contact' section grid shows 12 rows of data. The 'Name' column is highlighted in blue for the first row (Amanda Kingston).

#	Name (required)	Custom ID	First Name	Last Name	Email	2nd Email	Phone	2nd Phone	Mobile Phc	Role	Time Zone
0	Aaron Hughes	6c-004	Aaron	Hughes	aaron@6c...	support@...	1-408-555...	1-408-555...	1-408-555...	Technical	PT
1	John Parker	6c-004	John	Parker	john@gm...		234.634.1...		888-call-now	Technical	ET
2	Tom Taylor	6c-004	Tom	Taylor	ttaylor@io...		503-555-1...		866-555-1...	Technical	ET
3	Bob Smith	6c-007	Bob	Smith	bsmith@a...		888-call-now		703-555-1...	Technical	ET
4	Maurice Carmichael	6c-008	Maurice	Carmichael	mc@mail...		866-555-1...		888-nice-...	Abuse	GMT
5	Vince Bunch	6c-009	Vince	Bunch	vbunch@...	ops@hap...	703-555-1...		234-555-6...	Sales	CT
6	Mark Tompson	6c-010	Mark	Tompson	tompson...		888-nice-...		354-555-1...	Billing	PT
7	Herold Waters	6c-011	Herold	Waters	hwaters@l...		234-555-6...		17 145 12...	Billing	PT
8	Michael Sanders	6c-012	Michael	Sanders	pm@myb...		354-555-1...		234-234 1...	Sales	PT
9	Jill Keller	6c-013	Jill	Keller	jill.keller@...		17 145 12...		44 123 55...	Technical	PT
10	Sarah Campbell	6c-014	Sarah	Campbell	sa.camp...		234-234 1...		888-call-now	Technical	PT
11	Amanda Kingston	6c-015	Amanda	Kingston	akingston...		44 123 55...		866-555-1...	Technical	PT

After moving your data into the Section grid, proceed to Step 6 - Importing into ProVision.

## Resource Importer Walkthrough - Step 6

### Importing Resources

#### Step 6: Import into ProVision

When all of the data is under the Section Grid tab, and any required field data filled in, you can import the data into Provision! From the Data menu, select "Import active Sheet into ProVision". You will see an import progress bar. Once complete, you data will be in provision, filled into the Section fields for your chosen Resource.

The screenshot shows the 'Resource Importer' application window. The 'Data' menu is open, and the option 'Import active Sheet to ProVision' is highlighted with a red circle. The background shows a grid of contact data with columns for Last Name, Email, 2nd Email, Phone, 2nd Phone, Mobile Phd, Role, and Time Zone.

#	<input type="checkbox"/>	Name	Sort sheet by column Name, A → Z	Last Name	Email	2nd Email	Phone	2nd Phone	Mobile Phd	Role	Time Zone
0	<input type="checkbox"/>	Aaron H	Sort sheet by column Name, Z → A	Hughes	aaron@6c...	support@...	1-408-555...	1-408-555...	1-408-555...	Technical	PT
1	<input type="checkbox"/>	John Pa		Parker	john@gm...		234.634.1...		888-call-now	Technical	ET
2	<input type="checkbox"/>	Tom Ta	Search...	Taylor	ttaylor@to...		503-555-1...		866-555-1...	Technical	ET
3	<input type="checkbox"/>	Bob Smi		Smith	bsmith@a...		888-call-now		703-555-1...	Technical	ET
4	<input type="checkbox"/>	Maurice Carmichael		Carmichael	mc@mail...		866-555-1...		888-nice...	Abuse	GMT
5	<input type="checkbox"/>	Vince Bunch		Bunch	vbunch@...	ops@hap...	703-555-1...		234-555-6...	Sales	CT
6	<input type="checkbox"/>	Mark Tompson		Tompson	tompson...		888-nice...		354-555-1...	Billing	PT
7	<input type="checkbox"/>	Herold Waters		Waters	hwaters@i...		234-555-6...		17 145 12...	Billing	PT
8	<input type="checkbox"/>	Michael Sanders		Sanders	pm@myb...		354-555-1...		234-234 1...	Sales	PT
9	<input type="checkbox"/>	Jill Keller		Keller	jill.keller@...		17 145 12...		44 123 55...	Technical	PT
10	<input type="checkbox"/>	Sarah Campbell		Campbell	sa.camp...		234-234 1...		888-call-now	Technical	PT
11	<input type="checkbox"/>	Amanda Kingston		Kingston	akingston...		44 123 55...		866-555-1...	Technical	PT

The screenshot shows a modal dialog box titled 'Importing'. It features a blue progress bar that is nearly full, with the text '11 / 12' displayed in the center. The background shows the 'Data' menu of the Resource Importer application.

# Users & Permissions

## Overview

Users & Permissions is accessed from the Admin screen under the [Users](#) tab. Here, you will find tools for adding and managing permissions groups, users, and running queries for verifying a user's specific permissions.

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.

### Table of contents

- [Global Permissions](#)
- [Resource Permissions](#)
- [Users and Groups](#)
- [Verifying Permissions](#)

## 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

**Group Information**

Name: test group 1

Enabled: ☒

**Group Users**

test@6connect.com test mcTest

**Resource Permissions (Hide 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 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"/>
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"/>

[Add More Group Permissions](#)

**Save**

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 a 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

**Group Information**

Name: test group 1

Enabled: ☒

**Group Users**

test@6connect.com test mcTest

**Resource Permissions (Hide 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 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"/>
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"/>

[Add More Group Permissions](#)

**Save**

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 a 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.

Users			
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	  
fu@6connect.com	Fry Chen	AAA Group	  
<a href="#">Add User</a>			

### 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

test@6connect.com

First Name

test

Last Name

mcTest

☒ 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
4
✕

## 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

//

Send

## 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.

Groups		
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 
<span style="background-color: #28a745; color: white; padding: 5px 10px; border-radius: 3px;">Add Group</span>		



### Overlapping group and user permissions

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

## Add a Group

After hitting the "Add Group" button, the Group Information screen will pop up.

Group Information

Name

Enabled
☒

Resource Permissions (Show Details)

	IPAM	DNS	Peer	Resource	User	
Resource						
Top-Level (Global)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="X"/>

Add More Group Permissions

Save

Add in the name of the new group, and set the permissions for that group by defining the resource(s) in the dropdown menu, checking the functional areas that you want accessible. Click "Show Details" to fine tune the functional areas into Create/Read/Update/Delete level permissions. To add permissions for additional Resources, click "Add More Group Permissions", select the Resource, and check the desired permissions. To delete a Resource from the permissions list, simply click the red icon.

In the example below, New Group has top-level (Global) permissions to Create, Read, and Update IPAM and Resource functions, but not to delete or access other areas. For more detail on top-level and resource permissions, see [Global Permissions](#) and [Resource Permissions](#).

Group Information

Name

Enabled
☒

Resource Permissions (Hide Details)

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

C: Create R: Read U: Update D: Delete

Add More Group Permissions

Save

Click "Save" when complete. After adding the group, you can add users to the group by selecting it when editing a user account.



# 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:

Query

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

Groups effecting this user on this resource: Global Read-Only

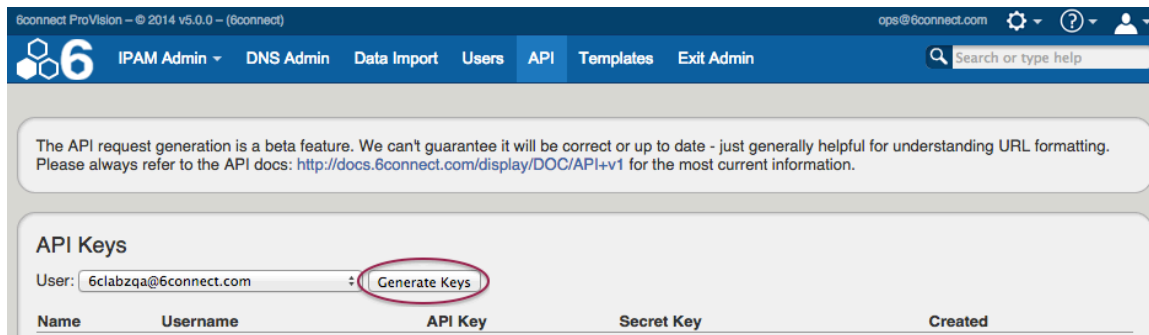
C: Create R: Read U: Update D: Delete

## API Tab

### API Tab

The **API** tab allows you to create and manager API keys for users.

To create a key, simply select the user, and click on "Generate Keys". The Name, Username, API Key, Secret Key, and Created date information will be added to the list below.



To revoke a user's key, click "Revoke" at the end of their entry.



For detailed information on working with API features, please refer to [ProVision Developer Tools](#) and [API v1](#).

# Templates

## Overview

The **Templates** tab under the Admin section allows you to view and edit DNS Templates.

When creating a new DNS zone via the gadget, the user can then specify which zone template to use.

1 Name	2 Records	3 Created By	4 Modified	5	6
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		

### DNS Template UI:

- 1) **Name:** Name given to the template
- 2) **Records:** Number of Zone Records associated with the template
- 3) **Created By:** Template creator
- 4) **Modified:** Last date of modification
- 5) **Edit:** Click to bring up the template detail below and edit information
- 6) **Delete:** Click to delete the DNS template

## Adding or Editing a DNS Template

The Admin can either create a new template or edit an existing template by clicking on the "Edit" icon in the DNS Template list for the entry.

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

**Add Template**

Name

**SOA Record**

Serial	Refresh	Retry	Expiry	Minimum
<input type="text"/>	<input type="text" value="14400"/>	<input type="text" value="3600"/>	<input type="text" value="604800"/>	<input type="text" value="3600"/>

**Zone Records**

Host:	TTL	Type	Priority	Value
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

```
@      IN      SOA      ns1.dns.6connect.net. hostmaster.6connect.net. (
                                <SERIAL> ; serial
                                14400      ; refresh
                                3600       ; retry
                                604800     ; expire
                                3600       ; minimum
                                )
```

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	⊖ ▲ ▼
					⊕

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.

**DNS**

**New DNS Zone**   -- no template --

**Zone Delegation**

**Delegated Zone**   **Slave IP**

-- no template --  
 Anna's Template  
**Demo Template**  
 Equinix  
 testing  
 VM Turnup

For more information on DNS Administration functions, refer to the [DNS Administration](#) section of the documentation.

# ProVision Developer Tools

## Developer Tools

6connect ProVision can integrate with your existing tools and workflow through use of the API and CLI. The 6connect API allows you to access the data and functions of the 6connect web tools to run advanced commands in ProVision, and supports a wide variety of update and deletion conditions not available in the UI.

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

### Table of contents

- [API v1](#)
- [CLI \(Alpha\)](#)

## 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 - Peering
- API Module - Resource
- How Do I...

# 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.

An API request looks like this:

<https://cloud.6connect.com/ex/api/v1/api.php?target=ipam&action=get&type=IP&mask=24>

An API response is a JSON-encoded text string, and looks like this:

```
{ "success":1, "message":"1 blocks found",
  "data":[{"id":"7539","oct1":"1","oct2":"2","oct3":"3","oct4":"0","mask":"24","child1":null
  "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":
  00:00:00","assigned_time":"2012-03-20 09:45:12"}]}
```

Instructions on decoding this return data can be found in the API endpoint documentation pages.

#### Using API Keys:

When using the API without pre-established authentication to ProVision, you must include both your API Key and a specially-prepared query hash parameter, like so:

<https://cloud.6connect.com/ex/api/v1/api.php?target=ipam&action=get&type=IP&mask=24&apiKey=116-MX15LUYY78ZZTW5&hash=8jxj4IApYmgb5IZC>

API Keys can be generated from your ProVision instance by navigating to the Admin panel by using the gear icon in the upper right hand corner, then navigating to the API tab. The API tab will present the API authentication information in the following format:

API Key: 38-TMHQV8CV2XZYC2ZS

Secret Key: 6e04e5822ce90feaa8947ded46c46878

The secret key serves as an API password and is used in the creation of the API Authentication hash. The formula for creating a API query hash from an API query and a Secret Key is the following:

Hash = Base64Encode( Sha256HMACHash ( QueryString, SecretKey ) )

In PHP, this would be performed with the following line of code:

```
$hash = base64_encode(hash_hmac('sha256', $_SERVER['QUERY_STRING'], $secretKey, TRUE));
```



**Because the hash function is computed based on the query string, you must calculate a unique hash for every API request!**



#### Example

Lets say you wanted to create a hash for the following API request:

[https://cloud.6connect.com/6c\\_375/api/v1/api.php?target=ipam&action=get&type=IP&mask=24](https://cloud.6connect.com/6c_375/api/v1/api.php?target=ipam&action=get&type=IP&mask=24)

And that your API Key and Secret Key are as follows:

API Key: 32-5DAYTJQY2TZHOFOB

Secret Key: 48b278ec873bda4738923dbc467f8669

The first step is to append your API Key to the URL. The API Key indicates which user is executing the API query.

[https://cloud.6connect.com/6c\\_375/api/v1/api.php?target=ipam&action=get&type=IP&mask=24&apiKey=32-5DAYTJQY2TZHOFOB](https://cloud.6connect.com/6c_375/api/v1/api.php?target=ipam&action=get&type=IP&mask=24&apiKey=32-5DAYTJQY2TZHOFOB)

The first step is to isolate the Query String from the request URL. The Query String is everything which follows the question mark. So,



Query String: target=ipam&action=get&type=IP&mask=24&apiKey=32-5DAYTJQY2TZHOFOB

The next step is to calculate the SHA256 hash of this string with your Secret Key. In PHP, this would be:

```
$sha256 = hash_hmac('sha256', "target=ipam&action=get&type=IP&mask=24&apiKey=32-5DAYTJQY2TZHOFOB",  
"48b278ec873bda4738923dbc467f8669", TRUE);
```

As this value has been 256-bit hashed, it will contain many unprintable characters. The solution to this is to encode it in base 64 for transport. Again, in PHP:

```
$hash = base64_encode($sha256);
```

Calculating it out yields the completed hash:

```
$hash = yneSFMyxPPe+3W4IOkVp50K3VStatBcRRak+2ygDUWQ=
```

The calculated hash can then be appended to the full API Query URL to form a completed request:

```
https://cloud.6connect.com/6c\_375/api/v1/api.php?target=ipam&action=get&type=IP&mask=24&apiKey=32-5DAYTJQY2TZHOFOB
```



#### A Note on False Positives

ProVision utilizes several possible authentication schemes of which key-based API authentication is only one. Session-based, username/password authentication is used for the majority of user interaction with the ProVision front end. Because session information is stored in browsers cookies, a browser can be authenticated to execute API commands as long as the session is active.

Unfortunately, this can lead to confusion when using a machine-based API as the user might use an authenticated browser session to test API-Key based API queries. These queries will always succeed regardless of whether the API Query Hash was calculated correctly as the system defaults to Session-based authentication when it is available.

To ensure that session-based authentication is not polluting your API-Key based testing, always use a separate browser which is not logged in to your ProVision instance to test API queries.

#### Other Languages

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 a licensed 6connect ProVision application. If you would like access to a ProVision license 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 "API" tab.
- 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 5.5 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 [6connect-PHP-SDKv2.tar.gz](#)

#### 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-examples.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

### Admin and Audit

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

- [Authentication Testing](#)
- [Log Management](#)
- [Zone Templates](#)

## Authentication Testing

- Authentication Testing
  - testSSH
  - testLDAP
  - testSecure64

## 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	<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td colspan="3">{ "success":1, "message":"Success!" }</td></tr><tr><td>ERROR</td><td colspan="3">{ 'success':0, 'message':'error message' }</td></tr></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><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><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></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><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><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></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=123456																							

testLDAP	
URL	/api/v1/api.php?target=auth&action=testLDAP

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><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><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></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	/api/v1/api.php?target=auth&action=testLDAP&ldapPort=389&ldapServer=																

testSecure64															
URL	/api/v1/api.php?target=auth&action=testSecure64														
Description	Returns success or failure of a connection to an Secure64 DNS appliance.														
Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td colspan="3">{<span>"</span>success<span>"</span>:1,<span>"</span>message<span>"</span>:<span>"</span>Successful<span>"</span>}</td></tr><tr><td>ERROR</td><td colspan="3">{<span>'</span>success<span>'</span>:0, <span>'</span>message<span>'</span>:<span>'</span>error message<span>'</span>}</td></tr></table>			SUCCESSFUL	{ <span>"</span> success <span>"</span> :1, <span>"</span> message <span>"</span> : <span>"</span> Successful <span>"</span> }			ERROR	{ <span>'</span> success <span>'</span> :0, <span>'</span> message <span>'</span> : <span>'</span> error message <span>'</span> }						
SUCCESSFUL	{ <span>"</span> success <span>"</span> :1, <span>"</span> message <span>"</span> : <span>"</span> Successful <span>"</span> }														
ERROR	{ <span>'</span> success <span>'</span> :0, <span>'</span> message <span>'</span> : <span>'</span> error message <span>'</span> }														
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><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></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				
	<b>Name</b>	<b>Type</b>	<b>Example</b>	<b>Description</b>
	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=/tmp			

Log Management

- Log Management
  - Get

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	<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td>{<i>"success":1,"message":"Search Successful.",</i><i>"data":{"logId":"31568"</i><i>"2012-05-07 17:44:43",</i><i>"logLevel":"INFO","userId":"39","userEmail":"anna@6connect.com"</i><i>,"logCategory":</i><i>"User",</i><i>"message": "Anna Claiborne (anna@6connect.com ) logged in via local authentication",</i><i>"ip":"107.111.0.228"</i></td></tr><tr><td>ERROR</td><td>{<i>'success':0, 'message':'error message'</i>}</td></tr></table> <div>Data Detail</div>	SUCCESSFUL	{ <i>"success":1,"message":"Search Successful.",</i> <i>"data":{"logId":"31568"</i> <i>"2012-05-07 17:44:43",</i> <i>"logLevel":"INFO","userId":"39","userEmail":"anna@6connect.com"</i> <i>,"logCategory":</i> <i>"User",</i> <i>"message": "Anna Claiborne (anna@6connect.com ) logged in via local authentication",</i> <i>"ip":"107.111.0.228"</i>	ERROR	{ <i>'success':0, 'message':'error message'</i> }
SUCCESSFUL	{ <i>"success":1,"message":"Search Successful.",</i> <i>"data":{"logId":"31568"</i> <i>"2012-05-07 17:44:43",</i> <i>"logLevel":"INFO","userId":"39","userEmail":"anna@6connect.com"</i> <i>,"logCategory":</i> <i>"User",</i> <i>"message": "Anna Claiborne (anna@6connect.com ) logged in via local authentication",</i> <i>"ip":"107.111.0.228"</i>				
ERROR	{ <i>'success':0, 'message':'error message'</i> }				



	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>logId</td><td>INTEGER</td><td>24</td><td>Unique log entry id.</td></tr><tr><td>time</td><td>DATETIME</td><td>2012-05-07 22:10:07</td><td>Date and time year to second.</td></tr><tr><td>logLevel</td><td>STRING</td><td>NOTICE</td><td>Standard syslog log levels in verbose format (EMERG, ALERT, CRIT, ERR, WARNING, NOTICE, INFO, DEBUG).</td></tr><tr><td>userId</td><td>Integer</td><td>11</td><td>The unique user id associated with the log entry.</td></tr><tr><td>userName</td><td>STRING</td><td>anna@6connect.com</td><td>The unique user name associated with the log entry.</td></tr><tr><td>logCategory</td><td>STRING</td><td>IPAM</td><td>The 6connect category for the log entry (User, IPAM, Resource Holder, DNS, Peering, Assistant, NTP, Reporting).</td></tr><tr><td>message</td><td>STRING</td><td>Created new children from 1.0.0.0/24</td><td>The detailed log message.</td></tr><tr><td>ip</td><td>STRING</td><td>107.111.0.228</td><td>The remote IP address of the user who took the action being logged.</td></tr></table>	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.
	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	<div>None</div>																																				
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr></table>	Name	Type	Example	Description																																
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="#">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
  - Get
  - Update
  - Delete

## Zone Templates

Get																
URL	/api/v1/api.php?target=zoneTemplate&action=get															
Description	Returns success or failure of a connection to an external server via SSH.															
Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td colspan="3">{ "success":1,"message":"Found 1 records for template \nAwesome Template\n","data":{"templateId":"11","template":"Template","created":"2013-07-31 14:01:24","modified":"2013-07-31 14:01:24","userId":"112","soa":null,"</td></tr><tr><td>ERROR</td><td colspan="3">{ 'success':0, 'message': 'error message' }</td></tr></table>				SUCCESSFUL	{ "success":1,"message":"Found 1 records for template \nAwesome Template\n","data":{"templateId":"11","template":"Template","created":"2013-07-31 14:01:24","modified":"2013-07-31 14:01:24","userId":"112","soa":null,"			ERROR	{ 'success':0, 'message': 'error message' }						
SUCCESSFUL	{ "success":1,"message":"Found 1 records for template \nAwesome Template\n","data":{"templateId":"11","template":"Template","created":"2013-07-31 14:01:24","modified":"2013-07-31 14:01:24","userId":"112","soa":null,"															
ERROR	{ 'success':0, 'message': 'error message' }															
Required Parameters	None															
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><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></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	<div>Examples:</div> <div>SUCCESSFUL: {"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"}}</div> <div>ERROR: {'success':0, 'message':'Error updating template: error details'}&gt;</div>																												
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>name</td><td>STRING</td><td>Test Template</td><td>The name of the template to be created or updated.</td></tr><tr><td></td><td></td><td></td><td></td></tr></table>	Name	Type	Example	Description	name	STRING	Test Template	The name of the template to be created or updated.																				
Name	Type	Example	Description																										
name	STRING	Test Template	The name of the template to be created or updated.																										
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>soa</td><td>STRING</td><td>ns1.test.net hostmaster.ns1.test.net</td><td>A valid SOA for the template in for format</td></tr><tr><td>ttd</td><td>INTEGER</td><td>86400</td><td>The TTL for the zone template, which is the default expiration time for all records without their own TTL.</td></tr><tr><td>refresh</td><td>INTEGER</td><td>14400</td><td>The time period for slaves to refresh the zone.</td></tr><tr><td>retry</td><td>INTEGER</td><td>3600</td><td>Time that a slave should retry refreshing the zone in case of incident.</td></tr><tr><td>expire</td><td>INTEGER</td><td>604800</td><td>Time for a slave to expire a zone.</td></tr><tr><td>mininum</td><td>INTEGER</td><td>3600</td><td>The maximum caching time in the event of failed lookups.</td></tr></table>	Name	Type	Example	Description	soa	STRING	ns1.test.net hostmaster.ns1.test.net	A valid SOA for the template in for format	ttd	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.
Name	Type	Example	Description																										
soa	STRING	ns1.test.net hostmaster.ns1.test.net	A valid SOA for the template in for format																										
ttd	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.
	t1l_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=10118&refresh=14400&retry=3600&expire=604800&minimum=3600&value_0:			

Delete	
URL	/api/v1/api.php?target=zoneTemplate&action=delete
Description	Deletes a DNS template.

Returns	<b>Examples:</b> <table><tr><td>SUCCESSFUL</td><td colspan="3">{ "success":1,"message":"Template \\"Test Template\\" delete."}</td></tr><tr><td>ERROR</td><td colspan="3">{ "success":0,"message":"No template found for templateId \\"1005\\"."}</td></tr></table>				SUCCESSFUL	{ "success":1,"message":"Template \\"Test Template\\" delete."}			ERROR	{ "success":0,"message":"No template found for templateId \\"1005\\"."}		
SUCCESSFUL	{ "success":1,"message":"Template \\"Test Template\\" delete."}											
ERROR	{ "success":0,"message":"No template found for templateId \\"1005\\"."}											
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>templateId</td><td>INTEGER</td><td>3</td><td>ID of the template to delete.</td></tr></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	/api/v1/api.php?target=zoneTemplate&action=delete&templateId=1005											

## API Module - DHCP

- DHCP Server Control
  - get
  - update
  - add
  - delete
  - testConnection
  - push
- DHCP Entry Control
  - get
  - update
  - updateOption
  - add
  - addOption
  - delete
  - deleteOption

## DHCP Server Control

get					
URL	/api/v1/api.php?target=DHCP&action=get				
Description	Accepts search criteria to retrieve a list of all matching DHCP Servers.				
Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td><pre>{   "success": 1,   "message": "Search Successful.",   "data": [     {       "DHCPID": "1",       "trace.foo.com": "DHCPPort": "22",       "h}k*c))jwqhgd",       "DHCPType": "ISC",       "Vdhcpd.conf",       "DHCPServerStop": "s",       "kill -9 `cat VvarVrunVdhcpd.pid`",       "DHCPServerStart": "sudo",       "VusrVlocalVdhcpVsbinVdhcpd -p",       "75",       "DHCPDefaultLease": null,       "DHCPMaxLease": null,       "DHCP": null,       "DHCPDomainName": null,       "DHCPN"     }   ] }</pre></td></tr><tr><td>ERROR:</td><td><pre>{   "success": 0,   "message": "error message" }</pre></td></tr></table> <div>Data Detail:</div>	SUCCESSFUL:	<pre>{   "success": 1,   "message": "Search Successful.",   "data": [     {       "DHCPID": "1",       "trace.foo.com": "DHCPPort": "22",       "h}k*c))jwqhgd",       "DHCPType": "ISC",       "Vdhcpd.conf",       "DHCPServerStop": "s",       "kill -9 `cat VvarVrunVdhcpd.pid`",       "DHCPServerStart": "sudo",       "VusrVlocalVdhcpVsbinVdhcpd -p",       "75",       "DHCPDefaultLease": null,       "DHCPMaxLease": null,       "DHCP": null,       "DHCPDomainName": null,       "DHCPN"     }   ] }</pre>	ERROR:	<pre>{   "success": 0,   "message": "error message" }</pre>
SUCCESSFUL:	<pre>{   "success": 1,   "message": "Search Successful.",   "data": [     {       "DHCPID": "1",       "trace.foo.com": "DHCPPort": "22",       "h}k*c))jwqhgd",       "DHCPType": "ISC",       "Vdhcpd.conf",       "DHCPServerStop": "s",       "kill -9 `cat VvarVrunVdhcpd.pid`",       "DHCPServerStart": "sudo",       "VusrVlocalVdhcpVsbinVdhcpd -p",       "75",       "DHCPDefaultLease": null,       "DHCPMaxLease": null,       "DHCP": null,       "DHCPDomainName": null,       "DHCPN"     }   ] }</pre>				
ERROR:	<pre>{   "success": 0,   "message": "error message" }</pre>				



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
DHCPTType	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.
DHCPTType	STRING	MSDHCP	The DHCP Server Type to search for.
DHCPConfigPath	STRING	/where/is/it/	The Config Path to search for.
DHCPServerStop	STRING	/path/to/server/stop	Search by server stop command.
DHCPServerStart	STRING	/path/to/server/start	Search by server start command.
DHCPDefaultLease	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.
DHCPDomainName	STRING	domain.name.servers	Search by domain name servers.
DHCPNameServers	STRING	ns.domain.com	Search by name servers.
DHCPUseText	BOOL	1	Search by using text configs or not.
DHCPConfigText	STRING	Text File	Search by text file contents.

Example URL

/api/v1/api.php?target=DHCPServer&action=get&likeFlag=1&DHCPUse

## 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	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td colspan="3">{<i>"success":1,"message":"Update Successful."</i>}</td></tr><tr><td>ERROR:</td><td colspan="3">{<i>"success":0, "message":"error message"</i>}</td></tr></table>				SUCCESSFUL:	{ <i>"success":1,"message":"Update Successful."</i> }			ERROR:	{ <i>"success":0, "message":"error message"</i> }		
SUCCESSFUL:	{ <i>"success":1,"message":"Update Successful."</i> }											
ERROR:	{ <i>"success":0, "message":"error message"</i> }											
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><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></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.
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.servers	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.
	UpdateDomainNameServers	STRING	domain.name.servers	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.
Example URL	/api/v1/api.php?target=DHCPServer&action=update&SearchPort=43&UpdateServer=192.168.1.100&UpdatePort=43&UpdateUsername=kjennings&UpdatePassword=*****&UpdateType=ISC&UpdateConfigPath=/etc/dhcpd.conf&UpdateServerStop=/usr/sbin/service dhcpd stop&UpdateServerStart=/usr/sbin/service dhcpd start&UpdateDefaultLease=64000&UpdateMaxLease=128000&UpdateAuthoritative=1&UpdateLogFacility=local7&UpdateDomainNameServers=domain.name.servers&UpdateNameServers=ns.domain.com&UpdateUseText=1&UpdateConfigText=Text File			

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

Returns

Examples:

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

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	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>DHCPConfigPath</td><td>STRING</td><td>/where/is/it/</td><td>The new config path.</td></tr><tr><td>DHCPServerStop</td><td>STRING</td><td>/path/to/server/stop</td><td>The new server stop command.</td></tr><tr><td>DHCPServerStart</td><td>STRING</td><td>/path/to/server/start</td><td>The new server start command.</td></tr><tr><td>DHCPDefaultLease</td><td>INT</td><td>64000</td><td>The new default lease.</td></tr><tr><td>DHCPMaxLease</td><td>INT</td><td>128000</td><td>The new max lease.</td></tr><tr><td>DHCPAuthoritative</td><td>BOOL</td><td>1</td><td>The new Authoritative status.</td></tr><tr><td>DHCPLogFacility</td><td>STRING</td><td>local7</td><td>The new logging facility.</td></tr><tr><td>DHCPDomainNameServers</td><td>STRING</td><td>domain.name.servers</td><td>The new domain name servers.</td></tr><tr><td>DHCPNameServers</td><td>STRING</td><td>ns.domain.com</td><td>The new name servers.</td></tr><tr><td>DHCPUseText</td><td>BOOL</td><td>1</td><td>The new use text file setting.</td></tr><tr><td>DHCPConfigText</td><td>STRING</td><td>Text File</td><td>The new use text file content.</td></tr></table>	Name	Type	Example	Description	DHCPConfigPath	STRING	/where/is/it/	The new config path.	DHCPServerStop	STRING	/path/to/server/stop	The new server stop command.	DHCPServerStart	STRING	/path/to/server/start	The new server start command.	DHCPDefaultLease	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.	DHCPDomainNameServers	STRING	domain.name.servers	The new domain name servers.	DHCPNameServers	STRING	ns.domain.com	The new name servers.	DHCPUseText	BOOL	1	The new use text file setting.	DHCPConfigText	STRING	Text File	The new use text file content.	
	Name	Type	Example	Description																																														
	DHCPConfigPath	STRING	/where/is/it/	The new config path.																																														
	DHCPServerStop	STRING	/path/to/server/stop	The new server stop command.																																														
	DHCPServerStart	STRING	/path/to/server/start	The new server start command.																																														
	DHCPDefaultLease	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.																																														
	DHCPDomainNameServers	STRING	domain.name.servers	The new domain name servers.																																														
	DHCPNameServers	STRING	ns.domain.com	The new name servers.																																														
DHCPUseText	BOOL	1	The new use text file setting.																																															
DHCPConfigText	STRING	Text File	The new use text file content.																																															
Example URL	/api/v1/api.php?target=DHCPServer&action=addDHCPServer=IP/domain																																																	

delete					
URL	/api/v1/api.php?target=DHCPServer&action=delete				
Description	Performs a search over the DHCP Servers dataset and deletes all found matches.				
Returns	<p><b>Examples:</b></p> <table> <tr> <td>SUCCESSFUL:</td><td><code>{"success":1,"message":"DHCPServer Deleted."}</code></td></tr> <tr> <td>ERROR:</td><td><code>{"success":0, "message":"error message"}</code></td></tr> </table>	SUCCESSFUL:	<code>{"success":1,"message":"DHCPServer Deleted."}</code>	ERROR:	<code>{"success":0, "message":"error message"}</code>
SUCCESSFUL:	<code>{"success":1,"message":"DHCPServer 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.
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.
DHCPTType	STRING	MSDHCP	The DHCP Server Type to search for.
DHCPConfigPath	STRING	/where/is/it/	The Config Path to search for.
DHCPServerStop	STRING	/path/to/server/stop	Search by server stop command.
DHCPServerStart	STRING	/path/to/server/start	Search by server start command.
DHCPDefaultLease	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.
DHCPDomainName	STRING	domain.name.servers	Search by domain name servers.
DHCPNameServer	STRING	ns.domain.com	Search by name servers.
DHCPUseText	BOOL	1	Search by using text configs or not.
DHCPConfigText	STRING	Text File	Search by text file contents.

Example URL

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

### *testConnection*

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	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td colspan="3">{ "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."]] }</td></tr><tr><td>ERROR:</td><td colspan="3">{ "success":0, "message": "error message" }</td></tr></table>				SUCCESSFUL:	{ "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."]] }			ERROR:	{ "success":0, "message": "error message" }						
SUCCESSFUL:	{ "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."]] }															
ERROR:	{ "success":0, "message": "error message" }															
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><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></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.
DHCPTType	STRING	MSDHCP	The DHCP Server Type to search for.
DHCPConfigPath	STRING	/where/is/it/	The Config Path to search for.
DHCPServerStop	STRING	/path/to/server/stop	Search by server stop command.
DHCPServerStart	STRING	/path/to/server/start	Search by server start command.
DHCPDefaultLease	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.
DHCPDomainName	STRING	domain.name.servers	Search by domain name servers.
DHCPNameServers	STRING	ns.domain.com	Search by name servers.
DHCPUseText	BOOL	1	Search by using text configs or not.
DHCPConfigText	STRING	Text File	Search by text file contents.

Example URL

/api/v1/api.php?target=DHCPServer&action=testConnection&DHCPId=1

## *push*

### 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

#### Examples:

##### SUCCESSFUL:

```
{"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."]]}
```

##### 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.
DHCPTType	STRING	MSDHCP	The DHCP Server Type to search for.
DHCPConfigPath	STRING	/where/is/it/	The Config Path to search for.
DHCPServerStop	STRING	/path/to/server/stop	Search by server stop command.
DHCPServerStart	STRING	/path/to/server/start	Search by server start command.
DHCPDefaultLease	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.
DHCPDomainName	STRING	domain.name.servers	Search by domain name servers.
DHCPNameServers	STRING	ns.domain.com	Search by name servers.
DHCPUseText	BOOL	1	Search by using text configs or not.
DHCPConfigText	STRING	Text File	Search by text file contents.

Example URL

/api/v1/api.php?target=DHCPServer&action=push&DHCPServer=IP/dor

DHCP Entry Control

<i>get</i>					
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	<div><div>Examples:</div><table><tr><td>SUCCESSFUL:</td><td><pre>{"EntryId": "27", "EntryParent": null, "EntryName": "host", "EntryNetmask": "255.255.255.0", "EntryOptions": [{"OptionId": "46", "OptionSubnetId": "1", "OptionValue": "11:23:45:67:89:ab"}, {"OptionId": "4", "OptionValue": "10.20.30.158"}]}</pre></td></tr><tr><td>ERROR:</td><td><pre>{"success": 0, "message": "error message"}</pre></td></tr></table><div>Data Detail:</div></div>	SUCCESSFUL:	<pre>{"EntryId": "27", "EntryParent": null, "EntryName": "host", "EntryNetmask": "255.255.255.0", "EntryOptions": [{"OptionId": "46", "OptionSubnetId": "1", "OptionValue": "11:23:45:67:89:ab"}, {"OptionId": "4", "OptionValue": "10.20.30.158"}]}</pre>	ERROR:	<pre>{"success": 0, "message": "error message"}</pre>
SUCCESSFUL:	<pre>{"EntryId": "27", "EntryParent": null, "EntryName": "host", "EntryNetmask": "255.255.255.0", "EntryOptions": [{"OptionId": "46", "OptionSubnetId": "1", "OptionValue": "11:23:45:67:89:ab"}, {"OptionId": "4", "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.

Example URL

/api/v1/api.php?target=DHCPEntry&amp;action=get&amp;EntryID=123

## 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	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td colspan="3">{<i>"success":1,"message":"Update Successful."</i>}</td></tr><tr><td>ERROR:</td><td colspan="3">{<i>"success":0, "message":"error message"</i>}</td></tr></table>				SUCCESSFUL:	{ <i>"success":1,"message":"Update Successful."</i> }			ERROR:	{ <i>"success":0, "message":"error message"</i> }		
SUCCESSFUL:	{ <i>"success":1,"message":"Update Successful."</i> }											
ERROR:	{ <i>"success":0, "message":"error message"</i> }											
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><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></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.
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.
SearchServerId	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.
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.
Example URL		/api/v1/api.php?target=DHCPEntry&action=update&SearchID=123&Upd			

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	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td colspan="3">{ "success":1, "message": "Update Successful." }</td></tr><tr><td>ERROR:</td><td colspan="3">{ "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" }											
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><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></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.
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.									

	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>SearchId</td><td>INT</td><td>123</td><td>The ID of the DHCP Entry to search for.</td></tr><tr><td>SearchParent</td><td>INT</td><td>123</td><td>The parent Entry to search for.</td></tr><tr><td>SearchServerId</td><td>INT</td><td>123</td><td>The DHCP Server to search for.</td></tr><tr><td>SearchType</td><td>STRING</td><td>subnet</td><td>The Entry type to search for.</td></tr><tr><td>SearchName</td><td>STRING</td><td>30.20.10.1</td><td>The name to search for.</td></tr><tr><td>SearchNetmask</td><td>STRING</td><td>255.255.255.0</td><td>The subnet mask to search for.</td></tr><tr><td>SearchKey</td><td>STRING</td><td>range</td><td>The key portion of the option key-value pairing to search for.</td></tr><tr><td>SearchValue</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></table>	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.	SearchServerId	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.	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																																	
	SearchId	INT	123	The ID of the DHCP Entry to search for.																																	
	SearchParent	INT	123	The parent Entry to search for.																																	
	SearchServerId	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.																																	
	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.																																	
<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>UpdateKey</td><td>STRING</td><td>range</td><td>The new key portion of the option key-value pairing.</td></tr><tr><td>UpdateValue</td><td>STRING</td><td>30.20.10.10 30.20.10.40</td><td>The new value portion of the option key-value pairing.</td></tr></table>	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.																									
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.																																		
Example URL	/api/v1/api.php?target=DHCPEntry&action=updateOption&SearchNetmask=																																				
add																																					
URL	/api/v1/api.php?target=DHCPEntry&action=add																																				
Description	Adds a new DHCP Entry and returns the new ID.																																				

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 Entry.

Required Parameters

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.
EntryNetmask	STRING	255.255.255.0	The subnet mask of this new Entry.

Optional Parameters

Name	Type	Example	Description
EntryParent	INT	123	The parent Entry to search for.

Example URL

/api/v1/api.php?target=DHCPEntry&action=add&EntryServerId=123&En

addOption

URL

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

Description

Creates a new DHCP Option and returns the new ID.

Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td><i>{"success":1,"message":"Add Successful.", "data":123}</i></td></tr><tr><td>ERROR:</td><td><i>{"success":0, "message":"error message"}</i></td></tr></table> <div>Data Detail:</div> <table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td>data</td><td>INT</td><td>The ID of the new DHCP Option.</td></tr></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 Option.						
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 Option.															
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>OptionSubnetId</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></table>	Name	Type	Example	Description	OptionSubnetId	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														
OptionSubnetId	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.														
Example URL	/api/v1/api.php?target=DHCPEntry&action=addOption&OptionSubnetId=																

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	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td><i>{"success":1,"message":"DHCPEntry Deleted."}</i></td></tr><tr><td>ERROR:</td><td><i>{"success":0, "message":"error message"}</i></td></tr></table>	SUCCESSFUL:	<i>{"success":1,"message":"DHCPEntry Deleted."}</i>	ERROR:	<i>{"success":0, "message":"error message"}</i>
SUCCESSFUL:	<i>{"success":1,"message":"DHCPEntry Deleted."}</i>				
ERROR:	<i>{"success":0, "message":"error message"}</i>				

## 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.

Example URL

`/api/v1/api.php?target=DHCPEntry&action=delete&EntryParent=123`***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	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td><i>{"success":1,"message":"DHCP Option(s) Deleted."}</i></td></tr><tr><td>ERROR:</td><td><i>{"success":0, "message":"error message"}</i></td></tr></table>		SUCCESSFUL:	<i>{"success":1,"message":"DHCP Option(s) Deleted."}</i>	ERROR:	<i>{"success":0, "message":"error message"}</i>
SUCCESSFUL:	<i>{"success":1,"message":"DHCP Option(s) Deleted."}</i>					
ERROR:	<i>{"success":0, "message":"error message"}</i>					

## 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.

Example URL

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

## API Module - DNS

- DNS Server Control
  - get
  - add
  - delete
  - update
  - transferByServer
  - transferSingle
- DNS Zone Control
  - get
  - search
  - update
  - add
  - delete
  - getRecordTypes
  - getFile
  - getDSFile
  - checkZone
  - getArchivedZone
- DNS Record Control
  - get
  - update
  - add
  - delete
  - switch
- Server-Zone Linkage
  - get
  - add
  - delete
- Name Server Control
  - get
  - add
  - delete
  - setDefault
  - orderUp
  - orderDown

### DNS Server Control

<i>get</i>	
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	<b>Examples:</b>

SUCCESSFUL:	{ "success":1,"message":"Fetch Sucessful.", "data":{"id":"10","server","username":"user", "password":"vwvddp","port":"2600", "SCP","remote_directory":"zones", "named_conf_path":"Vetc\\zones", ":null,"dyn_DNSSEC_contact":null, "powerdns_backend":"Bind","db_us "server_type":"slave","SOA":null,"m {"customer_name":"","server_ty ,"SOA":"","remote_directory":"","z \\named_conf_path":"","Vetc\\zone \\dyn_DNSSEC_contact\\ ":"","post_command":"","\\ ,"pre_command":"","\\powerdns_t \\db_username":"","db_password "testID":"963","zoneCount":"8","view \\server_id":"","10","name":"","_6con }}
ERROR:	{ "success":0, "message":"error message"}

#### Data Detail:

Name	Type	Description
id	INTEGER	Server ID
server	STRING	Server Name
username	STRING	Login Name
password	CRYPT	Login Password
port	INTEGER	Port the Server listens on
zoneCount	INTEGER	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	Protocol used for transfer of DNS zones and records. 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 path to the zone files used within the named.conf 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	INTEGER	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	INTEGER	The View ID
server_id	INTEGER	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	INTEGER	The UNIX timestamp of when the view was created.

	<table><tr><td>extras</td><td>JSON</td><td>A JSON-encoded array of the extra attributes printed out in the view definition in the config file.</td></tr></table>	extras	JSON	A JSON-encoded array of the extra attributes printed out in the view definition in the config file.					
extras	JSON	A JSON-encoded array of the extra attributes printed out in the view definition in the config file.							
Required Parameters	None								
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>id</td><td>INTEGER</td><td>15</td><td>The server id to fetch.</td></tr></table>	Name	Type	Example	Description	id	INTEGER	15	The server id to fetch.
Name	Type	Example	Description						
id	INTEGER	15	The server id to fetch.						
Example URL	/api/v1/api.php?target=dnsServer&action=get&id=15								

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

Name	Type	Example	Description
server	STRING	dns.yourdomain.com	Full FQDN of the DNS Server
password	STRING	password1	Login password for Server
transferType	STRING	SCP	Protocol used for transfer of DNS zones and records. Valid settings include SCP, PowerDNS, Secure64, Secure64Signer
serverType	STRING	Master	Values are 'Master' or 'Slave' only
displayName	STRING	Primary NS	The name displayed representing the DNS server, can be the same as server or different
SOA	STRING	ns1.6connect.com hostmaster.6connect.com	Server of Authority record for DNS server

## Optional Parameters

These optional parameters vary according to what type of server is being configured.

Name	Type	Example	Description
customerName	STRING	/tmp/zones	Customer Name
remoteDirectory	STRING	/tmp/zones	Zone Directory on Server
port	INTEGER	22	Port for ssh or scp access to server
namedConfPath	STRING	/tmp	The path to the zone files used within the named.conf file.

	preCommand	STRING	/path/to/stuff/preCommand	Command to execute before zone transfer
	postCommand	STRING	/path/to/stuff/postCommand	Command to execute after zone transfer
	DNSSECContact	STRING	joeuser	For use with Dyn dns service
	username	STRING	bobuser	Login name for Server
	active	INTEGER	0	Values 0 or 1 only, sets the server to inactive on 0 value
	masterid	INTEGER	53	Master server ID. If a server is a slave, masterid points to its master.
	powerDNSBackend	STRING	Bind or MySQL	pDNS server backend type
	dbDatabaseName	STRING	pdns_1	DB name for pDNS servers with MySQL powerDNSBackend type
	dbPort	INTEGER	3306	Port for for pDNS servers with MySQL powerDNSBackend type
	dbUsername	STRING	someuser	DB username for pDNS servers with MySQL powerDNSBackend type
	dbPassword	STRING	somepass	DB password for pDNS servers with MySQL powerDNSBackend type
Example URL	/api/v1/api.php?target=dnsServer&action=add&server=dns.yourdomain.com&transferType=Secure64&displayName=PrimaryNS&serverType=master&password=password1&SOA=ns1.6connect.com.+hostmaster.6connect.com.			



## *delete*

URL	/api/v1/api.php?target=dnsServer&action=delete											
Description	Deletes a DNS Server											
Returns	<div><b>Examples:</b></div> <table><tr><td>SUCCESSFUL:</td><td colspan="3">{ "success":1, "message":"Delete Successful." }</td></tr><tr><td>ERROR:</td><td colspan="3">{ "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><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>id</td><td>INTEGER</td><td>5</td><td>ID of server to delete</td></tr></table>				Name	Type	Example	Description	id	INTEGER	5	ID of server to delete
Name	Type	Example	Description									
id	INTEGER	5	ID of server to delete									
Optional Parameters	None											
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	<div><b>Examples:</b></div> <table><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

Name	Type	Example	Description
id	INTEGER	5	ID of server
server	STRING	dns.yourdomain.com	Full FQDN of the DNS Server
SOA	STRING	ns1.6connect.com hostmaster.6connect.com	Server of Authority record for DNS server
transferType	STRING	SCP	Protocol used for transfer of DNS zones and records. Valid settings include SCP, PowerDNS, Secure64, Secure64Signer

## Optional Parameters

These optional parameters vary according to what type of server is being configured.

Name	Type	Example	Description
active	INTEGER	0	Values 0 or 1 only, sets the server to inactive on 0 value
customerName	STRING	/tmp/zones	Customer Name
dbDatabaseName	STRING	pdns_1	DB name for pDNS servers with MySQL powerDNSBackend type
dbPassword	STRING	somepass	DB password for pDNS servers with MySQL powerDNSBackend type
dbPort	INTEGER	3306	Port for for pDNS servers with MySQL powerDNSBackend type

dbUsername	STRING	someuser	DB username for pDNS servers with MySQL powerDNSBackend type
displayName	STRING	Primary NS	The name displayed representing the DNS server, can be the same as server or different
DNSSECContact	STRING	joeuser	For use with Dyn dns service
enable_views	INTEGER	1	Whether or not Views are enabled. Valid values are '1' for enable or '0' for do not enable
masterid	INTEGER	53	Master server ID. If a server is a slave, masterid points to its master.
namedConfPath	STRING	/tmp	The path to the zone files used within the named.conf file.
password	STRING	password1	Login password for Server
port	INTEGER	22	Port for ssh or scp access to server
powerDNSBackend	STRING	Bind or MySQL	pDNS server backend type
postCommand	STRING	/path/to/stuff/postCommand	Command to execute after zone transfer
preCommand	STRING	/path/to/stuff/preCommand	Command to execute before zone transfer
remoteDirectory	STRING	/tmp/zones	Zone Directory on Server

	<table><tr><td>serverType</td><td>STRING</td><td>Master</td><td>Values are 'Master' or 'Slave' only</td></tr><tr><td>username</td><td>STRING</td><td>bobuser</td><td>Login name for Server</td></tr></table>	serverType	STRING	Master	Values are 'Master' or 'Slave' only	username	STRING	bobuser	Login name for Server
serverType	STRING	Master	Values are 'Master' or 'Slave' only						
username	STRING	bobuser	Login name for Server						
Example URL	/api.php?target=dnsServer&action=update&id=74&transferType=SCP&dns.yourdomain.com &SOA=ns1.6connect.com.+hostmaster.6connect.com.								

### *transferByServer*

URL	/api/v1/api.php?target=dnsServer&action=transferByServer											
Description	Performs a full zone push on a DNS Server, executing pre and post commands, transferring files, and restarting services.											
Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td colspan="3">{ "success":1, "message": "Transfer Successful." }</td></tr><tr><td>ERROR:</td><td colspan="3">{ "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><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>push</td><td>INTEGER</td><td>1</td><td>The ID of the server to push zones to</td></tr></table>				Name	Type	Example	Description	push	INTEGER	1	The ID of the server to push zones to
Name	Type	Example	Description									
push	INTEGER	1	The ID of the server to push zones to									
Optional Parameters	None											
Example URL	/api/v1/api.php?target=dnsServer&action=transferByServer&push=1											

### *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> <tr> <td>SUCCESSFUL:</td><td>{"success":1,"message":"Updated Zone: \$name.zone on \$server via SCP"}</td></tr> <tr> <td>ERROR:</td><td>{"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><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>zoneid</td><td>INTEGER</td><td>35</td><td>The ID of the zone to push</td></tr></table>	Name	Type	Example	Description	zoneid	INTEGER	35	The ID of the zone to push
Name	Type	Example	Description						
zoneid	INTEGER	35	The ID of the zone to push						
Optional Parameters	None								
Example URL	/api/v1/api.php?target=dnsServer&action=transferSingle&zoneid=35								

## 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	<p><b>Examples:</b></p> <table><tr><td>SUCCESSFUL:</td><td><pre>{ "success":1, "message":"Search Successful.", "data":[ { "zoneId":"932", "zoneName":"185.160.209.inaddr.arpa.", "zoneSerial":"2013040302", "zoneResourceId":"1", "zoneTags":null, "zoneTTL":"28800", "recordHost":"185.160.209.inaddr.arpa.", "recordTTL":"28800", "recordOrdering":"1", "recordErrors":null, "assetId":"0", "userId":"root" } ] }</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><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td>zoneId</td><td>INTEGER</td><td>The Id of the Zone entry. A single Zone entry might have multiple Records.</td></tr><tr><td>zoneName</td><td>STRING</td><td>The Zone name.</td></tr><tr><td>zoneResourceId</td><td>INTEGER</td><td>The resource Id associated with this Zone.</td></tr><tr><td>zoneSerial</td><td>INTEGER</td><td>Zone Serial.</td></tr></table>		SUCCESSFUL:	<pre>{ "success":1, "message":"Search Successful.", "data":[ { "zoneId":"932", "zoneName":"185.160.209.inaddr.arpa.", "zoneSerial":"2013040302", "zoneResourceId":"1", "zoneTags":null, "zoneTTL":"28800", "recordHost":"185.160.209.inaddr.arpa.", "recordTTL":"28800", "recordOrdering":"1", "recordErrors":null, "assetId":"0", "userId":"root" } ] }</pre>	ERROR:	<pre>{ "success":0, "message":"error message" }</pre>	Name	Type	Description	zoneId	INTEGER	The Id of the Zone entry. A single Zone entry might have multiple Records.	zoneName	STRING	The Zone name.	zoneResourceId	INTEGER	The resource Id associated with this Zone.	zoneSerial	INTEGER	Zone Serial.
SUCCESSFUL:	<pre>{ "success":1, "message":"Search Successful.", "data":[ { "zoneId":"932", "zoneName":"185.160.209.inaddr.arpa.", "zoneSerial":"2013040302", "zoneResourceId":"1", "zoneTags":null, "zoneTTL":"28800", "recordHost":"185.160.209.inaddr.arpa.", "recordTTL":"28800", "recordOrdering":"1", "recordErrors":null, "assetId":"0", "userId":"root" } ] }</pre>																				
ERROR:	<pre>{ "success":0, "message":"error message" }</pre>																				
Name	Type	Description																			
zoneId	INTEGER	The Id of the Zone entry. A single Zone entry might have multiple Records.																			
zoneName	STRING	The Zone name.																			
zoneResourceId	INTEGER	The resource Id associated with this Zone.																			
zoneSerial	INTEGER	Zone Serial.																			

zoneRefresh	INTEGER	Zone Refresh.
zoneRetry	INTEGER	Zone Retry.
zoneExpire	INTEGER	Zone Expire.
zoneMinimum	INTEGER	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	INTEGER	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	INTEGER	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	INTEGER	If pagination is used, this value will contain a total count of records had the pagination not been used.
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.
generalFlag	BOOL	1	When 1, searches over the provided parameters using OR. If 0 or omitted, uses AND.
selectCount	INTEGER	30	When supplied only returns the first X entries
selectOffset	INTEGER	10	When supplied, only returns entries after record X
sortArray	JSON	<code>{"zoneName":"dest","zoneMask":"a</code>	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	INTEGER	123	The Zone Id to search for.
zoneName	STRING	foo	The Zone Name to search for.
zoneResourceId	INTEGER	5	The Resource Id to search for.
zoneSerial	INTEGER	2012033001	The Zone Serial to search for.



	zoneRefresh	INTEGER	36000	The Zone Refresh to search for.
	zoneRetry	INTEGER	800	The Zone Retry to search for.
	zoneExpire	INTEGER	6090000	The Zone Expire to search for.
	zoneMinimum	INTEGER	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	INTEGER	3600	The Zone TTL to search for.
	zoneEnableDNSSEC	INTEGER	1	Search based on DNSSEC settings.
	recordId	INTEGER	123	The Record Id to search for.
	recordZoneId	INTEGER	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.
Example URL	/api/v1/api.php?target=zone&action=get&zoneId=123			

## 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	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td><code>{"success":1,"message":"Search Successful.", "data":{"zoneId":"123"}}</code></td></tr><tr><td>ERROR:</td><td><code>{"success":0, "message":"error message"}</code></td></tr></table> <div>Data Detail:</div> <table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td>zoneId</td><td>INTEGER</td><td>The Id of the Zone entry. A single Zone entry might have multiple Records.</td></tr><tr><td>zoneName</td><td>STRING</td><td>The Zone name.</td></tr><tr><td>zoneResourceId</td><td>INTEGER</td><td>The resource Id associated with this Zone.</td></tr><tr><td>zoneSerial</td><td>INTEGER</td><td>Zone Serial.</td></tr><tr><td>zoneRefresh</td><td>INTEGER</td><td>Zone Refresh.</td></tr><tr><td>zoneRetry</td><td>INTEGER</td><td>Zone Retry.</td></tr><tr><td>zoneExpire</td><td>INTEGER</td><td>Zone Expire.</td></tr><tr><td>zoneMinimum</td><td>INTEGER</td><td>Zone Minimum.</td></tr><tr><td>zoneSOA</td><td>STRING</td><td>Zone SOA.</td></tr><tr><td>zoneTags</td><td>STRING</td><td>All the tags associated with this Zone.</td></tr><tr><td>zoneTTL</td><td>STRING</td><td>Zone TTL.</td></tr><tr><td>zoneEnableDNSSEC</td><td>BOOL</td><td>Whether or not DNSSEC is enabled for this Zone.</td></tr><tr><td>zoneAutoCheck</td><td>BOOL</td><td>Whether or not this zone is configured to be automatically validated on load/edit.</td></tr><tr><td>recordCount</td><td>INTEGER</td><td>How many records are associated with this zone.</td></tr></table>	SUCCESSFUL:	<code>{"success":1,"message":"Search Successful.", "data":{"zoneId":"123"}}</code>	ERROR:	<code>{"success":0, "message":"error message"}</code>	Name	Type	Description	zoneId	INTEGER	The Id of the Zone entry. A single Zone entry might have multiple Records.	zoneName	STRING	The Zone name.	zoneResourceId	INTEGER	The resource Id associated with this Zone.	zoneSerial	INTEGER	Zone Serial.	zoneRefresh	INTEGER	Zone Refresh.	zoneRetry	INTEGER	Zone Retry.	zoneExpire	INTEGER	Zone Expire.	zoneMinimum	INTEGER	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	INTEGER	How many records are associated with this zone.
SUCCESSFUL:	<code>{"success":1,"message":"Search Successful.", "data":{"zoneId":"123"}}</code>																																																	
ERROR:	<code>{"success":0, "message":"error message"}</code>																																																	
Name	Type	Description																																																
zoneId	INTEGER	The Id of the Zone entry. A single Zone entry might have multiple Records.																																																
zoneName	STRING	The Zone name.																																																
zoneResourceId	INTEGER	The resource Id associated with this Zone.																																																
zoneSerial	INTEGER	Zone Serial.																																																
zoneRefresh	INTEGER	Zone Refresh.																																																
zoneRetry	INTEGER	Zone Retry.																																																
zoneExpire	INTEGER	Zone Expire.																																																
zoneMinimum	INTEGER	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	INTEGER	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	BOOI	Whether or not the user has DNS DELETE permissions on this zone's resource
	unpagedRows	INTEGER	If pagination is used, this value will contain a total count of records had the pagination not been used.
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.
generalFlag	BOOL	1	When 1, searches over the provided parameters using OR. If 0 or omitted, uses AND.
selectCount	INTEGER	30	When supplied only returns the first X entries
selectOffset	INTEGER	10	When supplied, only returns entries after record X
sortArray	JSON	<code>{"zoneName":"desc","zoneMask":"a</code>	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	INTEGER	123	The Zone Id to search for.
zoneName	STRING	foo	The Zone Name to search for.
zoneResourceId	INTEGER	5	The Resource Id to search for.
zoneSerial	INTEGER	2012033001	The Zone Serial to search for.

	zoneRefresh	INTEGER	36000	The Zone Refresh to search for.
	zoneRetry	INTEGER	800	The Zone Retry to search for.
	zoneExpire	INTEGER	6090000	The Zone Expire to search for.
	zoneMinimum	INTEGER	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	INTEGER	3600	The Zone TTL to search for.
	zoneEnableDNSSEC	INTEGER	1	Search based on DNSSEC settings.
	recordId	INTEGER	123	The Record Id to search for.
	recordZoneId	INTEGER	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.
Example URL		/api/v1/api.php?target=zone&action=search&zoneId=123		

### update

URL	/api/v1/api.php?target=zone&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	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td colspan="3">{ "success":1, "message":"Update Successful." }</td></tr><tr><td>ERROR:</td><td colspan="3">{ "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	None																																											
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><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></table> <table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>searchZoneId</td><td>INTEGER</td><td>123</td><td>The Zone Id to search for.</td></tr><tr><td>searchZoneName</td><td>STRING</td><td>foo</td><td>The Zone Name to search for.</td></tr><tr><td>searchZoneResourceId</td><td>INTEGER</td><td>5</td><td>The Resource Id to search for.</td></tr><tr><td>searchZoneSerial</td><td>INTEGER</td><td>2012033001</td><td>The Zone Serial to search for.</td></tr><tr><td>searchZoneRefresh</td><td>INTEGER</td><td>36000</td><td>The Zone Refresh to search for.</td></tr><tr><td>searchZoneRetry</td><td>INTEGER</td><td>800</td><td>The Zone Retry to search for.</td></tr></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	searchZoneId	INTEGER	123	The Zone Id to search for.	searchZoneName	STRING	foo	The Zone Name to search for.	searchZoneResourceId	INTEGER	5	The Resource Id to search for.	searchZoneSerial	INTEGER	2012033001	The Zone Serial to search for.	searchZoneRefresh	INTEGER	36000	The Zone Refresh to search for.	searchZoneRetry	INTEGER	800	The Zone Retry to search for.
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	INTEGER	123	The Zone Id to search for.																																									
searchZoneName	STRING	foo	The Zone Name to search for.																																									
searchZoneResourceId	INTEGER	5	The Resource Id to search for.																																									
searchZoneSerial	INTEGER	2012033001	The Zone Serial to search for.																																									
searchZoneRefresh	INTEGER	36000	The Zone Refresh to search for.																																									
searchZoneRetry	INTEGER	800	The Zone Retry to search for.																																									

searchZoneExpire	INTEGER	6090000	The Zone Expire to search for.
searchZoneMinimum	INTEGER	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	INTEGER	3600	The Zone TTL to search for.
searchZoneEnableDNSSEC	INTEGER	1	Search based on DNSSEC settings.
searchRecordId	INTEGER	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.
updateZoneResource	INTEGER	5	The Resource Id to replace into the searched rows.

updateZoneSerial	INTEGER	2012033001	The Zone Serial to replace into the searched rows.
updateZoneRefresh	INTEGER	36000	The Zone Refresh to replace into the searched rows.
updateZoneRetry	INTEGER	800	The Zone Retry to replace into the searched rows..
updateZoneExpire	INTEGER	6090000	The Zone Expire to replace into the searched rows.
updateZoneMinimum	INTEGER	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	INTEGER	3600	The Zone TTL to replace into the searched rows.
updateZoneEnableDNSSEC	BOOLEAN	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.



	updateRecordValue	STRING	ns1.dns.6connect.com	The Record Value to replace into the searched rows.								
	updateRecordDescription	STRING	Description	Update Record Descriptions.								
	updateRecordTTL	STRING	3600	The Record TTL to replace into the searched rows.								
	updateZoneAutoCheck	BOOL	1	Whether or not this zone is configured to be automatically validated on load/edit.								
<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>recordZoneId</td><td>INTEGER</td><td>123</td><td>The parent zone ID</td></tr></table>					Name	Type	Example	Description	recordZoneId	INTEGER	123	The parent zone ID
Name	Type	Example	Description									
recordZoneId	INTEGER	123	The parent zone ID									
Example URL	/api/v1/api.php?target=zone&action=update&searchZoneId=123&updateZoneExpire=6090000											

Required Parameters

Name	Type	Example	Description
zoneName	STRING	254.221.67.in-addr.arpa	The name for the new Zone.
zoneResourceId	STRING	123	Resource Id for the new Zone.

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.
zoneIpver	STRING	IPv6	The IP Version.
zoneLocalSigning	BOOL	1	Whether or not this zone should be signed by the ProVision server when DNSSEC is enabled. If set to false, ProVision will deliver the zone unsigned to the DNS server and the signing / updating process should be triggered by the post-push command
zoneSerial	INTEGER	2012033001	Serial for the new Zone.
zoneRefresh	INTEGER	36000	Refresh for the new Zone.
zoneRetry	INTEGER	800	Retry for the new Zone.
zoneExpire	INTEGER	6090000	Expire for the new Zone.
zoneMinimum	INTEGER	10	Minimum for the new Zone.
zoneSOA	STRING	200	SOA for the new Zone.
zoneTags	STRING	client,production	Tags for the new Zone.
zoneTTL	STRING	3600	TTL for the new Zone.
zoneEnableDNSSEC	INTEGER	1	Whether or not this new zone uses DNSSEC.

Example URL	/api/v1/api.php?target=zone&action=add&zoneName=254.221.67.in-addr.arpa&zoneResourceId=123&zoneSerial=2012033001
-------------	--

## *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	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td colspan="3">{ "success":1, "message":"Zones and Associated Records Deleted." }</td></tr><tr><td>ERROR:</td><td colspan="3">{ "success":0, "message":"error message" }</td></tr></table>				SUCCESSFUL:	{ "success":1, "message":"Zones and Associated Records Deleted." }			ERROR:	{ "success":0, "message":"error message" }																																		
SUCCESSFUL:	{ "success":1, "message":"Zones and Associated Records Deleted." }																																											
ERROR:	{ "success":0, "message":"error message" }																																											
Required Parameters	No specific parameter is required, however, one or more optional parameters must be used for a successful return																																											
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>deleteZoneId</td><td>INTEGER</td><td>123</td><td>The Zone Id to search for.</td></tr><tr><td>deleteZoneName</td><td>STRING</td><td>foo</td><td>The Zone Name to search for.</td></tr><tr><td>deleteZoneResourceId</td><td>INTEGER</td><td>5</td><td>The Resource Id to search for.</td></tr><tr><td>deleteZoneSerial</td><td>INTEGER</td><td>2012033001</td><td>The Zone Serial to search for.</td></tr><tr><td>deleteZoneRefresh</td><td>INTEGER</td><td>36000</td><td>The Zone Refresh to search for.</td></tr><tr><td>deleteZoneRetry</td><td>INTEGER</td><td>800</td><td>The Zone Retry to search for.</td></tr><tr><td>deleteZoneExpire</td><td>INTEGER</td><td>6090000</td><td>The Zone Expire to search for.</td></tr><tr><td>deleteZoneMinimum</td><td>INTEGER</td><td>10</td><td>The Zone Minimum to search for.</td></tr><tr><td>deleteZoneSOA</td><td>STRING</td><td>200</td><td>The Zone SOA to search for.</td></tr></table>				Name	Type	Example	Description	deleteZoneId	INTEGER	123	The Zone Id to search for.	deleteZoneName	STRING	foo	The Zone Name to search for.	deleteZoneResourceId	INTEGER	5	The Resource Id to search for.	deleteZoneSerial	INTEGER	2012033001	The Zone Serial to search for.	deleteZoneRefresh	INTEGER	36000	The Zone Refresh to search for.	deleteZoneRetry	INTEGER	800	The Zone Retry to search for.	deleteZoneExpire	INTEGER	6090000	The Zone Expire to search for.	deleteZoneMinimum	INTEGER	10	The Zone Minimum to search for.	deleteZoneSOA	STRING	200	The Zone SOA to search for.
Name	Type	Example	Description																																									
deleteZoneId	INTEGER	123	The Zone Id to search for.																																									
deleteZoneName	STRING	foo	The Zone Name to search for.																																									
deleteZoneResourceId	INTEGER	5	The Resource Id to search for.																																									
deleteZoneSerial	INTEGER	2012033001	The Zone Serial to search for.																																									
deleteZoneRefresh	INTEGER	36000	The Zone Refresh to search for.																																									
deleteZoneRetry	INTEGER	800	The Zone Retry to search for.																																									
deleteZoneExpire	INTEGER	6090000	The Zone Expire to search for.																																									
deleteZoneMinimum	INTEGER	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	INTEGER	3600	The Zone TTL to search for.
	deleteZoneEnableDNSSEC	BOOLEAN	1	Search based on DNSSEC settings.
	deleteRecordId	INTEGER	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.
	deleteRecordZoneId	INTEGER	123	The parent zone ID
Example URL	/api/v1/api.php?target=zone&action=delete&deleteZoneId=123			

getRecordTypes												
URL	/api/v1/api.php?target=zone&action=getRecordTypes											
Description	Returns a list of all Record Types allowed by the system.											
Returns	<div><div>Examples:</div><table><tr><td>SUCCESSFUL:</td><td>{ "success":1, "message": "Search Successful. ", "data": [{"recordType": " "}</td></tr><tr><td>ERROR:</td><td>{ "success":0, "message": "error message" }</td></tr></table></div> <div><div>Data Detail:</div><table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td>recordType</td><td>STRING</td><td>A Record Type</td></tr></table></div>		SUCCESSFUL:	{ "success":1, "message": "Search Successful. ", "data": [{"recordType": " "}	ERROR:	{ "success":0, "message": "error message" }	Name	Type	Description	recordType	STRING	A Record Type
SUCCESSFUL:	{ "success":1, "message": "Search Successful. ", "data": [{"recordType": " "}											
ERROR:	{ "success":0, "message": "error message" }											
Name	Type	Description										
recordType	STRING	A Record Type										
Required Parameters	None											

Optional Parameters	None
Example URL	/api/v1/api.php?target=zone&action=getRecordTypes

### *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				
	Name	Type	Example	Description
	zoneId	INTEGER	50	The Id of the zone to retrieve.
	format	ENUMERATED	'html' or "	If html, the zone file will be formatted for display via a web browser. If blank or omitted, the zone file will be formatted for display in a file system.
	unsigned	BOOL	1	For a DNSSEC-enabled zone, determines whether or not the system retrieves the signed or unsigned zone file. Ignored for non-DNSSEC zones.
Optional Parameters	None			
Example URL	/api/v1/api.php?target=zone&action=getFile&zoneId=50&zoneId=50&format=html&unsigned=1			

### *getDSFile*

URL	/api/v1/api.php?target=zone&action=getDSFile
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><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>zoneld</td><td>INTEGER</td><td>50</td><td>The Id of the zone whose DS keys are to be retrieved.</td></tr></table>	Name	Type	Example	Description	zoneld	INTEGER	50	The Id of the zone whose DS keys are to be retrieved.
	Name	Type	Example	Description					
zoneld	INTEGER	50	The Id of the zone whose DS keys are to be retrieved.						
Optional Parameters	None								
Example URL	/api/v1/api.php?target=zone&action=getDSFile&zoneld=50								

### *checkZone*

URL	/api/v1/api.php?target=zone&action=checkZone											
Description	Runs a zone file through Named checkzone											
Returns	<div><b>Examples:</b></div> <table><tr><td>SUCCESSFUL:</td><td colspan="3">{ "success":1, "message": "No errors found." }</td></tr><tr><td>ERROR:</td><td colspan="3">{ "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::\V32': 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::\V32': 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::\V32': bad IPv6 address " }											
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>zoneld</td><td>INTEGER</td><td>50</td><td>The Id of the zone to check.</td></tr></table>				Name	Type	Example	Description	zoneld	INTEGER	50	The Id of the zone to check.
Name	Type	Example	Description									
zoneld	INTEGER	50	The Id of the zone to check.									
Optional Parameters	None											
Example URL	/api/v1/api.php?target=zone&action=checkZone&zoneld=50											

### *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	<b>Examples:</b>

SUCCESSFUL:	{ "success":1,"message":"Search Successful.", "data":{ "zoneArchiveId":2768,"zoneId":1227,"zoneArchiveTimestamp":1375298,"zoneArchiveFingerprint":"d060e59","zoneMask":null,"zoneSerial":2013073105,"zoneRefresh":1440,"zoneMinimum":3600,"zoneSOA":null,"zoneTags":null,"zoneResourceId":1013,"zonePreviousViewLinkage":null }}
ERROR:	{ "success":0,"message":"error message"}

#### Data Detail:

Name	Type	Description
zoneId	INTEGER	The Id of the Zone entry to find archived versions of.
zoneArchiveId	INTEGER	The ID of the Archive Entry
zoneArchiveTimestamp	INTEGER	A timestamp marking when this zone was archived.
zoneArchiveFingerprint	STRING	A hash value identifying this zone. Used for comparing versions.
zoneName	INTEGER	Zone Name.
zoneSerial	INTEGER	Zone Serial.
zoneRefresh	INTEGER	Zone Refresh.
zoneRetry	INTEGER	Zone Retry.
zoneExpire	INTEGER	Zone Expire.
zoneMinimum	INTEGER	Zone Minimum.
zoneSOA	STRING	Zone SOA.
zoneTags	STRING	Zone Tags.
zoneTTL	INTEGER	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.



Required Parameters	None																				
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>zoneId</td><td>INTEGER</td><td>123</td><td>The Zone Id to search for.</td></tr><tr><td>zoneArchiveId</td><td>INTEGER</td><td>123</td><td>The Zone Archive Id</td></tr><tr><td>zoneArchiveTimestamp</td><td>INTEGER</td><td>2012033001</td><td>The Zone Archive Timestamp</td></tr><tr><td>fetchArchiveFile</td><td>BOOL</td><td>1</td><td>Whether or not to return the full Zone file with the result set..</td></tr></table>	Name	Type	Example	Description	zoneId	INTEGER	123	The Zone Id to search for.	zoneArchiveId	INTEGER	123	The Zone Archive Id	zoneArchiveTimestamp	INTEGER	2012033001	The Zone Archive Timestamp	fetchArchiveFile	BOOL	1	Whether or not to return the full Zone file with the result set..
Name	Type	Example	Description																		
zoneId	INTEGER	123	The Zone Id to search for.																		
zoneArchiveId	INTEGER	123	The Zone Archive Id																		
zoneArchiveTimestamp	INTEGER	2012033001	The Zone Archive Timestamp																		
fetchArchiveFile	BOOL	1	Whether or not to return the full Zone file with the result set..																		
Example URL	/api/v1/api.php?target=zone&action=getArchivedZone&zoneId=123																				

## DNS Record Control

<i>get</i>	
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:	{ "success":1, "message":"Search Successful.", "data":[{"recordId":"300"}]}
ERROR:	{ "success":0, "message":"error message" }

Data Detail:

Name	Type	Description
recordId	INTEGER	The ID of this Record Entry. It is always included with its parent Zone.
recordZoneId	INTEGER	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.

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.
selectCount	INTEGER	30	When supplied only returns the first X entries
selectOffset	INTEGER	10	When supplied, only returns entries after record X

Name	Type	Example	Description
recordId	INTEGER	123	The Record ID to search for.
recordZoneId	INTEGER	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.

		Name	Type	Example	Description
		zoneId	INTEGER	123	The Zone Id to search for.
		zoneName	STRING	foo	The Zone Name to search for.
		zoneResourceId	INTEGER	5	The Resource Id to search for.
		zoneCustName	STRING	foo	The Customer Name to search for.
		zoneIpver	STRING	IPv6	The IP Version to search for.
		zoneSerial	INTEGER	2012033001	The Zone Serial to search for.
		zoneRefresh	INTEGER	36000	The Zone Refresh to search for.
		zoneRetry	INTEGER	800	The Zone Retry to search for.
		zoneExpire	INTEGER	6090000	The Zone Expire to search for.
		zoneMinimum	INTEGER	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	INTEGER	3600	The Zone TTL to search for.
		zoneEnableDNSSEC	INTEGER	1	Search based on DNSSEC settings.
Example URL		/api/v1/api.php?target=record&action=get&selectCount=30&zoneId=123			

### 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	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td colspan="3">{ "success":1, "message":"Update Successful." }</td></tr><tr><td>ERROR:</td><td colspan="3">{ "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	None																																											
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><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></table> <table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>searchZoneId</td><td>INTEGER</td><td>123</td><td>The Zone ID to search for.</td></tr><tr><td>searchZoneName</td><td>STRING</td><td>foo</td><td>The Zone Name to search for.</td></tr><tr><td>searchZoneCustomerId</td><td>INTEGER</td><td>5</td><td>The Customer ID to search for.</td></tr><tr><td>searchZoneSerial</td><td>INTEGER</td><td>2012033001</td><td>The Zone Serial to search for.</td></tr><tr><td>searchZoneRefresh</td><td>INTEGER</td><td>36000</td><td>The Zone Refresh to search for.</td></tr><tr><td>searchZoneRetry</td><td>INTEGER</td><td>800</td><td>The Zone Retry to search for.</td></tr></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	searchZoneId	INTEGER	123	The Zone ID to search for.	searchZoneName	STRING	foo	The Zone Name to search for.	searchZoneCustomerId	INTEGER	5	The Customer ID to search for.	searchZoneSerial	INTEGER	2012033001	The Zone Serial to search for.	searchZoneRefresh	INTEGER	36000	The Zone Refresh to search for.	searchZoneRetry	INTEGER	800	The Zone Retry to search for.
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	INTEGER	123	The Zone ID to search for.																																									
searchZoneName	STRING	foo	The Zone Name to search for.																																									
searchZoneCustomerId	INTEGER	5	The Customer ID to search for.																																									
searchZoneSerial	INTEGER	2012033001	The Zone Serial to search for.																																									
searchZoneRefresh	INTEGER	36000	The Zone Refresh to search for.																																									
searchZoneRetry	INTEGER	800	The Zone Retry to search for.																																									

searchZoneExpire	INTEGER	6090000	The Zone Expire to search for.
searchZoneMinimum	INTEGER	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	INTEGER	3600	The Zone TTL to search for.
searchZoneEnableDNSSEC	INTEGER	1	Search based on DNSSEC settings.
searchRecordId	INTEGER	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.
searchZoneResourceId	INTEGER	5	The Resource Id to search for.
searchRecordZoneId	INTEGER	123	The Zone ID of the Record to search for.

Name	Type	Example	Description
updateZoneName	STRING	foo	The Zone name to replace into the searched rows.

updateZoneSerial	INTEGER	2012033001	The Zone Serial to replace into the searched rows.
updateZoneRefresh	INTEGER	36000	The Zone Refresh to replace into the searched rows.
updateZoneRetry	INTEGER	800	The Zone Retry to replace into the searched rows..
updateZoneExpire	INTEGER	6090000	The Zone Expire to replace into the searched rows.
updateZoneMinimum	INTEGER	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	INTEGER	3600	The Zone TTL to replace into the searched rows.
updateZoneEnableDNSSEC	BOOLEAN	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.

	updateRecordValue	STRING	ns1.dns.6connect.com	The Record Value to replace into the searched rows.
	updateRecordDescription	STRING	Description	Update Record Descriptions.
	updateRecordTTL	STRING	3600	The Record TTL to replace into the searched rows.
	updateZoneResource	INTEGER	5	The Resource Id to replace into the searched rows.
	updateZoneAutoCheck	BOOL	1	Whether or not this zone is configured to be automatically validated on load/edit.
Example URL	/api/v1/api.php?target=record&action=update&searchZoneId=123&searchZoneTags=client&updateZoneTTL=3600			

add

URL	/api/v1/api.php?target=record&action=add														
Description	Adds a new Record to a supplied Zone.														
Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td colspan="2">{ "success":1, "message":"Add Successful.", "data":123 }</td></tr><tr><td>ERROR:</td><td colspan="2">{ "success":0, "message":"error message" }</td></tr></table> <div>Data Detail:</div> <table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td>data</td><td>INTEGER</td><td>The ID of the new Record entry.</td></tr></table>			SUCCESSFUL:	{ "success":1, "message":"Add Successful.", "data":123 }		ERROR:	{ "success":0, "message":"error message" }		Name	Type	Description	data	INTEGER	The ID of the new Record entry.
SUCCESSFUL:	{ "success":1, "message":"Add Successful.", "data":123 }														
ERROR:	{ "success":0, "message":"error message" }														
Name	Type	Description													
data	INTEGER	The ID of the new Record entry.													



Required Parameters				
	<b>Name</b>	<b>Type</b>	<b>Example</b>	<b>Description</b>
	newRecordZoneId	INTEGER	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.
*newRecordValue required only for certain Record Types				
Optional Parameters				
	<b>Name</b>	<b>Type</b>	<b>Example</b>	<b>Description</b>
	likeFlag	BOOL	1	When 1, string searches are done via LIKE with wildcards at both ends. When 0, strict comparison is used.
	newRecordDescription	STRING	Description.	Notes for the Record.
Example URL	newRecordTTL	INTEGER	3600	Record TTL.
	/api/v1/api.php?target=record&action=add&newRecordZoneId=123&newRecordHost=@host&newRecordType=PTR&newRecordTTL=3600			

<i>delete</i>												
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	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td colspan="3">{ "success":1, "message":"Deletion Successful." }</td></tr><tr><td>ERROR:</td><td colspan="3">{ "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" }											
Required Parameters	None											
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr></table>				Name	Type	Example	Description				
Name	Type	Example	Description									

deleteZoneId	INTEGER	123	The Zone ID to search for.
deleteZoneName	STRING	foo	The Zone Name to search for.
deleteZoneCustomerId	INTEGER	5	The Customer ID to search for.
deleteZoneIpver	STRING	IPv6	The IP Version to search for.
deleteZoneSerial	INTEGER	2012033001	The Zone Serial to search for.
deleteZoneRefresh	INTEGER	36000	The Zone Refresh to search for.
deleteZoneRetry	INTEGER	800	The Zone Retry to search for.
deleteZoneExpire	INTEGER	6090000	The Zone Expire to search for.
deleteZoneMinimum	INTEGER	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	INTEGER	3600	The Zone TTL to search for.
deleteZoneEnableDNSSEC	DNSSEC	1	Search based on DNSSEC settings.
deleteRecordId	INTEGER	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.
	deleteZoneResourceId	INTEGER	5	The Resource Id to search for.
	deleteZoneCustName	STRING	foo	The Customer Name to search for.
Example URL	/api/v1/api.php?target=record&action=delete&deleteZoneName=foo			

switch																
URL	/api/v1/api.php?target=record&action=switch															
Description	Switches the order of two record entries.															
Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td colspan="3">{ "success":1, "message":"Record Moved." }</td></tr><tr><td>ERROR:</td><td colspan="3">{ "success":0, "message":"error message" }</td></tr></table>				SUCCESSFUL:	{ "success":1, "message":"Record Moved." }			ERROR:	{ "success":0, "message":"error message" }						
SUCCESSFUL:	{ "success":1, "message":"Record Moved." }															
ERROR:	{ "success":0, "message":"error message" }															
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>moveWhichId</td><td>INTEGER</td><td>123</td><td>The Record Id to be moved.</td></tr><tr><td>moveAfterId</td><td>INTEGER</td><td>42</td><td>The Id of the Record the first Record is to be moved after.</td></tr></table>				Name	Type	Example	Description	moveWhichId	INTEGER	123	The Record Id to be moved.	moveAfterId	INTEGER	42	The Id of the Record the first Record is to be moved after.
Name	Type	Example	Description													
moveWhichId	INTEGER	123	The Record Id to be moved.													
moveAfterId	INTEGER	42	The Id of the Record the first Record is to be moved after.													
Optional Parameters	None															
Example URL	/api/v1/api.php?target=record&action=switch&moveWhichId=123&moveAfterId=42															

## Server-Zone Linkage

<b>get</b>	
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	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td>{ "success":1, "message":"2 rows retrieved. ", "data":[{"id":"285", "zoneId":15, "serverId":1, "serverName":"server1", "serverType":"server", "serverMasterType":"master", "zoneName":"zone1", "resourceId":1}]} </td></tr><tr><td>ERROR:</td><td>{ "success":0, "message":"error message"}</td></tr></table> <div>Data Detail:</div> <table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td>id</td><td>INTEGER</td><td>The Linkage Id.</td></tr><tr><td>zoneId</td><td>INTEGER</td><td>The ZoneId involved in this link.</td></tr><tr><td>serverId</td><td>INTEGER</td><td>The ServerId involved in this link.</td></tr><tr><td>serverName</td><td>STRING</td><td>The server name</td></tr><tr><td>serverType</td><td>STRING</td><td>The server transfer type</td></tr><tr><td>serverMasterType</td><td>STRING</td><td>Whether this server is a master or a slave.</td></tr><tr><td>zoneName</td><td>STRING</td><td>The zone name</td></tr><tr><td>resourceId</td><td>INTEGER</td><td>The Resource Id the Zone is attached to.</td></tr></table>	SUCCESSFUL:	{ "success":1, "message":"2 rows retrieved. ", "data":[{"id":"285", "zoneId":15, "serverId":1, "serverName":"server1", "serverType":"server", "serverMasterType":"master", "zoneName":"zone1", "resourceId":1}]}	ERROR:	{ "success":0, "message":"error message"}	Name	Type	Description	id	INTEGER	The Linkage Id.	zoneId	INTEGER	The ZoneId involved in this link.	serverId	INTEGER	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	INTEGER	The Resource Id the Zone is attached to.
SUCCESSFUL:	{ "success":1, "message":"2 rows retrieved. ", "data":[{"id":"285", "zoneId":15, "serverId":1, "serverName":"server1", "serverType":"server", "serverMasterType":"master", "zoneName":"zone1", "resourceId":1}]}																															
ERROR:	{ "success":0, "message":"error message"}																															
Name	Type	Description																														
id	INTEGER	The Linkage Id.																														
zoneId	INTEGER	The ZoneId involved in this link.																														
serverId	INTEGER	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	INTEGER	The Resource Id the Zone is attached to.																														
Required Parameters	None																															
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>id</td><td>INTEGER</td><td>15</td><td>Fetches the linkage with the matching id.</td></tr><tr><td>serverId</td><td>INTEGER</td><td>15</td><td>Fetches all linkages with the matching serverId.</td></tr><tr><td>zoneId</td><td>INTEGER</td><td>15</td><td>Fetches all linkages with the matching zoneId.</td></tr></table>	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	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.																													
Example URL	/api/v1/api.php?target=zoneLinkage&action=get&id=15																															

### *add*

URL	/api/v1/api.php?target=zoneLinkage&action=add
-----	---

Description	Adds a new link between a DNS Server and a Zone																			
Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td colspan="3">{ "success":1, "message":"Link Added." }</td></tr><tr><td>ERROR:</td><td colspan="3">{ "success":0, "message":"error message" }</td></tr></table>				SUCCESSFUL:	{ "success":1, "message":"Link Added." }			ERROR:	{ "success":0, "message":"error message" }										
SUCCESSFUL:	{ "success":1, "message":"Link Added." }																			
ERROR:	{ "success":0, "message":"error message" }																			
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>serverId</td><td>INTEGER</td><td>16</td><td>The DNS Server Id.</td></tr><tr><td>zoneId</td><td>INTEGER</td><td>105</td><td>The Zone Id.</td></tr><tr><td>serverSlave</td><td>BOOL</td><td>1</td><td>Whether or not this zone is a master or a slave on the linked server. Values are: 1 for slave, 0 for master.</td></tr></table>				Name	Type	Example	Description	serverId	INTEGER	16	The DNS Server Id.	zoneId	INTEGER	105	The Zone Id.	serverSlave	BOOL	1	Whether or not this zone is a master or a slave on the linked server. Values are: 1 for slave, 0 for master.
Name	Type	Example	Description																	
serverId	INTEGER	16	The DNS Server Id.																	
zoneId	INTEGER	105	The Zone Id.																	
serverSlave	BOOL	1	Whether or not this zone is a master or a slave on the linked server. Values are: 1 for slave, 0 for master.																	
Optional Parameters	None																			
Example URL	/api/v1/api.php?target=zoneLinkage&action=add&serverId=16&zoneId=105&serverSlave=0																			

### *delete*

URL	/api/v1/api.php?target=zoneLinkage&action=delete				
Description	Deletes a link between a DNS Server and a Zone				
Returns	<b>Examples:</b> <table> <tr> <td>SUCCESSFUL:</td><td>{"success":1,"message":"Link Deleted."}</td></tr> <tr> <td>ERROR:</td><td>{"success":0,"message":"error message"}</td></tr> </table>	SUCCESSFUL:	{"success":1,"message":"Link Deleted."}	ERROR:	{"success":0,"message":"error message"}
SUCCESSFUL:	{"success":1,"message":"Link Deleted."}				
ERROR:	{"success":0,"message":"error message"}				
Required Parameters	None				

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.
Example URL	/api/v1/api.php?target=zoneLinkage&action=delete			

## Name Server Control

get																								
URL	/api/v1/api.php?target=nameServer&action=get																							
Description	Fetches a list of all stored Name Servers																							
Returns	<div><b>Examples:</b></div> <table><tr><td>SUCCESSFUL:</td><td><pre>{"success":1,"message":"Fetch Successful.", "data":{"id":"1","nameserver":1}}</pre></td></tr><tr><td>ERROR:</td><td><pre>{"success":0, "message":"error message"}</pre></td></tr></table> <div><b>Data Detail:</b></div> <table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td>id</td><td>INTEGER</td><td>Server ID</td></tr><tr><td>nameserver</td><td>STRING</td><td>Server Name</td></tr><tr><td>add_to_zones_default</td><td>BOOL</td><td>Whether or not this is a default server.</td></tr><tr><td>ordering</td><td>INTEGER</td><td>Display order</td></tr><tr><td>uses</td><td>INTEGER</td><td>How many zones have been assigned to this NameServer</td></tr></table>		SUCCESSFUL:	<pre>{"success":1,"message":"Fetch Successful.", "data":{"id":"1","nameserver":1}}</pre>	ERROR:	<pre>{"success":0, "message":"error message"}</pre>	Name	Type	Description	id	INTEGER	Server ID	nameserver	STRING	Server Name	add_to_zones_default	BOOL	Whether or not this is a default server.	ordering	INTEGER	Display order	uses	INTEGER	How many zones have been assigned to this NameServer
SUCCESSFUL:	<pre>{"success":1,"message":"Fetch Successful.", "data":{"id":"1","nameserver":1}}</pre>																							
ERROR:	<pre>{"success":0, "message":"error message"}</pre>																							
Name	Type	Description																						
id	INTEGER	Server ID																						
nameserver	STRING	Server Name																						
add_to_zones_default	BOOL	Whether or not this is a default server.																						
ordering	INTEGER	Display order																						
uses	INTEGER	How many zones have been assigned to this NameServer																						
Required Parameters	None																							

Optional Parameters	<b>Name</b>	<b>Type</b>	<b>Example</b>	<b>Description</b>
	default	INTEGER	1	Set server as default
Example URL	/api/v1/api.php?target=nameServer&action=get&default=1			

add												
URL	/api/v1/api.php?target=nameServer&action=add											
Description	Adds a new DNS Server											
Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td colspan="3">{<i>"success":1,"message":"Add Successful."</i>}</td></tr><tr><td>ERROR:</td><td colspan="3">{<i>"success":0,"message":"error message"</i>}</td></tr></table>				SUCCESSFUL:	{ <i>"success":1,"message":"Add Successful."</i> }			ERROR:	{ <i>"success":0,"message":"error message"</i> }		
SUCCESSFUL:	{ <i>"success":1,"message":"Add Successful."</i> }											
ERROR:	{ <i>"success":0,"message":"error message"</i> }											
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>newServer</td><td>STRING</td><td>ns.yourdomain.com</td><td>Name of the NameServer</td></tr></table>				Name	Type	Example	Description	newServer	STRING	ns.yourdomain.com	Name of the NameServer
Name	Type	Example	Description									
newServer	STRING	ns.yourdomain.com	Name of the NameServer									
Optional Parameters	None											
Example URL	/api/v1/api.php?target=nameServer&action=add&newServer=ns.yourdomain.com											

delete

URL	/api/v1/api.php?target=nameServer&action=delete							
Description	Deletes a NameServer							
Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td><code>{"success":1,"message":"Server Deleted."}</code></td></tr><tr><td>ERROR:</td><td><code>{"success":0,"message":"error message"}</code></td></tr></table>			SUCCESSFUL:	<code>{"success":1,"message":"Server Deleted."}</code>	ERROR:	<code>{"success":0,"message":"error message"}</code>	
SUCCESSFUL:	<code>{"success":1,"message":"Server Deleted."}</code>							
ERROR:	<code>{"success":0,"message":"error message"}</code>							
Required Parameters								
	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>id</td><td>INTEGER</td><td>5</td><td>ID of server to delete.</td></tr></table>	Name	Type	Example	Description	id	INTEGER	5
Name	Type	Example	Description					
id	INTEGER	5	ID of server to delete.					
Optional Parameters	None							
Example URL	/api/v1/api.php?target=nameServer&action=delete&id=5							

### *setDefault*

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	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td colspan="3">{<i>"success":1,</i> <i>"message":<i>"Success."</i></i>}</td></tr><tr><td>ERROR:</td><td colspan="3">{<i>"success":0,</i> <i>"message":<i>"error message"</i></i>}</td></tr></table>				SUCCESSFUL:	{ <i>"success":1,</i> <i>"message":<i>"Success."</i></i> }			ERROR:	{ <i>"success":0,</i> <i>"message":<i>"error message"</i></i> }						
SUCCESSFUL:	{ <i>"success":1,</i> <i>"message":<i>"Success."</i></i> }															
ERROR:	{ <i>"success":0,</i> <i>"message":<i>"error message"</i></i> }															
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>id</td><td>INTEGER</td><td>5</td><td>ID of server to modify.</td></tr><tr><td>value</td><td>INTEGER</td><td>1</td><td>1 = Default, 0 = Normal</td></tr></table>				Name	Type	Example	Description	id	INTEGER	5	ID of server to modify.	value	INTEGER	1	1 = Default, 0 = Normal
Name	Type	Example	Description													
id	INTEGER	5	ID of server to modify.													
value	INTEGER	1	1 = Default, 0 = Normal													
Optional Parameters	None															
Example URL	/api/v1/api.php?target=nameServer&action=setDefault&id=3&value=1															

### *orderUp*

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	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td colspan="3">{<i>"success":1,"message":"Reordering Successful."</i>}</td></tr><tr><td>ERROR:</td><td colspan="3">{<i>"success":0, "message":"error message"</i>}</td></tr></table>				SUCCESSFUL:	{ <i>"success":1,"message":"Reordering Successful."</i> }			ERROR:	{ <i>"success":0, "message":"error message"</i> }		
SUCCESSFUL:	{ <i>"success":1,"message":"Reordering Successful."</i> }											
ERROR:	{ <i>"success":0, "message":"error message"</i> }											
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>id</td><td>INTEGER</td><td>5</td><td>ID of server to modify.</td></tr></table>				Name	Type	Example	Description	id	INTEGER	5	ID of server to modify.
Name	Type	Example	Description									
id	INTEGER	5	ID of server to modify.									
Optional Parameters	None											
Example URL	/api/v1/api.php?target=nameServer&action=orderUp&id=3											

### *orderDown*

URL	/api/v1/api.php?target=nameServer&action=orderDown
-----	--



Description	Swaps the index order of the targeted NameServer with that of the one below it.											
Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL:</td><td colspan="3">{<i>"success":1,"message":"Reordering Successful."</i>}</td></tr><tr><td>ERROR:</td><td colspan="3">{<i>"success":0, "message":"error message"</i>}</td></tr></table>				SUCCESSFUL:	{ <i>"success":1,"message":"Reordering Successful."</i> }			ERROR:	{ <i>"success":0, "message":"error message"</i> }		
SUCCESSFUL:	{ <i>"success":1,"message":"Reordering Successful."</i> }											
ERROR:	{ <i>"success":0, "message":"error message"</i> }											
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>id</td><td>INTEGER</td><td>5</td><td>ID of server to activate.</td></tr></table>				Name	Type	Example	Description	id	INTEGER	5	ID of server to activate.
Name	Type	Example	Description									
id	INTEGER	5	ID of server to activate.									
Optional Parameters	None											
Example URL	/api/v1/api.php?target=nameServer&action=orderDown&id=5											

## API Module - IPAM

- IP Address Management (IPv4 and IPv6)
  - Get
  - Add
  - Update
  - Delete
  - Add Tag
  - Delete Tag
  - Smart Assign
  - Direct Assign
  - Unassign
  - Simple Reassign
  - Get Tags List
  - Add Tag To List
  - Get RIRs List
  - Get Regions List
  - Add Region To List
  - Get Utilization
  - Get Host Utilization
  - Aggregate
  - Split
  - Scan Block
  - Get Scan Results
  - Get Options
  - Get Resource Hierarchy
  - Get VLAN
  - Process Holding Tank
- IPAM API Calls Subject to Change:
  - Get Attribute List

### IP Address Management (IPv4 and IPv6)

<i>Get</i>	
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

```
{ "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" ] } ] }
```

ERROR

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

Required Parameters

None

Optional Parameters

Name	Type	Example	Description
address	INTEGER	1125449728	IP address of the block in decimal format
asn	INTEGER	1000	Filters blocks based on their ASN

allowSubAssignment	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	Code X	User-defined block code as defined in Admin-IPAM settings: Generic Code Per Block Name
endAddress	INTEGER	1125453823	End IP address of the block in decimal format
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"
lirId	INTEGER	101	The numeric ID of an LIR resource the block should be linked to
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	cust-001	<p><b>(Deprecated:</b> Use resourceQuery instead)</p> <p>A custom ID which can be used to link resources in the 6Connect database back to your organization.</p>
resourceId	INTEGER	1234	The ID of the resource the block is assigned to
resourceQuery	JSON	<pre>{ "custom_id": "cust-001" }</pre>	<p>A JSON object representing a valid resource query. Any parameters that can be used for a <a href="#">Resource GET API call</a> can be used. Use of the resourceQuery parameter will return blocks assigned to any of the resources returned by that query.</p>
rir	STRING	ARIN	Acceptable values: ARIN, RIPE, APNIC, AfriNIC, LACNIC, 1918
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
	tagsMode	STRING	"Strict" or "Exclude"	"strict" - matches ONLY blocks that have the EXACT set of tags of specified. "exclude" - matches ONLY blocks which are NOT tagged with any of the blocks specified.
	topAggregateId	INTEGER	1234	The ID of the aggregate block to which the block belongs
	type	STRING	"ipv4" or "ipv6"	IP type
	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	<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td><pre>{ "success":1, "message": "Block 192.168.0.0/24 (12345) added", "id":12345, "data":{ "id":12345, "cidr": "192.168.0.0/24", ...} }</pre></td></tr><tr><td>ERROR</td><td><pre>{ "success":0, "message": "error message" }</pre></td></tr></table>	SUCCESSFUL	<pre>{ "success":1, "message": "Block 192.168.0.0/24 (12345) added", "id":12345, "data":{ "id":12345, "cidr": "192.168.0.0/24", ...} }</pre>	ERROR	<pre>{ "success":0, "message": "error message" }</pre>								
SUCCESSFUL	<pre>{ "success":1, "message": "Block 192.168.0.0/24 (12345) added", "id":12345, "data":{ "id":12345, "cidr": "192.168.0.0/24", ...} }</pre>												
ERROR	<pre>{ "success":0, "message": "error message" }</pre>												
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>block</td><td>STRING</td><td>213.37.29.0/24</td><td>CIDR block description</td></tr><tr><td>rir</td><td>STRING</td><td>ARIN</td><td>Acceptable values: ARIN, RIPE, APNIC, AfriNIC, LACNIC, 1918</td></tr></table>	Name	Type	Example	Description	block	STRING	213.37.29.0/24	CIDR block description	rir	STRING	ARIN	Acceptable values: ARIN, RIPE, APNIC, AfriNIC, LACNIC, 1918
Name	Type	Example	Description										
block	STRING	213.37.29.0/24	CIDR block description										
rir	STRING	ARIN	Acceptable values: ARIN, RIPE, APNIC, AfriNIC, LACNIC, 1918										

## Optional Parameters

Name	Type	Example	Description
allowDuplicate	BOOL	true	Allow the creation of duplicate blocks. The default behavior is to reject duplicates.
allowSubAssignment	BOOL	true	Does the block allow sub-assignments? If the block is assigned and allowSubAssignment 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	Code X	User-defined block code as defined in Admin-IPAM settings: Generic Code Per Block Name
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&block=213.37.29.0/24
&rir=ARIN
```

**Update**



URL	/api/v1/api.php?target=ipam&action=update																			
Description	Updates detail data about an IP block.																			
Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td>SINGLE BLOCK</td><td colspan="3"><pre>{ "success":1, "message": "192.168.0.0/24 (12345) updated", "data":{ "id":12345, "cidr":192.168.0.0/24", ...} }</pre></td></tr><tr><td>SUCCESSFUL</td><td>MULTIPLE BLOCKS</td><td colspan="3"><pre>{ "success":1, "message": "blocks updated", "data":[ { "id":12345, "cidr":192.168.0.0/24", ...}, { "id":12346, "cidr": "192.168.0.1/32", ...} ] }</pre></td></tr><tr><td>ERROR</td><td></td><td colspan="3"><pre>{ "success":0, "message": "error message" }</pre></td></tr></table>					SUCCESSFUL	SINGLE BLOCK	<pre>{ "success":1, "message": "192.168.0.0/24 (12345) updated", "data":{ "id":12345, "cidr":192.168.0.0/24", ...} }</pre>			SUCCESSFUL	MULTIPLE BLOCKS	<pre>{ "success":1, "message": "blocks updated", "data":[ { "id":12345, "cidr":192.168.0.0/24", ...}, { "id":12346, "cidr": "192.168.0.1/32", ...} ] }</pre>			ERROR		<pre>{ "success":0, "message": "error message" }</pre>		
SUCCESSFUL	SINGLE BLOCK	<pre>{ "success":1, "message": "192.168.0.0/24 (12345) updated", "data":{ "id":12345, "cidr":192.168.0.0/24", ...} }</pre>																		
SUCCESSFUL	MULTIPLE BLOCKS	<pre>{ "success":1, "message": "blocks updated", "data":[ { "id":12345, "cidr":192.168.0.0/24", ...}, { "id":12346, "cidr": "192.168.0.1/32", ...} ] }</pre>																		
ERROR		<pre>{ "success":0, "message": "error message" }</pre>																		
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Allow Multiple</th><th>Description</th></tr><tr><td>id*</td><td>INTEGER</td><td>125</td><td>Yes</td><td>ID of the IP block. Multiple block IDs can be specified in a comma-separated list.</td></tr><tr><td>block*</td><td>STRING</td><td>192.0.0.0/24</td><td>Yes</td><td>CIDR or the block. Multiple CIDRs can be specified in a comma-separated list.</td></tr></table> <div>*Either block or id can be used, but only one must be provided</div>					Name	Type	Example	Allow Multiple	Description	id*	INTEGER	125	Yes	ID of the IP block. Multiple block IDs can be specified in a comma-separated list.	block*	STRING	192.0.0.0/24	Yes	CIDR or the block. Multiple CIDRs can be specified in a comma-separated list.
Name	Type	Example	Allow Multiple	Description																
id*	INTEGER	125	Yes	ID of the IP block. Multiple block IDs can be specified in a comma-separated list.																
block*	STRING	192.0.0.0/24	Yes	CIDR or the block. Multiple CIDRs can be specified in a comma-separated list.																
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr></table>					Name	Type	Example	Description											
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	Code X	Arbitrary user-defined block code
lirId	INTEGER	101	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.
propagate	BOOL	true	Propagates all attribute values to any smaller child blocks of the block being updated. <b>Available in version 5.1.0</b>
rir	STRING	ARIN	Acceptable values: ARIN, RIPE, APNIC, AfriNIC, LACNIC, 1918
tags	STRING	Customer, vpn	Comma-separated list of tags

	tags_action	STRING	replace	What action to take on the supplied tags. This action must be taken in conjunction with the tags parameter. Valid settings for tags_action are: replace, add, delete. When tags_action is set to 'replace', all tags on an IP block are replaced with those
	vlan	NUMERIC	123	VLAN for the block
Example URL	/api/v1/api.php?target=ipam&action=update&block=192.0.0.0/24 &notes=Notes_here			

Delete															
URL	/api/v1/api.php?target=ipam&action=delete														
Description	Deletes an aggregate block														
Returns	<div>Examples</div> <table><tr><td>SUCCESSFUL</td><td colspan="3"><pre>{ "success":1, "message": "Aggregate deleted: 192.168.0.0/24", "data": { "id":12345, "cidr": "192.168.0.0/24", ... } }</pre></td></tr><tr><td>ERROR</td><td colspan="3"><pre>{ "success":0, "message": "error message" }</pre></td></tr></table>			SUCCESSFUL	<pre>{ "success":1, "message": "Aggregate deleted: 192.168.0.0/24", "data": { "id":12345, "cidr": "192.168.0.0/24", ... } }</pre>			ERROR	<pre>{ "success":0, "message": "error message" }</pre>						
SUCCESSFUL	<pre>{ "success":1, "message": "Aggregate deleted: 192.168.0.0/24", "data": { "id":12345, "cidr": "192.168.0.0/24", ... } }</pre>														
ERROR	<pre>{ "success":0, "message": "error message" }</pre>														
Required Parameters	<table><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>INTEGER</td><td>125</td><td>ID of the IP block</td></tr></tbody></table> <div>*Either block or id can be used, but only one must be provided</div>			Name	Type	Example	Description	block*	STRING	213.37.29.0/24	CIDR block description	id*	INTEGER	125	ID of the IP block
Name	Type	Example	Description												
block*	STRING	213.37.29.0/24	CIDR block description												
id*	INTEGER	125	ID of the IP block												

Optional Parameters	Name	Type	Example	Description
	force	BOOL	true	Forces the aggregate block to be deleted even if the block is split or contains sub blocks which are assigned. The default behavior is to reject attempts to delete blocks which have been split or are assigned.
Example URL	/api/v1/api.php?target=ipam&action=delete&block=213.37.29.0/24			

Add Tag												
URL	/api/v1/api.php?target=ipam&action=add											
Description	Adds a tag to an IP block.											
Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td colspan="3">{ "success":1,"message":"Tag Added.", "data":{"id":12345, "cidr":192.168.0.0/24", ...} }</td></tr><tr><td>ERROR</td><td colspan="3">{ "success":0, "message":"error message" }</td></tr></table>				SUCCESSFUL	{ "success":1,"message":"Tag Added.", "data":{"id":12345, "cidr":192.168.0.0/24", ...} }			ERROR	{ "success":0, "message":"error message" }		
SUCCESSFUL	{ "success":1,"message":"Tag Added.", "data":{"id":12345, "cidr":192.168.0.0/24", ...} }											
ERROR	{ "success":0, "message":"error message" }											
Required Parameters												
	Name	Type	Example	Description								
	id*	INTEGER	125	ID of the block								
	block*	STRING	192.0.0.0/24	CIDR of the block								
	*Either block or id can be used, but only one must be provided											
	tag	STRING	Customer	The tag to add								
Optional Parameters	None											
Example URL	/api/v1/api.php?target=ipam&action=addTag&id=125&tag=Customer											

Delete Tag	
URL	/api/v1/api.php?target=ipam&action=deleteTag

Description	Removes a tag from an IP block.																							
Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td colspan="3">{ "success":1, "message":"Tag Removed.", "data":{"id":12345, "cidr":192.168.0.0/24", ...} }</td></tr><tr><td>ERROR</td><td colspan="3">{ "success":0, "message":"error message" }</td></tr></table>				SUCCESSFUL	{ "success":1, "message":"Tag Removed.", "data":{"id":12345, "cidr":192.168.0.0/24", ...} }			ERROR	{ "success":0, "message":"error message" }														
SUCCESSFUL	{ "success":1, "message":"Tag Removed.", "data":{"id":12345, "cidr":192.168.0.0/24", ...} }																							
ERROR	{ "success":0, "message":"error message" }																							
Required Parameters	<table><thead><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr></thead><tbody><tr><td>id*</td><td>INTEGER</td><td>125</td><td>ID of the block</td></tr><tr><td>block*</td><td>STRING</td><td>192.0.0.0/24</td><td>CIDR of the block</td></tr><tr><td colspan="4">*Either block or id can be used, but only one must be provided</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*	INTEGER	125	ID of the block	block*	STRING	192.0.0.0/24	CIDR of the block	*Either block or id can be used, but only one must be provided				tag	STRING	Customer	The tag to delete
Name	Type	Example	Description																					
id*	INTEGER	125	ID of the block																					
block*	STRING	192.0.0.0/24	CIDR of the block																					
*Either block or id can be used, but only one must be provided																								
tag	STRING	Customer	The tag to delete																					
Optional Parameters	None																							
Example URL	/api/v1/api.php?target=ipam&action=deleteTag&id=125&tag=Customer																							

### 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	<b>Examples:</b> <table> <tr> <td>SUCCESSFUL</td><td><code>{ "success":1, "message":"Assigned 192.168.0.0/24 to Resource (1234) via Smart Assign", "id":12345, "data":{"id":12345, "cidr":192.168.0.0/24", ...} }</code></td></tr> <tr> <td>ERROR</td><td><code>{ "success":0, "message":"error message" }</code></td></tr> </table>	SUCCESSFUL	<code>{ "success":1, "message":"Assigned 192.168.0.0/24 to Resource (1234) via Smart Assign", "id":12345, "data":{"id":12345, "cidr":192.168.0.0/24", ...} }</code>	ERROR	<code>{ "success":0, "message":"error message" }</code>
SUCCESSFUL	<code>{ "success":1, "message":"Assigned 192.168.0.0/24 to Resource (1234) via Smart Assign", "id":12345, "data":{"id":12345, "cidr":192.168.0.0/24", ...} }</code>				
ERROR	<code>{ "success":0, "message":"error message" }</code>				

## Required Parameters

Name	Type	Example	Description
mask	INTEGER	24	The size of the block to be assigned
rir	STRING	ARIN	Acceptable values: ARIN, RIPE, APNIC, AfriNIC, LACNIC, 1918
resourceId*	INTEGER	1234	Integer ID of the resource to assign the block to
resourceQuery*	JSON	<pre>{"custom_id": "cust-001"}</pre>	A JSON object representing a valid resource query. Any parameters that can be used for a Resource GET API call can be used. Use of the resourceQuery parameter will return blocks assigned to any of the resources returned by that query.
*Either resourceId or resourceQuery can be used, but only one must be provided			
resourceHolderId	STRING	cust-001	<b>(Deprecated:</b> Use resourceQuery instead)  A custom ID which can be used to link resources in the 6Connect database back to your organization.
type	STRING	"IPv4" or "IPv6"	The type of block to assign

## Optional Parameters\*

Name	Type	Example	Description
assignedResourceId	INTEGER	123	The ID of the resource the block is assigned to
code	STRING	Code X	Arbitrary user-defined block code
lirId	INTEGER	101	The ID of an LIR resource
region	STRING	Ashburn	Region to assign from
tags	STRING	customer,vpn	Comma separated string of tags. Matches blocks which have at least the set of tag specified by this parameter
tagsMode	STRING	"strict" or "exclude"	"strict" - matches ONLY blocks that have the EXACT set of tags of specified.  "exclude" - matches ONLY blocks which are NOT tagged with any of the blocks specified.
vlan	INTEGER	1023	VLAN designated to a given block

\*Additional or fewer "optional" parameters may be required in order to result in a successful assignment, depending on the attributes of available blocks.

Example URL

/api/v1/api.php?target=ipam&amp;action=smartAssign&amp;mask=24&amp;type=IPv4

**Direct Assign**

URL

/api/v1/api.php?target=ipam&amp;action=directAssign

Description

Assigns a block to an Resource Holder

Returns

Examples:

SUCCESSFUL	SINGLE BLOCK	<pre>{   "success":1,"message":     192.168.0.0/24 to     Resource (1234)",   "id":12345, "data":{     "id":12345,     "cidr":192.168.0.0/24",     ...} }</pre>
SUCCESSFUL	MULTIPLE BLOCKS	<pre>{   "success":1,"message":     5 blocks to Resource     (1234) via Direct     Assign", "data":{     "ids":[12345, 12346,     12347, ...] } }</pre>
ERROR		<pre>{ "success":0,   "message":"error message" }</pre>



## Required Parameters

Name	Type	Example	Description
block*	STRING	213.37.29.0/24	CIDR block description
id*	INTEGER	125	ID of the IP block, comma separated list of ids, or json encoded array of ids
*Either block or id can be used, but only one must be provided			
resourceHolderId*	STRING	cust-001	<b>(Deprecated:</b> Use resourceQuery instead)  A custom ID which can be used to link resources in the 6Connect database back to your organization.
resourceId**	INTEGER	1234	Integer ID of the resource to assign the block to
resourceQuery**	JSON	<pre>{"custom_id": "cust-001"}</pre>	A JSON object representing a valid resource query. Any parameters that can be used for a Resource GET API call can be used. Use of the resourceQuery parameter will return blocks assigned to any of the resources returned by that query.
**Either resourceId, resourceQuery, or resourceHolderId can be used, but only one must be provided			

## Optional Parameters\*

Name	Type	Example	Description
code	STRING	Code X	Arbitrary user-defined block code
lirId	INTEGER	101	The ID of an LIR resource
region	STRING	Ashburn	Region to assign from
rir	STRING	ARIN	Acceptable values: ARIN, RIPE, APNIC, AfriNIC, LACNIC, 1918
tags	STRING	customer,vpn	Comma separated string of tags. Matches blocks which have at least the set of tag specified by this parameter
tagsMode	STRING	"strict" or "exclude"	"strict" - matches ONLY blocks that have the EXACT set of tags of specified.  "exclude" - matches ONLY blocks which are NOT tagged with any of the blocks specified.
vlan	INTEGER	1023	VLAN designated to a given block

\*Additional or fewer "optional" parameters may be required in order to result in a successful assignment, depending on the attributes of available blocks.

## Example URL

/api/v1/api.php?target=ipam&action=directAssign&block=213.37.29.0/24

**Unassign**

## URL

/api/v1/api.php?target=ipam&action=unassign

## Description

Reclaims the specified block to be reassigned in the future

Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td><pre>{   "success":1,"message":"192.168.0.unassigned", "id":12345,   "data":{ "id":12345,             "cidr":192.168.0.0/24", ...} }</pre></td></tr><tr><td>ERROR</td><td><pre>{ "success":0, "message":"error message" }</pre></td></tr></table>	SUCCESSFUL	<pre>{   "success":1,"message":"192.168.0.unassigned", "id":12345,   "data":{ "id":12345,             "cidr":192.168.0.0/24", ...} }</pre>	ERROR	<pre>{ "success":0, "message":"error message" }</pre>								
SUCCESSFUL	<pre>{   "success":1,"message":"192.168.0.unassigned", "id":12345,   "data":{ "id":12345,             "cidr":192.168.0.0/24", ...} }</pre>												
ERROR	<pre>{ "success":0, "message":"error message" }</pre>												
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><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>INTEGER</td><td>125</td><td>ID of the IP block</td></tr></table> <div>*Either block or id can be used, but only one must be provided</div>	Name	Type	Example	Description	block*	STRING	213.37.29.0/24	CIDR block description	id*	INTEGER	125	ID of the IP block
Name	Type	Example	Description										
block*	STRING	213.37.29.0/24	CIDR block description										
id*	INTEGER	125	ID of the IP block										
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>skipHolding</td><td>BOOL</td><td>true</td><td><div>If set to true (skipHolding=true) then the holding tank is skipped. If set to false, or not included, normal holding tank rules apply. <b>Available in version 5.1.0</b>  Acceptable values: "true" or "false"</div></td></tr></table>	Name	Type	Example	Description	skipHolding	BOOL	true	<div>If set to true (skipHolding=true) then the holding tank is skipped. If set to false, or not included, normal holding tank rules apply. <b>Available in version 5.1.0</b>  Acceptable values: "true" or "false"</div>				
Name	Type	Example	Description										
skipHolding	BOOL	true	<div>If set to true (skipHolding=true) then the holding tank is skipped. If set to false, or not included, normal holding tank rules apply. <b>Available in version 5.1.0</b>  Acceptable values: "true" or "false"</div>										
Example URL	/api/v1/api.php?target=ipam&action=unassign&block=213.37.29.0/24												

### *Simple Reassign*

URL	/api/v1/api.php?target=ipam&action=simpleReassign
Description	ARIN SWIP - simple reassign. Creates an ARIN customer record for the assigned resource and reassigns the block to the ARIN customer record.

Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td><code>{ "success":1, "message":"Sent ARIN SWIP with action simpleReassign for 67.221.244.0/28 for Acme, Message: Success" }</code></td></tr><tr><td>ERROR</td><td><code>{ "success":0, "message":"error message" }</code></td></tr></table>	SUCCESSFUL	<code>{ "success":1, "message":"Sent ARIN SWIP with action simpleReassign for 67.221.244.0/28 for Acme, Message: Success" }</code>	ERROR	<code>{ "success":0, "message":"error message" }</code>																
SUCCESSFUL	<code>{ "success":1, "message":"Sent ARIN SWIP with action simpleReassign for 67.221.244.0/28 for Acme, Message: Success" }</code>																				
ERROR	<code>{ "success":0, "message":"error message" }</code>																				
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>blockId</td><td>INTEGER</td><td>1234</td><td>ID of the block to reassing</td></tr><tr><td>resourceId</td><td>INTEGER</td><td>1234</td><td>ID of resource representing the customer to reassign to</td></tr><tr><td>lirId</td><td>INTEGER</td><td>1234</td><td>ID of the LIR to use for reassignment</td></tr><tr><td>entityHandle</td><td>STRING</td><td>CONNE-81</td><td>The Org ID for the LIR.</td></tr></table>	Name	Type	Example	Description	blockId	INTEGER	1234	ID of the block to reassing	resourceId	INTEGER	1234	ID of resource representing the customer to reassign to	lirId	INTEGER	1234	ID of the LIR to use for reassignment	entityHandle	STRING	CONNE-81	The Org ID for the LIR.
Name	Type	Example	Description																		
blockId	INTEGER	1234	ID of the block to reassing																		
resourceId	INTEGER	1234	ID of resource representing the customer to reassign to																		
lirId	INTEGER	1234	ID of the LIR to use for reassignment																		
entityHandle	STRING	CONNE-81	The Org ID for the LIR.																		
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>netName</td><td>STRING</td><td>NET-ACME-67-221-244-0-28</td><td>Optional name for the network to override the default. The default net name will be created using the Net Name Prefix and IP address for the block.</td></tr></table>	Name	Type	Example	Description	netName	STRING	NET-ACME-67-221-244-0-28	Optional name for the network to override the default. The default net name will be created using the Net Name Prefix and IP address for the block.												
Name	Type	Example	Description																		
netName	STRING	NET-ACME-67-221-244-0-28	Optional name for the network to override the default. The default net name will be created using the Net Name Prefix and IP address for the block.																		
Example URL	/api/v1/api.php?target=ipam&action=simpleReassign&resourceId=121&																				

### Get Tags List

URL	/api/v1/api.php?target=ipam&action=getTagList
Description	Returns a list of all valid IP Tags in the database.

Returns	<div><b>Examples:</b></div> <table><tr><td>SUCCESSFUL</td><td><pre>{   "success": 1,   "message": "Tags Retrieved.",   "data": {     {       "value": "IT",       "name": "Mobile"     },     {       "value": "PTP",       "name": "Point to Point"     },     {       "value": "Prod",       "name": "Production Machines"     },     {       "value": "VOIP",       "name": "Voice over IP"     }   } }</pre></td></tr><tr><td>ERROR</td><td><pre>{   'success': 0,   'message': 'error message' }</pre></td></tr></table>	SUCCESSFUL	<pre>{   "success": 1,   "message": "Tags Retrieved.",   "data": {     {       "value": "IT",       "name": "Mobile"     },     {       "value": "PTP",       "name": "Point to Point"     },     {       "value": "Prod",       "name": "Production Machines"     },     {       "value": "VOIP",       "name": "Voice over IP"     }   } }</pre>	ERROR	<pre>{   'success': 0,   'message': 'error message' }</pre>
SUCCESSFUL	<pre>{   "success": 1,   "message": "Tags Retrieved.",   "data": {     {       "value": "IT",       "name": "Mobile"     },     {       "value": "PTP",       "name": "Point to Point"     },     {       "value": "Prod",       "name": "Production Machines"     },     {       "value": "VOIP",       "name": "Voice over IP"     }   } }</pre>				
ERROR	<pre>{   'success': 0,   'message': 'error message' }</pre>				

### Examples:

SUCCESSFUL	<pre>{   "success":1,"message":"Tags Retrieved.",   "data":{     {"value":"IT","name":"Mobile"},     {"value":"PTP","name":"Point to Point"},     {"value":"Prod","name":"Production Machines"},     {"value":"VOIP","name":"Voice over IP"}   } }</pre>
ERROR	<pre>{   'success':0,   'message':'error message' }</pre>

### Add Tag To List

URL	/api/v1/api.php?target=ipam&action=addTagToList											
Description	Adds a tag to the IPAM tag list											
Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td colspan="3">{<i>"success":1,"message":"Tag Added."</i>}</td></tr><tr><td>ERROR</td><td colspan="3">{<i>'success':0, 'message':'error message'</i>}</td></tr></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	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>newTag</td><td>STRING</td><td>Loopback C</td><td>The value to add to the list of name/value pairs which make up the list of available regions</td></tr></table>				Name	Type	Example	Description	newTag	STRING	Loopback C	The value to add to the list of name/value pairs which make up the list of available regions
Name	Type	Example	Description									
newTag	STRING	Loopback C	The value to add to the list of name/value pairs which make up the list of available regions									
Optional Parameters	None											
Example URL	/api/v1/api.php?target=ipam&action=addTagToList&newTag=Loopback C											

```
/api/v1/api.php?target=ipam&action=addTagToList
```

### Description

Adds a tag to the IPAM tag list

## Returns

### Examples:

SUCCESSFUL	<code>{"success":1,"message":"Tag Added."}</code>
ERROR	<code>{'success':0, 'message':'error message'}</code>

### Required Parameters

Name	Type	Example	Description
newTag	STRING	Loopback C	The value to add to the list of name/value pairs which make up the list of available regions

## Optional Parameters

None

### Example URL

```
/api/v1/api.php?target=ipam&action=addTagToList&newTag=Loopback
```

### Get RIRs List

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><i>{ "success":1, "message":"RIRs Retrieved. ", "data":{"value":"ARIN",</i></td></tr> <tr> <td>ERROR</td><td><i>{ 'success':0, 'message':'error message' }</i></td></tr> </table>	SUCCESSFUL	<i>{ "success":1, "message":"RIRs Retrieved. ", "data":{"value":"ARIN",</i>	ERROR	<i>{ 'success':0, 'message':'error message' }</i>
SUCCESSFUL	<i>{ "success":1, "message":"RIRs Retrieved. ", "data":{"value":"ARIN",</i>				
ERROR	<i>{ 'success':0, 'message':'error message' }</i>				

```
/api/v1/api.php?target=ipam&action=getRIRList
```

### Description

Returns a list of all valid RIRs in the database.

## Returns

### Examples:

SUCCESSFUL	<code>{"success":1,"message":"RIRs Retrieved.", "data":{"value":"ARIN",</code>
ERROR	<code>{'success':0, 'message': 'error message'}</code>

### Get Regions List

URL	/api/v1/api.php?target=ipam&action=getRegionList				
Description	Returns a list of all valid Regions in the database.				
Returns	<b>Examples:</b> <table><tr><td>SUCCESSFUL</td><td><pre>{"success":1,"message":"Regions Retrieved.", "data":{"value":"ANY", "Region":{"value":"ASH1", "name":"ASH1"}, {"value":"VA", "name":"Boston"}, {"value":"BOS", "name":"Boston"}, {"value":"CHI", "name":"Chicago"}, {"value":"DAL", "name":"Dallas"}, {"value":"DEN", "name":"Denver"}, {"value":"FRKT", "name":"Frankfurt"}, {"value":"LON1", "name":"London"}, {"value":"MIA", "name":"Miami"}, {"value":"PAR", "name":"Paris"}, {"value":"SFO", "name":"San Francisco"}, {"value":"SEA", "name":"Seattle"}, {"value":"Tokyo", "name":"Tokyo"}}</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", "Region":{"value":"ASH1", "name":"ASH1"}, {"value":"VA", "name":"Boston"}, {"value":"BOS", "name":"Boston"}, {"value":"CHI", "name":"Chicago"}, {"value":"DAL", "name":"Dallas"}, {"value":"DEN", "name":"Denver"}, {"value":"FRKT", "name":"Frankfurt"}, {"value":"LON1", "name":"London"}, {"value":"MIA", "name":"Miami"}, {"value":"PAR", "name":"Paris"}, {"value":"SFO", "name":"San Francisco"}, {"value":"SEA", "name":"Seattle"}, {"value":"Tokyo", "name":"Tokyo"}}</pre>	ERROR	<pre>{'success':0, 'message':'error message'}</pre>
SUCCESSFUL	<pre>{"success":1,"message":"Regions Retrieved.", "data":{"value":"ANY", "Region":{"value":"ASH1", "name":"ASH1"}, {"value":"VA", "name":"Boston"}, {"value":"BOS", "name":"Boston"}, {"value":"CHI", "name":"Chicago"}, {"value":"DAL", "name":"Dallas"}, {"value":"DEN", "name":"Denver"}, {"value":"FRKT", "name":"Frankfurt"}, {"value":"LON1", "name":"London"}, {"value":"MIA", "name":"Miami"}, {"value":"PAR", "name":"Paris"}, {"value":"SFO", "name":"San Francisco"}, {"value":"SEA", "name":"Seattle"}, {"value":"Tokyo", "name":"Tokyo"}}</pre>				
ERROR	<pre>{'success':0, 'message':'error message'}</pre>				

### Add Region To List

URL	/api/v1/api.php?target=ipam&action=addRegionToList											
Description	Adds a region to the IPAM region list											
Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td colspan="3">{<i>"success":1,"message":"Region Added."</i>}</td></tr><tr><td>ERROR</td><td colspan="3">{<i>'success':0, 'message':'error message'</i>}</td></tr></table>				SUCCESSFUL	{ <i>"success":1,"message":"Region Added."</i> }			ERROR	{ <i>'success':0, 'message':'error message'</i> }		
SUCCESSFUL	{ <i>"success":1,"message":"Region Added."</i> }											
ERROR	{ <i>'success':0, 'message':'error message'</i> }											
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>newRegion</td><td>STRING</td><td>SFO</td><td>The value to add to the list of name/value pairs which make up the list of available regions</td></tr></table>				Name	Type	Example	Description	newRegion	STRING	SFO	The value to add to the list of name/value pairs which make up the list of available regions
Name	Type	Example	Description									
newRegion	STRING	SFO	The value to add to the list of name/value pairs which make up the list of available regions									
Optional Parameters	None											
Example URL	/api/v1/api.php?target=ipam&action=addRegionToList&newRegion=SFO											

<i>Get Utilization</i>	
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.

Returns

Examples:

SUCCESSFUL	<pre>{   "success": 1,   "totalBlocks": 1,   "totalHosts": "256",   "hostsAssigned": 0,   "hostsAllocated": "256",   "hostsAvailable": "256",   "hostsInHolding": 0,   "availablePercentage": "100.00",   "assignedPercentage": "0.00",   "allocatedPercentage": "100.00",   "inHoldingPercentage": "0.00",   "resources": [{     "id": 351,     "name": "Customer 1",     "type": "entry",     "hosts": "256",     "blocks": "1",     "percentage": "100.00"   }],   "blocksAssigned": 0,   "blocksAllocated": 1,   "blocksAvailable": "1",   "blocksInHolding": null,   "blocksAssignedPercentage": "0.00",   "blocksAllocatedPercentage": "100.00",   "blocksAvailablePercentage": "100.00",   "blocksInHoldingPercentage": "0.00" }</pre>
ERROR	<pre>{'success':0, 'message':'error message'}</pre>



Required Parameters				
	<b>Name</b>	<b>Type</b>	<b>Example</b>	<b>Description</b>
	block*	STRING	213.37.29.0/24	CIDR block description
	id*	INTEGER	125	ID of the IP block
	*Either block or id can be used, but only one must be provided			
Optional Parameters				
	<b>Name</b>	<b>Type</b>	<b>Example</b>	<b>Description</b>
	mask	INTEGER	24	The specific mask size to retrieve utilization for. If using this parameter, the id parameter should be the id of the aggregate.
Example URL	/api/v1/api.php?target=ipam&action=utilization&id=125			

<b><i>Get Host Utilization</i></b>	
URL	/api/v1/api.php?target=ipam&action=getHostUtilization
Description	Gets the host utilization statistics with support for filters.

Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td><pre>{    "success": 1,    "totalHosts": "256",    "hostsAssigned": 0,    "hostsAllocated": "256",    "hostsAvailable": "256",    "hostsInHolding": 0,    "availablePercentage": "100.00",   "assignedPercentage": "0.00",   "allocatedPercentage": "100.00",   "inHoldingPercentage": "0.00",    "resources": [{      "id": 351,     "name": "Customer 1",     "type": "entry",     "hosts": "256",     "blocks": "1",     "percentage": "100.00"   } ] }</pre></td></tr><tr><td>ERROR</td><td><pre>{'success':0, 'message':'error message'}</pre></td></tr></table>	SUCCESSFUL	<pre>{    "success": 1,    "totalHosts": "256",    "hostsAssigned": 0,    "hostsAllocated": "256",    "hostsAvailable": "256",    "hostsInHolding": 0,    "availablePercentage": "100.00",   "assignedPercentage": "0.00",   "allocatedPercentage": "100.00",   "inHoldingPercentage": "0.00",    "resources": [{      "id": 351,     "name": "Customer 1",     "type": "entry",     "hosts": "256",     "blocks": "1",     "percentage": "100.00"   } ] }</pre>	ERROR	<pre>{'success':0, 'message':'error message'}</pre>				
SUCCESSFUL	<pre>{    "success": 1,    "totalHosts": "256",    "hostsAssigned": 0,    "hostsAllocated": "256",    "hostsAvailable": "256",    "hostsInHolding": 0,    "availablePercentage": "100.00",   "assignedPercentage": "0.00",   "allocatedPercentage": "100.00",   "inHoldingPercentage": "0.00",    "resources": [{      "id": 351,     "name": "Customer 1",     "type": "entry",     "hosts": "256",     "blocks": "1",     "percentage": "100.00"   } ] }</pre>								
ERROR	<pre>{'success':0, 'message':'error message'}</pre>								
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>type</td><td>STRING</td><td>"ipv4" or "ipv6"</td><td>IP type</td></tr></table>	Name	Type	Example	Description	type	STRING	"ipv4" or "ipv6"	IP type
Name	Type	Example	Description						
type	STRING	"ipv4" or "ipv6"	IP type						

Optional Parameters	Name	Type	Example	Multiple Values	Description
	code	STRING	"code-1"	Yes	User-defined block code as defined in Admin-IPAM settings: Generic Code Per Block Name
	region	STRING	"SFO"	Yes	Region to assign from
	rir	STRING	ARIN	No	Acceptable values: ARIN, RIPE, APNIC, AfriNIC, LACNIC, 1918
	tags	STRING	"Customer"	Yes	Comma separated string of tags
	vlan	INTEGER	1000	Yes	VLAN designated to a given block
	<p>NOTE: to filter using multiple values, pass the values as a JSON-encoded string representation of an array.</p> <p>For example, to get utilization data for multiple tags, you could use the following URL:</p> <p><i>/api/v1/api.php?target=ipam&amp;action=getHostUtilization&amp;type=ipv4&amp;tags=</i></p>				
Example URL	<i>/api/v1/api.php?target=ipam&amp;action=getHostUtilization&amp;type=ipv4&amp;tags=["Customer","PTP"]&amp;region=SMF</i>				

<i>Aggregate</i>	
URL	<i>/api/v1/api.php?target=ipam&amp;action=aggregate</i>
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	<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td><code>{"success":1,"message":"10 aggregated into 10.2.0.0\24","id":16326}</code></td></tr><tr><td>ERROR</td><td><code>{'success':0,'message':'error message'}</code></td></tr></table>	SUCCESSFUL	<code>{"success":1,"message":"10 aggregated into 10.2.0.0\24","id":16326}</code>	ERROR	<code>{'success':0,'message':'error message'}</code>								
SUCCESSFUL	<code>{"success":1,"message":"10 aggregated into 10.2.0.0\24","id":16326}</code>												
ERROR	<code>{'success':0,'message':'error message'}</code>												
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>id*</td><td>INTEGER</td><td>125</td><td>ID of the IP block.</td></tr><tr><td>block*</td><td>STRING</td><td>213.37.29.0/24</td><td>CIDR block.</td></tr></table> <div>*Either block or id can be used, but only one must be provided</div>	Name	Type	Example	Description	id*	INTEGER	125	ID of the IP block.	block*	STRING	213.37.29.0/24	CIDR block.
Name	Type	Example	Description										
id*	INTEGER	125	ID of the IP block.										
block*	STRING	213.37.29.0/24	CIDR block.										
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>autoAggregateToMask</td><td>INTEGER</td><td>24</td><td>All blocks and IPs smaller than this netmask will be aggregated.</td></tr><tr><td>ignoreAssignment</td><td>BOOL</td><td>TRUE</td><td>If the ignoreAssignment flag is not set the aggregation operation will fail if any children beneath the supplied autoAggregateToMask are assigned or otherwise unavailable. If this option is set, it will unassign blocks prior to reaggregation.</td></tr></table>	Name	Type	Example	Description	autoAggregateToMask	INTEGER	24	All blocks and IPs smaller than this netmask will be aggregated.	ignoreAssignment	BOOL	TRUE	If the ignoreAssignment flag is not set the aggregation operation will fail if any children beneath the supplied autoAggregateToMask are assigned or otherwise unavailable. If this option is set, it will unassign blocks prior to reaggregation.
Name	Type	Example	Description										
autoAggregateToMask	INTEGER	24	All blocks and IPs smaller than this netmask will be aggregated.										
ignoreAssignment	BOOL	TRUE	If the ignoreAssignment flag is not set the aggregation operation will fail if any children beneath the supplied autoAggregateToMask are assigned or otherwise unavailable. If this option is set, it will unassign blocks prior to reaggregation.										
Example URL	/api/v1/api.php?target=ipam&action=aggregate&id=125&autoAggregateToMask=24												

### Split

URL	<code>/api/v1/api.php?target=ipam&amp;action=split</code>
-----	---

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	<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td><pre>{"success":1,"message":"10 split into 10.1.0.0\25 and 10.1.0.128\25","data":{"child1":23451,"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 split into 10.1.0.0\25 and 10.1.0.128\25","data":{"child1":23451,"child2":23451}}</pre>	ERROR	<pre>{'success':0, 'message':'error message'}</pre>								
SUCCESSFUL	<pre>{"success":1,"message":"10 split into 10.1.0.0\25 and 10.1.0.128\25","data":{"child1":23451,"child2":23451}}</pre>												
ERROR	<pre>{'success':0, 'message':'error message'}</pre>												
Required Parameters	<table><thead><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr></thead><tbody><tr><td>id*</td><td>INTEGER</td><td>125</td><td>ID of the IP block.</td></tr><tr><td>block*</td><td>STRING</td><td>213.37.29.0/24</td><td>CIDR block.</td></tr></tbody></table> <div>*Either block or id can be used, but only one must be provided</div>	Name	Type	Example	Description	id*	INTEGER	125	ID of the IP block.	block*	STRING	213.37.29.0/24	CIDR block.
Name	Type	Example	Description										
id*	INTEGER	125	ID of the IP block.										
block*	STRING	213.37.29.0/24	CIDR block.										
Optional Parameters	<table><thead><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr></thead><tbody><tr><td>autoSplitToMask</td><td>INTEGER</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>autoSplitLimit</td><td>INTEGER</td><td>4</td><td>A number the power of 2 (^2).</td></tr></tbody></table>	Name	Type	Example	Description	autoSplitToMask	INTEGER	24	Auto aggregate the block back to this mask size. Note all blocks up this mask size must be Available or call will fail.	autoSplitLimit	INTEGER	4	A number the power of 2 (^2).
Name	Type	Example	Description										
autoSplitToMask	INTEGER	24	Auto aggregate the block back to this mask size. Note all blocks up this mask size must be Available or call will fail.										
autoSplitLimit	INTEGER	4	A number the power of 2 (^2).										
Example URL	/api/v1/api.php?target=ipam&action=split&block=213.37.29.0/24&autoSplitLimit=4												

### Scan Block

URL	/api/v1/api.php?target=ipam&action=scanBlock
Description	Initiates an asynchronous ping (ICMP) scan of the target block specified. Results of the scan can be checked with get.

Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td><pre>{ "success":1, "message": "Pi scan started for 8.8.8.0\27" }</pre></td></tr><tr><td>ERROR</td><td><pre>{'success':0, 'message':'error message'}</pre></td></tr></table>	SUCCESSFUL	<pre>{ "success":1, "message": "Pi scan started for 8.8.8.0\27" }</pre>	ERROR	<pre>{'success':0, 'message':'error message'}</pre>												
SUCCESSFUL	<pre>{ "success":1, "message": "Pi scan started for 8.8.8.0\27" }</pre>																
ERROR	<pre>{'success':0, 'message':'error message'}</pre>																
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>id*</td><td>INTEGER</td><td>125</td><td>ID of the IP block.</td></tr><tr><td>block*</td><td>STRING</td><td>213.37.29.0/24</td><td>CIDR block.</td></tr><tr><td colspan="4">*Either block or id can be used, but only one must be provided</td></tr></table>	Name	Type	Example	Description	id*	INTEGER	125	ID of the IP block.	block*	STRING	213.37.29.0/24	CIDR block.	*Either block or id can be used, but only one must be provided			
Name	Type	Example	Description														
id*	INTEGER	125	ID of the IP block.														
block*	STRING	213.37.29.0/24	CIDR block.														
*Either block or id can be used, but only one must be provided																	
Optional Parameters	None																
Example	/api/v1/api.php?target=ipam&action=scanBlock&block=213.37.29.0/24																

### Get Scan Results

URL	/api/v1/api.php?target=ipam&action=getScanResults											
Description	Initiates an asynchronous ping (ICMP) scan of the target block specified. Results of the scan can be checked with get											
Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td colspan="3">{ "success":1, "data":{ "block": "213.37.29.0/24", "start": "2011-07-10T00:00:00", "end": "2011-07-11T00:00:00", "data": [ { "address": "8.8.8.8" } ] }</td></tr><tr><td>ERROR</td><td colspan="3">{ 'success':0, 'message':'error message' }</td></tr></table>				SUCCESSFUL	{ "success":1, "data":{ "block": "213.37.29.0/24", "start": "2011-07-10T00:00:00", "end": "2011-07-11T00:00:00", "data": [ { "address": "8.8.8.8" } ] }			ERROR	{ 'success':0, 'message':'error message' }		
SUCCESSFUL	{ "success":1, "data":{ "block": "213.37.29.0/24", "start": "2011-07-10T00:00:00", "end": "2011-07-11T00:00:00", "data": [ { "address": "8.8.8.8" } ] }											
ERROR	{ 'success':0, 'message':'error message' }											
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>block</td><td>STRING</td><td>213.37.29.0/24</td><td>CIDR block.</td></tr></table>				Name	Type	Example	Description	block	STRING	213.37.29.0/24	CIDR block.
Name	Type	Example	Description									
block	STRING	213.37.29.0/24	CIDR block.									
Optional Parameters	None											
Example	/api/v1/api.php?target=ipam&action=getScanResults&block=213.37.29.0/24											

### Get Options

URL	/api/v1/api.php?target=ipam&action=getOptions
Description	Returns a list of options available for the block

Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td><code>{"success":1,"message":"Options for 14.0.0.0V25 (125)","options":{"actions":["aggregate","Split"],"masks":[26,27,28,29,30,31,32]}}</code></td></tr><tr><td>ERROR</td><td><code>{'success':0, 'message':'error message'}</code></td></tr></table>	SUCCESSFUL	<code>{"success":1,"message":"Options for 14.0.0.0V25 (125)","options":{"actions":["aggregate","Split"],"masks":[26,27,28,29,30,31,32]}}</code>	ERROR	<code>{'success':0, 'message':'error message'}</code>				
SUCCESSFUL	<code>{"success":1,"message":"Options for 14.0.0.0V25 (125)","options":{"actions":["aggregate","Split"],"masks":[26,27,28,29,30,31,32]}}</code>								
ERROR	<code>{'success':0, 'message':'error message'}</code>								
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>id</td><td>INTEGER</td><td>125</td><td>ID of the IP block</td></tr></table>	Name	Type	Example	Description	id	INTEGER	125	ID of the IP block
Name	Type	Example	Description						
id	INTEGER	125	ID of the IP block						
Optional Parameters	None								
Example URL	/api/v1/api.php?target=ipam&action=getOptions&id=125								

### Get Resource Hierarchy

URL	/api/v1/api.php?target=ipam&action=getResourceHierarchy											
Description	Returns the resource hierarchy for the block											
Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td colspan="3">{<span>"success":1,"data":{"id":402,"name":"Labz"}}</span>}</td></tr><tr><td>ERROR</td><td colspan="3">{<span>'success':0, 'message':'error message'}</span>}</td></tr></table>				SUCCESSFUL	{ <span>"success":1,"data":{"id":402,"name":"Labz"}}</span> }			ERROR	{ <span>'success':0, 'message':'error message'}</span> }		
SUCCESSFUL	{ <span>"success":1,"data":{"id":402,"name":"Labz"}}</span> }											
ERROR	{ <span>'success':0, 'message':'error message'}</span> }											
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>id</td><td>INTEGER</td><td>125</td><td>ID of the IP block</td></tr></table>				Name	Type	Example	Description	id	INTEGER	125	ID of the IP block
Name	Type	Example	Description									
id	INTEGER	125	ID of the IP block									
Optional Parameters	None											
Example URL	/api/v1/api.php?target=ipam&action=getResourceHierarchy&id=125											

### Get VLAN

URL	/api/v1/api.php?target=ipam&action=getVlan
Description	Returns the VLAN for the block

Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td><pre>{<i>"success":1,"message":</i>"Found VLAN 1002 (14.0.0.0V25)",<i>"data":{"id":125,"type":</i>Labz",<i>"last_updated_time":</i>"2015-01-12:30:37",<i>"description":</i>null,<i>"parent":</i>Notes",<i>"generic_code":</i>"Datacenter:GA",<i>"vlan":</i>1002,<i>"arin_net_id":</i>null,<i>"arin_net_id":</i>10:30:31",<i>"asn":</i>"143",<i>"allowSubAssig":</i>14.0.0.127",<i>"tags":{"Customer"}}</i>}</pre></td></tr><tr><td>ERROR</td><td><pre>{<i>'success':0, 'message':</i>'error message'}</pre></td></tr></table>	SUCCESSFUL	<pre>{<i>"success":1,"message":</i>"Found VLAN 1002 (14.0.0.0V25)",<i>"data":{"id":125,"type":</i>Labz",<i>"last_updated_time":</i>"2015-01-12:30:37",<i>"description":</i>null,<i>"parent":</i>Notes",<i>"generic_code":</i>"Datacenter:GA",<i>"vlan":</i>1002,<i>"arin_net_id":</i>null,<i>"arin_net_id":</i>10:30:31",<i>"asn":</i>"143",<i>"allowSubAssig":</i>14.0.0.127",<i>"tags":{"Customer"}}</i>}</pre>	ERROR	<pre>{<i>'success':0, 'message':</i>'error message'}</pre>												
SUCCESSFUL	<pre>{<i>"success":1,"message":</i>"Found VLAN 1002 (14.0.0.0V25)",<i>"data":{"id":125,"type":</i>Labz",<i>"last_updated_time":</i>"2015-01-12:30:37",<i>"description":</i>null,<i>"parent":</i>Notes",<i>"generic_code":</i>"Datacenter:GA",<i>"vlan":</i>1002,<i>"arin_net_id":</i>null,<i>"arin_net_id":</i>10:30:31",<i>"asn":</i>"143",<i>"allowSubAssig":</i>14.0.0.127",<i>"tags":{"Customer"}}</i>}</pre>																
ERROR	<pre>{<i>'success':0, 'message':</i>'error message'}</pre>																
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>id*</td><td>INTEGER</td><td>125</td><td>ID of the IP block</td></tr><tr><td>block*</td><td>STRING</td><td>213.37.29.0/24</td><td>CIDR block.</td></tr><tr><td colspan="4">*Either block or id can be used, but only one must be provided</td></tr></table>	Name	Type	Example	Description	id*	INTEGER	125	ID of the IP block	block*	STRING	213.37.29.0/24	CIDR block.	*Either block or id can be used, but only one must be provided			
Name	Type	Example	Description														
id*	INTEGER	125	ID of the IP block														
block*	STRING	213.37.29.0/24	CIDR block.														
*Either block or id can be used, but only one must be provided																	
Optional Parameters	None																
Example URL	/api/v1/api.php?target=ipam&action=getVlan&id=125																

### Process Holding Tank

URL	/api/v1/api.php?target=ipam&action=processHoldingTank				
Description	Processes the Holding Tank, returning held blocks to available status				
Returns	<b>Examples:</b> <table> <tr> <td>SUCCESSFUL</td><td>{<i>"success":1,"message":</i>"1 IPv4 and 0 IPv6 blocks would be moved to the available pool.",<i>"data":{"id":77712,"type":</i>"ipv4",<i>"type":</i>holding",<i>"last_updated_time":</i>"2014-11:25:41",<i>"description":</i>null,<i>"parent":</i>AZ",<i>"vlan":</i>null,<i>"arin_net_id":</i>null,<i>"arin_net_id":</i>11:20:34",<i>"asn":</i>null,<i>"allowSubAssig":</i>23.92.0.127",<i>"tags":{"Customer",</i>"D</td></tr> <tr> <td>ERROR</td><td>{<i>'success':0, 'message':</i>'error message'}</td></tr> </table>	SUCCESSFUL	{ <i>"success":1,"message":</i> "1 IPv4 and 0 IPv6 blocks would be moved to the available pool.", <i>"data":{"id":77712,"type":</i> "ipv4", <i>"type":</i> holding", <i>"last_updated_time":</i> "2014-11:25:41", <i>"description":</i> null, <i>"parent":</i> AZ", <i>"vlan":</i> null, <i>"arin_net_id":</i> null, <i>"arin_net_id":</i> 11:20:34", <i>"asn":</i> null, <i>"allowSubAssig":</i> 23.92.0.127", <i>"tags":{"Customer",</i> "D	ERROR	{ <i>'success':0, 'message':</i> 'error message'}
SUCCESSFUL	{ <i>"success":1,"message":</i> "1 IPv4 and 0 IPv6 blocks would be moved to the available pool.", <i>"data":{"id":77712,"type":</i> "ipv4", <i>"type":</i> holding", <i>"last_updated_time":</i> "2014-11:25:41", <i>"description":</i> null, <i>"parent":</i> AZ", <i>"vlan":</i> null, <i>"arin_net_id":</i> null, <i>"arin_net_id":</i> 11:20:34", <i>"asn":</i> null, <i>"allowSubAssig":</i> 23.92.0.127", <i>"tags":{"Customer",</i> "D				
ERROR	{ <i>'success':0, 'message':</i> 'error message'}				
Required Parameters	None				



Optional Parameters				
	<b>Name</b>	<b>Type</b>	<b>Example</b>	<b>Description</b>
	preview	BOOL	true	Shows what is going to be removed from the holding tank.  Acceptable values: "true" or "false"
Example URL	/api/v1/api.php?target=ipam&action=processHoldingTank&preview=true			

### IPAM API Calls Subject to Change:

Calls below this point are subject to change, and are not recommended for use in production code.

Get Attribute List												
URL	/api/v1/api.php?target=ipam&action=getAttributeLists											
Description	Returns a list of attributes											
Returns	<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td colspan="3">{ "asns": [], "masks": [ "24" ], "rirs": [ "1911", "Lab 1", "slug": "quito-lab-1", "type": "dhcp" ] }</td></tr><tr><td>ERROR</td><td colspan="3">{ 'success': 0, 'message': 'error message' }</td></tr></table>				SUCCESSFUL	{ "asns": [], "masks": [ "24" ], "rirs": [ "1911", "Lab 1", "slug": "quito-lab-1", "type": "dhcp" ] }			ERROR	{ 'success': 0, 'message': 'error message' }		
SUCCESSFUL	{ "asns": [], "masks": [ "24" ], "rirs": [ "1911", "Lab 1", "slug": "quito-lab-1", "type": "dhcp" ] }											
ERROR	{ 'success': 0, 'message': 'error message' }											
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>id</td><td>INTEGER</td><td>125</td><td>ID of the IP block</td></tr></table>				Name	Type	Example	Description	id	INTEGER	125	ID of the IP block
Name	Type	Example	Description									
id	INTEGER	125	ID of the IP block									
Optional Parameters	None											
Example URL	/api/v1/api.php?target=ipam&action=getAttributeLists&id=125											

# API Module - LIR

- LIR Management
  - Get
  - Delete

## LIR Management

Get	
URL	/api/v1/api.php?target=lir&action=get
Description	Returns a list of LIRs
Returns	<div><div>Examples:</div><div><div>SUCCESSFUL</div><div><pre>{   "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",</pre></div></div></div>

```
"admin_poc":  
"TEST-ARIN",  
  
"net_poc": "TEST-ARIN",  
  
"abuse_poc": "",  
  
"net_name_prefix":  
"PRFX",  
  
"api_key":  
"API-XXXX-YYYY-ZZZZ-1234"  
}  
  
    ],  
    "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	<div><b>Examples:</b><table><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></div>	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 - Peering

- Peering
  - getCommunications
  - getPeers
  - getRequests
  - getSessions
  - addSession
  - configureSession
  - deleteSession
  - updateSession
  - resetPeerStatus
  - sendRequest
  - sendEmail
  - updatePeer

## Peering

getCommunications												
Base URL		/api/v1/api.php?target=peering&action=getCommunications										
Description		Returns all communication data on peers at a particular exchange.										
Returns		<div>Examples:</div> <table><tr><td>SUCCESSFUL</td><td colspan="3">{ "success":1, "message":"8 records found.", "data":[{"name":"1&amp;1 Internet", "asn":"8560", "request_status":"pending", "request_id":1}, {"name":"Level 3 Communications", "asn":"3356", "request_status":"pending", "request_id":2}, {"name":"Cogent Communications", "asn":"14061", "request_status":"pending", "request_id":3}, {"name":"Abovenet Communications Inc.", "asn":"6461", "request_status":"pending", "request_id":4}, {"name":"Telecom", "asn":"4323", "request_status":"pending", "request_id":5} ] }</td></tr><tr><td>ERROR</td><td colspan="3">{ 'success':0, 'message':'error message' }</td></tr></table>			SUCCESSFUL	{ "success":1, "message":"8 records found.", "data":[{"name":"1&1 Internet", "asn":"8560", "request_status":"pending", "request_id":1}, {"name":"Level 3 Communications", "asn":"3356", "request_status":"pending", "request_id":2}, {"name":"Cogent Communications", "asn":"14061", "request_status":"pending", "request_id":3}, {"name":"Abovenet Communications Inc.", "asn":"6461", "request_status":"pending", "request_id":4}, {"name":"Telecom", "asn":"4323", "request_status":"pending", "request_id":5} ] }			ERROR	{ 'success':0, 'message':'error message' }		
SUCCESSFUL	{ "success":1, "message":"8 records found.", "data":[{"name":"1&1 Internet", "asn":"8560", "request_status":"pending", "request_id":1}, {"name":"Level 3 Communications", "asn":"3356", "request_status":"pending", "request_id":2}, {"name":"Cogent Communications", "asn":"14061", "request_status":"pending", "request_id":3}, {"name":"Abovenet Communications Inc.", "asn":"6461", "request_status":"pending", "request_id":4}, {"name":"Telecom", "asn":"4323", "request_status":"pending", "request_id":5} ] }											
ERROR	{ 'success':0, 'message':'error message' }											
Required Parameters		<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>public_id</td><td>INTEGER</td><td>1</td><td>The unique numerical identifier of the exchange to retrieve peering communication records for.</td></tr></table>			Name	Type	Example	Description	public_id	INTEGER	1	The unique numerical identifier of the exchange to retrieve peering communication records for.
Name	Type	Example	Description									
public_id	INTEGER	1	The unique numerical identifier of the exchange to retrieve peering communication records for.									
Optional Parameters		None										
Example URL		/api/v1/api.php?target=peering&action=getCommunications&public_id=1										

**getPeers**

URL	/api/v1/api.php?target=peering&action=getPeers																																								
Description	Returns a list of all peers available at an exchange																																								
Returns	<p><b>Examples:</b></p> <p>SUCCESSFUL: {"success":1,"message":"184 peers found.", "data":{"id":"262", "public_id":"1", "asn":"8560", "name":"1 &amp; 1 Internet", "qualified":true, "is_peer":0, "request_status":"sent", "info_prefixe": "Clearing House", "qualified":true, "is_peer":0, "request_status":null, "info_prefixes":"1 Hosting", "qualified":true, "is_peer":0, "request_status":null, "info_prefixes": "Communications, Inc.", "qualified":true, "is_peer":0, "request_status":null, "info_prefixes": "20 Communications", "qualified":true, "is_peer":0, "request_status":null, "info_prefixes": "60 LLC", "qualified":true, "is_peer":0, "request_status":null, "info_prefixes": "60 Inc", "qualified":true, "is_peer":0, "request_status":null, "info_prefixes": "5", " (Abovenet Communications Inc.)", "qualified":true, "is_peer":0, "request_status":null, "info_prefixes": "20 Game Network, Inc.", "qualified":true, "is_peer":0, "request_status":null, "info_prefixes":null.</p> <p>ERROR: {"success":1, "message":"No peers found."}</p>																																								
Required Parameters	None																																								
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>public_id</td><td>INTEGER</td><td>1</td><td>The unique numerical identifier of the exchange to retrieve peering communication records for.</td></tr><tr><td>id</td><td>INT</td><td>1</td><td>The unique numerical identifier of the peer in peeringDB.</td></tr><tr><td>asn</td><td>INT</td><td>4436</td><td></td></tr><tr><td>name</td><td>STRING</td><td>GTT</td><td></td></tr><tr><td>aka</td><td>STRING</td><td>nLayer</td><td></td></tr><tr><td>website</td><td>STRING</td><td><a href="http://www.gt-t.net">http://www.gt-t.net</a></td><td></td></tr><tr><td>notes_public</td><td>STRING</td><td></td><td></td></tr><tr><td>notes_private</td><td>STRING</td><td></td><td></td></tr><tr><td>irr_as_set</td><td>STRING</td><td>AS-NLAYER</td><td></td></tr></table>	Name	Type	Example	Description	public_id	INTEGER	1	The unique numerical identifier of the exchange to retrieve peering communication records for.	id	INT	1	The unique numerical identifier of the peer in peeringDB.	asn	INT	4436		name	STRING	GTT		aka	STRING	nLayer		website	STRING	<a href="http://www.gt-t.net">http://www.gt-t.net</a>		notes_public	STRING			notes_private	STRING			irr_as_set	STRING	AS-NLAYER	
Name	Type	Example	Description																																						
public_id	INTEGER	1	The unique numerical identifier of the exchange to retrieve peering communication records for.																																						
id	INT	1	The unique numerical identifier of the peer in peeringDB.																																						
asn	INT	4436																																							
name	STRING	GTT																																							
aka	STRING	nLayer																																							
website	STRING	<a href="http://www.gt-t.net">http://www.gt-t.net</a>																																							
notes_public	STRING																																								
notes_private	STRING																																								
irr_as_set	STRING	AS-NLAYER																																							

info_traffic	ENUM	1 Tbps+	enum('Not Disclosed','0-20 Mbps','20-100Mbps','50-100 Gbps','100+ Gbps','100-200 Gbps','200-300 Gbps','300-500 Gbps','500-1000 Gbps','1 Tbps+') DEFAULT 'Not Disclosed'
info_ratio	ENUM	Mostly Outbound	enum('Not Disclosed','Heavy Outbound','Mostly Outbound','Balance Inbound','Heavy Inbound') DEFAULT 'Not Disclosed'
info_scope	ENUM	Global	enum('Not Disclosed','Region America','Asia Pacific','Europe','/ America','Global') DEFAULT NULL
info_type	ENUM	NSP	enum('Not Disclosed','NSP','') DEFAULT 'Not Disclosed'
info_prefixes	INT	10000	
info_lookingglass	STRING	<a href="http://lg.nlayer.net/">http://lg.nlayer.net/</a>	
info_routeserver	STRING	<a href="telnet://route-server.nlayer.net">telnet://route-server.nlayer.net</a>	
info_unicast	CHAR	1	
info_multicast	CHAR		
info_ipv6	CHAR	1	
policy_url	STRING	<a href="http://www.gt-t.net/Peering_policies">http://www.gt-t.net/Peering_policies</a>	
policy_general	ENUM	Selective	enum('Open','Selective') DEFAULT NULL
policy_locations	ENUM	Required - International	enum('Not Required','Preferred - US','Required - International') DEFAULT NULL

	policy_ratio	ENUM	No	enum('Yes','No') DEFAULT NULL
	policy_contracts	ENUM	Not Required	enum('Not Required','Private Only','Required') DEFAULT NULL
	policy_nopublic	ENUM	N	enum('Y','N') NOT NULL DEFAULT 'N'
	policy_noprivate	ENUM	N	enum('Y','N') NOT NULL DEFAULT 'N'
	date_created	DATETIME	2013-03-21 15:36:42	Date the peeringdb entry was created
	date_lastupdated	DATETIME	2013-03-21 15:36:42	Date the peeringdb entry was last updated
	include_public_ips	BOOL	TRUE	Returns a list of all public facing IPs
	include_contacts	BOOL	TRUE	Returns a list of all contacts associated with peer(s)
	include_log_data	BOOL	TRUE	Returns a list of all log data associated with the peer(s) (use with care)
Example URL	/api/v1/api.php?target=peering&action=getPeers&public_id=1			

<i>getRequests</i>	
URL	/api/v1/api.php?target=peering&action=getRequests
Description	Returns a list of all peering requests issued



Returns	<p><b>Examples:</b></p> <p>SUCCESSFUL: {"success":1,"message":"1 request found.", "data":{"id":"131","public_id":"5","source_participant_id":"2335", <a href="mailto:ops@6connect.com">ops@6connect.com</a>","email_to":"<a href="mailto:halinmk@gmail.com">halinmk@gmail.com</a>", ,"subject":"Peering request from 6connect, Inc.", "body":"Peering,\n\n6connect, Inc., 8038, would like to peer with Amazon.com at our common locations.\n\nFacility, IP Address\nEquinix Ashburn - 206.126.236.68\nEquinix Palo Alto - 198.32.176.36\nEquinix Ashburn - 206.126.236.35\nEquinix San Jose - 206.223.116.177\nLINX Juniper LAN - 195.66.225.175\n\nSincerely,\nOperations\nops@6connect.com\n\n6connect, Inc. information:\nEquinix Palo Alto, 2001:504:d::33\nEquinix Palo Alto, 198.32.176.51\n\nPeeringDB: http://as8038.peeringdb.com\n", "status":null, "created":"2014-04-23 10:31:33", "modified":"2014-04-23 10:31:33"}}}</p> <p>ERROR: {"success":1,"message":"No request found.", "data":{}}</p>								
Required Parameters	None								
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>peer_participant_id</td><td>INTEGER</td><td>1</td><td>The numerical id of the peer</td></tr></table>	Name	Type	Example	Description	peer_participant_id	INTEGER	1	The numerical id of the peer
Name	Type	Example	Description						
peer_participant_id	INTEGER	1	The numerical id of the peer						
Example URL	/api/v1/api.php?target=peering&action=getRequests&peer_participant_id=1								

getSessions											
URL	/api/v1/api.php?target=peering&action=getSessions										
Description	Returns a list of all bgp peering sessions										
Returns	<p><b>Examples:</b></p> <p>SUCCESSFUL: {"success":1,"message":"1 sessions found.", "data":{"id":"51","source_asn":"32787","source_ipaddr":"1.2.3.4 Technologies", "peer_participant_id":"2", "peer_ipaddr":"206.126.236.102 b", "public_id":"1", "public_name":"Equinix Ashburn", "ip_type":"ipv4", "type":"Peer", "state":"not configured", "prfx_max":"20", "prfx_received":null, "password":"0", "note":null}}</p> <p>ERROR: {"success":1,"message":"No peers found."}</p>										
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>public_id</td><td>INTEGER</td><td>1</td><td>The unique numerical identifier of the exchange to retrieve peering communicaiton records for.</td></tr></table>			Name	Type	Example	Description	public_id	INTEGER	1	The unique numerical identifier of the exchange to retrieve peering communicaiton records for.
Name	Type	Example	Description								
public_id	INTEGER	1	The unique numerical identifier of the exchange to retrieve peering communicaiton records for.								

## Optional Parameters

Name	Type	Example	Description
id	INTEGER	41	
public_id	INTEGER		
source_asn	INTEGER		
source_ipaddr	STRING		
resource_id	INTEGER		
peer_asn	INTEGER		
peer_name	STRING		
peer_participant_id	INTEGER		
peer_ipaddr	STRING		
peer_hostname	STRING		
peer_group	STRING		
password	INTEGER		
type	STRING		
state	STRING		
prfx_max	INTEGER		
prfx_received	INTEGER		
ip_type	ENUM		enum('ipv4','ipv6') NOT NULL DEFAULT 'ipv4'
note	STRING		
created	TIMESTAMP		
modified	TIMESTAMP		
deleted	INTEGER		
public_id	INTEGER		

Example URL

/api/v1/api.php?target=peering&amp;action=getPeers&amp;public\_id=1

***addSession***

URL

/api/v1/api.php?target=peering&amp;action=addSession

Description

Adds a bgp session

Returns	<div><b>Examples:</b> SUCCESSFUL: {"success":1,"message":"Session added: Amazon.com (AS8038V1.2.3.5) - (AS16509V206.126.236.68)","data":{"id":111,"source_asn":"8038","source_configured","prfx_max":"200","prfx_received":null,"password":"ace12345 a fancy note."}}  ERROR: {"success":1,"message":"No request found.","data":{}}</div>																																																																
Required Parameters	None																																																																
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>source_asn</td><td>INTEGER</td><td>1</td><td>The numerical id of the peer</td></tr><tr><td>source_ipaddr</td><td>STRING</td><td></td><td></td></tr><tr><td>resource_id</td><td>INTEGER</td><td></td><td></td></tr><tr><td>peer_asn</td><td>INTEGER</td><td></td><td></td></tr><tr><td>peer_name</td><td>STRING</td><td></td><td></td></tr><tr><td>peer_participant_id</td><td>INTEGER</td><td></td><td></td></tr><tr><td>peer_ipaddr</td><td>STRING</td><td></td><td></td></tr><tr><td>peer_hostname</td><td>STRING</td><td></td><td></td></tr><tr><td>peer_group</td><td>STRING</td><td></td><td></td></tr><tr><td>public_id</td><td></td><td></td><td></td></tr><tr><td>type</td><td>STRING</td><td></td><td></td></tr><tr><td>ip_type</td><td>ENUM</td><td></td><td>enum('ipv4','ipv6')</td></tr><tr><td>state</td><td>STRING</td><td></td><td></td></tr><tr><td>prfx_max</td><td>INTEGER</td><td></td><td></td></tr><tr><td>note</td><td>STRING</td><td></td><td></td></tr></table>	Name	Type	Example	Description	source_asn	INTEGER	1	The numerical id of the peer	source_ipaddr	STRING			resource_id	INTEGER			peer_asn	INTEGER			peer_name	STRING			peer_participant_id	INTEGER			peer_ipaddr	STRING			peer_hostname	STRING			peer_group	STRING			public_id				type	STRING			ip_type	ENUM		enum('ipv4','ipv6')	state	STRING			prfx_max	INTEGER			note	STRING		
Name	Type	Example	Description																																																														
source_asn	INTEGER	1	The numerical id of the peer																																																														
source_ipaddr	STRING																																																																
resource_id	INTEGER																																																																
peer_asn	INTEGER																																																																
peer_name	STRING																																																																
peer_participant_id	INTEGER																																																																
peer_ipaddr	STRING																																																																
peer_hostname	STRING																																																																
peer_group	STRING																																																																
public_id																																																																	
type	STRING																																																																
ip_type	ENUM		enum('ipv4','ipv6')																																																														
state	STRING																																																																
prfx_max	INTEGER																																																																
note	STRING																																																																
Example URL	/api/v1/api.php?target=peering&action=getRequests&peer_participant_id=																																																																

### *configureSession*

URL	/api/v1/api.php?target=peering&action=configureSession											
Description	Configure a BGP session on the router											
Returns	<b>Examples:</b> SUCCESSFUL:  ERROR: {"success":0,"message":"Unable to authenticate "}											
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>session_id</td><td>INTEGER</td><td>1</td><td></td></tr></table>				Name	Type	Example	Description	session_id	INTEGER	1	
Name	Type	Example	Description									
session_id	INTEGER	1										

Optional Parameters				
	<b>Name</b>	<b>Type</b>	<b>Example</b>	<b>Description</b>
	session_id	INTEGER	1	The numerical id of the peer
	source_ipaddr	STRING		
	resource_id	INTEGER		
	peer_asn	INTEGER		
	peer_name	STRING		
	peer_participant_id	INTEGER		
	peer_ipaddr	STRING		
	peer_hostname	STRING		
	peer_group	STRING		
	public_id			
	type	STRING		
	ip_type	ENUM		enum('ipv4','ipv6')
	state	STRING		
	prfx_max	INTEGER		
	note	STRING		
Example URL	/api/v1/api.php?target=peering&action=configureSession&session_id=5			

<b><i>deleteSession</i></b>	
URL	/api/v1/api.php?target=peering&action=deleteSession
Description	Delete sessions matching criteria
Returns	<b>Examples:</b> SUCCESSFUL: {"success":1,"message":"1 sessions deleted."} ERROR: {"success":0,"message":"No sessions found to delete."}
Required Parameters	None

## Optional Parameters

Name	Type	Example	Description
id	INTEGER	41	
public_id	INTEGER		
source_asn	INTEGER		
source_ipaddr	STRING		
resource_id	INTEGER		
peer_asn	INTEGER		
peer_name	STRING		
peer_participant_id	INTEGER		
peer_ipaddr	STRING		
peer_hostname	STRING		
peer_group	STRING		
password	INTEGER		
type	STRING		
state	STRING		
prfx_max	INTEGER		
prfx_received	INTEGER		
ip_type	ENUM		enum('ipv4','ipv6') NOT NULL DEFAULT 'ipv4'
note	STRING		
created	TIMESTAMP		
modified	TIMESTAMP		
deleted	INTEGER		
public_id	INTEGER		

Example URL

/api/v1/api.php?target=peering&amp;action=deleteSession&amp;id=171

***updateSession***

URL

/api/v1/api.php?target=peering&amp;action=updateSession

Description

Updates session values with any new values specified

Returns	<p><b>Examples:</b></p> <p>SUCCESSFUL:{"success":1,"message":"Session updated: 123.net (AS32787V1.2.3.4) - (AS12129V206.126.236.70)","data":{"id":"41","source_asn":"32787","source_ipaddr":"123.net","public_id":"1","public_name":"Equinix Ashburn","ip_type":"ipv4","type":"Peer","state":"not configured","prfx_max":"10","prfx_received":null,"password":"0","note":"/an awesome note."}}</p> <p>ERROR:</p>																																																																																															
Required Parameters	None																																																																																															
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>id</td><td>INTEGER</td><td>41</td><td></td></tr><tr><td>public_id</td><td>INTEGER</td><td></td><td></td></tr><tr><td>source_asn</td><td>INTEGER</td><td></td><td></td></tr><tr><td>source_ipaddr</td><td>STRING</td><td></td><td></td></tr><tr><td>resource_id</td><td>INTEGER</td><td></td><td></td></tr><tr><td>peer_asn</td><td>INTEGER</td><td></td><td></td></tr><tr><td>peer_name</td><td>STRING</td><td></td><td></td></tr><tr><td>peer_participant_id</td><td>INTEGER</td><td></td><td></td></tr><tr><td>peer_ipaddr</td><td>STRING</td><td></td><td></td></tr><tr><td>peer_hostname</td><td>STRING</td><td></td><td></td></tr><tr><td>peer_group</td><td>STRING</td><td></td><td></td></tr><tr><td>password</td><td>INTEGER</td><td></td><td></td></tr><tr><td>type</td><td>STRING</td><td></td><td></td></tr><tr><td>state</td><td>STRING</td><td></td><td></td></tr><tr><td>prfx_max</td><td>INTEGER</td><td></td><td></td></tr><tr><td>prfx_received</td><td>INTEGER</td><td></td><td></td></tr><tr><td>ip_type</td><td>ENUM</td><td></td><td>enum('ipv4','ipv6') NOT NULL DEFAULT 'ipv4'</td></tr><tr><td>note</td><td>STRING</td><td></td><td></td></tr><tr><td>created</td><td>TIMESTAMP</td><td></td><td></td></tr><tr><td>modified</td><td>TIMESTAMP</td><td></td><td></td></tr><tr><td>deleted</td><td>INTEGER</td><td></td><td></td></tr><tr><td>public_id</td><td>INTEGER</td><td></td><td></td></tr></table>				Name	Type	Example	Description	id	INTEGER	41		public_id	INTEGER			source_asn	INTEGER			source_ipaddr	STRING			resource_id	INTEGER			peer_asn	INTEGER			peer_name	STRING			peer_participant_id	INTEGER			peer_ipaddr	STRING			peer_hostname	STRING			peer_group	STRING			password	INTEGER			type	STRING			state	STRING			prfx_max	INTEGER			prfx_received	INTEGER			ip_type	ENUM		enum('ipv4','ipv6') NOT NULL DEFAULT 'ipv4'	note	STRING			created	TIMESTAMP			modified	TIMESTAMP			deleted	INTEGER			public_id	INTEGER		
Name	Type	Example	Description																																																																																													
id	INTEGER	41																																																																																														
public_id	INTEGER																																																																																															
source_asn	INTEGER																																																																																															
source_ipaddr	STRING																																																																																															
resource_id	INTEGER																																																																																															
peer_asn	INTEGER																																																																																															
peer_name	STRING																																																																																															
peer_participant_id	INTEGER																																																																																															
peer_ipaddr	STRING																																																																																															
peer_hostname	STRING																																																																																															
peer_group	STRING																																																																																															
password	INTEGER																																																																																															
type	STRING																																																																																															
state	STRING																																																																																															
prfx_max	INTEGER																																																																																															
prfx_received	INTEGER																																																																																															
ip_type	ENUM		enum('ipv4','ipv6') NOT NULL DEFAULT 'ipv4'																																																																																													
note	STRING																																																																																															
created	TIMESTAMP																																																																																															
modified	TIMESTAMP																																																																																															
deleted	INTEGER																																																																																															
public_id	INTEGER																																																																																															
Example URL	/api/v1/api.php?target=peering&action=updateSession&note=Adding+ar																																																																																															

### *resetPeerStatus*

URL	/api/v1/api.php?target=peering&action=resetPeerStatus												
Description													
Returns	<p><b>Examples:</b></p> <p>SUCCESSFUL: {"success":1,"message":"1&amp;1 Internet status reset","data":{"id":"262","public_id":"1","asn":"8560","name":"1&amp;1 Internet","qualified":true,"is_peer":0,"request_status":"none","info_prefix":status reset","time":"2014-05-22 23:14:54","request_id":null,"session_id":null,"public_id":"1"},"message":status reset","time":"2014-05-22 23:14:18","request_id":null,"session_id":null,"public_id":"1"},"message":deleted: 1&amp;1 Internet (AS32787V1.2.3.4) - (AS8560V206.126.236.200)","time":"2014-05-22 22:39:43","request_id":null,"session_id":"71","public_id":"1"},"message":sent: ","time":"2014-04-12 13:24:43","request_id":"121","session_id":null,"public_id":"1"},"message":added: 1&amp;1 Internet (AS32787V1.2.3.4) - (AS8560V206.126.236.200)","time":"2014-04-07 11:32:37","request_id":null,"session_id":"71","public_id":"1"}}}</p> <p>ERROR: {"success":0,"message":"Could not find peer matching parameters"}</p>												
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>participant_id</td><td>INTEGER</td><td>262</td><td>The id of the peer in from the peeringDB peerParticipants table.</td></tr><tr><td>public_id</td><td>INTEGER</td><td>1</td><td>The id of the exchange point from the peeringDB mgmtPublics table.</td></tr></table>	Name	Type	Example	Description	participant_id	INTEGER	262	The id of the peer in from the peeringDB peerParticipants table.	public_id	INTEGER	1	The id of the exchange point from the peeringDB mgmtPublics table.
Name	Type	Example	Description										
participant_id	INTEGER	262	The id of the peer in from the peeringDB peerParticipants table.										
public_id	INTEGER	1	The id of the exchange point from the peeringDB mgmtPublics table.										
Optional Parameters	None												
Example URL	/api/v1/api.php?target=peering&action=resetPeerStatus&participant_id=												

### *sendRequest*

URL	/api/v1/api.php?target=peering&action=sendRequest
Description	Send a peering request (email) to a prospective peer. This will be deprecated in the next version for a simpler call, strongly suggest against using.

Returns	<p><b>Examples:</b></p> <p>SUCCESSFUL: {"success":1,"message":"Request sent", "data":{"id":"922","public_id":"1","asn":"10933","name":"ATX Communications, Inc.", "qualified":true,"is_peer":0,"request_status":"sent", "info_prefixes":n sent to ", "time":"2014-05-27 16:59:01", "request_id":"181", "session_id":null, "public_id":"1"}}, {"message sent to ", "time":"2014-05-27 16:49:30", "request_id":"171", "session_id":null, "public_id":"1"}]}</p> <p>ERROR: {"success":0,"message":"Internal error"}</p>																																																
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>public_id</td><td>INTEGER</td><td></td><td></td></tr><tr><td>peer_participant_id</td><td>INTEGER</td><td></td><td></td></tr><tr><td>source_participant_id</td><td>INTEGER</td><td></td><td></td></tr><tr><td>peer_name</td><td>STRING</td><td></td><td></td></tr><tr><td>peer_asn</td><td>INTEGER</td><td></td><td></td></tr><tr><td>email_from</td><td>STRING</td><td>262</td><td></td></tr><tr><td>email_to</td><td>STRING</td><td>1</td><td></td></tr><tr><td>subject</td><td>STRING</td><td></td><td></td></tr><tr><td>body</td><td>STRING</td><td></td><td></td></tr><tr><td>type</td><td>ENUM</td><td>html</td><td>enum('text','html')</td></tr><tr><td>status</td><td>ENUM</td><td>sent</td><td>enum('sent','acce</td></tr></table>	Name	Type	Example	Description	public_id	INTEGER			peer_participant_id	INTEGER			source_participant_id	INTEGER			peer_name	STRING			peer_asn	INTEGER			email_from	STRING	262		email_to	STRING	1		subject	STRING			body	STRING			type	ENUM	html	enum('text','html')	status	ENUM	sent	enum('sent','acce
Name	Type	Example	Description																																														
public_id	INTEGER																																																
peer_participant_id	INTEGER																																																
source_participant_id	INTEGER																																																
peer_name	STRING																																																
peer_asn	INTEGER																																																
email_from	STRING	262																																															
email_to	STRING	1																																															
subject	STRING																																																
body	STRING																																																
type	ENUM	html	enum('text','html')																																														
status	ENUM	sent	enum('sent','acce																																														
Optional Parameters	None																																																
Example URL	https://ops.6connect.com/peering-demo/api/v1/api.php?target=peering& &public_id=1&type=text&email_from=ops%406connect.com&email_to= &body=%0D%0APeering%2C%0D%0A%0D%0A6connect%2C+Inc.%2																																																

<i>sendEmail</i>															
URL	/api/v1/api.php?target=peering&action=sendEmail														
Description	Send a peering request (email) to a prospective peer. This will be deprecated in the next version for a simpler call, strongly suggest against using.														
Returns	<b>Examples:</b> SUCCESSFUL:  ERROR:														
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>participant_id</td><td>INTEGER</td><td>262</td><td></td></tr><tr><td>public_id</td><td>INTEGER</td><td>1</td><td></td></tr></table>			Name	Type	Example	Description	participant_id	INTEGER	262		public_id	INTEGER	1	
Name	Type	Example	Description												
participant_id	INTEGER	262													
public_id	INTEGER	1													



Optional Parameters	None
Example URL	

### *updatePeer*

URL	/api/v1/api.php?target=peering&action=updatePeer															
Description																
Returns	<b>Examples:</b> SUCCESSFUL:  ERROR:															
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr><tr><td>participant_id</td><td>INTEGER</td><td>262</td><td></td></tr><tr><td>public_id</td><td>INTEGER</td><td>1</td><td></td></tr></table>				Name	Type	Example	Description	participant_id	INTEGER	262		public_id	INTEGER	1	
Name	Type	Example	Description													
participant_id	INTEGER	262														
public_id	INTEGER	1														
Optional Parameters	None															
Example URL																

## API Module - Resource

- Resources
  - get
  - add
  - update
  - delete

### Resources

<i>get</i>			
URL	/api/v1/api.php?target=resource&action=get		
Description	Get a resource or resources		
Returns	<b>Examples:</b> SUCCESSFUL: <pre>{"success":1,"message":"Search successful","data":[{"id":"57","name":"2nd Email","slug":"6c-contact-email2","type":"field","parent_id":"1","category_id":null,"attr":[]}]}</pre> ERROR: <pre>{"success":0,"message":"Search failed"}</pre>		
Optional Parameters			
	<b>Name</b>	<b>Type</b>	<b>Notes/Example</b>
	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> </ul>



When `attr_compare` is set to `IN`, `NOT IN`, `BETWEEN`, `NOT BETWEEN`, then `attr_value` must either be an array or a comma separated string.

You can search on multiple attributes by including an array of attribute options:

Name	Type	Notes/Example
attributes	ARRAY	<pre>var data = {   "type":   "entry",   "   attributes   ": [     {       "attr_key":       "_section",       "attr_value":       "105",     },     {       "attr_key":       "address-mail-state",       "attr_value":       "CA",     }   ],   "resources_per_page":   10 }</pre>

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	<p><b>Examples:</b></p> <p><i>/api/v1/api.php?target=resource&amp;action=add&amp;meta[name]=apitest&amp;meta[slug]=apitest</i></p> <p>SUCCESSFUL: {"success":1,"message":"Resource added", "data":{"id":1077,"name":"apitest", "slug":"apitest", "type":"entry", "meta":{"name":"apitest", "slug":"apitest", "type":"entry"}}</p> <p><i>/api/v1/api.php?target=resource&amp;action=add&amp;meta[name]=apitest&amp;meta[slug]=apitest</i></p> <p>ERROR:{"success":0,"message":"Entries must be assigned to a section"}</p>											
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Notes/Example</th></tr><tr><td>meta[name]</td><td>STRING</td><td>Name of the resource</td></tr><tr><td>meta[type]</td><td>STRING</td><td>Type of resource (entry, section, field, ect)</td></tr></table>			Name	Type	Notes/Example	meta[name]	STRING	Name of the resource	meta[type]	STRING	Type of resource (entry, section, field, ect)
Name	Type	Notes/Example										
meta[name]	STRING	Name of the resource										
meta[type]	STRING	Type of resource (entry, section, field, ect)										

Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Notes/Example</th></tr><tr><td>meta[parent_id]</td><td>INTEGER</td><td>ID of the parent resource</td></tr><tr><td>meta[category_id]</td><td>INTEGER</td><td>ID of the category</td></tr></table>	Name	Type	Notes/Example	meta[parent_id]	INTEGER	ID of the parent resource	meta[category_id]	INTEGER	ID of the category
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)	<div>One of the following:</div> <table><tr><th>Name</th><th>Type</th><th>Notes/Example</th></tr><tr><td>meta[section_id]</td><td>INTEGER</td><td>ID of the section that the entry will be assigned to</td></tr><tr><td>meta[section]</td><td>STRING</td><td>Slug of the section that the entry will be assigned to</td></tr></table>	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
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-inst  If the field instance is left blank, it will simply be the next value in the instance array. For example:</p> <pre>fields[network-fqdn][]=e</pre> <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	<p>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.</p>

Required Parameters												
(meta[type] = field)	<table><tr><th>Name</th><th>Type</th><th>Notes/Example</th></tr><tr><td>meta[field_type]</td><td>STRING</td><td>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></td></tr></table>	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>					
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)	<table><tr><th>Name</th><th>Type</th><th>Notes/Example</th></tr><tr><td>meta[help_block]</td><td>STRING</td><td>Fields can have a line of text under them with instructions</td></tr><tr><td>meta[options]</td><td>ARRAY</td><td>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&amp;meta[options]=array(<ul style="list-style-type: none"><li>choicebox</li><li>dropdown options of Blue and Green.</li></ul>)</td></tr></table>	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[options]=array( <ul style="list-style-type: none"><li>choicebox</li><li>dropdown options of Blue and Green.</li></ul> )		
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[options]=array( <ul style="list-style-type: none"><li>choicebox</li><li>dropdown options of Blue and Green.</li></ul> )										

update											
URL	/api/v1/api.php?target=resource&action=update										
Description	Update a resource.										
Returns	<b>Examples:</b> SUCCESSFUL: <code>{"success":1,"message":"Resource Updated","data":{"id":"1055","name":"87-child-1","slug":"87-child-1","type":1}}</code>  ERROR: <code>{"success":0,"message":"No resource found with ID: 1079"}</code>										
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Notes/Example</th></tr><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></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)

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> <p>fields[position][key]</p> <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> <p>fields[0][id]=2 fields[0][slug]=asset-ser fields[0][help_block]=so fields[0][new]=false</p> <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> <p>fields[10][name]=TextA fields[10][field_type]=te fields[10][new]=true</p>

*delete*

URL	/api/v1/api.php?target=resource&action=delete								
Description	Delete a resource.								
Returns	<b>Examples:</b> SUCCESSFUL: <code>{"success":1,"message":"Resource deleted."}</code> ERROR: <code>{"success":0,"message":"No resource found with ID: 57"}</code>								
Required Parameters	<table><tr><th>Name</th><th>Type</th><th>Notes/Example</th></tr><tr><td>id</td><td>INTEGER</td><td>ID of the resource</td></tr></table>			Name	Type	Notes/Example	id	INTEGER	ID of the resource
Name	Type	Notes/Example							
id	INTEGER	ID of the resource							
Optional Parameters	<table><tr><th>Name</th><th>Type</th><th>Notes/Example</th></tr><tr><td>recursive</td><td>BOOL</td><td>When 1, deletes parent and child entries for the resource</td></tr></table> <p>A recursive delete will delete all resources, which are permitted to be deleted, from the bottom up.</p> <p>Imagine the following hierarchy:</p> <div><div>A<div><div>B1<div>C11C12</div></div><div>B2<div>C21</div></div></div></div></div> <p>If a recursive delete is performed on A, but C21 is not deletable, the following resources would still be deleted: (B1, C11, C12, C22).</p> <p>B2 would not be deleted because it depends on C21 and A would not be deleted because it depends on B2.</p>			Name	Type	Notes/Example	recursive	BOOL	When 1, deletes parent and child entries for the resource
Name	Type	Notes/Example							
recursive	BOOL	When 1, deletes parent and child entries for the resource							
Example URL	/api/v1/api.php?target=resource&action=delete&id=57								

## How Do I...

If you want to get a jumpstart on common API use cases, you came to the right place! Expand the text areas below for walkthroughs and code samples of API calls...

Context: I unassigned an IP address and now it's in the Holding Tank. Now I want to assign an IP from the Holding Tank. I don't want to unassign an IP randomly, in case it is allocated to a Resource. What are my options?

▼ [Click here to expand...](#)

There are 3 options:

- 1) If you know the specific IP, you can use the ipam-get api call to determine if it is in Holding:

```
/api/v1/api.php?target=ipam&action=get&cidr=1.2.3.4/32

{
  id:1234,
  cidr:"1.2.3.4",
  ...
  resource_name:"Holding"
}
```

- 2) If you want to show all blocks/IPs in Holding, you can use the following ipam-get API call:

```
/api/v1/api.php?target=ipam&action=get&resourceQuery={"name":"Holding"}
```

- 3) If you know the block is in Holding, you can issue another ipam-unassign API call to move it from Holding to Available:

```
/api/v1/api.php?target=ipam&action=unassign&block=1.2.3.4/32
```

Context: I need to create a Resource Holder, assign them an IP block, then subassign some IPs out of that block to two new Resource Holders. What does this look like in Python?

▼ [Click here to expand...](#)

We broke this up in a few steps so it's easier to link together.

- 1) Let's create a Resource Holder called "Ned"

```
query_string =
'target=resource&action=add&meta[type]=entry&meta[section]=resource-holder&meta[name]=Ned'
query_string += '&apiKey=' + api_key
hash = base64.b64encode( hmac.new(api_secret_key, query_string,
hashlib.sha256).digest() )
url = base_url + '?' + query_string + '&hash=' + hash
print 'Create Ned resource holder'
print url, "\n"
data = json.load(urllib2.urlopen(url))
ned_resource_id = data['data']['id']
```

- 2) Now let's add the 213.29.27.0/24 IP block

```
query_string = 'target=ipam&action=add&rir=RIPE&block=213.29.27.0/24'
query_string += '&apiKey=' + api_key
hash = base64.b64encode( hmac.new(api_secret_key, query_string,
hashlib.sha256).digest() )
url = base_url + '?' + query_string + '&hash=' + hash
print 'Create 213.29.27.0/24 block'
print url, "\n"
data = json.load(urllib2.urlopen(url))
```

3) With the block in the system, we can assign 213.29.27.0/24 to "Ned" the Resource Holder

```
query_string = "target=ipam&action=directAssign&block=213.29.27.0/24&resourceId=%d"
% (ned_resource_id)
query_string += '&apiKey=' + api_key
hash = base64.b64encode( hmac.new(api_secret_key, query_string,
hashlib.sha256).digest() )
url = base_url + '?' + query_string + '&hash=' + hash
print 'Assign 213.29.27.0/24 block to Ned'
print url, "\n"
data = json.load(urllib2.urlopen(url))
```

4) Since we plan on assigning IPs out of this block, we should enable subassignments for 213.29.27.0/24

```
query_string =
'target=ipam&action=update&block=213.29.27.0/24&allowSubAssignments=true'
query_string += '&apiKey=' + api_key
hash = base64.b64encode( hmac.new(api_secret_key, query_string,
hashlib.sha256).digest() )
url = base_url + '?' + query_string + '&hash=' + hash
print 'Update 213.29.27.0/24 to allow sub assignments'
print url, "\n"
data = json.load(urllib2.urlopen(url))
```

5) Now let's create a Resource Holder "Tara"

```
query_string =
"target=resource&action=add&meta[type]=entry&meta[section]=resource-holder&meta[name]=T"
% (ned_resource_id)
query_string += '&apiKey=' + api_key
hash = base64.b64encode( hmac.new(api_secret_key, query_string,
hashlib.sha256).digest() )
url = base_url + '?' + query_string + '&hash=' + hash
print 'Create Tara resource holder'
print url, "\n"
data = json.load(urllib2.urlopen(url))
tara_resource_id = data['data']['id']
```

6) To keep it interesting, let's create another Resource Holder "Una"

```
query_string =
"target=resource&action=add&meta[type]=entry&meta[section]=resource-holder&meta[name]=U"
% (ned_resource_id)
query_string += '&apiKey=' + api_key
hash = base64.b64encode( hmac.new(api_secret_key, query_string,
hashlib.sha256).digest() )
url = base_url + '?' + query_string + '&hash=' + hash
print 'Create Una resource holder'
print url, "\n"
data = json.load(urllib2.urlopen(url))
una_resource_id = data['data']['id']
```

7) Assign a /28 block from Ned's 213.29.27.0/24 to Tara

```

query_string =
"target=ipam&action=smartAssign&type=ipv4&rir=RIPE&mask=28&&resourceId=%d&assignedResou
% (tara_resource_id, ned_resource_id)
query_string += '&apiKey=' + api_key
hash = base64.b64encode( hmac.new(api_secret_key, query_string,
hashlib.sha256).digest() )
url = base_url + '?' + query_string + '&hash=' + hash
print 'Assign block from Ned\'s 213.29.27.0/24 to Tara'
print url, "\n"
data = json.load(urllib2.urlopen(url))

```

8) Then assign another /28 block from Ned's 213.29.27.0/24 to Una

```

query_string =
"target=ipam&action=smartAssign&type=ipv4&rir=RIPE&mask=28&&resourceId=%d&assignedResou
% (una_resource_id, ned_resource_id)
query_string += '&apiKey=' + api_key
hash = base64.b64encode( hmac.new(api_secret_key, query_string,
hashlib.sha256).digest() )
url = base_url + '?' + query_string + '&hash=' + hash
print 'Assign block from Ned\'s 213.29.27.0/24 to Una'
print url, "\n"
data = json.load(urllib2.urlopen(url))

```

Context: I need to set up a DNS server using ProVision's API in PHP, create a zone with a few simple records, and push it to the server.

✓ [Click here to expand...](#)

1) Start with providing instance information, API key, Secret Key, and DNS Server IP

```

<?php
//
//
// supply the URL of your ProVision instance, your API key and your Secret key.
$proVisionURL = "https://ops.6connect.com/qa-4.2.2";
$apiKey = "Nnvz8xKZDQUWke6gDxb";
$apiSecretKey = "2YojRbrHnToPZ7cDeFBzcTAvcfMbPVMX";
// this example uses 6connect's PHP APIClient
require_once("APIClient.php");
// set up the connection
$apiClient = new APIClient($proVisionURL, $apiKey, $apiSecretKey);

// save this. IP of the DNS Server we're creating.
$serverIp = "208.39.106.184";

```

2) Add a DNS server



```

// begin making api calls. We begin by adding a simple DNS server.
$params = array();
$params['displayName'] = "Example Server";           // the pretty name of the DNS server
$params['server'] = "208.39.106.184";               // the IP of the DNS Server
$params['active'] = 1;                             // whether or not this server is currently enabled
$params['transferType'] = "SCP";                    // we are using an
ISC Bind server which we will communicate with via SCP
$params['username'] = "6connect";                    // the username used to SCP zones to this
server
$params['password'] = "password";                    // the password used to SCP zones to this
server
$params['port'] = 22;                               // the port used
to SCP zones to this server
$params['serverType'] = "master";                    // whether this server is a master or a
slave
$params['SOA'] = "ns1.dns.6connect.net. hostmaster.6connect.net."; // the default SOA
$params['remoteDirectory'] = "/tmp/";               // where to place the zone files on the
server
$params['namedConfPath'] = "/tmp/";                 // the path to the zones within the
configuration file. Usually the same as 'remoteDirectory'
$params['postCommand'] = "touch /tmp/allFinished";   // the command to execute on
the server after the transfer is complete.
// add the server
$apiResponse = $apiClient->sendRequest('dnsServer', 'add', $params);
if ($apiResponse->status == 1) {
    echo "Successfully added DNS Server '" . $params['displayName'] . "'\n";
} else {
    echo "Could not add DNS Server '" . $params['displayName'] . "' !\n";
    die();
}

// now we fetch the id of our newly created server
$params = array();
$apiResponse = $apiClient->sendRequest('dnsServer', 'get', $params);
$data = $apiResponse->data;
for ($i = 0; $i < count($data); $i++) {
    if ($data[$i]['server'] == $serverIp) {
        // we save the id for later.
        $serverId = $data[$i]['id'];
        break;
    }
}
echo "Server Id is: $serverId \n";

```

### 3) Create a zone

```
// okay, DNS server is set up -- time to create a zone.
$params = array();
$params['zoneName'] = "atestzone.com";    // zone name
$params['zoneResourceId'] = 1;           // the owner of the zone; 1 is default
$response = $apiClient->sendRequest('zone', 'add', $params);
if ($response->status == 1) {
    echo "Successfully added DNS Zone '" . $params['zoneName'] . "'\n";
} else {
    echo "Could not add DNS Zone '" . $params['zoneName'] . "' !\n";
    die();
}
// snag the zoneId for later.
$zoneId = $response->data;
```

#### 4) Add Zone records

```

// Lets add some records to our new zone!
$params = array();
$params['newRecordZoneId'] = $zoneId;           // parent zone id
$params['newRecordType'] = 'A';                 // record type
$params['newRecordHost'] = "www";               // the host field of the record
$params['newRecordValue'] = "1.2.3.4";          // the value field of the
record
$params['newRecordTTL'] = "3600";                // the value of the TTL field
$apiResponse = $apiClient->sendRequest('record', 'add', $params);
if ($apiResponse->status == 1) {
    echo "Successfully added Record to zone #{$zoneId}\n";
} else {
    echo "Could not add Record to zone #{$zoneId}!\n";
    die();
}

$params = array();
$params['newRecordZoneId'] = $zoneId;           // parent zone id
$params['newRecordType'] = 'A';                 // record type
$params['newRecordHost'] = "dev";               // the host field of the
record
$params['newRecordValue'] = "2.3.4.5";          // the value field of the record
$params['newRecordTTL'] = "3600";                // the value of the TTL field
$apiResponse = $apiClient->sendRequest('record', 'add', $params);
if ($apiResponse->status == 1) {
    echo "Successfully added Record to zone #{$zoneId}\n";
} else {
    echo "Could not add Record to zone #{$zoneId}!\n";
    die();
}

$params = array();
$params['newRecordZoneId'] = $zoneId;           // parent zone id
$params['newRecordType'] = 'A';                 // record type
$params['newRecordHost'] = "cloud";             // the host field of the record
$params['newRecordValue'] = "3.4.5.6";          // the value field of the
record
$params['newRecordTTL'] = "3600";                // the value of the TTL field
$apiResponse = $apiClient->sendRequest('record', 'add', $params);
if ($apiResponse->status == 1) {
    echo "Successfully added Record to zone #{$zoneId}\n";
} else {
    echo "Could not add Record to zone #{$zoneId}!\n";
    die();
}

```

4) Link the Zone to the new DNS server and push

```
// Okay, we have some zones with records. Time to link this zone to the new DNS
Server
$params = array();
$params['serverId'] = $serverId;      // the server id
$params['zoneId'] = $zoneId;         // the zone id
$params['serverSlave'] = 0;          // not a slave zone
$apiResponse = $apiClient->sendRequest('zoneLinkage', 'add', $params);
if ($apiResponse->status == 1) {
    echo "Successfully linked Zone #{$zoneId} to server #{$serverId}\n";
} else {
    echo "Could not link Zone #{$zoneId} to server #{$serverId}!\n";
    die();
}
// now we can push the zone to the server
$params = array();
$params['zoneId'] = $zoneId;          // the zone id to push
$apiResponse = $apiClient->sendRequest('dnsServer', 'transferSingle', $params);
if ($apiResponse->status == 1) {
    echo "Zone pushed!\n";
} else {
    echo "Could not push zone!\n";
    die();
}
?>
```

Context: How do I update the notes field of an IP block using the API in PHP?

▼ [Click here to expand...](#)

- 1) Start with providing instance information, API key, Secret Key, and DNS Server IP; set up the connection

```
<?php
//
// This file walks through an example of how to look up a block id number
// in ProVision, and then use it to attach a notes field
//
// supply the URL of your ProVision instance, your API key and your Secret key.
$provisionURL = "https://ops.6connect.com/qa-4.2.2";
$apiKey = "32-5DAYTJEE2TZHOFOB";
$apiSecretKey = "48b278ec873bda473a323dbc467f8669";
// this example uses 6connect's PHP APIClient
require_once("APIClient.php");
// set up the connection
$apiClient = new APIClient($provisionURL, $apiKey, $apiSecretKey);
```

- 2) Split the metadata you want to have showing in the notes, and find the block with which it should associate

```
// lets imagine we have some metadata in the following format:
//
$string = "10.1.245.5|DFW7|HP a5820x|its-erp.dfw7.us.corp|";
//
// And we want to insert the Colo, Server type, and hostname into the Notes field of
the IP block

// first we split everything up
$pieces = explode("|", $string);
$ip = $pieces[0];
$colo = $pieces[2];
$type = $pieces[3];
$host = $pieces[4];

// then we pull the IP block using the API.
$params = array();
$params['block'] = "$ip/32"; // the IP block we're looking for, with netmask
// make the call to the IPAM-GET endpoint
$apiResponse = $apiClient->sendRequest('ipam', 'get', $params);
if ($apiResponse->status != 1) {
    echo "Could not pull information for block: $ip/32 !\n";
    die();
}
if (trim($apiResponse->message) == "No blocks found.") {
    echo "IP block $ip/32 not found in ProVison!\n";
    die();
}

// we now have the ipObject associated with this IP block. Lets get its block id.
$blockId = $apiResponse->data[0]['id'];
echo "IP block id: $blockId \n";
```

### 3) Update the block with the notes

```
// it is time to update the block with the new notes.
$notes = "$colo,$type,$host";
$params = array();
$params['id'] = $blockId;
$params['notes'] = $notes;
// make the call to the IPAM-UPDATE endpoint
$apiResponse = $apiClient->sendRequest('ipam', 'update', $params);

// and done!
echo $apiResponse->message . "\n";
```

# CLI (Alpha)

## Command Line Interface - 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.

## Help & Support

# Help & Support

For setup assistance or additional information, you can contact our support team at [support@6connect.com](mailto:support@6connect.com).

For tutorials, frequently asked questions, feedback, or additional resources such as import templates and previous documentation versions, please follow the links listed below.

### Table of contents

- [Tutorials](#)
- [FAQ](#)
- [Additional Resources](#)
- [Feedback and Feature Requests](#)



## 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).

### *Table of Contents*

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

## Common Tasks

### IPAM

[Adding/Editing blocks](#)

[Reserve IP space](#)

[Aggregating/Splitting blocks](#)

[SWIP configuration and use](#)

[RPSL configuration and use](#)

### DNS

[Importing DNS Zones](#)

### Peering

[Adding routers](#)

[Adding sessions](#)

[Importing sessions](#)

### Importing Data

[Resource Importer Walkthrough](#)

[Import Aggregate Blocks](#)

[Import DNS Zones](#)

## UI Tours

### Administration

#### *Managing Group and User Permissions*



### DNS

#### *PowerDNS w/ MySQL Support*



## FAQ

## 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) if you have any additional questions or this does not resolve the issue.

### ✓ How can I 'reserve' IP space?

To create a reserved pool of IP space, you can create a Section called "Reserved", add the IPAM gadget to it, then create an Entry with that

Section to be the address group. From there, use the IPAM gadget and the IPAM Manage page to assign and unassign IP space from that pool.

The workflow for this would be:

1. Assign IP space to the "Reserved" Section.
2. When you are ready to pull space from "Reserved", unassign the desired block. This moves it to the holding tank.
3. Override the holding tank to make the space "available". This can be done in the IPAM manager via the "Override Holding" wrench option, or a manual 'pull out of holding' API call.
4. Assign the block to the desired Resource.

▼ [How do I change the URL of my ProVision instance?](#)

Depending on your version of ProVision, you may need both steps. Edit the file <6connect web root>/data/globals.php and:

- 1) Change the \$hostname variable to the new value
- 2) Change the \$base\_url to the new value

Please note that you may also need to update the SSL certs, httpd settings, etc.

## Additional Resources

- [Import Templates](#)
- [List of Abbreviations](#)
- [Previous Documentation Versions](#)

# Import Templates

## Import Templates

### Downloadable Import Templates

Below you can find CSV templates for uploading Resource, Contact and IP data.

For DNS Import examples and a walkthrough, visit the [DNS Import](#) page.

File	Me
>  contact-import-sample_v1.csv	De
>  import-zone-assign.csv	Nc
>  customer-import-sample.csv	Al
>  IP-import-sample_v1.csv	Ju

Drag and drop to upload or [browse for files](#)

 [Download All](#)

## List of Abbreviations

### List of Abbreviations:

[Edit Document](#)

<b>API</b>	Application Program Interface
<b>CLI</b>	Command-line interface
<b>DHCP</b>	Dynamic Host Configuration Protocol
<b>DNS</b>	Domain Name System
<b>DNSSec</b>	Domain Name System Security Extensions
<b>IP address</b>	Internet Protocol address
<b>IPAM</b>	Internet Protocol address management
<b>LDAP</b>	Lightweight Directory Access Protocol)
<b>SDK</b>	software development kit
<b>SSH</b>	Secure Shell

[Abbreviation List.xlsx](#)



# Previous Documentation Versions

## Documentation for Previous Versions of 6connect software:

Archived Online Documentation:

[ProVision v 5.0.3 Documentation](#)

Archived PDF Documentation:



[PDF Documentation](#)

File	Mc
>  6c-ProVision-v4.2.1_Manual.pdf	No
>  6C-v2.5.13-Manual.pdf	Au
>  6C-v3.0-Manual.pdf	Au
>  6connect-Service_Provider_Edition_3...	Au

 [Download All](#)

## 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)

## 6connect Scanlet (Beta)

### Scanlet (Beta)

#### Download Information

Download the latest version [here](#)

#### System Requirements

Java: [Download Here](#)

#### Known Issues

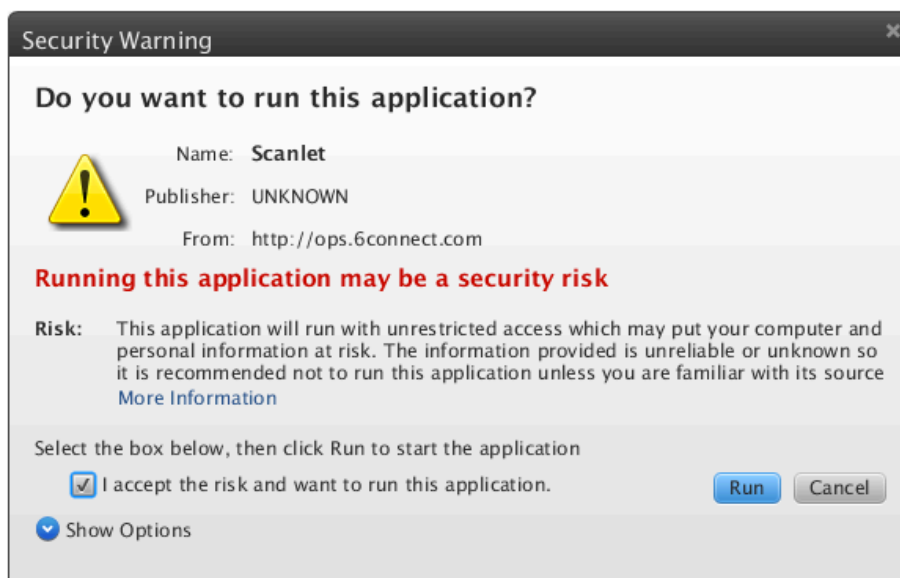
- In some cases, the application may appear to hang before it finishes. This is usually related to timeouts not being addressed quickly. We erred on the side of caution to ensure that devices were located, but the result is that sometimes unresponsive devices can make the scan take longer than expected.

### Documentation

This software is currently in Beta - please forward all feedback to [gary@6connect.com](mailto:gary@6connect.com)

#### Starting Scanlet

When you open the download URL, you should see the Java plugin activate and should be prompted with a security warning to confirm that you are knowingly running the application.

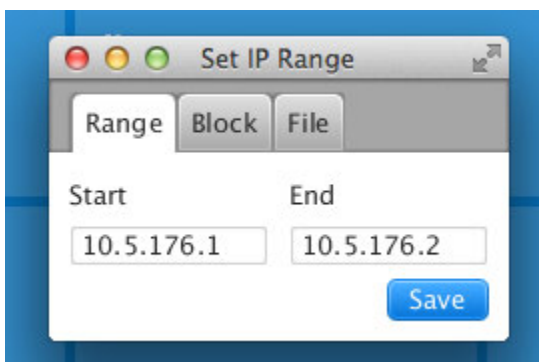


Upon checking the box and clicking the "Run" button, it should automatically bring up the UI in your browser.

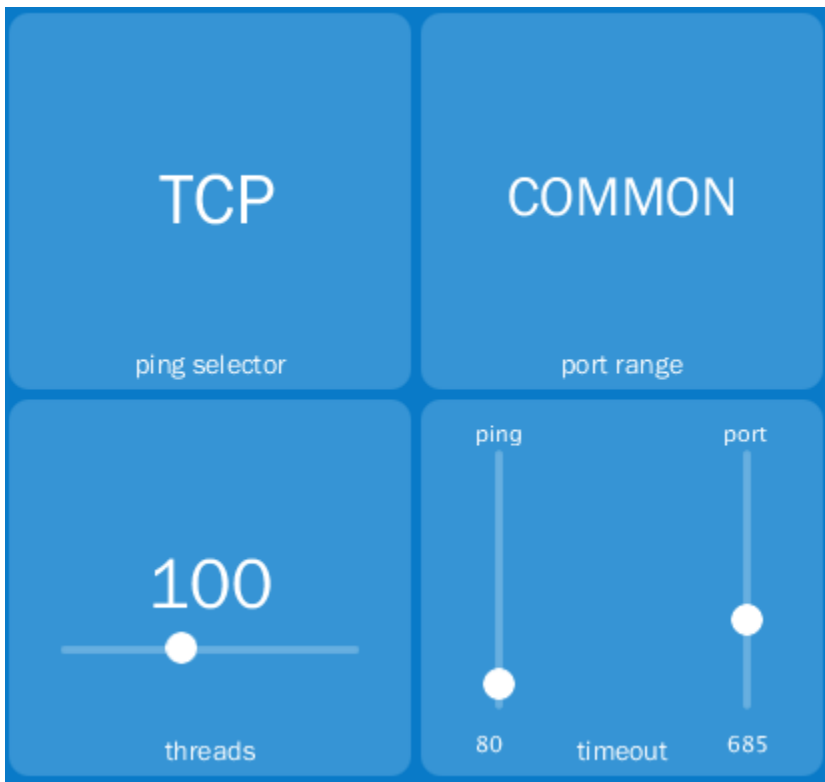
If this does not work, try to click the link at the bottom left of the webpage to manually open the application.

### Using Scanlet

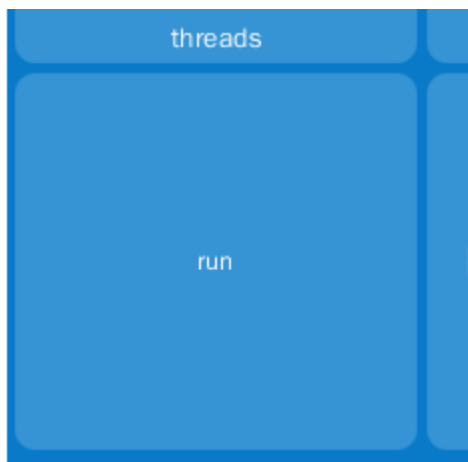
#### Running a Scan



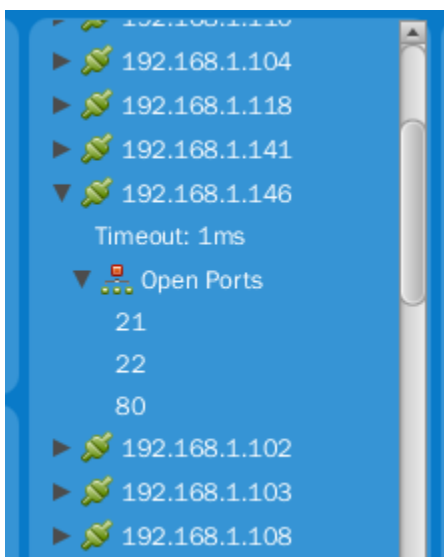
**STEP 1:** Set the IP range that you would like to scan. If you click on the "IP Range" square, you will have a window open to set the parameters.



**STEP 2:** Select the additional parameters for your scan (ping selector, Ports, thresholds, etc.)



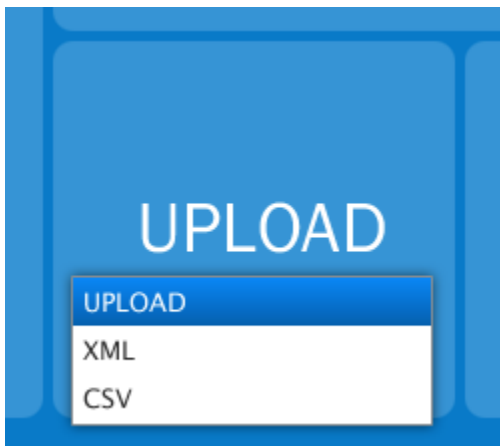
**STEP 3:** Run the scan!



**STEP 4:** View the scan results. You can expand the arrows per IP address, view Port information or MAC address data.

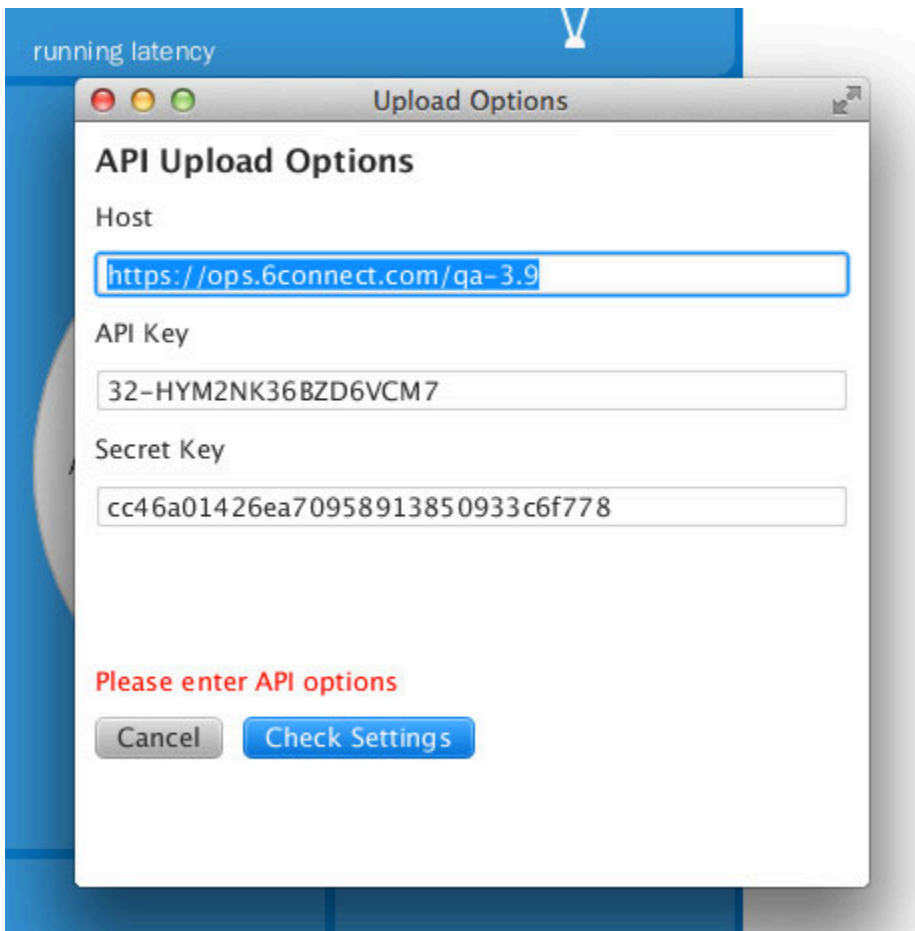
#### Uploading/Saving scan results

**Option 1:** If you have a 6connect ProVision instance to test with, you can upload your scan results into the platform using the API with the "UPLOAD" selection.



When you click on the "SAVE" button, you will be directed to enter the API information for your particular instance. Discovered devices will be imported as type "Scanlet". We will be pushing out an update that will allow you to modify different types and even allow some more detailed

hierarchies from the initial discovery model.



**Option 2:** If you would like to save an XML/CSV version locally to your computer, simply select the format you prefer. When you click on the Save button, it will prompt you for a location to save the file.

## **Release Notes**

### **Release Notes**

- ProVision 5.0.0
- ProVision 4.2.0
- ProVision 4.1.0
- ProVision 4.0.0
- 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 5.0.0

ProVision 5.0.0 is a major release with multiple customer feature requests and significant new features



### PHP Compatibility

Please note that ProVision 5.x requires php 5.5.+. For local installations, please upgrade php prior to installing the upgrade. Also ensure that the correct Sourceguardian php extension is loaded for the new version of php.

Contact 6connect at [info@6connect.com](mailto:info@6connect.com) to schedule a demo or get more information.

▼ [Click here to see 5.0.x Release Notes...](#)

- [ProVision 5.0.4](#)
- [ProVision 5.0.3](#)
- [ProVision 5.0.2](#)
- [ProVision 5.0.1](#)

## New Features

(CFR denotes customer requested)

### Resource Importer

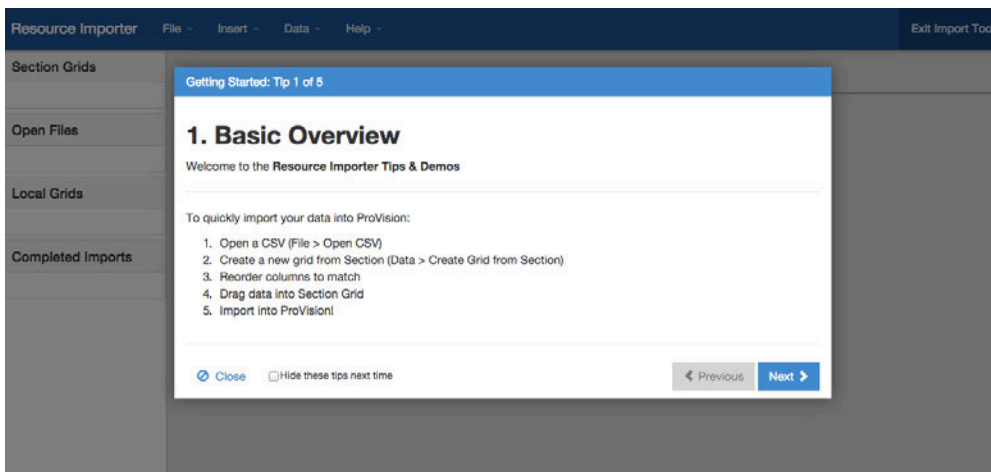
Click on the following links for more details on the [Resource Import Tool](#), and a [Resource Importer Walkthrough](#).

Updates include:

IM- 1466: Resource Importer opens in a new tab

IM- 1490: Improved ability to adjust column locations

IM- 1557: New on-screen guidance



IM- 1580: Opening a .csv searches the first 10 rows for a valid header

IM- 1581: Streamlined menu options

IM- 1582: Removed resource search

IM- 1589: Revised appearance of required data columns

IM- 1591: Added UTF-8 warning



## IM- 1643: Mac .csv files now supported

IM- 1593: New ability to set Peer Groups when adding a router and adding a peering session:

## Add Session

TypePeer ▾

ExchangeSelect Exchange... ▾

Note

SourceASNAS8038 ▾

RouterSelect Router... ▾

IP Address

DestinationPeer GroupSelect Peer Group... ▾

MD5

Max Prefixes

Select peer and public IP data PeeringDB or specify custom data for the session.

PeerPeer Name ▾

Public IPPublic IP (from PeeringDB) ▾

Peer

ASN

IP Address

☐ Configure router after saving?

Save

IM- 1552: New "Configure" menu option for one-click config of session on the router

IM- 1553: Sessions of type "Peer" are removed from the router when deleted in ProVision

IM- 1554: Peering importer differentiates between peer sessions matched to an exchange and unknown sessions

IM- 1556: Peering "Source IP Address" column changed to "Router"

IM- 1570: Import list of groups from router during session import

IM- 1574: Improved read-only peering permissions configurations

IM- 1594: Option to configure router when adding a peering session

IM- 1597: Multiple Peer Groups may be created when adding a router

IM- 1651: Improved Config Manager functionality

IM- 1653: Peering Import section added to Admin->Import

IM- 1654: Added Peer Group import button to session import

### Peer Group Gadget

The new Peer Group Gadget allows you to add peer groups for IPv4 and IPv6 for a selected exchange from a router's Resource Entry page. Peer Groups added from this gadget will be available to select in the "Add Session" dialog box. For more details on the Peer Group Gadget, see [Gadgets](#).

Note: Peer groups listed in the Gadget are for ProVision only and should reflect groups that exist on the router. Adding or deleting peer groups from the Gadget will not add or delete them on the router.

The screenshot shows the 'Peer Groups' gadget interface. At the top, there's a title 'Peer Groups'. Below it, the 'Add Peer Group:' section includes a dropdown menu labeled 'Select Exchange...', an empty text input field, radio buttons for 'IPv4' (selected) and 'IPv6', and an 'Add Group' button. Below this is a table with three columns: 'Exchange', 'Peer Group', and 'Type'. The table is currently empty.

### Additional Features

CFR-22: Smart Assign 'Exclude' tags

CFR-80: Smart Assign 'Strict Tag' and 'Standard' (match all) interpretation

The screenshot shows the 'IPAM' interface. Under the 'Assign Block:' section, there are two options: 'Browse To Assign' (with a link 'List available blocks') and 'Direct Assign' (with a text input field containing 'x.x.x.x/yy' and an 'Assign' button). Below this is the 'Smart Assign' section, which includes dropdown menus for 'IPv4', 'Size', 'RIR', and 'Region', followed by a 'Tags...' input field. To the right of these is the 'Tag selection mode:' section with three radio buttons: 'Standard' (selected), 'Strict', and 'Exclude'. At the bottom left of the 'Smart Assign' section is a 'Show advanced options' link and a 'Smart Assign' button.

CFR- 68: If IP or DNS are not assigned to a Resource, the Resource does not show up on the Dashboard

CFR- 90: Add "Allow subassignments" when editing multiple subnets in IPAM Manage

IM- 878: Removed Generate API Request

IM- 1363: TTL field added to DNS bulk records changes

**Update ALL of the above with new data:**

Host	Type	TTL	Value
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

**Replace Records**

IM- 1506: admin.php checks for a valid license key

IM- 1508: Indicator (blue arrow) on blocks in Resource Gadget and IPAM Manage when subassignments are enabled

[Export Current List To CSV](#)

<input type="checkbox"/>	10.4.0.64/26	64	Quito	Customer	Available	2013-11-01	
<input type="checkbox"/>	10.4.0.128/25	128	Quito	Customer	Available	2013-11-01	
<input type="checkbox"/>	10.4.1.0/32	1	Quito	Customer	Datacenter.12 →	2013-11-01	
<input type="checkbox"/>	10.4.1.1/32	1	Quito	Customer	Datacenter.12 →	2013-11-01	

IM- 1514: IPAM interface improved for updating/editing multiple block notes and tags. Select "Update Field" to apply change for all selected blocks.

**Edit selected blocks** **Assign selected blocks** [Export Current List](#)

**Updating information for 2 blocks**

RIR	LIR	DataCenter1	VLAN	Region	ASN
1918	Select LIR...	DS213	101	Quito	
<input type="checkbox"/> Update field	<input type="checkbox"/> Update field	<input type="checkbox"/> Update field	<input type="checkbox"/> Update field	<input type="checkbox"/> Update field	<input type="checkbox"/> Update field

**Notes**

Test 123

☒ Update field

**Select tags...**

☒ Ignore -- do not update tags information

☐ Replace tags -- replace tags for each block with the selected tags below

☐ Add tags -- add selected tags to each block

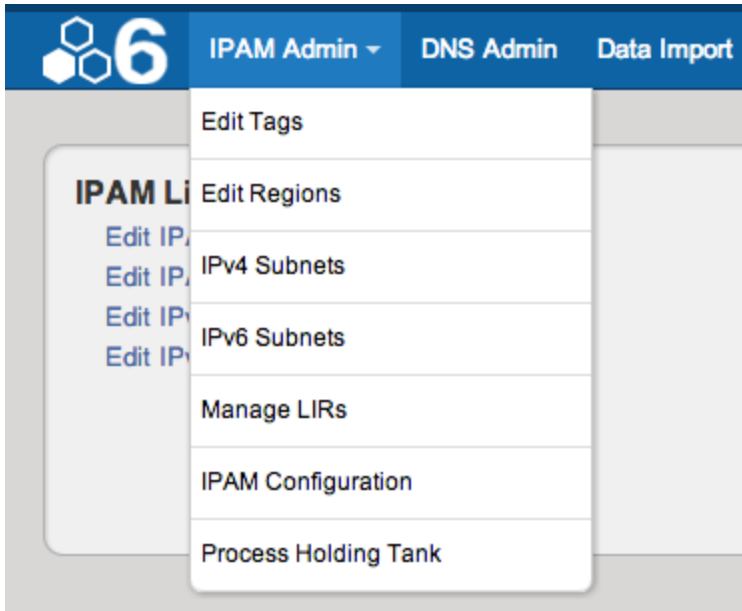
☐ Delete tags -- delete selected tags from each block

<input type="checkbox"/> MPLS	<input type="checkbox"/> Anycast	<input type="checkbox"/> Point to Point	<input type="checkbox"/> BGP
<input type="checkbox"/> Security	<input type="checkbox"/> Customer	<input type="checkbox"/> Cable	<input type="checkbox"/> Backbone
<input type="checkbox"/> Tower	<input type="checkbox"/> DSL	<input type="checkbox"/> DHCP	<input type="checkbox"/> DNS
<input type="checkbox"/> VOIP	<input type="checkbox"/> Management	<input type="checkbox"/> Internal	<input type="checkbox"/> Development

**Save** **Cancel**

## Bug Fixes/Improvements

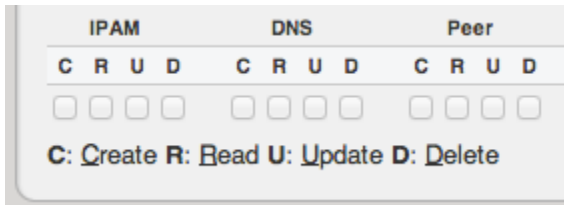
IM- 362: IPAM header tab highlighting, added dropdown menu



IM- 930: When marked peer "Not Qualified" or "Rejected, entry is no longer removed

IM- 983: Adding a new Section with blank name now provides error message

IM- 1000: Added "CRUD" legend under manager users



IM- 1061: Fixed navigation breadcrumbs

IM- 1165: Added 'logged in as' text to header

IM- 1096: Relabeled Secure64 entry in dropdown menu

IM- 1097: Updated Holding / Assignment wording on notifications

IM- 1273: configTest.php Deprecated

IM- 1282: Admin-api.php daemon user visible

IM- 1289: Treeview functionality restored

IM- 1325: Test zonesigner is working for DNSSEC

IM- 1470: Fixed error message in Admin Preferences

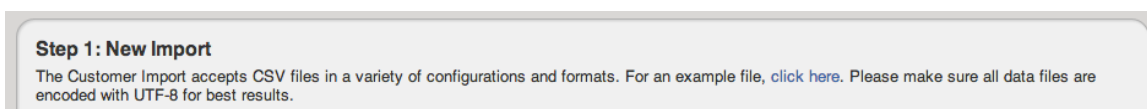
IM- 1480: Stop creation of "system" resources through the UI and API

IM- 1483: Moved EULA location

IM- 1504: Can no longer create API key for daemon user

IM- 1506: UpgradeUtil doesn't check for valid license key

IM- 1513: Included note on CSV import screens to encode with UTF-8 for best results



IM- 1519: Import directory structure updated

IM- 1526: Edit IPAM tags empty field save

IM- 1527: Resolved IPAM tags alphabetical sort

IM- 1528: Smart / Direct Assign buttons in IPAM gadget sent two requests when double-clicked

IM- 1530: Revised IPAM Gadget text display

IM- 1531: Delete aggregate button does not refresh page

IM- 1533: ConfigTest.php doesn't test for the writability of the web root

IM- 1536: IPAM gadget displays log information for IPv6 blocks

IM- 1537: Deleting a zone from the DNS Gadget removes it from the chart

IM- 1540: Edited gray text in text box for Direct Assign to be clearer for IPv6

IM- 1543: Removed LIR option from the section dropdown when creating an entry

IM- 1545: Add aggregate has "Available" as the default assignee

IM- 1549: Duplicate customer sample import link removed

IM- 1550: Import customer from salesforce link removed

IM- 1559: User activity report download .csv updated

IM- 1562: Admin up/down now works for Juniper routers

IM- 1565: Treeview top level folder renamed

IM- 1567: Additional countries available in drop down menu

IM- 1568: Block param without mask causes error for ipam-get API call

IM- 1575: Improved handling of a no permissions user on IPAM page

IM- 1583: Restricted creation/editing of LIRs through the Resource entries

IM- 1598: Improved handling of no permissions user on Sections page

IM- 1604: Updated handling of names containing apostrophes

IM- 1606: Improved 'Edit List: IPAM Tags' functionality and sorting

IM- 1613: Revised ARIN ORG handle to 'Org ID'

IM- 1620: Revised "Edited the file: regions" log message

IM- 1621: Added Custom ID field to Resource entry creation

IM- 1622: Revised the default Section and Category options when adding a Resource Entry

IM- 1624: Revised label location for 'Resource IP Assignment %' on Dashboard

IM- 1625: Revised user account count to disregard system user account

IM- 1626: API User drop down hides 'daemon' user

IM- 1627: Revised phrasing for Dashboard 'Resource IP Assignments' label

IM- 1631: Entry list page improvements

IM- 1632: Added custom ID to resources with duplicate names

IM- 1634: Changed "Type" To "Section" on Resource Entry page

IM- 1636: Improved handling of restricted permissions user on Resource Entry page

IM- 1637: Improved handling of restricted permissions user on Reporting page

IM- 1644: Improved handling of restricted permissions user on Sections page

IM- 1646: VLAN added to advanced options in IPAM gadget

IM- 1647: Revised display of Password field on routers in section field view

IM- 1652: Improved handling of restricted permissions user on Reporting page

## ProVision 5.0.4

ProVision 5.0.4 is a minor release with bug fixes and improvements



### PHP Compatibility

Please note that ProVision 5.x requires php 5.5.+. For local installations, please upgrade php prior to installing the upgrade. Also ensure that the correct Sourceguardian php extension is loaded for the new version of php.

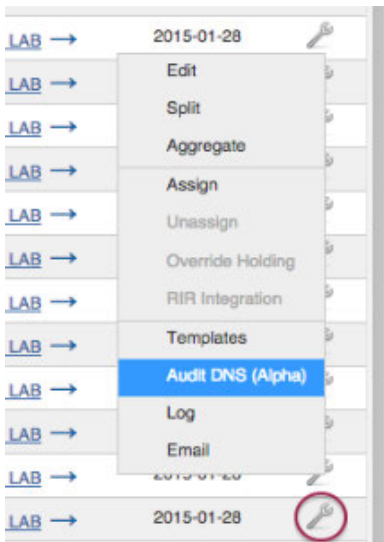
Contact 6connect at [info@6connect.com](mailto:info@6connect.com) to schedule a demo or get more information.

### New Features

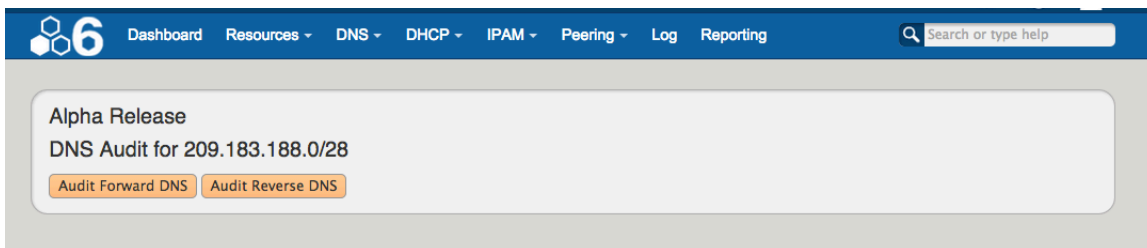
#### DNS Audit Tools (Alpha): *SEEKING USER FEEDBACK*

Introducing a first version of DNS audit tools to perform a simple audit of both forward and reverse DNS. The tools set includes UI, API end points, and a command line interface. The audit results include the DNS as found in the 6connect ProVision database, the results from a resolver, and if there is a conflict in these two pieces of information. 6connect is seeking user feedback on this tool; contact us at [support@6connect.com](mailto:support@6connect.com) with your thoughts.

Access the UI version of the Tool by going to the desired block in IPAM Manage, click on the Action Menu (wrench), then select "DNS Audit (Alpha)":



This takes you to the DNS Audit page.



From there, select the "Audit Forward DNS" or "Audit Reverse DNS" buttons to provide a list of IPs, the Reverse Values, Conflict Status, and Resolved Host(s).

Alpha Release			
DNS Audit for 209.183.188.0/28			
<div> <div>Audit Forward DNS</div> <div>Audit Reverse DNS</div> </div>			
IP	6connect Reverse	Conflict	Resolved Host(s) of IP
209.183.188.2	6lyco2.lycoming.edu.	No	209.183.188.2
209.183.188.3	6lyco3.lycoming.edu.	No	209.183.188.3

DNS Audit Tools: Additional File and Example Information:

▼ [Click here to expand...](#)

- **UI:**

`audit_dns.php`

Example:

`https://cloud.6connect.com/myinstance/audit_dns.php?block=209.183.188.0/28`

- **API:**

Files:

`api/v1/auditDNS/execute.php`

`api/v1/auditDNS/getStatus.php`

Examples:

`api/v1/api.php?target=auditDNS&action=execute&type=forward&block=209.183.188/28&jobId=1424218758`

`api/v1/api.php?target=auditDNS&action=execute&type=reverse&block=209.183.188/28&jobId=1424218758`

- **Command Line:**

Files:

`tools/audit_foward_dns.php` `tools/audit_reverse_dns.php`

Examples:

`php audit_forward_dns.php -b 209.183.188/28`

`php audit_reverse_dns.php -b 209.183.188/28`

## Juniper Logical Systems support for Peering:

ProVision now supports Juniper Logical Systems for Peering, with the Logical Systems information tied to the router resource. UI elements have been added as follows:

- A Logical Systems field appears in the the Peering "Add Router" dialog when a Juniper router is selected.

*To add Logical Systems to a router, go to the Peering Tab and click on the "Add Router" button. Select the Juniper router make/model, and the Logical System text field will appear. Enter the Logical System and router information, peer group information, and hit "Add Router" when complete. For routers with multiple associated Logical Systems, you may create duplicate router resources with different logical systems information.*



### Add Router

Parent Resource: TLR

Name:

Make: Juniper

Model: 7600 Series

**Logical System**:

Hostname:

IPv4 Address:

IPv6 Address:

Username:

Password:

**Peer Groups**

Exchange: Equinix Internet Exchange

Peer Group:

Type: ☒ IPv4 ☐ IPv6

[Add Group](#)

Exchange	Peer Group	Type
No groups specified		

[Add Router](#)



Associating the router with a peer group is necessary to link the router to a particular exchange.

Please be sure to add the Peer Group information either in the "Add Router" dialog or in the Peer Group Gadget prior to adding sessions.

- Peering - Main - Add Session: Router drop down indicates the router, then Logical System info in parenthesis (e.g. "Juniper (test)") .

After having added a Logical System to a router, that router + Logical System combination will be available to select in the Peering - Add Session dialog box. The Peer Group associated with that router / Logical System will automatically be selected. Continue to fill in your session information, then hit "Save".

### Add Session

Type: Peer

Exchange: Equinix Palo Alto

Note:

Peer Group: PeerGroup1 - ipv4

MD5:

Max Prefixes:

**Source**

Router: Juniper-LS test (test4) - 50

ASN: AS8038

**Logical System: test4**

**Destination**

Select peer and public IP data PeeringDB or specify custom data for the session.

Peer: Peer Name

Public IP: Public IP (from PeeringDB)

Peer:

ASN:

IP Address:

☐ Configure router after saving?

[Save](#)

When viewing your session in the Sessions list, your router will appear with the Logical Systems information in parenthesis.

AS8038 –	Juniper-LStest (test4)	Amazon.com	AS16509 – 198.32.176.36	PeerGroup1	Peer	0/0	not configured	
----------	------------------------	------------	-------------------------	------------	------	-----	----------------	--

[close](#)

- Peering - Main - Config Manager: When configuring a session that is part of an Logical System, the first line of the config displayed should be "set cli logical-system <name>"

## Config Manager

Router:
Username:
Password:

```
set cli logical-system test2
edit
```

- Peering - Import: When importing sessions from a Juniper router, the Router drop down indicates the router, then Logical System info in parenthesis (e.g. "Juniper (test)"). Also, there is now a Logical System column in the results.

### Peering Import

Exchange: Equinix Internet Exchange Palo
Router: Juniper-LStest (test)
Load Sessions

Importing sessions from Juniper-LStest (test) (50.240.195.137) at Equinix Palo Alto.

2 sessions found. 0 already imported or added.

#### Peer Groups

Name	Type
equinix-test	ipv4

Import Selected Groups

#### Sessions


<input checked="" type="checkbox"/>	Type	Source ASN	Peer	Peer ASN	Peer IP	Group	Logical System	State	
<input checked="" type="checkbox"/>	Unknown	12345		1234	1.2.3.4	equinix-test	test	Idle	
<input checked="" type="checkbox"/>	Peer	12345	M-ROOT DNS Server	7500	198.32.176.179	equinix-test	test	Idle	

Import Selected Sessions

## Improvements

### DNS UI Updates:

Various user interface improvements have been made to the DNS Admin section. Documentation has been updated to reflect these changes on the [DNS Admin](#) page.


[IPAM Admin](#)
[DNS Admin](#)
[Data Import](#)
[Users](#)
[API](#)
[Templates](#)
[Exit Admin](#)

### Manage DNS Servers

Server: Select Server New Server

### DNS Zone Transfers:

dns.6connect.net	12 Zones	<input type="checkbox"/>
DynECT Server	9 Zones	<input type="checkbox"/>
Secure64 Auth Server	9 Zones	<input type="checkbox"/>
cache.6connect.com	8 Zones	<input type="checkbox"/>
nalinmk.com	115 Zones	<input type="checkbox"/>
services1.tcp0.com	47 Zones	<input type="checkbox"/>
208.39.106.184	7 Zones	<input type="checkbox"/>
ns1.6clabs.com	86 Zones	<input type="checkbox"/>
ns2.6clabs.com	84 Zones	<input type="checkbox"/>
PowerDNS Server	0 Zones	<input type="checkbox"/>
208.39.106.184	44 Zones	<input type="checkbox"/>
ubuntu-testvm02	3 Zones	<input type="checkbox"/>

Push Zones to Selected Servers: Push

### DNS Defaults and Tools

- [Global DNS Zone Defaults](#)
- [DNS PTR Auto Generation Management](#)
- [DNS Record Types](#)
- [DNS View ACL Management](#)
- [Bulk DNS Change Tools](#)

### DNS Export Functions

- [Show all DS records for DNSSEC](#)
- [Generate zip file of all zones](#)

Selected changes include:

- "DNS Configuration" has been renamed "DNS Global Defaults", and is accessed from the "Global DNS Zone Defaults" link. The default options list has been simplified.
- "Nameserver Management" has been renamed "Default Nameservers", and is accessed from the "Global DNS Zone Defaults" link.
- "Bulk Zone Assignment" and "Bulk Record Changes" are now accessed from the "Bulk DNS Change Tools" link.

CFR- 97: Tests/ directory are now excluded from local and hosted builds

### Bug Fixes

IM- 1710: Improved integration with mysql

IM- 1742: Chart view now respects permissions

IM- 1770: Default nameservers can now be removed

IM- 1771: Resolved issue where DNS was missing pointer generation for valid PTR records

IM- 1772: Generating API key no longer displays array bounds error

IM- 1774: Resolved an issue in IP Import preventing row edits after defining columns

IM- 1781: Update DNS links appear below hostname field for applicable resources with an assigned /32 block

**Hostname:**  
br1.6connect.org  
Forward DNS | Reverse DNS | Both

## ProVision 5.0.3

ProVision 5.0.3 is a minor release with bug fixes and improvements



### PHP Compatibility

Please note that ProVision 5.x requires php 5.5.+. For local installations, please upgrade php prior to installing the upgrade. Also ensure that the correct Sourceguardian php extension is loaded for the new version of php.

Contact 6connect at [info@6connect.com](mailto:info@6connect.com) to schedule a demo or get more information.

## Improvements

### Peering VRF:

VRF for Peering now supported for Cisco routers.

Enabling "VRF Support" in the Admin home page under "Peering Settings" will automatically add the VRF gadget to the router Section.

The VRF gadget, available on a router's Resource Entry page, allows you to add VRFs and ASN to a router

Once VRFs are set up for a router, the source ASNs for the associated VRFs will appear in the Source ASN dropdown when adding or editing a session for that router.

Currently only supports Cisco routers.

## Bug Fixes

IM- 1758: Block without VLAN now returns a "No VLAN found" message from a getVlan API call

IM- 1764: Updated IPAM API "Get" call so that the search parameter supports tag filtering

IM- 1768: Resolved an issue where users without IPAM-update permission were able to edit multiple blocks

## ProVision 5.0.2

ProVision 5.0.2 is a minor release with new features, bug fixes, and improvements



### PHP Compatibility

Please note that ProVision 5.x requires php 5.5.+. For local installations, please upgrade php prior to installing the upgrade. Also ensure that the correct Sourceguardian php extension is loaded for the new version of php.

Contact 6connect at [info@6connect.com](mailto:info@6connect.com) to schedule a demo or get more information.

### Bug Fixes/Improvements

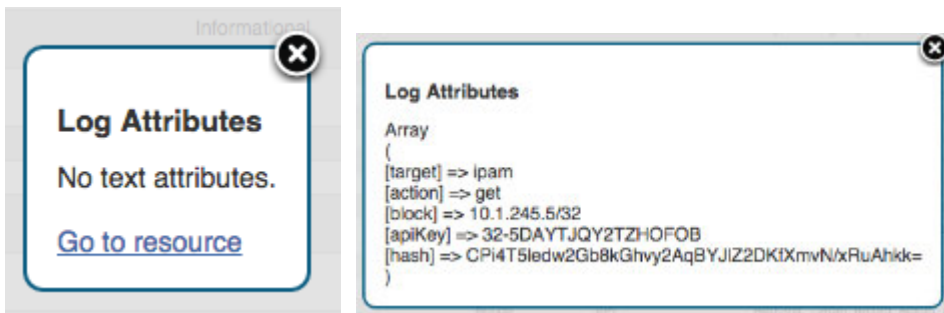
IM- 1706: API Direct Assign Tagging. See: [API Module- IPAM DirectAssign](#)

IM- 1705: API endpoint that will return utilization (assigned / available) across all aggregates based on a specific tag. See: [API Module - IPAM GetHostUtilization](#)

IM- 423: Improved logging for IPAM reporting actions

IM- 1252: Improved resource logging

Clicking on a resource log will provide additional log attributes (if available) and a link back to the Resource Entry page, API logs will display API detail:



IM- 1388: configTest improvements

IM- 1619: Additional information provided in logging: Log messages are more descriptive

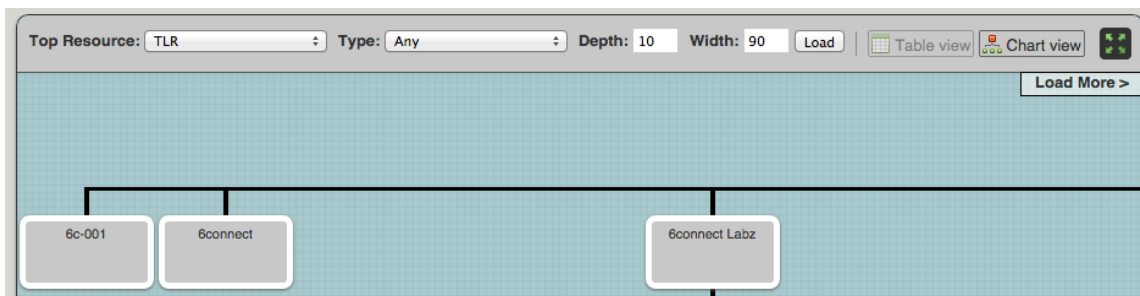
Assigned 2001:db4::/64 to San Salvador (2101) via Smart Assign

2001:db4::/40 (91471) updated

(API) 2001:db4::/36 split into all of its /40 blocks

IM- 1666: Expanded chart viewing width capabilities

Chart view now has an optional "Width" parameter, as well as a "Load More" button to continue viewing additional through the full tree width.



IM- 1675: Added API endpoint to do a recursive delete of resources (Parameter: &recursive=1 ). See: [API Module - Resource delete](#)

IM- 1688: Fixed password loading issue in Connection Configuration of the DHCP Gadget

IM- 1689: Resolved error in DHCP gadget after removing free line entries

IM- 1691: Resolved DHCP gadget duplicate error messages

IM- 1711: Blocks without parents can now be updated

IM- 1714: Resolved an issue with individual block assigning after a multi-assign

IM- 1716: Read only / Limited Permissions users can no longer add blocks via API

IM- 1717: Read only users can now access IPAM Manage dialog

IM- 1718: Read only users can no longer aggregate blocks in IPAM manage

IM- 1721: Resolved issues in 'IP Import- Import from .csv' with the "Generic Code" and "Allow Subassignments" fields.

IM- 1723: IPAM Gadget filter no longer duplicates mask entries in the filter drop down

## ProVision 5.0.1

ProVision 5.0.1 is a minor release with bug fixes and improvements



### PHP Compatibility

Please note that ProVision 5.x will require php 5.5.+. For local installations, please upgrade php prior to installing the upgrade. Also ensure that the correct Sourceguardian php extension is loaded for the new version of php.

Contact 6connect at [info@6connect.com](mailto:info@6connect.com) to schedule a demo or get more information.

### *Bug Fixes/Improvements*

IM- 1645: API attr\_key multiple attributes filter and hyphen conditions

IM- 1659: Assignment date for IPAM reporting is not inclusive

IM- 1660: Add Session dialog not pre-selecting exchange

IM- 1661: Peering - exception thrown for invalid exchange prefix

IM- 1662: Install - configBootstrap requires valid email for version list

IM- 1663: Install - configTest add mod\_deflate check and update version

IM- 1665: Subassignment indicator shows up on /32s (ipv4)

IM- 1678: Install - Fix install process with the secure directory

IM- 1679: Prevent deleting Resources which have children

IM- 1680: Updated "Getting Started" Link on Dashboard

IM- 1694: Updated default DNS sec resolver

IM- 1695: Session list doesn't refresh after clicking Update Session State

IM- 1696: Logging - Remote log attempting to log without host or ip

## ProVision 4.2.0

The 4.2.0 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 4.2.x Release Notes...](#)

- [ProVision 4.2.1](#)

### New Features

#### Peering Revamp

[Peering v2](#)

#### DHCP Integration into Resources

[DHCP Tab](#)

#### Bug Fixes

IM-1505 - IPAM Manage - Resource Filter improvements

IM-606 - Zone import optimizations for larger data sets



## ProVision 4.2.1

ProVision 4.2.1 is a minor release with some customer feature requests.

Contact 6connect at [info@6connect.com](mailto:info@6connect.com) to schedule a demo or get more information. Documentation (PDF version) - [Previous Documentation Versions](#)

### New Features

(CFR denotes customer requested)

CFR-86: Add "enable sub assignments" to IPAM gadget "edit" functions

1234 main road  
asdkjh dnm  
Khandahar, 24876 G2873

ARIN Info

10.48.0.0 - 10.63.255.255

☒ Allow sub assignments for this block

**Region** **DataCenter1** **VLAN**

Region

**Notes**

**Select Tags**

<input type="checkbox"/> Anycast	<input type="checkbox"/> BB	<input type="checkbox"/> BGP
<input type="checkbox"/> Cable	<input type="checkbox"/> Customer	<input type="checkbox"/> Dev
<input type="checkbox"/> DHCP	<input type="checkbox"/> DNS	<input type="checkbox"/> DSL
<input type="checkbox"/> Infrastructure	<input type="checkbox"/> Internal	<input type="checkbox"/> Loopback
<input type="checkbox"/> Management	<input type="checkbox"/> MPLS	<input type="checkbox"/> PTP
<input type="checkbox"/> Security	<input type="checkbox"/> Static	<input type="checkbox"/> Tower
<input type="checkbox"/> VMware	<input type="checkbox"/> VOIP	<input type="checkbox"/> VPN
<input type="checkbox"/> VRF	<input type="checkbox"/> Loopback A	<input type="checkbox"/> Loopback B

**Save**

Los Angeles, CA Anycast,PTP 2014-07-21

CFR-71: Require CRUD permissions for a user to delete DNS server -> zone links

IM-1261: Pagination for IPAM aggregates

Aggregate Blocks

All – IPv4 – IPv6 – DHCP

67.

67.21.0.0/20

67.221.241.0/24

67.221.255.0/24

Page: 1 2 3 4 5

65.39.196.0/23 – ARIN

Hosts

512

Blocks

33

Resources

3.12% assigned

3.12% allocated

Filter By Mask: All Masks

Go To Reporting

Available	496	96.87%	Available	31	93.93%	6connect Available	96.87%
Assigned	16	3.12%	Assigned	2	6.06%	6connect Labz	1.56%
Holding	0	0.00%	Holding	0	0.00%	Apple	1.56%
Allocated	16	3.12%	Allocated	2	6.06%		

IM-1496: DNS - Add zone UI revamp

Create a DNS Zone

Enter Zone Name

Select Customer...

IM-1184/IM-554: Option to allow imported zones to automatically add target server

Review Data

Please review the data for correctness. Invalid and ignored rows will be skipped.

View: All Valid Warnings Invalid Ignored Hide

Zone Name: nalinmk.com

Resource Holder: nalinmk.com

View Save

DNS Servers:	Enabled	Server Name	Master	Slave
<input type="checkbox"/>		dns.6connect.net (dns.6connect.net)	<input type="radio"/>	<input type="radio"/>
<input type="checkbox"/>		services1.tcp0.com (services1.tcp0.com)	<input type="radio"/>	<input type="radio"/>
<input type="checkbox"/>		ns1.sc2000.net (ns1.sc2000.net)	<input type="radio"/>	<input type="radio"/>
<input type="checkbox"/>		test.server (192.168.1.234)	<input type="radio"/>	<input type="radio"/>
<input checked="" type="checkbox"/>		6connect Test Server (208.39.106.184)	<input checked="" type="radio"/>	<input type="radio"/>
<input type="checkbox"/>		ns1.6clabs.com (ns1.6clabs.com)	<input type="radio"/>	<input type="radio"/>
<input type="checkbox"/>		ns2.6clabs.com (ns2.6clabs.com)	<input type="radio"/>	<input type="radio"/>

IM-1413: Reporting for IPAM, improve aggregate selection dropdown so it doesn't conflict with the IP Type checkboxes

Bug Fixes

IM-1500: Remove old remaining dns files from code base

IM-1154: Relabel "Transfer Type: SCP" to "ISC BIND"

IM-1126: Improved UTF-8 support for data imports

IM-944: Restructure Import Classes for more flexible data import formats

IM-323: logAction updates for global logging

IM-320: Cleanup of .txt file references/dependencies



## 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 4.1.23](#)
- [ProVision 4.1.22](#)
- [ProVision 4.1.21](#)
- [ProVision 4.1.20](#)
- [ProVision 4.1.19](#)
- [ProVision 4.1.18](#)
- [ProVision 4.1.17](#)
- [ProVision 4.1.16](#)
- [ProVision 4.1.15](#)
- [ProVision 4.1.14](#)
- [ProVision 4.1.13](#)
- [ProVision 4.1.12](#)
- [ProVision 4.1.11](#)
- [ProVision 4.1.10](#)
- [ProVision 4.1.9](#)
- [ProVision 4.1.8](#)
- [ProVision 4.1.7](#)
- [ProVision 4.1.6](#)
- [ProVision 4.1.5](#)
- [ProVision 4.1.4](#)
- [ProVision 4.1.3](#)
- [ProVision 4.1.2](#)
- [ProVision 4.1.1](#)

## 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"/>	<a href="#">198.11.8.0/24</a>	256		BOS		Customer	LIGHTTOWER-ISP-F	<a href="#">Has Children</a>
<input type="checkbox"/>	<a href="#">198.11.9.0/24</a>	256		BOS	13601-IP-BSTPMAME	Customer	LIGHTTOWER-ISP-F	<a href="#">Potamus Trading LLC</a>
<input type="checkbox"/>	<a href="#">198.11.10.0/24</a>	256		NYC		Customer	LIGHTTOWER-ISP-F	<a href="#">Has Children</a>
<input type="checkbox"/>	<a href="#">198.11.11.0/24</a>	256		NYC		Customer	LIGHTTOWER-ISP-F	<a href="#">Has Children</a>

## PowerDNS - MySQL Support

Full support for the MySQL PowerDNS backend. Zone updates are pushed out in easy SQL updates to the target server.

### Manage DNS Servers

Server:  New Server

Default:

Transfer Type:

Server Type:

SOA:

Username:

PowerDNS

ex: ns1.dns.6connect.r

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

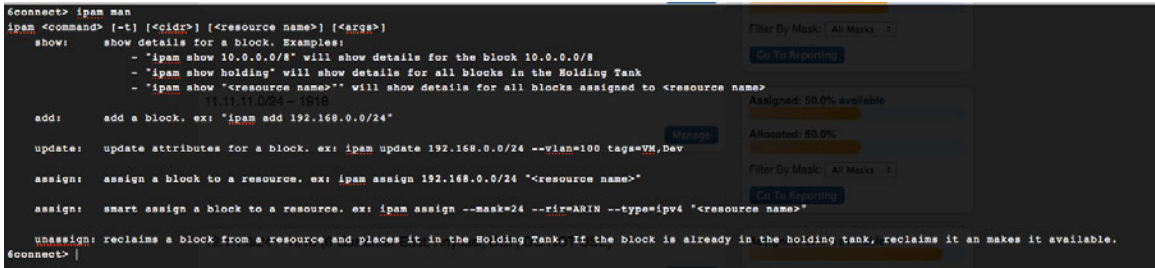
Update and delete tags globally from the primary tags list, just as nature intended.

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)

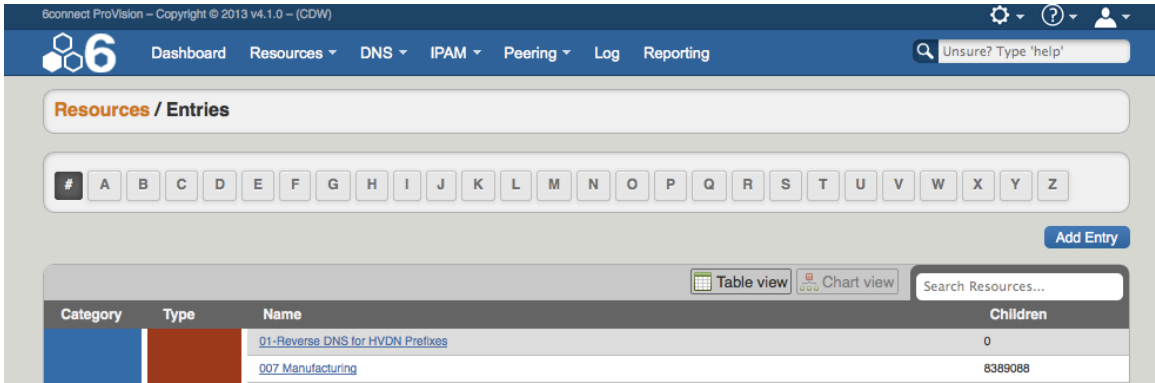
Hate clicking and have a general disdain for UI? This feature is for you! Just press ctrl+shift+s to try 6connect on the CLI with simple commands for quick no-frills interaction.



Feature Improvements

Resource UI Functionality

Easier navigation and button placement, alphabetical browse, data usage charts and more!

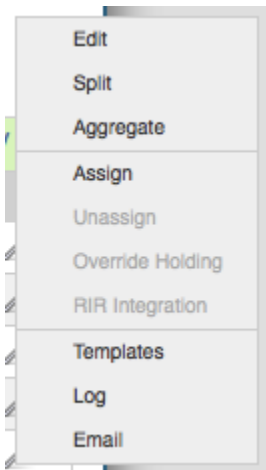


Header Redesign

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
Customer Handle	10.10.10.0	/30	PTP	1918
Generic Code	10.1.0.0	/24		ARIN
IP	10.2.0.0	/24	Development	1918
Mask	10.4.0.0	/24	Ticket 1515	1918
Net Handle				
Notes				
Org ID				
Region				

Default RIR:

**Step 3: Review and Import**  
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 4.1.23

### Bug Fixes

IM-1494 - IPAM - Incorrect SWIP parameters sent from IPAM gadget



## ProVision 4.1.22

### Bug Fixes/Feature Updates

IM-1493 - IPAM - Invalid LIR ID or entity handle error message when performing SWIP action

IM-1124 - DNS - PTR autogenerate - UPDATE

#### Reverse DNS

ProVision can be configured auto-generate missing IPv4 PTR records in reverse zones based on templates.

The default configuration values here can be overridden by local configuration values on individual zones.

At this time this feature is limited to zones which cover /24 sized blocks.

The variables '\$oct1', '\$oct2', '\$oct3', '\$oct4' may be used. They correspond to the first through fourth octets of the PTR's IP.

Generate missing IPv4  
PTR records by default ☒

PTR Host Template

PTR Value Template

## ProVision 4.1.21

### Improvements

#### Call Subnet Scans via the API

Subnet scan can now be issued and retrieved via the API. Examples of the two commands and links to documentation are listed below:

/api/v1/api.php?target=ipam&action=scanBlock&block=<cidr>

/api/v1/api.php?target=ipam&action=getScanResults&block=<cidr>

[API Module - IPAM](#)

#### Call Swap Functions via the API

[API Module - IPAM](#)

#### Delete Functionality for All Blocks

Aggregates can now be deleted regardless of state of sub-blocks. An aggregate can be deleted even if sub-blocks or hosts are assigned to resources.

11.1.2.0/24 – ARIN

Manage

Hosts	256		Blocks	9		Resources	3
Available	191	74.60%	Available	7	77.77%	LargerTest	74.60%
Assigned	65	25.39%	Assigned	2	22.22%	ABCAM	25.00%
Holding	0	0.00%	Holding	0	0.00%	1stdibs.com	0.39%
Allocated	256	100.00%	Allocated	9	100.00%		

#### IPAM Aggregate Statistics

All IPAM aggregates now display statistics on how they are assigned to resource on the main IPAM screen.

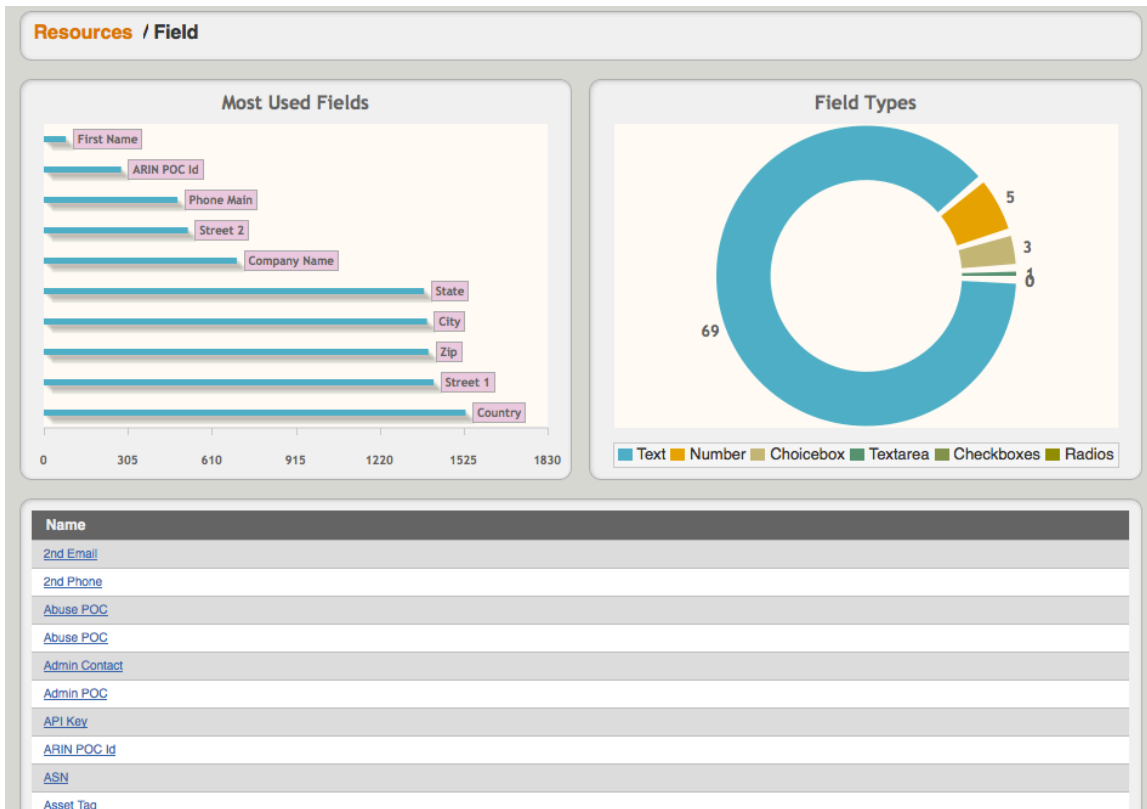
11.1.2.0/24 – ARIN

Manage

Hosts	256		Blocks	9		Resources	3
Available	191	74.60%	Available	7	77.77%	LargerTest	74.60%
Assigned	65	25.39%	Assigned	2	22.22%	ABCAM	25.00%
Holding	0	0.00%	Holding	0	0.00%	1stdibs.com	0.39%
Allocated	256	100.00%	Allocated	9	100.00%		

#### Resource Field Management

A new easy to read display of field usage statistics for usage of specific resource field names, as well as usage of field types, and a list of all available fields for improved resource field management.



## Bug Fixes

- CFR-87 - IPAM - Consistently display Region name across IPAM management and gadgets
- IM-1467 - Resource - Unable to enter Canadian Provinces
- IM-1476 - IPAM - Smart Assign generic code parameter being ignored
- IM-1477 - IPAM - Remove ability to dis-allow sub-assignments on blocks with children
- IM-1482 - Resource - Can't edit fields in Section in Chrome/Safari
- IM-1484 - Resource - Fix verbiage around Types and Sections, change all to Sections
- IM-1486 - Resource - Remove system TLR field from breadcrumb display

## ProVision 4.1.20

### *Bug Fixes*

IM-1463 - DNS - Create zones drop down pulling incorrect resource list (DHCP)

IM-1449 - IPAM - Add loading/waiting indicator when waiting for swip/allocate transaction

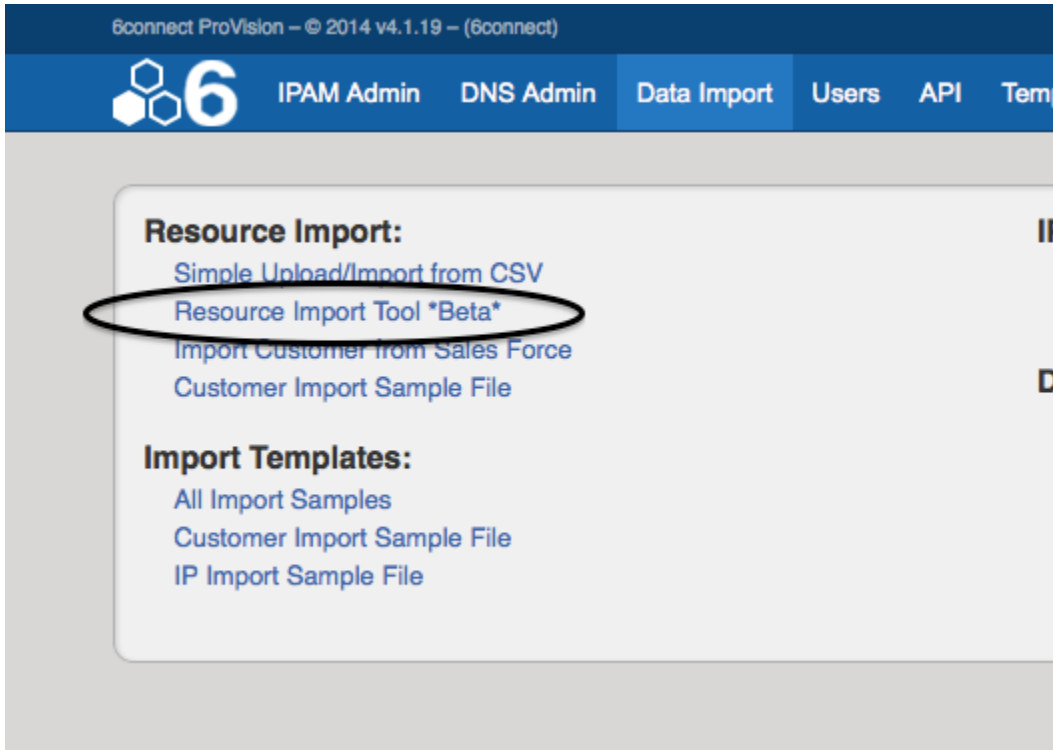
IM-1332 - Import - File type issues, can't import files that are not Windows csv

## ProVision 4.1.19

### Beta Features

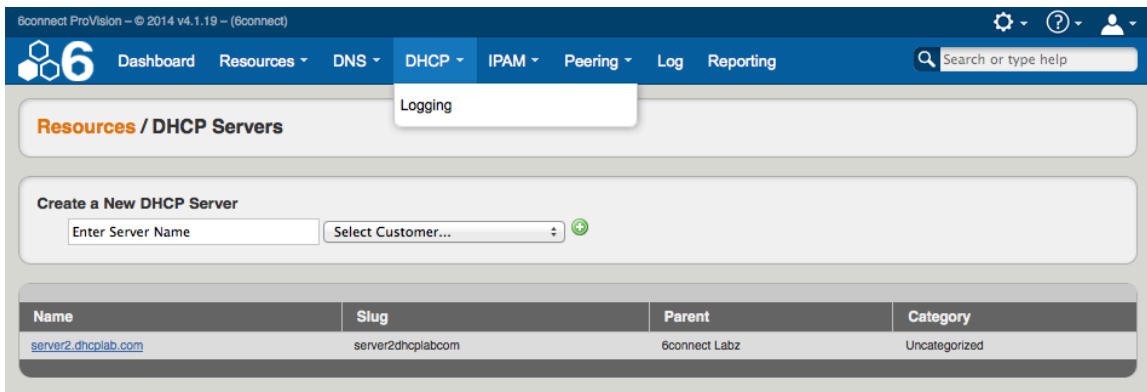
#### Resource Importer

A new stand alone application to import complex resource sets. Give it a test run! Go to Admin -> Data Import and click on the Link for "Resource Import Tool Beta". As always, please send feedback to [support@6connect.com](mailto:support@6connect.com).











#### DHCP Management

New DHCP Tab - See all your DHCP server entries in a single place. The usual search functions work as well.



DHCP Management (Now with more Resources!) - We have integrated DHCP functions into a "Gadget". This means you can assign DHCP functionality to a Server of your choice and manage it just like any other resource in ProVision.

<

+ DHCP Pools	
Linked Pools	
TestHost1 [Host]	
TestHost2 [Host]	
TestHost3 [Host]	
TestHost4 [Host]	
TestSub1 [Subnet]	
TestSub2 [Subnet]	
TestSub3 [Subnet]	
TestSub4 [Subnet]	

Create a new

Subnet

Subnet Name:

(ex: Lab #1)

New IP Assignment:

Smart

IPv4

Mask

RIR

Region

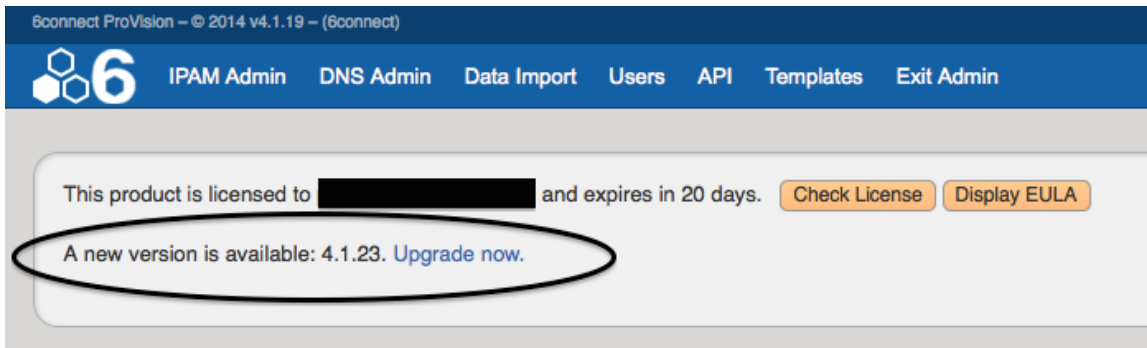
Free Lines:

No lines saved.

Add a New Line:

Add

Add Pool



### ARIN Swip Error Messages

ARIN swip error messages now show the full detail of the error in the GUI.

### Bug Fixes

IM-1453 - Resources - Unable to remove existing fields from field collection

IM-1454 - IPAM - IPv6 radio button typo

IM-1459 - IPAM - Manage screen does not auto scroll/highlight block after edit

## ProVision 4.1.18

### Beta Features

#### Resource Importer

A new stand alone application to import complex resource sets. Give it a test run!

### Improvements

Resource Importer - Add default name server columns to import

Resource - Add count of zone to Customer section display

### Bug Fixes

IM-1277 - Dashboard - Resource zone assignments display even with no data

IM-1291 - Resource - Ability to move system resources via Resource Chart View

IM-1292 - Resource - Ability to move TLR anywhere in tree - subsequent crash

IM-1328 - Admin - Radius dictionary and LDAP schema links broken

IM-1442 - Resource - Redirect non-existent categories to uncategorized category

IM-1446 - IPAM - Add Swip error message to error displayed in UI on failure



## ProVision 4.1.17

### Beta Features

#### Resource Importer

A new stand alone application to import complex resource sets. Screen shots TBA.

### Improvements

#### PowerDNS Server Connection Test

Test and diagnose connectivity issues to a PowerDNS server.

**Manage DNS Servers**

Server:  **New Server**

Display Name:

FQDN or IP:  ex: ns1.dns.6connect.net or 216.239.32.10

Default:

Transfer Type:

Server Type:

Backend Type:

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

Username:

Password:

DB Username:

DB Password:

DB Port:

DB Name:

**Test Config** **Update Server** **Delete Server**

Connected socket to 208.39.104.106:3306 but MySQL connection failed: could not find driver

### Bug Fixes

IM-1411 - DNS - Add "Test Config" option for PowerDNS servers in DNS Admin

IM-1430 - IPAM - Direct assign using block parameter throws error

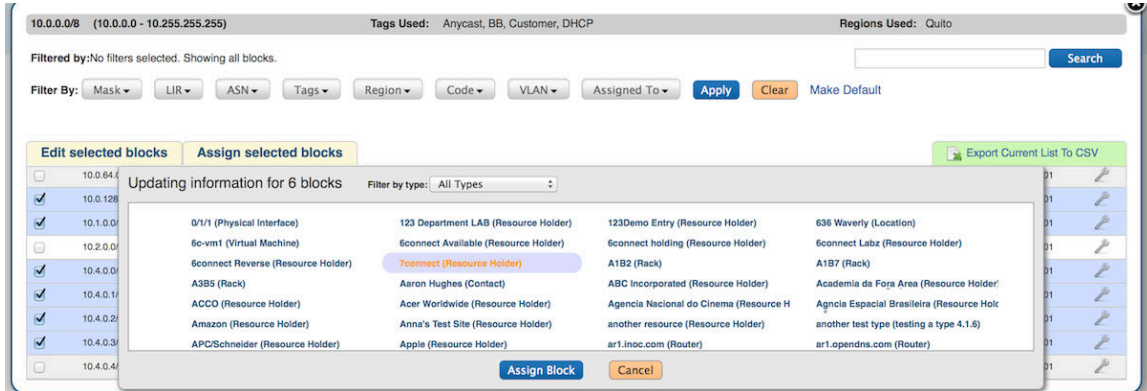
IM-1431 - Setup - configBootstrap allows an empty registration email

## ProVision 4.1.16

### New Features

#### IPAM - Multi-assign

Assign multiple IP addresses to a resource via a single API call, or just a few clicks in the user interface! When multiple IP addresses are selected in the UI, a new tab to assign all at once will appear.



### Bug Fixes

IM-1418 - IPAM - De-Swip estimated time too high

IM-1426 -IPAM - Filters resetting to default settings after multiple edits

IM-1427 - Resource - Chart view fatal error

IM-1428 - IPAM - Swip timer counting negative

## 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

## 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 4.1.13

### 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
-----
82 - 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 4.1.12

### 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

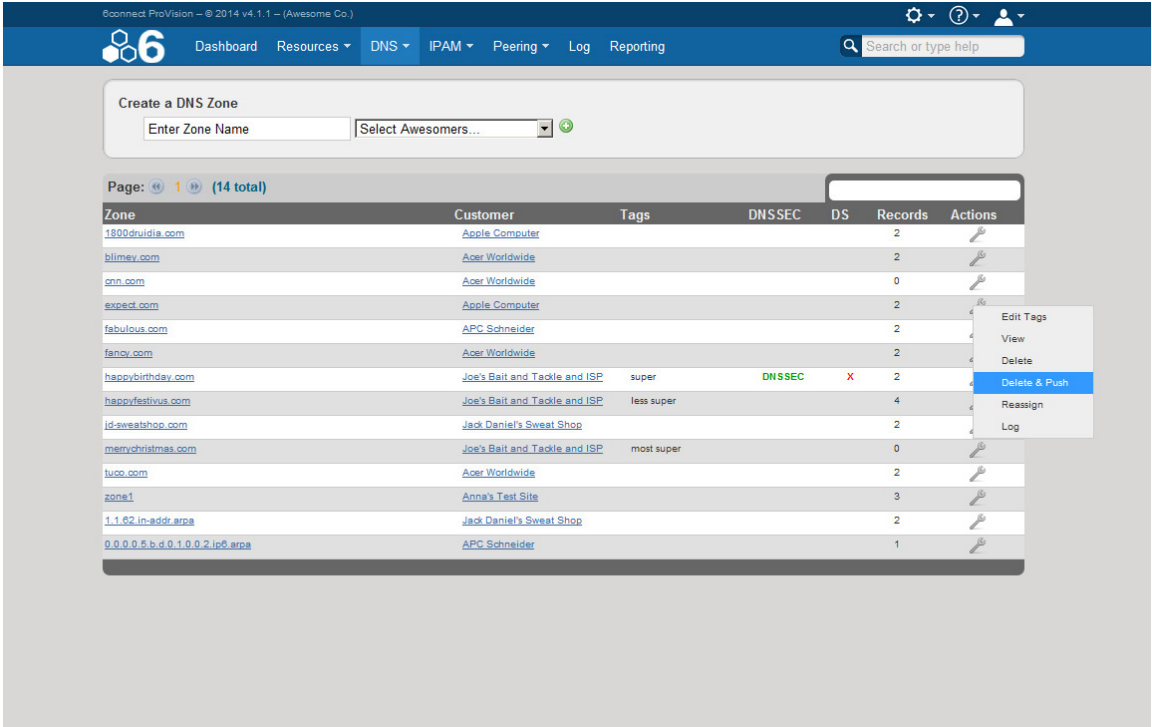
IM-1039 - Resource - ZIP/Postal code field missing in edit dialog

# ProVision 4.1.11

## New Features

### DNS Deletion Queue

Addition of a deletion queue for zones for flexible removal of zones from DNS Servers.



## Bug Fixes

IM-1344 to IM-1351 - All bug fixes related to a bad version of bootstrap.

## 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 4.1.9

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:  **New 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

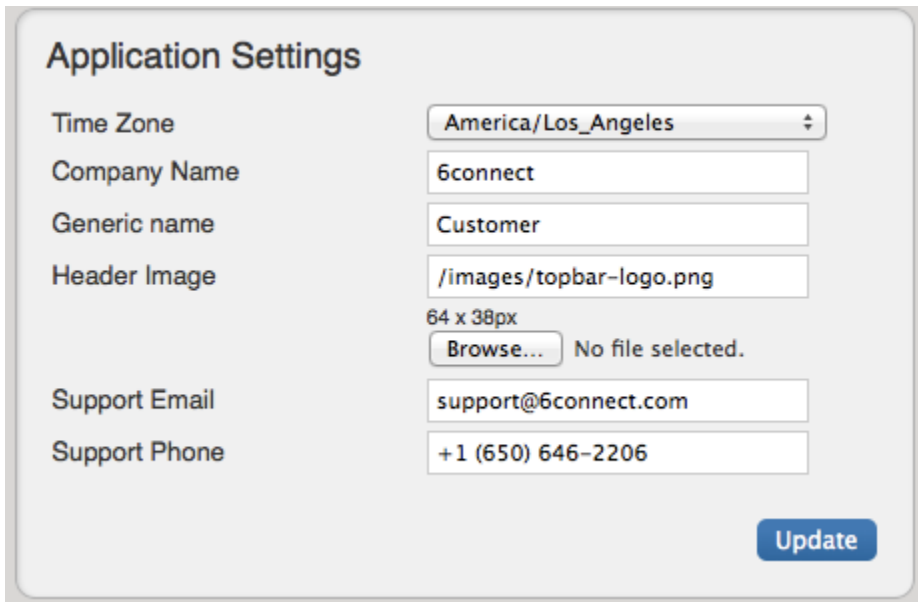
**IM-1142** - Center command results given by assistant

## ProVision 4.1.8

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.



The screenshot shows the 'Application Settings' form. It includes fields for Time Zone (set to 'America/Los\_Angeles'), Company Name ('6connect'), Generic name ('Customer'), Header Image (set to '/images/topbar-logo.png' with a 'Browse...' button and 'No file selected.' text), Support Email ('support@6connect.com'), and Support Phone ('+1 (650) 646-2206'). An 'Update' button is located at the bottom right.

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 4.1.7

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 4.1.6

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 4.1.5

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 4.1.4

The 4.1.4 is bug fix release only.

### Bug Fixes

IM-1270 - Fix table name in 4.1.0 upgrade script

## ProVision 4.1.3

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 4.1.2

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 4.1.1

The 4.1.1 is bug fix release only.

### Bug Fixes

IM-1193 - Path to zones incorrect in 6connect conf file

## ProVision 4.0.0

The 4.0.0 release is a major release that contains many new features and improvements. The improvements include: a complete revamp of the Resource (Asset) system, an LIR/RIR management system to manage multiple ASNs and RIR credential sets, a new permissions structure with user groups and Resource based access, global search auto-complete, and an improved GUI setup (available since 3.9.3).

### New Features

#### *RIR/LIR Manager*

If you have multiple ASNs, LIRs, or RPSL Maintainer Objects to manage - this is the feature for you. You can now enter an unlimited amount of ARIN or RIPE credentials and select which set of credentials to use when updating an IP block. Additional detailed error reporting has been added to provide specific information in the event of an update failure.

The credentials for all LIR's under a particular RIR are displayed when the RIR Integration icon is clicked, allowing quick and easy switches of blocks between different organizations or maintainers.

#### *Permissions*

Control and set access to your IP data, DNS, and Resources exactly how you want it. Full CRUD permissions for each section of ProVision can now be controlled on per Resource basis through user groups. Previously, permissions were based on sections of the tool: IPAM, DNS, Peering, Resources, etc. and applied globally. Permissions for each section are now attached to individual Resources. Additionally, user groups have been added to simplify permission assignment. Full CRUD permissions for each section of ProVision can now be controlled on per Resource basis through user groups.

The new permission structure has completely replaced the Customer Portal functionality in versions previous to 4.0.0. Customer Portal users will be migrated to standard directory of users that have permissions only on their specific customer resource.

Note: The Radius and LDAP integrations have changed significantly with the addition of user groups. The configurations for 3.9.3 and prior will NOT work with 4.0.

For more information, see the documentation: <http://docs.6connect.com/display/DOC/Manage+Users>

#### *Global Search Auto-complete*

Find what you are looking for quicker and easier. Global search will now provide a drop down of auto-complete answers, and the section that contains the relevant results.

#### *Resource (Asset) System*

ProVision has a new ultra flexible asset management system that looks good while doing it. The Resource system has a smoother look and feel with pagination, search, and quicker, more detailed methods for creating custom fields and data types. Both Categories and types have been added to help better organize Resources.

And if you are a fan of pretty things, there is a new graphical Resource view that displays the hierarchical layout of Resources, and allows for drag/drop to reorganize Resources.

#### *Documentation*

We have moved to a new and improved documentation system at: <http://docs.6connect.com/>. Help documents are now fully indexed and searchable. Video tutorials are being added soon!

## Improvements

No more changing text configuration files! 4.0.0 now has a GUI setup wizard to check the system configuration for errors, and load the configuration files and database from a few simple fields.

## Minor Improvements

IPAM - generic code field added to API Smart Assign call. (CFR-44)  
Authentication - set default login type via constant. (CFR-43)  
Reporting - Add reporting functionality on Top Level IP Aggregates. (CFR-42)  
DNS - List zones alphabetically, and then numerically by IP octet/nibble.

## Bug Fixes

IM-812 - Confirming the Org ID of IANA on welcome page produces blank screen.  
IM-922 - Successful Radius login rejected.  
IM-455 - Do not allow duplicate zone creation.  
IM-833 - Fix to save record on Zone View when hitting enter key.  
IM-937 - Fix to close record using view icon on Zone View.  
IM-941 - Fix for global search timeouts.  
IM-940 - Updated import CSV header type checking to accommodate .csv files generated from a variety of different sources.

## Minor Release 4.0.1

IM-1079 - IPs in Holding can be assigned via Browse Assign.  
IM-1088 - Daemon user visible in accounts section.

## Minor Release 4.0.2

IM-1099 - IPs in Holding can't be manually over ride reclaimed.  
IM-1100 - Latest import code not checked into 4.0.0 branch.

## Minor Release 4.0.3

IM-1102 - Contact info gadget information does not save on first click.  
IM-1101 - IP block scan Found hosts number does not refresh on completion.

## Minor Release 4.0.4

IM-1119 - Odd session behavior/results with api calls.

## Minor Release 4.0.5

IM-1120 - Generate forward/reverse DNS for resources not working.  
IM-1125 - Session Bug - User timeout/logout allows API request as user with no permissions  
IM-1131 - Warning in search.php line 92  
IM-1132 - Customer import from .csv field count issue

## Minor Release 4.0.6

Maintenance release - Peering code integration

## Minor Release 4.0.7

IM-1136 - ContactInfo gadget reading and saving incorrect field name for phone and fax

## ProVision 3.9.0

Released March 2013

### Network Discovery Scanlet (v1)

- IPv4 scanning agent for network discovery
- Discovers and imports IP, Ports and MAC Address as devices

### ProVision - IPAM

- IP Sub-assignments
- IP Templates

### ProVision - DNS

- DNS Views (Split Horizon Support): Includes ACL manager!
- Improved DNS Master/Slave support
- Secure64 DNS Integration (update)

### ProVision - Assets

- Browse, Direct, and Smart Assign added to Assets
- Asset UI cleanup (multiple)

### ProVision - Platform

- Logging fixed when entering duplicate blocks
- Propel ORM removal

## ProVision 3.9.3

### Overview

The 3.9.3 minor release provides a few new DNS features.

### *New Features*

#### Per Server DNS View Support

Enable, disable, and manage views on a server level. This enables every server to have it's own unique combination of views implemented, or none at all.

#### Secure64 DNS View Support

Support for views with the Secure64 DNS system.

### *Bug Fixes*

None.

## ProVision 3.9.2

### Overview

The 3.9.2 minor release provides a few new DNS features, additions to the API request generator, minor IPAM changes, and minor bug fixes.

### *New Features*

#### Secure64 DNS Integration

Secure64 Master/Slave integration and DNSSEC integration.

#### API Request Generator - DNS Utilities

DNS Zones, DNS Records, DNS Servers, Server-Zone Linkage, and Name Server API sections have been added to the API request generator.

### *Changes and Updates*

#### SWIP Information Display

When a block is swipped, it is now indicated by a green check mark next to the RIR name on the IP manage screen. Hovering over this check reveals the swip details.

### *Bug Fixes*

Fix for IM-807 - Missing email address and exchange information on some peering request forms.

Fix for IM-808 - Scan results not displaying on IPAM manage screen.

Fix for IM-809 - Setting incorrect path for imports temp directory on local installs.



## ProVision 3.9.1

### Overview

The 3.9.1 minor release provides a few new IPAM features and improved data visualization. It also contains some changes and improvements DNS Master/Slave relationships, and the file structure created on BIND servers during file push.

### *New Features*

#### IPAM Bulk Edit

There are now checkboxes by each IP block on the IP block management screen. When you select more than one check box, the option to edit multiple blocks appears at the top left of the IP list. Clicking "Edit selected blocks" will bring up the standard edit menu and all selected blocks will be updated to the properties chosen.

#### IPAM Export to .csv

The "Export Current List to .csv link is located at the upper right of the Manage list. When clicked it exports all blocks displayed on the screen to a .csv file. You can export select sub-sets of blocks simply by filtering or searching the list.

#### DNS Master/Slave Relationships

One the zone edit page, there is now the ability to push a zone out to a server as a master or a slave to accommodate complex and mixed master/slave environments.

#### Bulk Zone Server Assignments

On the DNS Admin tab, the second second provides the ability to search for zones, then perform a bulk assignment of those zones to a specific server.

#### IP Assignments - Hierarchical View

On the IPAM manage screen, blocks that allow sub-assignments, and have at least one sub-assigned block become a link that expands to show a hierarchical listing of space within the block and what resources the space is assigned to.

### *Changes and Updates*

#### BIND Directory Structure

The directory structure for BIND has been changed so that all zone files now go into a sub folder under ./6connectGeneric that is the first character of the zone files name, either 0-9 or a-z.

#### RIPE IP Look-up

IP based look-up on the welcome screen for RIPE blocks is now using an improved reverse lookup on mnt-ref and displays all discovered organizations associated with an IP, allowing the user to identify the correct organization.

### *Bug Fixes*

Fix for missing time zone, state, and country information in the admin section and customer address section. This would have only impacted new turn ups between 3/14/2013 and present.

## ProVision 3.8.0

Launched in December 2012

### ProVision - IPAM

- IP Import Wizard
- IP Templates/Splitting

### ProVision - DNS

- Updated Zone Editor
- Zone History and Versioning (with rollback)

### ProVision - Assets

- Migration to more flexible container structure

## ProVision 3.7.0

Launched in September 2012

### ProVision - IPAM

- Multiple UI updates, including new "Aggregate" view

### ProVision - DNS

- PowerDNS support (v1)
- Zone editor refresh
- Zonelist page UI update
- Selective zone push (single zone, bulk push, etc.)
- Multiple DNS Server Support
- RFC based zone validation

### ProVision - Assets

- Complete refresh, unlimited Asset Types and Custom Fields

### Network Automation - Peering

- Complete UI refresh of Communication Manager and Session Manager

### ProVision - Platform

- LDAP Support
- Revamped Tree View with real-time search

## ProVision 3.6.0

Launched in August 2012

### DNS

- Asynchronous zone push

### DHCP

- ISC dhcpd support
- Support for common DHCP features
- DHCP configuration pushes and execution of custom commands
- Support for custom DHCP configurations

### Portal

- Integrated UI updates from Resource Holder page

### Platform

- Revamped Resource Holder page
- Updated Smart Assign function
- Multiple data visualization updates

## ProVision 3.5.0

Launched in June 2012

### API

- IP block history lookup
- Support for lookup by Custom field

### IPAM

- tag editor refresh
- ARIN Reallocation support (via REST)
- RIPE block lookup for aggregate importing (welcome screen only)

### DNS

- Zone template editor/builder
- Zone editor refresh
- Zone importer refresh
- DNS Admin UI updates including pre/post commands per server
- DynECT integration (v1) for Dyn.com DNS customers

### Platform

- Customer/Delegate Portal - version 1 (includes permissions for IPAM/DNS functions)
- Real-time streaming syslog function (incl JSON support)
- Admin Tab UI updates (multiple)
- Reporting option (filter by Tag, Custom Field, Region, etc.)
- Timeserver refresh

## ProVision 3.4.0

Launched in April 2012

### API

- DNS functionality now accessible
- Log functionality
- php SDK v1

### IPAM

- IPv6 Direct Assign Refresh
- Prep for UI refresh

### DNS

- Multi-server support for zone export/push
- DNS Admin UI refresh

### Platform

- Authentication revamp (login, logging, etc.)
- Sitelist UI update
- Paging UI update
- Salesforce integration (version 1)
- Prep for RIR/LIR support

## ProVision 3.3.0

Launched in March 2012

### API

- IPAM functionality vis RESTful web service (including SWIP)
- Add/Remove Assignee (all fields)
- Add/Remove Assignee contacts (all fields)

### IPAM

- View/Edit Devices from detail screen
- SWIP data lookup (ARIN only)

### DNS

- SCP support for zone exports
- Secure64 Integration (version 1)

### Peering

- Support for multiple locations within an Exchange

### Platform

- Dashboard refresh
- Reporting Tab refresh
- Automated IP Import wizard
- Enhanced RADIUS VSA support
- RESTful API integration (application side)
- Multiple navigation updates (tab/subtab)
- FreeBSD Virtual Appliance released

# ProVision 3.2.0

Launched in February 2012

## IPAM

- RPI RPSL API integration
- SWIP as "private" for ARIN region
- Parent/Child relationships for Sites (multiple levels permitted)
- Revamped Site Editing screen
- Multiple Tag support for assignments
- Filter by NetMask option

## DNS

- Zone editor enhancements (multiple)
- Bulk DNS zone editor
- DNS Zone download (ZIP format)
- DNS zone file incrementer
- DNS Zone file pre-population

## Peering

- Integrated Communication Manager unto UI

## Asset and Device Tracking

- Rack level diagramming from Asset data
- Device tracking (IP, hostname, MAC address, VLAN, etc.)
- Generate DNS records from Device data

## Platform

- Revamped global search interface
- Assistant (integrated network tools and search)
- Additional Filtering enhancements for views
- Additional Header sorting enhancements
- Double byte support
- User Management enhancements
- Integrated documentation (HTML/PDF)
- Revamped "tree view" functionality
- Granular logging with filtering and highlighting
- New "Reporting" Tab for statistics, data export and RIR reporting



## ProVision 3.0.0

Launched in Fall 2011

### IPAM

- ARIN API integration
- Enhanced block search functionality
- v4 network host scanning

### DNS

- Zone searching and filtering enhancements
- Note storage and search per zone record entry

### Asset and Device Tracking

- Create/Delete and Modify Assets (beta)

### Platform

- Time Machine functionality for Zones, IP blocks and User Actions
- Local authentication engine
- Enhanced Global search functionality
- Enhanced Tree View functionality

## ProVision 2.5

Launched in Summer 2011

### IPAM

- Added support from /16 to /128 for IPv6
- Custom field per Block (accessed via Admin Tab)
- Custom field per Site (accessed via Admin Tab)
- Detail page retains IP settings for multiple allocations
- Added VLAN tracking per block
- Added customizable email templates to Admin Tab
- Direct Assign auto subnets for v4 and v6
- Track blocks by Region

### DNS

- DNSSEC support including DS key generation
- Added paste in forward IPv6 address to resolve IPv6 PTR while editing reverse zones

### Peering

- Production release (version 1)
- Support for Cisco, Juniper, Foundry, Brocade

### Platform

- Data importer (beta) - including DNS zones
- Updated Admin Tab functionality and layout
- Added permissions checks for working files/directories
- php 5.3.6 compatibility
- browser compatibility updates (various)

## Coming Soon

### Future Releases Roadmap

#### 5.1.0 Release - Q1 2015

- Tag unification and UI clean up - this is unifying the tagging system for IP and DNS
- VLAN system including VLAN management, VLAN IP view, and probably a VLAN gadget
- Contact Manager - this cleans up some advanced features for RIPE/ARIN integration
- Refactor of SWIP for use with contact manager (see previous)
- LIR refactor to move it to it's own resource type and out of a section (see previous)
- Search - select resource fields to be displayed in search results (UI enhancement)
- Resource templates - (creating multiple resources from a template, etc.)
- Peering Module - Cisco VRF support
- Peering Module - Juniper logical systems support
- IPAM Gadget option - select Tag Interpretation default behavior (Strict, etc.)
- Task Scheduler
- Auto-merge of IP aggregates on unassign
- Refresh auto generation of forward/reverse DNS for Resource with FQDN
- DNS Admin UI clean up
- DNS Audit Tools (UI clean up)

Above is a list of road mapped features for our next release. The roadmap is highly flexible, and subject to change at any time (even during the release). We receive a regular volume of feature requests, and will often change the roadmap to include customer feature requests, rather than a scheduled roadmap feature or improvement.