Platform Documentation

ProVision

Application Version 5.2.2

Covering:

- Installation Guide
- Getting Started
- User Guide
- Admin Guide
- Developer Tools
- Help & Support

For additional information, please visit http://docs.6connect.com or contact 6connect at support@6connect.com

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.

Table of Contents

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 10+(IE 8+ may have some display issues)

Backup and Redundancy

Backup Schedule

Both local and hosted instances are provided with a default Scheduler task to perform a backup every 24 hours, with a 1 month retention policy.

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.

> Previous versions of this installation documentation contained a dedicated page, "6connect Local Software Installation" for the ProVision-only section of the installation process. This information is now available under each OS-specific installation page as the last section ("Install 6connect ProVision Software:").

Table of Contents

System Requirements

ProVision System Requirements

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 5.6.x: {+}http://php.net/downloads.php
- MySQL 5.6.x: {+}http://www.mysql.com/downloads/

MySQL Triggers
6connect does not support custom MySQL triggers at this time - please email 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
Both local and hosted instances are provided with a default Scheduler task to perform a backup every 24 hours, with a 1 month retention policy.

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

ProVision Local Installation For CentOS 6

CentOS 6

Before You Begin

Ensure that System Requirements have been met prior to proceeding with the CentOS Configuration Guide.

Install Requirements

1) Upgrade your current packages

Upgrade your current packages.

```
yum update
```

2) Install Required Packages

PHP

All installations of ProVision require at least PHP 5.6 (and related extensions). CentOS 6 comes with PHP 5.3 by default. You can either add a repository which provides PHP 5.6 or install PHP manually.

The Webtatic and Remi repos both have versions of PHP which are newer than those in the official repos. For this example, we'll be using Webtatic

Add the repository:

```
rpm -Uvh https://mirror.webtatic.com/yum/el6/latest.rpm
```

Update:

```
yum update
```

Install:

Click here to expand...

PHP5 / Apache2 / extensions

- httpd
- php56w
- php56w-opcache

- php56w-mysqlnd
- php56w-pdo
- php56w-ldap
- php56w-pecl-memcache
- php56w-bcmath
- php56w-devel
- php56w-pear
- php56w-cli

Development tools for pecl / additional system packages:

- curl
- openssl
- memcached
- mod_ssl

```
yum install httpd php56w php56w-opcache php56w-mysqlnd php56w-pdo php56w-ldap
php56w-pecl-memcache php56w-bcmath php56w-devel php56w-pear php56w-cli curl openssl
memcached mod_ssl
```

MySQL

Install MySQL to use a local database.
MySQL is included with most CentOS installs, check for it with:

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

The default MySqL version included with most CentOS installs will need to be upgraded to the latest version:

```
rpm -Uvh https://mirror.webtatic.com/yum/el6/latest.rpm
```

If you have an existing installation, you can replace it with:

```
yum install mysql.`uname -i` yum-plugin-replace
yum replace mysql --replace-with mysql56w
```

For a fresh install:

```
   yum install mysql56w mysql56w-server
```

Then, re-start and configure.

```
   service mysqld start      chkconfig mysqld on
```

If a new install, set the MySQL root password:

```
    /usr/bin/mysqladmin -u root password 'new-password'
```

For an existing install / upgrade, you will need to upgrade the existing tables after the restart.

```
mysql_upgrade -u root -p
```

This will issue a password prompt for the user. If you don't have a root user password, remove the "-p".

DNS and Additional Utilities

5. Install the DNS and other remaining utilities:

Click here to expand...

- curl

- openssl
- nmap
- bind-utils
- bind
- expect

```
yum install curl openssl nmap bind-utils bind expect
```

DNSSEC-Tools

Install and Compile DNSSEC-Tools

```
yum groupinstall 'Development Tools'
yum install openssl-devel perl-devel perl-CPAN
cd /usr/src
wget https://www.dnssec-tools.org/download/dnssec-tools-2.1.tar.gz
tar -xzf dnssec-tools-2.1.tar.gz
./configure
make
make install
```

3) Configuring the requirements:

SSL

Self signed certificates in CentOS 6 by default have been already installed.

If you want to change it, follow the steps below:

> Note: For production install, it is **HIGHLY RECOMMENDED** to use organization signed certs

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:
(shown using the vi editor, though you may use the editor of your choice)

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

Find the lines that start with SSHCertificateFile and change them to be like:

```
SSLCertificateFile /etc/pki/tls/certs/ca.crt
SSLCertificateKeyFile /etc/pki/tls/private/ca.key
```

Then restart

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

Add 443 virtual hosts as needed in httpd.conf.

Apache

Allow overwrites in the apache vhosts

```
sed -i 's/AllowOverride None/AllowOverride All/g' /etc/httpd/conf/httpd.conf
```

Start Apache and make it to start on boot

```
   chkconfig httpd on     service httpd start
```

> **mod rewrite REQUIRED**
> Please note that mod_rewrite is required! If it is not enabled in Apache, key elements will not work as expected.

MySQL

Set the MySQL Configuration:

```
mysql -p -e "SET GLOBAL sql_mode='NO_ENGINE_SUBSTITUTION';SET SESSION
sql_mode='NO_ENGINE_SUBSTITUTION';"
```

then enter the MySQL root password when prompted.

4) Optional configurations:

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:

If you are going to run iptables, click here

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
```

Note
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.

Radius (Optional)

This section only needs to be followed if the customer will be using Radius for authentication.

If you are going to use radius authentication, click here.

Install radius module:

```
   pecl install radius     echo extension=radius.so > /etc/php.d/radius.ini
```

SSH

Install ssh module:

```
    yum install libssh2-devel     pecl install -f ssh2     echo extension=ssh2.so >
/etc/php.d/ssh2.ini
```

4) Install 6connect ProVision Software:

1. Remove the current contents in the ProVision web folder (currently the www root) and after extract the archive contents (where 5.x.x is the version number for the build) :

```
tar -xf productionBuild-5.x.x-php5.6.tar -C /var/www/html
```

2. Change the permissions to be the web user permissions

   chown -R apache.apache /var/www/html

3. Go to http://<web root>/install/configTest.php. Follow the provided instructions, correcting any configuration errors if they occur. Once all steps are completed, you are ready to use your ProVision instance!

ProVision Local Installation For CentOS 7

CentOS 7

Before You Begin

Ensure that System Requirements have been met prior to proceeding with the CentOS Configuration Guide.

Install Requirements

1) Upgrade your current packages

Upgrade your current packages.

```
yum update
```

2) Install Required Packages

PHP

All installations of ProVision require at least PHP 5.6 (and related extensions). CentOS 7 comes with PHP 5.4 by default. You can either add a repository which provides PHP 5.6 or install PHP manually.

The Webtatic and Remi repos both have versions of PHP which are newer than those in the official repos. For this example, we'll be using Webtatic

Add the repository:

```
rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
  rpm -Uvh https://mirror.webtatic.com/yum/el7/webtatic-release.rpm
```

Install PHP and extensions:

Click here for the list of extensions...

PHP5 / Apache2 / extensions

- httpd
- php56w
- php56w-opcache
- php56w-mysqlnd
- php56w-pdo
- php56w-ldap
- php56w-pecl-memcache
- php56w-bcmath
- php56w-devel

- php56w-pear
- php56w-cli

Development tools for pecl / additional system packages:

- curl
- openssl
- memcached
- mod_ssl

```
yum install httpd php56w php56w-opcache php56w-mysqlnd php56w-pdo php56w-ldap
php56w-pecl-memcache php56w-bcmath php56w-devel php56w-pear php56w-cli curl openssl
memcached mod_ssl
```

MySQL

Install MySQL/ MariaDB to use a local database.
MySQL is included with most CentOS installs, check for it with:

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

If MySQL is not installed, click here...

If it is not installed:

```
rpm -Uvh http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm
yum install mysql-server
service mysqld start
chkconfig mysqld on
```

Set the MySQL root password:

```
/usr/bin/mysqladmin -u root password 'new-password'
```

To install MariaDB instead of MySQL, click here...

If you prefer to install MariaDB:

```
yum install mariadb-server mariadb
systemctl start mariadb
```

Set the root password, as currently it is not set, just hit ENTER on the current password:

```
mysql_secure_installation
```

Set so that it starts on boot:

```
systemctl enable mariadb.service
```

DNS and Additional Utilities

5. Install the DNS and other remaining utilities:

Click here for the list of utilities...

- curl

- openssl
- nmap
- bind-utils
- bind
- expect
- wget
- bzip2

```
yum install curl openssl nmap bind-utils bind expect wget bzip2
```

DNSSEC-Tools

Install and Compile DNSSEC-Tools

```
yum groupinstall 'Development Tools'
 yum install openssl-devel perl-devel perl-CPAN
 cd /usr/src
 wget https://www.dnssec-tools.org/download/dnssec-tools-2.2.tar.gz
 tar -xzf dnssec-tools-2.2.tar.gz
 ./configure
 make
 make install
```

3) Configuring the requirements:

SSL

1. Self signed certificates in CentOS 7 by default have been already installed.

If you want to change it, follow the steps below:

> Note: For production install, it is **HIGHLY RECOMMENDED** to use organization signed certs

Click here to expand...

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:
(shown using the vi editor, though you may use the editor of your choice)

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

Find the lines that start with SSHCertificateFile and change them to be like:

```
SSLCertificateFile /etc/pki/tls/certs/ca.crt
SSLCertificateKeyFile /etc/pki/tls/private/ca.key
```

Then restart.

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

Add 443 virtual hosts as needed in httpd.conf.

Apache

Allow overwrites in the apache vhosts

```
sed -i 's/AllowOverride None/AllowOverride All/g' /etc/httpd/conf/httpd.conf
```

Start Apache and make it start on boot

```
systemctl start httpd.service
    systemctl enable httpd.service
```

> **mod rewrite REQUIRED**
> Please note that mod_rewrite is required! If it is not enabled in Apache, key elements will not work as expected.

MySQL

Set the MySQL Configuration:

```
mysql -p -e "SET GLOBAL sql_mode='NO_ENGINE_SUBSTITUTION';SET SESSION
sql_mode='NO_ENGINE_SUBSTITUTION';"
```

Then enter the MySQL root password when prompted.

4) Optional configurations:

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:

If you are going to run iptables, click here

To list all current IPTable rules:

```
iptables -L
```

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
```

Note
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.

Radius (Optional)

This section only needs to be followed if the customer will be using Radius for authentication.

If you are going to use radius authentication, click here.

Install radius module:

```
pecl install radius
    echo extension=radius.so > /etc/php.d/radius.ini
```

SSH

Install ssh module:

```
yum install libssh2-devel
    pecl install -f ssh2
    echo extension=ssh2.so > /etc/php.d/ssh2.ini
```

5) Install 6connect ProVision Software:

1. Remove the current contents in the ProVision web folder (currently the www root) and after extract the archive contents (where 5.x.x is the version number for the build):

```
tar -xf productionBuild-5.x.x-php5.6.tar -C /var/www/html
```

2. Change the permissions to be the web user permissions

```
chown -R apache.apache /var/www/html
```

3. If enabled SELinux you must execute the following command:

```
chcon -R -t httpd_sys_rw_content_t /var/www/html

chcon -R -t httpd_sys_rw_content_t /tmp
```

You must also execute the same command for the 6c secure path created from **configSecureKeys.sh**

4. Go to http://<web root>/install/configTest.php. Follow the provided instructions, correcting any configuration errors if they occur. Once all steps are completed, you are ready to use your ProVision instance!

ProVision Local Installation For Debian 7/8

Debian 7/ Debian 8

Before You Begin

Ensure that System Requirements have been met prior to proceeding with the Installation Guide.

Install Requirements

1) Upgrade your current packages

Upgrade your current packages.

```
apt-get update
   apt-get upgrade
```

2) Install Required Packages

PHP

> This step is only applicable to Debian 7 users. If you are using Debian 8, please skip this step.

All installations of ProVision require at least PHP 5.6 (and related extensions). Debian Wheezy comes with PHP 5.4 by default. You can either add a repository which provides PHP 5.6 or install PHP manually. In this example, we are going to use 3rd party repository that also upgrades the Apache to 2.4 as well as some other packages.

Edit sources.list:
(shown using the vi editor, though you may use the editor of your choice)

```
vi /etc/apt/sources.list
```

Add the following lines to sources.list:

```
deb http://packages.dotdeb.org wheezy-php56 all
deb-src http://packages.dotdeb.org wheezy-php56 all
```

Install the GPG key:

```
wget https://www.dotdeb.org/dotdeb.gpg
apt-key add dotdeb.gpg
```

And update the packages list:

```
apt-get update
```

Install Apache2/PHP5/MySQL and development tools for pecl:

Click here for the list of extensions...

PHP5 / Apache2 / extensions

- apache2
- libapache2-mod-php5
- php5
- php5-cgi
- php5-cli
- php5-gd
- php5-curl
- php5-ldap
- php5-mysqlnd
- php5-ssh2
- php-pear
- php5-dev

Development tools for pecl / additional system packages:

- curl
- openssl
- memcached
- php5-memcache

```
apt-get install apache2 libapache2-mod-php5 php5 php5-cgi php5-cli php5-gd php5-curl
php5-ldap php5-mysqlnd php5-ssh2 php-pear php5-dev curl openssl memcached php5-memcache
```

While installing you will be asked to set the MySQL root password.

MySQL

Install MySQL to use a local database.

```
apt-get install mysql-server
```

You will be asked to enter a root password for the mysql server.

DNS and Additional Utilities

Install the DNS and other utilities:

Click here to expand...

- curl
- openssl
- memcached
- php5-memcache
- nmap
- dnsutils
- bind9utils
- dnssec-tools
- expect

```
apt-get install curl openssl memcached php5-memcache nmap dnsutils bind9utils
dnssec-tools expect
```

3) Configuring the requirements:

SSL

Enable the necessary apache modules with the following command:

```
a2enmod php5 ssl rewrite
```

In order to create self signed certificates and to work properly, the hostname must be resolved properly.
Open the hosts file for editing:

```
    vi /etc/hosts
```

Add a record for your hostname

```
    yourhostname X.X.X.X
```

Create self signed certificate:

```
    make-ssl-cert generate-default-snakeoil --force-overwrite
```

Apache

> **mod rewrite REQUIRED**
> Please note that mod_rewrite is required! If it is not enabled in Apache, key elements will not work as expected.

Enable the default vhost for apache

```
    a2ensite default-ssl
```

Allow overwrites in the apache vhosts

```
    sed -i 's/AllowOverride None/AllowOverride All/g'
/etc/apache2/sites-available/default*
```

6. Reload apache

```
    service apache2 reload
```

MySQL

MySQL Configuration:

```
    mysql -p -e "SET GLOBAL sql_mode='NO_ENGINE_SUBSTITUTION';SET SESSION
sql_mode='NO_ENGINE_SUBSTITUTION';"
```

Then, enter the MySQL root password when prompted.

4) Optional configurations:

Radius (Optional)

This section only needs to be followed if the customer will be using Radius for authentication.
If you are going to use radius authentication, click here.

1. Install radius module:
Debian 7:

Debian 7:

```
pecl install radius
```

Debian 8:

Debian 8:

```
 apt-get install php5-radius
```

Create module loading configuration:
(shown using the vi editor, though you may use the editor of your choice)

```
   vi /etc/php5/mods-available/radius.ini
```

Add the following lines to radius.ini:

```
   ; configuration for php radius module
   ; priority=20
   extension=radius.so
```

To enable the radius module, type the following command:

```
   php5enmod radius
```

5) Install 6connect ProVision Software:

1. Remove the current contents in the ProVision web folder (currently the www root) and after extract the archive contents (where 5.x.x is the version number for the build):

```
tar -xf productionBuild-5.x.x-php5.6.tar -C /var/www/
```

2. Change the permissions to be the web user permissions

```
chown -R www-data.www-data /var/www
```

3. Go to http://<web root>/install/configTest.php. Follow the provided instructions, correcting any configuration errors if they occur. Once all steps are completed, you are ready to use your ProVision instance!

ProVision Local Installation For Ubuntu

Ubuntu 14.04 LTS

Before You Begin

Ensure that System Requirements have been met prior to proceeding with the Installation Guide.

> **Permissions**
> Before installing, verify that you have appropriate install permissions. Depending on your permissions and system setup, commands may need to be run as superuser by prepending "sudo" to the listed commands.

Install Requirements

1) Upgrade your current packages

Upgrade your current packages.

```
apt-get update
   apt-get upgrade
```

2) Install Required Packages

PHP

All installations of ProVision versions 5.2.0 or later require PHP 5.6 and related extensions. You can either add a repository which provides PHP 5.6 or install PHP manually. In this example, we use the default package sources for Ubuntu 14.04 to install Apache 2.4 and PHP 5.6. We also provide an option to install PHP 5.6 through a third-party repository.

If you are installing PHP 5.6 click here...

> If you are installing ProVision 5.2.0, or after, make sure that your build archive has "php5.6" as part of its name. For example " productionBuild-5.2.0-php5.6.tar".

1. Add PHP 5.6 package sources to your system:

```
add-apt-repository ppa:ondrej/php5-5.6
```

and confirm with ENTER to continue.

If you get an error here, you may need to install python-software-properties first and then repeat the repository add as follows:

```
apt-get update
apt-get install python-software-properties

add-apt-repository ppa:ondrej/php5-5.6
```

2. Update packages:

```
apt-get update
```

3. Install PHP:

```
apt-get install php5
```

and confirm with “y” and ENTER.

You’ll now see “PHP 5.6.x” in the scrolling installation logs.

To verify the installed version of PHP, run:

```
php5 -v
```

Install Apache2/PHP5/MySQL and development tools for pecl:

Click here for the list of extensions...

PHP5 / Apache2 / extensions

- apache2
- libapache2-mod-php5
- php5
- php5-cgi
- php5-cli
- php5-gd
- php5-curl
- php5-ldap
- php5-mysqlnd
- php-pear
- php5-dev

Development tools for pecl / additional system packages:

- curl
- openssl

```
apt-get install apache2 libapache2-mod-php5 php5 php5-cgi php5-cli libssh2-php php5-gd
php5-curl php5-ldap php5-mysql php-pear php5-dev curl openssl
```

MySQL

ProVision 5.2 and above require MySQL 5.6.20 and above.

To install MySQL 5.6.20+, click here...

Add new repository for MySQL 5.6 distribution.

```
apt-key adv --keyserver pgp.mit.edu --recv-keys 5072E1F5
echo "deb http://repo.mysql.com/apt/ubuntu/ trusty mysql-5.6" >>
/etc/apt/sources.list.d/mysql.list
```

Update current packages.

```
apt-get update
```

Install MySQL 5.6 to use a local database.

```
apt-get install mysql-server-5.6
```

You will be asked to enter a root password for the mysql server.

MySQL is now installed. You can proceed to next section DNS And Additional Utilities

Install MySQL to use a local database.

```
apt-get install mysql-server
```

You will be asked to enter a root password for the mysql server.

DNS and Additional Utilities

Install the DNS and other utilities:

Click here to expand...

- curl
- openssl

- nmap
- dnsutils
- bind9utils
- dnssec-tools
- expect

```
apt-get install curl openssl nmap dnsutils bind9utils dnssec-tools expect
```

3) Configuring the requirements:

SSL

Enable the necessary apache modules with the following command:

```
a2enmod php5 ssl rewrite headers
```

In order to create self signed certificates and to work properly, the hostname must be resolved properly.
Open the hosts file for editing:

```
vi /etc/hosts
```

Add a record for your hostname

```
X.X.X.X yourhostname
```

Create self signed certificate:

```
make-ssl-cert generate-default-snakeoil --force-overwrite
```

Apache

> **mod rewrite REQUIRED**
> Please note that mod_rewrite is required! If it is not enabled in Apache, key elements will not work as expected.

Enable the default vhost for apache

```
a2ensite default-ssl
```

Allow overwrites in the apache vhosts

```
sed -i 's/AllowOverride None/AllowOverride All/g' /etc/apache2/apache2.conf
```

Update site configurations to use /var/www for docroot

```
sed -i 's/\/var\/www\/html/\/var\/www/' /etc/apache2/sites-available/000-default.conf
sed -i 's/\/var\/www\/html/\/var\/www/' /etc/apache2/sites-available/default-ssl.conf
```

Restart apache

```
service apache2 restart
```

MySQL

MySQL Configuration:

```
mysql -p -e "SET GLOBAL sql_mode='NO_ENGINE_SUBSTITUTION';SET SESSION
sql_mode='NO_ENGINE_SUBSTITUTION';"
```

Then, enter the MySQL root password when prompted.

4) Optional configurations:

Radius (Optional)

This section only needs to be followed if the customer will be using Radius for authentication.
If you are going to use radius authentication, click here.

Install radius module:

Debian 8:

```
apt-get install php5-radius
```

Create module loading configuration:
(shown using the vi editor, though you may use the editor of your choice)

```
vi /etc/php5/mods-available/radius.ini
```

Add the following lines to radius.ini:

```
; configuration for php radius module
    ; priority=20
    extension=radius.so
```

To enable the radius module, type the following command:

```
    php5enmod radius
```

SSH

Install ssh module:

```
apt-get install libssh2-1-dev
```

then run

```
pecl install -f ssh2
```

Create module loading configuration:
(shown using the vi editor, though you may use the editor of your choice)

```
    vi /etc/php5/mods-available/ssh2.ini
```

Add the following lines to ssh2.ini (adding extension=ssh2.so to your ssh2.ini):

```
; configuration for php ssh2 module
    ; priority=20
    extension=ssh2.so
```

To enable, type the following command:

```
php5enmod ssh2
```

Reload apache

```
service apache2 reload
```

5) Install 6connect ProVision Software:

The latest version of ProVision can be found at https://cloud.6connect.com/Download/Latest/ and downloaded using the credentials provided to you.

If you need credentials provided to you, or any other assistance, please contact our support team atsupport@6connect.com.

1. Remove the current contents in the ProVision web folder location (currently the www root) and after extract the archive contents (where 5.x.x is the version number for the build):

```
tar -xf productionBuild-5.x.x-php5.6.tar -C /var/www/
```

2. Change the permissions to be the web user permissions

```
chown -R www-data.www-data /var/www
```

3. Go to http://<web root>/install/configTest.php. Follow the provided instructions, correcting any configuration errors if they occur. Once all steps are completed, you are ready to use your ProVision instance!

6connect Local Software Upgrades

Local Software Upgrades

Upgrade Methods for 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 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

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

7. Login and use!

Local Installations - Peering Setup

Peering Setup - Local Installations:

ProVision uses a locally-hosted mirror of the PeeringDB database in order to perform non-edit Peering functions. There are a few steps to take in order to set up your locally hosted instance to coordinate with PeeringDB information.

1) First, download the PeeringDB SQL dump file from the following location:

http://www.peeringdb.com/dbexport/peeringdb.sql

2) Then, create a new database to contain that data on your local MySQL installation. Make sure that the MySQL user configured in the data/globals.php file is allowed to read and make changes to the local copy of the PeeringDB database.

3) Once this has been done, the ProVision global configuration file located at

```
[ProVision Root]/data/globals.php
```

must be updated with the following variable to inform ProVision of the location of this new install:

```
$peeringdb_db_name = 'peeringdb';            // name of the database in MySQL
```

Since this creates a local mirror of an external database it must be periodically re-downloaded to be kept up-to-date. 6connect recommends that local system administrators add the following bash script to their crontab, set to run once a day.

Click here for the script...

```
#!/bin/bash



        DB="peeringdb"

        DBUSER="root"

        DBPASS="*******"

        WGET="/usr/bin/wget"

        MYSQL="/usr/bin/mysql"



        cd
            /tmp/

        $WGET
            http://www.peeringdb.com/dbexport/peeringdb.sql

        /bin/cat
            /tmp/peeringdb.sql | $MYSQL -u$DBUSER -p$DBPASS $DB

        /bin/rm
            /tmp/peeringdb.sql
```

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.

ProVision Getting Started

First Steps - Not sure where to begin? Here are some key items to consider when setting up your first ProVision instance, high level overviews of Resources, Permissions, and Importing data into ProVision.

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.

First Steps

ProVision First Steps

Overview Video: First Steps - Part 1

This video gives a high level overview of the "Before you Begin" content on this page, as well as an introduction to ProVision's Resource System.

This video may also be viewed at https://www.youtube.com/watch?v=apJRcQv3ZQ0.

Before you Begin

We recommend that new users work through the following questions with their internal team to plan their ProVision instance:

> **Need Help?**
> Remember that 6connect's engineers are here to help. If you have questions, or want to test out some ideas, our team has worked with a variety of data sets and can help get you started on the right path. You can reach us any time at support@6connect.com.

1) What type of physical and non-physical components do you wish to track?

Impacts: What Sections and Resources are created

An important first step is determining what items you currently are, or will be, tracking, and what relationship they have with each other. ProVision's flexible Resource system allows you create and customize detailed entries for any type of item you may want to track: Customers, Contacts, Data Centers, Routers, VMs, and more. These types of resources are labeled as "Sections" in ProVision. Once a section is made, individual items (resources) may be created as a part of that Section. Each Section may have different Gadgets selected, which then provide additional functionality on a resource entry page.

See: Working with Resources, Customizing Sections, Gadgets

2) What is your current data structure? What is your ideal data structure?

Impacts: Resource Hierarchy, Assignment Behavior

The resource hierarchy structure in ProVision allows for "child" resources to be created under a parent resource (for example: servers as children under a datacenter resource entry, or subsidiaries under a parent company, who then share IP aggregates). The structure decided upon will influence how resources are set up in ProVision, as well as the behaviors of functions while working with items such as IP blocks or DNS zones.

See: Working with Resources, Resource Concepts

3) Who needs access to what data?

Impacts: User and Group Permissions

In ProVision, standard user permissions are set by resource and functional area (IPAM, DNS, Peering, etc). Global Admin permissions give access to additional functions such as configuration settings, DNS management, Importing, and Scheduler Tasks. Determine which users will

require administrative access, and which will have access to only specific resources or functional areas. You will need to get more specific later, but having some high level groups to work with is a great start.

See: Users & Permissions

Overview Video: First Steps - Part 2

This video gives a high level overview of the "Getting your Data into ProVision" content on this page.

This video may also be viewed at https://www.youtube.com/watch?v=2e0H1H4rTTs.

Getting your Data into ProVision

After determining your internal goals and processes, it's time to get your data into ProVision!

> **Start Small**
> When importing data into ProVision, data validation is a key step to ensure that everything is accurate. Upon importing your data, you may see some errors that result in a stop in the import process! It is recommended that you break up your imports to both keep them manageable and give you a chance to normalize your data prior to importing.

1) Gather and Prep your data

Determine where your data will be from:

Excel / Spreadsheets:

May be used to import: Resources, IP Aggregates/ Blocks, DNS BIND Zones

If you currently use Excel or other spreadsheet program for tracking, you will need to verify that your spreadsheets are "cleaned up" according to the information on the Importing Your Data page under "Preparing for Data Import". Make sure that you use UTF-8 encoding, remove extraneous blank rows, and compare your data to the data fields available in ProVision, shown under "Which Import Tool Should I Use?". Review sample files if desired to see example formats. If you plan to track custom types of Resources by creating Sections, you will need to create a Section with custom fields, and verify that your spreadsheet contains the same fields.

Once your verification / cleanup is complete, export your spreadsheet as a .csv file.

See: Importing Your Data, Working with Resources, Import DNS Zones

RIR

May be used to import: IP Aggregates

No advance preparation is needed for aggregates imported from RIR. ProVision's built-in importer will ask for your ORGID or an IP, and then populate an aggregate list from that information. Simply choose which aggregates you wish to import.

See: Import Aggregate Blocks

PowerDNS Server or DynECT Server

May be used to import: DNS Zones

ProVision provides automated tools for importing DNS zones from a PowerDNS or DynECT server. However, prior to using those tools for the first time, the server must be added to ProVision. Review the DNS Administration documentation, and add your PowerDNS or DynECT server.

See: DNS Administration, PowerDNS Zone Import, DynECT Zone Import

Manually Adding Data

ProVision allows manual adding of data at any time. We recommend verifying that the item has not already been added beforehand (to prevent duplicates), and keeping your desired data structure in mind.

2) Import or Manually Add Data

The order in which items are added will depend on what ProVision functional area (Resources, IPAM, DNS) you will be using, and what the

current / desired data structure is. In general, the following order is recommended:

A) Resources

Importing or creating your resources first allows subsequent items to be associated with those resources.

> Note: If you already have "Resource" data associated with your IP block data (ie, as fields in a spreadsheet with Resource Name and Resource ID), you may choose not to create those resources ahead of time. The IP Import from CSV tool will give you an option to create those Resources during the IP import process.

Import customers, physical devices, locations, and so forth through the Resource Import from CSV tool. If you wish for additional customization, you can create a custom Section, add the desired fields, and import resources under that Section through the Resource Import Tool.

Adding resources manually may be done at any time under the **Resources** Tab, by clicking the "Add Entry" button.

See: Working with Resources, Importing Your Data

B) IP Aggregates and Blocks

Import your IP Aggregates through the Import from RIR tool, or from a .csv file via IP Import from CSV.

The Import from CSV tool will create Top-Level Aggregates and place blocks under those aggregates based on the following method:

First, the importer will parse through the provided data, order all blocks from largest to smallest, then attempt to split the largest block out of an existing block matching the IP space and RIR. If that fails (no larger block exists), then that block is added in the system as its own Top-Level-Aggregate. Subsequent blocks will undergo the same process.

This method ensures that your list of IP blocks does not need to be organized in any particular order.

> If your import includes a large number of small blocks (/30s, /32s), be sure you've included at least one large block which encompasses them (/24, /22). This will ensure that the smaller blocks are neatly organized under the larger block, rather than imported as their own Top-Level Aggregates.

See: Importing Your Data

C) Add DNS servers and zones

If using the DNS functionality in ProVision, you will need to add your servers prior to importing zones.

Add DNS Severs

Servers are added under the DNS Admin section of ProVision, under "Manage DNS Servers". Click on "New Server" and fill out the server information.

See: DNS Administration

Information for specific server types is available under the following sections:

Configuring ISC BIND Support

Configuring DynECT Support

Configuring PowerDNS Support

Configuring Secure64 Support

Import DNS Zones

After creating the applicable servers in ProVision, you may import or manually add DNS zones.

ProVision offers three DNS zone import options, available under the Data Import tab in the Admin section. For more information on importing DNS zones, see Importing your Data and Import DNS Zones.

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. See: BIND DNS Zone Upload / Import

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. See: DynECT Zone Import

PowerDNS Zone Import

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

Manually adding zones may be done at any time from the **DNS** Tab. See Working with DNS Zones - Common Tasks.

3) Add Users and Groups

In ProVision, the permissions structure is handled by assigning users to groups, then setting specific resource-level C/R/U/D permissions for that group. It 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

See: Users & Permissions, Users and Groups

Working In Provision

With the basic data now imported, and users set up, you (and your team) are ready to work in ProVision!

Refer to our User Guide and Admin Guide for details on standard user and admin level areas of ProVision. Or, follow the links below for additional details grouped by specific task areas:

Concepts:

Resource Concepts

Workflow Concepts

Resources:

Working with Resources

IPAM:

IPAM Tab

Working with IP Blocks

IPAM Administration

DNS:

DNS Tab

Working with DNS Zones - Common Tasks

DNS Administration

Peering:

Peering v2

Peering - Common Tasks

Import Sessions

Resource Concepts

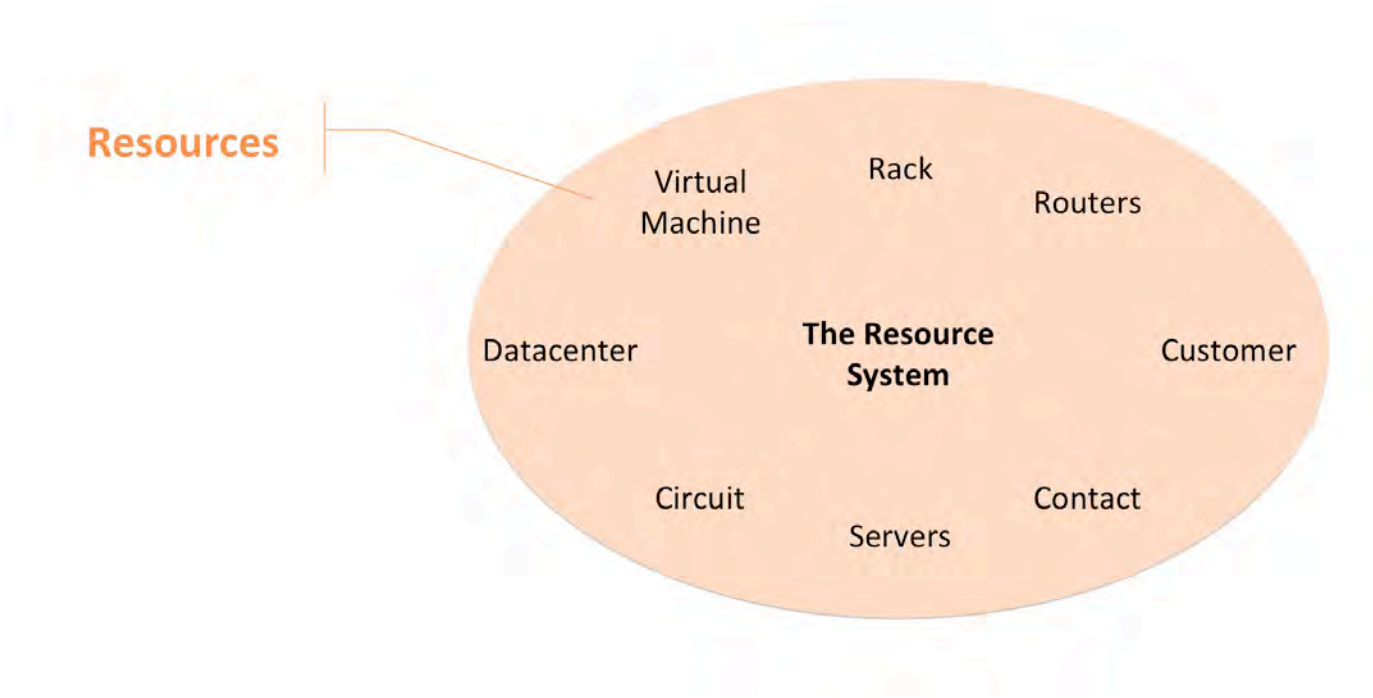
The Resource System

Resource System Components

Resources

In ProVision, a “Resource” is simply an umbrella term for the components that you are tracking. Resources may be:

- Physical assets such as servers and routers
- Individual people or companies (customers)
- Places holding your assets, such as Datacenters



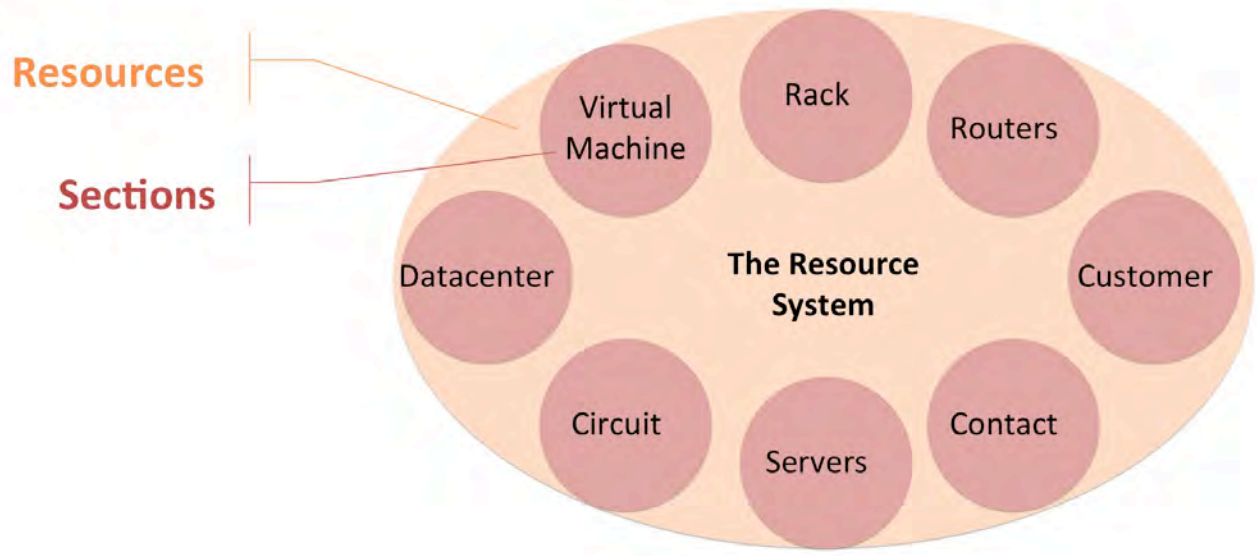
Section

The generic type of a resource is called a Section. General terms such as Rack, Routers, Customer, Servers, and Datacenters would be considered Sections in ProVision.

You might have an individual server named "Test Server", but its generic type of resource is that of a “Server” – so we would associate “Test Server” with the Section “Server”.

Think of Sections as templates for your resources. When a Section is created, you can associate specific fields and gadgets with that Section. In the case of our "Server" section, we would want to associate fields that are common server properties, such as Make, Model, Operating System,

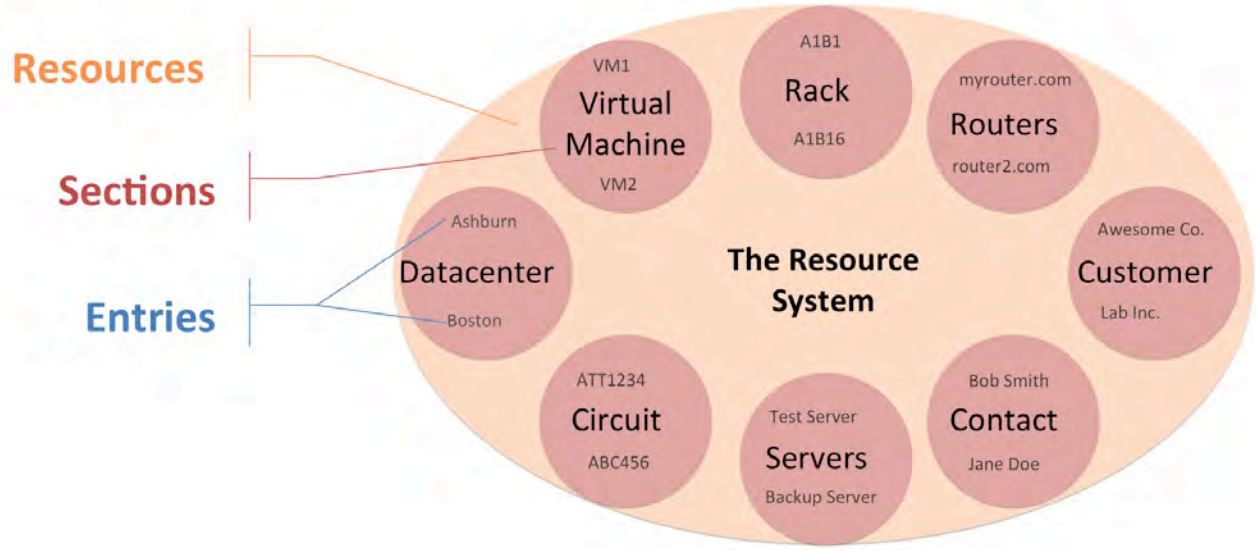
and Domain. Then, anytime we looked up a specific server in ProVision, we can see and update the Make, Model, OS, and Domain information.



Entry

Think of an Entry as the individual item you are tracking – it has a specific name, it might have an IP Address, a physical address, or additional descriptive data that is associated with it.

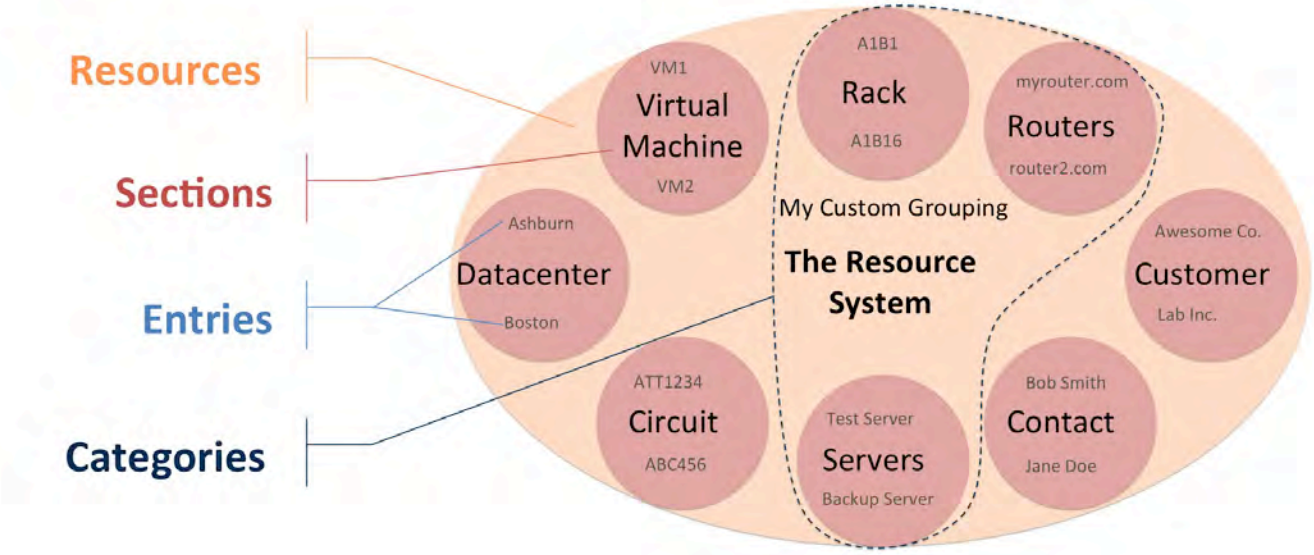
Our "Test Server" resource is an Entry under the "Server" Section.



Here, we can see that under the “Datacenter” section, we have two entries – one is the Ashburn Datacenter, the other is the Boston Datacenter. Under “Servers”, we have entries for “Test Server” and “Backup Server”.

Category

Another part of ProVision’s resource system is “Categories”. Categories allow you to group resources together under a name of your choosing to further organize your resources. A category that is often used is “Customer”, grouping customer resources together

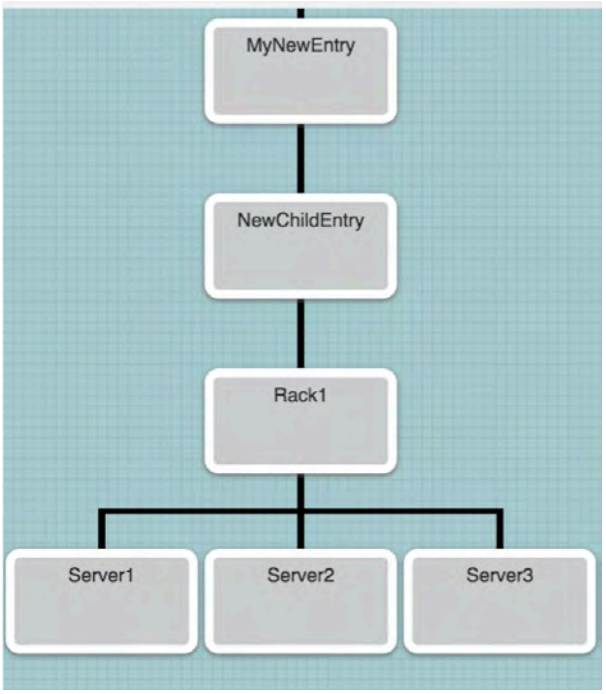


In this example, we are grouping Racks, Servers, and Routers under the “My Custom Grouping” Category.

Creating categories is entirely optional, so you can decide internally if the additional organization is needed in your instance.

Resource Hierarchy

In ProVision, "Child" Resources can be created underneath a "Parent" resource. This functionality allows for hierarchies to be created to match the desired organizational structure. An example of this would be to create racks and servers as children under a datacenter entry. Chart View, available from the **Resources** Tab, lets you see this structure graphically.



The structure decided upon will influence how resources are set up in ProVision, as well as behaviors of functions while working with items such as IP Blocks or DNS zones.

Additional Information:

Working with Resources

Working with Entries

Customizing Sections

Customizing Fields

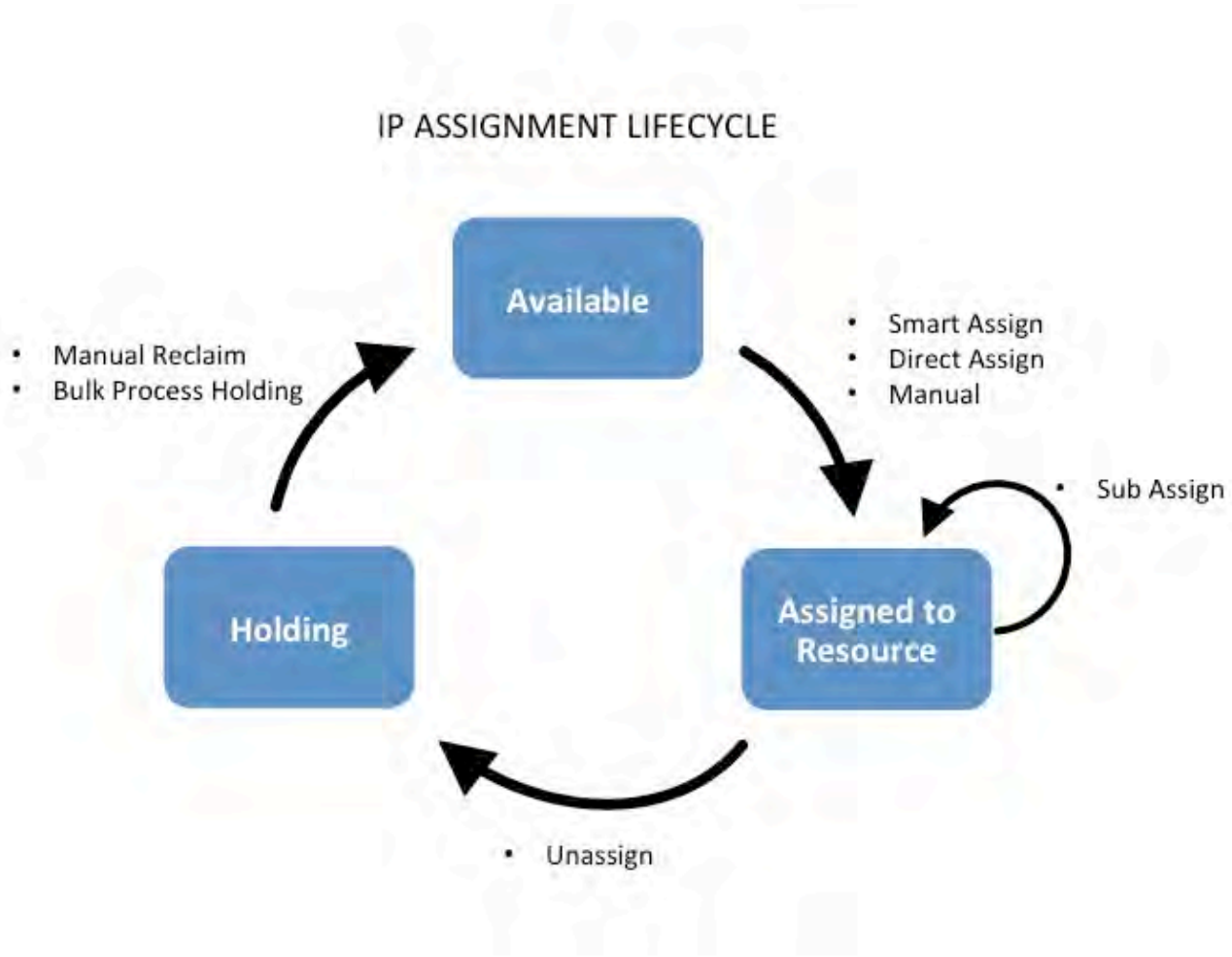
Gadgets

Workflow Concepts

Workflow Concepts

IP Assignment Lifecycle

In ProVision, the IP assignment lifecycle starts with an available block which is free to be assigned to any IPAM-enabled resource holder. There are multiple methods that may be used to assign a block to a resource holder: Smart Assign, Direct Assign, or Manual Assign (Smart Browse). Once an IP block is assigned, blocks can be further subassigned via the same methods if desired. When an assigned block is un-assigned it proceeds into the Holding Tank: a special resource 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:

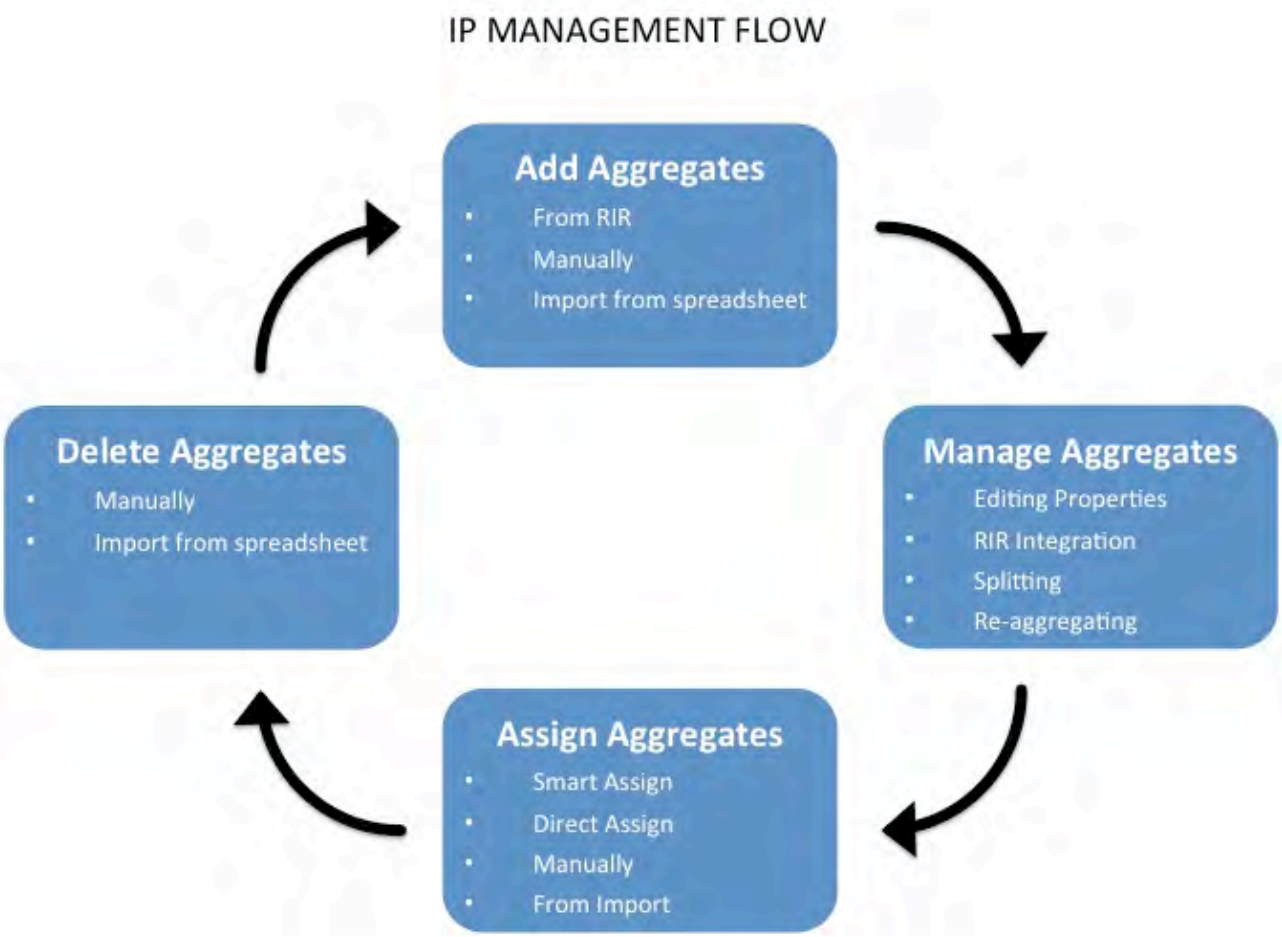
Working with IP Blocks

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.



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

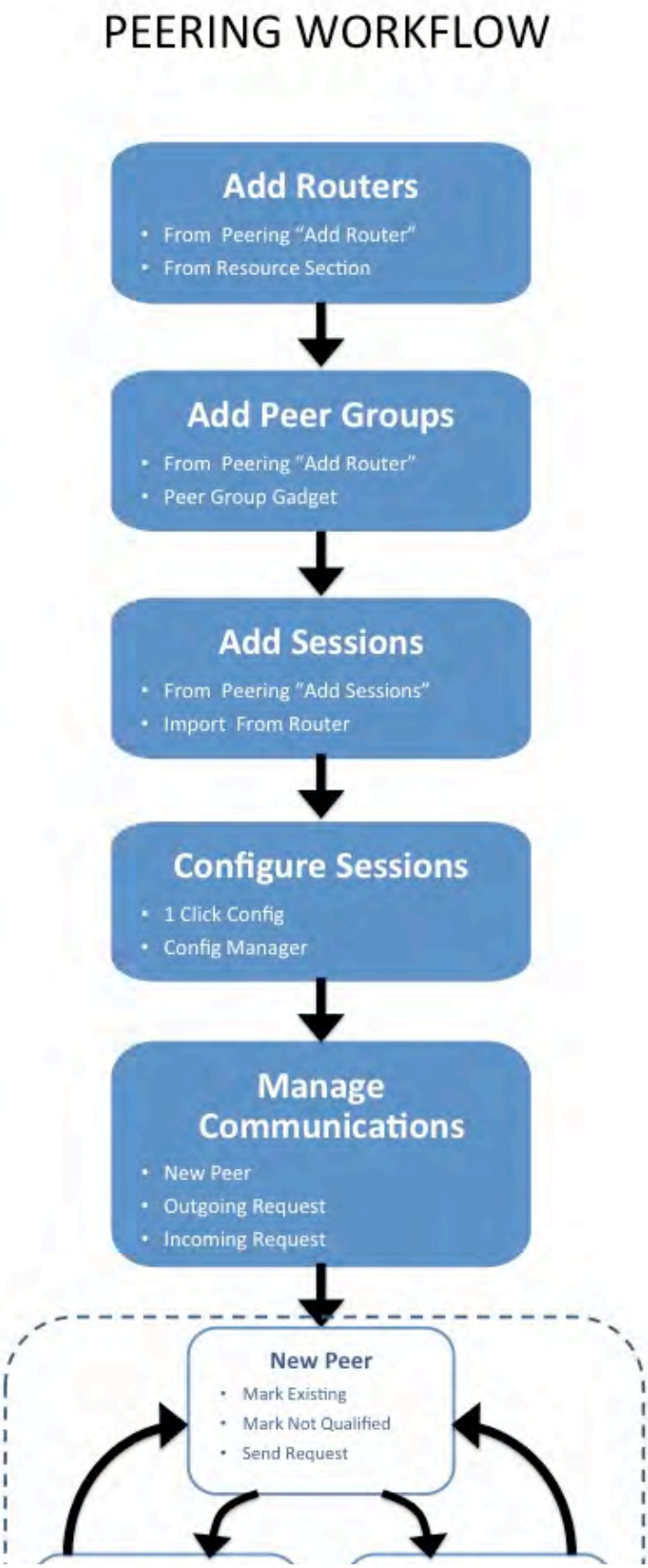
Import Aggregate Blocks

Peering

In ProVision, Peering starts with designating Routers. Routers may be added through the Resource system under a router Section, but the simpler approach is to add a new router via the "Add Router" button under the Peering Tab. Through this dialog, Peer Groups may be added at the same time.

After the router(s) and Peer Groups have been created in ProVision, Sessions need to be added. Sessions may be added manually through the "Add Session" dialog in the Peering tab, or Imported from a router (requires ProVision Admin permissions). Once a session has been created, it can be configured and managed through the Sessions list. The Peer that is associated with the session is added to the Peer Communications list,

where communications and status may be managed.



For more information on performing tasks under Peering, see the following documentation sections:

Peering - Common Tasks

Add Routers

Add Sessions

Import Sessions

Managing Peer Sessions

Managing Peer Communications

VLAN Manager

The VLAN Manager allows Admin users to add domains and VLANs to their ProVision instance, and associate them with IP Blocks. Optionally, VLAN tags may be associate with ranges or individual VLANs under a domain, to help limit VLAN search results when enabling VLANs.

Standard flow (without VLAN tags):

The workflow starts with creating a domain in the VLAN Admin section of ProVision. During creation, domains may be selected as "standard" or "extended" domains, declaring the size of the VLAN pool from which VLANs are available to be enabled. Next, VLANs must be enabled and added to the domain. This is done under the IPAM Tab -> VLAN section of ProVision. Add VLANs to the domains by clicking "Add", searching for the desired range of VLANs to enable from the standard / extended pool, and selecting the desired VLANs to enable. Once enabled, VLANs may be edited or have IP blocks associated with that VLAN. Editing VLANs and Direct / Smart Browse / Search IPs functions for adding blocks to VLANs are available from the Domain/ VLAN list under under the IPAM Tab -> VLAN section of ProVision by expanding the desired domain and clicking on the VLAN link. IP blocks may also be edited individually through the IPAM gadget, IPAM Manage, and VLAN Manage areas to add domain and VLAN information to the block.

Optional flow (with VLAN tags):

The workflow starts with creating a domain in the VLAN Admin section of ProVision. During creation, domains may be selected as "standard" or "extended" domains, declaring the size of the VLAN pool from which VLANs are available to be enabled. VLAN tags may be added (from the VLAN Admin Tab -> Edit Tags submenu) before or after domain creation. Tags created in this area are available to all domains and VLANs.

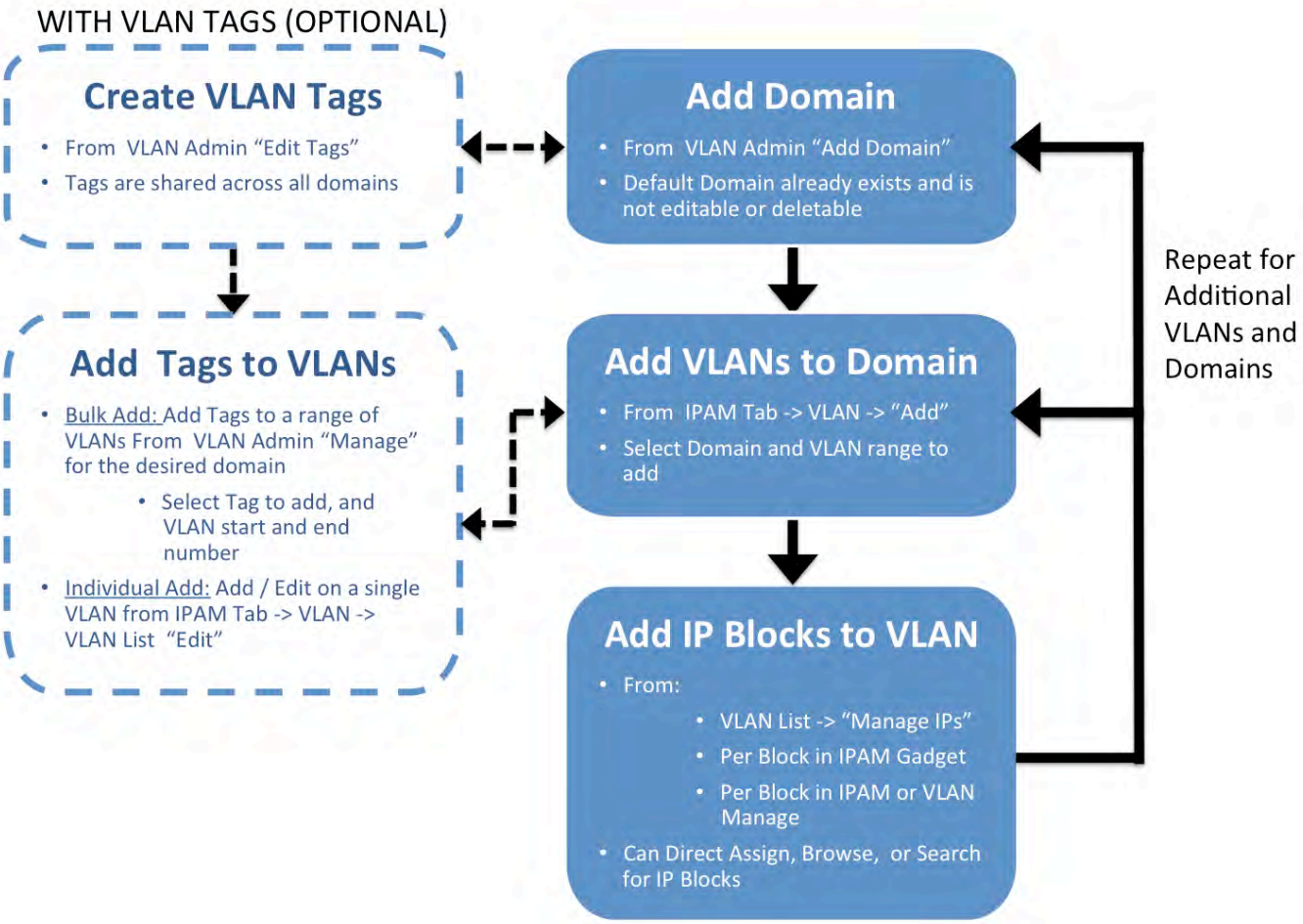
If the tags created need to be added to a large range of VLANs, the next step would be to add tags to the desired range(s) of VLANs from the VLAN Admin -> Domain "Manage" button. Domain Manage allows you to select an existing tag, type in the numeric VLAN start and end point, and assign that tag to that entire range of VLANs under the current domain.

Once a bulk range of tags is associated with VLANs, you can use the tag as a search criteria when adding VLANs to a domain. This is done under the IPAM Tab -> VLAN section of ProVision. Add VLANs to the domains by clicking "Add", using the tag name as a search field for the desired range of VLANs, and then selecting the desired VLANs to enable.

If tags are only desired for individual VLANs, they may be added when editing a VLAN after the VLAN has already been added/enabled for the domain. Editing VLANs may be accessed from the IPAM Tab -> VLAN section, then expanding the desired domain and clicking on the VLAN link, selecting "Edit".

Next, VLANs must be enabled and added to the domain. This is done under the IPAM Tab -> VLAN section of ProVision. Add VLANs to the domains by clicking "Add", searching for the desired range of VLANs to enable from the standard / extended pool, and selecting the desired VLANs to enable. Once enabled, VLANs may be edited or have IP blocks associated with that VLAN. Editing VLANs and Direct / Smart Browse / Search IPs functions for adding blocks to VLANs are available from the Domain/ VLAN list under under the IPAM Tab -> VLAN section of ProVision. IP blocks may also be edited individually through the IPAM gadget, IPAM Manage, and VLAN Manage areas to add domain and VLAN information to the block.

VLAN MANAGER WORKFLOW



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.

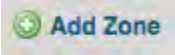
Interactive Icons:

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.

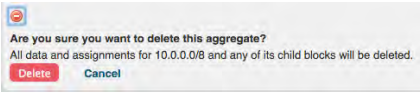
"Add" Button

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.



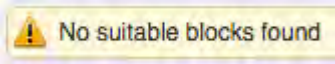
Status Icons:

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

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. The Dashboard is comprised of modular "Widgets" that show different types of information. These Widgets may be added, moved, edited, and customized to create personalized Dashboards.

There are two levels of Dashboards:

Default Dashboard: The Default Dashboard is a shared dashboard for all users of the ProVision instance. It may only be edited by admin level users, but it is viewable by all users.

Individual Users' Dashboards: Additional Dashboards may be created by individual users. These dashboards are tied to the user logged in, and are only viewable / editable by that user.

> **Dashboards and Permissions**
> Note: An individual's permission levels may limit what that user can view in both default and personal Dashboards. Although all users can view the Default Dashboard, they may be not be able to view all elements or data inside the dashboard.

ProVision Default Dashboard Overview:

Although each user may have access to or create wildly differing dashboards, below are descriptions of the Widgets available shown on the ProVision Default Dashboard (from left to right):

Activity Chart (Viewable by Admins Only):

Illustrates activity level for API, IPAM, DNS, Peering, and Resource areas of ProVision.

Select the desired ProVision area by clicking on the radio buttons. Then, hover over the chart, and use the mouse scroll wheel to zoom in or out of specific date / times. Clicking on a bubble will show activity detail, and resource / IP block links if applicable.

Only one Activity Chart Widget may exist on a dashboard.

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.

Only one IP Charts Widget may exist on a dashboard.

Clock:

Shows the current time and date for the selected time zone, based on format set by the user.

Multiple Clock Widgets may exist on a dashboard.

Status:

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

Only one Status Widget may exist on a dashboard.

Resource Tree:

An interactive horizontal or vertical treeview of a selected Resource and its children. Select a top-level resource from the Widget's Edit menu. Users may click and drag to move within the Widget, as well as zoom in and out using a scroll wheel. Clicking on filled dots will expand the tree to show the children of that resource. Double-clicking on a resource in the tree will give you the option to navigate to the resource's entry page.

Multiple Resource Tree Widgets may exist on a dashboard.

Contact Us:

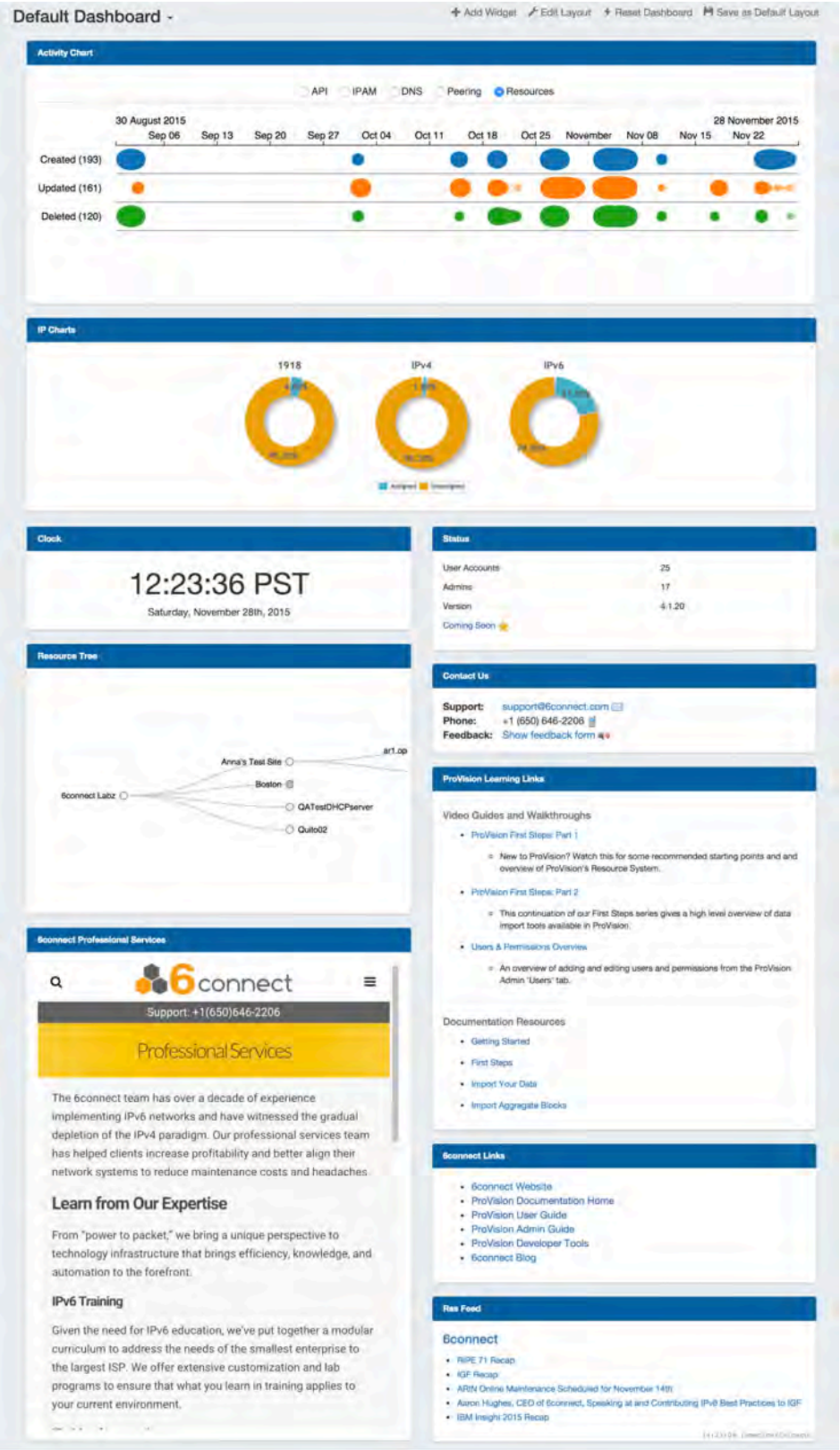
Provides support email, phone, and feedback form information. Support phone number and email address are set from the ProVision Admin settings. The feedback form email address is editable within the Widget.

Only one Contact Us Widget may exist on a dashboard.

ProVision Learning Links: (Markdown Widget)

Pre-created Markdown Widget containing links to commonly referenced ProVision documentation sections, video guides, and walkthroughs.

The Markdown Widget is a blank



slate where users may add links, text, or other content through use of the Markdown language.

Multiple Markdown Widgets may exist on a dashboard.

6connect Professional Services: (iFrame Widget)

Pre-created iFrame Widget accessing the 6connect Professional Services website.

The iFrame Widget sets a https URL to be viewed in an iFrame, to view stats or commonly referenced websites.

Multiple iFrame Widgets may exist on a dashboard.

6connect Links (Links Widget):

Pre-created Links Widget listing 6connect ProVision company and documentation links.

Link URL and display may be set from within the Widget.

Multiple Links Widgets may exist on a dashboard.

6connect RSS (RSS Feed Widget):

Pre-created RSS Feed Widget accessing the 6connect Blog RSS.

Shows the most recent five entries of a selected https RSS Feed. Feed links open in a new window when clicked.

Multiple RSS Feed Widgets may exist on a dashboard.

Customizing the Dashboard

Dashboard Customization

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. The Dashboard is comprised of modular "Widgets" that show different types of information. These Widgets may be added, moved, edited, and customized to create personalized Dashboards.

There are two levels of Dashboards:

Default Dashboard: The Default Dashboard is a shared dashboard for all users of the ProVision instance. It may only be edited by admin level users, but it is viewable by all users.

Individual Users' Dashboards: Additional personal dashboards may be created by individual users. These dashboards are tied to the user logged in, and are only viewable / editable by that user.

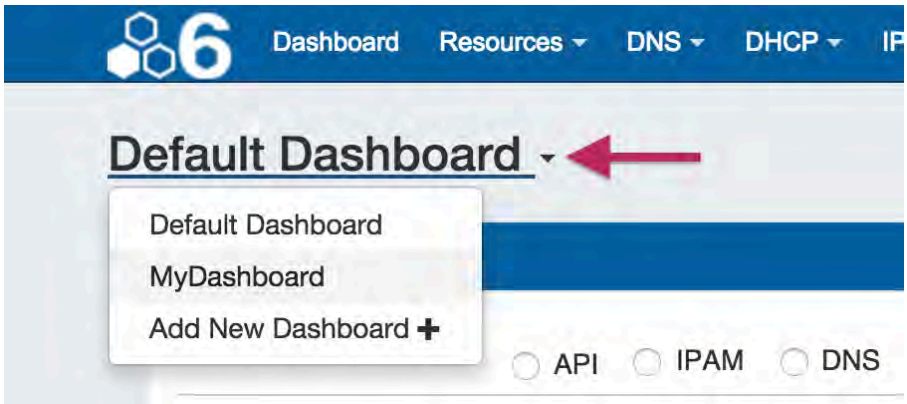
> **Dashboards and Permissions**
> Note: An individual's permission levels may limit what that user can view in both default and personal Dashboards. Although all users can view the Default Dashboard, they may be not be able to view all elements or data inside the dashboard.

> **Browser Compatibility**
> It is recommended to use the most recent versions of the Chrome or Firefox browsers when working with the dashboard. Due to incompatibilities with Internet Explorer, the dashboard is set to read-only mode and is not editable when accessed from IE at this time.

Working with Dashboards

Viewing an Existing Dashboard

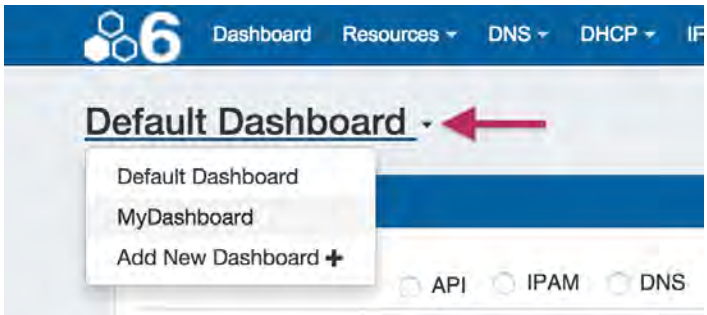
The Default Dashboard will be the primary dashboard that appears when clicking on the **Dashboard** Tab. To view another, already created dashboard, click on the dashboard name at the top left of the Dashboard page. This will open the list of existing dashboards. From there, click on the name of the dashboard you wish to view.



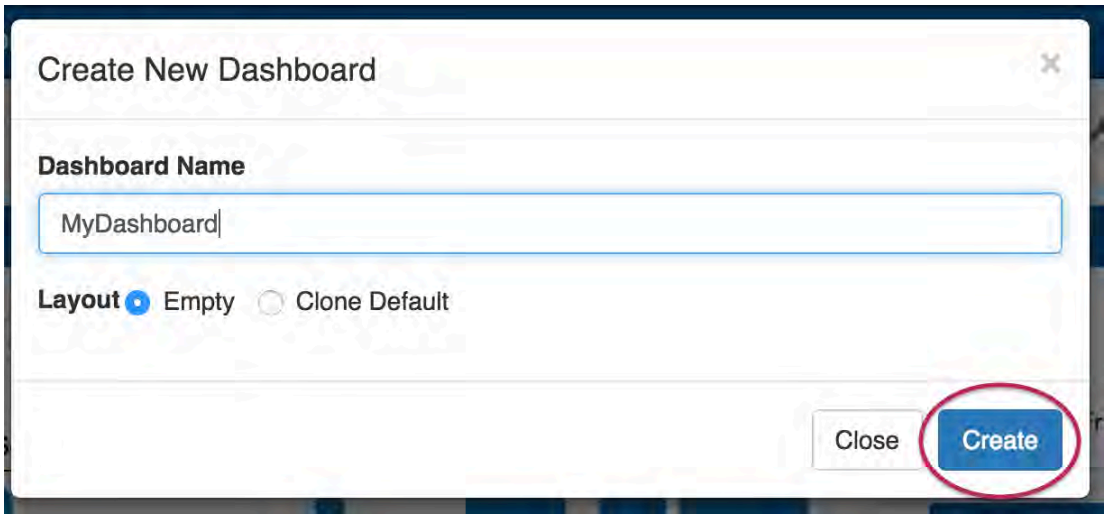
Adding a Dashboard

To add a new personal dashboard, click on the dashboard name at the top left of the Dashboard page. This will open the list of existing dashboards.

Click on "Add New Dashboard"



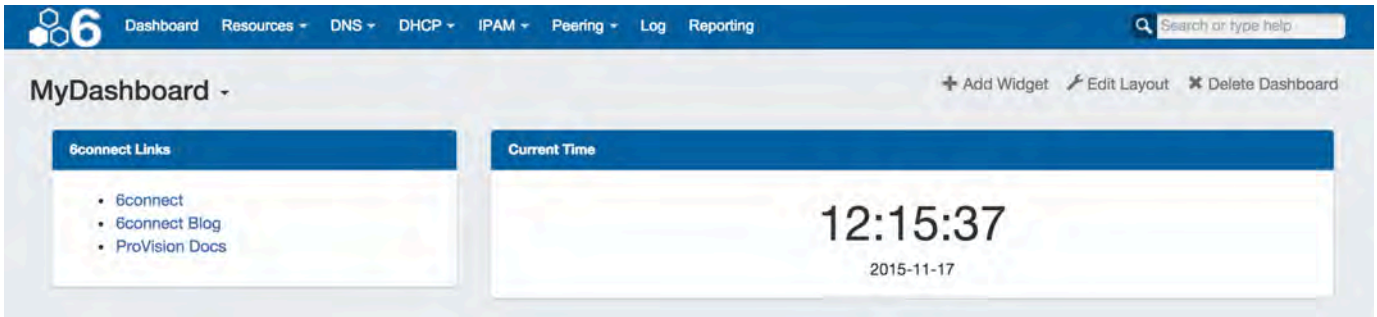
Fill out the desired dashboard name, and choose whether the dashboard will be empty or cloned from the default dashboard. Selecting "Empty" will provide a (mostly) blank canvas for selecting and placing Widgets from scratch; selecting "Clone" will duplicate the current state of the Default Dashboard, which the user can then edit as desired.



Note
Note: If "Empty" is selected, a default Widget will still show in the head of each page column, this is to assist in both visualizing the current page layout and to aid in placing new Widgets. These widgets may be edited or deleted as desired during the process of selecting layout and adding other Widgets.

For best performance of the dashboard, ensure that at least one Widget is placed in each column, or columns may be difficult to "detect" when moving Widgets to different page areas.

MyDashboard has now been created as an "Empty" dashboard, is accessible from the dropdown list of dashboards, and is ready to set up!

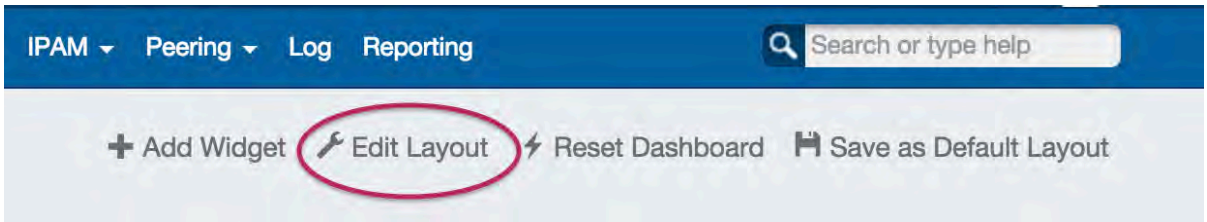


Editing a Dashboard

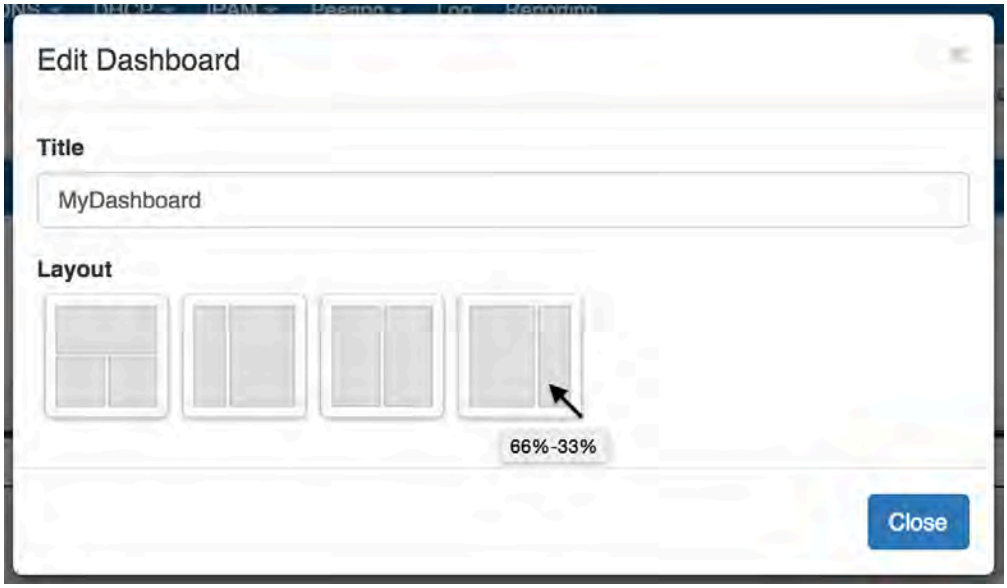
You can change the layout of your dashboard by selecting overall page layout options (for column size and location on the page), as well as move individual widgets around the page.

To Edit the Page Layout:

At the top of the page, click "Edit Layout".



This will bring up an option box where you can choose to rename your dashboard, and / or select a column organization structure.



Title: Type in how you want your dashboard title to display.

Layout: Click on a layout option to immediately apply that layout to your dashboard. Hovering over the option will show the column width percentages.

100% / (50% / 50%): Creates a page with a full width section at the top, and two equal sized columns below that.

33% / 66%: Creates a page with two columns, with the left column being one-third of the page width, and the right column as two-thirds of the page width.

50% / 50%: Creates a page with two equal sized columns, split down the middle.

66% / 33%: Creates a page with two columns, with the left column as two-thirds of the page width, and the right column as one-third of the page width.

To Edit the Location of Individual Widgets:

Individual Widgets may be moved around the page by clicking and dragging on the "Move" icon on the right side of the Widget header.

While mousing over the move icon, when the cursor changes to a hand icon, click and hold your left mouse button, then drag the widget up or down to the desired location. To move the Widget to another column, drag the widget sideways to the column, then up to briefly hover over a Widget already in the desired column (this "selects" the column for the Widget). Then, drop the Widget in the desired location in that column.

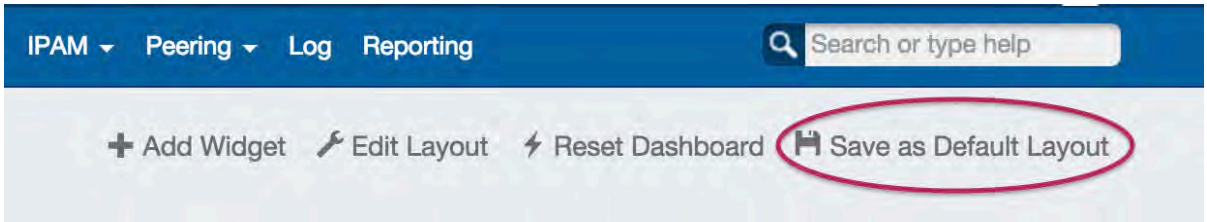
Saving a Dashboard

Personal dashboards are automatically saved after making layout or Widget edits (the exception is that Widget resizing must be manually saved -see Working with Dashboard Widgets).

Default Dashboards must be manually saved after edits for those changes to apply to all dashboard users.

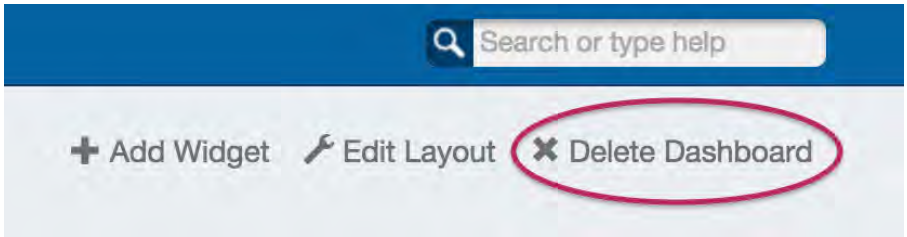
Admin users can save Default Dashboard changes by clicking "Save as Default Layout" at the top right section of the dashboard page, and the

changes will be applied for all users.



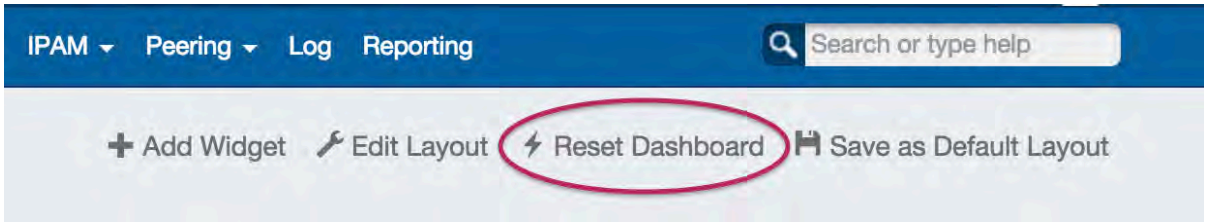
Deleting a Dashboard

Personal dashboards may be deleted by clicking the "Delete Dashboard" button at the top right of the page. Default Dashboards may not be deleted.



Reset the Default Dashboard

The Default Dashboard may be reset to the ProVision default dashboard by clicking "Reset Dashboard". This will remove any admin changes, and set the Default dashboard to its original state. This action may only be made by admin users.



Next Step: Widgets

Continue on to Working with Dashboard Widgets for information on each Widget, as well as how to add, move, update, and delete individual Widgets.

- Working with Dashboard Widgets

Working with Dashboard Widgets

Dashboard Widgets

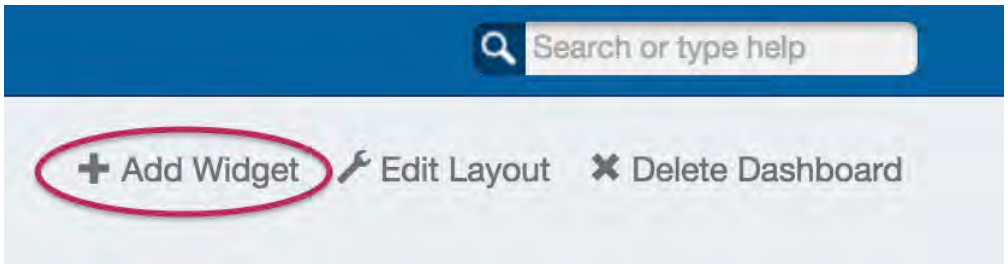
The previous section, Customizing the Dashboard, gave a high level overview of the dashboard types, how to create new personal dashboards, edit the dashboard layout, and remove personal dashboards. This section will give an overview of how to work with individual Widgets, and options available in each Widget type.

Before you begin working with individual Widgets, make sure that you have reviewed the information contained in Customizing the Dashboard and are familiar with the tasks outlined there.

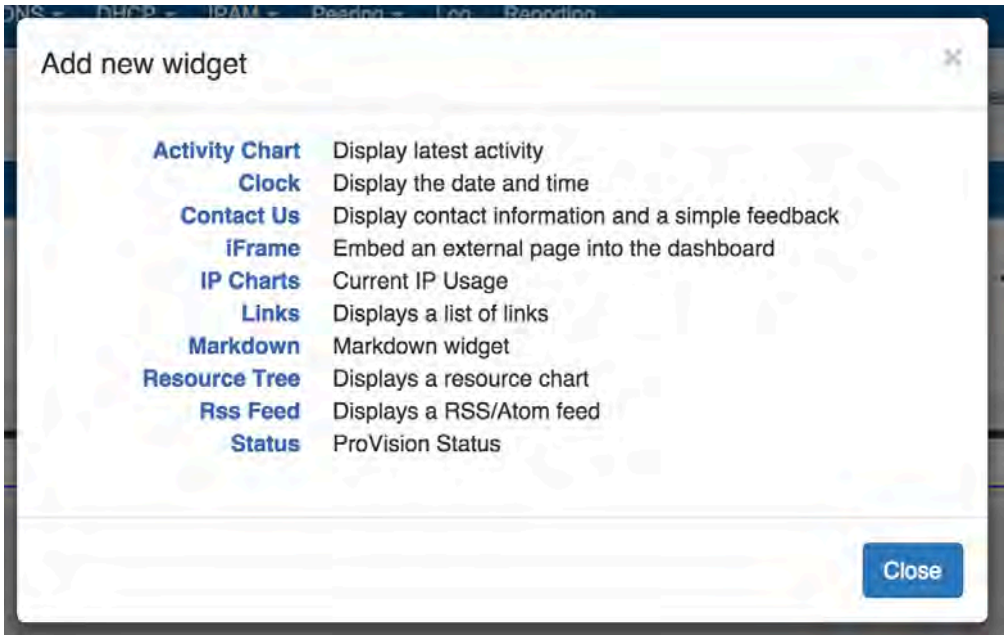
Working with Widgets

Adding Dashboard Widgets

To add a widget to personalize your dashboard, click on the "Add Widget" button at the top of the page.



Then, click on the name of the Widget you want to appear on your page. It will automatically be added into the leftmost column of your page.

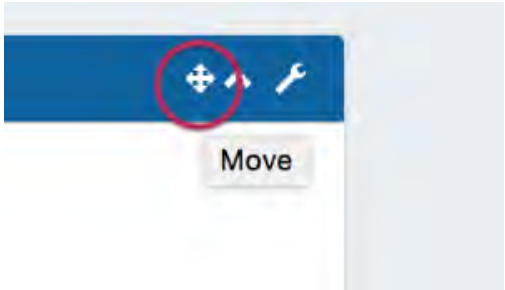


Some Widgets may only have one instance per dashboard, whereas other Widgets may be able to have multiple instances per dashboard. The Add New Widget dialog will only show you the widgets you have available to add.

For detailed information on each individual Widget, view the Available Widgets section on this page.

Move a Widget:

Individual Widgets may be moved around the page by clicking and dragging on the "Move" icon on the right side of the Widget header.

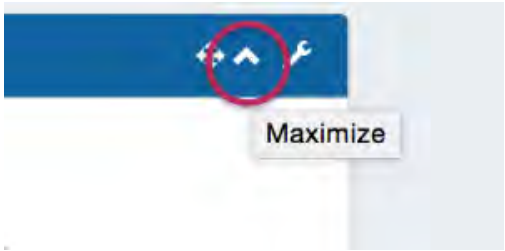


While mousing over the move icon (when the cursor changes to a hand icon) click and hold your left mouse button, then drag the widget up or down to the desired location.

To move the Widget to another column, drag the widget sideways to the column, then up to briefly hover over a Widget already in the desired column (this "selects" the column for the Widget). Then, drop the Widget in the desired location in that column.

Maximize / Pop-Out a Widget:

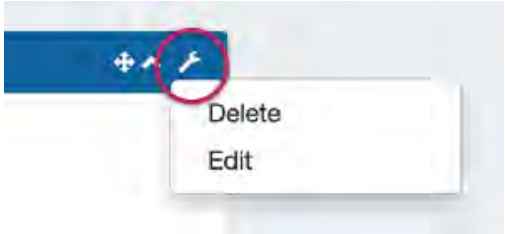
Widgets can temporarily be viewed at a larger size by hitting the "Maximize" icon on the Widget header. This will "pop out" the widget information to a larger page size for easier viewing.



While maximized, you may interact with the widget as normal. When done, hit the "Close" button.

Edit a Widget (Widget Action Menu):

To open the edit menu for a widget, click on the Action Menu (wrench icon) on the right side of the Widget header. A drop down menu will appear with some of the following options:



Edit: Opens an edit widget pop up, where widget-specific options may be changed. Options may include changing the title, links, applicable resources, and so forth depending on the particular Widget.

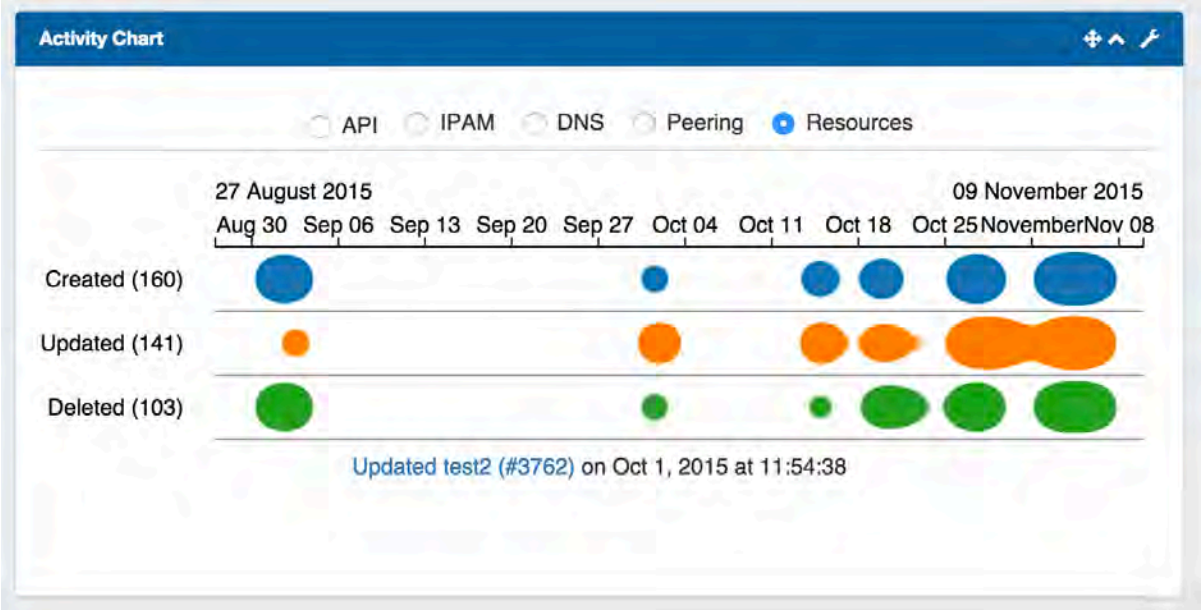
Resize: For Widgets able to be resized vertically, brings up a draggable black bar at the bottom of the widget. Drag the bar up or down to the desired widget height, then select the Action Menu again and click on "Save Resize" to save and complete the size edit.

Save Resize: Available only after the "Resize" action has been used. After resizing a Widget, selecting the Action Menu again and clicking on "Save Resize" will save the Widget size.

Delete: Deletes the widget.

Available Widgets

Activity Chart (Admin only)



Illustrates activity level for API, IPAM, DNS, Peering, and Resource areas of ProVision. The Activity Chart is only visible to Admin users. Only one Activity Chart Widget may exist on a dashboard.

Activity Chart Actions:

Radio Buttons: Select the desired ProVision activity area to view by clicking on the radio buttons (API, IPAM, DNS, Peering, or Resources).

Chart: While hovering over the chart, you can:

Use the Scroll Wheel to zoom in or out of the date/time detail level

Double click to zoom into the next date/time detail level

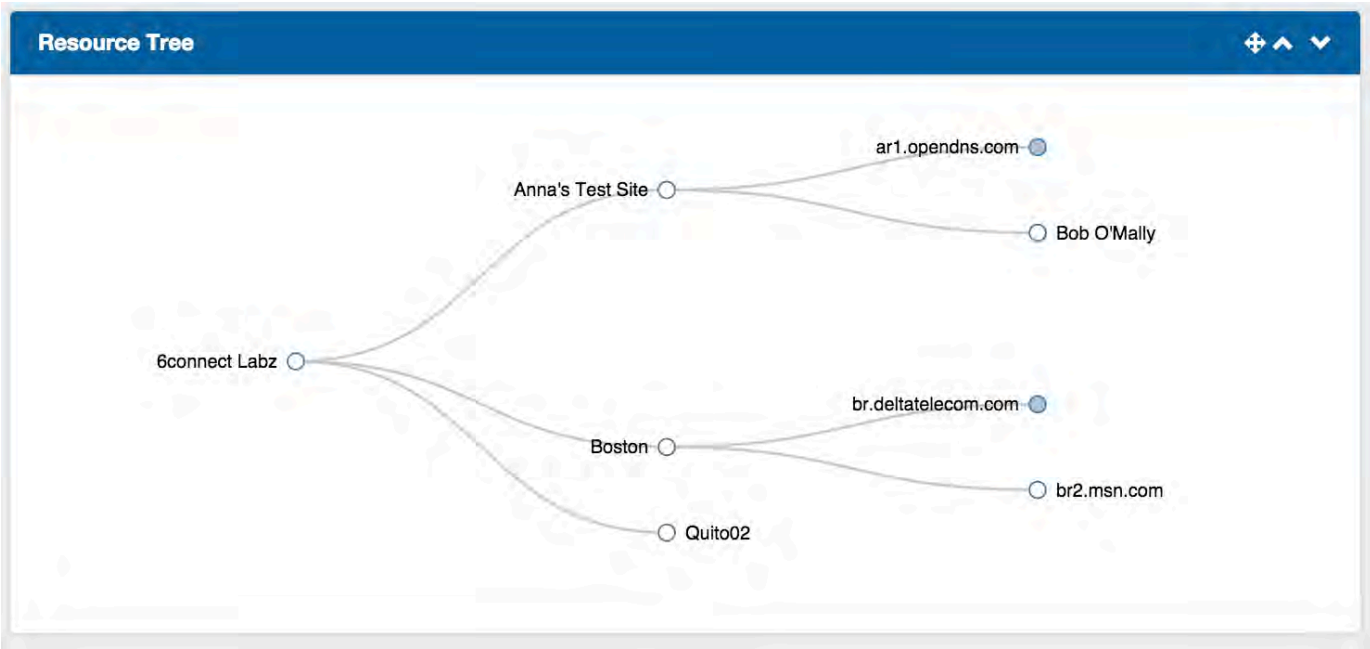
Click & Drag left or right to move to an earlier or later date/time at the current zoom level

Single Click on a bubble will show the activity detail, and resource / IP block links if applicable.

Activity Chart Edit Options:

Title: Edits the title of the Widget shown on the header.

Resource Tree



Interactive graphical display of the resource structure for a parent resource.

Resource Tree Actions:

Users may:

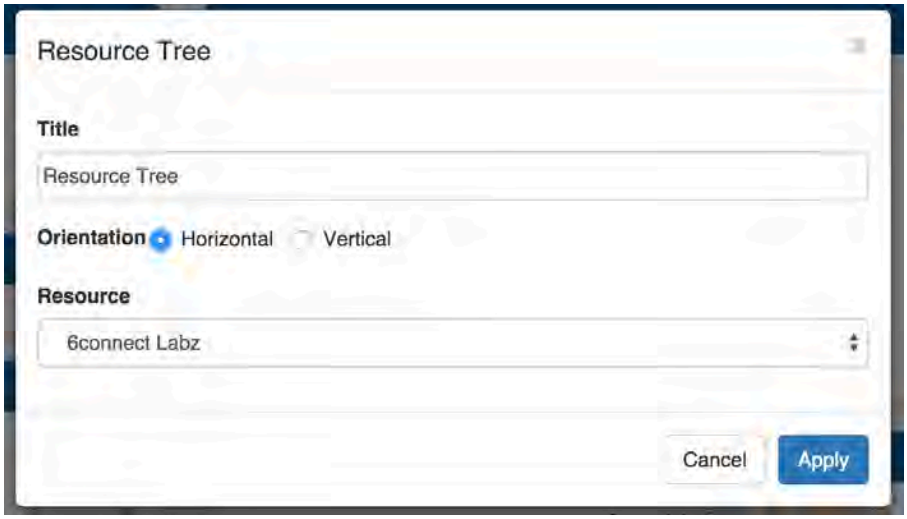
Click & Drag to move the chart within the Widget,

Use the Scroll Wheel to zoom in or out

Click on Filled dots to expand the tree to show the children of that resource.

Double click on a resource in the tree to give you the option to navigate to the resource's entry page.

Resource Tree Edit Options:



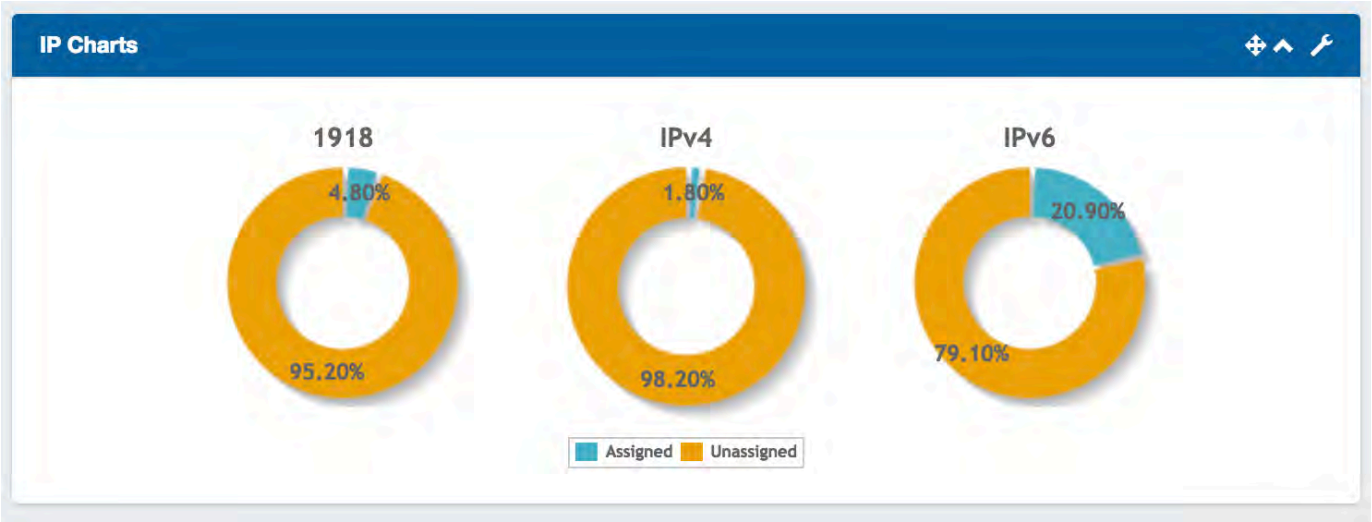
Title: Edits the title of the Widget shown on the header.

Orientation: Select "Horizontal" to view the tree in horizontal form, with the top level resource on the leftmost side of the widget, and children expanding to the right. "Vertical" will show the tree with the top level parent at the top of the widget, with children expanding downward.

Resource: A text search box to select the resource that will be the top level resource for the tree.

When done editing, hit "Apply" to save your changes, or "Cancel" to exit without saving.

IP Charts

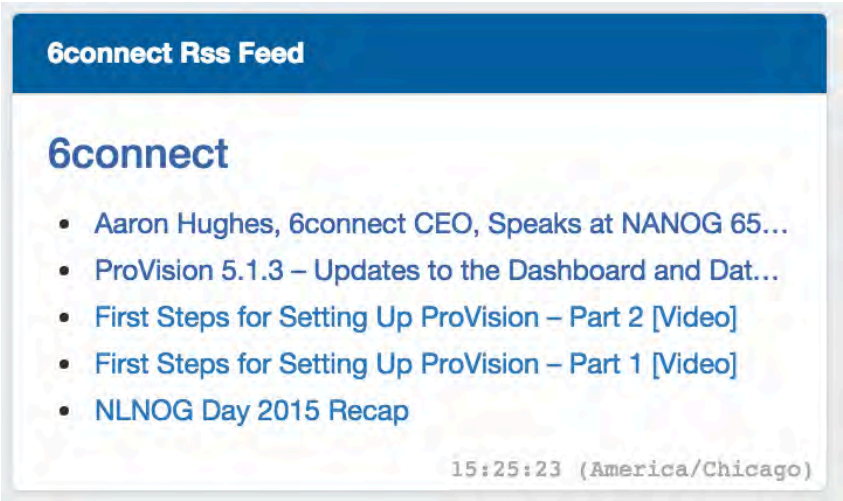


Illustrates the percentage of assigned vs unassigned hosts for 1918 / IPv4 / IPv6 space out of the total available hosts in ProVision viewable by the user.

IP Charts Edit Options:

Title: Edits the title of the Widget shown on the header.

RSS Feed



Shows the most recent five entries of a selected RSS Feed. Feed links open in a new window when clicked.

RSS Feed Edit Options:

Title: Edits the title of the Widget shown on the header.

Feed URL: Edits the URL of the RSS Feed to show.

Status

Shows 6connect ProVision status information, including number of user accounts, number of admin accounts, current version number, and a link to the "Coming Soon" section of the ProVision documentation.

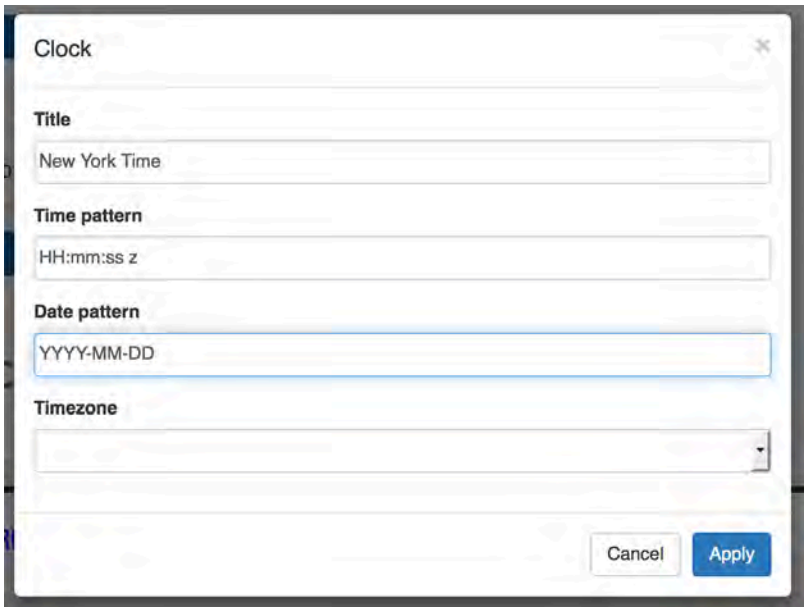
Status Edit Options:

Title: Edits the title of the Widget shown on the header.

Clock

Shows the current time for the selected time zone.

Clock Edit Options:



Title: Edits the title of the Widget shown on the header.

Time Pattern: Sets the desired format for the current time. Example: "HH:mm:ss z" shows hours, then minutes, then seconds, then time zone.

Date Pattern: Sets the desired format for the current date. Example: "YYYY-MM-DD" shows 4 digit year, then month, then day.

Time Zone: The desired time zone to show the current time for.

Click here for detail on date / time patterns...

Time / Date Patterns
For applicable time / date patterns, see http://momentjs.com/docs/#/displaying/format/

Some common date patterns might be:

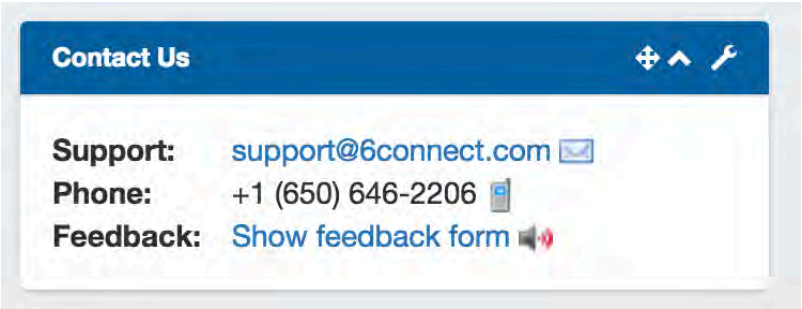
Display	Pattern
Wednesday, October 10th, 2015	dddd, MMMM Do, YYYY
10/21/2015	l

October 21, 2015	MMMM DD, YYYY
10-21-15	MM-DD-YY
2015-10-21	YYYY-MM-DD

Some common time patterns might be:

Display	Pattern
15:36:14 EDT	HH:mm:ss z
3:36:12 PM EDT	h:mm:ss A z
03:36:12 pm EDT	hh:mm:ss a z
15:36 -04:00	HH:mm Z

Contact Us



Provides support email, phone, and feedback form information. Only one Contact Us Widget may exist on a dashboard.

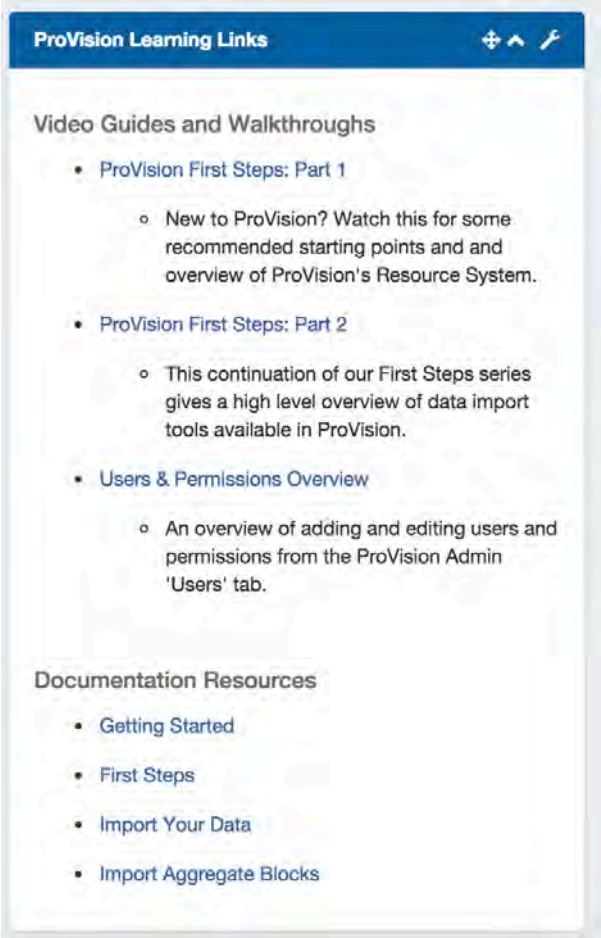
Contact Us Edit Options:

Title: Edits the title of the Widget shown on the header.

Feedback Email: Sets the desired email address for the feedback form.

> Note: Support phone number and email address are set from the ProVision Admin settings.

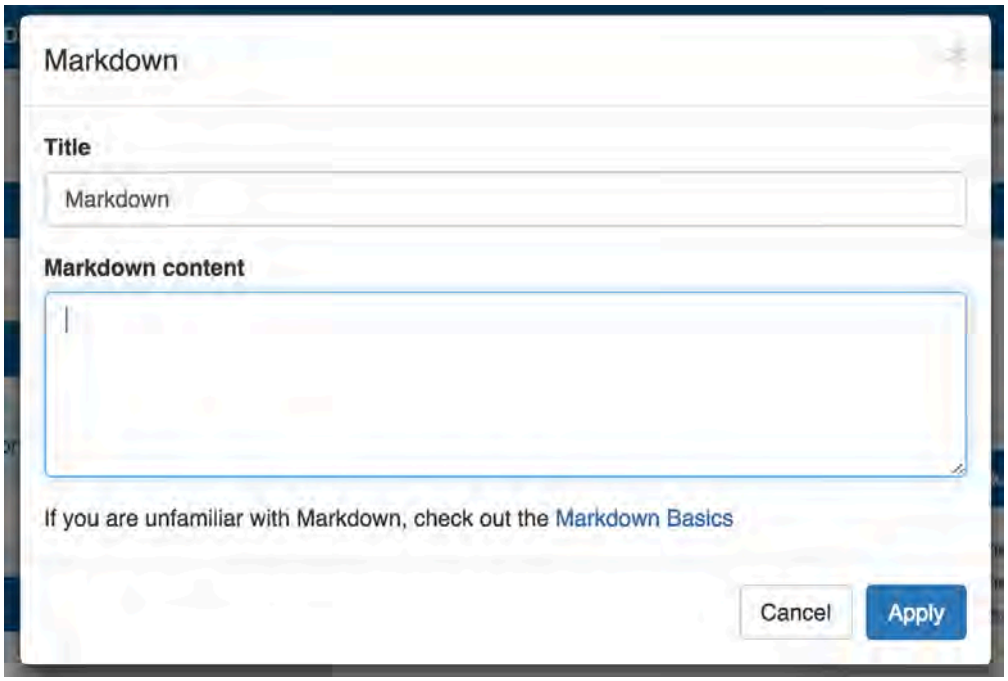
Markdown



The Markdown Widget is a blank slate where users may add links, text, or other content through use of the Markdown language. Multiple Markdown Widgets may exist on a dashboard.

> If you are unfamiliar with Markdown, check out Markdown Basics

Markdown Edit Options:



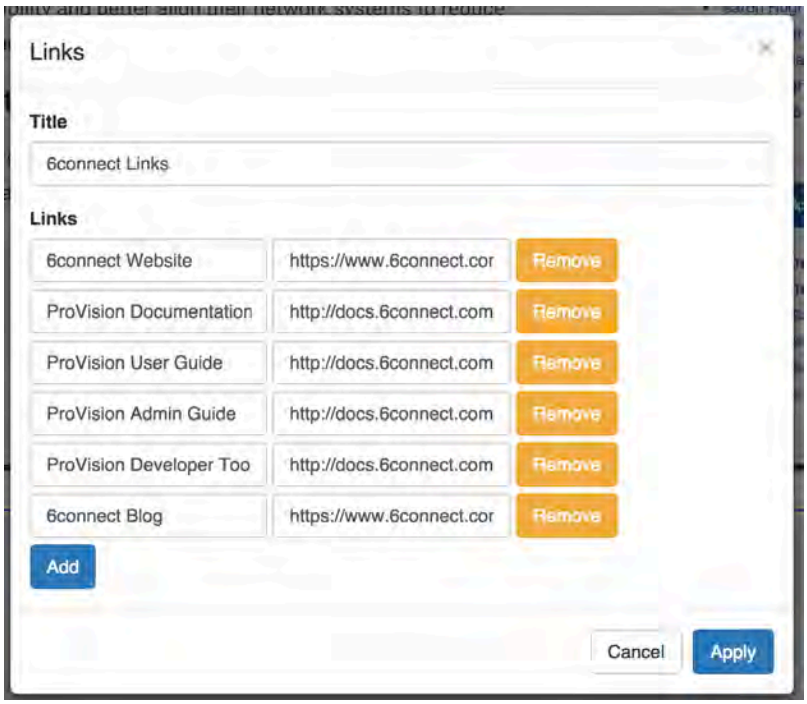
Title: Edits the title of the Widget shown on the header.

Markdown Content: Area to enter the markdown content.

Links

Provides a list of website links. Multiple Links Widgets may exist on a dashboard.

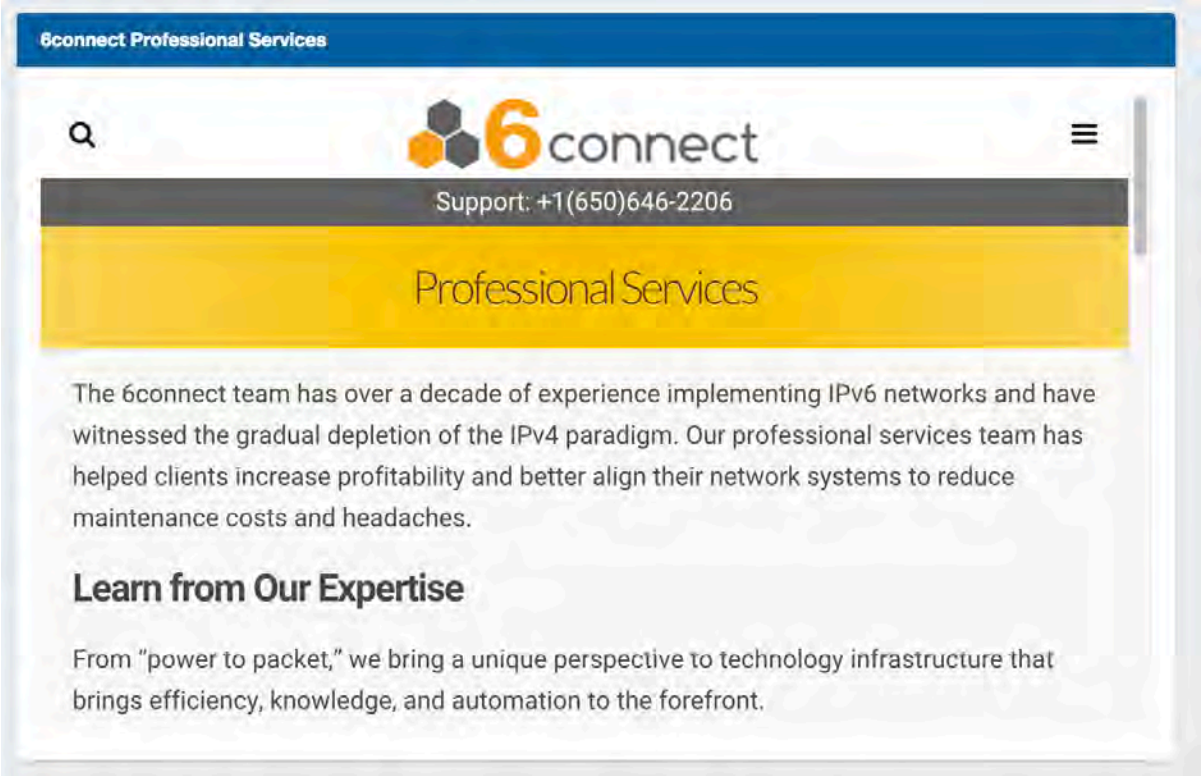
Link Edit Options:



Title: Edits the title of the Widget shown on the header.

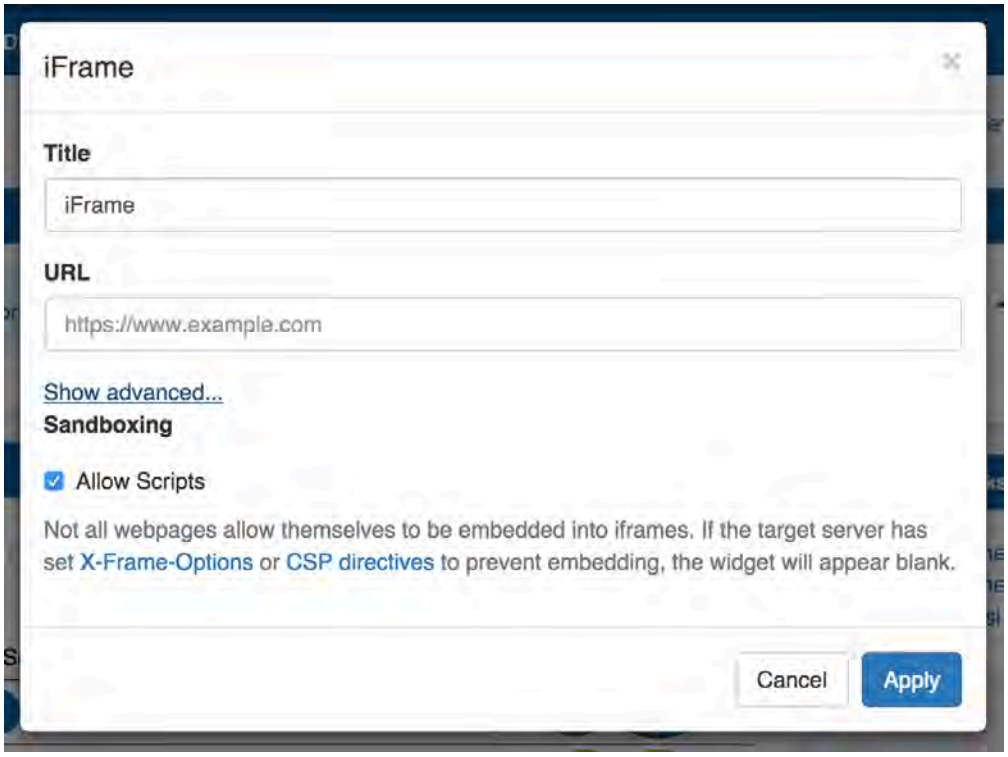
Links: Enter the desired link text, then the URL of the website to link. Click "Add" to add a new link, or "Remove" to remove an individual link. Hit "Apply" to save your changes.

iFrame



Set a https URL to be viewed in an iFrame. Useful to view stats or commonly referenced websites.

iFrame Edit Options:



Title: Edits the title of the Widget shown on the header.

URL: Sets the desired https address of the website to display in the frame

Advanced:

Sandboxing: Allow Scripts: Check to allow scripts.

> Not all webpages allow themselves to be embedded into iframes. If the target server has set X-Frame-Options or CSP directives to prevent embedding, the widget will appear blank.

Working with Resources

Resources

What is a Resource?

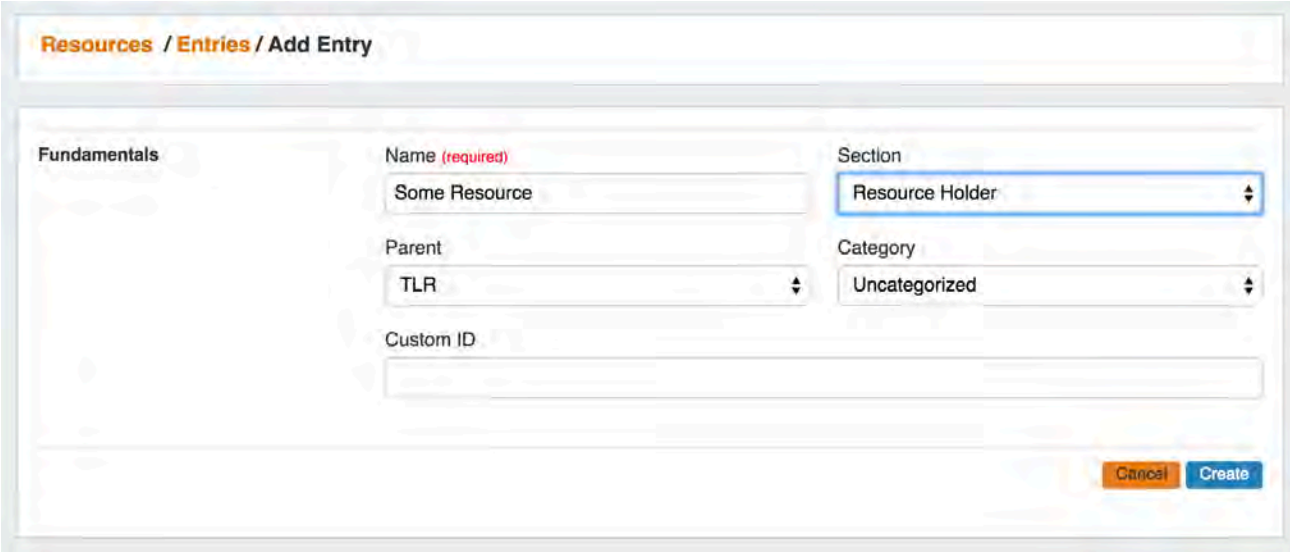
The "Resource" system is tied to the Permissions structure. What this means is that you get granular control on a resource level and can create groups around a single resource or even groups of resources. Since Resources can inherit permissions from others - it can be an easy way to categorize generic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

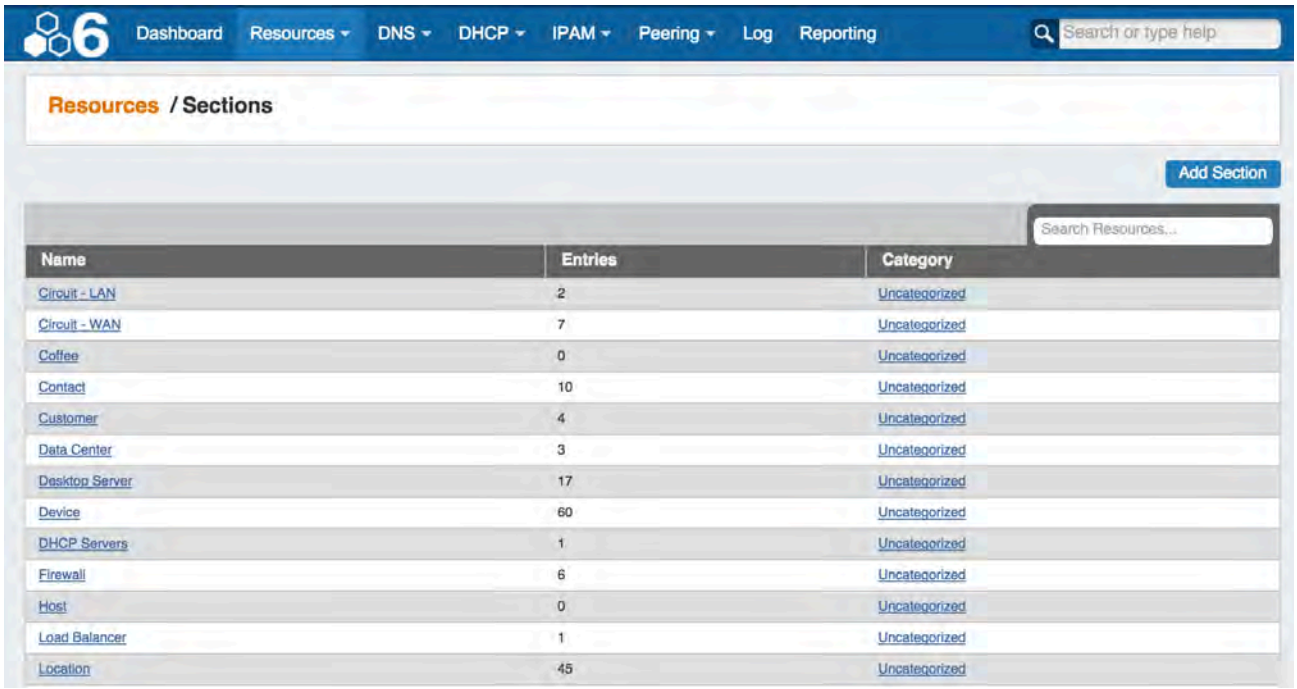
How to Work with Resources?

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:

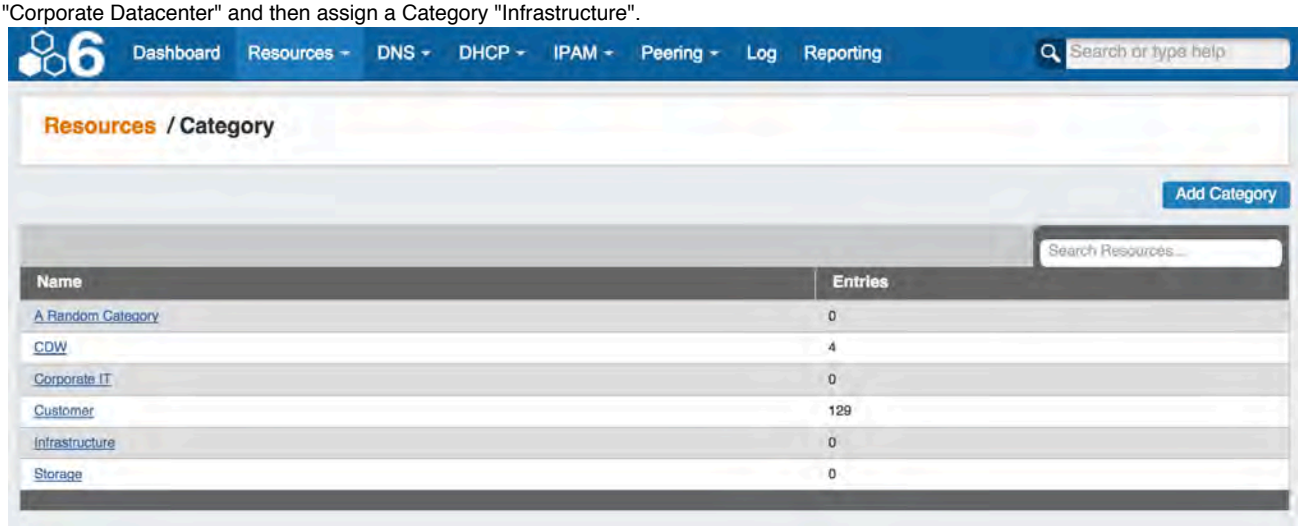
- **Resource Entries:** These are the actual Resource names. When you click the "Add Entry" button you can customize various elements of the entry and assign the Parent Resource, Section and Category from their respective dropdown menus. You may also add a Custom ID if desired. When done, hitting "Create" will pull up the field set for the chosen Section 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.



- **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".



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

Additional Information

- Working with Entries
- Customizing Sections
- Customizing Fields
- Gadgets
  - XML Specifications

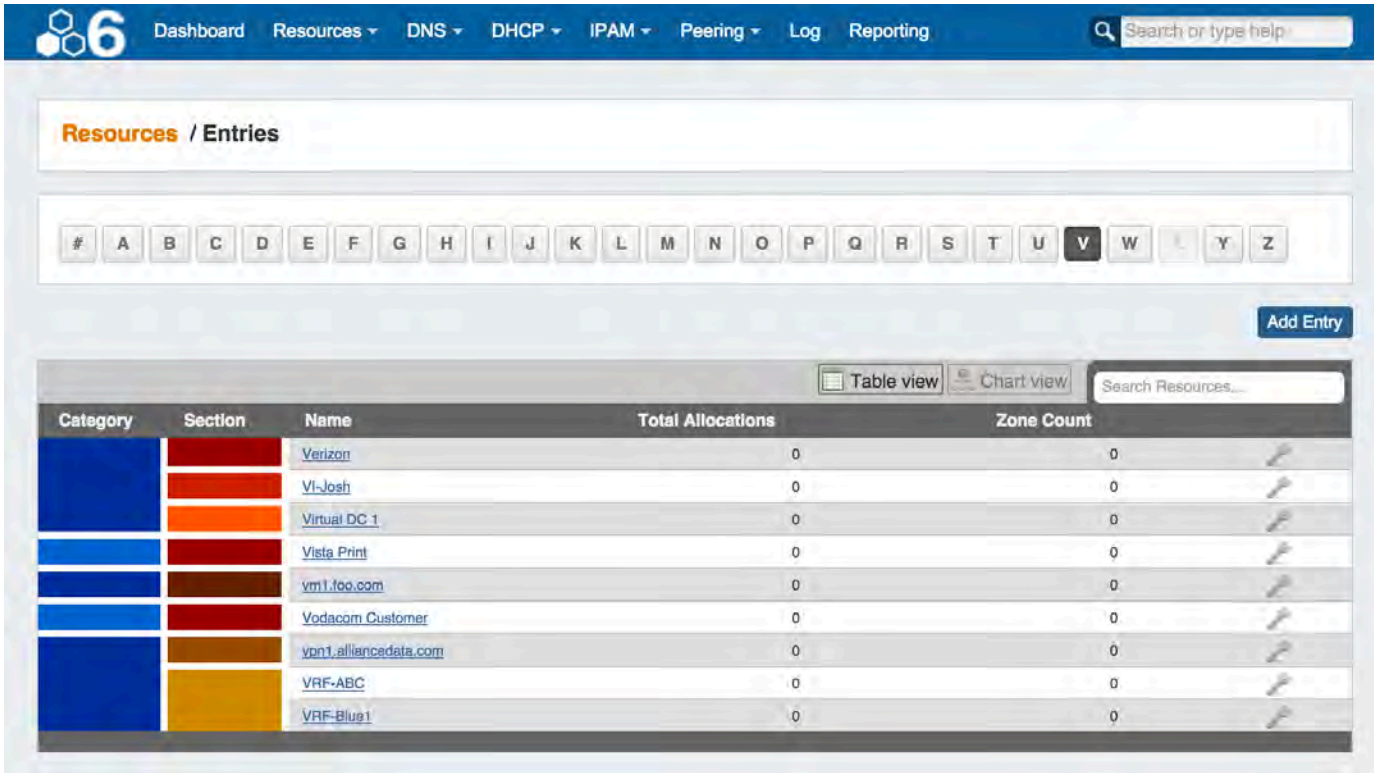
- Contact Manager

Working with Entries

Working with Entries

The list of Resource Entries is under the **Resource** Tab. To access it, you may either click on the Resource Tab, or select "Entries" from the Resource Tab dropdown menu.

Resource Tab / Entry List User Interface



1) Alphabetical Filter: Entries are organized by the first letter of the name. Click on a letter to see entries starting with that letter.

2) Add Entry Button: Click to add a new entry.

3) Pagination: Click to view additional pages of Entries under the selected letter.

4) Viewing Options: Select "Table View" or "Chart View" .

Table View (shown above), lists the Entries, Allocations, and Zone Count along with color-corded Category and Section information.

Chart View graphically shows the entries in filterable, hierarchy form.

5) Search Box: This text box allows the user to enter in criteria to filter the list of Entries.

6) Category: Color-coded Category indicator. The Category name will appear upon mouseover. Clicking on the Category box when the hand icon appears will redirect to a filtered list of entries of that Category.

7) Section: Color-coded Section indicator. The Section name will appear upon mouseover. Clicking on the Section box when the hand icon appears will redirect to a filtered list of entries of that Section.

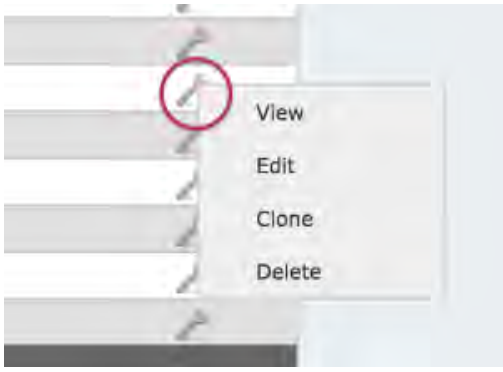
8) Name: A list of the Resource Entry names. Clicking on a name will take the user to that resource's individual entry page.

9) Allocations: The number of IP blocks assigned to that resource.

10) Zone Count: The number of zones assigned to that resource.

11) Action Menu: The Action Menu (wrench icon) gives a list of additional actions to perform on the zone

Entry List Action Menu



Clicking on the Action Menu in the Entry List View will bring up the following options:

View: Opens the resource's Entry page

Edit: Opens to the resource's Edit page

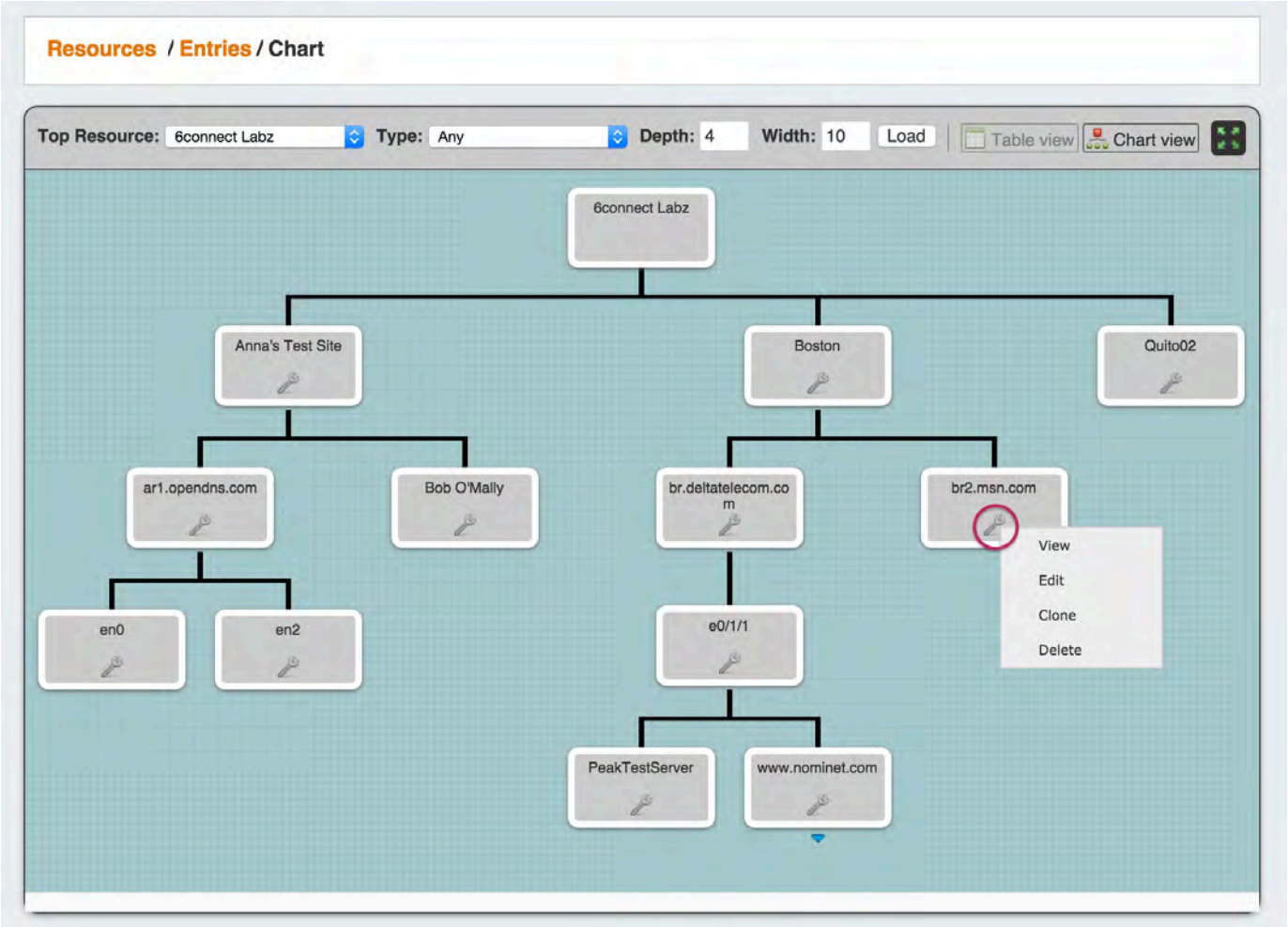
Clone: Opens the clone resource page for the selected resource

Delete: Deletes the resource, if appropriate permissions exist.

Chart View

Chart View illustrates the resources created in a graphical hierarchy. Select the Top Resource from the dropdown and the top of the chart, and / or select the Section type to further limit the view, if desired.

Expand the view of the chart by changing the Depth and Width fields to the number of resources to view in that axis, then hit "Load". You may also maximize the view by clicking the maximize button (top right of the chart, with four arrows).



Clicking on the Action Menu in the Chart View will bring up the following options:

View: Opens the resource's Entry page

Edit: Opens to the resource's Edit page

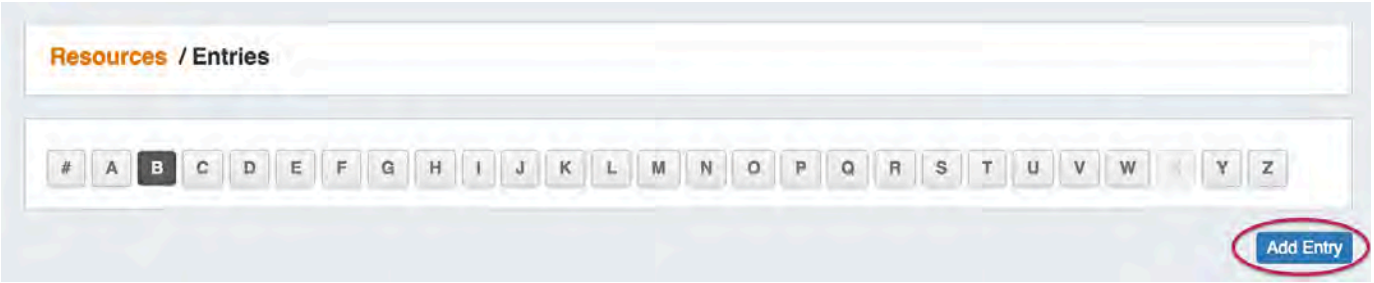
Clone: Opens the clone resource page for the selected resource

Delete: Deletes the resource, if appropriate permissions exist.

From Chart View, you can return to the Table View by clicking on "Table View" at the top of the chart.

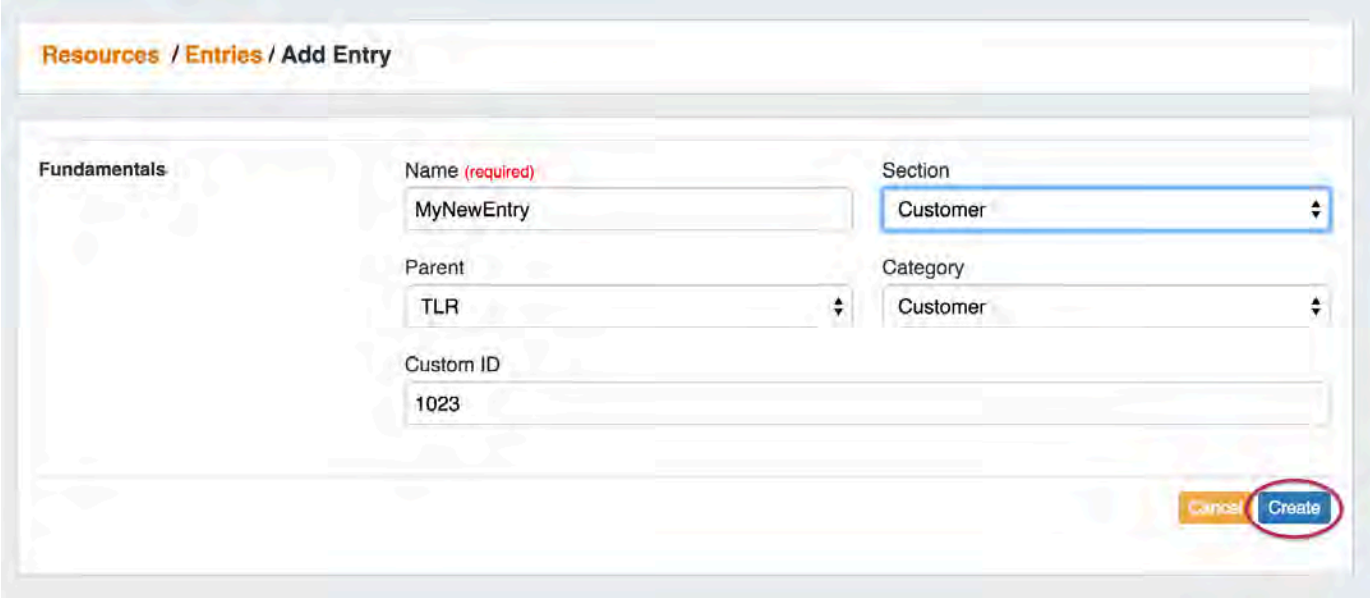
Create an Entry

1) To create a new Entry, click on the "Add Entry" button from the Resource Tab / Entry List Page.

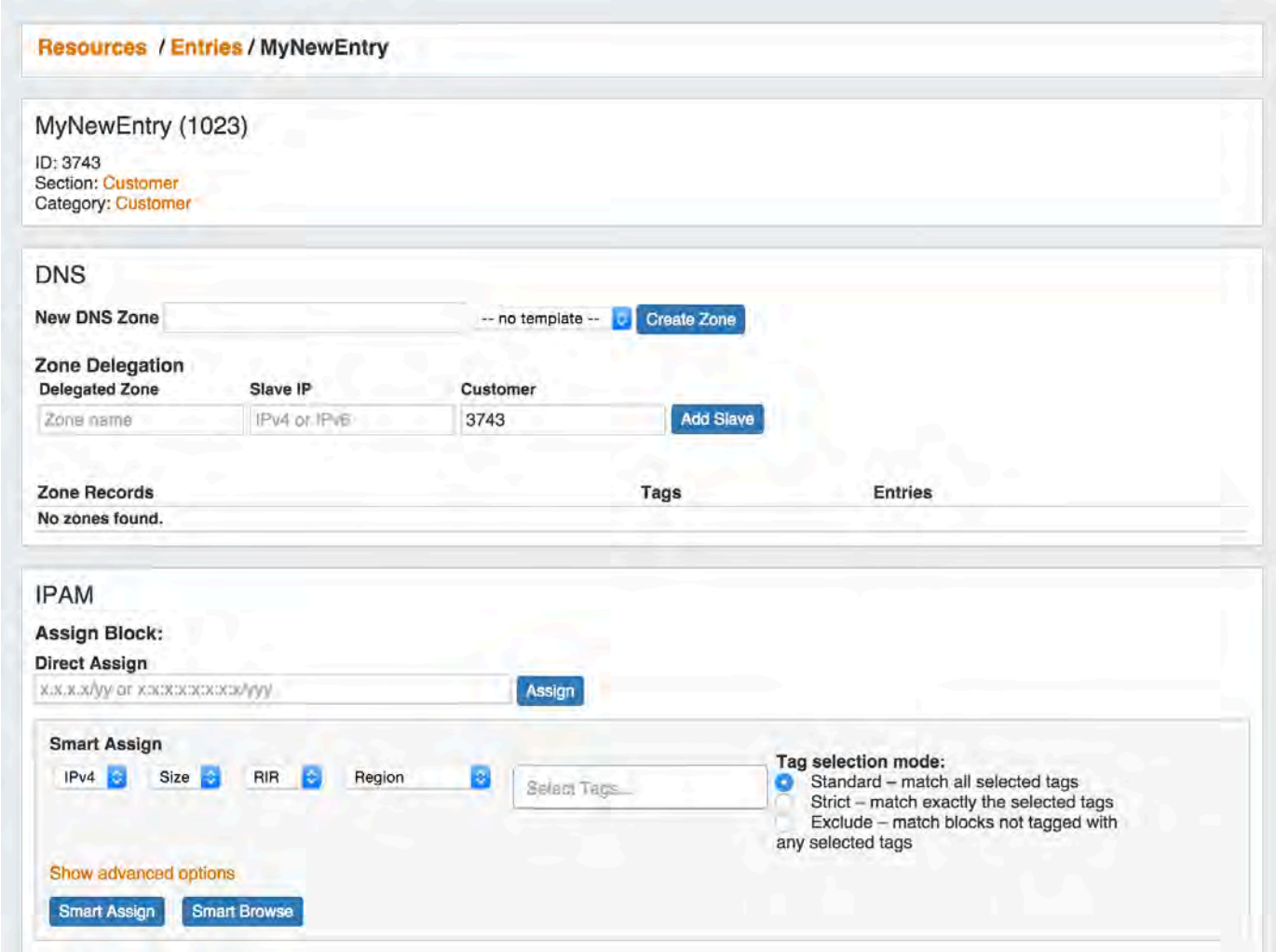


This will take you to the Add Entry page.

2) Fill in the Name of your new Resource Entry, select a Section (the chosen Section determines what gadgets will show on your new resource's Entry page), the Parent Resource, Category, and enter a Custom ID number if desired. When complete, click the "Create" button. If you wish to exit without saving your changes, click "Cancel".



Once the Entry is created, the Resource Entry page appears for that Resource, with the gadgets applicable to the selected Section.



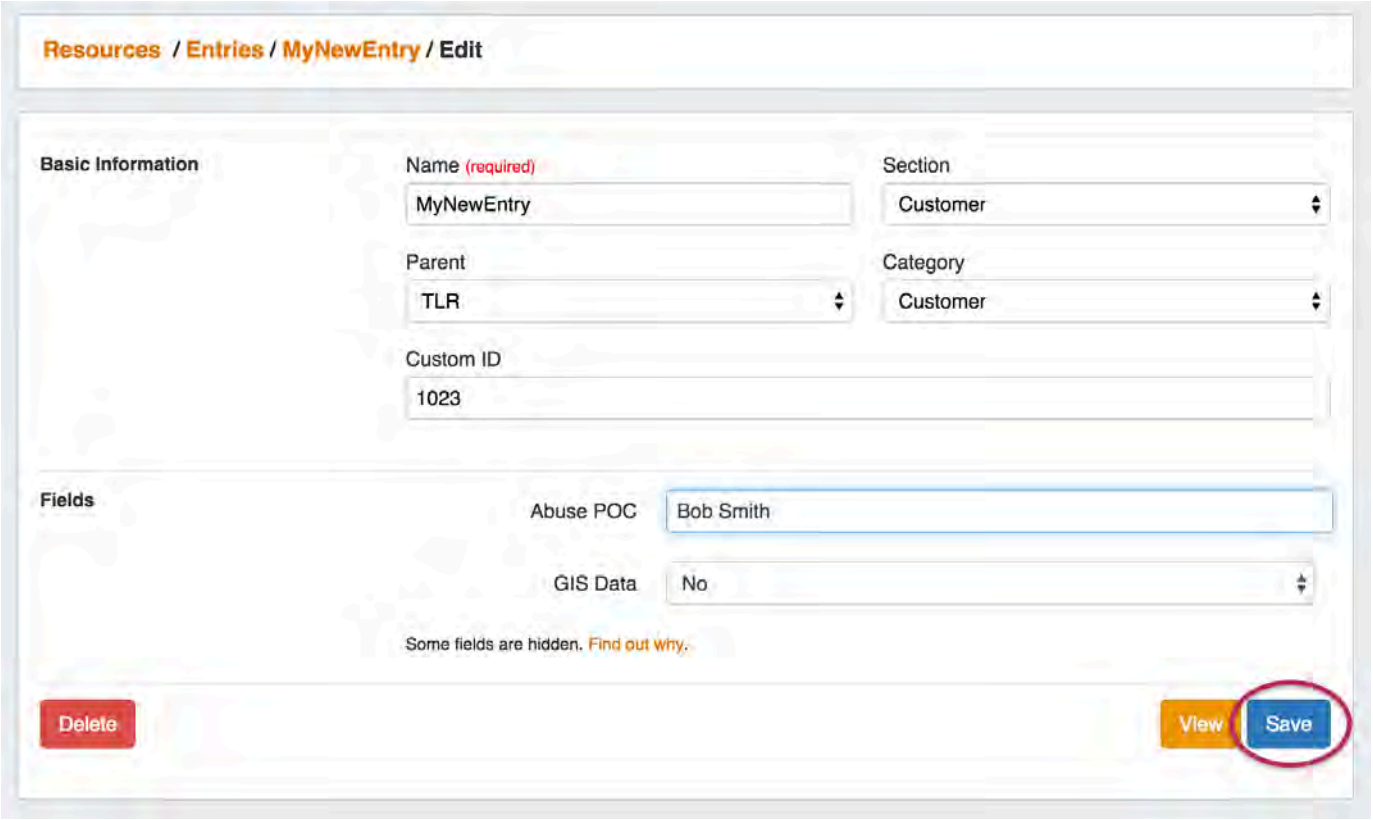
From here, you can work with your selected Gadgets, performing tasks such as assigning DNS zones, IP Blocks, assign Contacts, and more. See the Gadgets Page for detail on working with individual Gadgets.

Edit or Delete an Entry

To edit or delete an Entry, go to the bottom of that resource's Entry Page, and select "Edit".

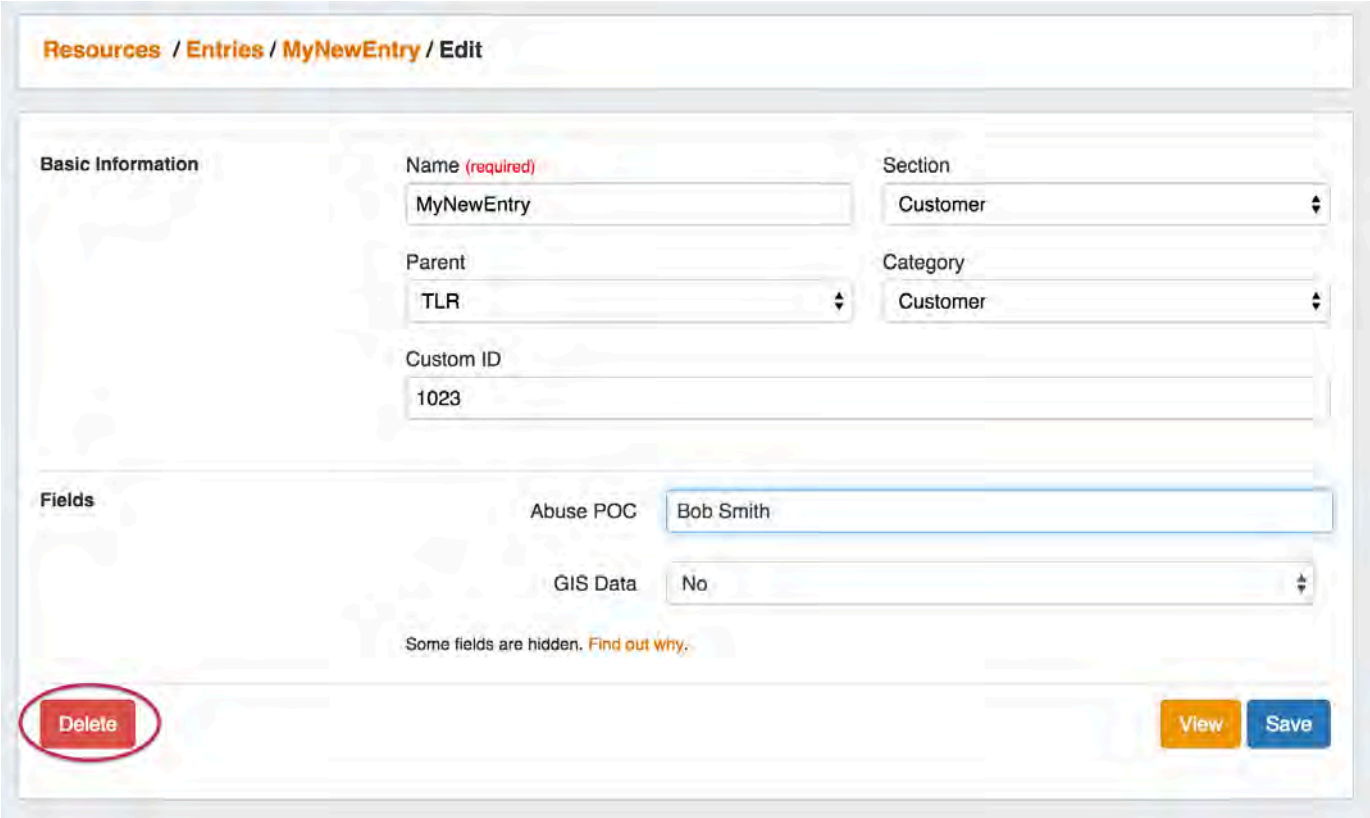
This will bring up the Edit Entry Page, where you can edit basic information such as Name, Section, Parent, Category, or Custom ID. If Fields are associated with the Section type, those may be edited as well.

When done, hit the "Save" button. If information was added to Fields, that information will now show up on the Entry Page.



If you wish to delete the Entry, select the "Delete" button.

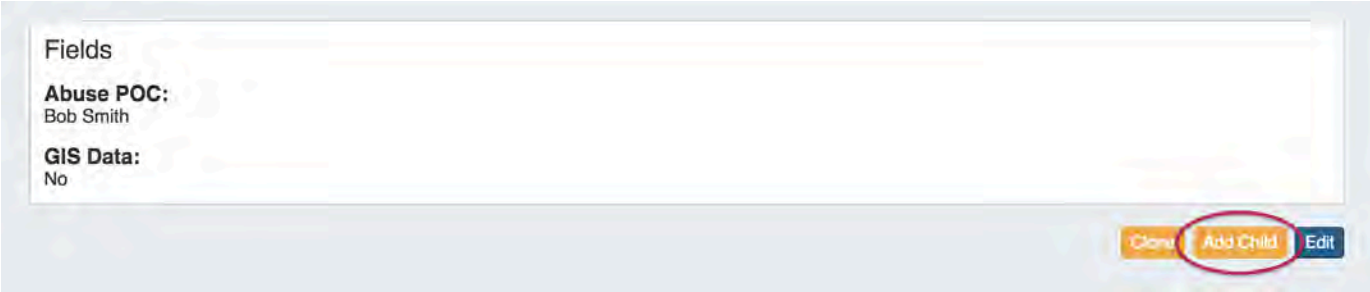
> **Note**
> If an entry has children, the delete option will not be available. Child Entries must be deleted before a Parent Entry can be deleted.



Add Child Entries

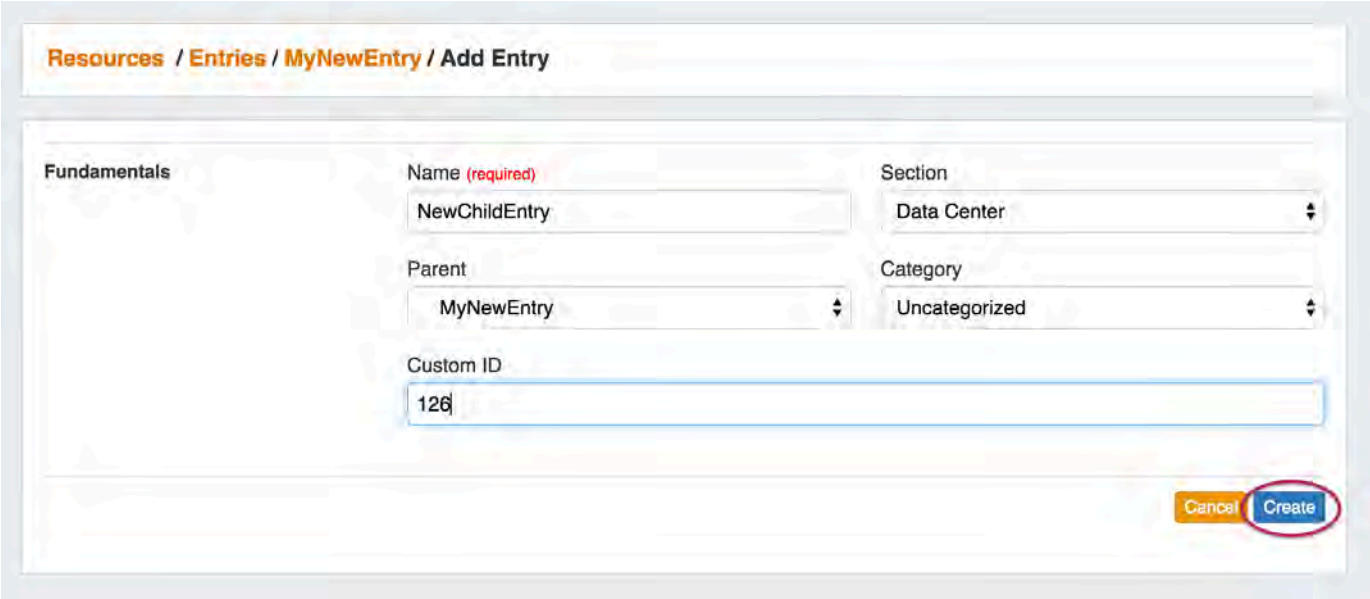
Adding a Child Entry creates a resource underneath the current resource in the hierarchy.

To add a Child Entry, at the bottom of the Resource Entry page, click "Add Child".



This brings up the Add Entry dialog. Looking at the top navigation breadcrumbs, you can see that we are adding an Entry underneath the resource "MyNewEntry".

Fill out the Name, Section, Category, and Custom ID (if desired), and hit the "Create" button. The Parent field is pre-selected for you.



Once created, the Child Entry will show in a list at the bottom of the Parent Entry page.

Resource Clone

Cloning an Entry duplicates the Entry and, if desired, any Child Entries existing under it.

This feature can be used to quickly and efficiently create multiple resources with same format, field information, or similar hierarchy structure. One example may be a datacenter with x number racks with y number of servers on each. One rack, with its servers as child entries could be created, and then cloned until the necessary quantity is reached. As text field information is copied during the clone, changes in individual server information would still need reviewed and updated. Think of cloning as creating a template from an existing Resource Entry.

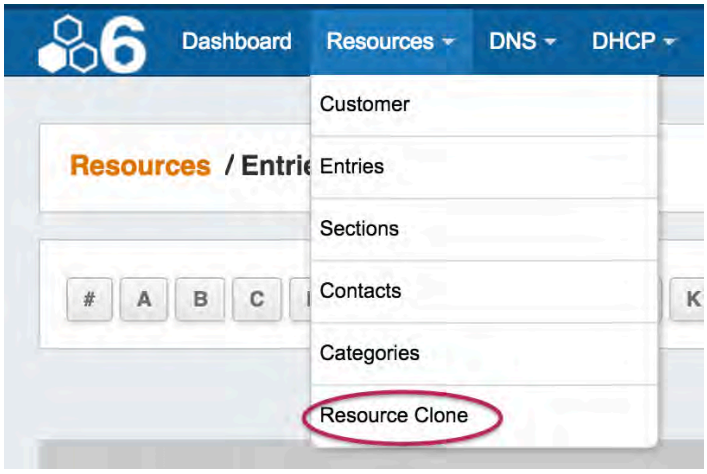
Information Cloned: Format of the original Resource (which gadgets are enabled, location, fields), contact / tech information, field information, and (if selected) Child Entries, and Child Entry sections, fields, and field informaiton - in general, items that are chosen or input at the entry creation, or input into text fields.

Information not Cloned: IP Blocks, DNS Zones, uploaded documents - in general, items that are *assigned* to the entry rather than input into fields.

To Clone an Entry

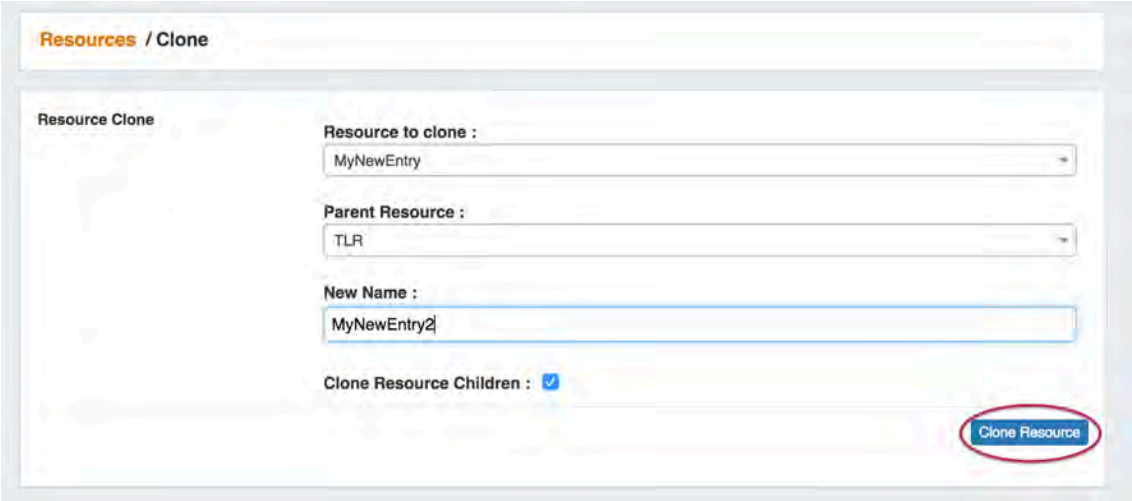
The Resource Clone feature may be accessed from either the **Resources** Tab dropdown menu, or from a Resource Entry page.

From the **Resources** Tab dropdown, select "Resource Clone". This will take you to the Resource Clone page.



You may also select "Clone" from the bottom of a Resource Entry page, in which case the Entry page you clicked "Clone" from will be automatically selected as the resource to clone.

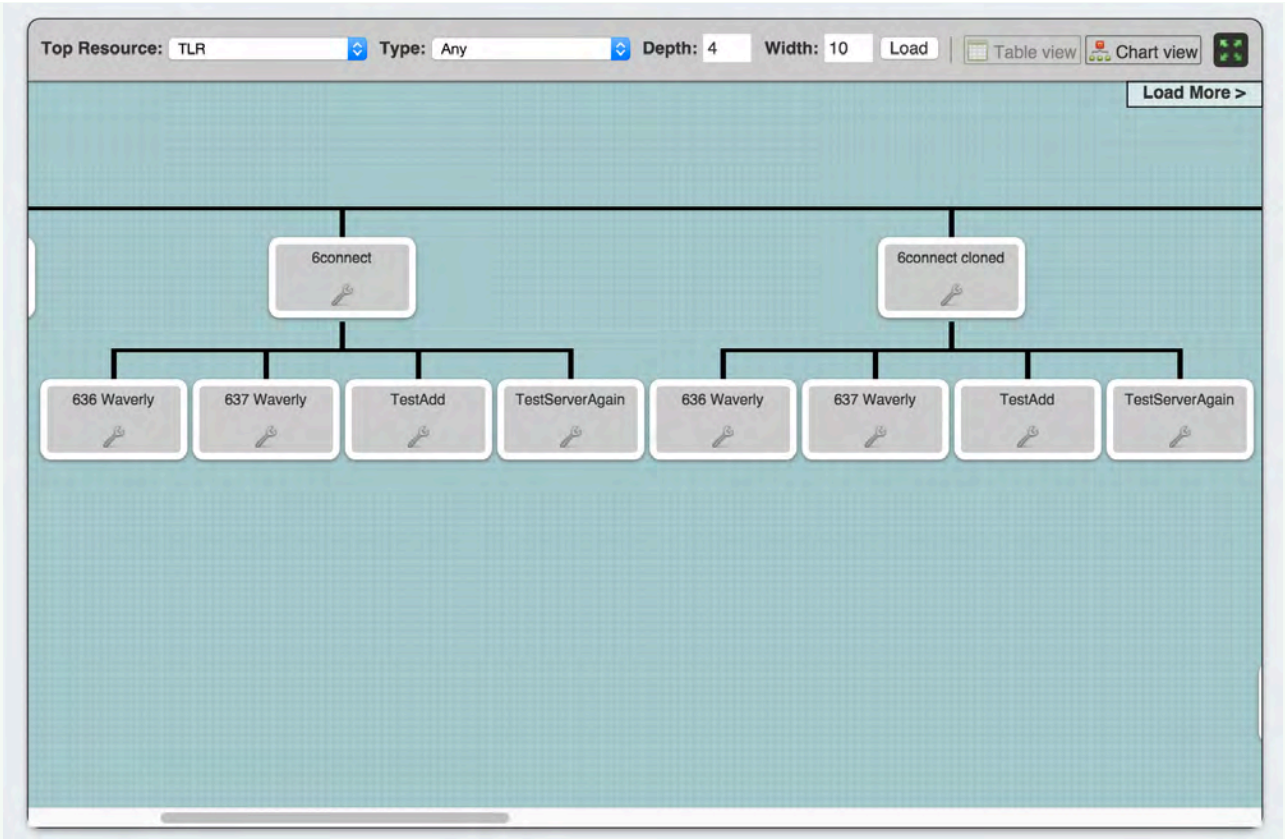
Once on the Resource Clone Page, enter the Resource to clone, Parent Resource, and the New Name for the clone. If you wish for all children of that resource (and their children, if applicable) to be cloned, check the "Clone Resource Children" checkbox. When complete, click the "Clone Resource" button.



To repeatedly clone the resource, simply change the "New Name" field (if you want the subsequent clones to have a different name), and click "Clone Resource" again. You may repeatedly click "Clone Resource" and each click will produce a new clone.

When done, you may click on the link provided for the most recently created clone, or hit the XXXX button to return to the parent resource.

> **Tip**
> Review the cloned resources using the chart view to see the hierarchy structure!


Top Resource:
TLR
Type:
Any
Depth:
4
Width:
10
Load
Table view
Chart view
Load More >
6connect
6connect cloned
636 Waverly
637 Waverly
TestAdd
TestServerAgain
636 Waverly
637 Waverly
TestAdd
TestServerAgain

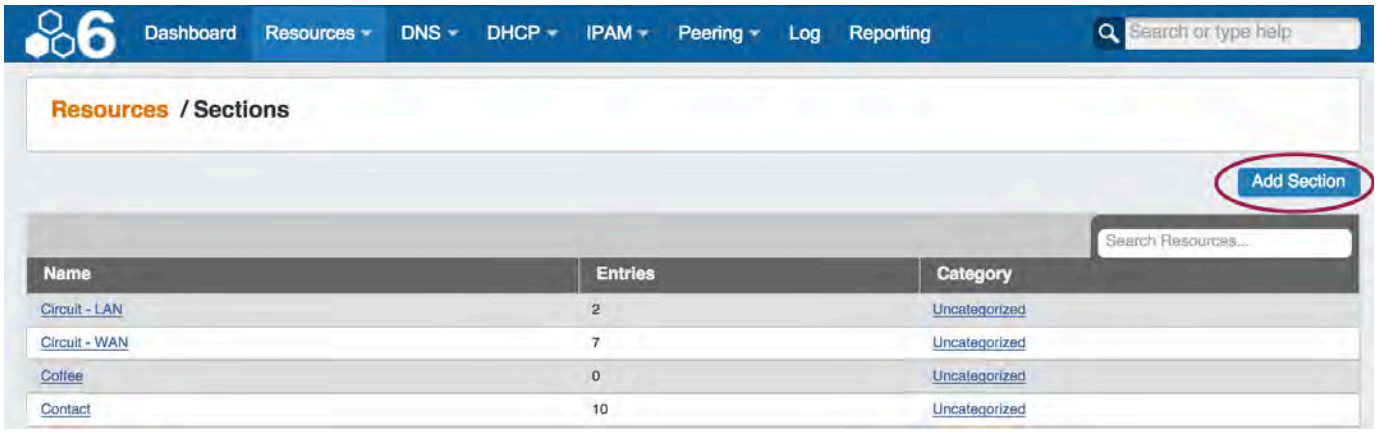
Customizing Sections

Customizing Sections

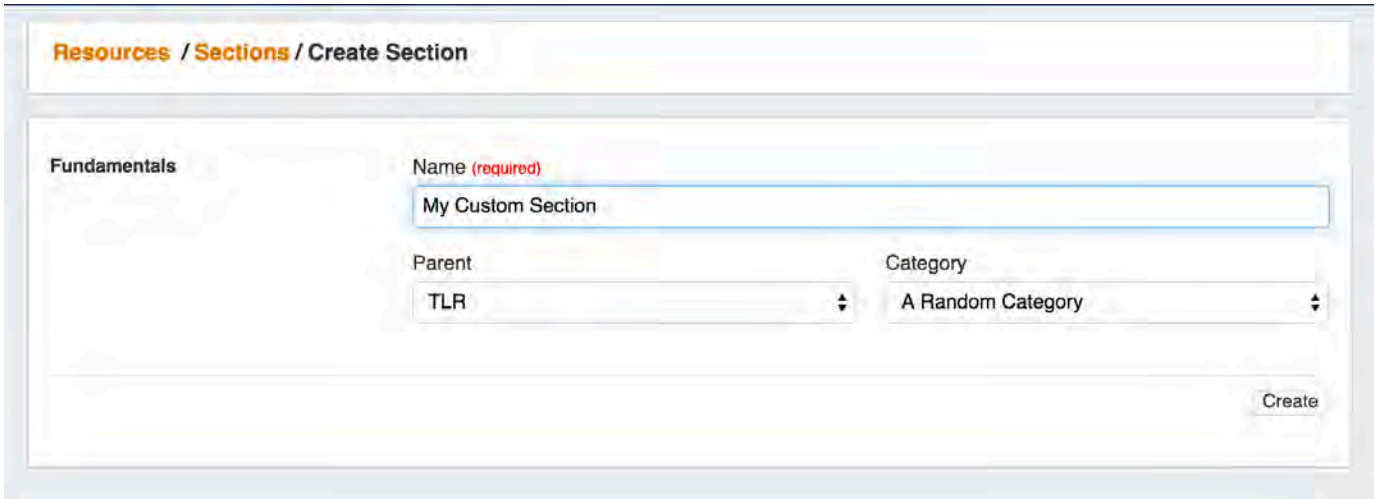
You can create as many Sections as you wish (Firewall, Server, VM, Virtual Interface, etc.) and customize the fields that you care about for each Section. For example, you may not need to track the console port for your virtual firewall, so you would simply not use that field for the "Virtual Firewall"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

Step 1: Create a New Section

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



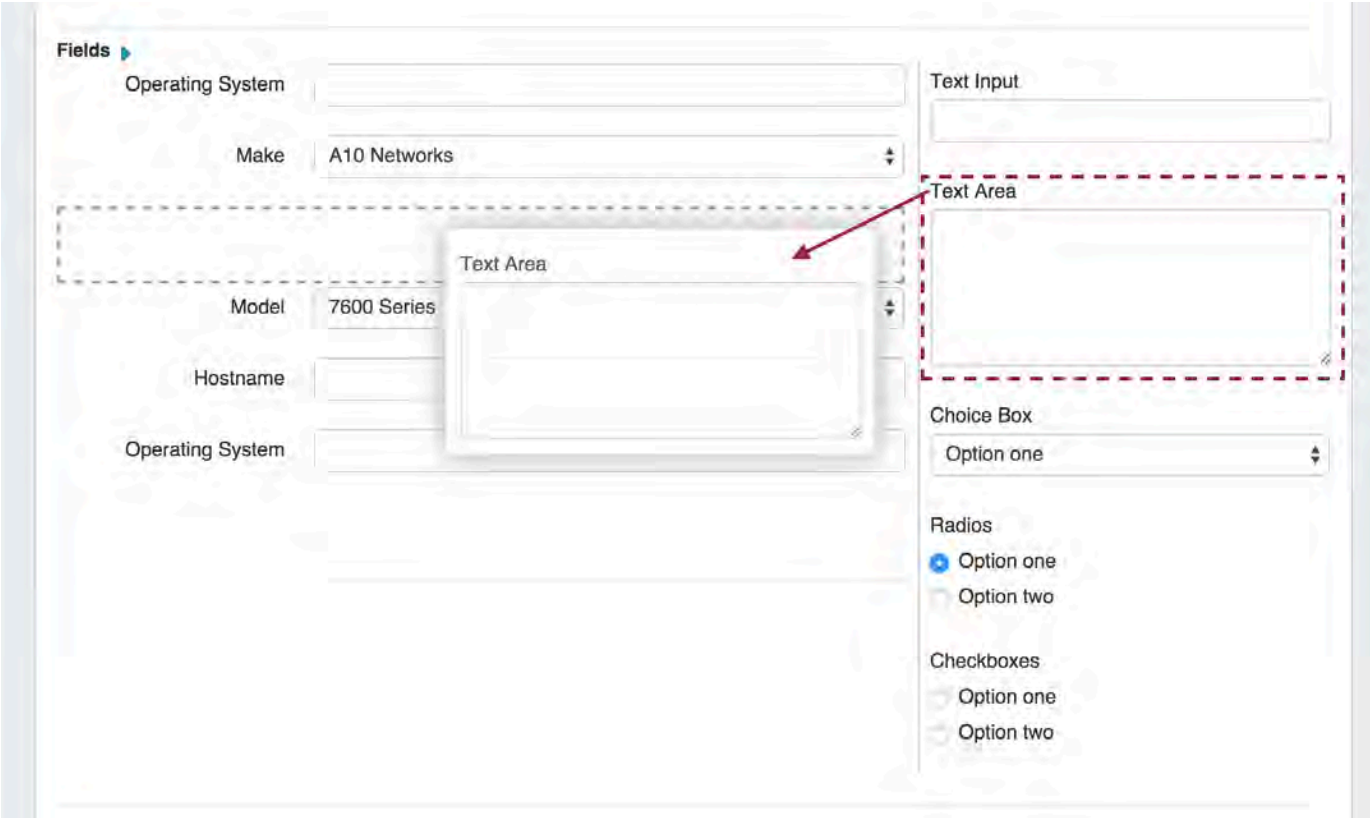
Create a new Section by specifying a Name, Parent, and Category. Then hit the "Create" button.



Step 2: Add a Custom Field to a Section

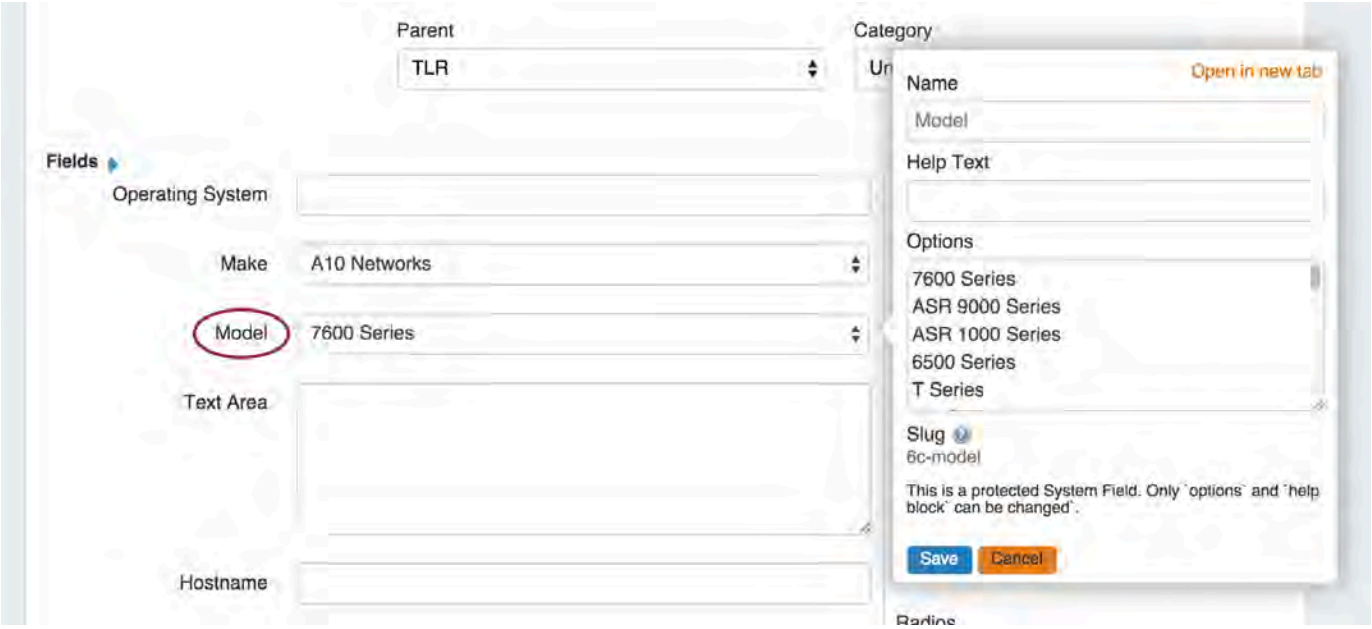
Manage existing fields and add custom fields for the selected Section by clicking "Edit Section"

Add existing or Custom Fields for your Section. You can add new Custom Fields of different types (text input, text area, choice box, radios, checkbox) by dragging and dropping the fields as well as using any existing fields that are available. See the Customizing Fields page for more details.



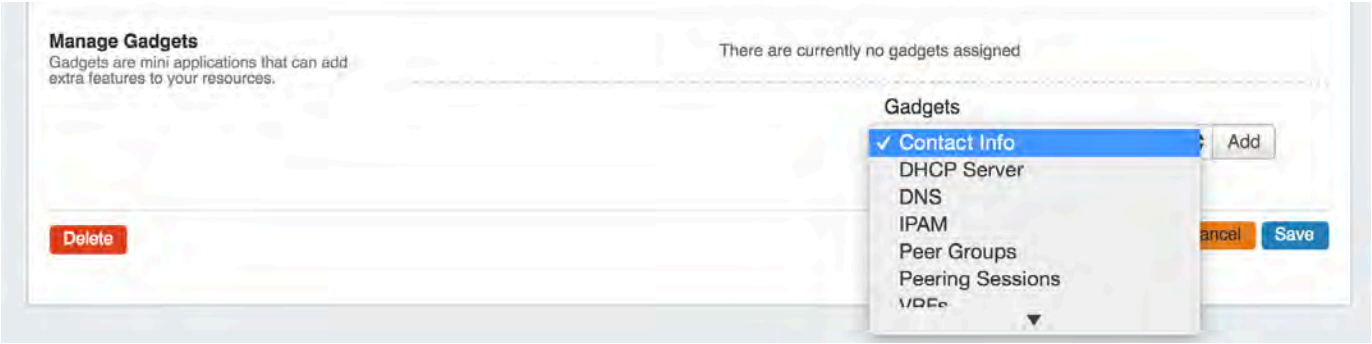
Step 3: Edit Custom Field Data

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 that you also have an area for Gadgets. Gadgets are modules of additional functionality that can be added to the UI of a given Resource. Simply select the Gadget you want to show for that section, hit "Add", then organize by dragging into the order you wish them to appear on the page. Once added to the Section, Gadgets will be visible for all Resources of that Section.



For a detailed list of gadgets and descriptions, see the Gadgets page.

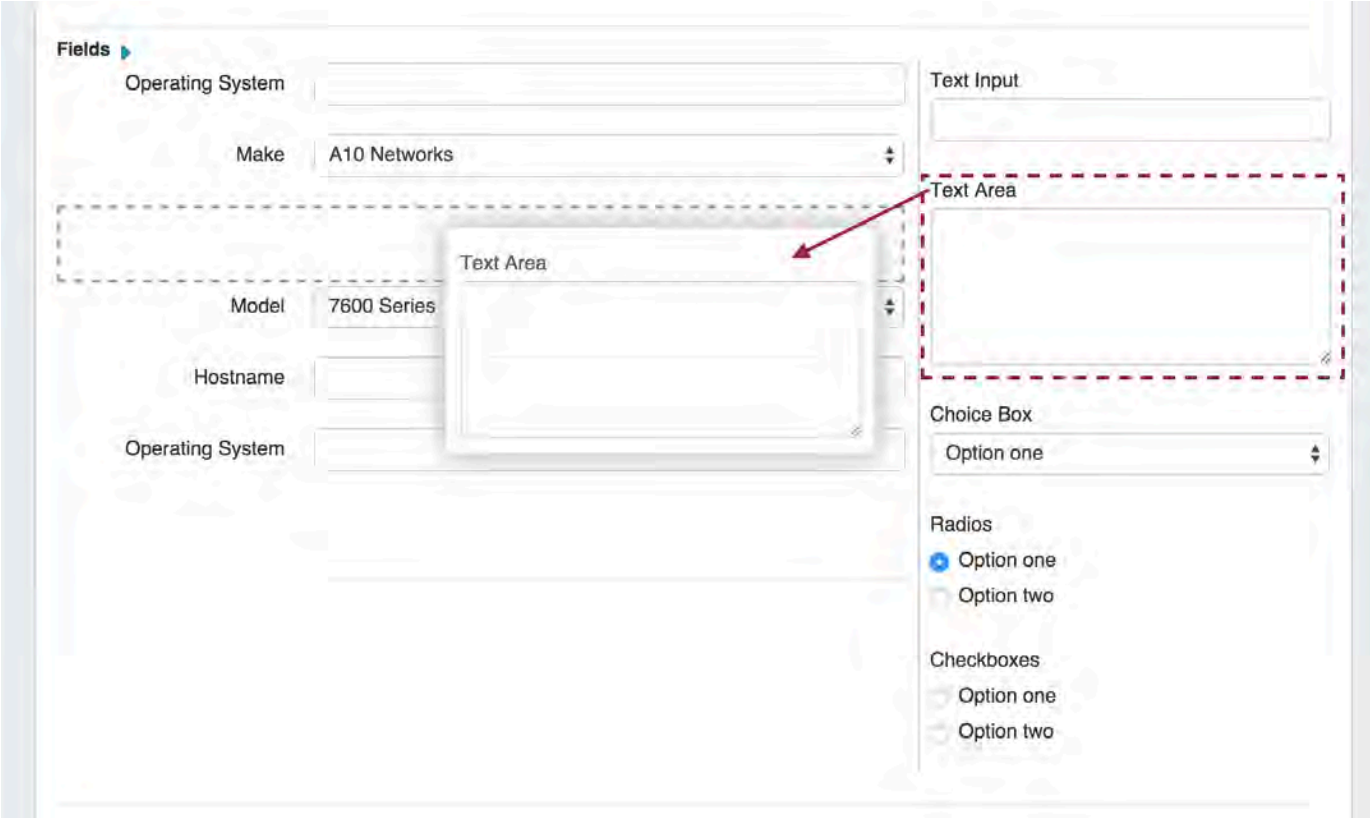
Customizing Fields

Working with Fields

Creating Fields

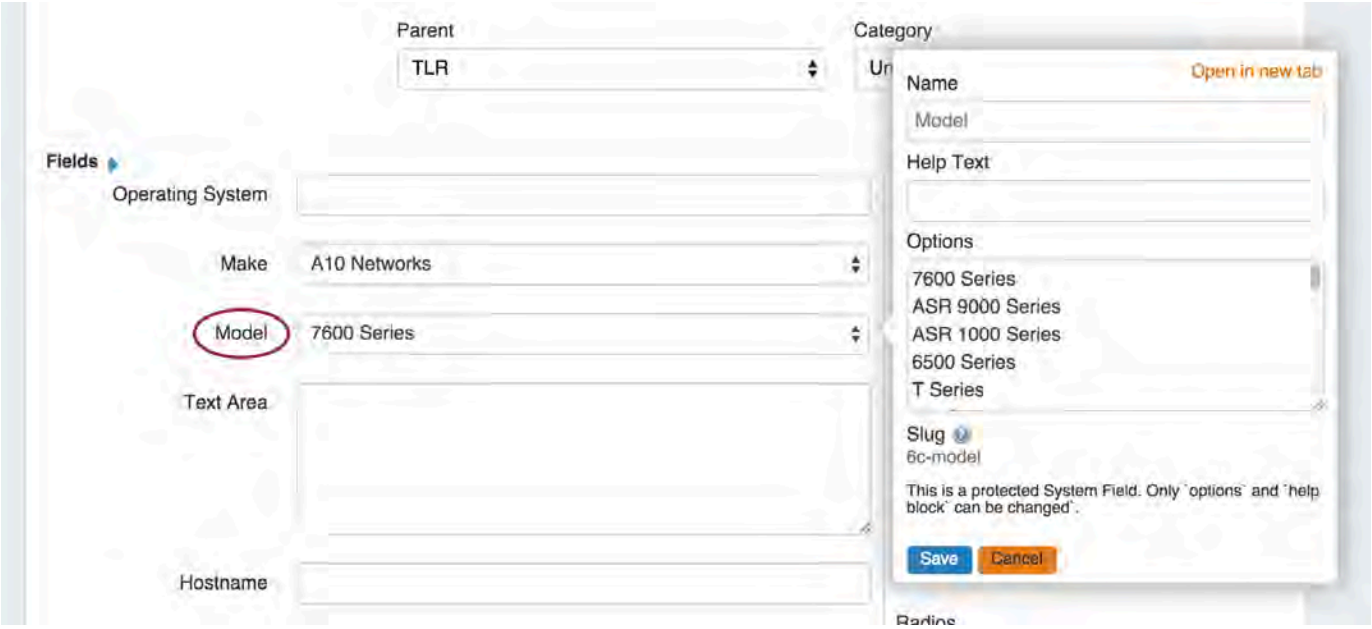
To add an existing field to a Section, select the field name from the dropdown menu and click on the "Add Field" button.

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.



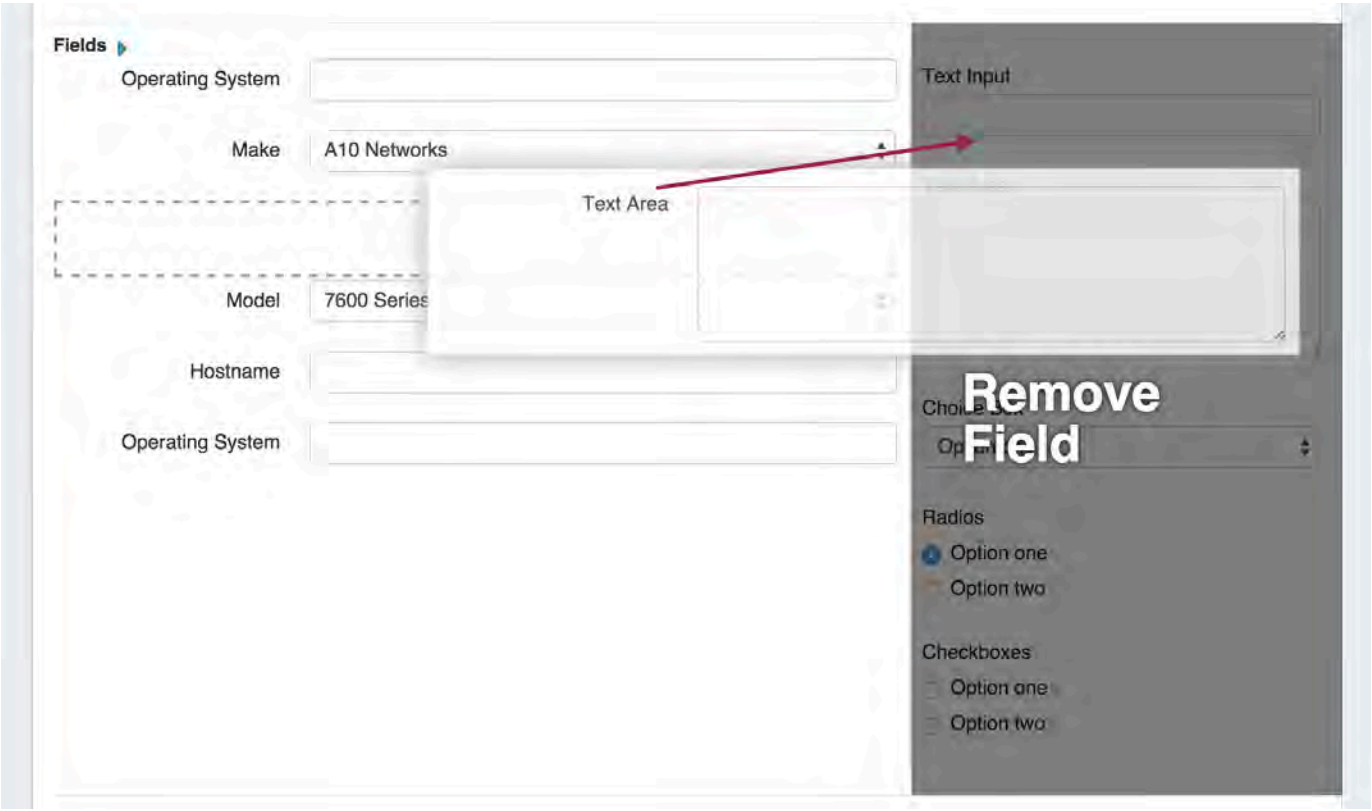
Editing/Removing Fields

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.



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

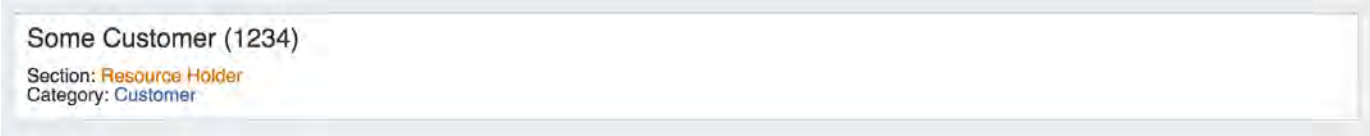
What are Gadgets?

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 (see Customizing Sections). Gadgets are best described as self contained webapps; w idgets 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.



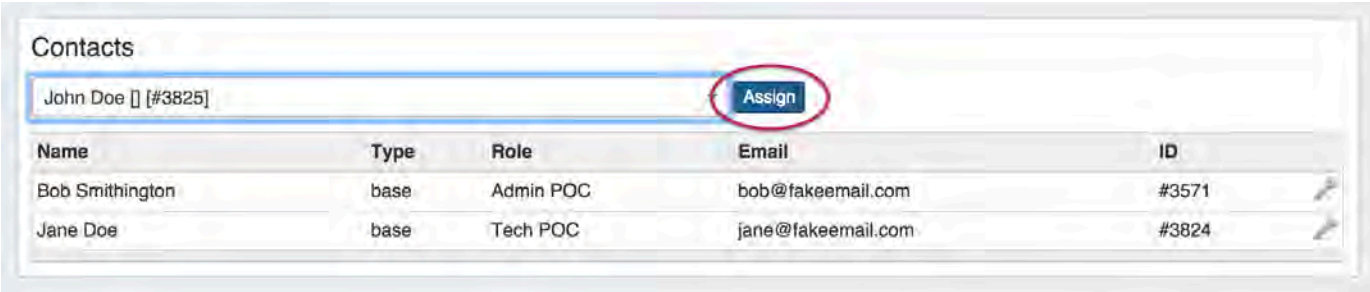
Contact Info

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.

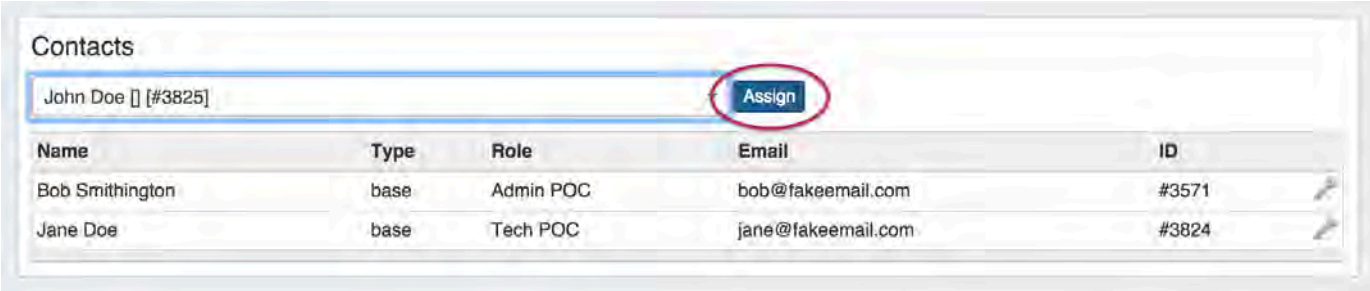
Contacts

The Contacts gadget may be used on any Section type to assign a contact (from the Contact Manager) to a Resource.

To assign a contact, search for and select the desired contact in the search box, then click "Assign". The Contact will show in the list below.



Contacts Action Menu:



The Action Menu (wrench icon) in the contacts gadget has three options:

Set Role: Opens a dialog box to select a role to assign to the contact, add a new role, or delete an existing role.

Unassign: Unassigns the contact from the resource

View: Redirects to the contact's detail page in the Contact Manager

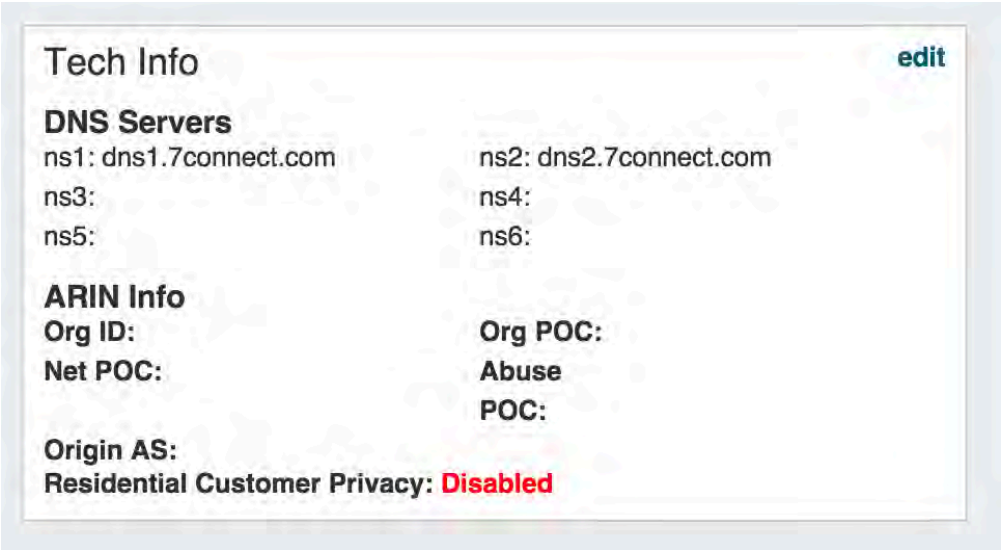
Refer to the Contact Manager page in the documentation for more detail on working with the Contacts gadget and the Contact Manager.

> **A Note on Permissions**
> The ability to view and assign a contact to a resource is restricted by the Parent assigned to the contact upon creation. If a user does

not have permissions for a contact's Parent Resource, that contact will not be able to be viewed / assigned via the gadget.

Tech Info

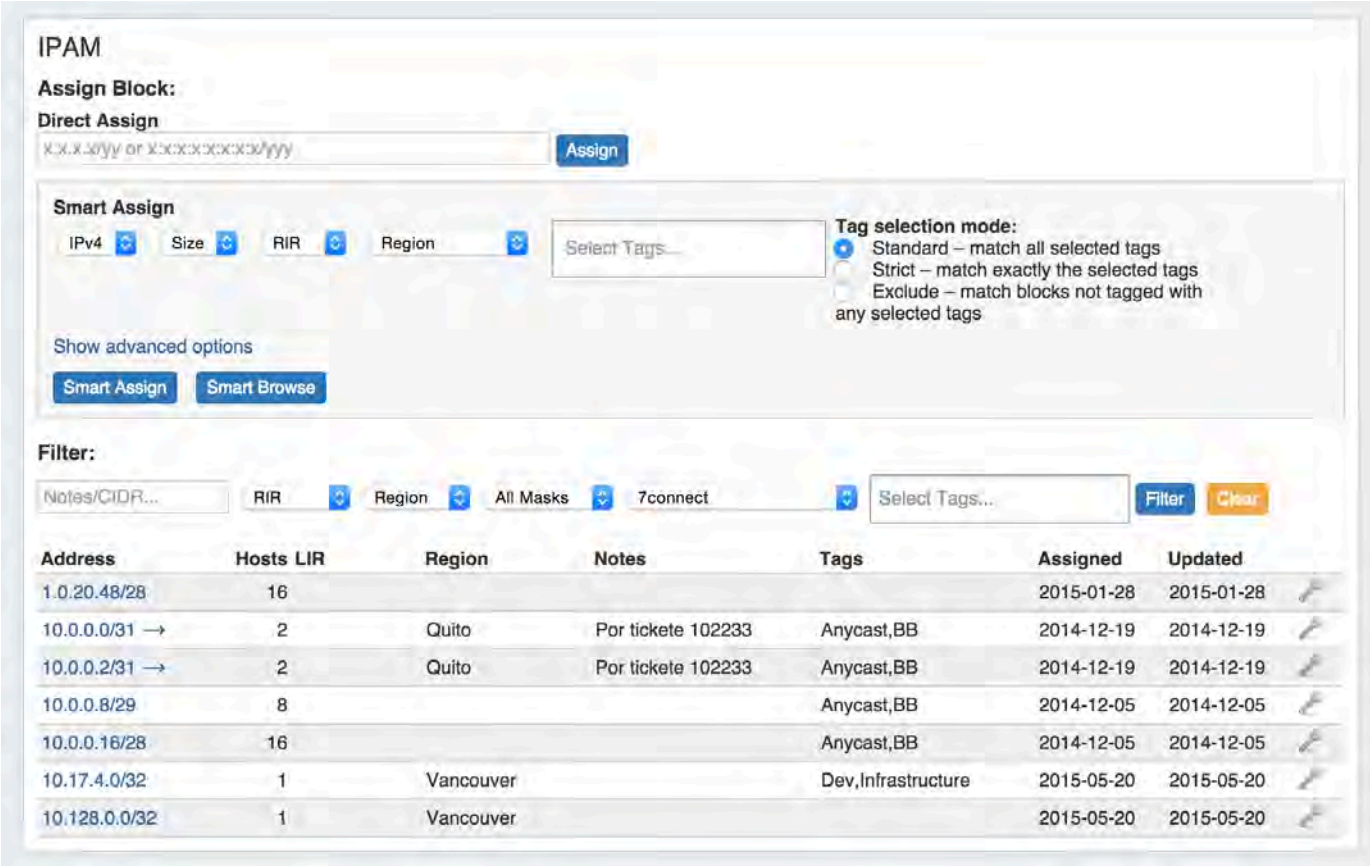
This visual element is used on the Resource Holder Section type. This Gadget allows you to list DNS servers, ARIN information, and enable/disable customer privacy.



IPAM

This gadget 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.



Document Storage

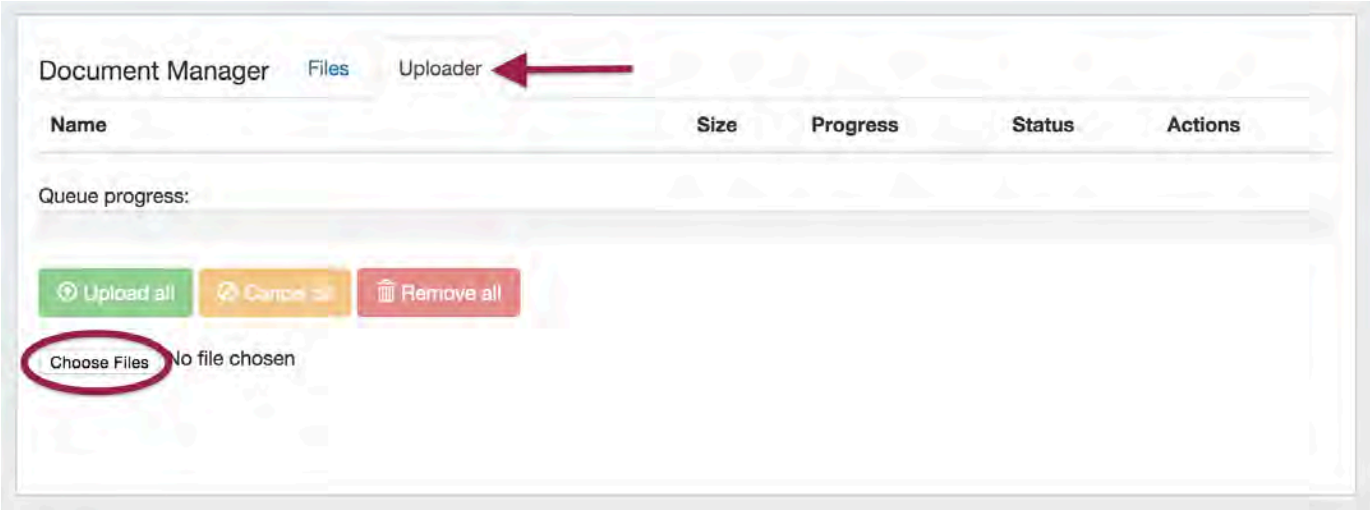
The Document Storage gadget allows you to upload documents to a resource, and have those documents accessible to download from the 6connect cloud to your local machine. It may be enabled on any Section type.

> Note: The uploader has a file size limit of 14mb, files above this size will not successfully upload.

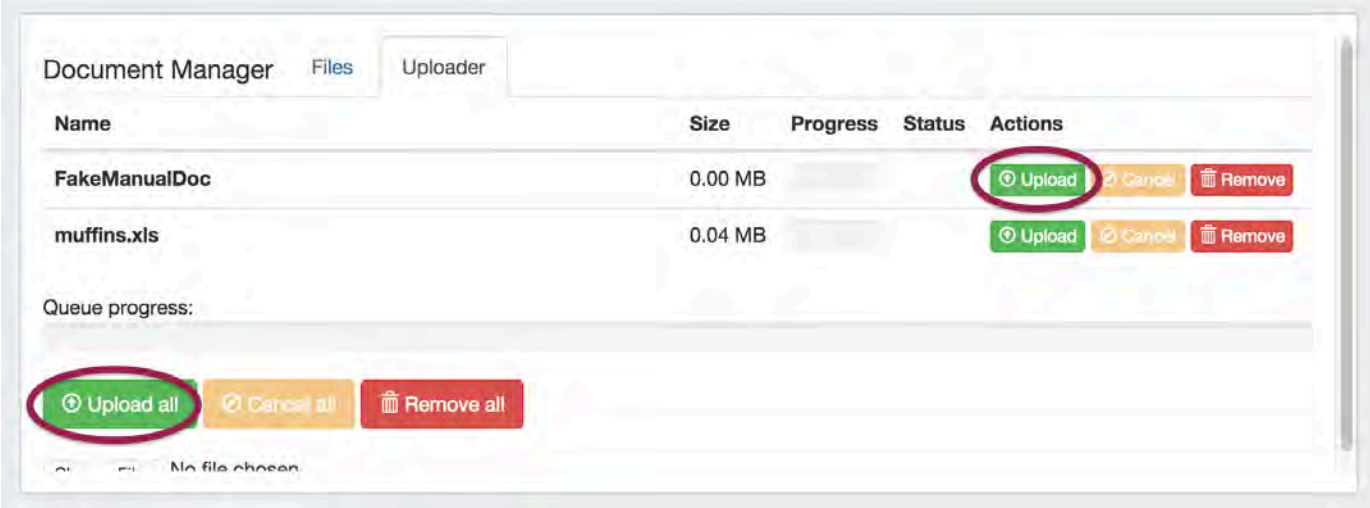
Upload Files

First, ensure the Document Storage gadget is enable for the Section type. Then, select the "Uploader" tab under the Document Manager gadget.

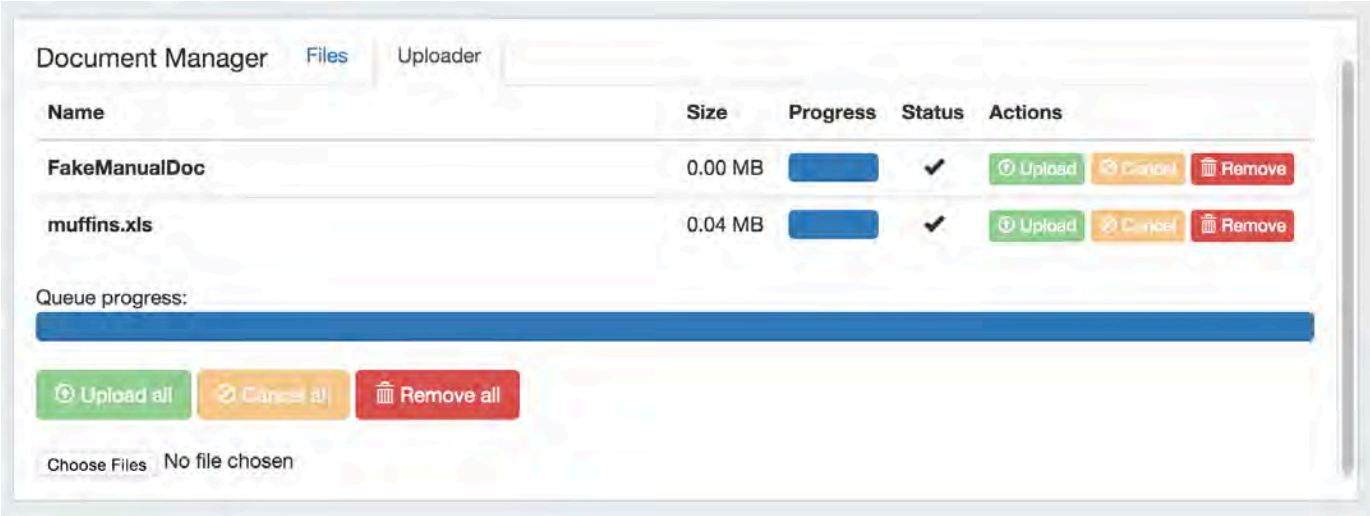
Click on the "Choose Files" button, and select the file(s) you would like to upload.



The selected files will show in a list under the Uploader tab. To upload the file(s), click on either the "Upload" button adjacent to the file to upload the individual file, or the "Upload All" button to upload all files listed. You may also choose to remove files from the upload list, or cancel.



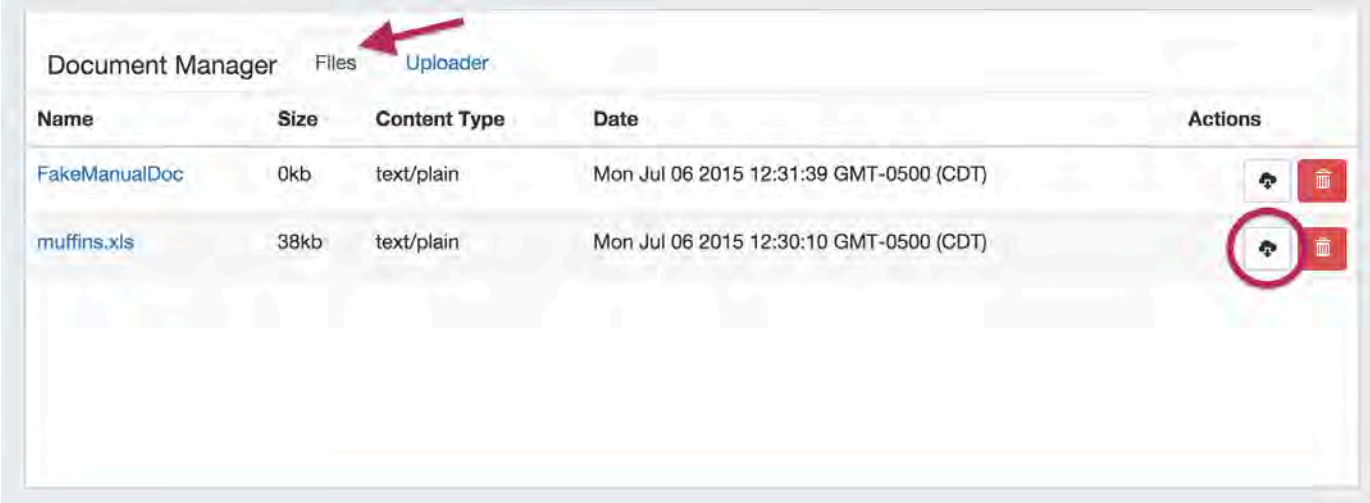
Once the progress bar is filled in, your upload is complete! Your files will show under the "Files" tab.



Download Available Files

Successfully uploaded files are listed under the "Files" tab of the Document Manager gadget. From here, you can see the file Name, Size, Type, Date, and available Actions.

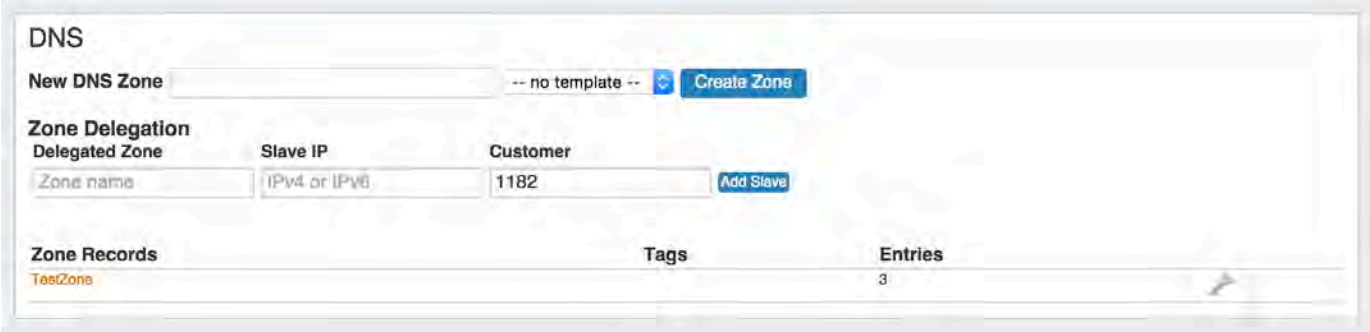
To download a file to your local machine, ensure you are on the "Files" tab, then click on the "Download from Cloud" icon.



If a file is no longer needed, you may delete the file from the Document Manager by clicking on the Delete (trash can) icon.

DNS

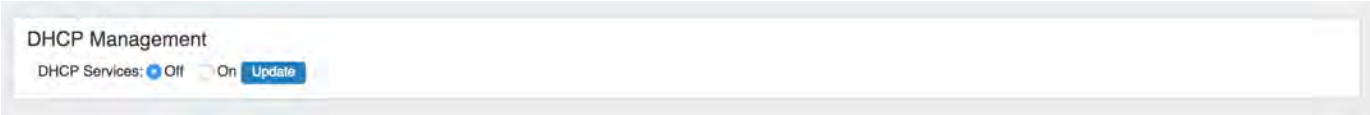
This gadget is used on the Resource Holder Section type. The DNS Gadget allows you to add new Zones as well as view and manage existing zones. For more information on DNS functions and managing zones, refer to the documentation for the DNS Tab.



DHCP Management

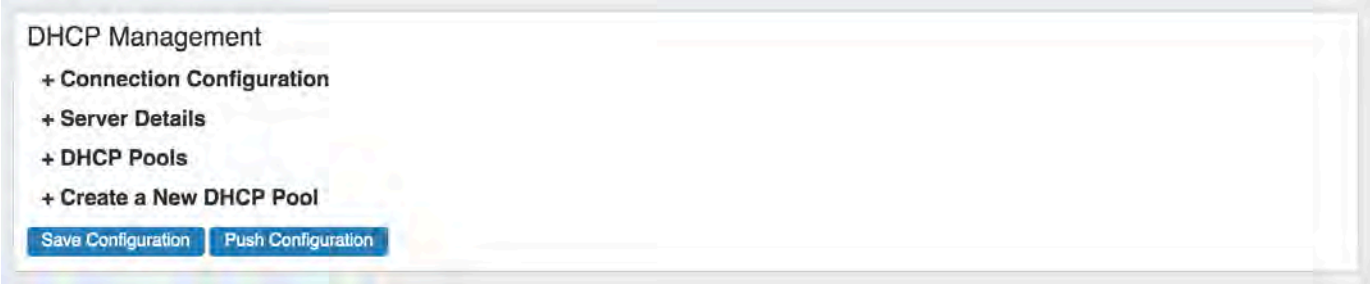
This visual element is used on the Server Section type.

The DHCP Management Gadget in the "Off" configuration:



To enable the DHCP Management Gadget, select the "On" radio button next to "DHCP Services", and click the "Update" Button.

The DHCP Management Gadget in the "On" configuration:

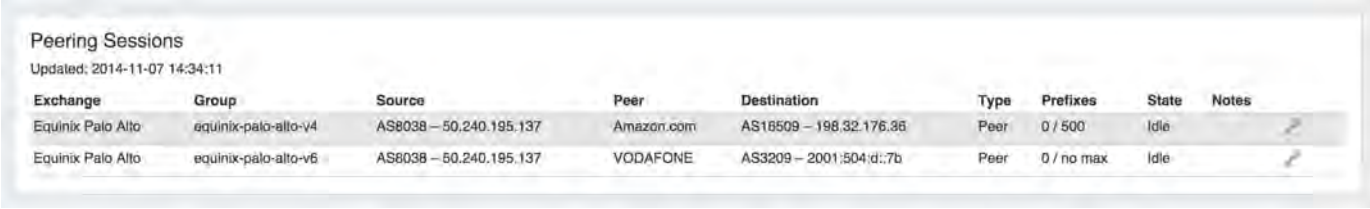


Each of the four sections in the DHCP Management Gadget - Connection Configuration, Server Details, DHCP Pools, and Create a New DHCP Pool - may be expanded to change settings in each area.

For detailed information on working with the DHCP Management Gadget, see DHCP Tab - Managing DHCP Server Configurations.

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.

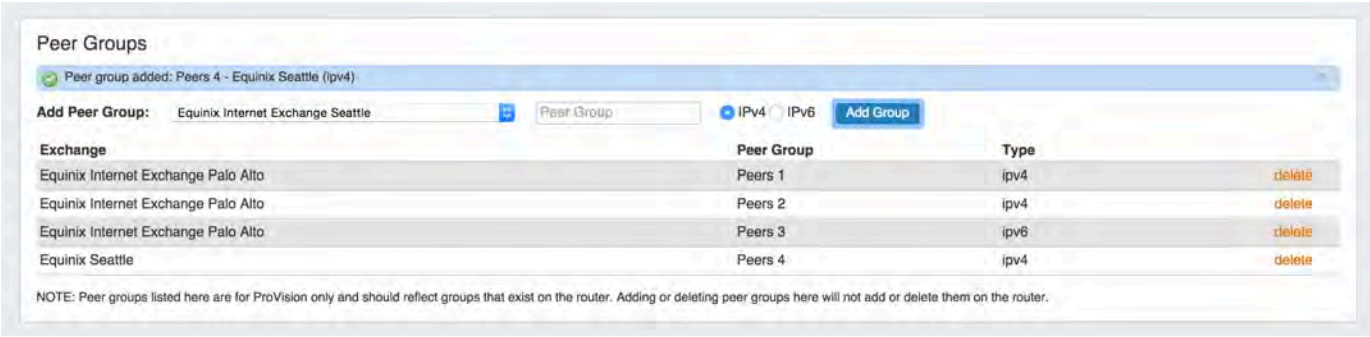


Peer Groups

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.

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.



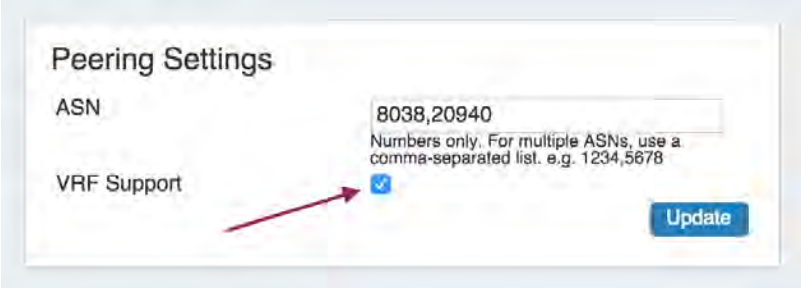
> **Note**
> Peer groups listed in the Gadget are for ProVision only and should reflect groups that exist on the router.
>
> Adding or deleting peer groups from the Gadget will not add or delete them on the router.

For additional information on Peering, see Peering v2.

Peering VRF

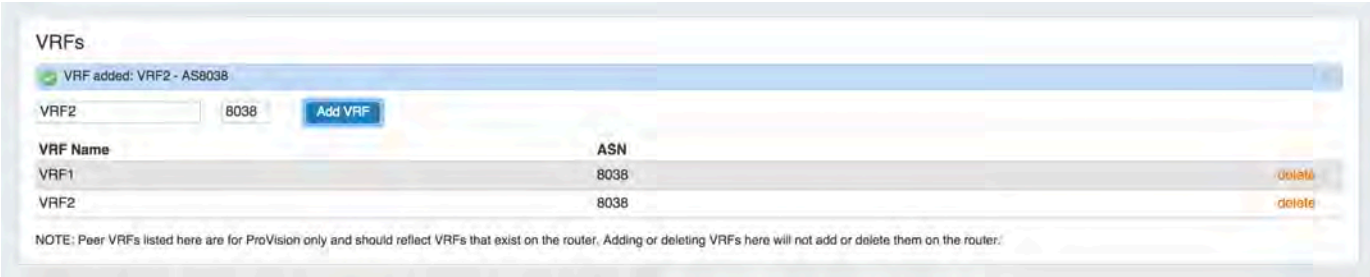
The Peering VRF Gadget allows you to add VRFs from a router's Resource Entry page.

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



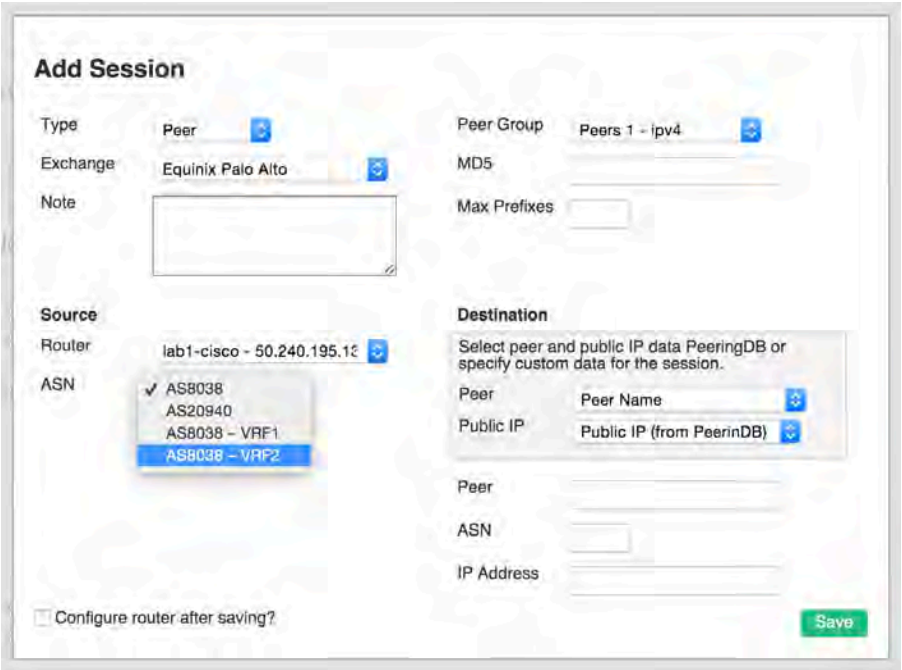
The VRF gadget will then be accessible in a router's Resource Entry page.

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



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.

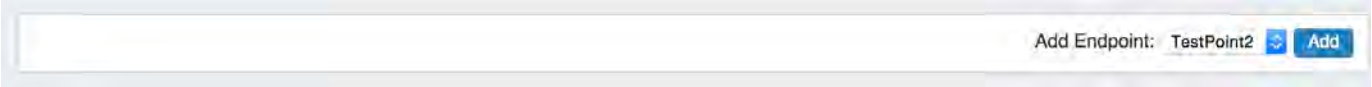


Peering VRF currently only supports Cisco routers.

Reverse API Console

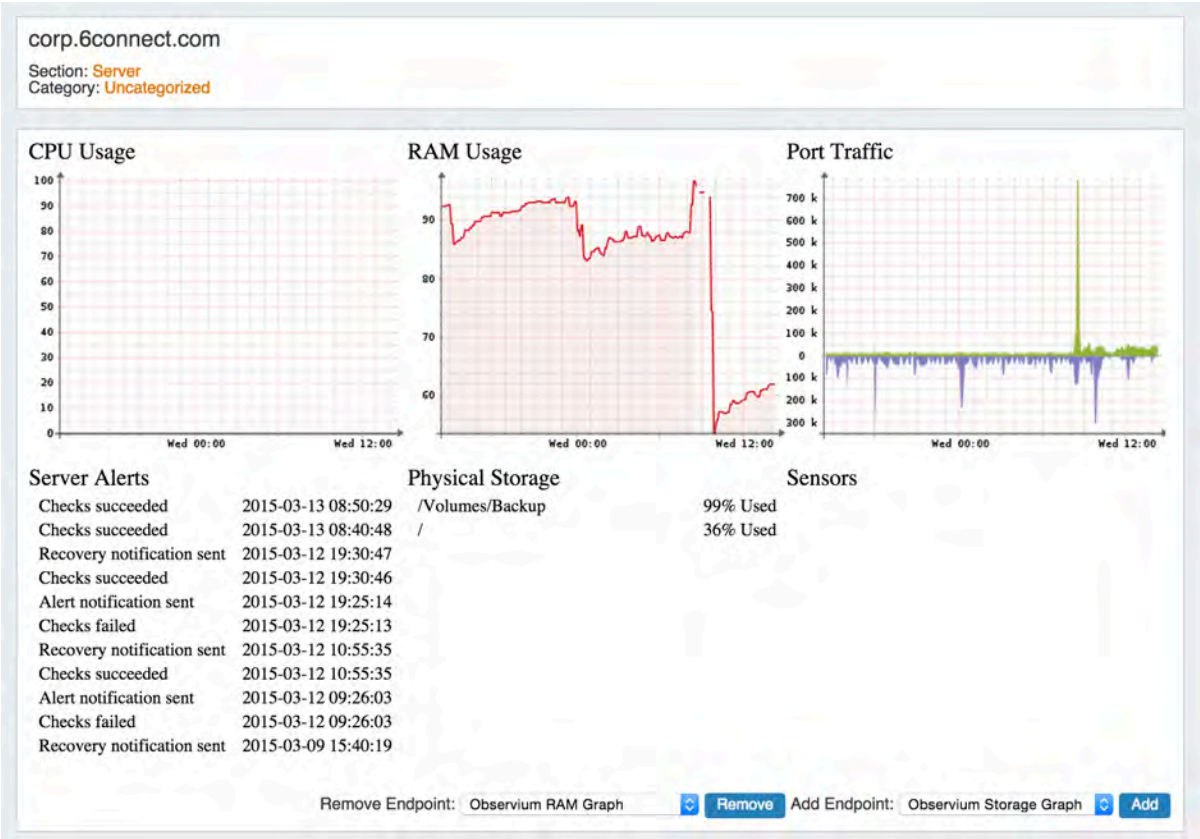
The Reverse API Console Gadget allows you to add endpoints from a Resource Entry page, and view customizable javascript displays set for the selected endpoint. The Reverse API Console may be added to any type of Section / Resource, and must be enabled for a Resource for certain rAPI processes to successfully complete.

After adding the gadget to a section, the Reverse API console will appear as whitespace with the option to add an endpoint display. Endpoints must already be created via rAPI or through the Admin Reverse API interface in ProVision to be selectable via the gadget.



Selecting an Endpoint and clicking "Add" will allow the javascript display code (added via rAPI or the Admin Reverse API interface) associated with that endpoint to display on the Resource Entry page.

An example display might be integrating Observium to display server status modules and alerts for Resources with the "Server" Section:



As a completely customizable area, displays can be designed to meet individual Resource needs - create charts, alerts, command buttons, or any other type of data that you wish to view.

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 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

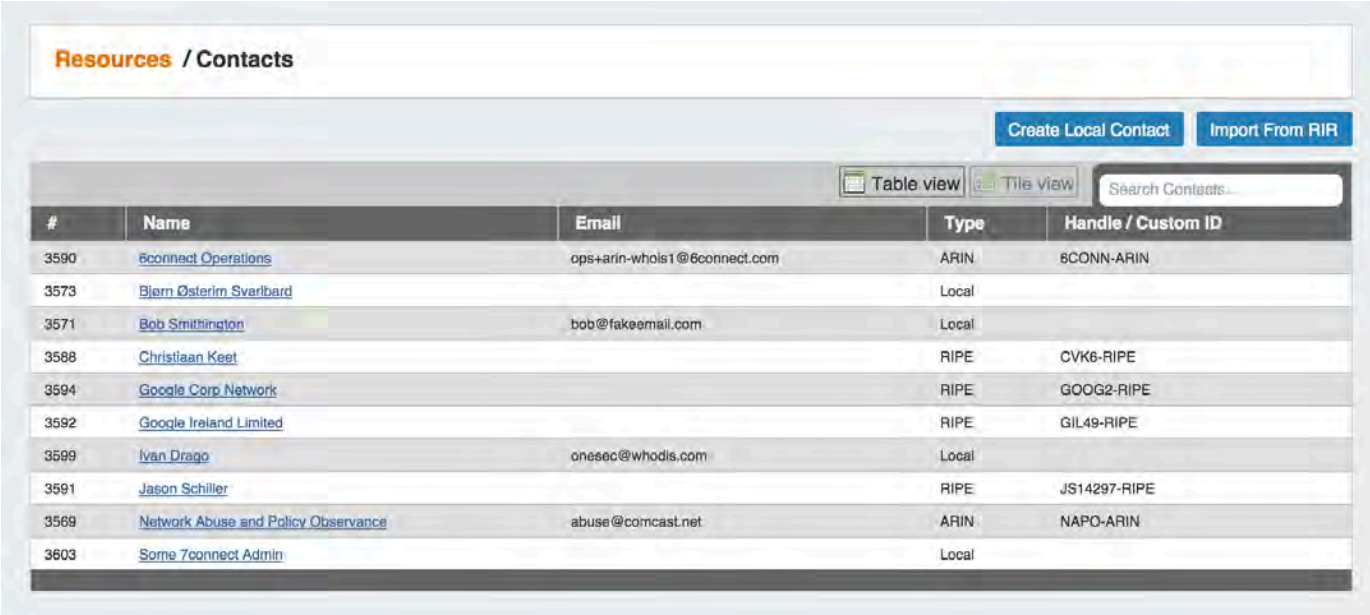
If a gadget uses fields, you can optionally add the slug of the field in this section to hide it from the main field list.

This can be very useful and make your Resource Sections 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.

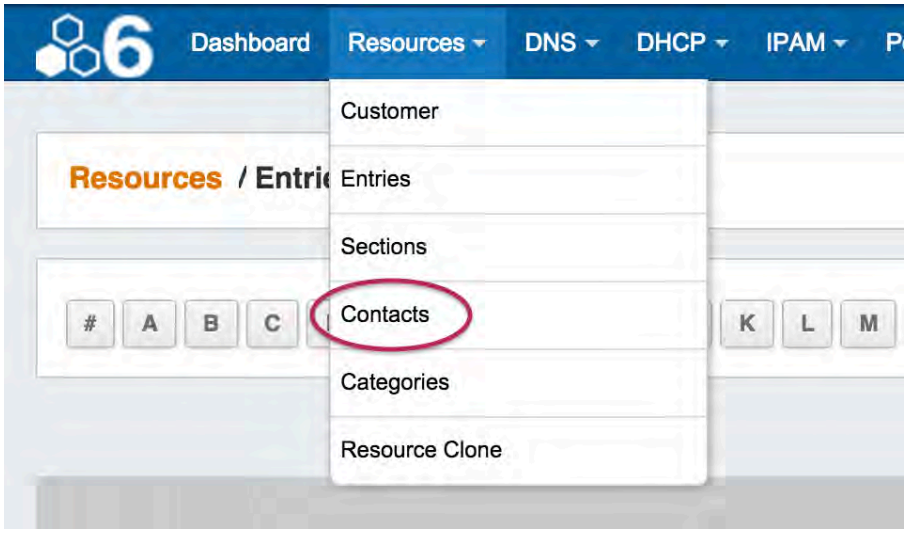
Contact Manager

Contact Manager

The Contact Manager lists and imports contacts into ProVision. You can create new contacts manually, or import contacts from RIR. Contacts may then be assigned to ProVision resources through the Contacts Gadget.



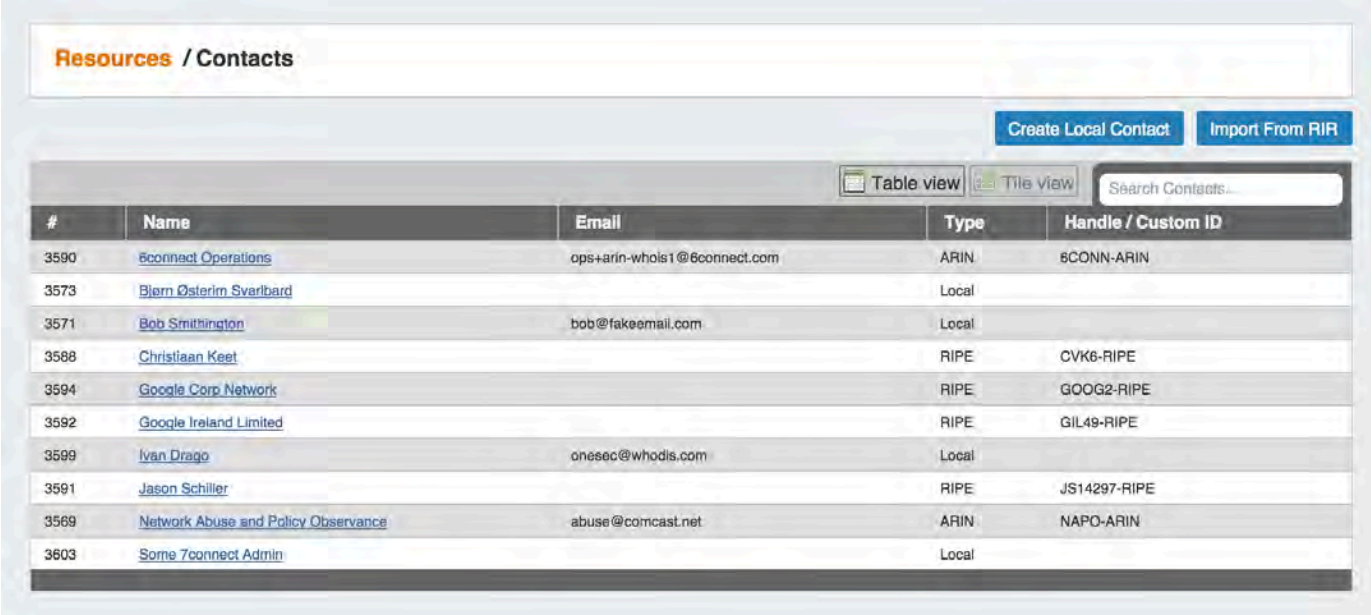
The Contact Manager is accessed from the Resource tab dropdown menu, under "Contacts"



UI Overview

Table View

Table View shows contacts in list form, their ID number, email, contact type, and their handle /custom ID. If the contact was manually created through the "Create New Contact" button, it will show as type "Local", whereas a contact imported from RIR will show of type "RIPE" or "ARIN".



#	Name	Email	Type	Handle / Custom ID
3590	6connect Operations	ops+arin-whois1@6connect.com	ARIN	6CONN-ARIN
3573	Bjørn Østerim Svarlbard		Local	
3571	Bob Smithington	bob@fakeemail.com	Local	
3588	Christiaan Keet		RIPE	CVK6-RIPE
3594	Google Corp Network		RIPE	GOOG2-RIPE
3592	Google Ireland Limited		RIPE	GIL49-RIPE
3599	Ivan Drago	onesec@whodis.com	Local	
3591	Jason Schiller		RIPE	JS14297-RIPE
3569	Network Abuse and Policy Observance	abuse@comcast.net	ARIN	NAPO-ARIN
3603	Some 7connect Admin		Local	

Tile View

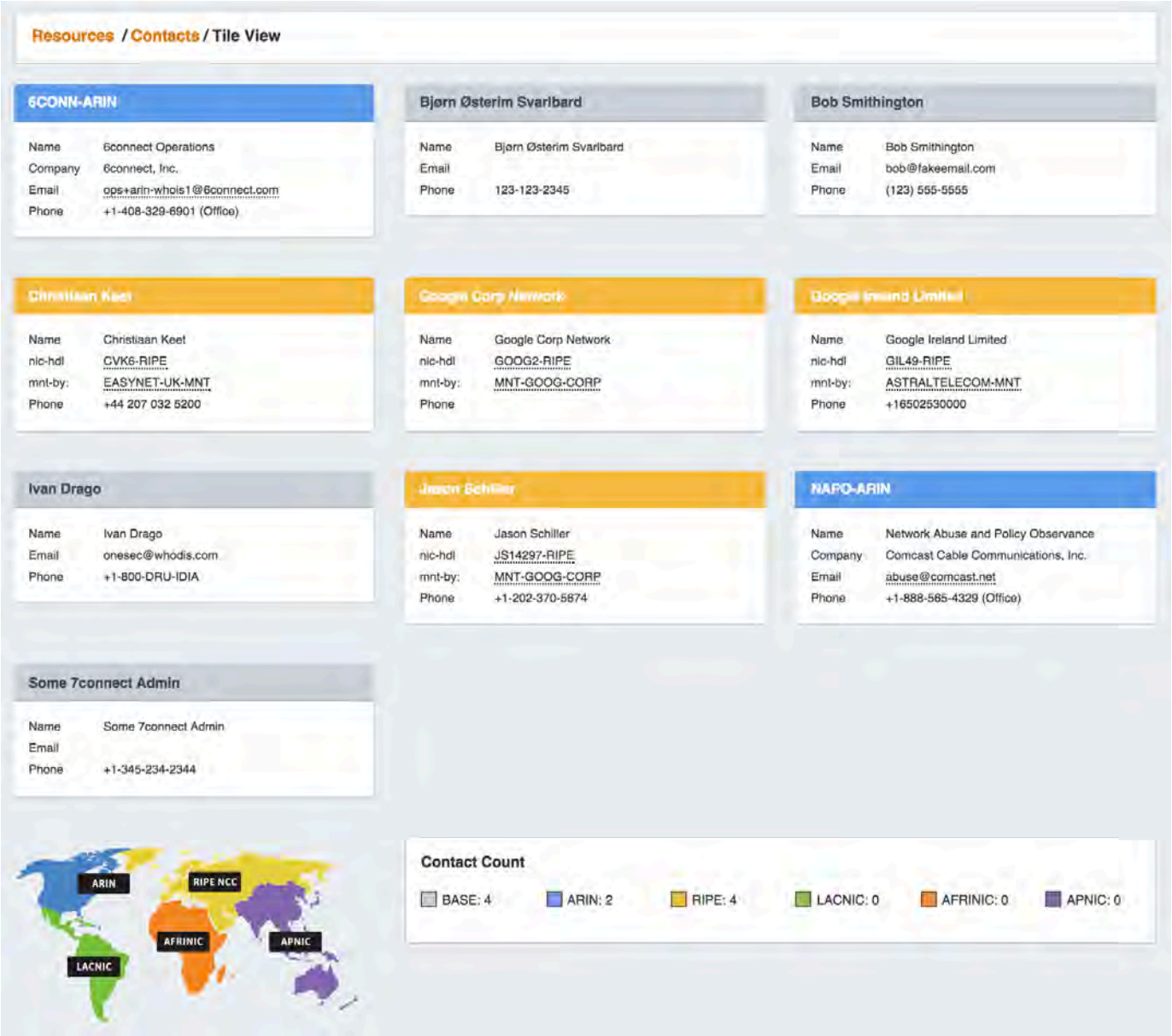
Tile view allows you to see all current contacts as a snapshot of basic information, based on contact type:

Base Contacts: Shows Contact Name, Email, and Phone

ARIN Contacts: Shows Contact Name, Company, Email, and Phone (if provided in ARIN)

RIPE Contacts: Shows Contact Name, nic-handle, maintained by, and phone (if provided in RIPE).

Clicking on the contact name on the header of the tile takes you to the contact's detail information page.



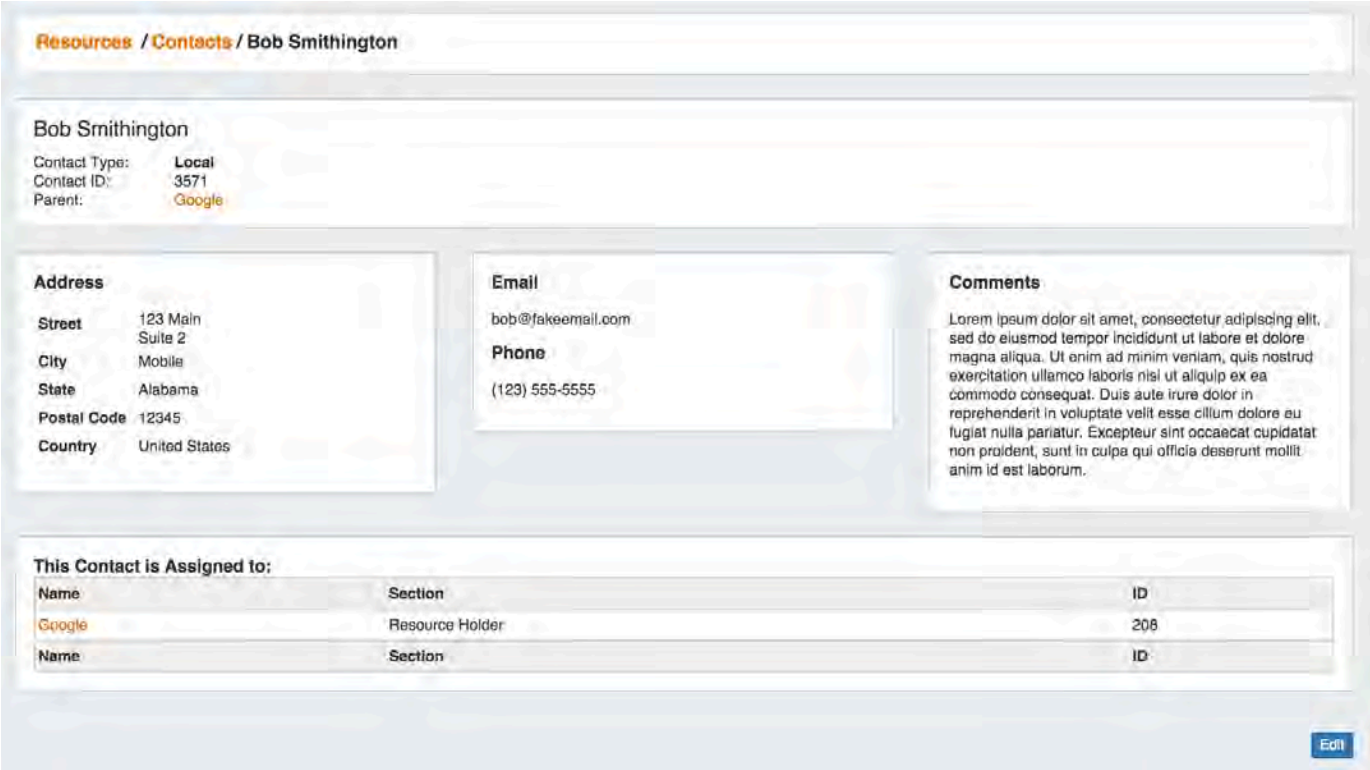
At the bottom of the Tile View page, counts of current contacts by RIR as well as an image map are provided. At this time, only Base (Local), ARIN, and RIPE contacts are supported.

View Contact Details

To view the contact details, click on the contact's name in either list or tile view. You will be taken to the contact details page.

A Local contact created in ProVision will have modules shown for Address, Email/Phone, and Comments, as well as a list of any resources that contact is assigned to via the Contacts gadget.

You may edit or delete a Local contact by clicking the "Edit" button at the bottom of the page.



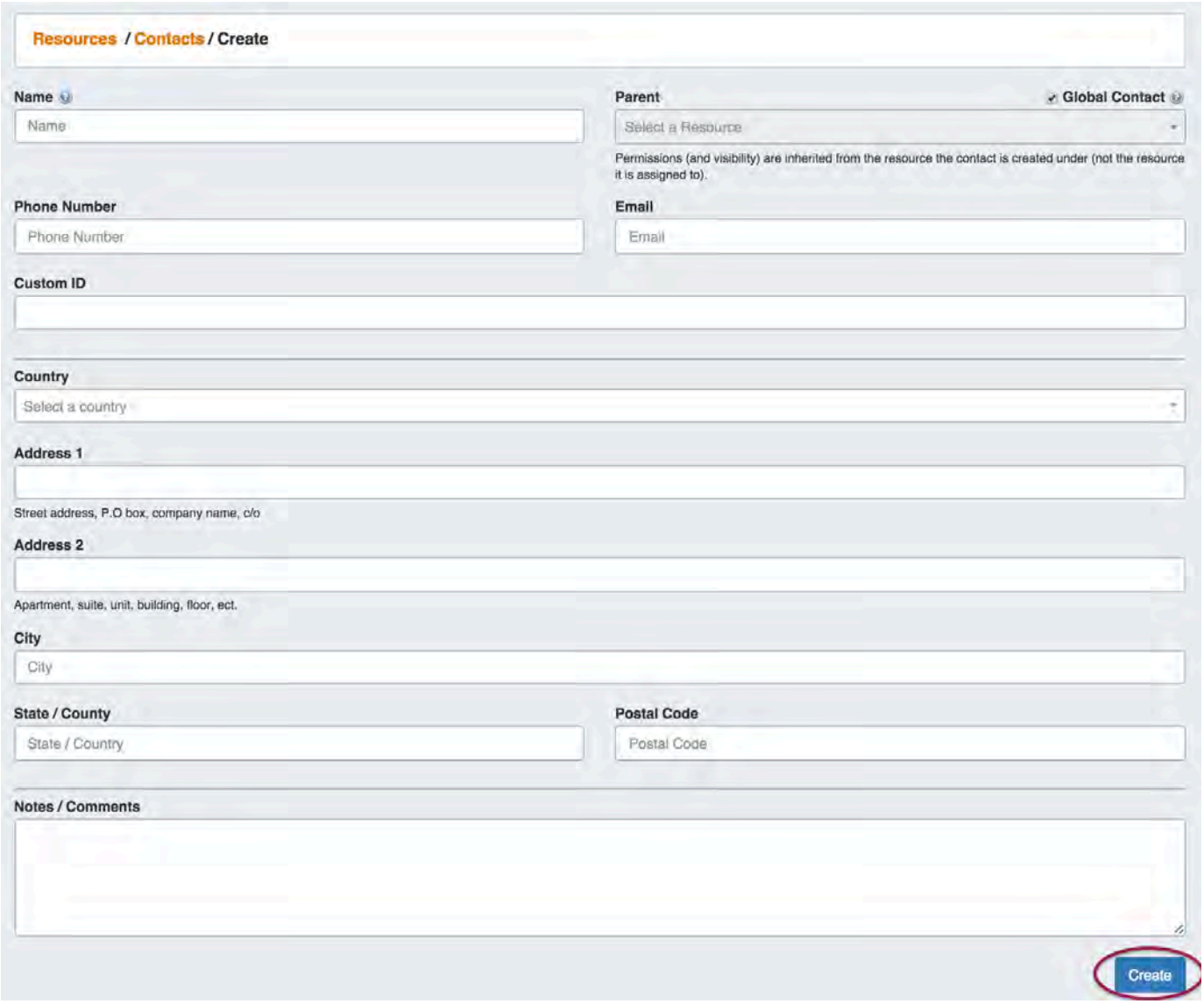
A contact imported from a RIR will have its details shown as determined by the imported RIR module - this data is not editable in ProVision.

You may delete the contact by clicking the "Delete" button at the bottom of the page.

Create New Contact

You can manually create a new contact in ProVision by clicking on "Create Local Contact" under the contacts page.

Fill in the contact's information for Name, Parent Resource (if desired), Phone, Custom ID, Email, Address, and Comments if desired.



When done filling out the information fields, hit the "Create" button to save your contact.

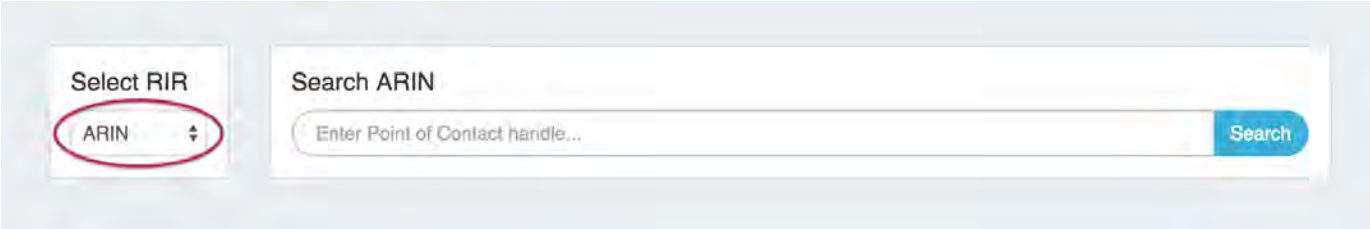
> **Parent**
> The parent field relates the contact to a resource's permissions structure, only allowing users with permissions for that resource to view and assign the affiliated contact(s). To allow the contact to be viewed globally in ProVision, check the "Global Contact" box above the Parent Field.

Import Contact from RIR

To import a contact from ARIN or RIPE, click the "Import from RIR" button from the contacts page.

1) Select RIR / Type

Once on the the Import from RIR page, on the left hand side of the page select the RIR from which you wish to import - ARIN or RIPE.

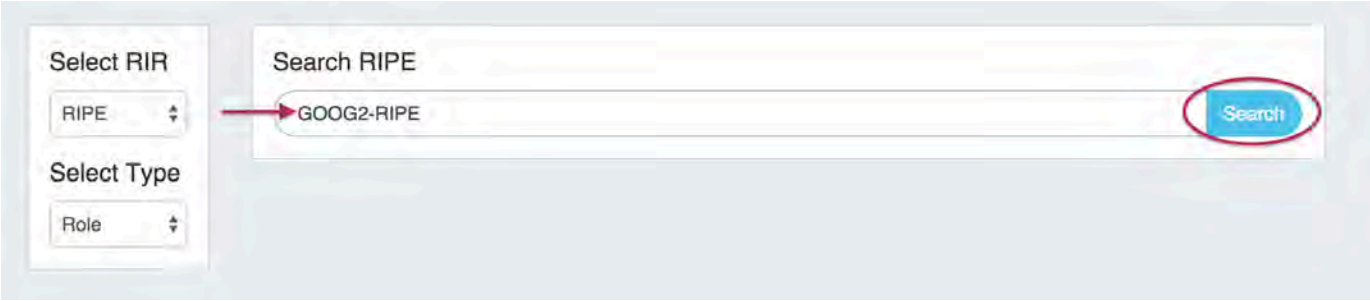


If you select RIPE, you will also be prompted to choose whether you are importing a "Person" or "Role".

2) Enter Point of Contact Handle

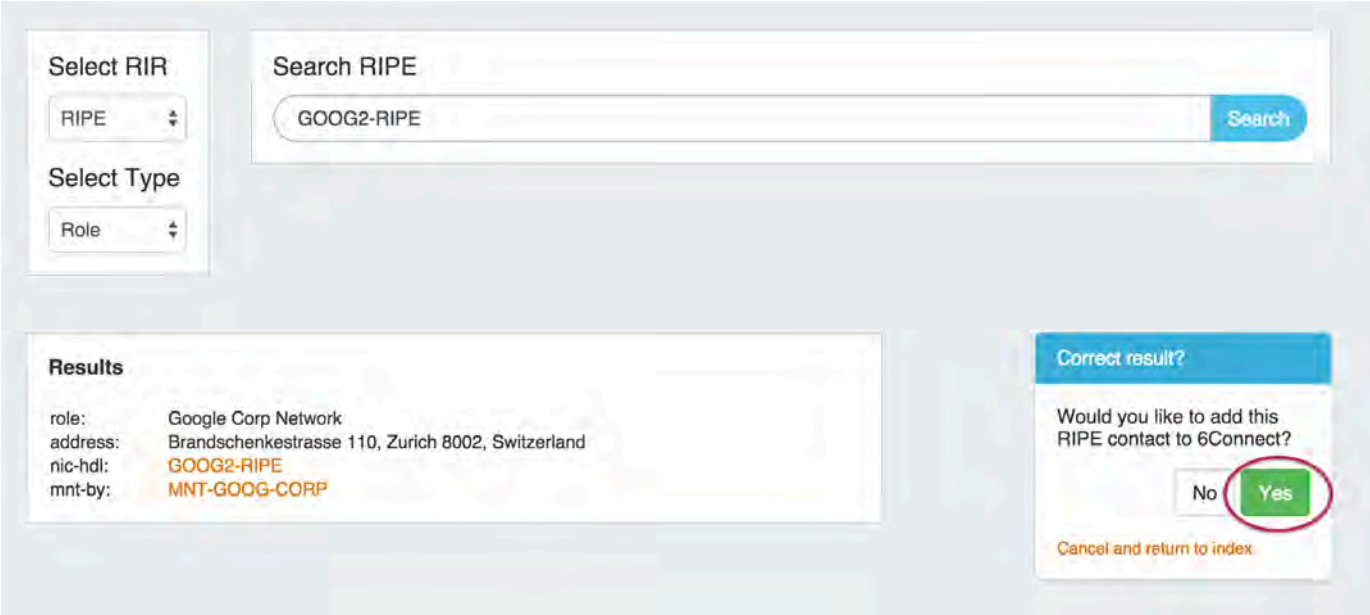
After selecting the RIR / Type (if applicable), enter the ARIN or RIPE Point of Contact Handle for the contact you are importing, then click "Search".

> Point of Contact Handle is the only search term that is valid. Attempting to search by name or keyword is not valid, and will return an error.



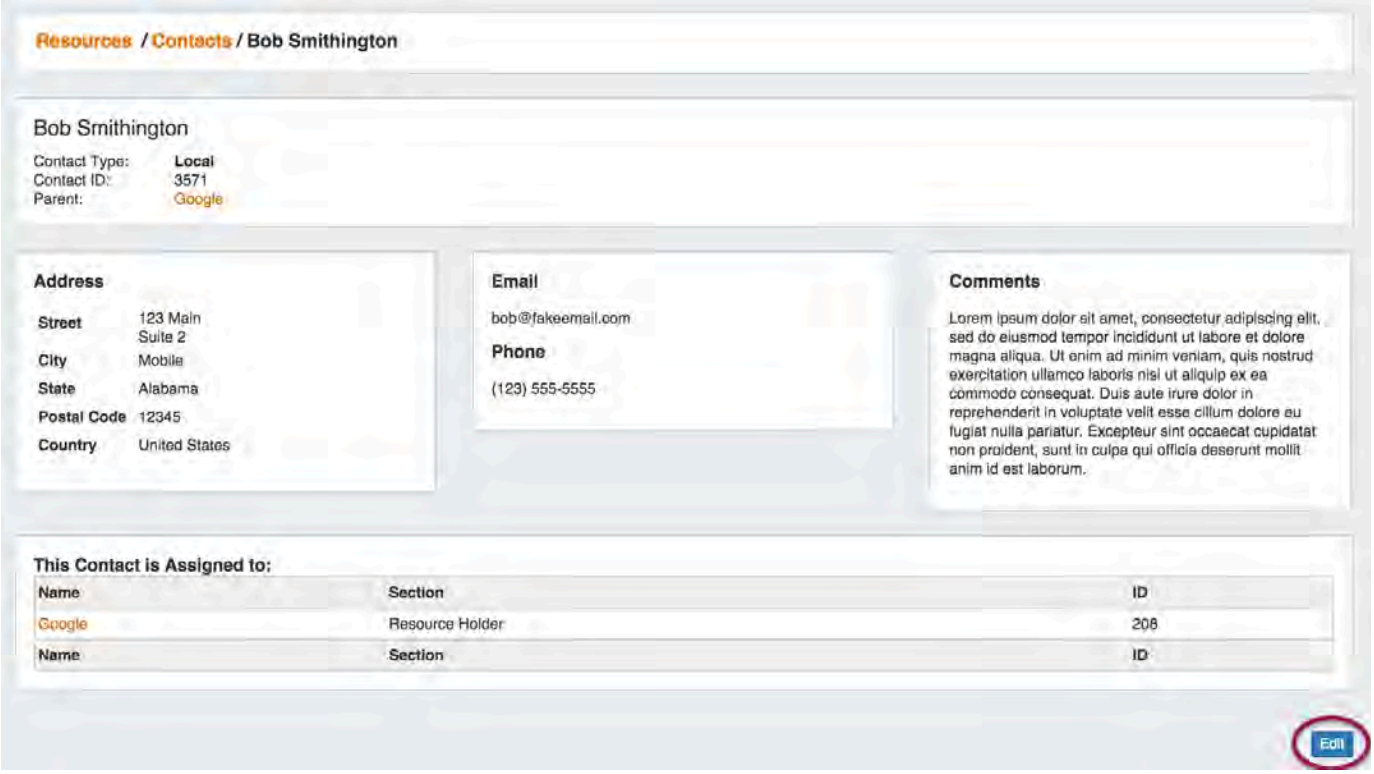
3) Verify and Add

If the search is successful, the found contact information will display at the bottom of the screen. If this is the information you wish to add into ProVision, click the "Yes" button on the right hand side of the page under "Would you like to add this RIPE contact to 6connect?". Otherwise, you may hit "No" to decline adding, or cancel and return to the index.

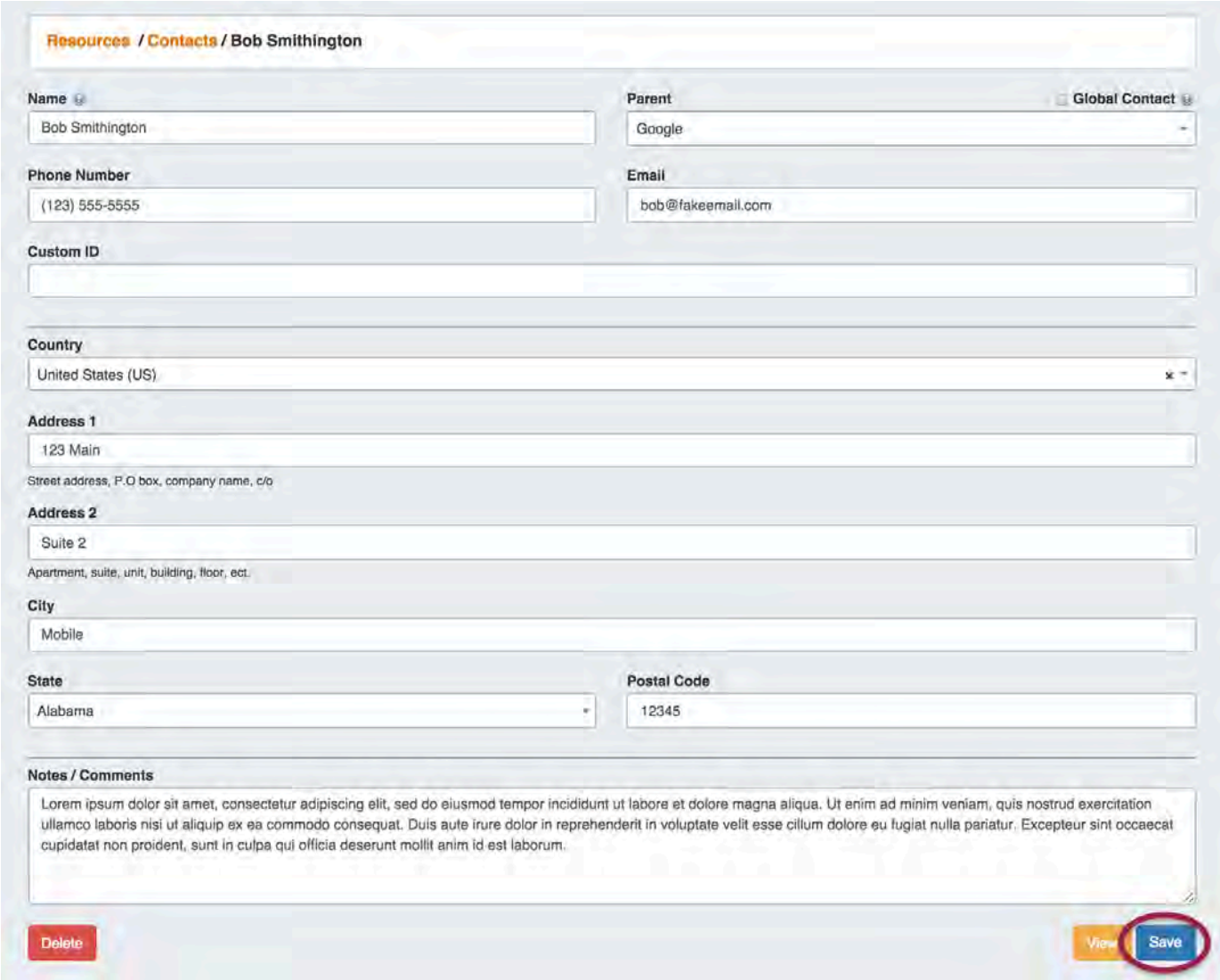


Edit or Delete Contact

To Edit or Delete a base contact, click on the contact's name in the contact list to go to its detail view page. From there, click "Edit".



This will take you to the detail information page, where you may edit fields as desired. When done editing, hit "Save" to save your changes, or "View" to cancel without saving and view the contacts list.



To delete the contact, click the "Delete" button while in the contact details page.

Working with the Contacts Gadget

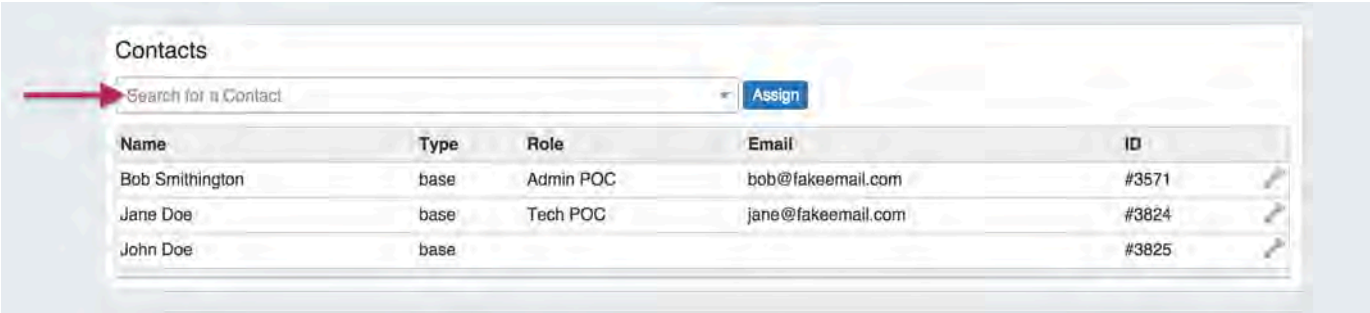
Assigning a Contact to a Resource

Contacts may be assigned to a resource through the Contacts gadget. To add a gadget to a Resource's Entry page, see Customizing Sections and Gadgets.

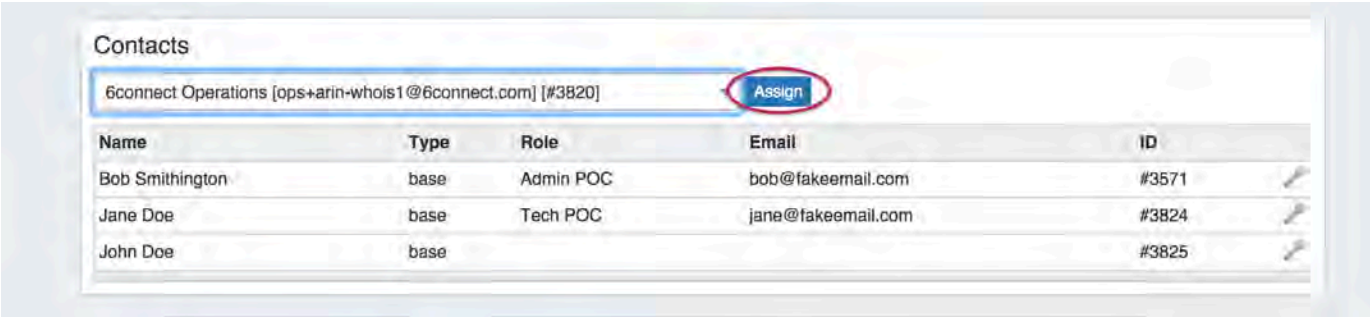
Once the Contacts gadget has been added to the desired Section, go to the Resource Entry page for the Resource you wish to assign the contact to, and navigate to the Contacts gadget.

Assign the Contact

In the Contact Gadget, search for a contact by typing the first few letters of the contact name in the search box, then click on the desired contact.



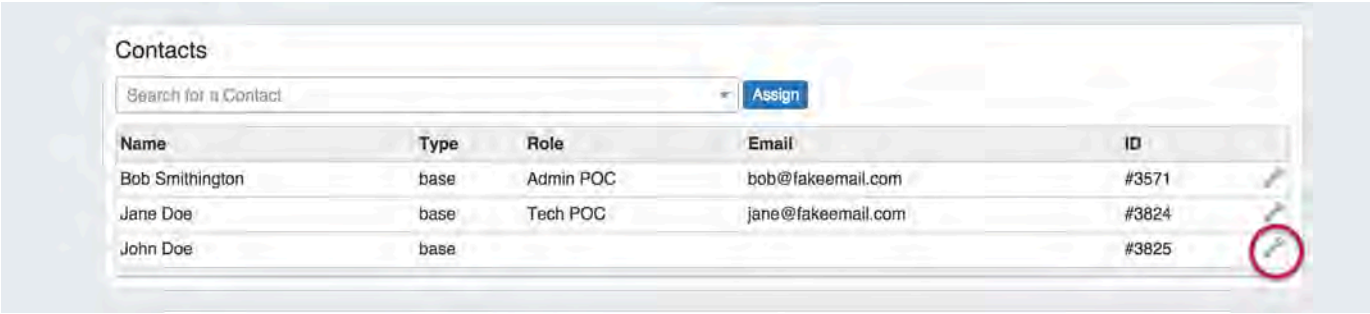
Then hit "Assign".



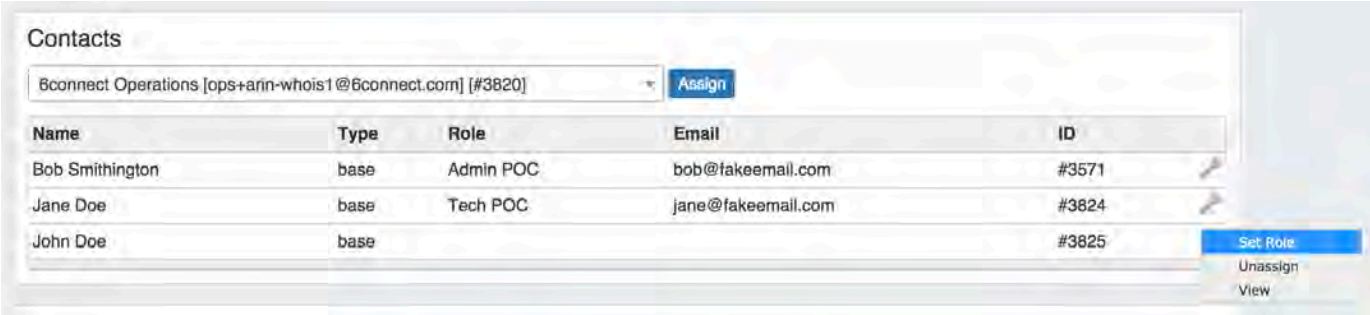
The Contact will show in the list below.

Create a Contact Role

Once a contact has been assigned, you can create and set a role for the contact by clicking on the Action Menu (wrench icon).



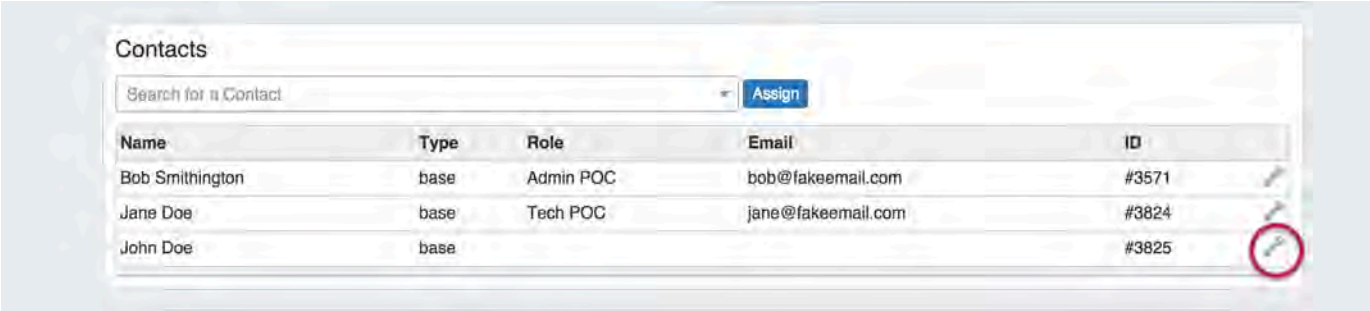
Select "Set Role"



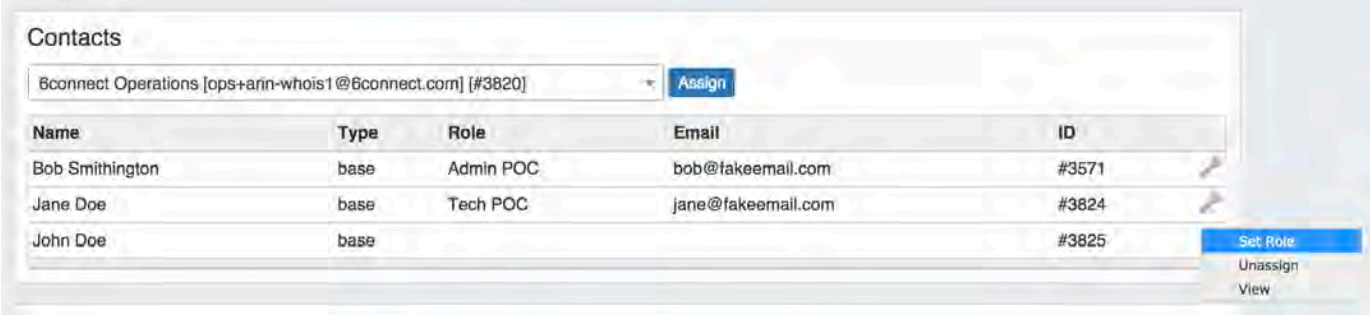
Under the "Add New Role" section, type in the desired tole name and click the "Add" button.

Assign a Role to a Contact

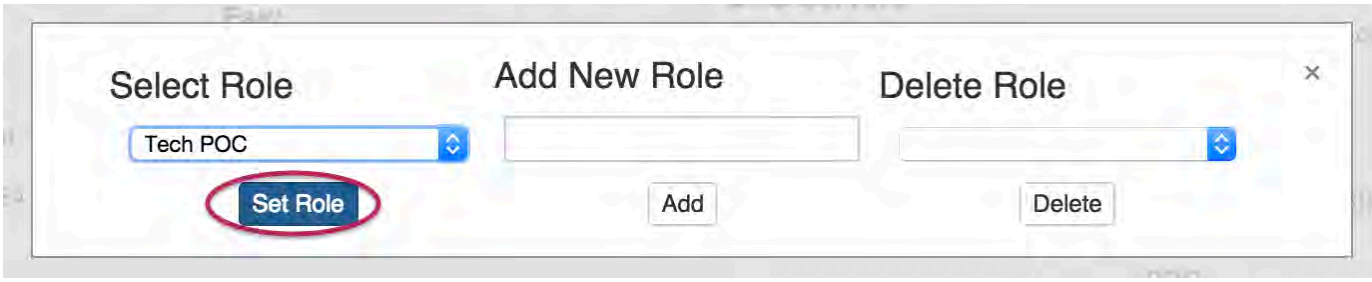
Once a contact has been assigned, you can set a role for the contact by clicking on the Action Menu (wrench icon).



Select "Set Role"

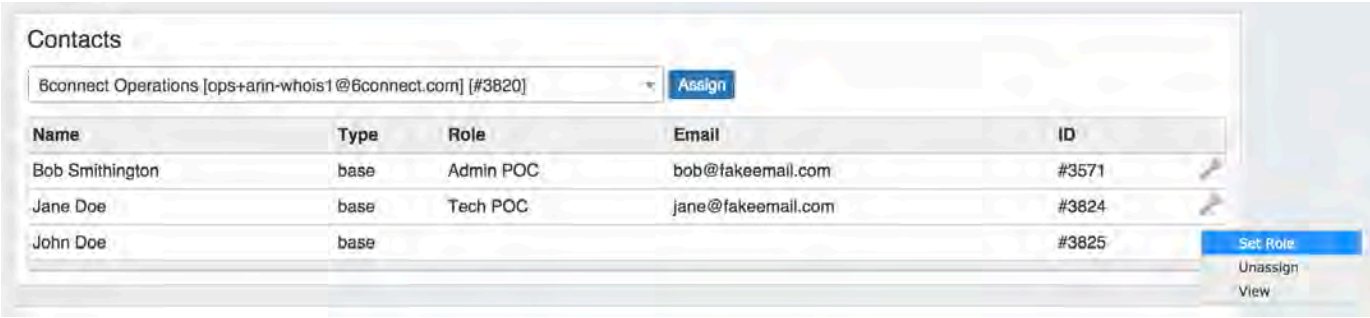


Under the "Select Role" section, choose the desired Role from the dropdown list, and click the "Set Role" button.



Delete a Role from the Roles List

Select "Set Role"

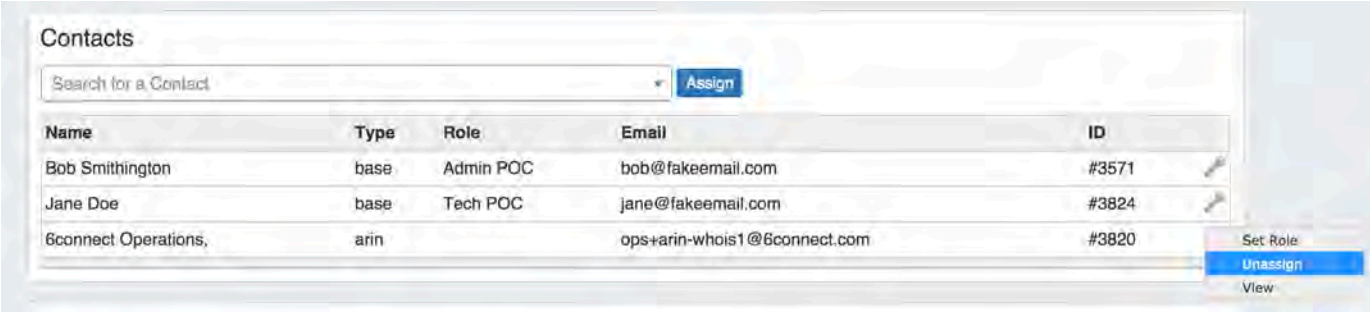


Under the "Delete Role" section, choose the desired Role from the dropdown list, and click the "Delete" button.

This removes the role from the "Select Role" list.

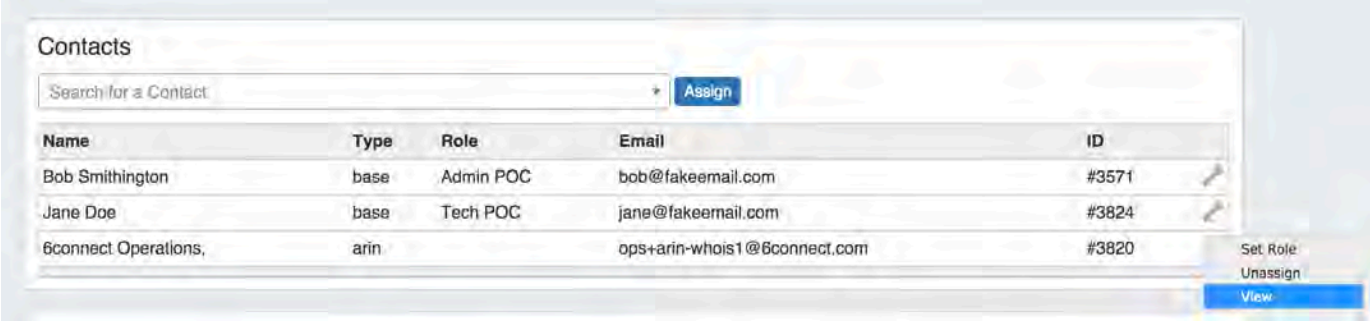
Unassign a Contact

To unassign a contact from a resource, in the Contacts gadget click the Action Menu, then select "Unassign".



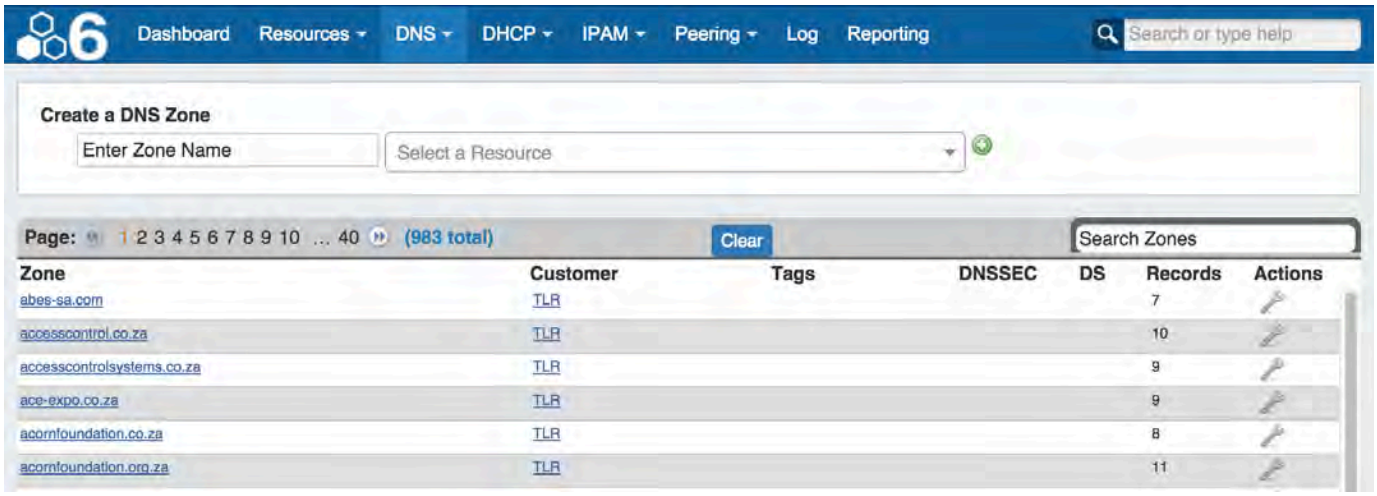
View the Contact Details Page

To view the contact's details, click the Action Menu, then select "View". You will be redirected to the contact's detail page.



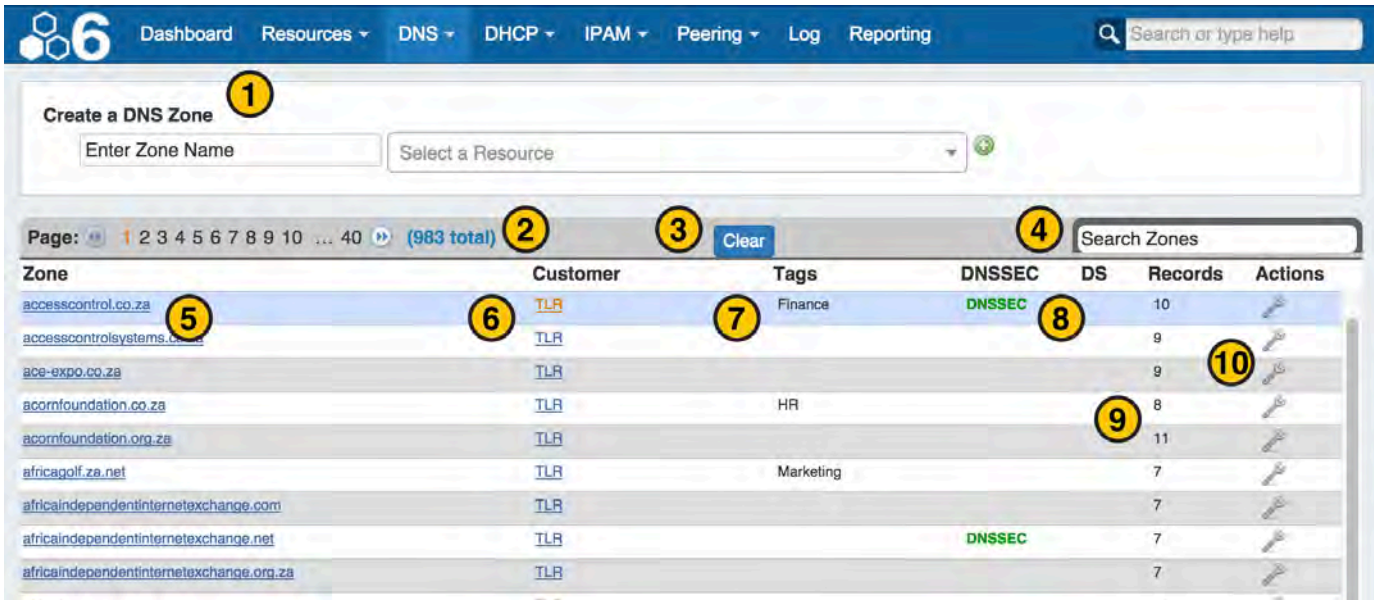
DNS Tab

DNS



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

DNS Tab User Interface



1) Create Zone Interface: Dialog for creating a DNS zone

2) Paging: this allows for easier browsing of large lists of DNS zones

3) Clear Button: Clears out the search box text

4) Search: this text box allows the user to enter in criteria to filter the list of zones

5) The Zone list: A click-able list of zone names - if clicked, the user will be directed to the DNS zone editing page

6) The Customer list: iA click-able list of Resource names that the zone is assigned to

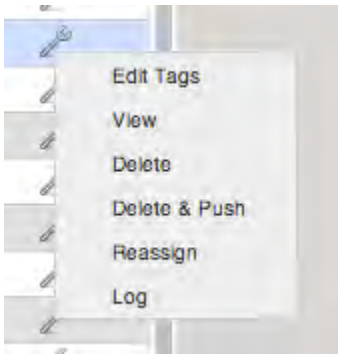
7) Tags: lists the tags associated with the zone

8) DNSSEC / DS Status: The **DNSSEC** column will show green if the zone has been signed and pushed successfully, the **DS** column will provide a status "X" to acknowledge that the zone was verified by an authenticated DNS server

9) Records: The records value is the number of zone records in the given zone

10) Action Menu: The Action Menu (wrench icon) gives a list of additional actions to perform on the 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.

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

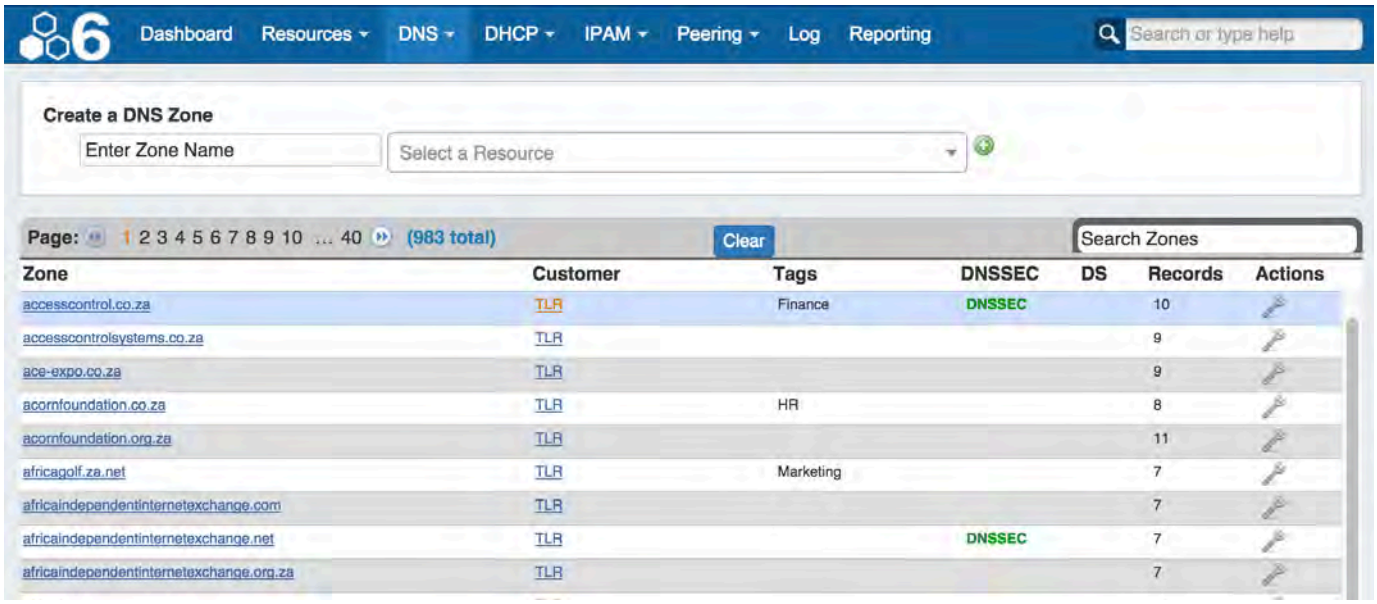
6) **Log:** Brings the user to the Log Tab with the results filtered for the specific zone

Additional Information

- Working with DNS Zones - Common Tasks
- Editing DNS Zones

Working with DNS Zones - Common Tasks

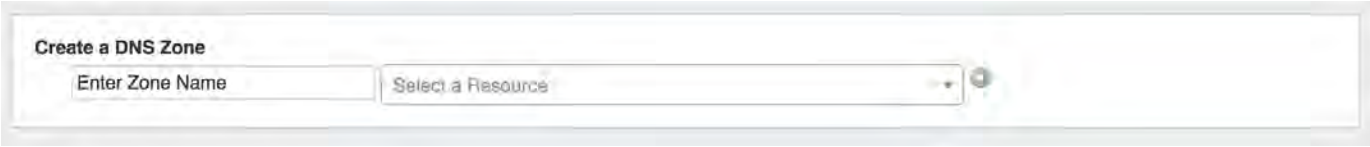
Working with DNS Zones - Common Tasks



This page details some common DNS tasks performed from the DNS tab and DNS Gadget. For tasks that require Admin access in ProVision, see Working with DNS Zones.

Creating/Adding Zones

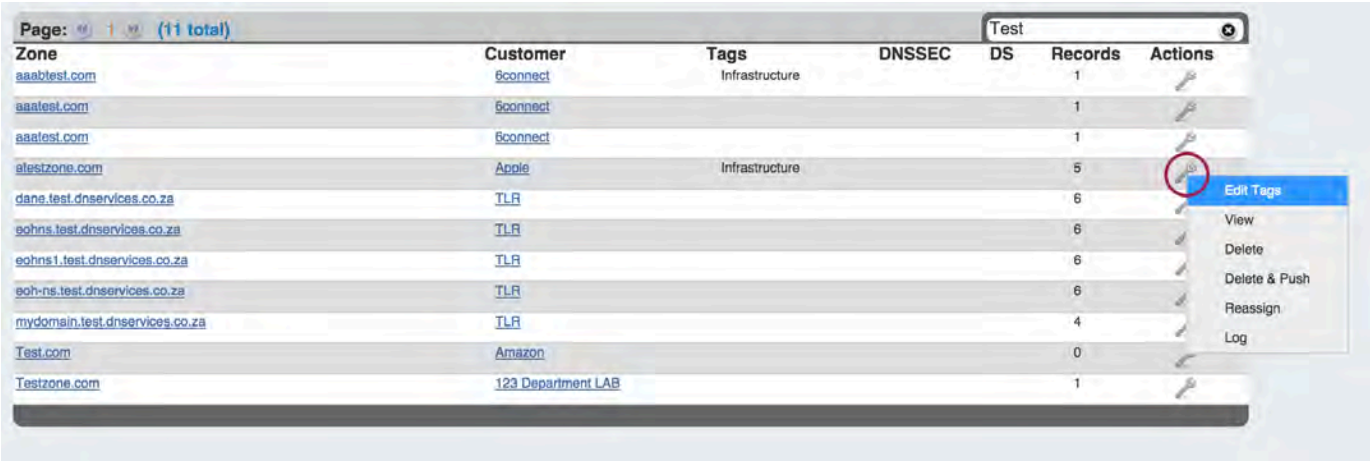
After clicking on the main **DNS** Tab, at the top of the page will be the "Create a DNS Zone" UI. To create a zone, enter the name of the zone and select the Resource you want to assign the zone to. Click on the green plus sign to be taken to the newly created zone file. There you can edit the zone, assign views, etc.



Editing DNS Tags

> Before you are able to add tags to a zone, the list of available tags must be set up in the Admin - DNS Admin section. See our DNS Administration documentation for more information on managing the DNS tags list.

To add tags to a zone, first find the zone in the DNS Tab zone list. Click on the Action Menu (wrench icon), and select "Edit Tags".

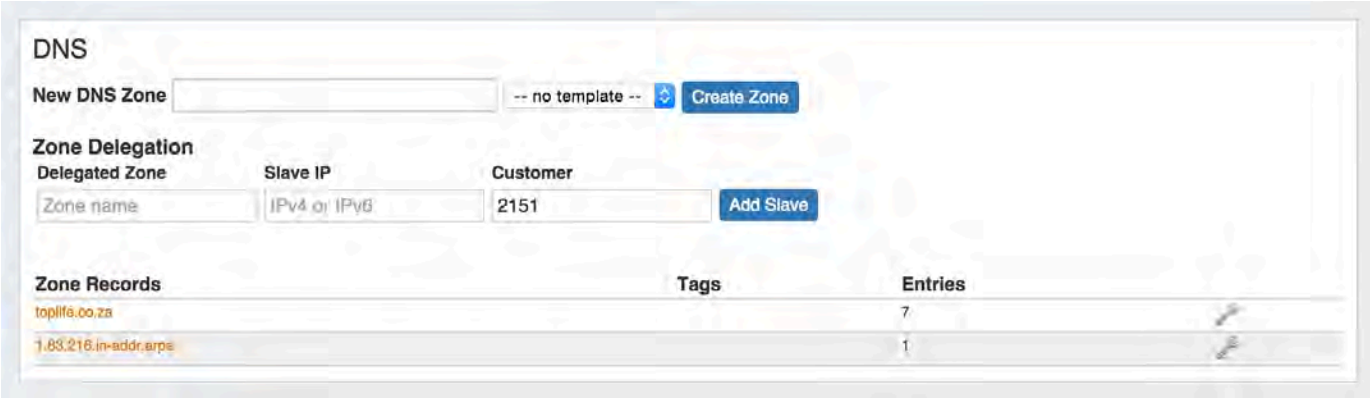


The "Edit Tags" interface will pop up.

Click inside the tag listing box, and select the desired tag(s) from the dropdown list. To remove a tag from the zone, click on the "x" in front of the tag. When you are done with your edits, hit the "Update" button.

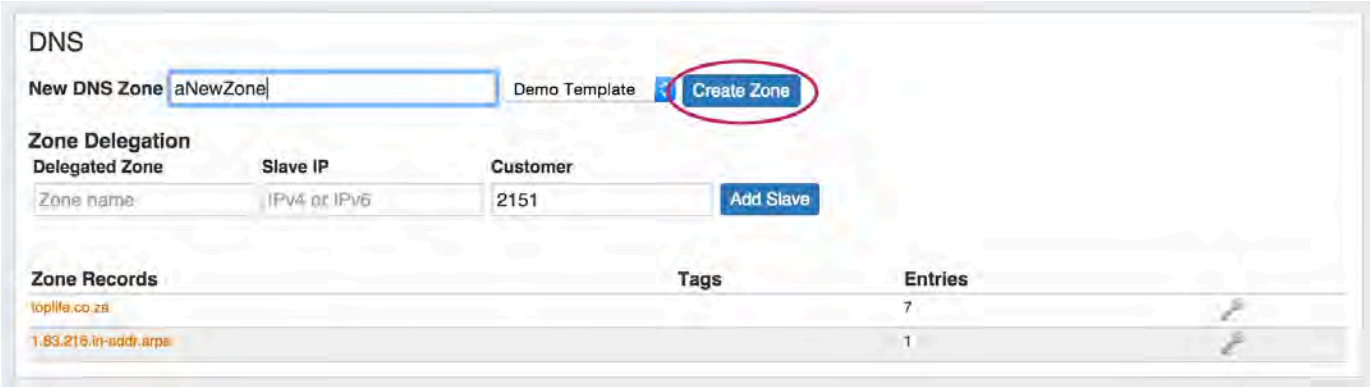
Using the DNS Gadget

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. Then, from the Resource's Entry page, you will see the DNS gadget and associated zones. From this interface, you can create new zones (with or without a Zone template) or assign Zone delegation specific information.

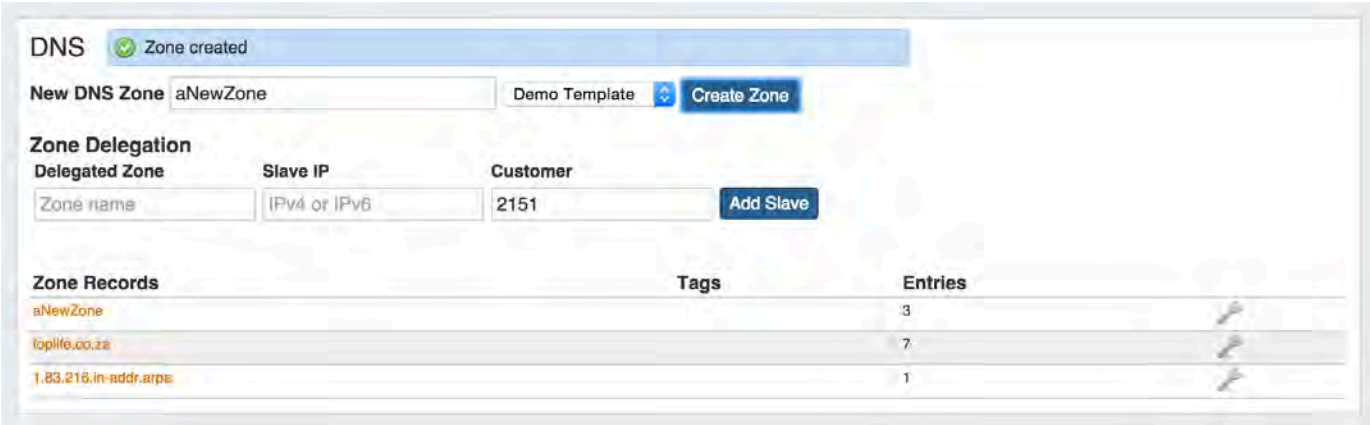


Creating/Adding Zones - DNS Gadget

To add a new DNS Zone from the DNS Gadget, enter the zone name in the field next to "New DNS Zone", select a template if applicable, and hit "Create Zone".

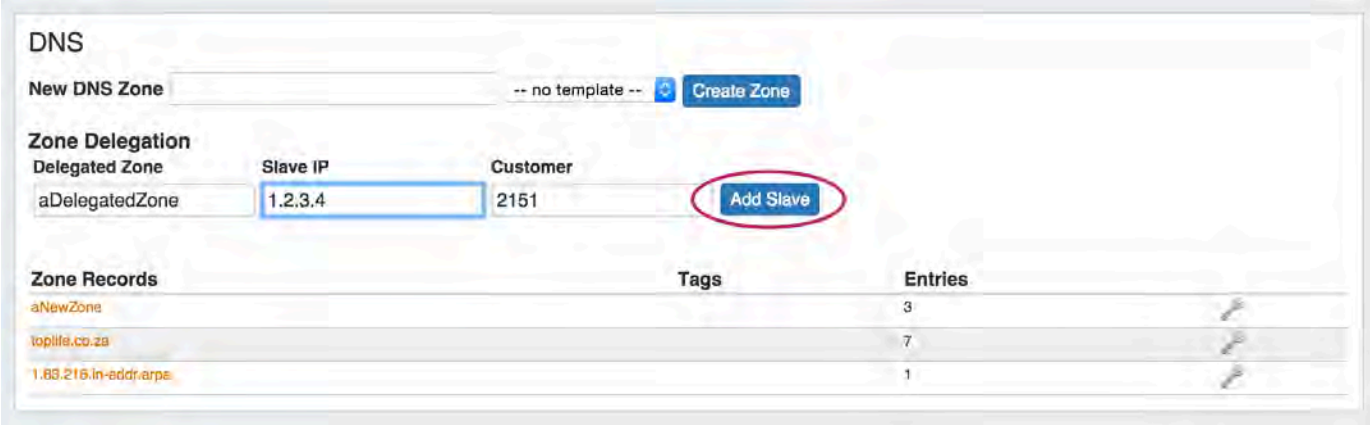


Your zone will be added to the list below, and a success message will appear at the top of the gadget.

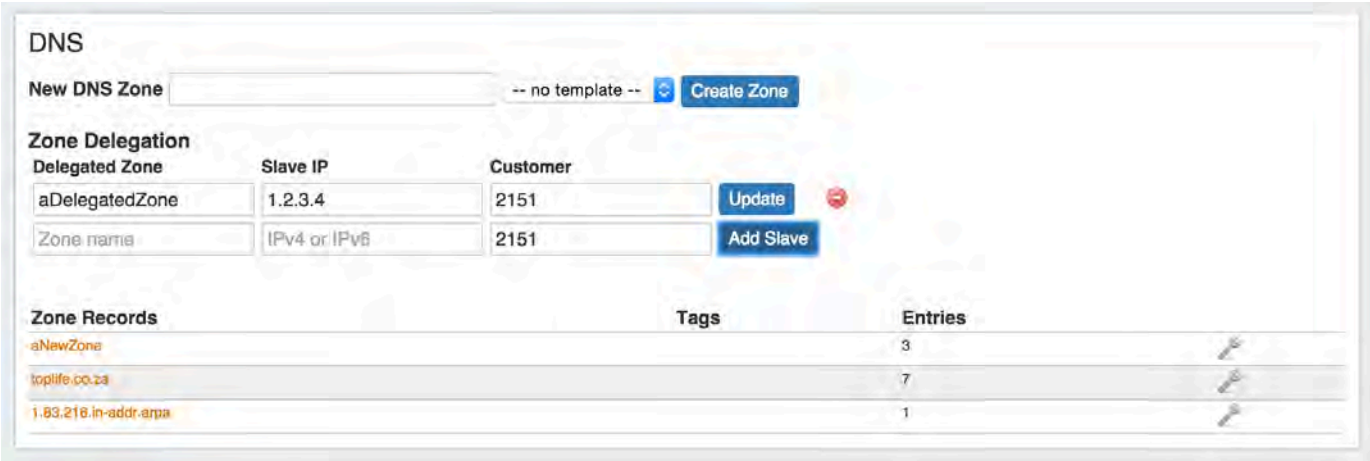


Zone Delegation - DNS Gadget

To delegate a zone, enter in the delegated zone name and the slave IP, and hit "Add Slave". The Customer number is the numerical ID for the Resource you are under, and it is automatically filled in.



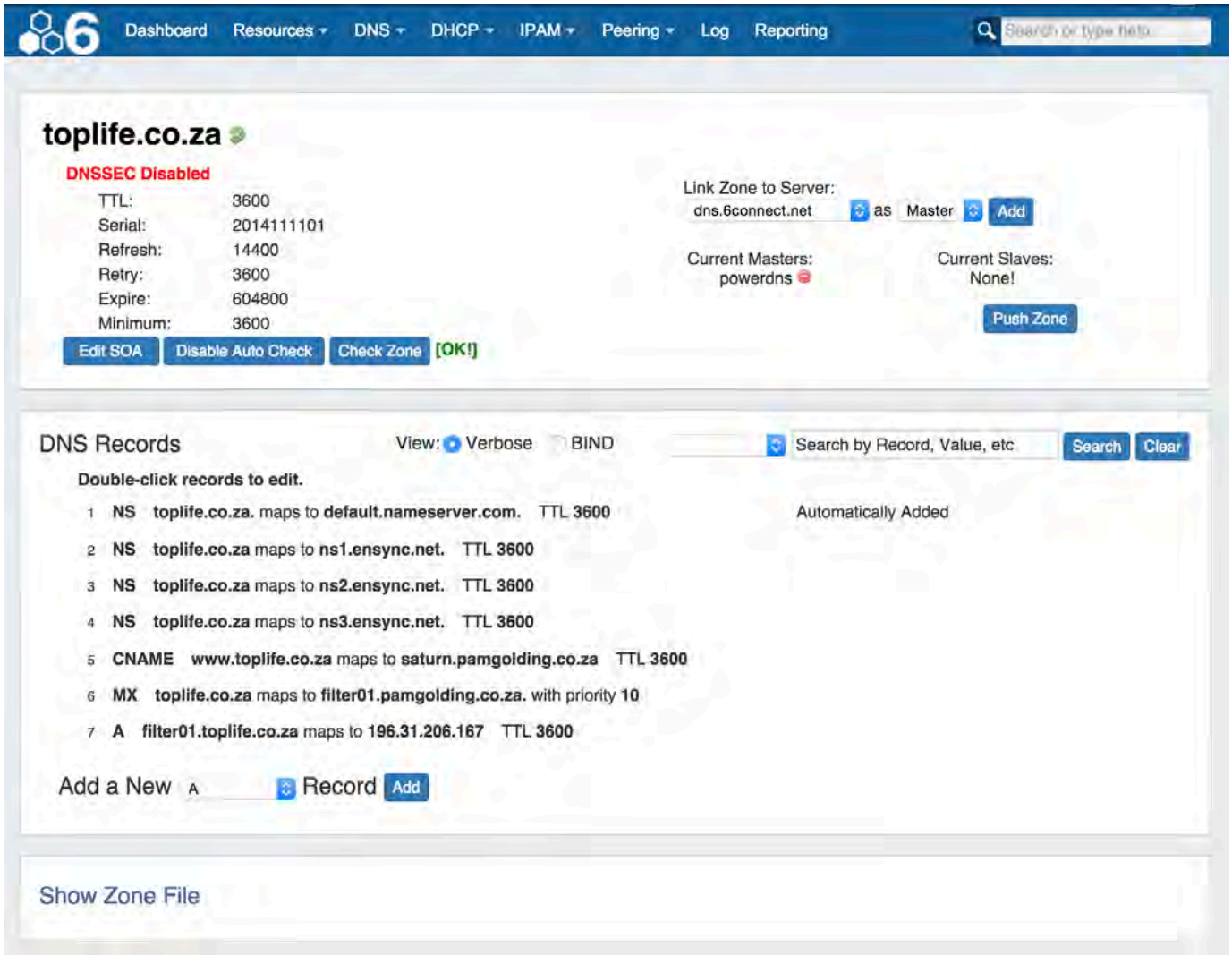
After adding the slave, it will show under the Zone Delegation list. You can make edits to the fields at any time simply by clicking in the field and typing your edits, then clicking the "Update" button. To Delete an entry, click the red "Delete" icon.



Edit Zones - DNS Gadget

Clicking on the Action Menu (wrench icon) next to a zone record and selecting "DNS View", "DNS Edit", or clicking on a zone name, will take you to the zone view / edit page.

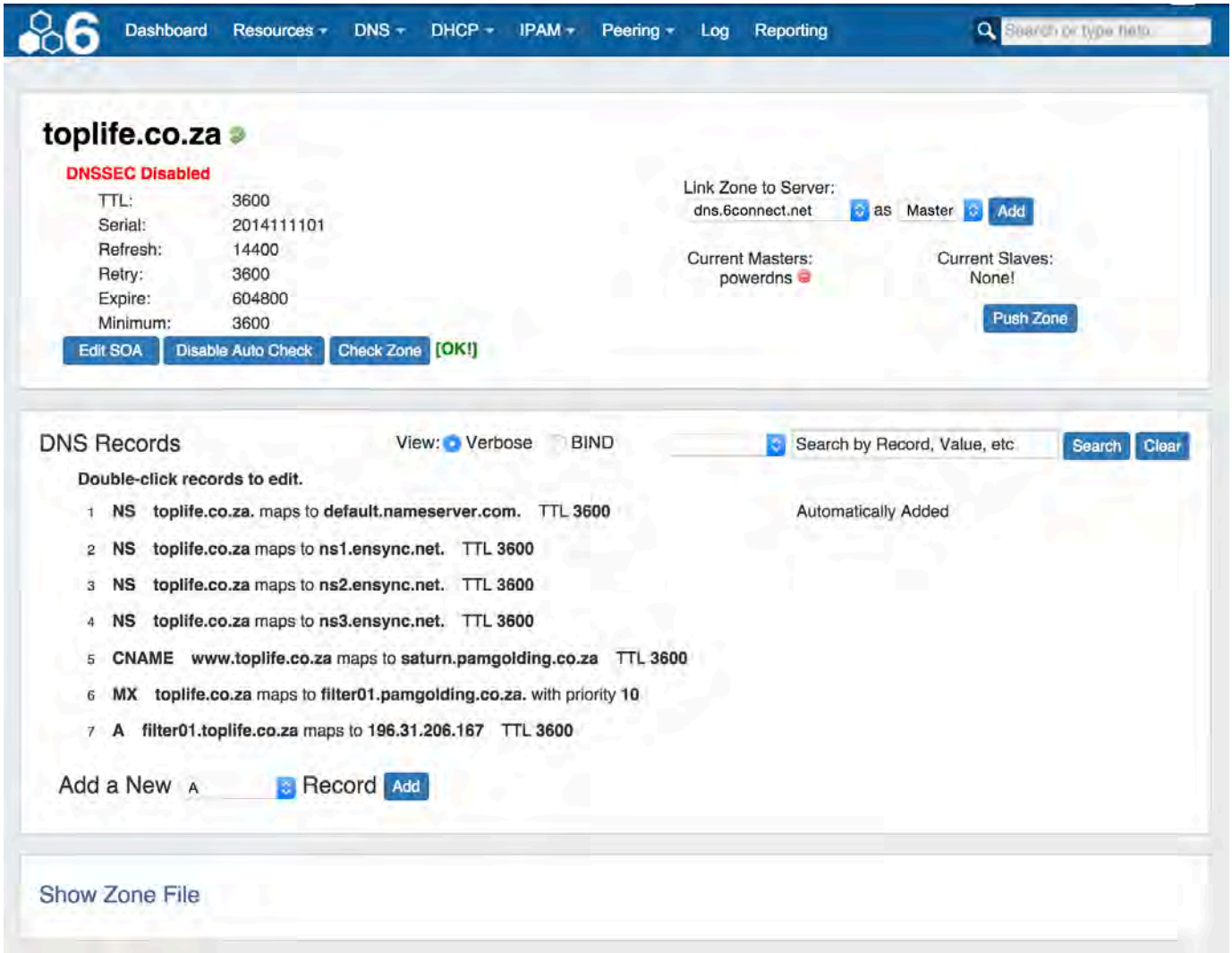
From this page you can edit DNSSEC, SOA, check zone status, link zones to servers, edit records, view the zone file, and view the zone history.



For a more detailed description on the view zone page, continue on to Editing DNS Zones.

Editing DNS Zones

Editing DNS Zones - The Zone View Page



Editing a Zone Record is done on the Zone View page. You can access this page two different ways:

1. From the DNS Gadget, click on the Action Menu (Wrench Icon) and select "View Zone" or "Edit Zone". This will take you directly to the Zone View page.
2. Click on the zone name from the DNS Tab, DNS Gadget, or the DNS Admin Bulk Zone search. The name links directly to the Zone View page.

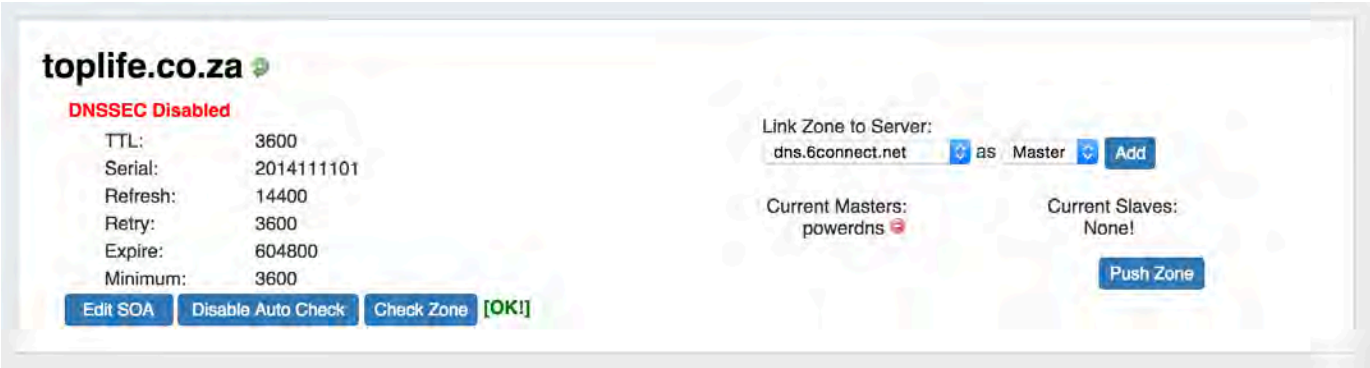
Areas of the Zone View page include:

- Editing DNS Zones - The Zone View Page
  - Zone Management:
    - Enable / Disable DNSSEC
    - Edit SOA
    - Link Zone to Server
    - Schedule Push
  - DNS Zone Record Data:
    - Edit a Zone Record
  - Show Zone File:
  - Show Zone History

Zone Management:

This area is at the top of the screen and provides direct access to confirm zone file changes.

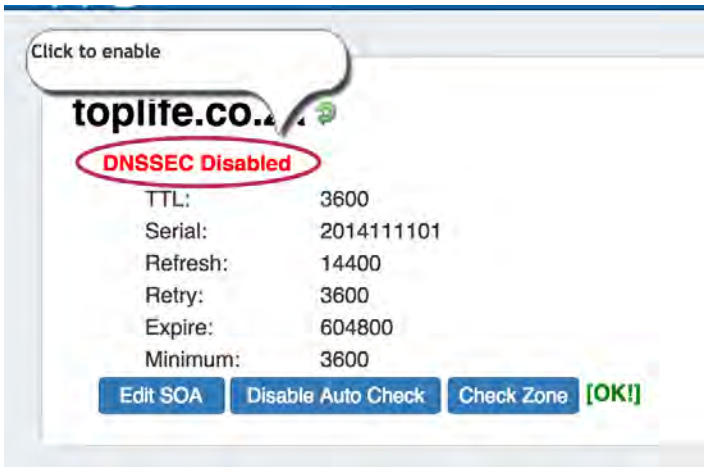
By clicking the "Check Zone" button, we automatically confirm that your zone is verified and highlight any problem entries. If no errors are detected, it will result in an "OK!" message. Once verified, you have the option to Push the Zone to the specified server(s) selected. You may disable autochecking the zone by clicking "Disable Auto Check".
***Note: When zones are written the serial number is incremented and DNSSEC refreshed (if enabled)**



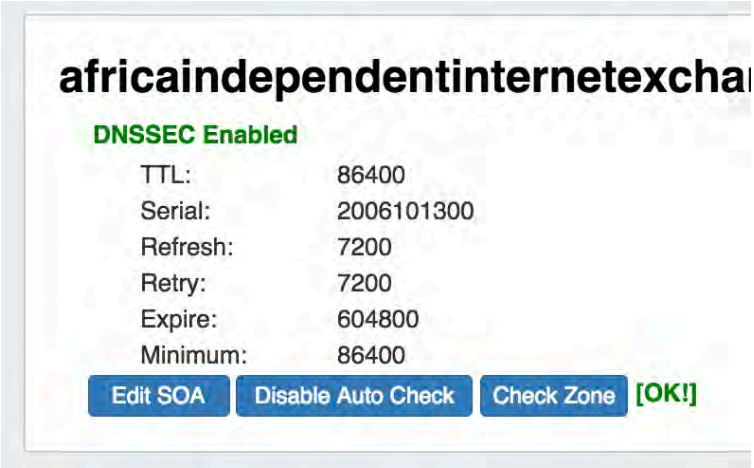
If errors are detected, the relevant zone record entries will be highlighted to show the error condition and the user will be prompted to fix them before being able to push the zone. The validation is for RFC compliance.

Enable / Disable DNSSEC

DNSSEC for a zone can be enabled / disabled by clicking on the DNSSEC status below the zone name.

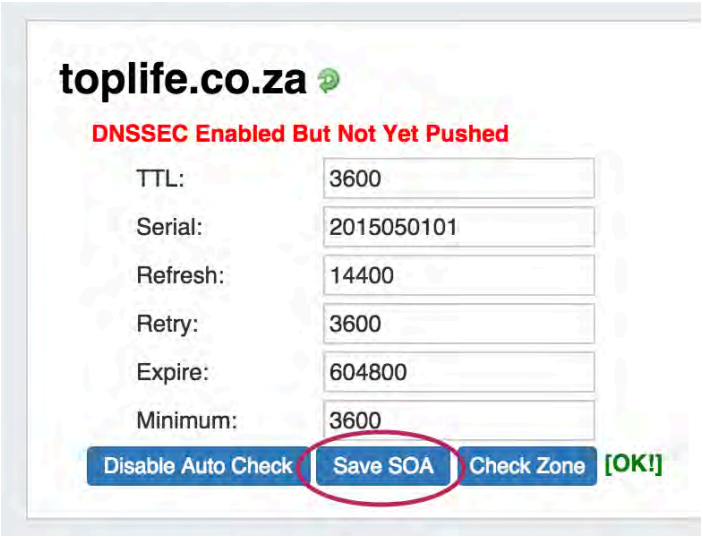


Once enabled, the message will change to either "DNSSEC Enable but not yet Pushed" or "DNSSEC Enabled" if pushed.



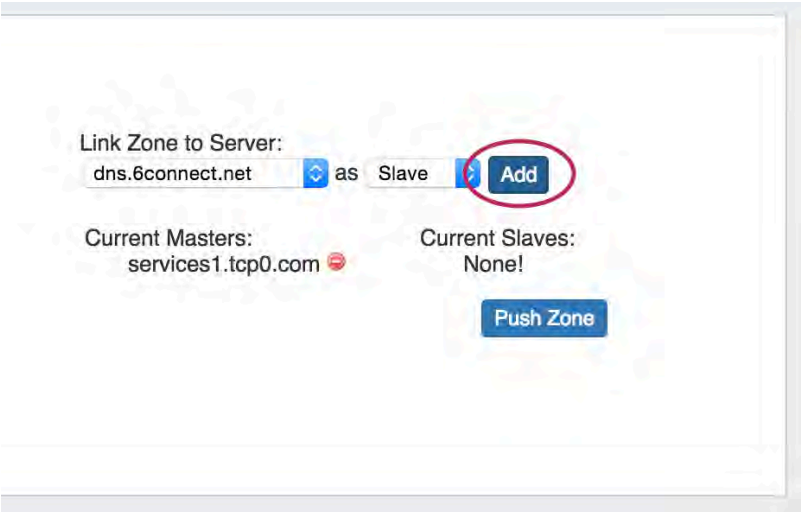
Edit SOA

Click on the "Edit SOA" button to make the SOA fields editable. Make your changes, then click "Save SOA" to save your changes.

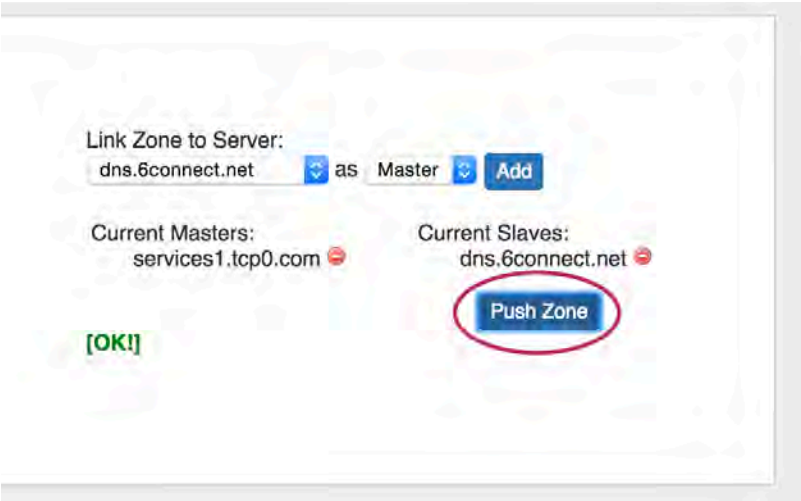


Link Zone to Server

To Link the zone to a server, select the sever from the dropdown list, and then specify whether it should be a "Master" or "Slave". The, hit the "Add" button. The desired server must already be created in the **DNS Admin** Tab.

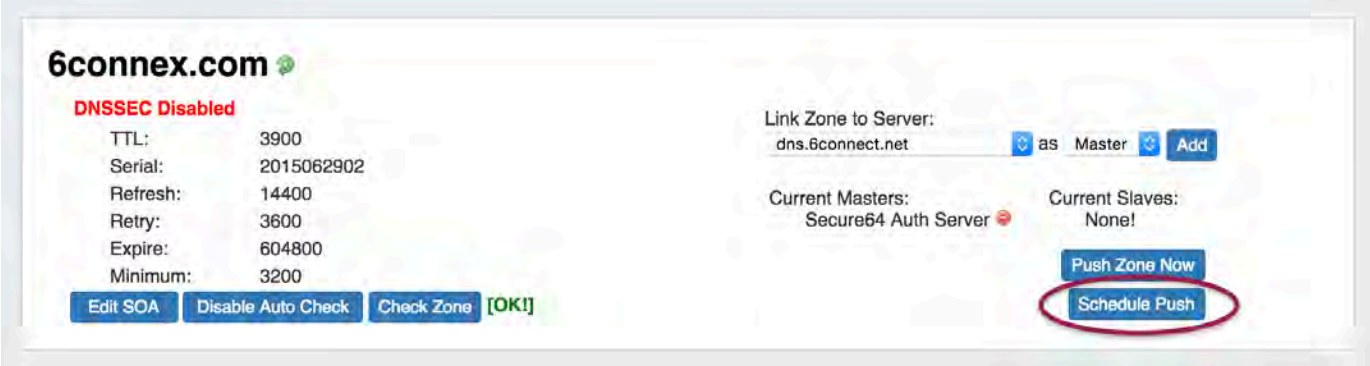


When done adding Master / Slave servers to the list, hit "Push Zone Now". If the push is successful, you will see an "OK!" message pop up under the linked server list.

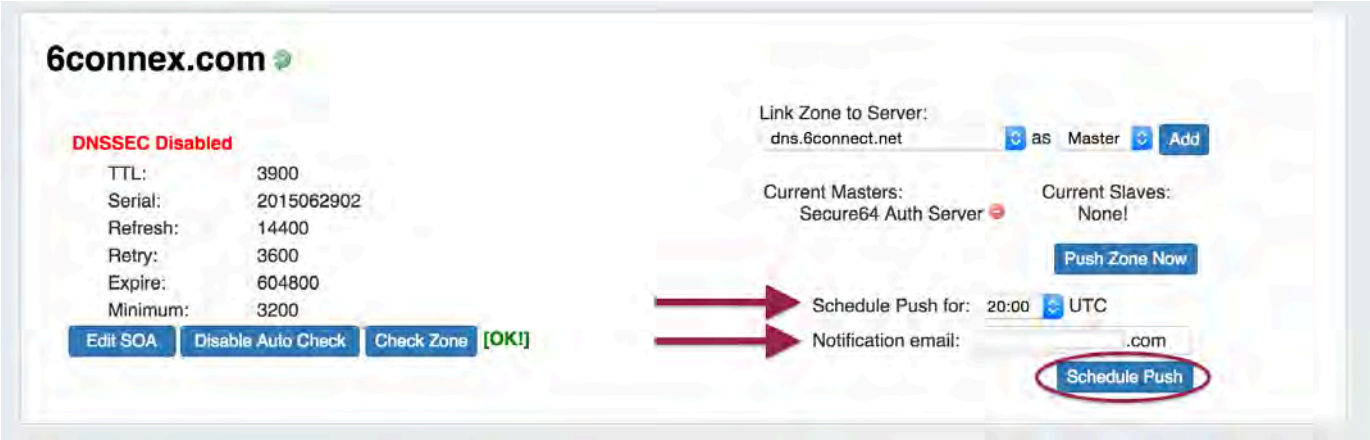


Schedule Push

If you have Admin access, a "Schedule Push" button will be visible below the "Push Zone Now" Button. To schedule a DNS Zone push, hit "Schedule Push".



Select the desired UTC time and fill in the notification email address. When done, click on "Schedule Push".



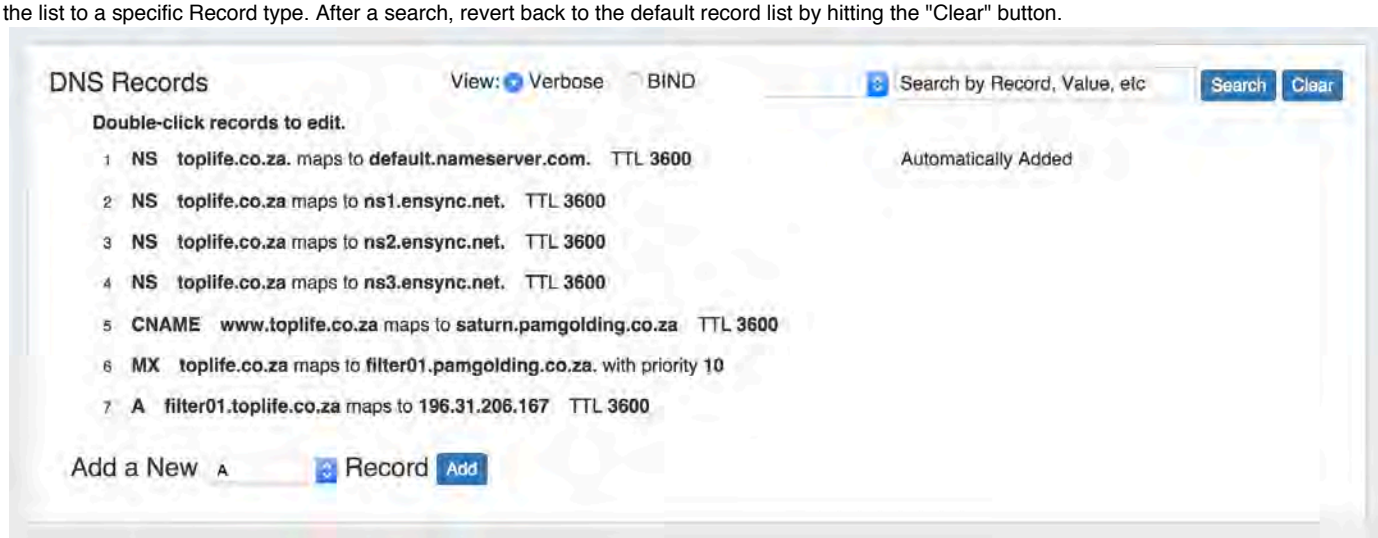
When done, you will see a notification that a push is scheduled. It will also show in the scheduled task list on the Admin Scheduler.

You may edit or delete the scheduled push via the Scheduler, similarly to any other scheduled task.

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. You can use the dropdown next to the search box to narrow the list to a specific Record type. After a search, revert back to the default record list by hitting the "Clear" button.



Edit a Zone Record

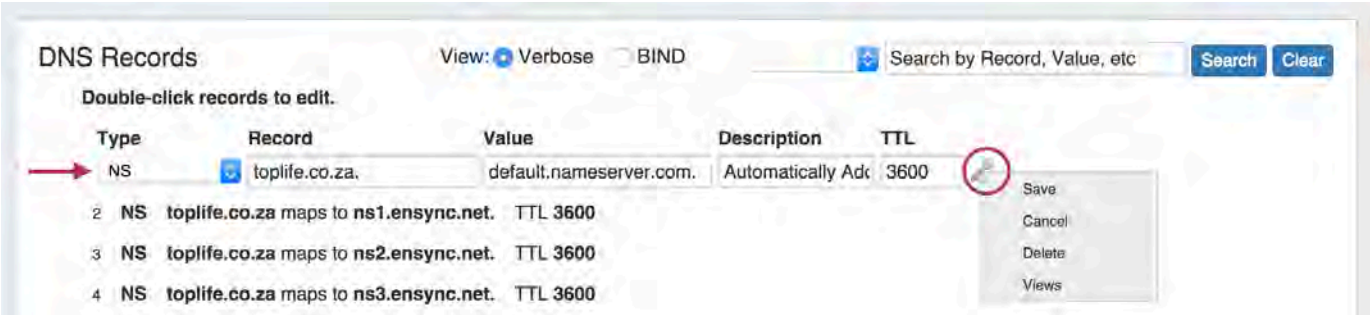
To Edit a Zone Record, simply double-click on it the entry and make any required edits. Use the "Wrench" icon for the action context menu to:

1) **Save** your changes to the zone records

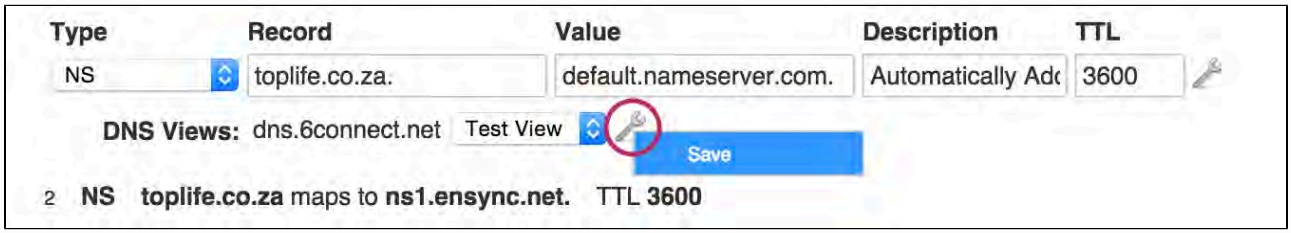
2) **Cancel** your edits to the zone record

3) **Delete** the zone record

4) See **Views** set up (if applicable) and apply to the record.



Configuring Views per DNS Zone
If Views are enabled and created on the DNS server assigned to this zone, you will also have the "Views" option that will bring up the view assignment menu. You will be able to select the existing View(s) that you wish to apply to the zone record here. After selecting the view, click the Action Menu (wrench icon) to save your changed. Note: If views are enabled on the DNS server assigned, but not yet created with a View Name, no list will appear.

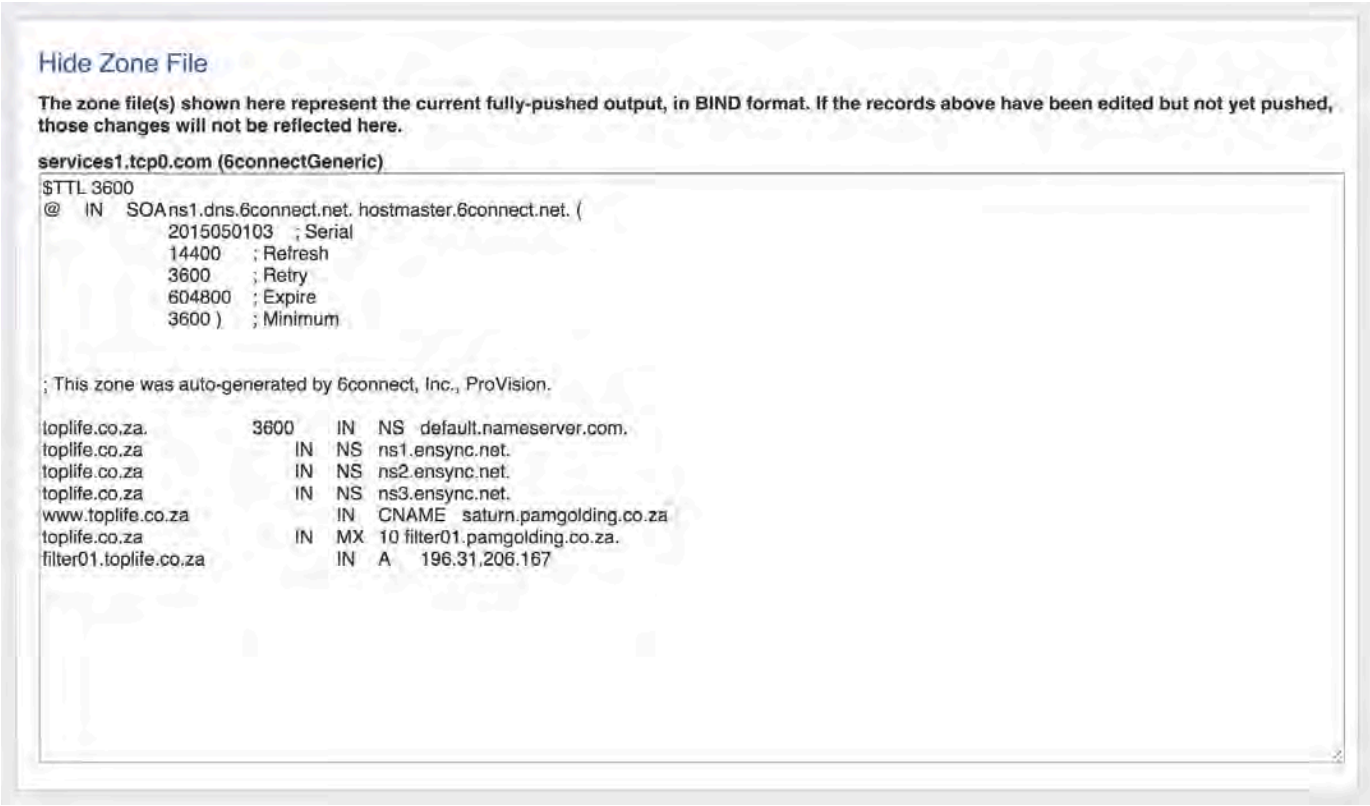


For more information on setting up Split Horizon/Views support - go here.

Show Zone File:

Click on the "Show Zone File" link to open the view.

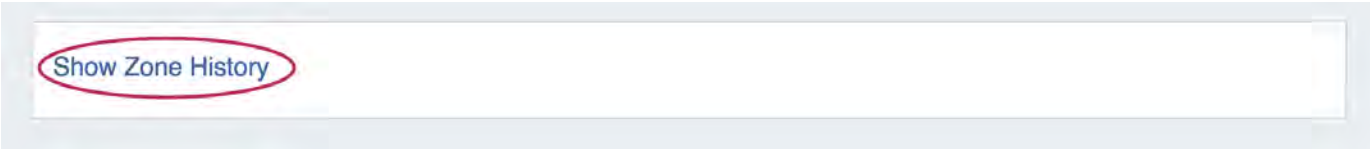
This view gives you a "CLI type" view of the zone file. If views are enabled, you will see those zone files as well. Please note that this is a read only screen.



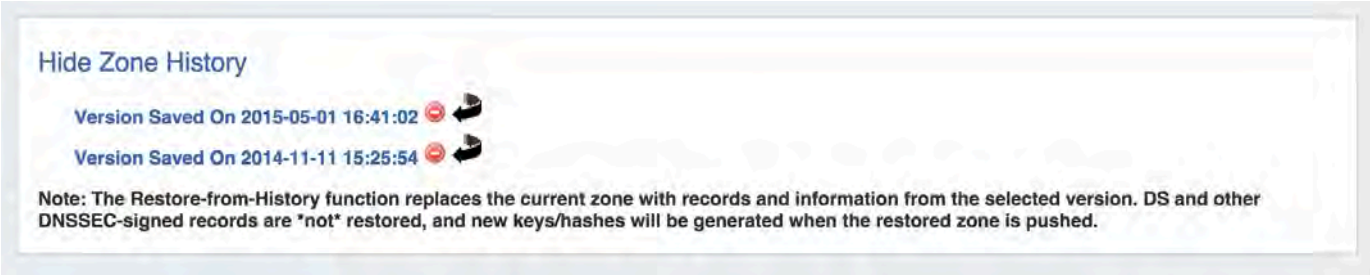
When done viewing the zone, you can click "Hide Zone File" to once again collapse the view.

Show Zone History

Click on the Show Zone History link to expand the view. Note that the zone has to actually be pushed for the Zone History area to show up on the screen.

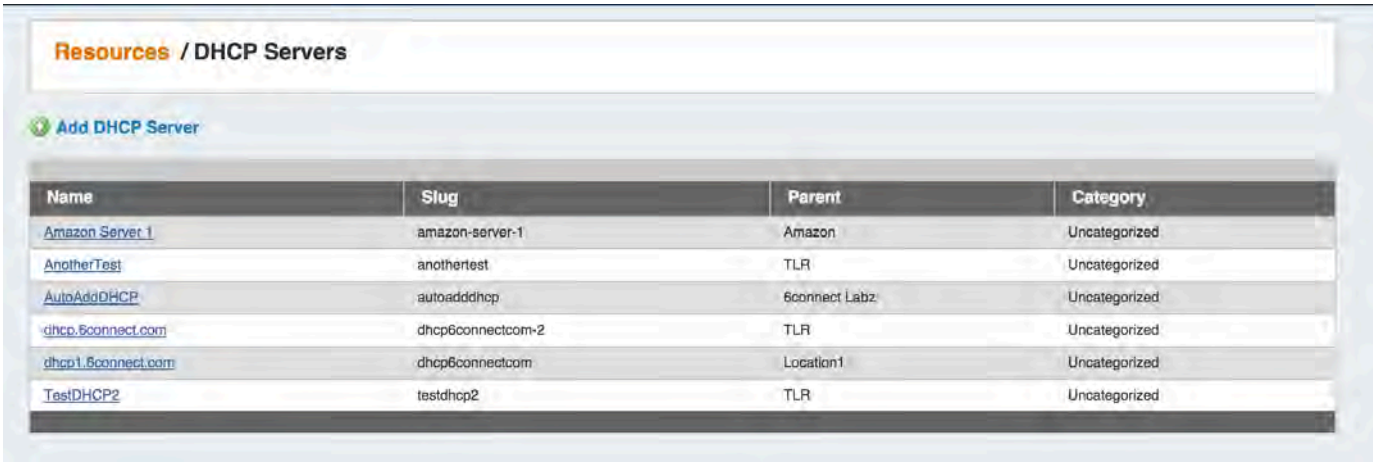


The feature allows you to revert/reload previous zone versions. Click the arrow icon to revert, or the red delete icon to delete the version.



DHCP Tab

The DHCP Tab



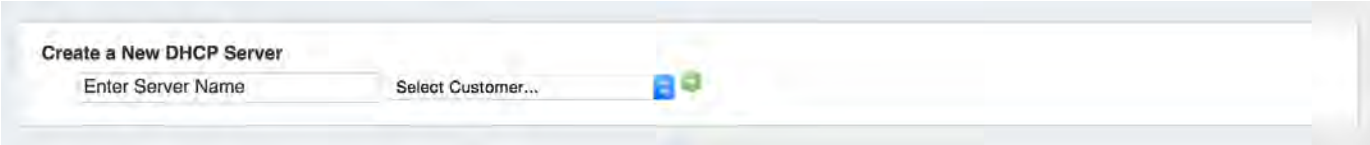
The **DHCP** Tab provides an overview of DHCP servers currently existing in ProVision, and links to the Resource pages for those servers. In ProVision, DHCP server configurations are tied into the Resource Manager system, so it is essential to associate servers with the proper parent Resource.

Adding DHCP Servers

To add a DHCP Server to ProVision from the **DHCP** Tab, click on the "Add a DHCP Server" link. The create server dialog area will open.

Type the server name, then under "Select Customer", choose the resource to which the DHCP server belongs. This creates a hierarchical relationship, with the server as a child resource under the selected parent.

When you are done providing this information, click on the green "+" icon to add the server.

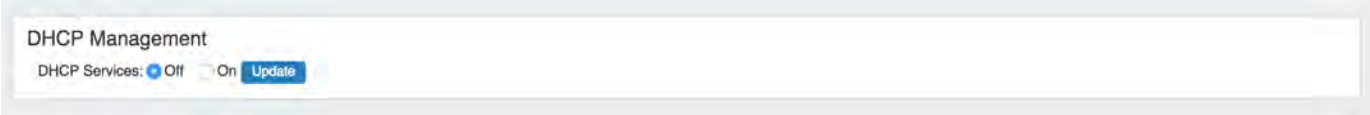


Working with the DHCP Gadget

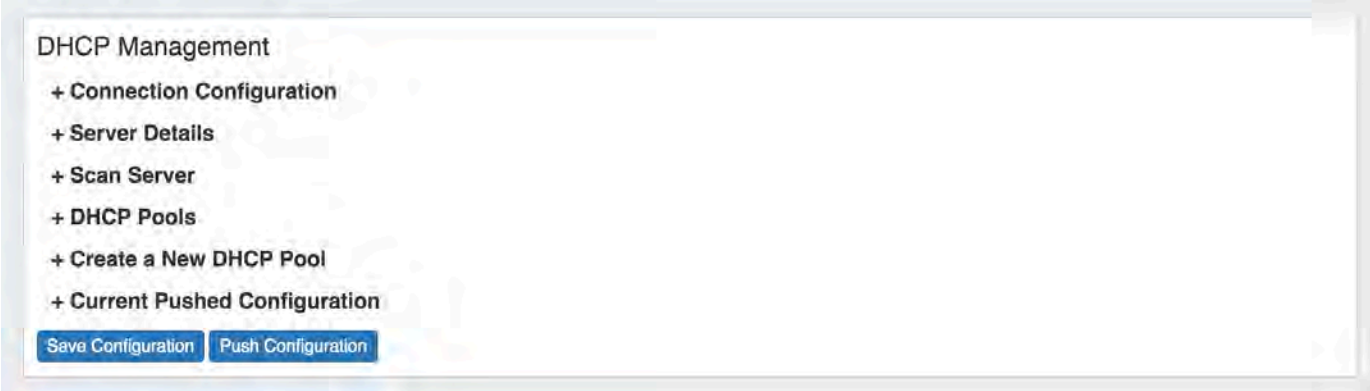
Before You Begin:

Attach / Enable the DHCP Gadget

Verify that the DHCP Gadget is attached to the Resource Section. Then, you can enable the gadget by selecting the radio button next to "On", and click "Update".



Once enabled, the DHCP Management Gadget will show sections for Connection Configuration, Server Details, Scan Server, DHCP Pools, Create a New DHCP Pool, and Current Pushed Configuration.

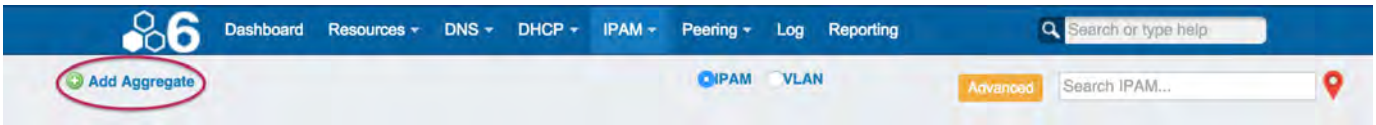


These sections will be reviewed in detail further on in this section.

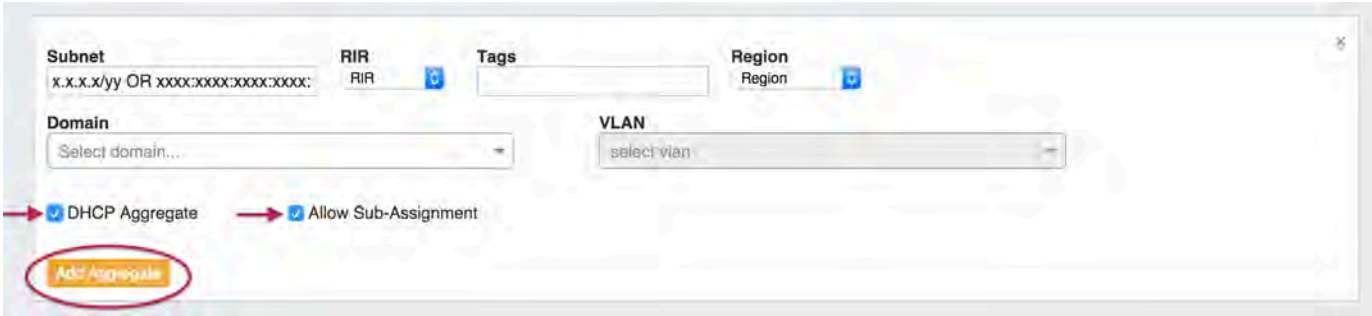
Defining DHCP Scopes - Create DHCP Aggregates

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

Under the **IPAM** tab, select "Add Aggregate". It will open the Add Aggregate dialog.



Fill in the aggregate information, and select the "DHCP Aggregate" checkbox option as outlined below. "Allow Sub-Assignment" will be selected automatically,so that smaller blocks may be assigned to DHCP pools.



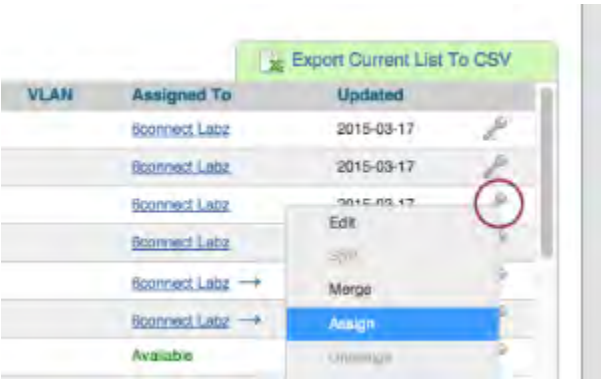
This will ensure the block is automatically added to the DHCP Available Resource, and thus usable when building DHCP Server Configurations and defining DHCP Pools.

When done, click "Add Aggregate"

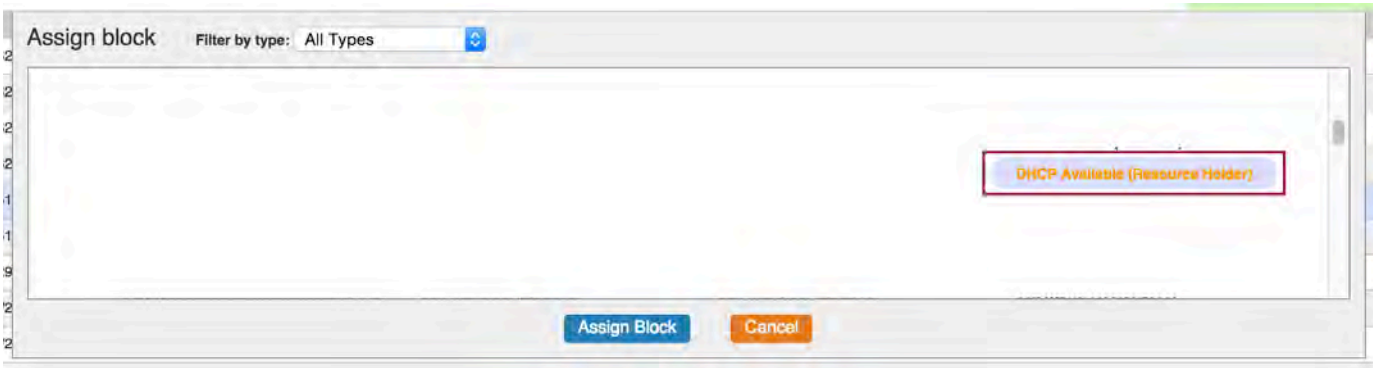
Using an Existing Aggregate

If you would like to use an existing aggregate or part of an existing aggregate, you simply need to "assign" the block to the Resource Holder "DHCP Available".

1) From the IPAM Manage screen, click on the Action Menu, then select "Assign".



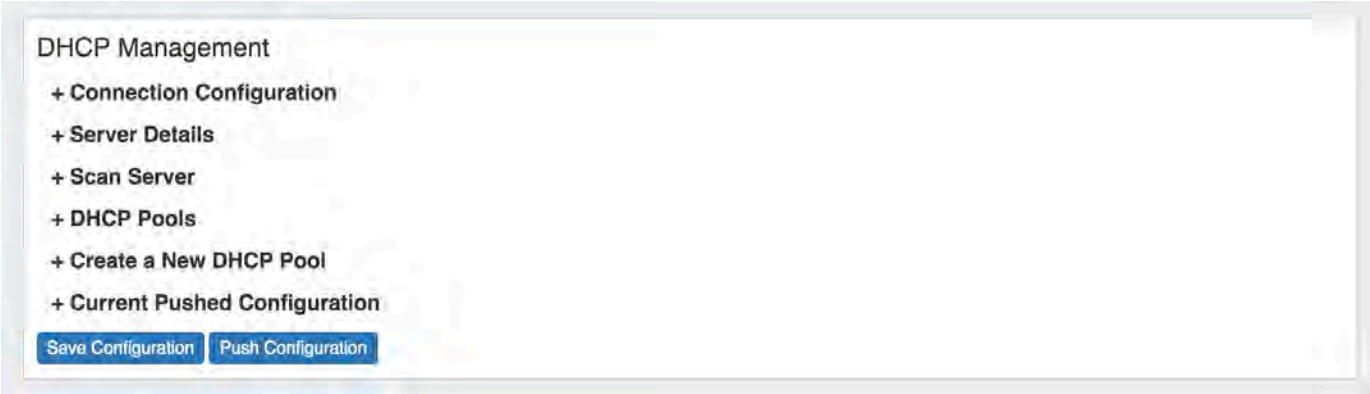
2) In the Assign Block dialog, scroll until you find "DHCP Available". Select it, then click on "Assign Block".



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

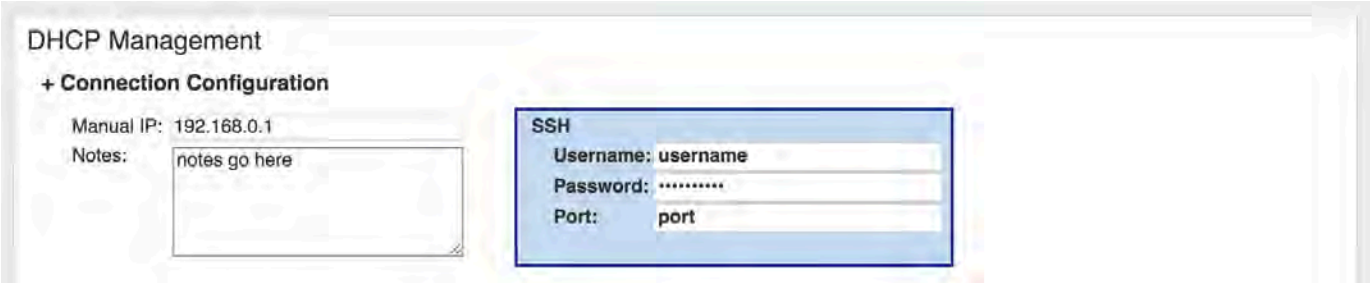
Managing DHCP Server Configurations

Once DHCP functions are enabled for a Resource Section, you will be able to manage configurations per Resource by expanding the relevant areas on the Resource's Entry page.



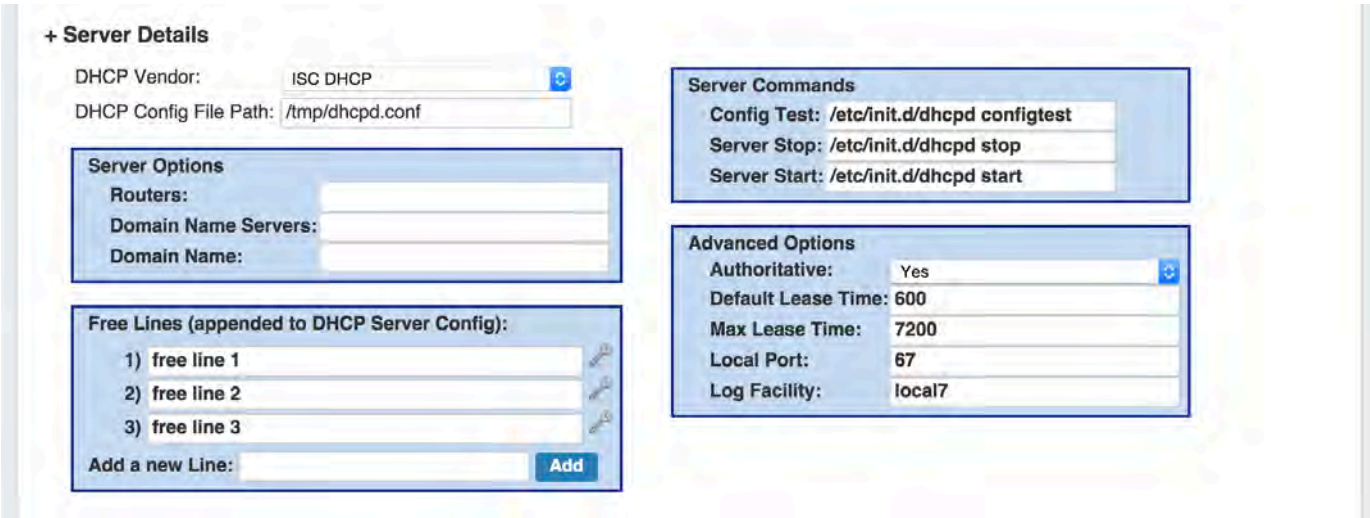
Connection Configuration

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



Server Details

Server details and advanced options may be entered under this portion of the gadget.



Server Command: Config Test
A note on the Server Command: Config Test option:

This command is run after the newly-written DHCP Config file has been transferred to the server but before the server is restarted. Due to the wide variety of DHCP install configurations that can arise on different systems, it is best if the Config Test option is as explicit as possible. For example:

sudo /usr/sbin/dhcpd -t -cf /etc/dhcp/dhcpd.conf

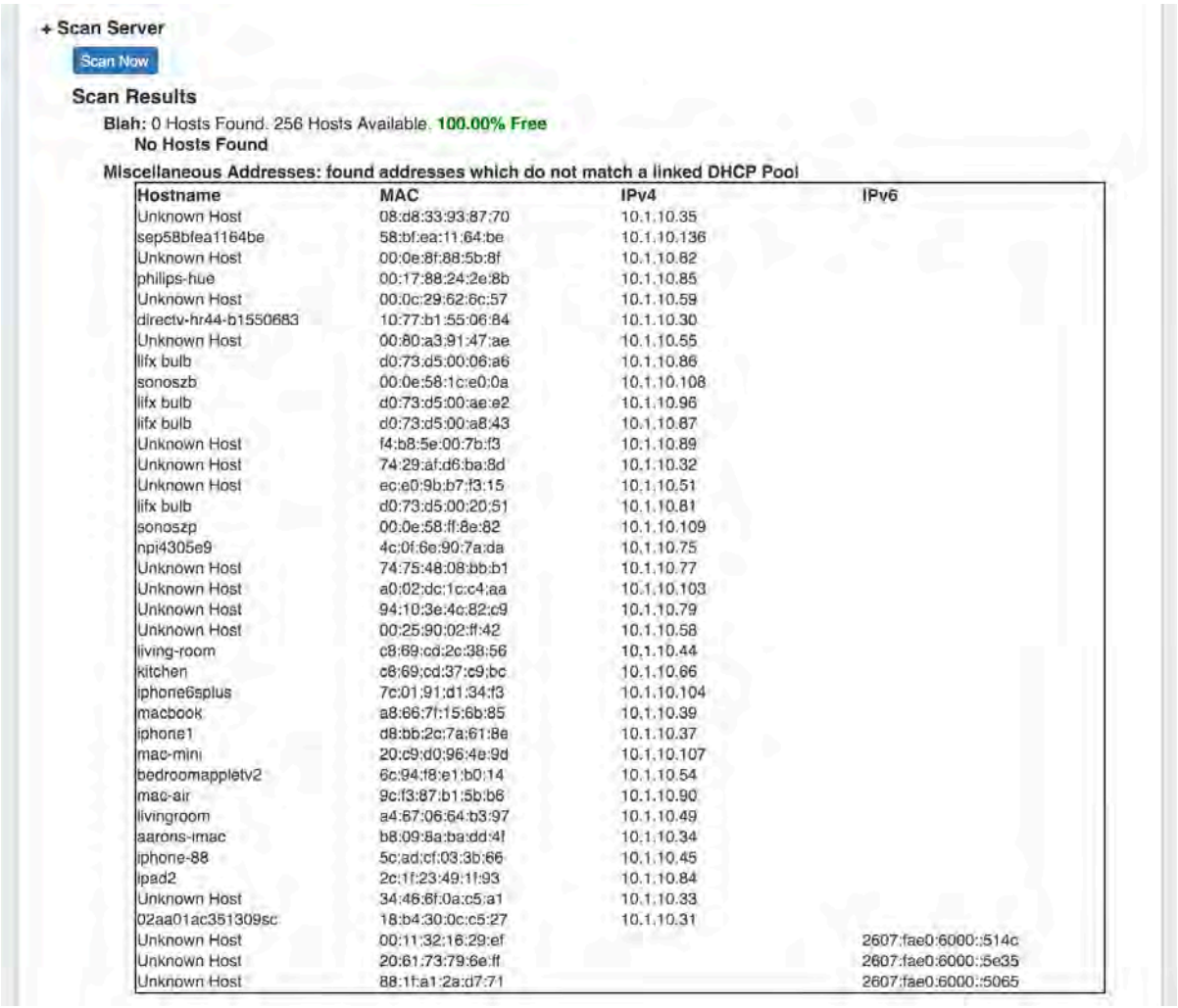
This command explicitly requests the DHCP daemon start in test-only mode (-t) with a specific config file (-cf). This config file should be

the location supplied in the "DHCP Config File Path" section.

Scan Server

Scan Server provides a "Scan Now" button, which scans the DHCP server and returns found IPs under DHCP Pools.

The returned IPs are divided by Pools and IPv4/IPv6. Returned fields include MAC address, IP address, and name. The percentage utilization of IP space out of the available DHCP Pool space is also included at the top of the list.



Hostname	MAC	IPv4	IPv6
Unknown Host	08:d8:33:93:87:70	10.1.10.35	
sep58bfea1164be	58:bf:ea:11:64:be	10.1.10.136	
Unknown Host	00:0e:8f:88:5b:8f	10.1.10.82	
philips-hue	00:17:88:24:2e:8b	10.1.10.85	
Unknown Host	00:0c:29:62:6c:57	10.1.10.59	
directv-hr44-b1550683	10:77:b1:55:06:84	10.1.10.30	
Unknown Host	00:80:a3:91:47:ae	10.1.10.55	
lifx bulb	d0:73:d5:00:06:a6	10.1.10.86	
sonoszb	00:0e:58:1c:e0:0a	10.1.10.108	
lifx bulb	d0:73:d5:00:ae:e2	10.1.10.96	
lifx bulb	d0:73:d5:00:a8:43	10.1.10.87	
Unknown Host	f4:b8:5e:00:7b:f3	10.1.10.89	
Unknown Host	74:29:af:d6:ba:8d	10.1.10.32	
Unknown Host	ec:e0:9b:b7:f3:15	10.1.10.51	
lifx bulb	d0:73:d5:00:20:51	10.1.10.81	
sonoszp	00:0e:58:ff:8e:82	10.1.10.109	
npi4305e9	4c:0f:6e:90:7a:da	10.1.10.75	
Unknown Host	74:75:48:08:bb:b1	10.1.10.77	
Unknown Host	a0:02:dc:1c:c4:aa	10.1.10.103	
Unknown Host	94:10:3e:4c:82:c9	10.1.10.79	
Unknown Host	00:25:90:02:ff:42	10.1.10.58	
living-room	c8:69:cd:2c:38:56	10.1.10.44	
kitchen	c8:69:cd:37:c9:bc	10.1.10.66	
iphone6splus	7c:01:91:d1:34:f3	10.1.10.104	
macbook	a8:66:7f:15:6b:85	10.1.10.39	
iphone1	d8:bb:2c:7a:61:8e	10.1.10.37	
mac-mini	20:c9:d0:96:4e:9d	10.1.10.107	
bedroomappletv2	6c:94:f8:e1:b0:14	10.1.10.54	
mac-air	9c:f3:87:b1:5b:b6	10.1.10.90	
livingroom	a4:67:06:64:b3:97	10.1.10.49	
aarons-imac	b8:09:8a:ba:dd:4l	10.1.10.34	
iphone-88	5c:ad:cf:03:3b:66	10.1.10.45	
ipad2	2c:1f:23:49:1f:93	10.1.10.84	
Unknown Host	34:46:6f:0a:c5:a1	10.1.10.33	
02aa01ac351309sc	18:b4:30:0c:c5:27	10.1.10.31	
Unknown Host	00:11:32:16:29:ef		2607:fae0:6000::514c
Unknown Host	20:61:73:79:6e:ff		2607:fae0:6000::5e35
Unknown Host	88:1f:a1:2a:d7:71		2607:fae0:6000::5065

If the scan is unable to be completed, an error message will appear in the DHCP Management section.

DHCP Pools

In this area, 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.

Use the Action menu to Link to Server, Delete, or Remove Link from Pools.

Additional edits may be made to pools by clicking on the Pool name. This allows the Pool name, assignment, and lines to be changed. Be sure to hit "Save" after making changes.

DHCP Pool Search:

Both Linked and Existing DHCP pools have field-specific search options to filter the DHCP pool list(s). Select the search term type from the dropdown list, type the search term in the search box, then hit "Search".

For both Linked and Existing pools, search terms include:

Name: The full or partial name of the pool (ex: "Lab")

MAC: The full (ex: 00:11:22:33:44:55:66:77) or partial (:22) MAC address, for pools that were created as "Host" type. MAC address searches must include a colon.

IP: Search by IP Block by providing the CIDR assigned to the pool. *(ex: 10.0.0.0/24)*

For Existing Pools, additional filter terms may be included to further filter the IP search by Region or assigned Resource.

After entering your search terms, click "Search", and the pool list will filter to the results.

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.

Enter the name of the Pool / Subnet, and select the assignment criteria:

For **Smart Assign**: select IPv4 / IPv6, Mask, then optional Region, Resource assignment, Domain, and VLAN criteria, add free lines if desired, then click "Add Pool".

For **Direct Assign**: select IPv4 / IPv6 and enter the block in CIDR format, add free lines if desired, then click "Add Pool".

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 (smart assign) or a specific IP address (direct assign).

Enter the name of the Hostname and MAC address, then select the assignment criteria:

For **Smart Assign**: select IPv4 / IPv6, then optional Region, Resource assignment, Domain, and VLAN criteria, add free lines if desired, then click "Add Pool".

For **Direct Assign**: select IPv4 / IPv6 and enter the block in CIDR format, add free lines if desired, then click "Add Pool".

Current Pushed Configuration

Expanding "Current Pushed Configuration" allows you to view the detailed text (read-only) of the last sucessfully pushed configuration.

Current Pushed Configuration does not reflect any un-pushed changes or failed pushes.

Saving/Pushing DHCP Server Configurations

It is recommended that you save your configuration after changes. Save your changes by clicking the "Save Configuration" button at the bottom of the DHCP Management Gadget.

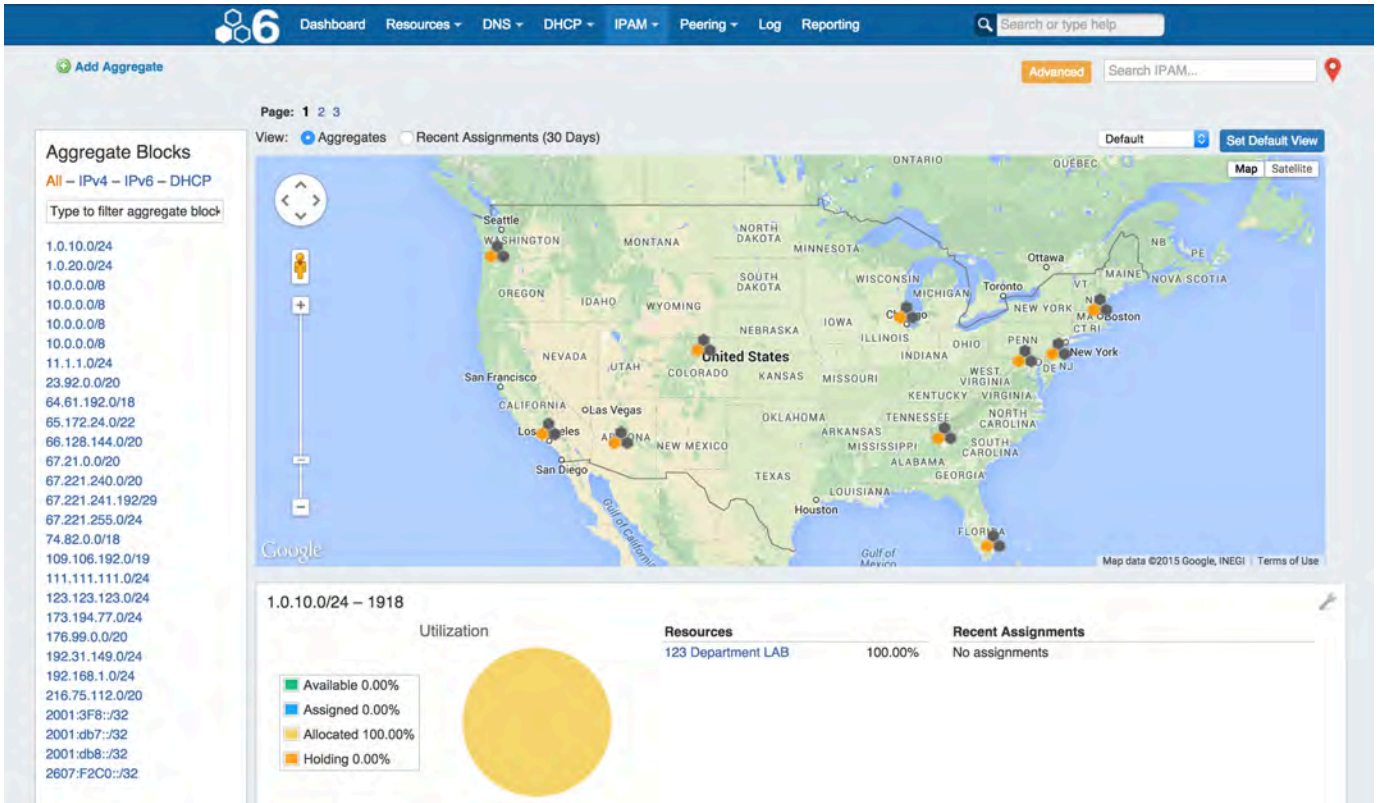
When you Push a Configuration (by clicking "Push Configuration") the configuration is automatically saved.

Permissions

DHCP Management integrates with ProVision's resource and permissions hierarchy, as well as the IP Management system. Individual DHCP servers can be assigned via Resource Permissions to different internal user groups, to be managed by only the appropriate parties.

IPAM Tab

IPAM

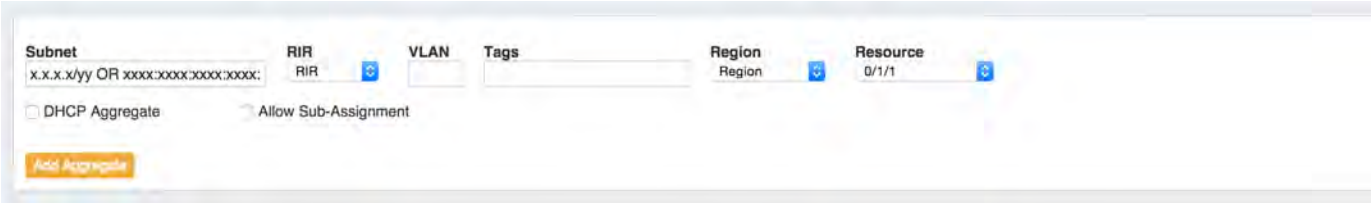


The **IPAM** tab provides a listing of aggregate blocks and tools to add and manage aggregates.

UI Overview:

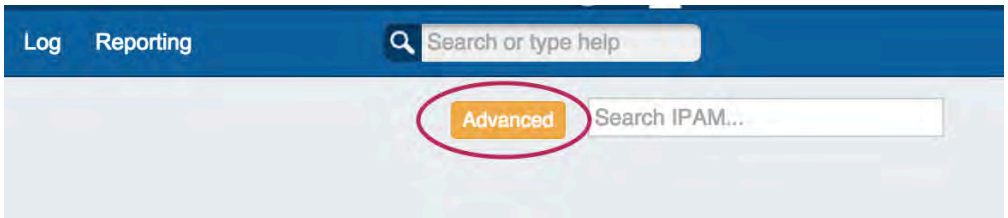
Add Aggregate:

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



"Advanced" Button":

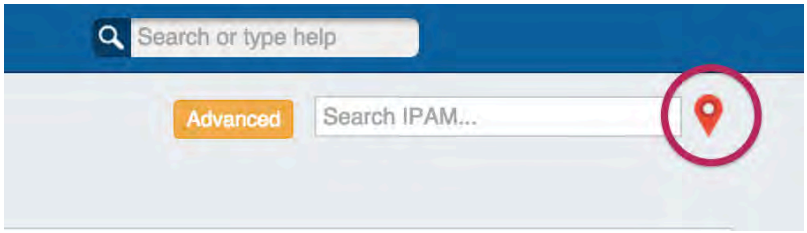
Opens the IPAM Manage screen for all blocks. See Working with IP Blocks - Architecting IP Address Blocks for more information on working in IPAM Manage.



Map View:

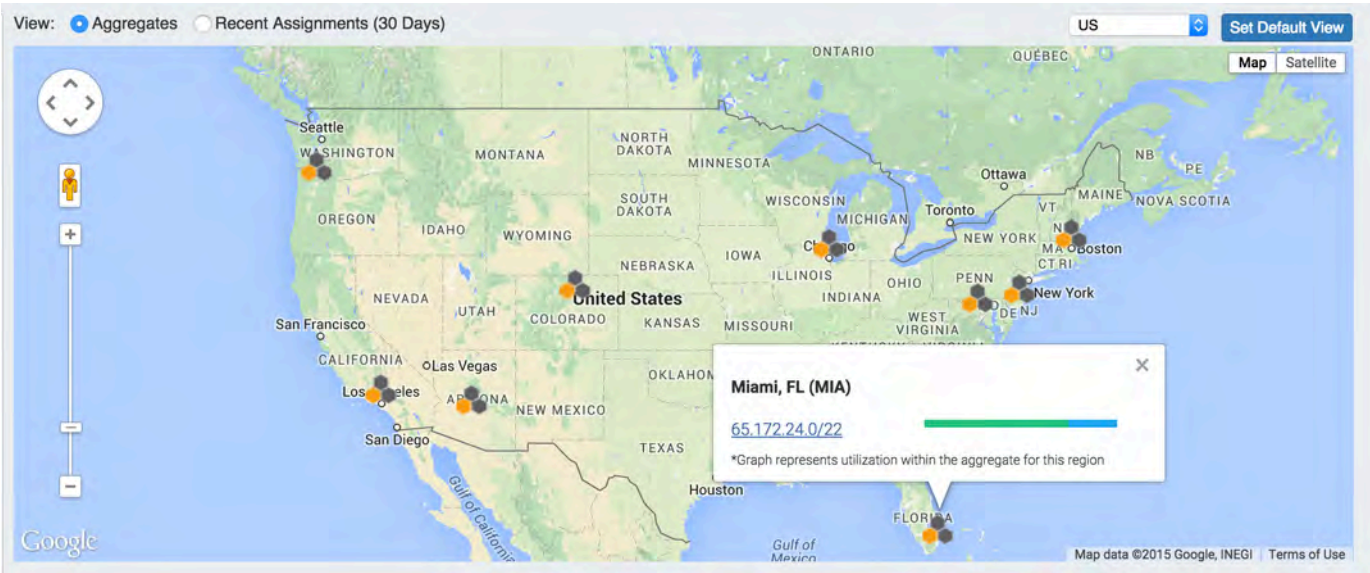
The IPAM aggregate map view may be enabled / disabled from the IPAM Configuration page. Once enabled, a map pin will be visible on the IPAM Tab next to the search box.

Click on the pin to show / hide the map.



Map View Overview:

Map View will show aggregate locations / recent assignments in geographical format, based on regions assigned to those aggregates. Clicking on a location icon will bring up a current utilization graph and a link to that aggregate's IPAM Manage page if in Aggregates view, or assignment details if in Recent Assignments view. A default map view may be set by either selecting a predefined map area from the dropdown (US, US & Europe, South America, All), and clicking on "Set Default View", or a custom area set by zooming / navigating to the desired map area and clicking "Set Default View".



View: Select "Aggregates" to view all available aggregates with region data, or "Recent Assignments" made within the past 30 days.

View Range Dropdown: Select a predefined area for the map view range: US, US & Europe, South America, or All.

Set Default View: Click to save the current map view range as the default view.

Aggregate Detail Box: Provides utilization data and a shortcut link to the IPAM Manage screen for that aggregate if in "Aggregates" view, or assignment details if in "Recent Assignments" view.

> Regions must be enabled, the region address field populated, and regions assigned to aggregates / blocks for the IPAM Map view to populate data. See Edit Regions for additional detail.

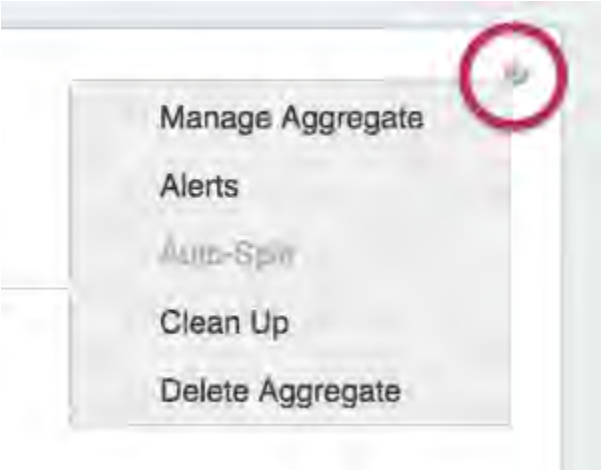
Aggregate Blocks List:

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, the top five Resources assigned under that aggregate, and recent assignments.

Top Level Aggregate Action Menu (wrench icon):



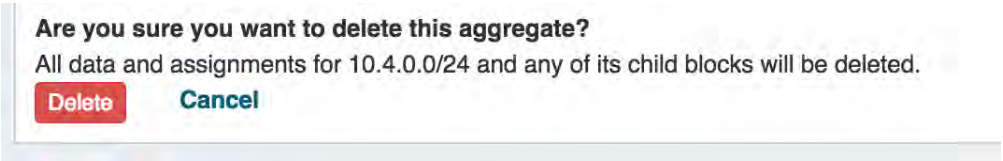
"Manage Aggregate" opens the IPAM Manage screen for blocks under that aggregate. See Working with IP Blocks - Architecting IP Address Blocks for more information on working in IPAM Manage.

"Alerts" opens the "Manage Alerts" dialog, where you can set up an email alert to send when available space reaches less than a certain percentage. An IPAM Alert task must be set up in the Scheduler detailing the alert schedule for the alert to send.

"Auto-Split" opens the template to split the aggregate into specific sized blocks.

"Clean Up" opens the template to merge the aggregate into specific sized blocks.

"Delete Aggregate" The provides the option to delete the aggregate.



Working with IP Blocks

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

Additional Information

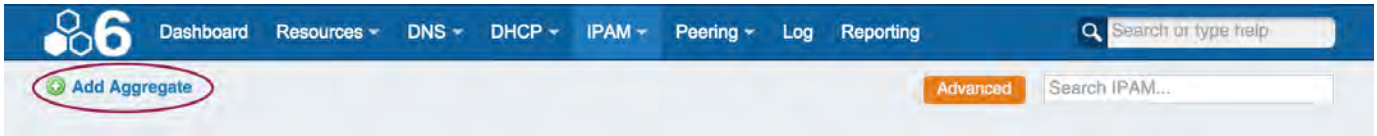
- Working with IP Blocks

Working with IP Blocks

Working with IP Blocks

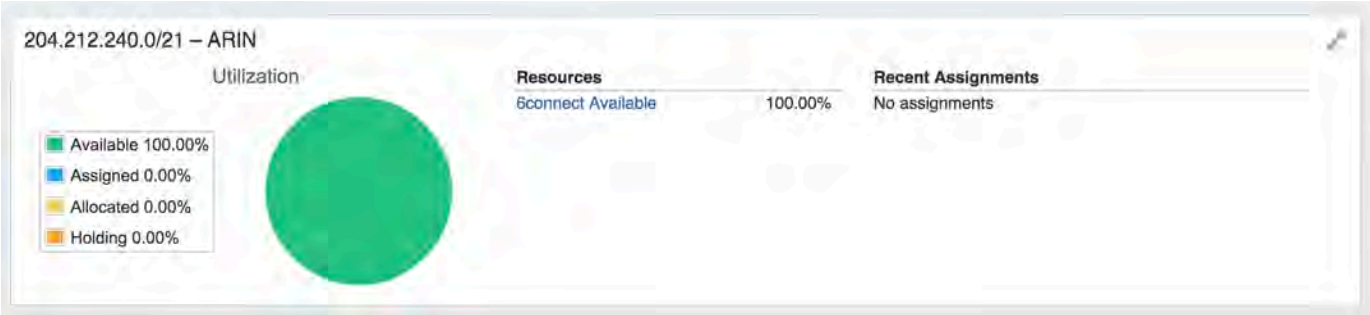
Adding/Deleting IP Address Aggregates

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

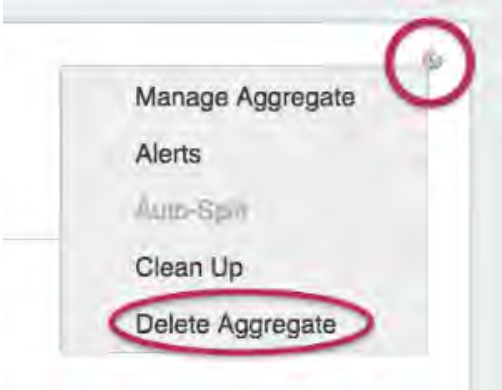


Once clicked, you get a more detailed screen to add an aggregate block.

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



To delete the aggregate - click on the Action Menu (wrench icon) for the aggregate in the top right corner of the box, and you will have the option delete the aggregate.



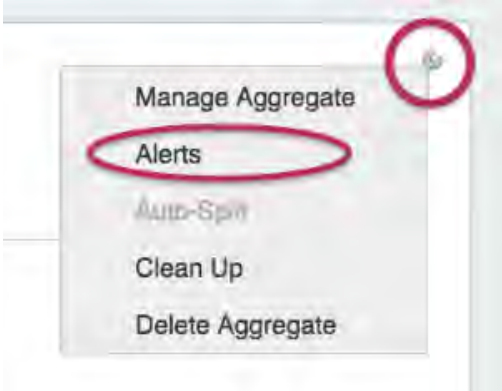
After selecting the "Delete Aggregate", a message will show up in the aggregate box verifying that you would like to delete. Deleting will remove all data and existing assignments from the aggregate and its child blocks. If certain you want to delete, click the "Delete" button. Otherwise, select "Cancel".

Set Aggregate Alerts

Aggregate alerts allows you to set up an email notification to send once an aggregate's available space hits a chosen percentage threshold.

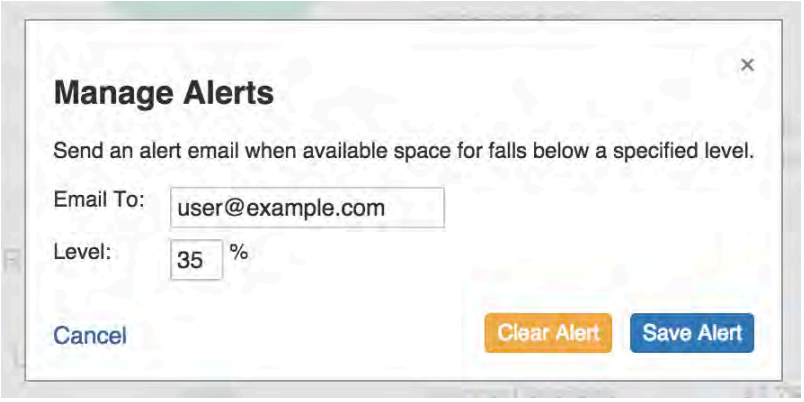
> Prior to setting the alert, verify that an IPAM Alerts task has been created in the Scheduler, detailing the alert schedule / frequency. Alerts will not send unless the scheduled task has been created to set the desired alert frequency.

To set the alert, select the Action Menu (wrench icon) for an aggregate, and click on "Alerts".



This brings up the "Manage Alerts" dialog. Enter the email address for the alert recipient and the available space percentage at which the alert is triggered.

When done, hit "Save Alert". You may also cancel at any time.

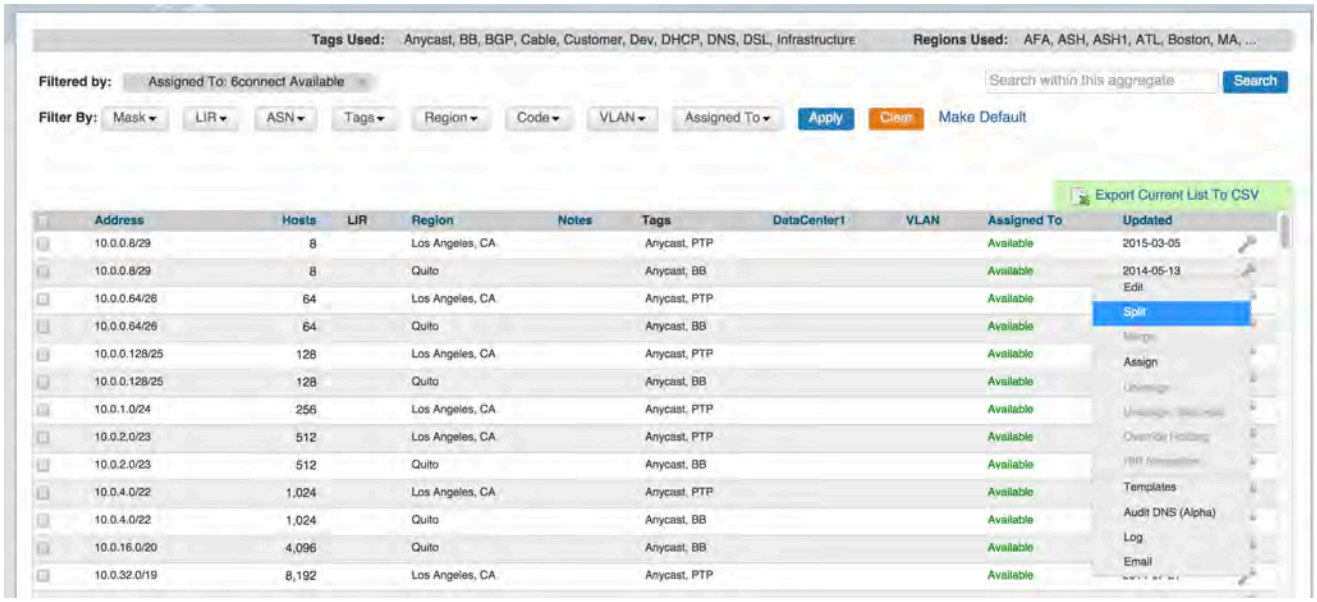


To clear an existing alert, hit "Clear Alert", and the alert information will be removed and status saved.

Architecting IP Address Blocks

Splitting/Merging blocks manually

To split a block manually - While in the IPAM Mange screen, click on the Action Menu (wrench icon) for the block you wish to modify. Then select the "Split" function. To aggregate blocks, select "Merge" from the same menu.



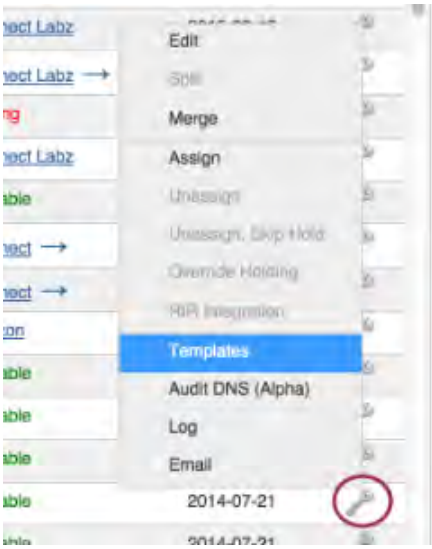
Splitting/Aggregating blocks with Templates

Templates for a block are available under the Action Menu (wrench icon) for that block. There are two templates available: Cleanup, which auto-aggregates the block, and Auto-split.

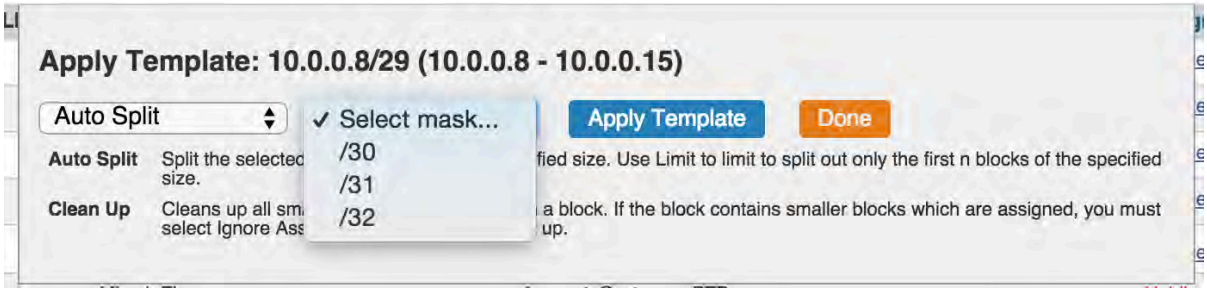
Select the desired template from the action menu.

Select the Split / Clean Up size desired, and hit "Apply".

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: 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.

Domain: The VLAN Domain to associate with the block.

VLAN: VLAN information for the block, must have domain selected to view available VLANs.

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 mutil-level situations (Parent -> Child -> Grandchild), VLAN should be updated at the top tier parent level.

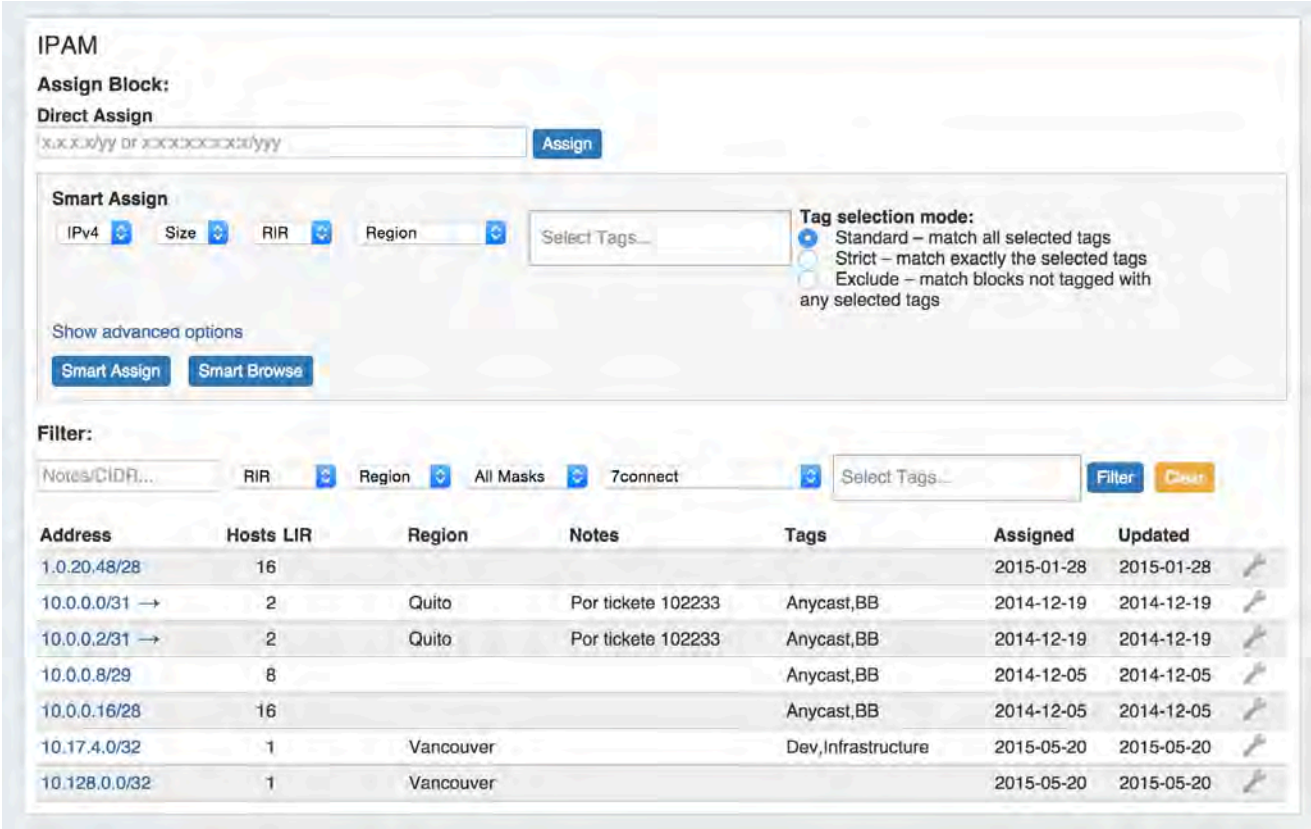
After editing the desired attributes for the block, simply hit "Save".

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 Direct Assign, Smart Browse, and Smart Assign with advanced options, and is the primary tool for space assignment.

Assigning Space from the IPAM Gadget

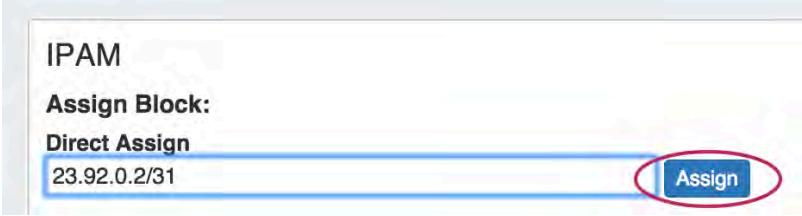
The IPAM Gadget is accessed from a Resource Entry page, once enabled for the Section (to add Gadgets, see Customizing Sections and Add Gadgets to your Section).



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

Direct Assign

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



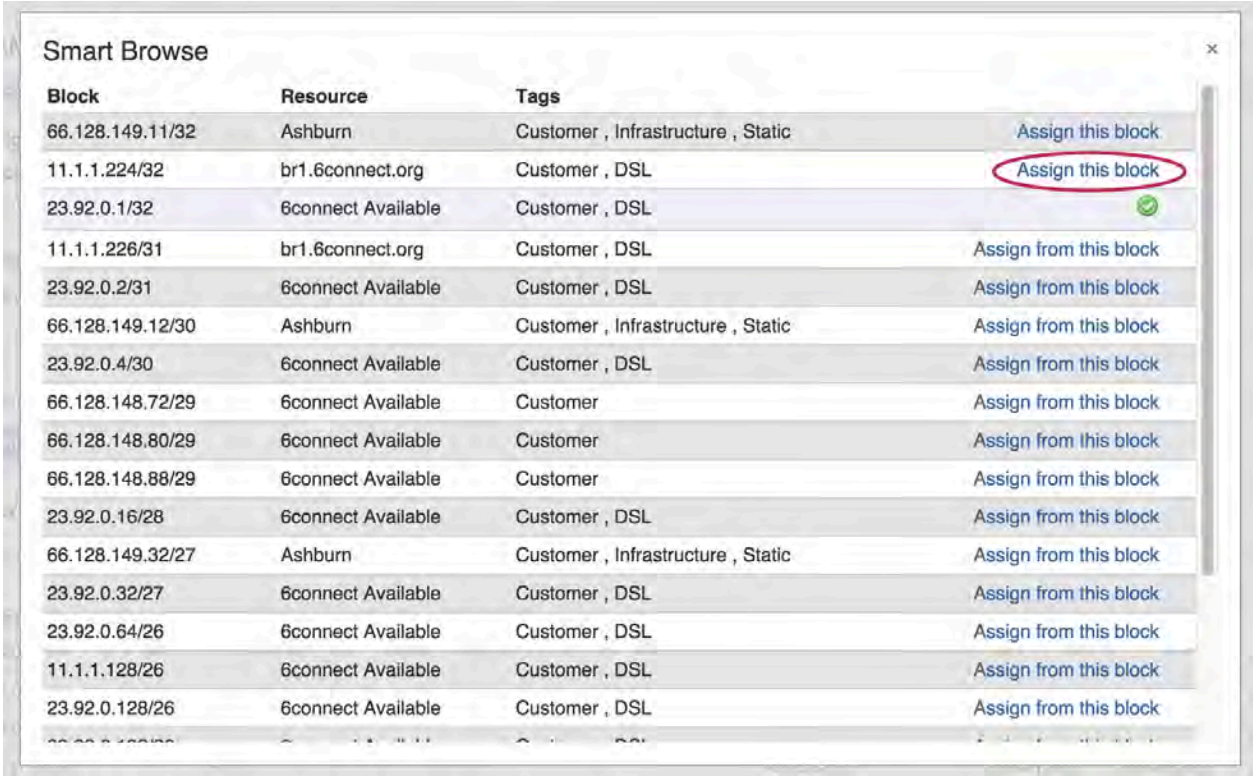
Note

The default maximum function nesting level is '100'. If you experience issues resulting from recursion limits and require a different maximum, the local admin can increase the recursion limit setting in php.ini. Refer to http://xdebug.org/docs/all_settings#max_nesting_level.

Smart Browse

Smart Browse utilizes the smart assign parameters as search filter criteria. Under the "Smart Assign" area, select the IPv4/IPv6, Size, RIR, Region, and/or Tags that you wish to filter the available blocks list.

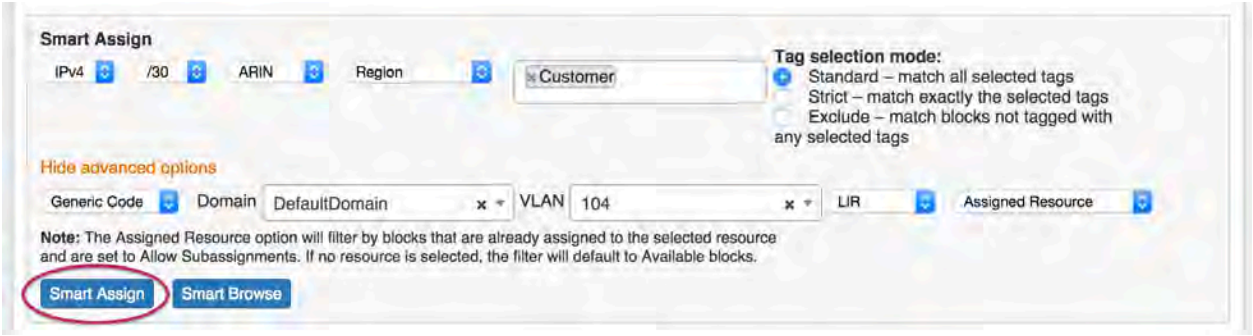
Click on "Smart Browse" to bring up a list of IP aggregates meeting that criteria, which you can select the block(s) to assign. A green check will show next to the block once assigned. You may assign multiple blocks per browse session.



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.

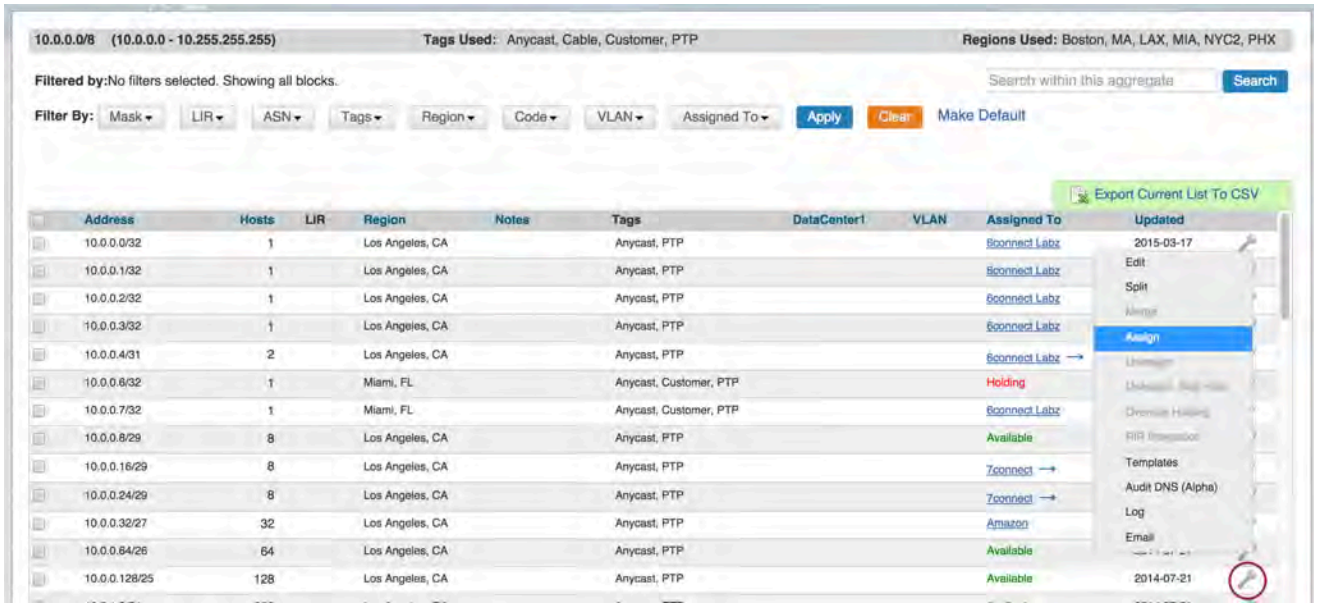
Additional advanced Smart Assign options are available under "Advanced Options", including VLAN and LIR.



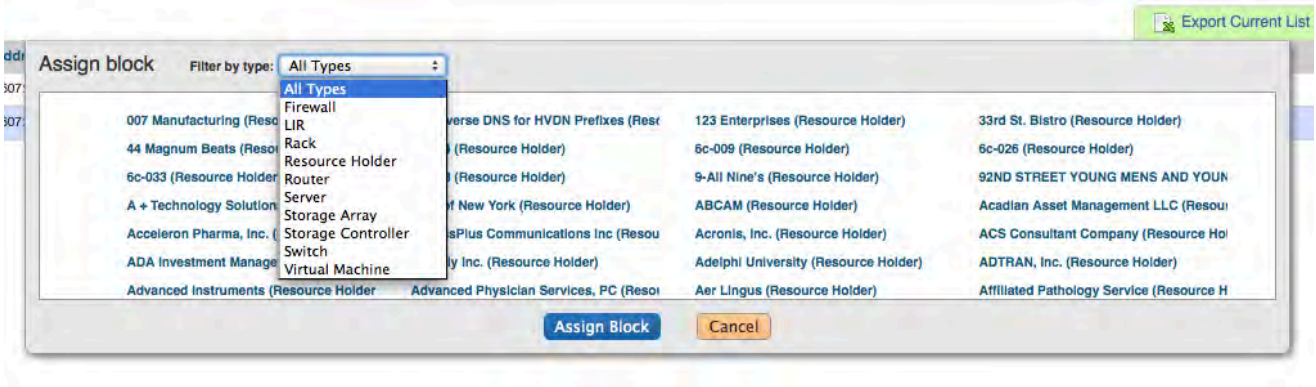
Once your criteria has been set, click the "Smart Assign" button.

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".



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



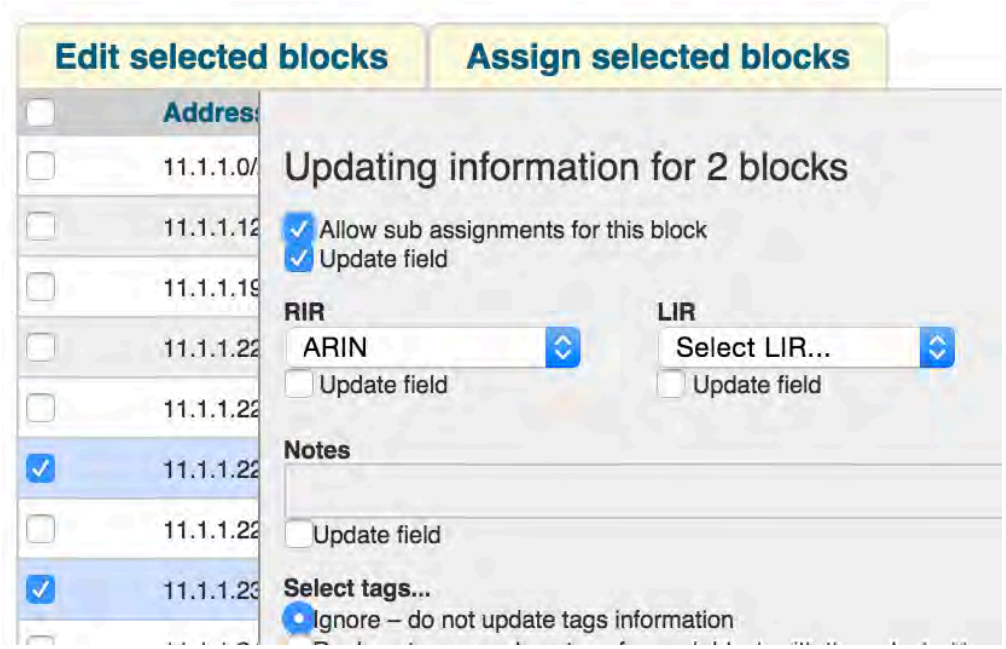
After assigning, you can further edit the block attributes or subassign space.

> **'Reserving' IP Space**
> If you need to create a 'reserved' pool from which to assign blocks, you can achieve this by creating a "Reserved" Section. 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.
>
> For more details, see the FAQ entry: "How do I 'reserve' IP 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 subsidaries, or locations.

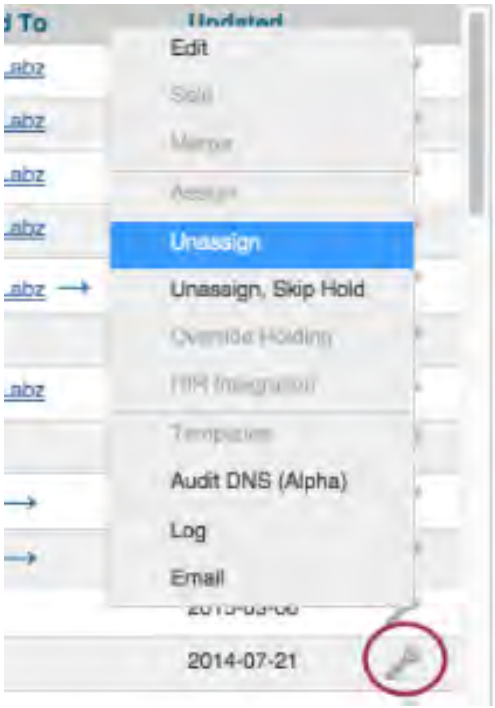
To allow sub assignments for multiple blocks at once, open the Manage screen for the aggregate. Then, select the desired blocks and click "Edit Selected Blocks". The Multi-block edit interface will open. In that interface, select the check boxes next to "Allow sub assignments for this block" and the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



Unassigning IP Space

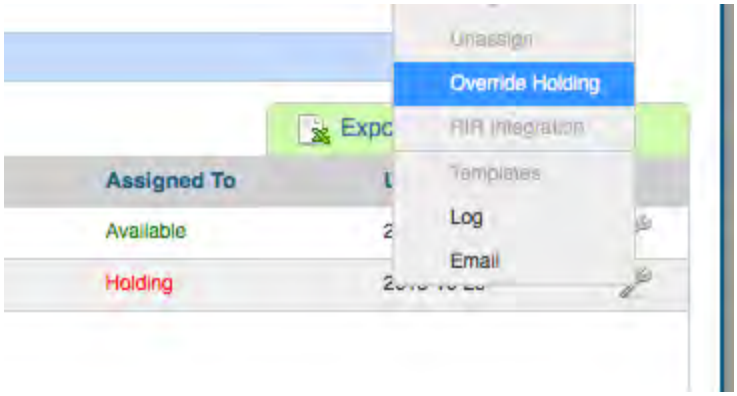
When a block is assigned, you will have the option of unassigning the block from the resource and returning it to the Holding Tank.

To unassign the block, simply click on the Action Menu (wrench icon) for the block and select "Unassign". You may also chose "Unassign, Skip Hold", which unassigns the block and immediately returns it to available, bypassing the holding tank. After unassigning blocks / overriding holding, newly available blocks will be merged upon next page refresh.

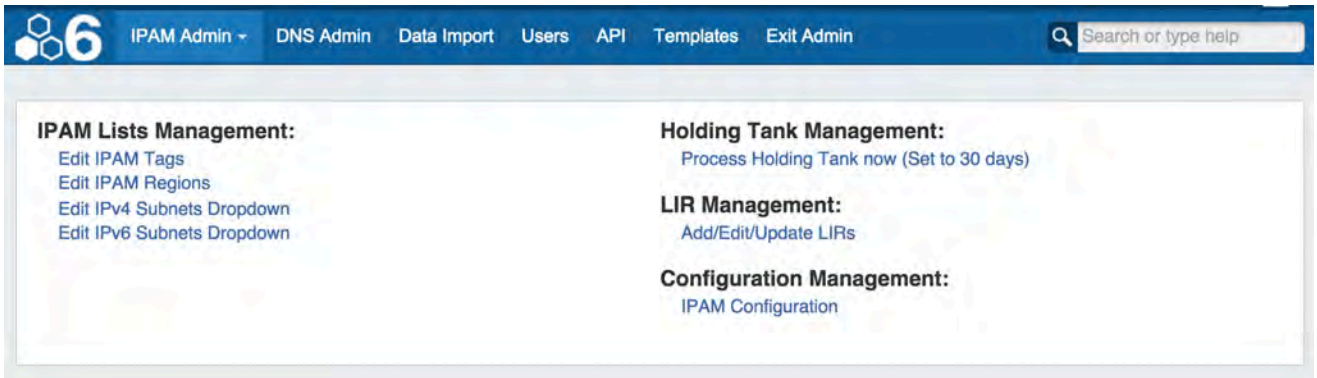


To return IP space in the Holding Tank to the Available Pool - there are two methods:

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).

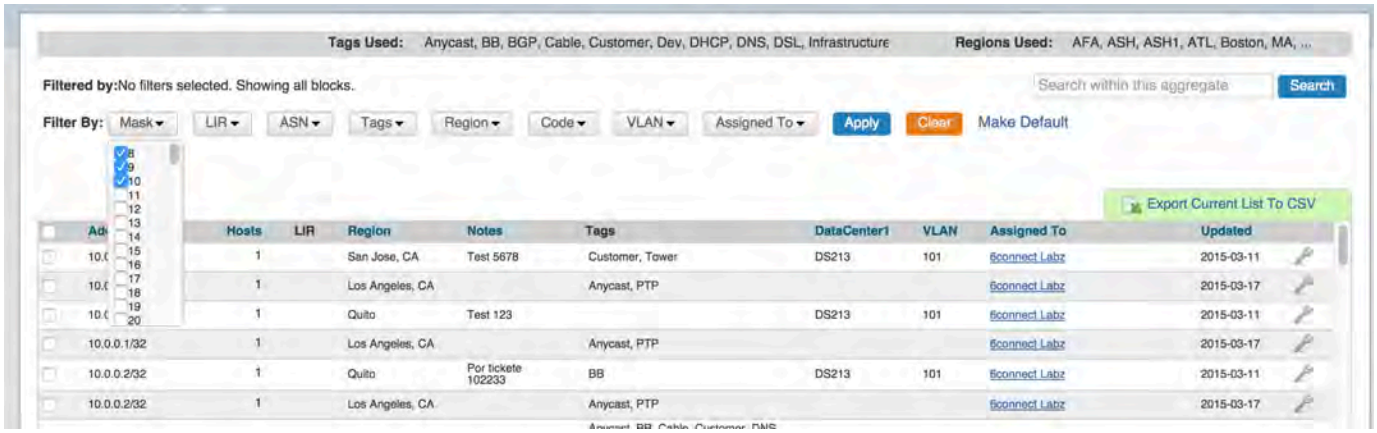


For more information on the Holding Tank, see Holding Tank Management.

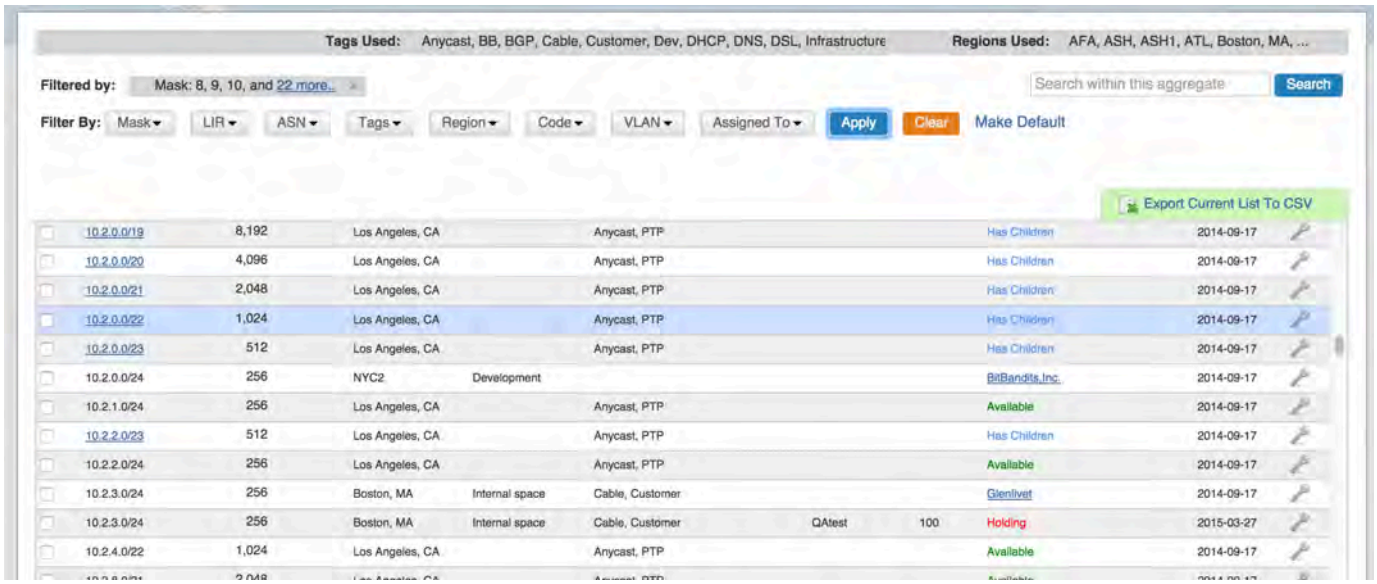
View Parent Blocks in IPAM Manage

On the IPAM Manage screen, you have an option to Filter the view by selected Subnet Mask (dropdown). By default, blocks in ProVision are displayed at their child size, the lowest point in the block tree available for that block.

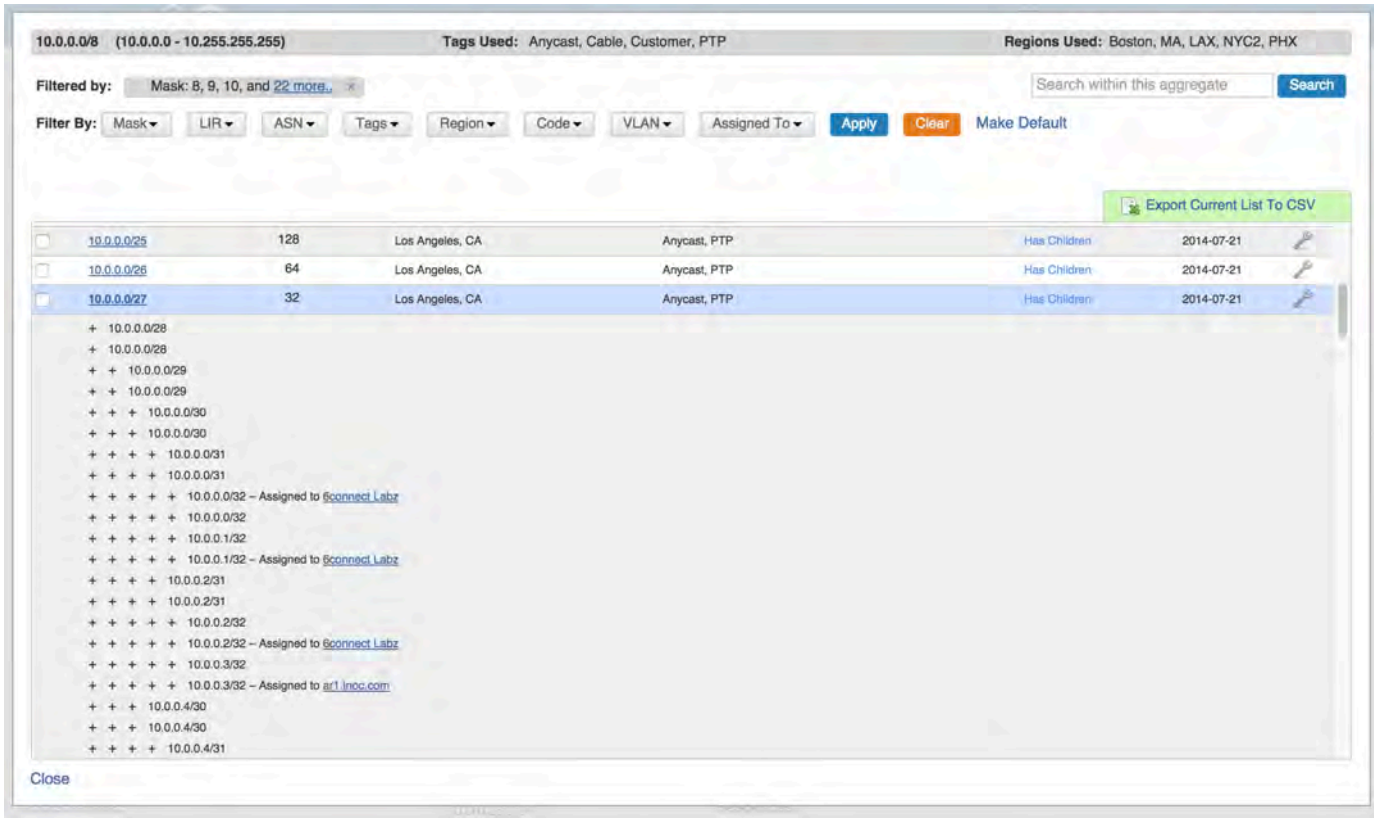
With the Filter By Masks enabled, and multiple or all masks selected, the view changes to include the "Parent" blocks, showing all masks throughout the assignment tree.



A parent block with children will show "Has Children" in the column where assignment status is shown, and the block CIDR will show as a clickable link.



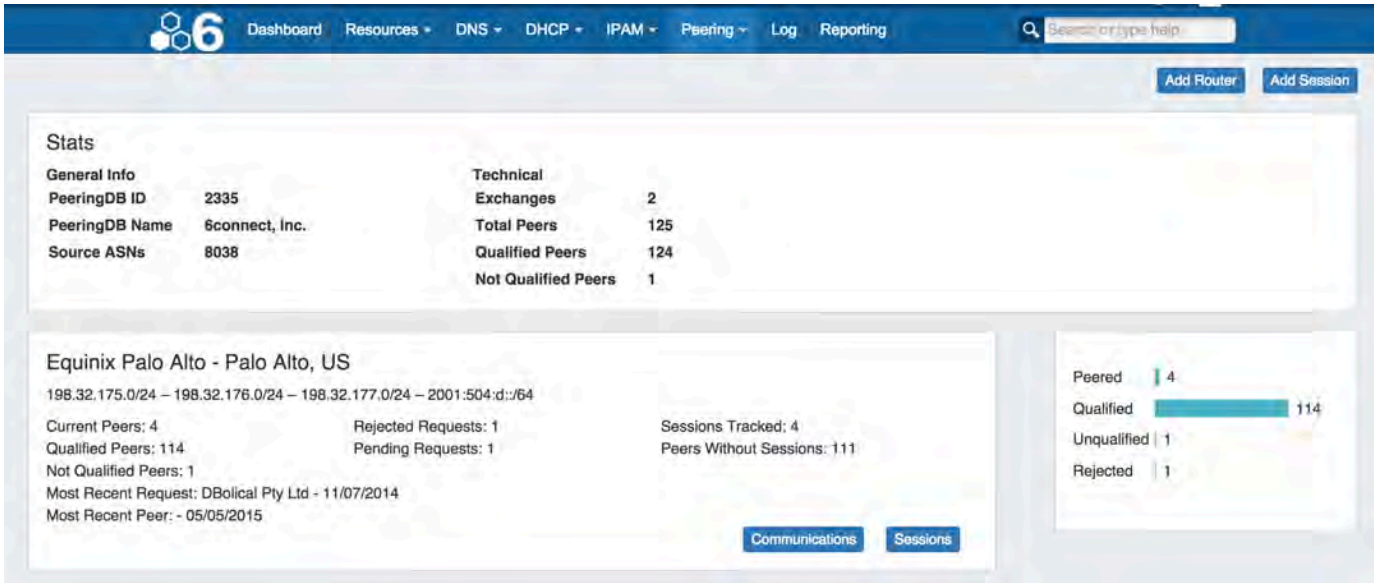
Then, the user can click on the block link once to view the additional assignments/allocations underneath it.



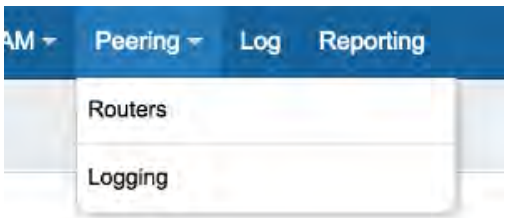
Peering v2

6connect Peering

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



Two other sections are available via the drop down menu:



Routers - Links to the resource list of routers

Logging - View peering related logs

> **Peering Setup - Local Installations**
> If you are using a locally hosted instance of ProVision, verify that you have followed the instructions on the Local Installations - Peering Setup page to set up Peering for your instance.

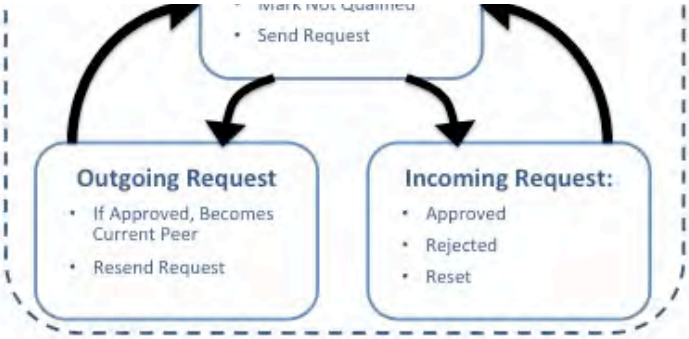
Table of contents

Peering - Common Tasks

Peering Workflow

The ProVision peering workflow encompasses working with routers, peer groups, sessions, and managing communications.

Click here to see the graphical workflow...



Below is an outline of the steps needed for a basic workflow process.

- Peering Workflow
    - 1) Add Router(s)
    - 2) Add Peer Groups
        - With a New Router:
        - With an Existing Router:
    - 3) Add Session(s)
    - 4) Configure Sessions
    - 5) Manage Communications
        - New Peer:
        - Outgoing Peer Request:
        - Incoming Peer Request:
    - 6) Repeat for new Routers, Sessions, and Peers.
- Peering Common Tasks

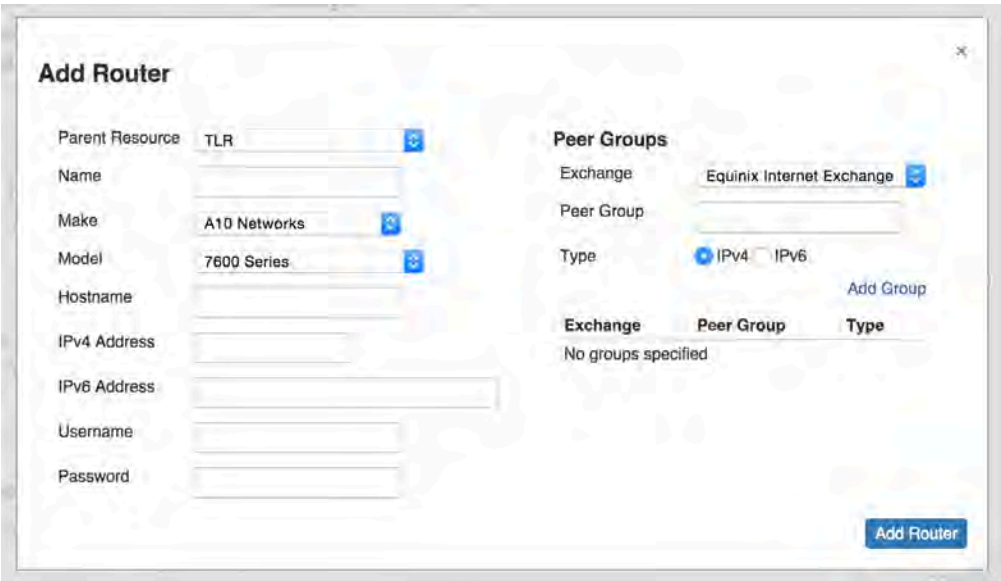
1) Add Router(s)

Associated Task(s): Add Routers

Adding a router is the fundamental base and first step of working with peering in ProVision. Routers may be added through either the "Add Router" button in the **Peering** Tab, or through creating a new entry through the Resources system under the router Section.

The recommended method is to start with the "Add Router" button in the **Peering** Tab.

This dialog will also allow you to add a Peer Group at the same time, merging steps one and two together.



For a detailed breakdown of this task, see Add Routers.

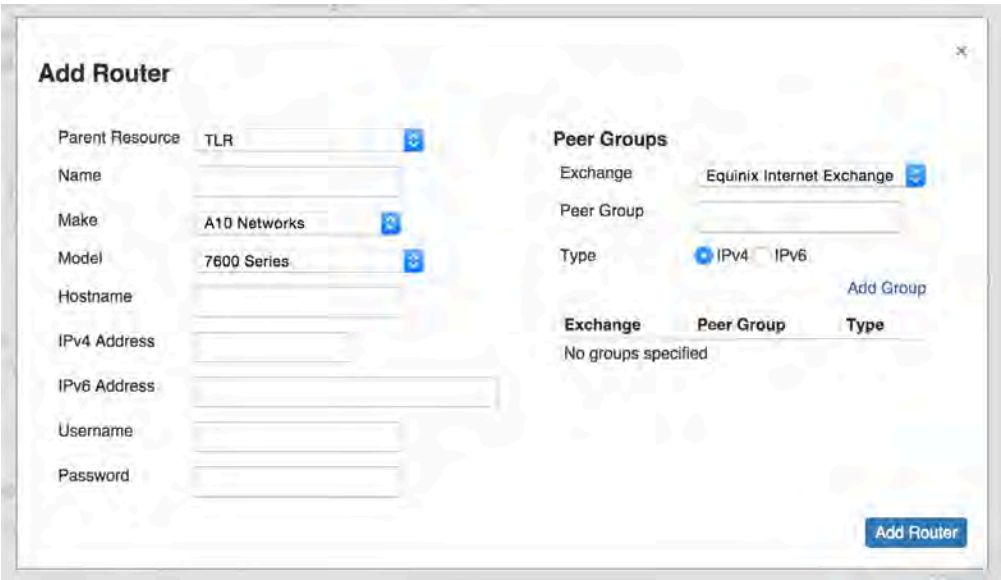
2) Add Peer Groups

Associated Task(s): Add Routers, Gadgets

Associating the router with a peer group is necessary to link the router to a particular exchange. You may add the Peer Group information either in the "Add Router" dialog or in the Peer Group Gadget prior to adding sessions.

With a New Router:

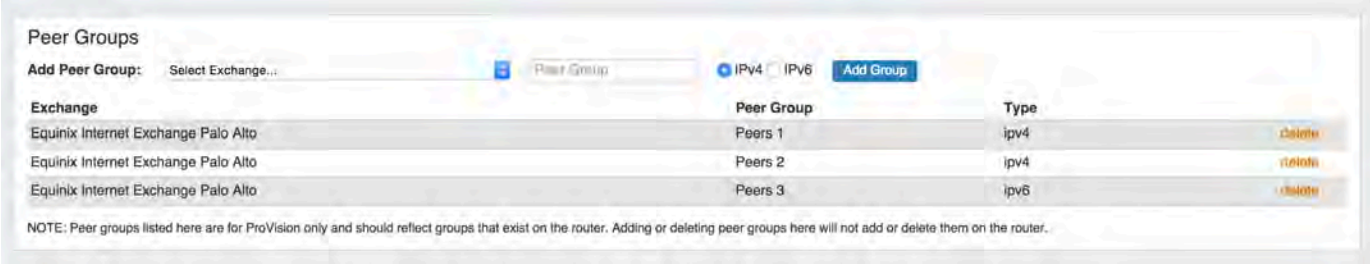
To add a peer group through the "Add Router" dialog: After inputting the Router information, under Peer Groups, select the exchange, type in the name of the desired Peer Group name, select whether it is IPv4 / IPv6, and click "Add Group". Lastly, click "Add Router".



With an Existing Router:

If you need to add a Peer Group to an existing router, you may use the Peer Group gadget.

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.



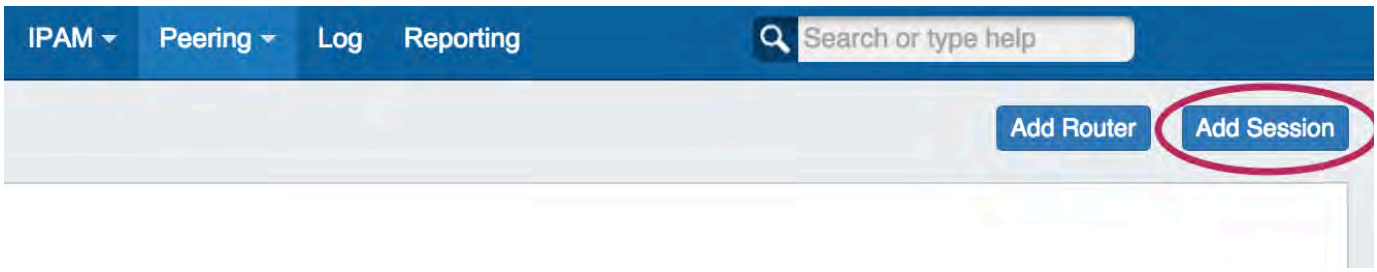
To do this, simply select the exchange, type in a Peer Group name in the text box, select IPv4 or IPv6, the click "Add Group".

Peer Groups added from this gadget will be then be available to select in the "Add Session" dialog box in the **Peering** tab.

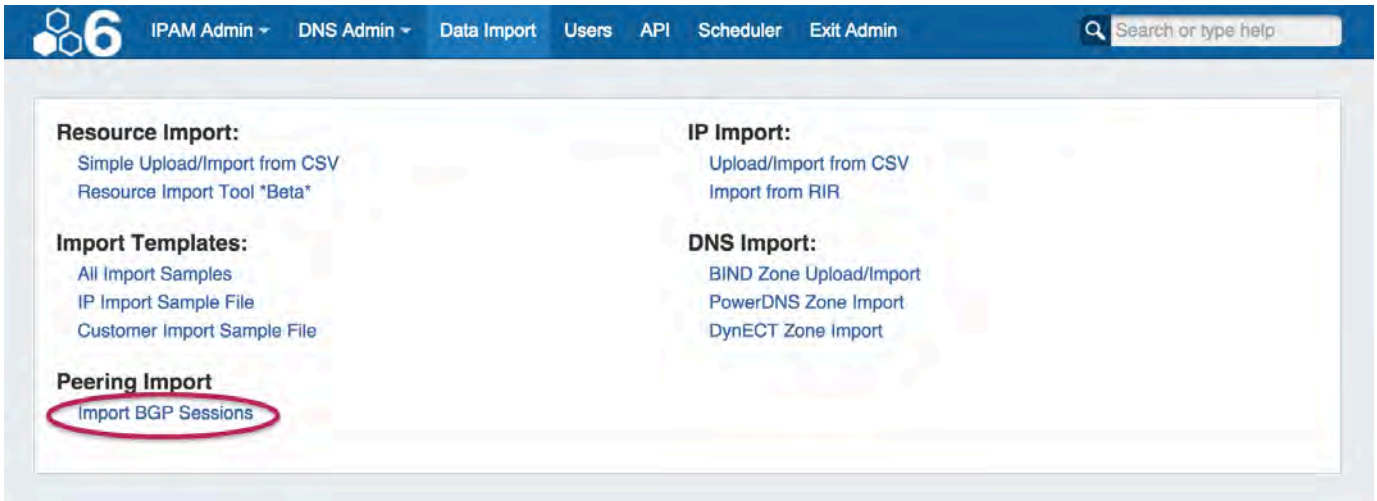
3) Add Session(s)

Associated Task(s): Add Sessions, Import Sessions

After Routers and Peer Groups have been set up, the next step is Adding Sessions to ProVision.



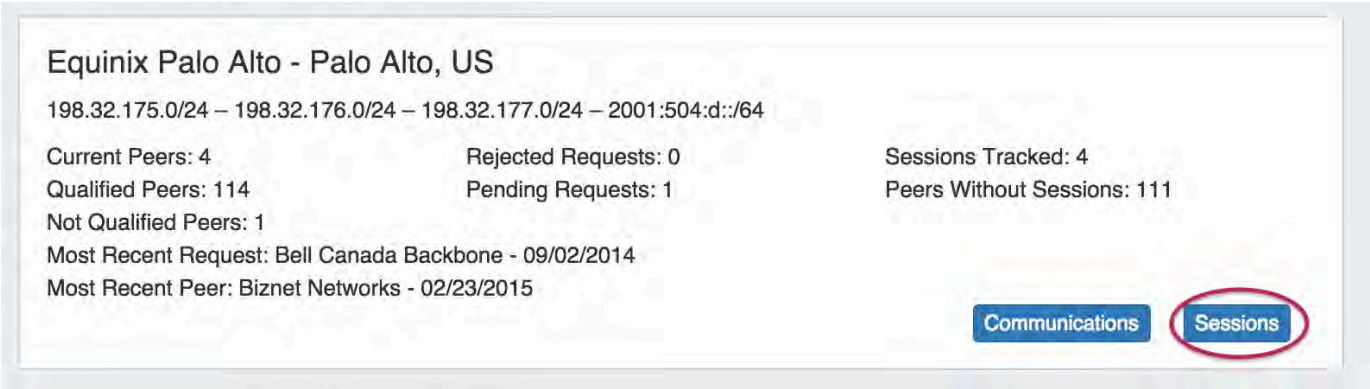
You may add a new session through the "Add Session" button in the **Peering** tab, or you may Import Sessions from an existing router. Importing sessions requires Admin level permissions, and is accessed through the **Data Import** Tab in the Admin section of ProVision.



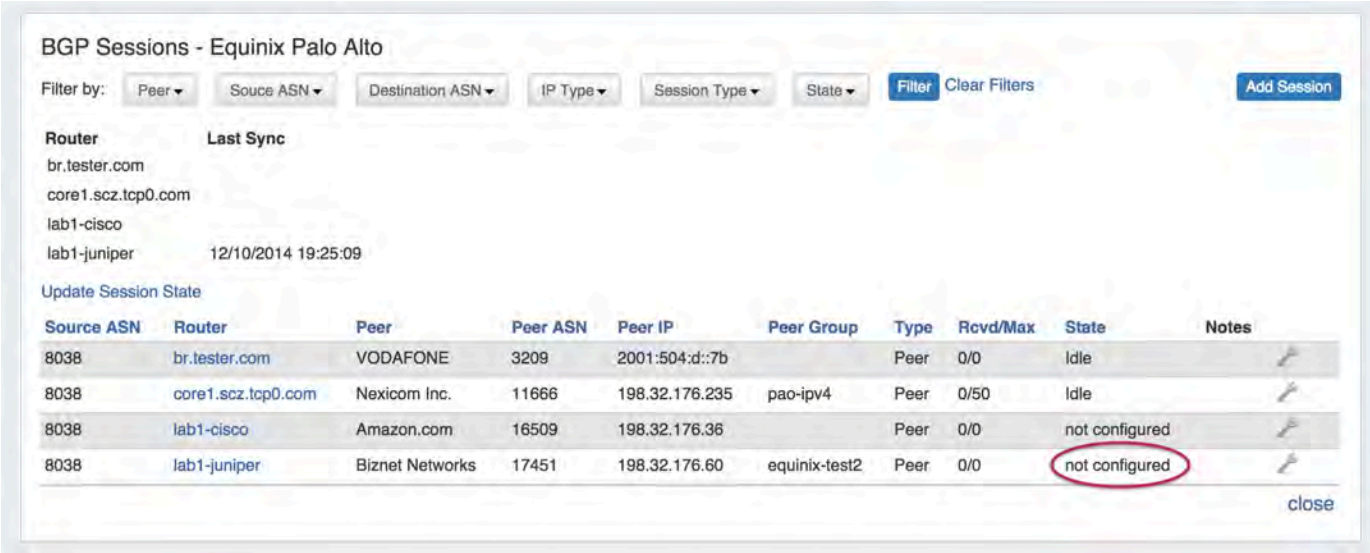
4) Configure Sessions

Associated Task(s): Managing Peer Sessions

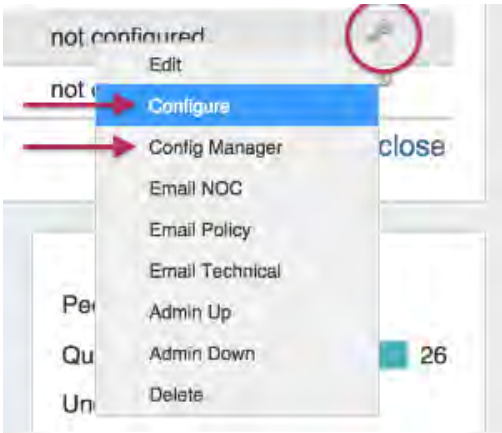
Once a session has been added, you will see it in the session list from the Managing Peer Sessions module. Open the Sessions manager by clicking on the "Sessions" button for the exchange that hosts your session.



For a newly added session, the status will show as "Not Configured".



You may configure the session through 1 click config or the Config Manager, accessed by clicking on the Action Menu (wrench icon) for the session. 1 click config uses the default config settings, while the Config Manager allows you to manually create one time use unique settings.

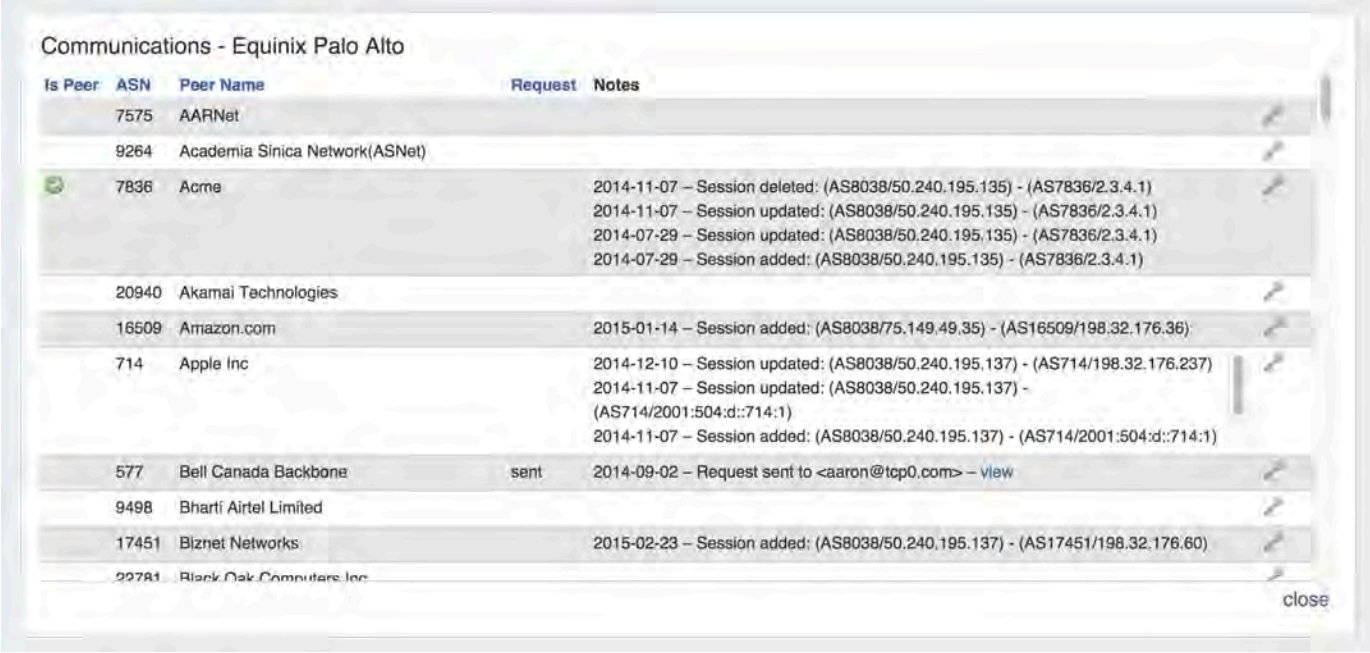


5) Manage Communications

Associated Task(s): Managing Peer Communications

Once a session has been added with a peer, you will see the peer listed the communications list. Open the Peer Communications Manager by clicking on the "Communications" button for the exchange that hosts your session.

You will see the list of Peers, their ASNs, Name, Request Status, Notes (log), and the Action Menu (wrench icon).



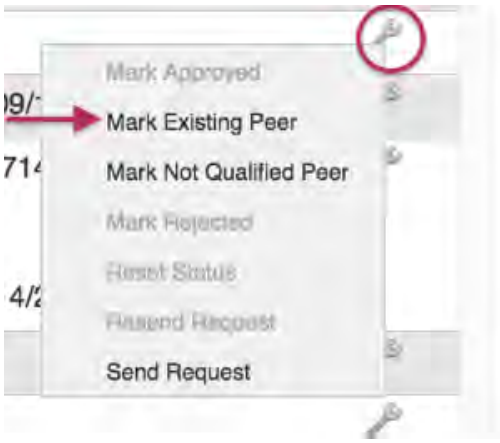
From here, you may manage peer requests and mark peer status through the Action Menu (wrench icon).

From here the flow depends on the Peer Status and whether a request is Outgoing or Incoming.

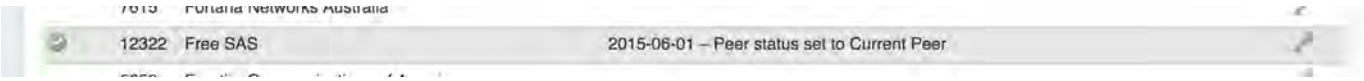
New Peer:

For an existing peer that was just added, you can mark the peer as:

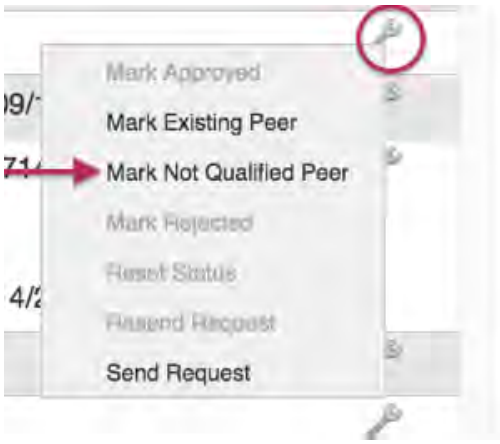
a) Mark Existing Peer



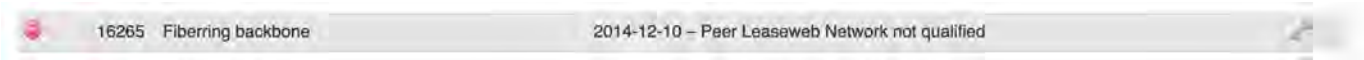
This places a green check icon next to the peer in the communications list (seen below) and removes email request options.



or b) Mark Not Qualified



This places a red icon next to the peer in the communications list and notes as not qualifed



You may also choose to send an outgoing peer request.

Outgoing Peer Request:

To send out a peer request, select "Send Request" from the Action Menu (wrench icon) . This sends an initial peering request email to the peering coordinator for that peer. 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.

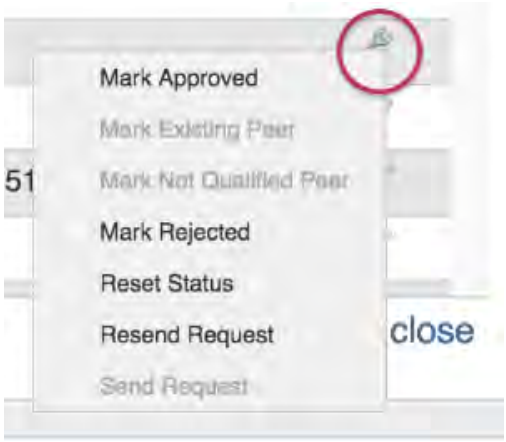
Once a Request has been sent out, it can proceed two ways:

a) The request is accepted and Peering is established (becomes current peer)

b) The request is not accepted / responded to, and the "Resend Request" option becomes available, allowing you to repeat the request.

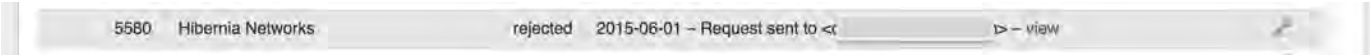
Incoming Peer Request:

If a request is received, you have a few options from which you may mark the peer from the Action Menu:



a) If you accept the request, you may "Mark Approved" and Peering is established (becomes current peer).

b) You may "Mark Rejected", in which case a peer status is marked rejected.



c) "Resend Request" may be available to resend a request

d) "Reset Status" is available at various points in the communications process, if you want to reset the peer back to the beginning state and re-establish a different status condition. This reopens the initial options from which you may select a different peer mark.

6) Repeat for new Routers, Sessions, and Peers.

To add additional Routers, Sessions, and Peers repeat steps 1, 2, or 3, and manage your newly added sessions and peer communications similarly with steps 4 and 5.

Peering Common Tasks

Some of the commonly performed peering tasks are listed below. Click on the links for additional information.

- Add Routers
- Add Sessions

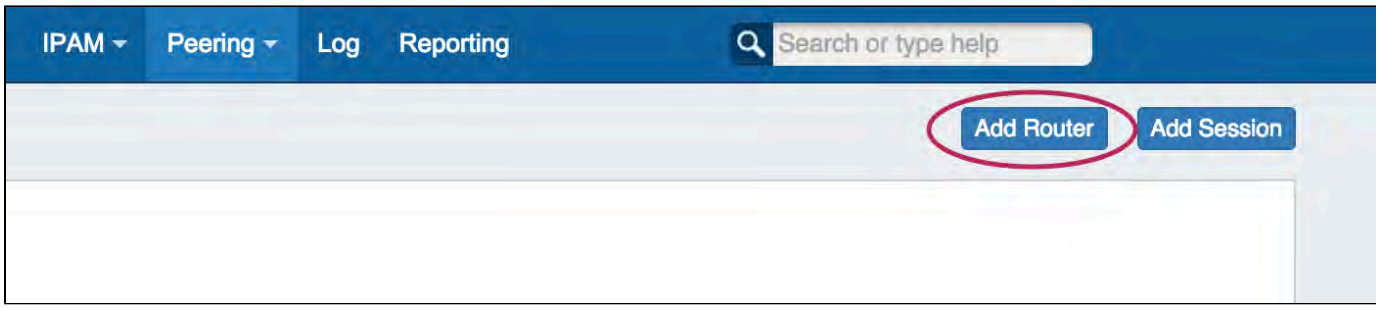
- Import Sessions
- Managing Peer Sessions
- Managing Peer Communications

Add Routers

Adding Routers

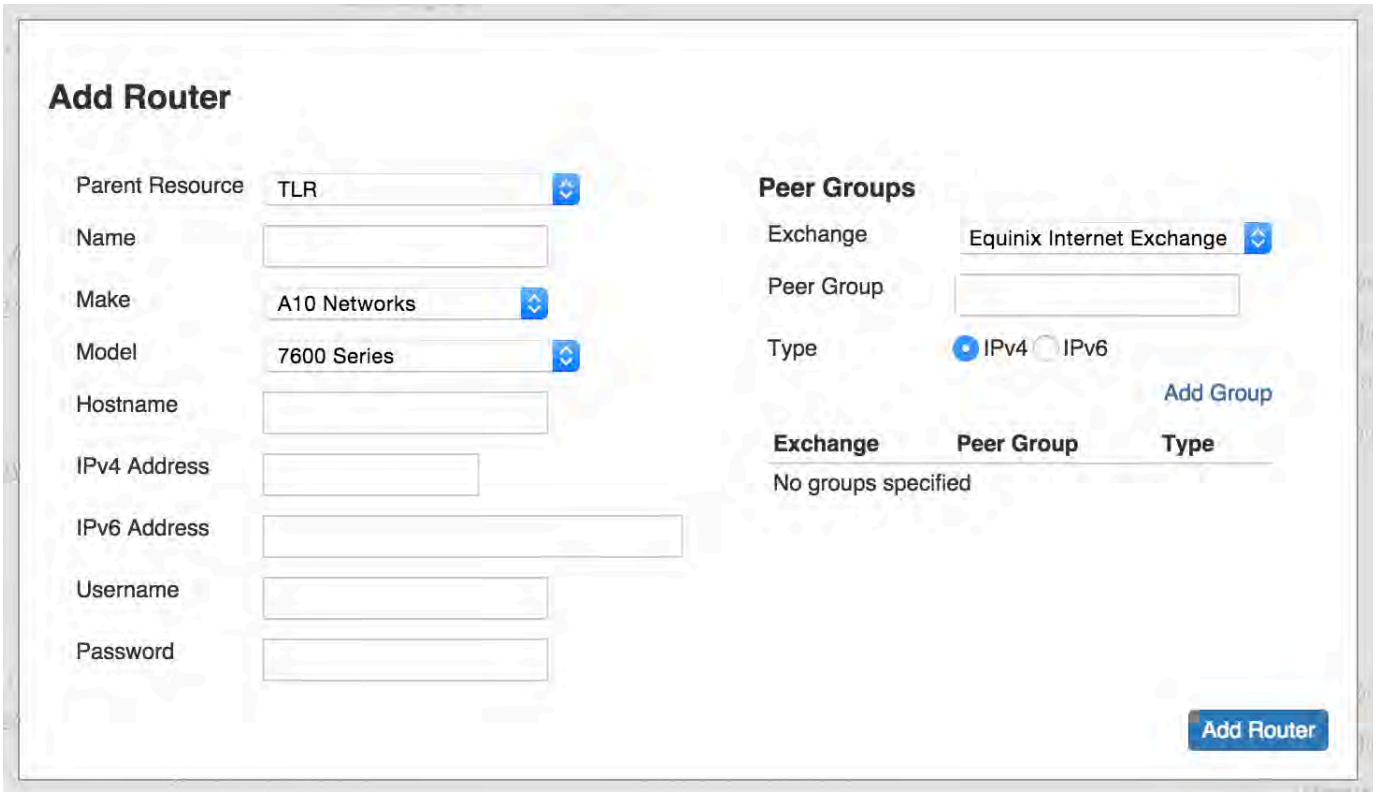
Add a Router

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



The Add Router dialog will pop up. 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".

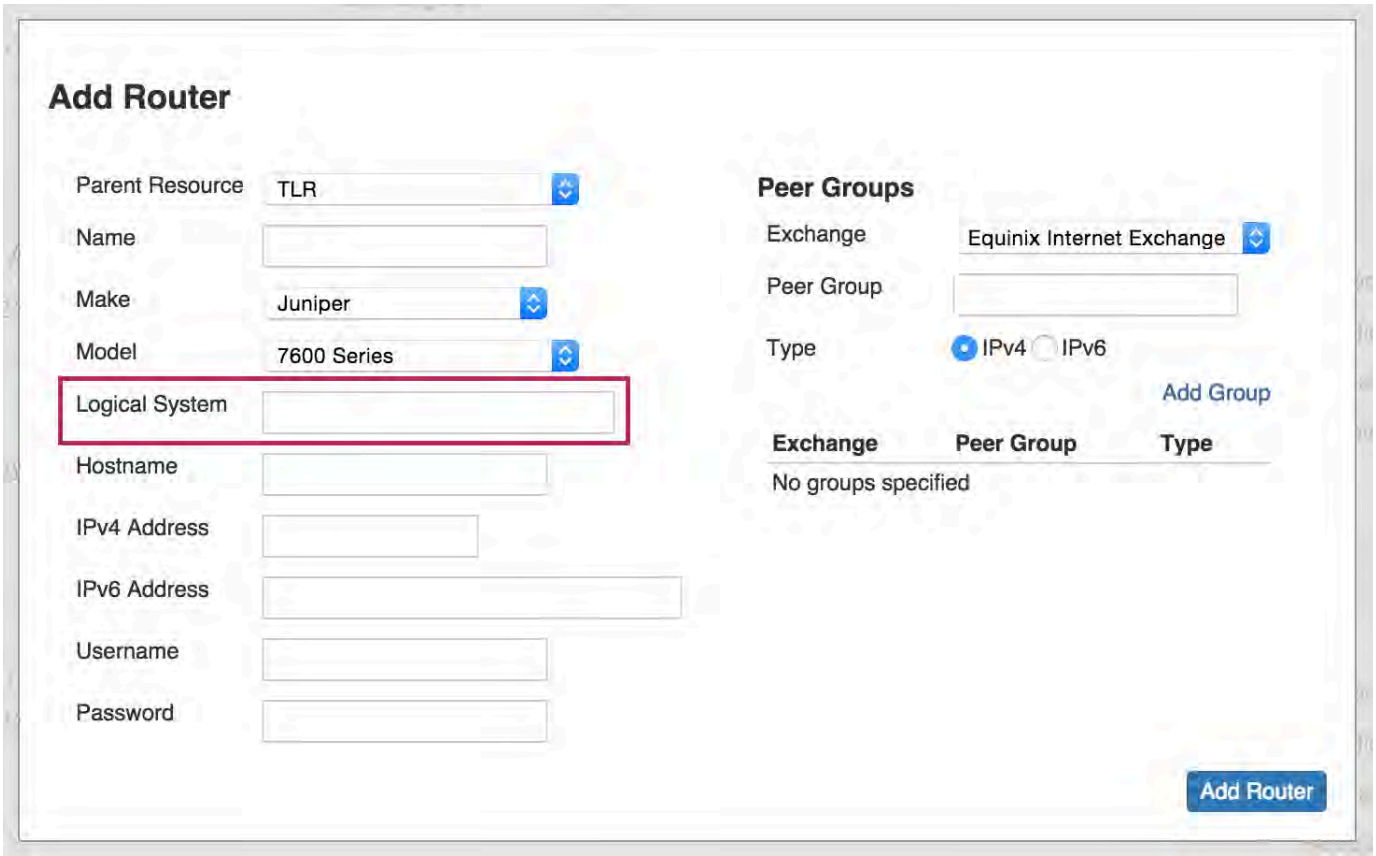


> Associating the router with a peer group is necessary to link the router to a particular exchange.
>
> Please be sure to add the Peer Group information either in the "Add Router" dialog or in the Peer Group Gadget prior to adding sessions.

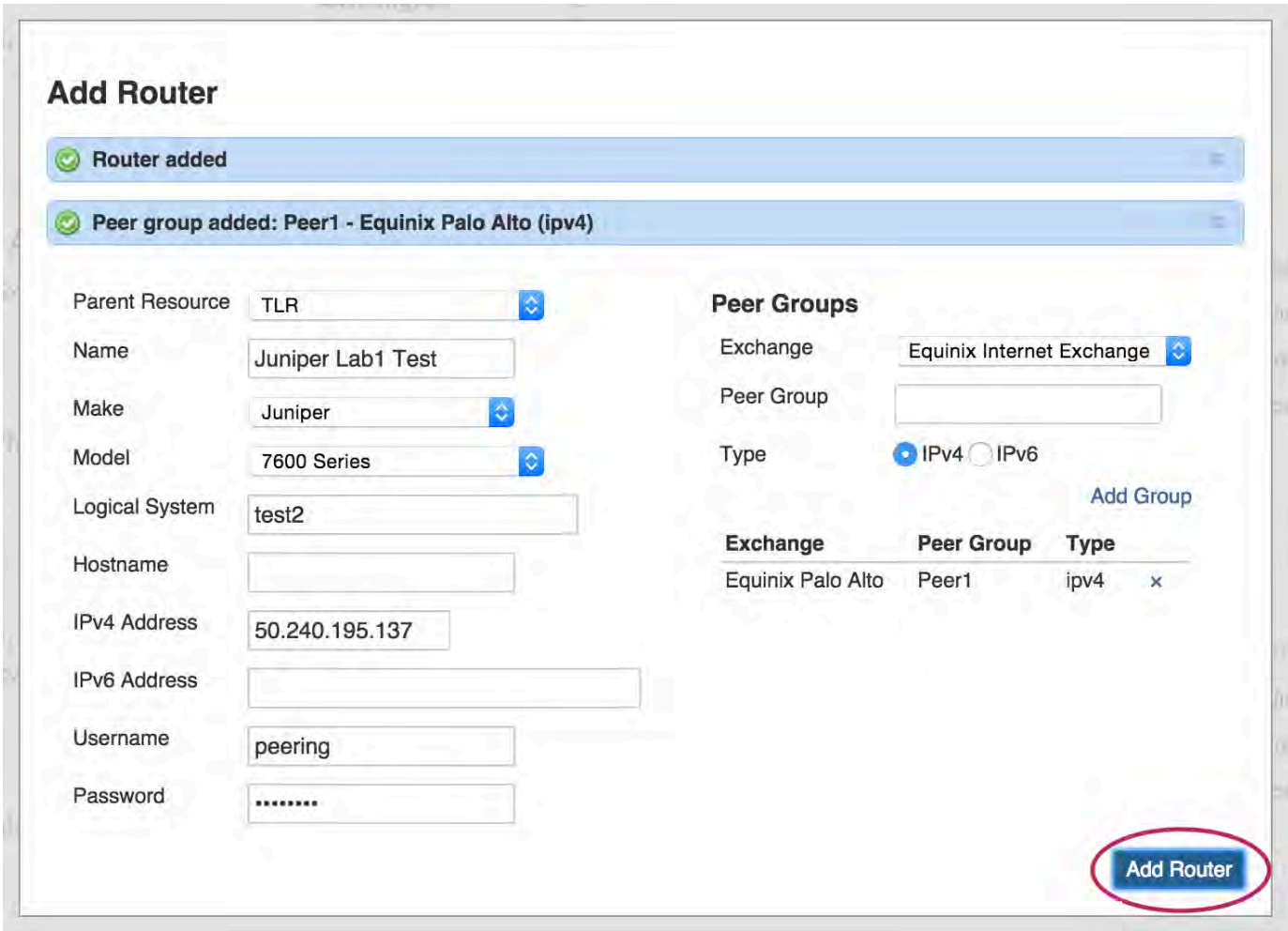
Adding Juniper Routers with Logical Systems

Adding a Juniper router with Logical Systems follows the standard process listed above, with one difference - adding in the Logical Systems information.

When you select a Juniper router make/model, the Logical System text field will appear.



Type the Logical Systems information for the router, then resume entering the rest of the router information and peer groups. Hit "Add Router" when complete.



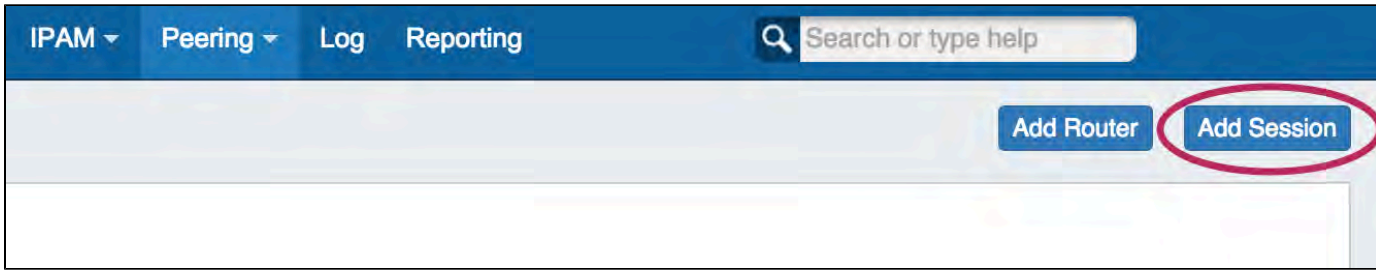
Routers with Multiple Logical Systems
For routers with multiple associated Logical Systems, you may create duplicate router resources utilizing the same router information, but with different logical systems entries.

Add Sessions

Add Sessions

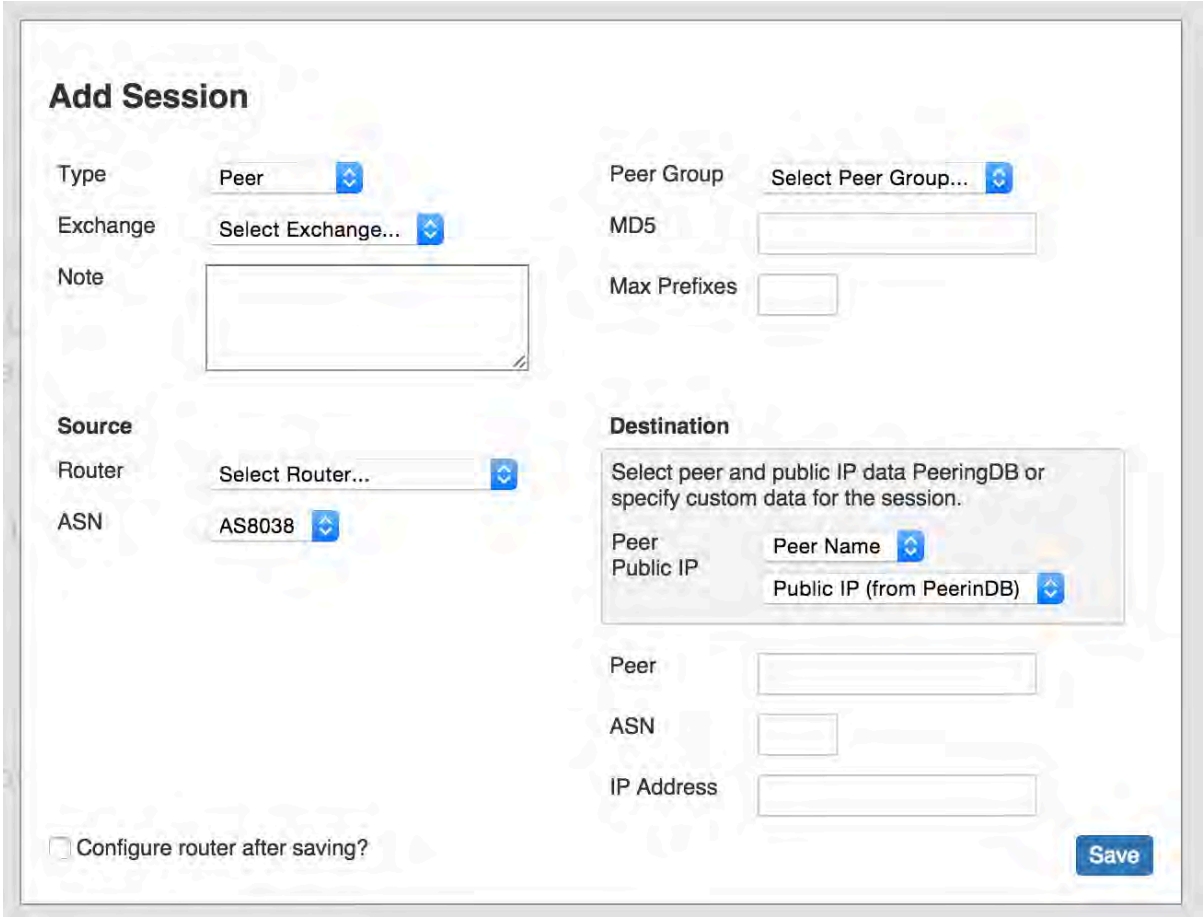
Adding a Peering Session

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



In the Add Session dialog, 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.

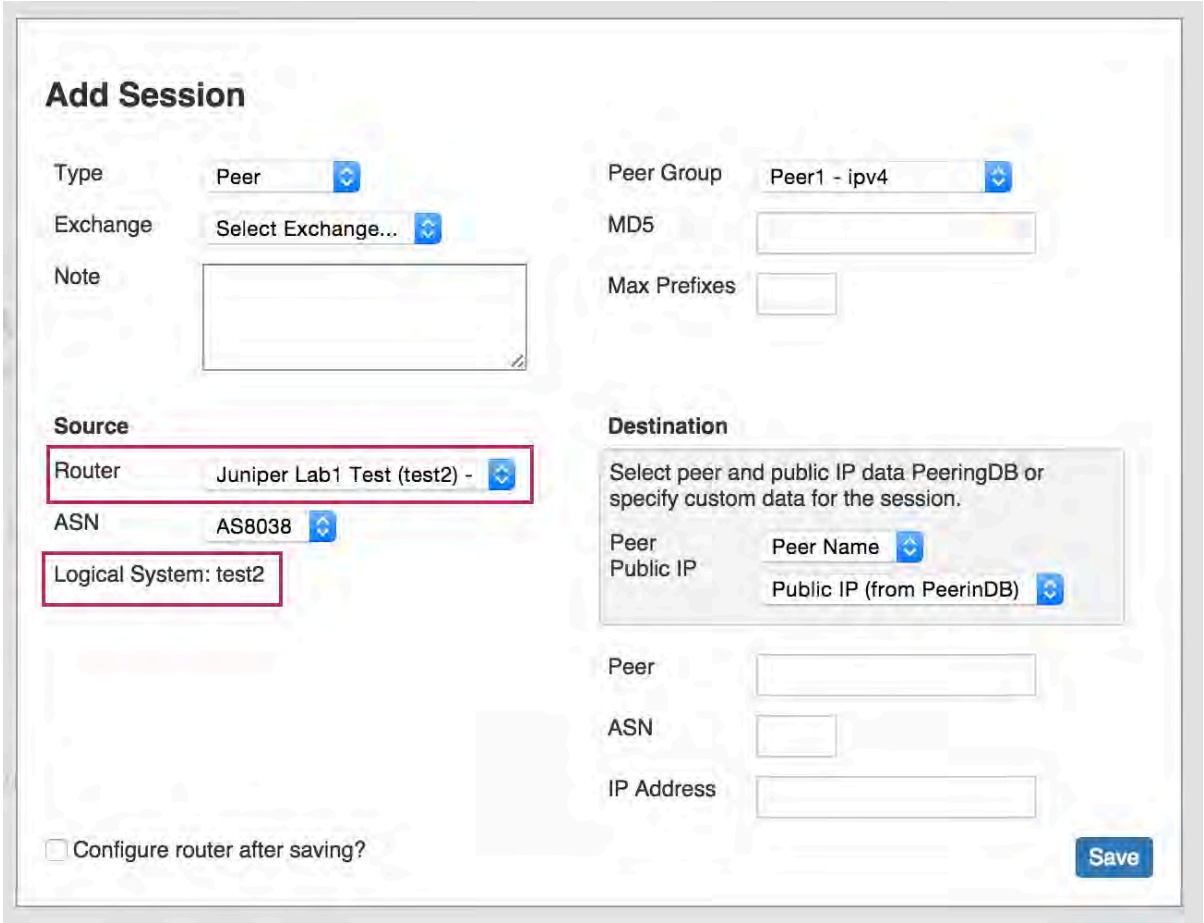


If you would like for the router to be automatically configured when adding your session, check the "Configure Router After Saving" box, then hit "Save". If left unchecked, the session can always be configured later in the Peering Manager.

Adding Sessions with Logical Systems Routers

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. Look for the router name, with the Logical System info in parenthesis (e.g. "Juniper (test)").

The Peer Group associated with that router / Logical System will automatically be selected. Continue to fill in your session information, then hit "Save".



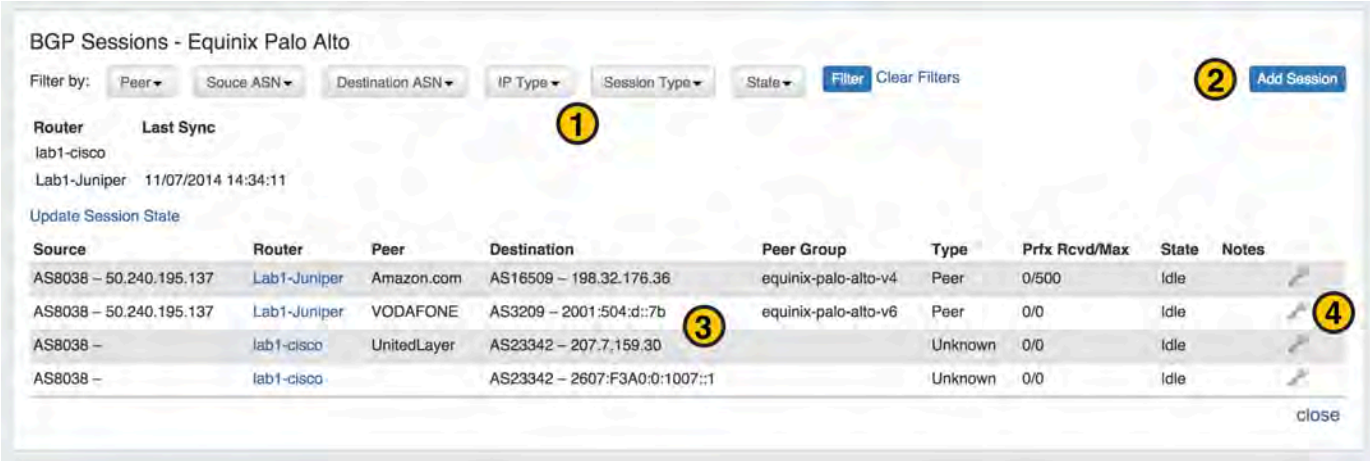
Managing Peer Sessions

Managing Peer Sessions

The Peering Manager:

To bring up the Peering Manager, click on "Sessions" for the desired exchange in the **Peering** tab.

The Peering Manager UI:



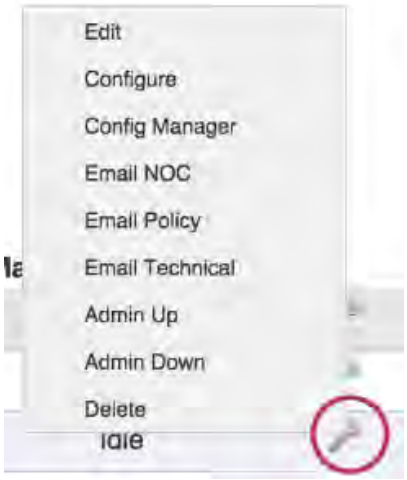
1) Filter Options: The sessions list may be filtered by Peer, Source ASN, Destination ASN, IP Type, Session Type, or State. Once you've chosen the filter criteria, click on "Filter". Select "Clear Filters" to return to the full session list.

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) Action Menu (Wrench Icon): Clicking on the wrench icon will bring up the following tools to manage your sessions:

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.

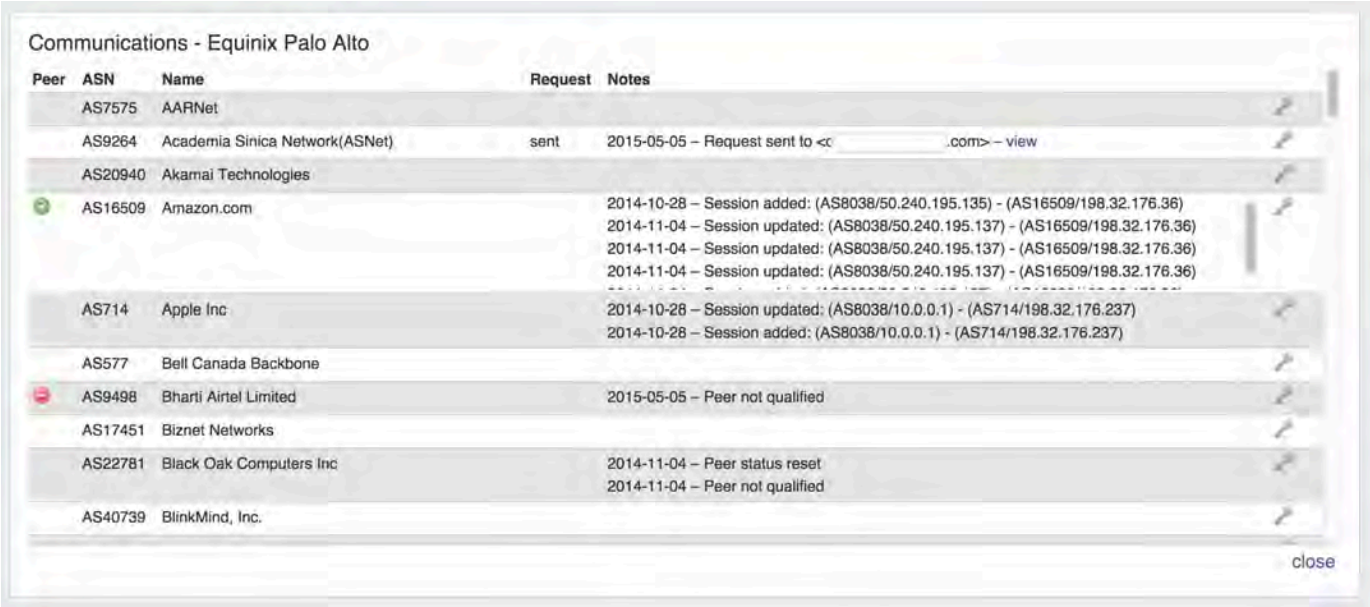
Managing Peer Communications

Managing Peer Communications

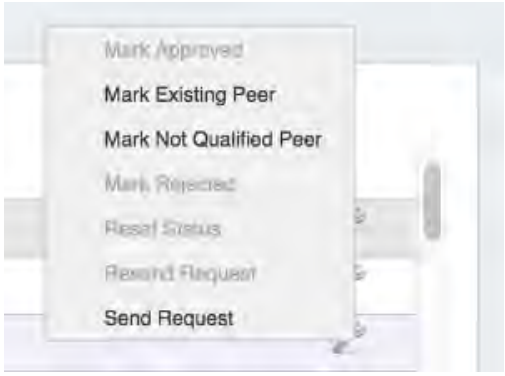
Communications Manager

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

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**.



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.

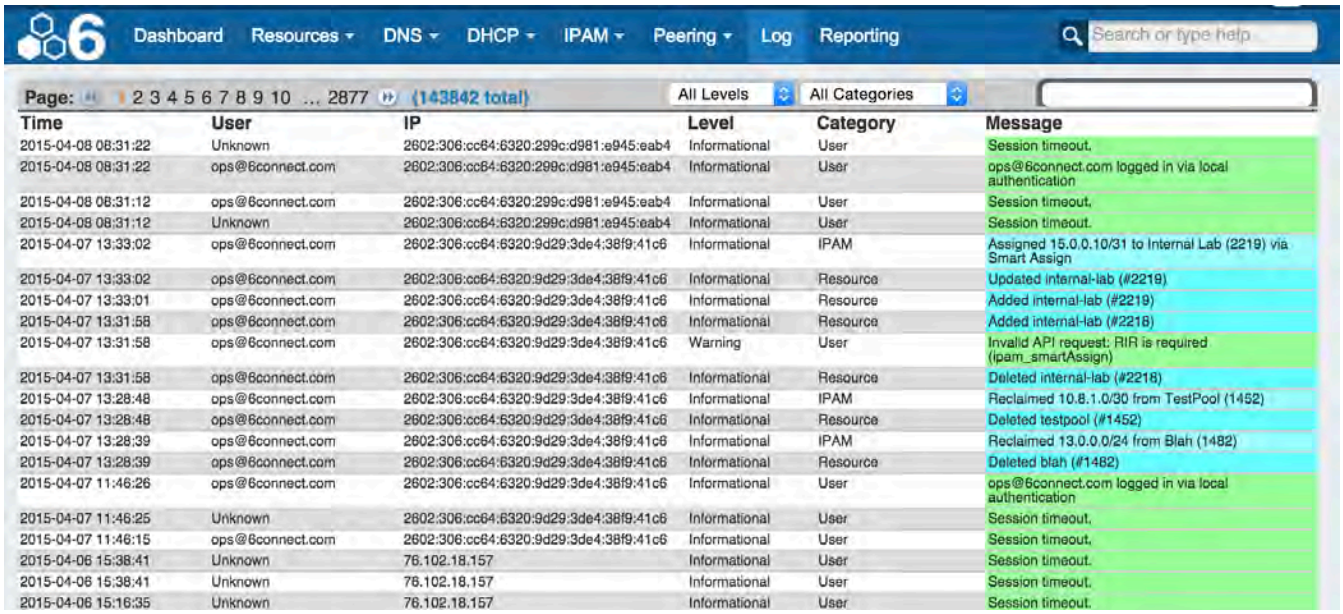
Resend Request: Resends the peering request.

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.

Log

Log

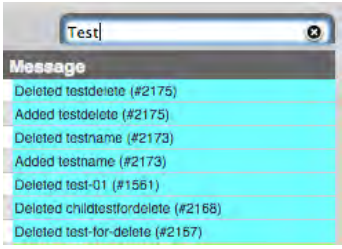
The 6connect ProVision log provides detailed information on actions performed in ProVision.



Log Features

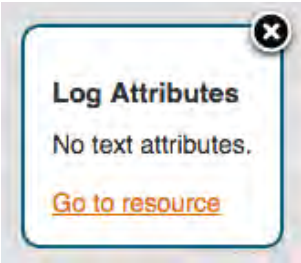
Search:

- Enter the search text above the Message column

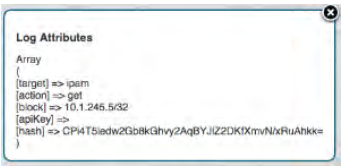


Additional Details:

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

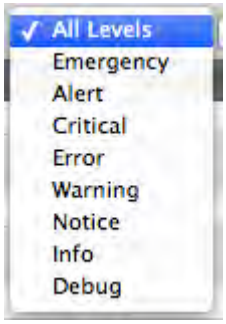


- Clicking on an API log item provides additional details about the API call

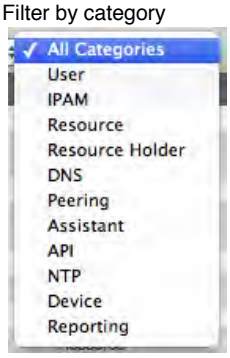


Filters:

- Filter by notification level



- Filter by category



Reporting

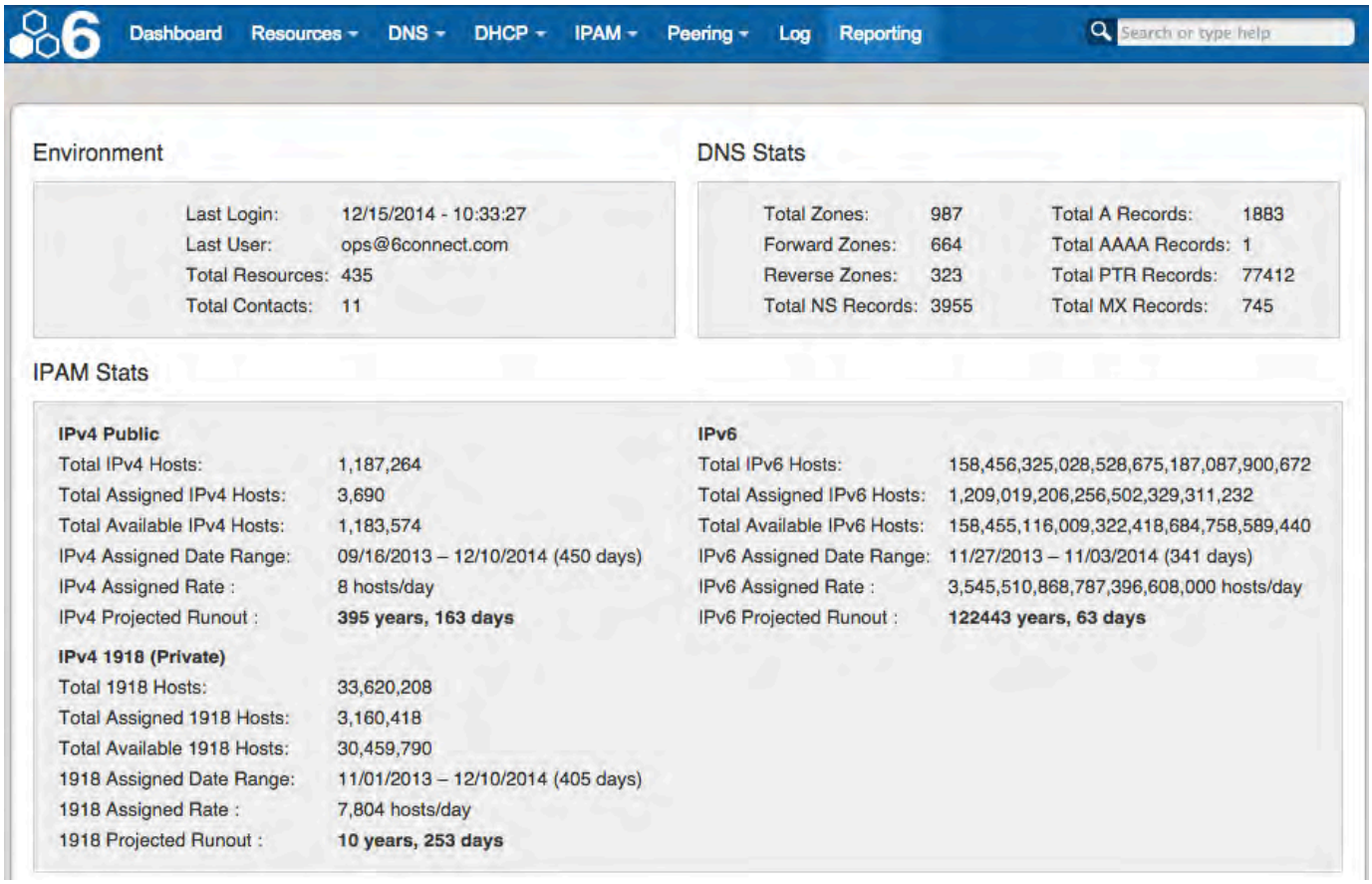
Reporting

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

- Reporting
  - Stats
  - Reports
    - User Activity
    - Customer List
    - IPAM

Stats

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



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".

Customer List

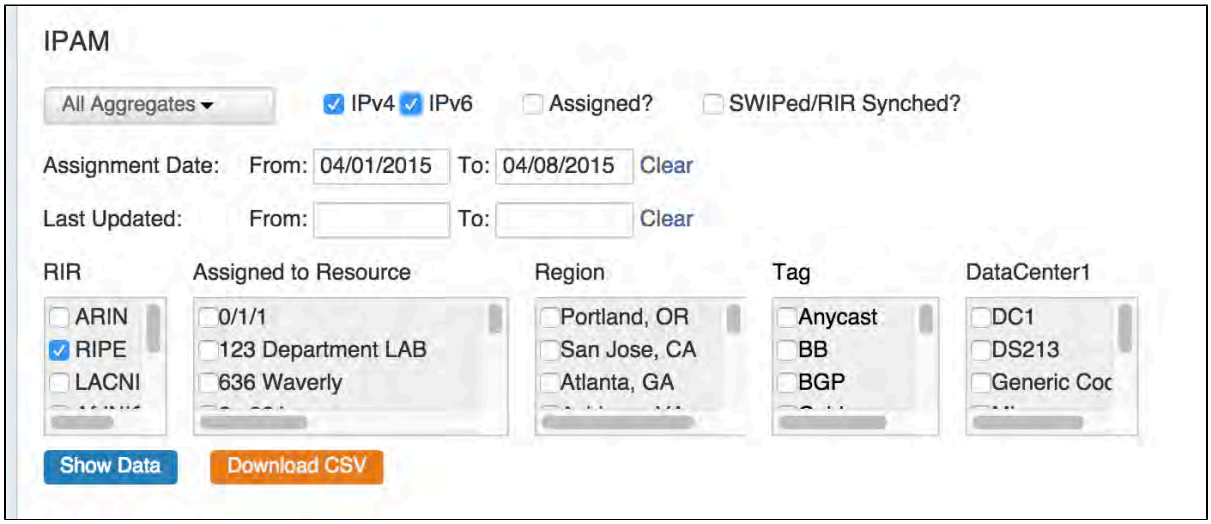
The Customer List report reflects all Resources created under the Category of "Customer". Clicking on "Show Data" will show information collected from the Contact Info and Tech Info gadgets, parent information, and IP / zone assignment counts. To export the data to .csv, simply select "Download CSV".

IPAM

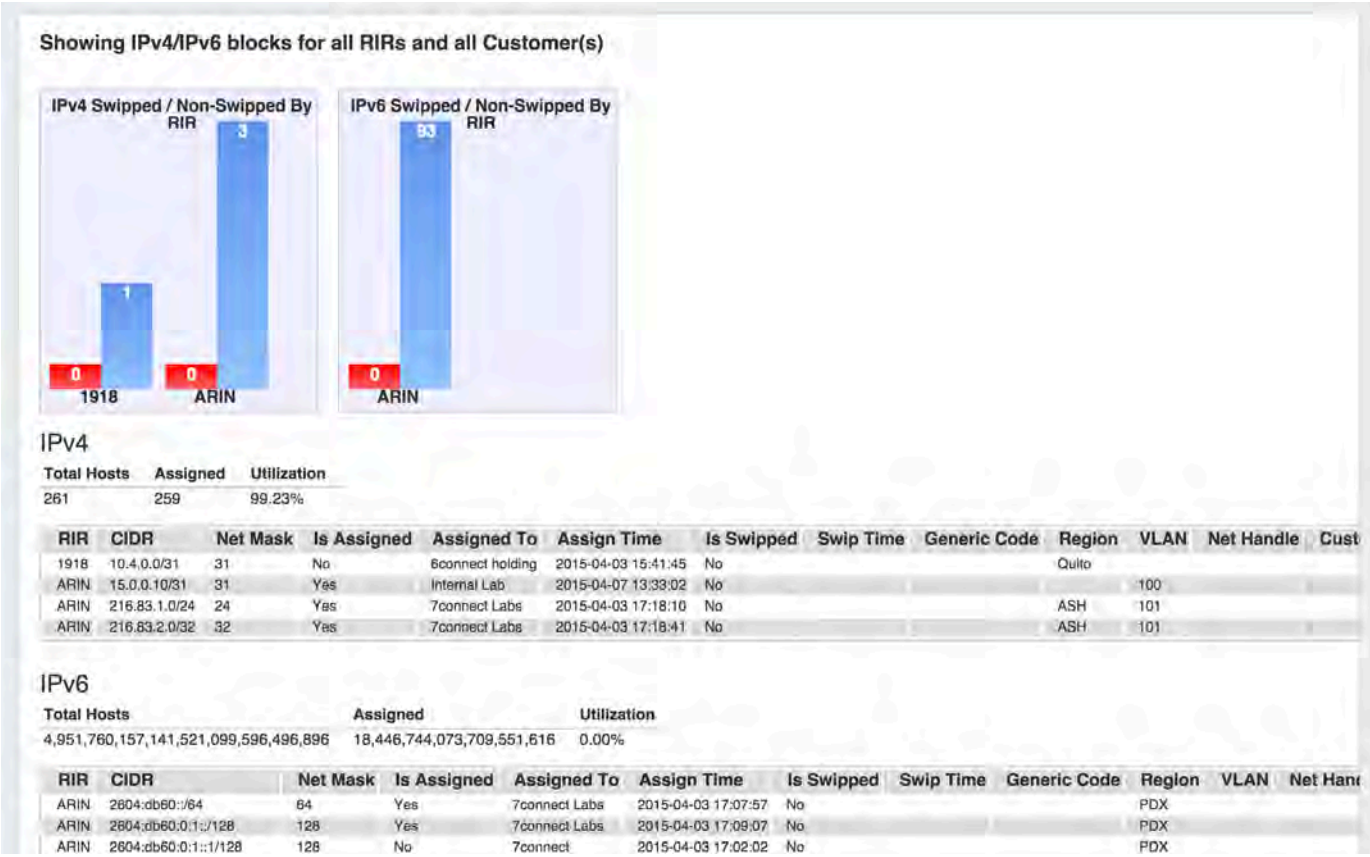
The IPAM report is highly customizable, allowing you to view information for all aggregates or selected blocks.

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.



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".



Note: The default number of entries returned in the Reporting section for either html or .csv is limited to 5000 rows. If more rows are needed, the ProVision API may be used to retrieve larger datasets.

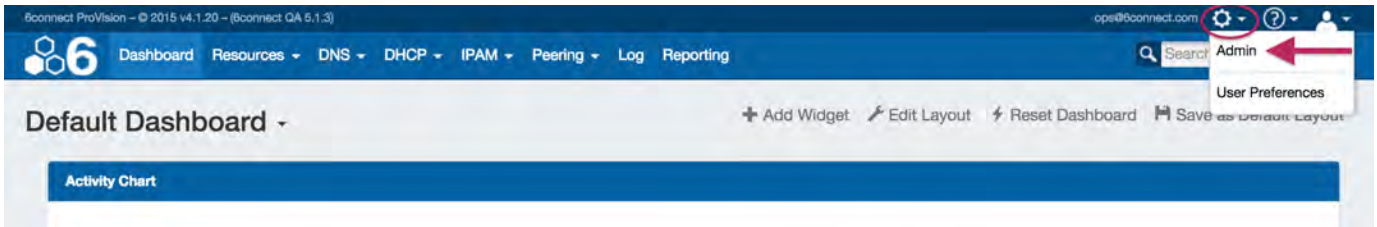
ProVision Admin Guide

Admin Guide

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

Accessing the ProVision Admin section

To access the Admin area, click the gear icon at the top right section of the header from any page. From there, select "Admin". You will then have access the Admin section tabs of ProVision, and will see the Admin Preferences page.



Leaving the ProVision Admin section

To leave the Admin area, simply click the "Exit Admin" link in the navigation bar. You will be redirected to the Dashboard.

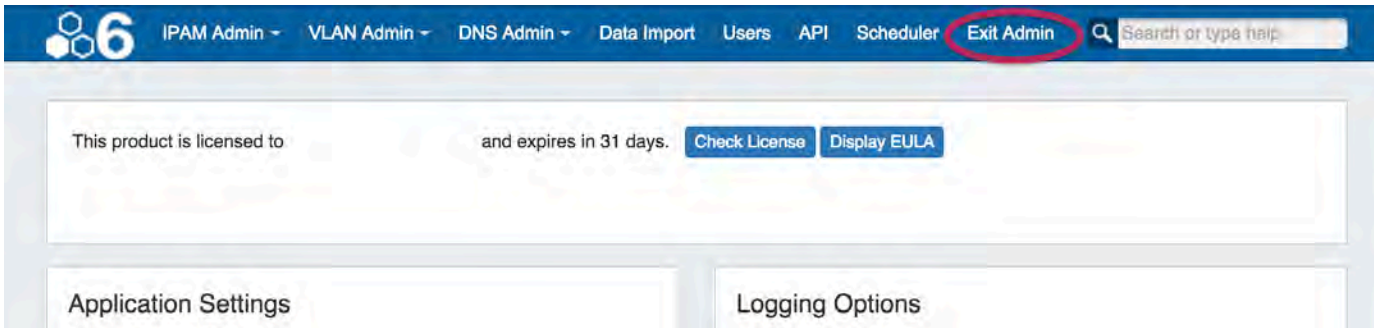
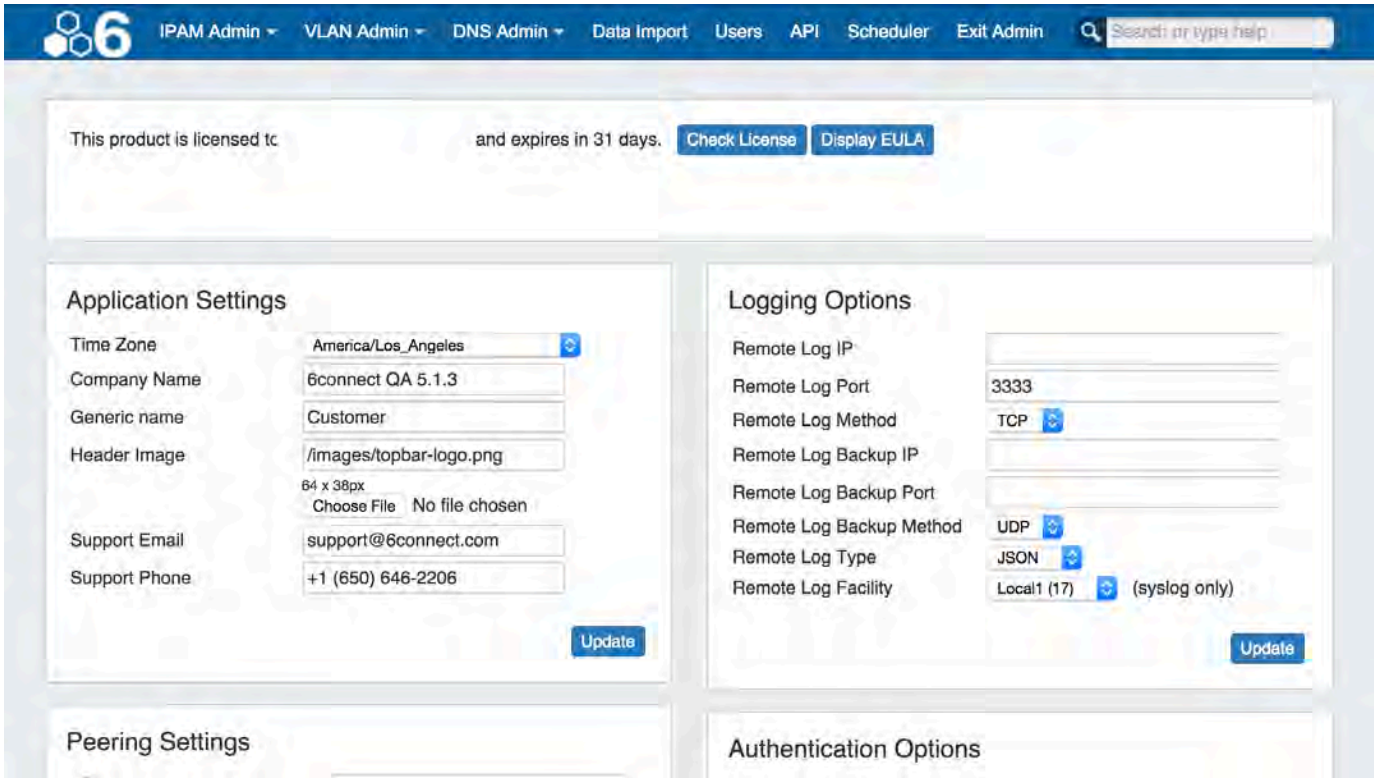


Table of Contents

Admin Preferences

Overview

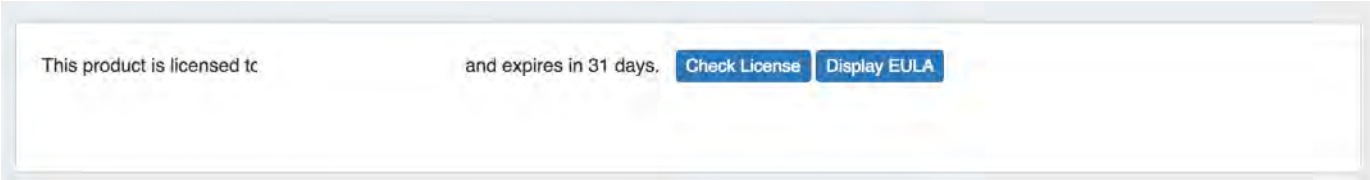


The Admin Preferences page is the home page of the Admin section of ProVision where general platform preferences may be set.

To access it, click the gear icon at the top right section of the header. From there, select "Admin". You will then have access the Admin section tabs of ProVision, and will see the Admin Preferences page.

Video Walkthrough

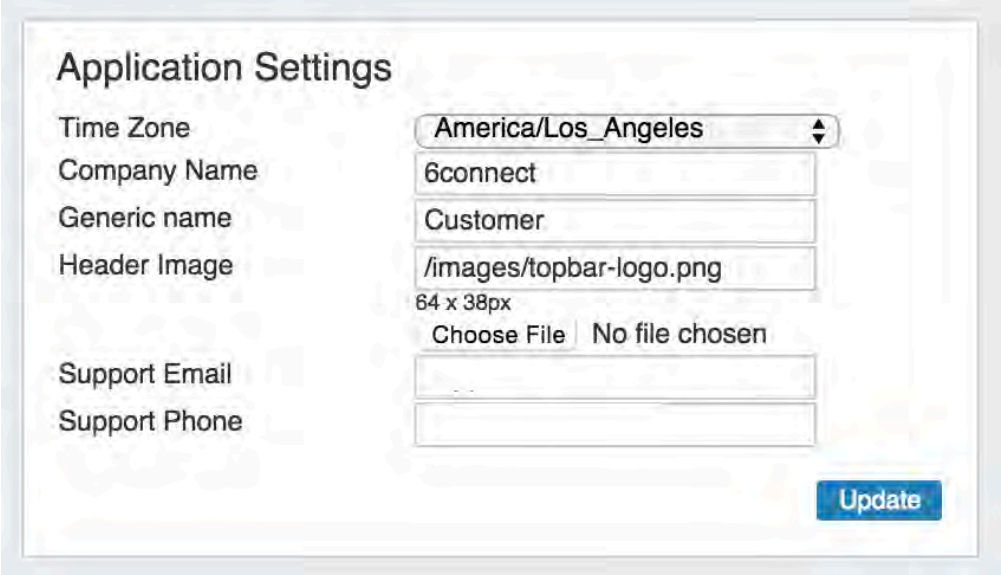
License Info



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

Application Settings

Application Settings is where you provide company specific info that appears in the header and Dashboard.



Time Zone: Supported Time zones are listed here: {EXT} 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.

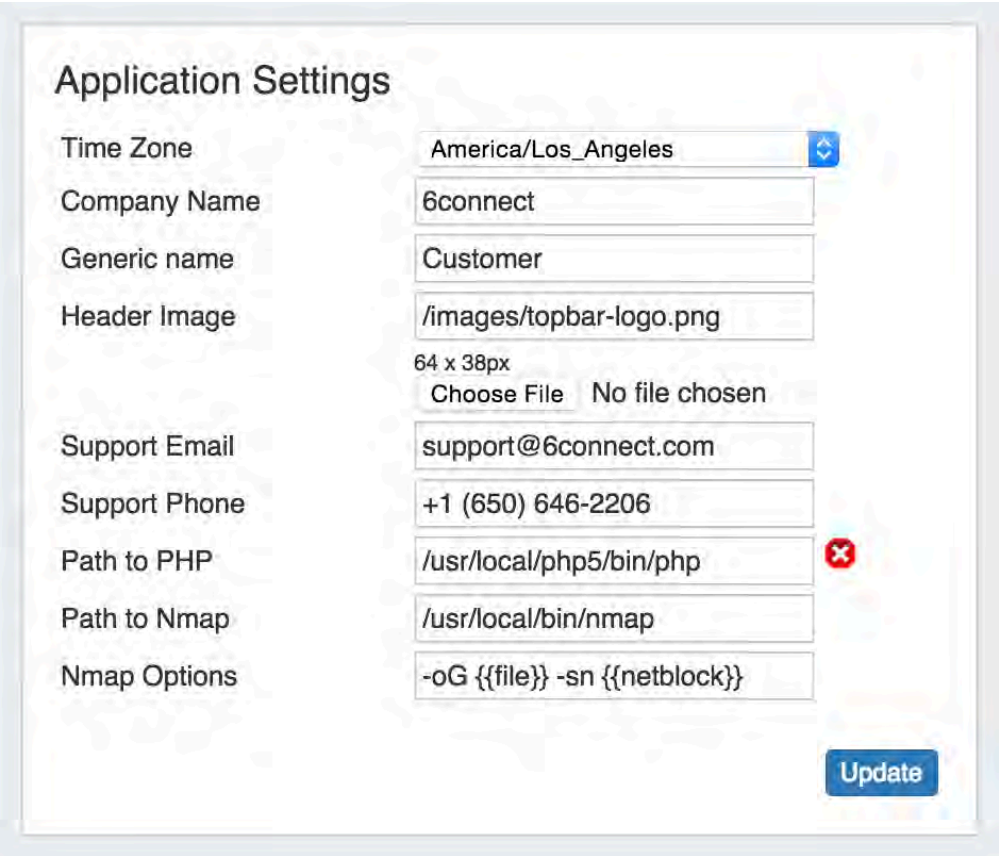
Header Image: Select an image file for the header

Support Email: Support Email address that displays on the Dashboard

Support Phone: Support phone number that displays on the Dashboard

Application Settings - Local Installation

Additional settings are available for local installations:
Local installations: Click here for additional settings...



Path to PHP (Local Installation): The directory path to php location

Path to Nmap (Local Installation): The directory path to nmap location

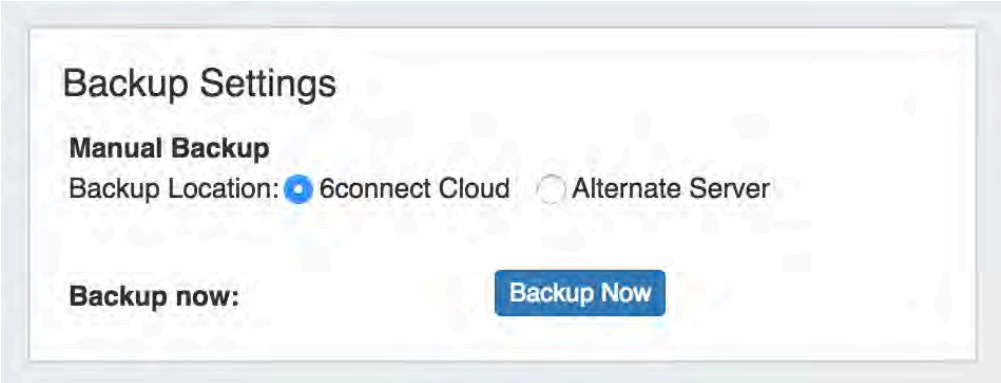
Nmap Options (Local Installation): The directory path to nmap location

Peering Settings

ASN : Enter the ASN(s) that will be used for Peering. Separate multiple ASNs with a comma.

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

Backup Settings

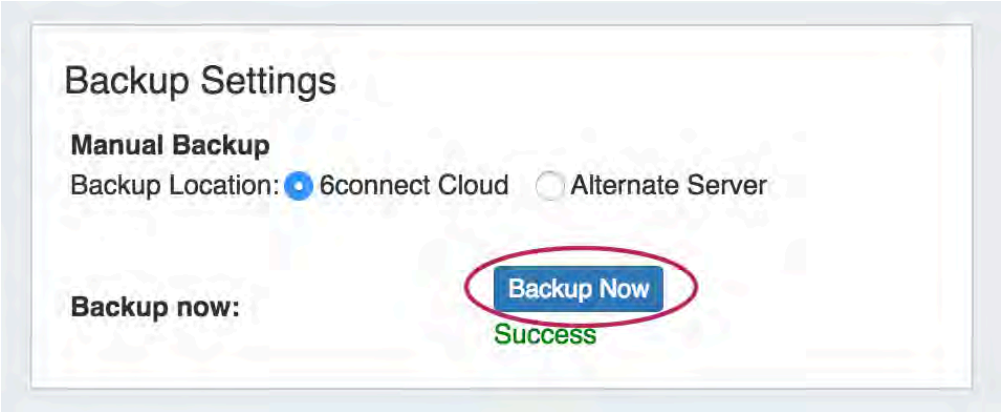


For cloud users, regular backups can be set up through the Scheduler. However, prior to imports or other large changes, you may wish to manually perform a backup.

Backup Location: The backups may be sent to the 6connect cloud, or to a specific server in the Resource system. Select the radio button for the desired location.

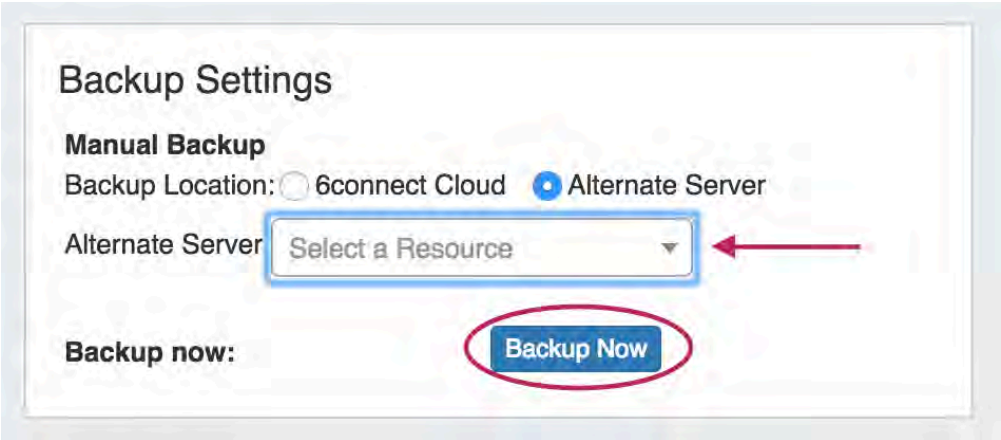
Manual Backup - 6connect Cloud:

Select "6connect Cloud" as your backup location, then click on the "Backup Now" button. You will see a success message below the button if successful.



Manual Backup - Alternate Server:

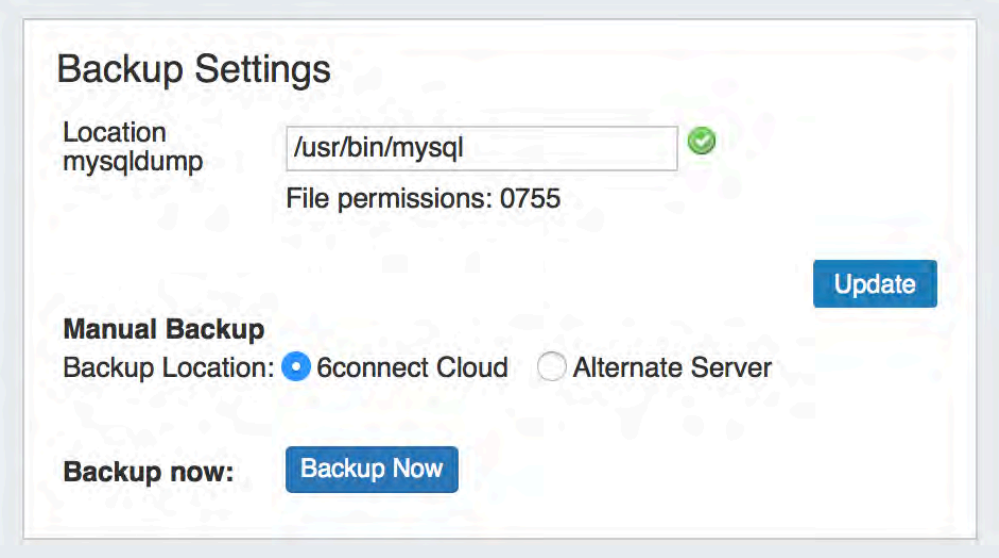
Select "Alternate Server" as your backup location, then select a server Resource that exists in ProVision to send the backup. After selecting your server, click on the "Backup Now" Button.



If selecting a server Resource in ProVision for the backup, verify that the server fields Hostname, Username, and Password are filled in and correct.

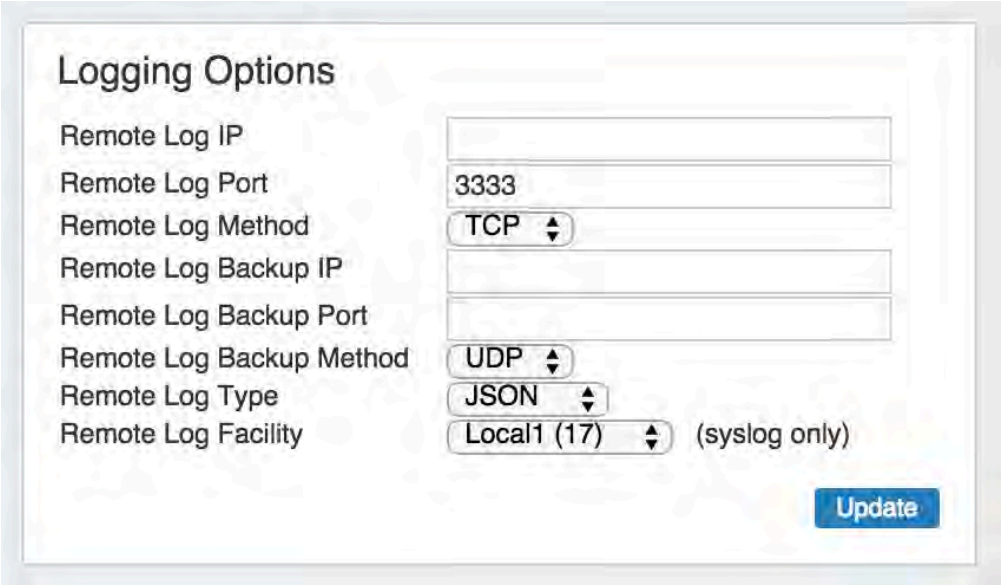
Backup Settings - Local Installation

Additional settings are available for local installations:
Local installations: Click here for additional settings...



Location of mysqldump (Local Installation): This is the location of the mysqldump directory.

Logging Options



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

Authentication Options

Maximum Session Idle: 60 (minutes) This value controls how long a session can stay idle before being forced to log in again.

RADIUS functions are available.

Radius Enable	☑
Radius Server Address	208.39.104.106
Radius Authentication Port	1812
Radius Accounting Port	1813
Radius Key	6connect

The Radius key is the Radius Server Secret.

Radius servers must be configured with the 6connect dictionary, located here.

LDAP functions are available.

LDAP Enable	☑
LDAP Server Address	dev1-dfw.6connect.com
LDAP Port	389
LDAP Security	None ⬍ Test Server
LDAP Auth DN List	cn=%LOGIN%,ou=people,dc=6con
LDAP Fetch DN	cn=%LOGIN%
LDAP Group Attribute	memberOf

The DN strings used to first authenticate the 6connect user and then to retrieve their permissions.

The string '%LOGIN%' should be inserted in place of the user's common name in both strings.

ex: cn=%LOGIN%,ou=people,dc=6connect,dc=com

Each Auth DN string will be tried in order until a user successfully authenticates.

LDAP servers must either be configured with the 6connect schema, located here, or have an internal list of user groups defined by the LDAP Group Attribute above. If a Group Attribute is set it will be used. If no Group Attribute is present the 6connect schema will be used. If both fail then users will not be able to log in to ProVision.

Update

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)

Note: 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

Note: 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

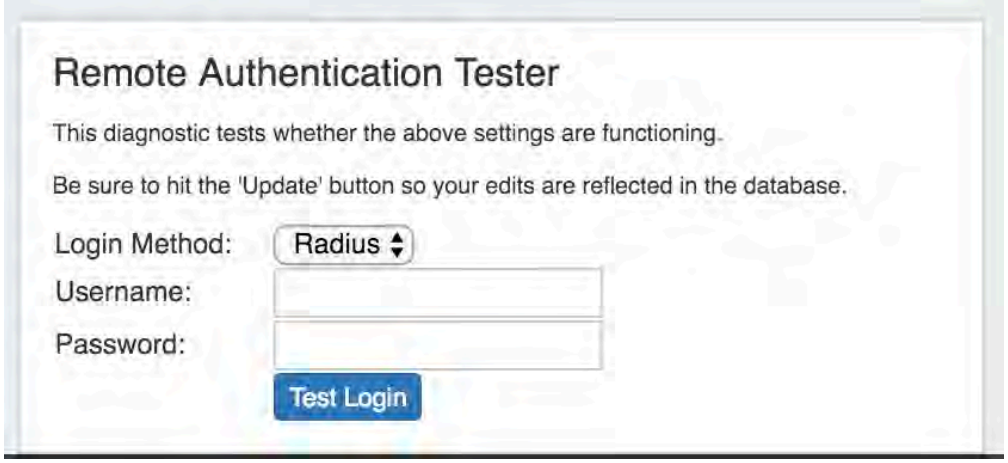
LDAP Security: Select the security method of your LDAP server - SSL, TLS or None

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)

LDAP Group Attribute: If using an internal list of user groups instead of 6connect groups, enter the attribute name for the LDAP groups here. If a Group Attribute is set, it will be used first, otherwise the 6connect schema will be used.

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.

Remote Authentication Tester



The Remote Authentication Tester checks Radius / LDAP settings for a user.

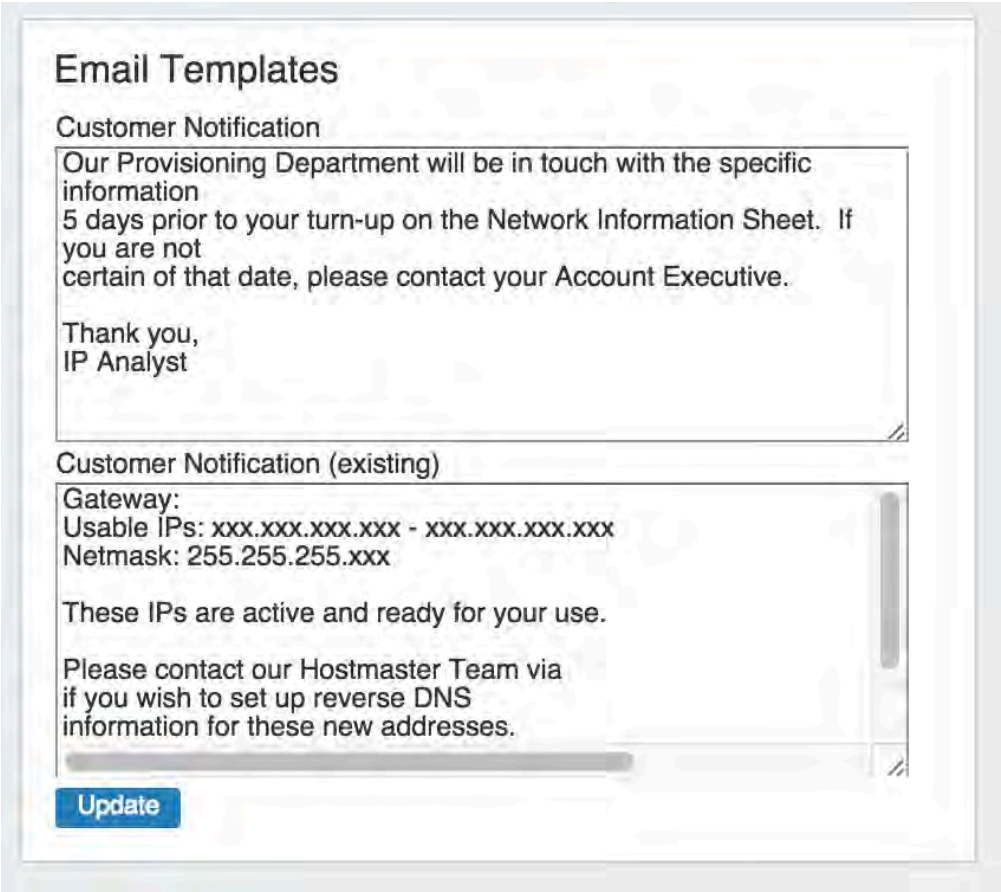
Select the Login Method (Radius or LDAP), enter the Username and Password for the user, and then click "Test Login".

Login Method: Select Radius or LDAP, according to your authentication settings.

Username: The username for the user you are testing.

Password: Password for the user you are testing.

Templates



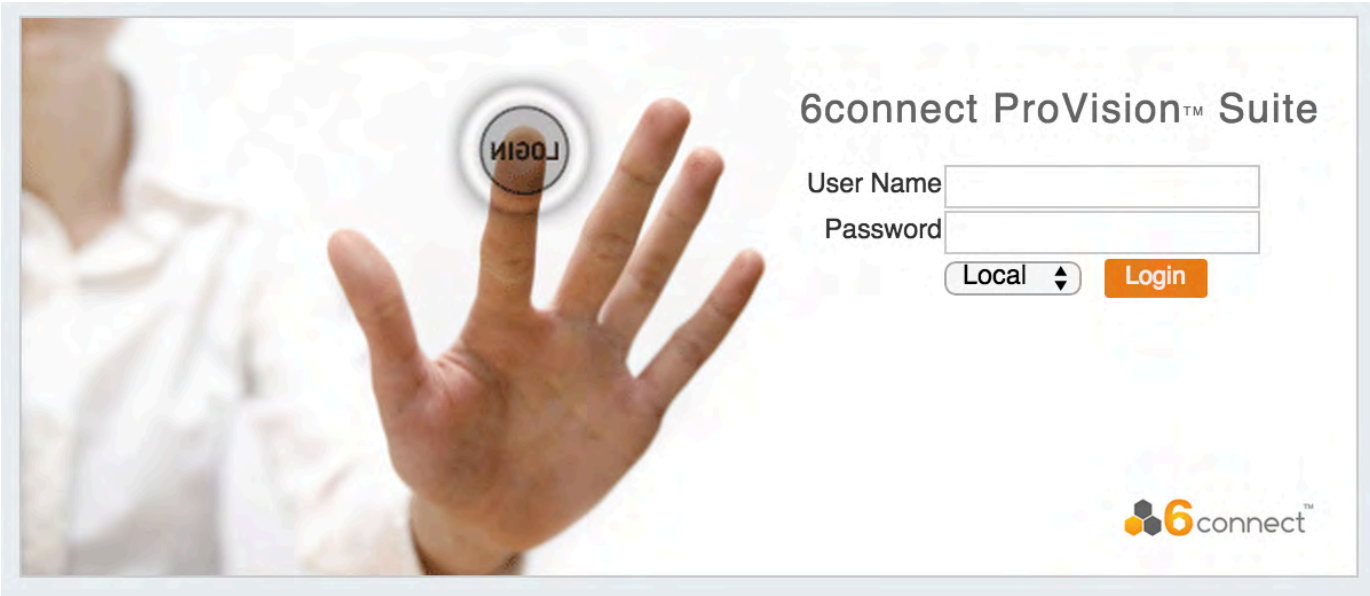
This is where you can edit outgoing email templates for IP block assignments.

To edit, click inside the text area box, make the desired changes, then click the "Update" button.

> After making changes to settings, clicking any "Update" button will save your changes for all sections of the page, not just the section the update button is contained in.

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 following sections for more detail authentication configurations:

- LDAP Authentication
- LDAP Authentication on Windows Server
- RADIUS Authentication

LDAP Authentication

LDAP Authentication

To setup an LDAP server for authentication, you must perform the following three procedures:

Configure the LDAP Server:

Configuring the LDAP server involves ((adding the schema)) and adding LDAP users / groups to the server:

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 ) )
```

Creating a LDAP User:

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 -b [BASE] -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.

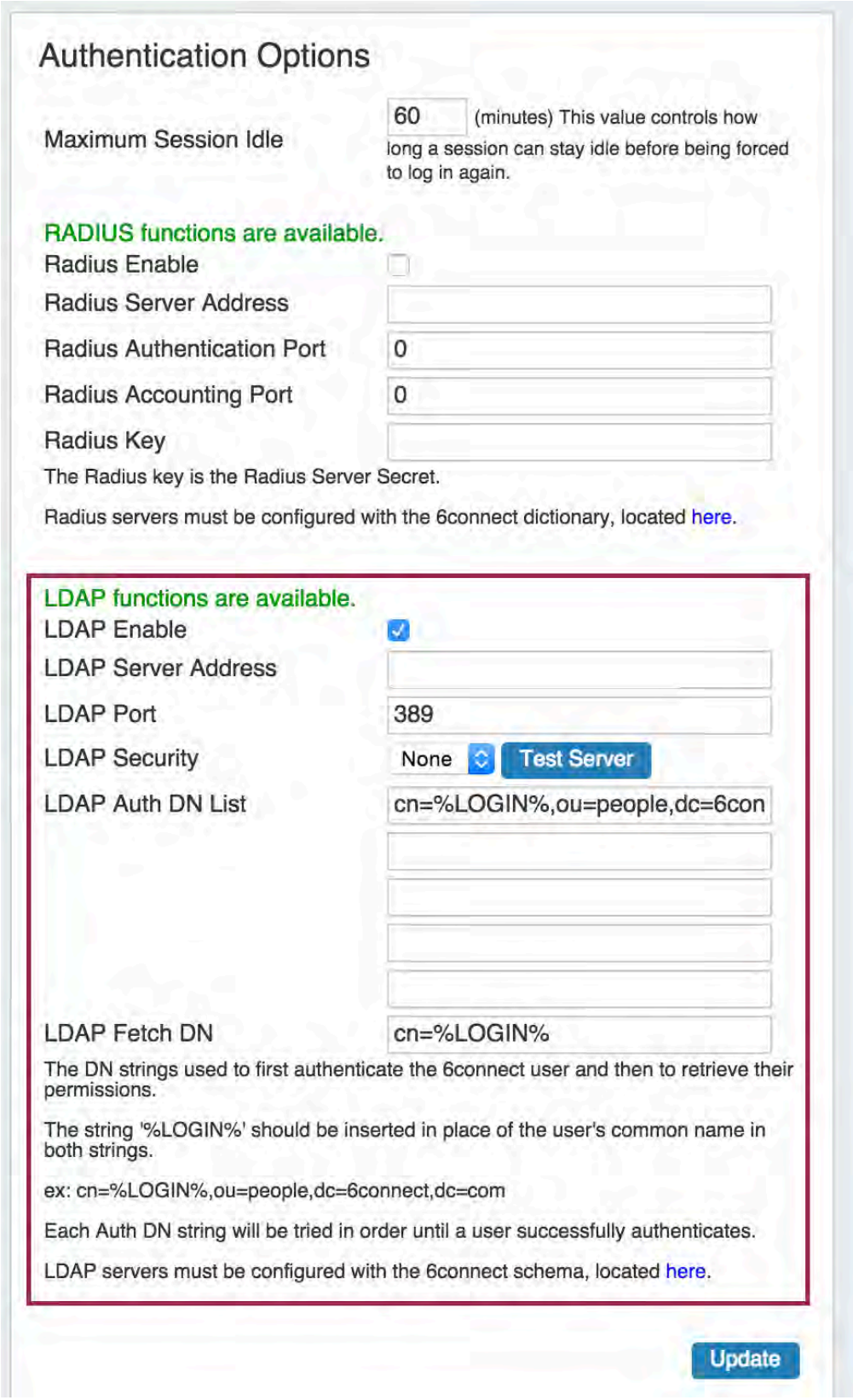
Example:

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

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.



Example values in this case would be:

LDAP Enable: (Checked)

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:

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 -b [BASE] -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.

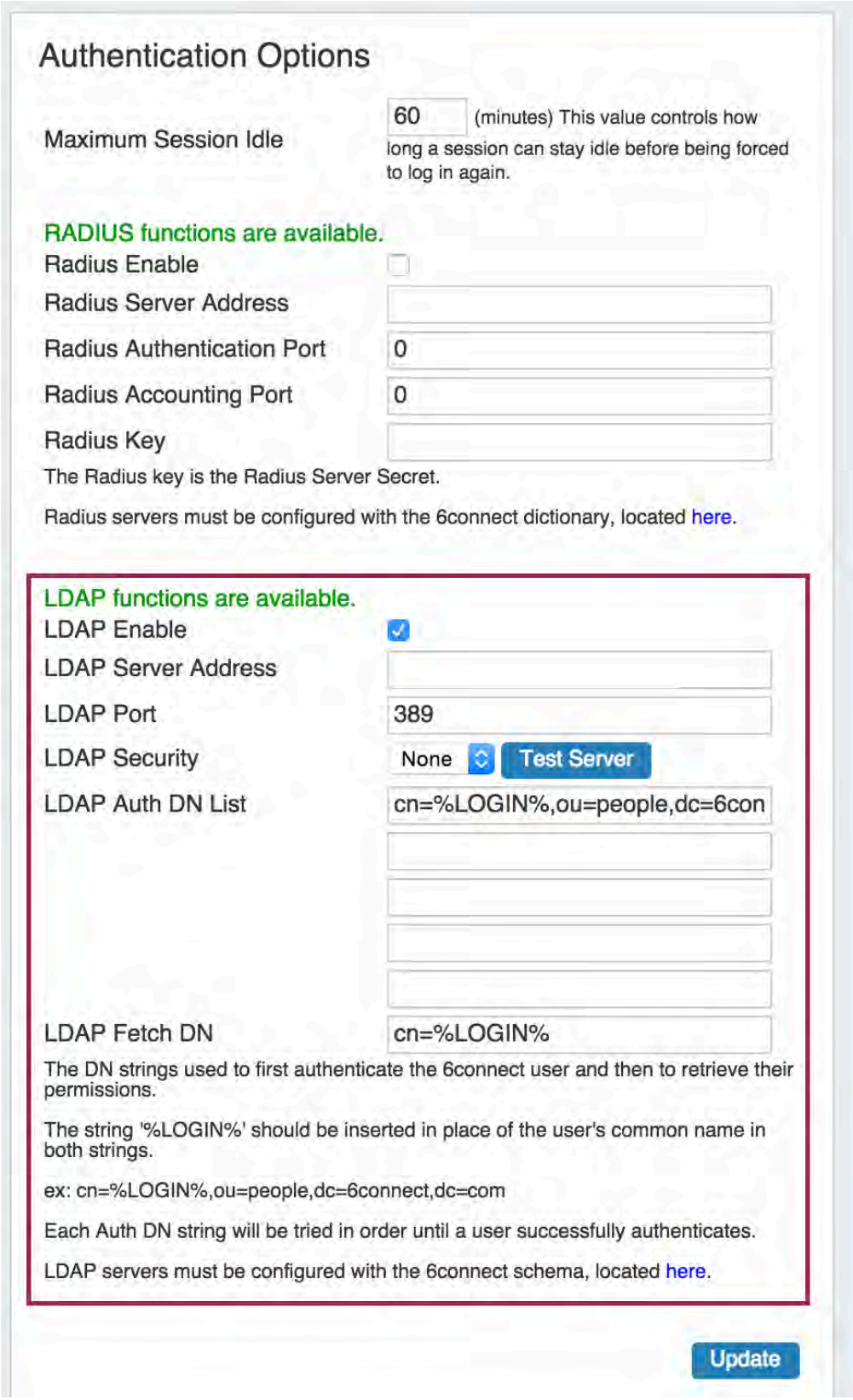
Example:

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

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.



Example values in this case would be:

LDAP Enable: (Checked)

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

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

Add the 6connect VSA to the Radius Installation

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:

ProVision 4.0 and greater:

```
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 ProVision 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 Nonexistant"
```

The Radius server must be restarted every time you add, remove, or modify users. To restart the Radius server, use this command:

```
/etc/init.d/radiusd restart
```

> **Note on RADIUS attributes**
> There are many Radius attributes, but '6connect_user_group' is the one used by 6connect ProVision. It is a comma-separated list of all the group names that the user belongs to.

Test Radius Accounts

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

To query a radius server, use the following command format:

```
radtest [USERNAME] [USERPASSWORD] [SERVER] 0 [SECRET]
```

Example:

```
radtest bobber hello 208.39.140.106 0 6connect
```

A successful response will look like this:

```
Sending Access-Request of id 198 to 208.39.140.106 port 1812
User-Name = "bobber"
User-Password = "hello"
NAS-IP-Address = 67.221.240.229
NAS-Port = 0
Message-Authenticator = 0x00000000000000000000000000000000
rad_recv: Access-Accept packet from host 208.39.104.106 port 1812, id=198, length=69
Attr-10 =
0x476c6f62616c2041646d696e732c47726f757020322c47726f757020312c47726f7570204e6f6e6578697374
```

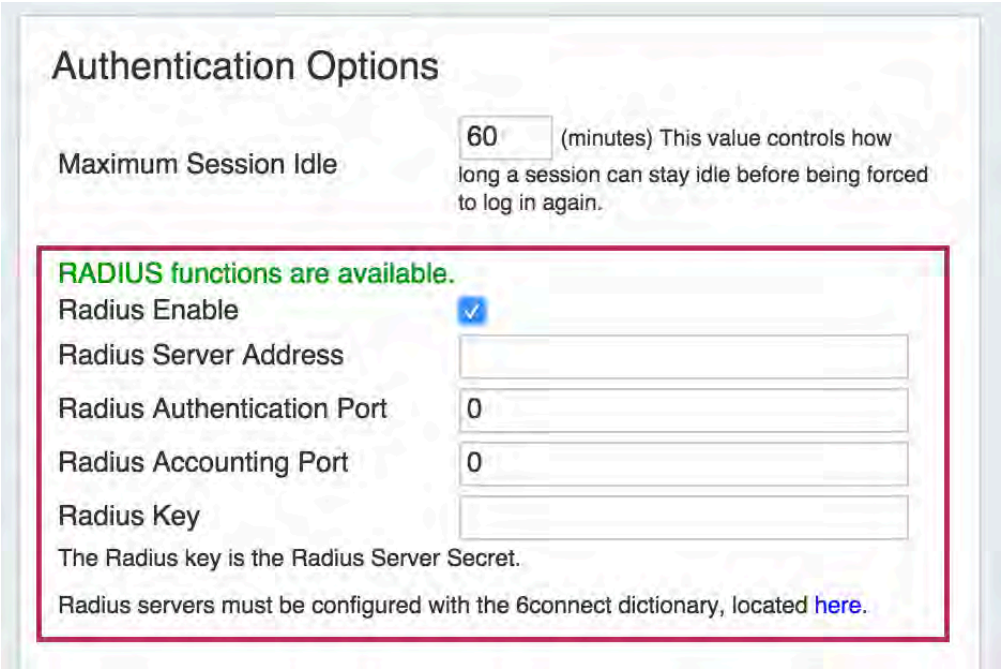
A rejected response may look like this:

```
Sending Access-Request of id 68 to 208.39.140.106 port 1812
User-Name = "bobberbro"
User-Password = "hello"
NAS-IP-Address = 67.221.240.229
NAS-Port = 0
Message-Authenticator = 0x00000000000000000000000000000000
rad_recv: Access-Reject packet from host 208.39.104.106 port 1812, id=68, length=20
```

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 -> 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:

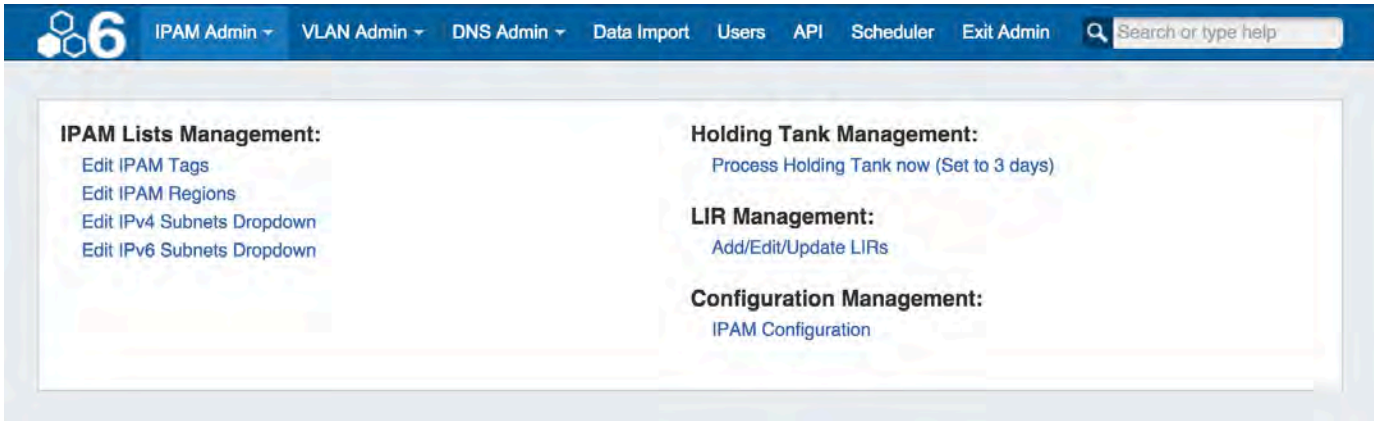
Go to the /data/globals.php and open in vi (or other editor). Add in the following text as the last line of the file (before the closing ?>)

```
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

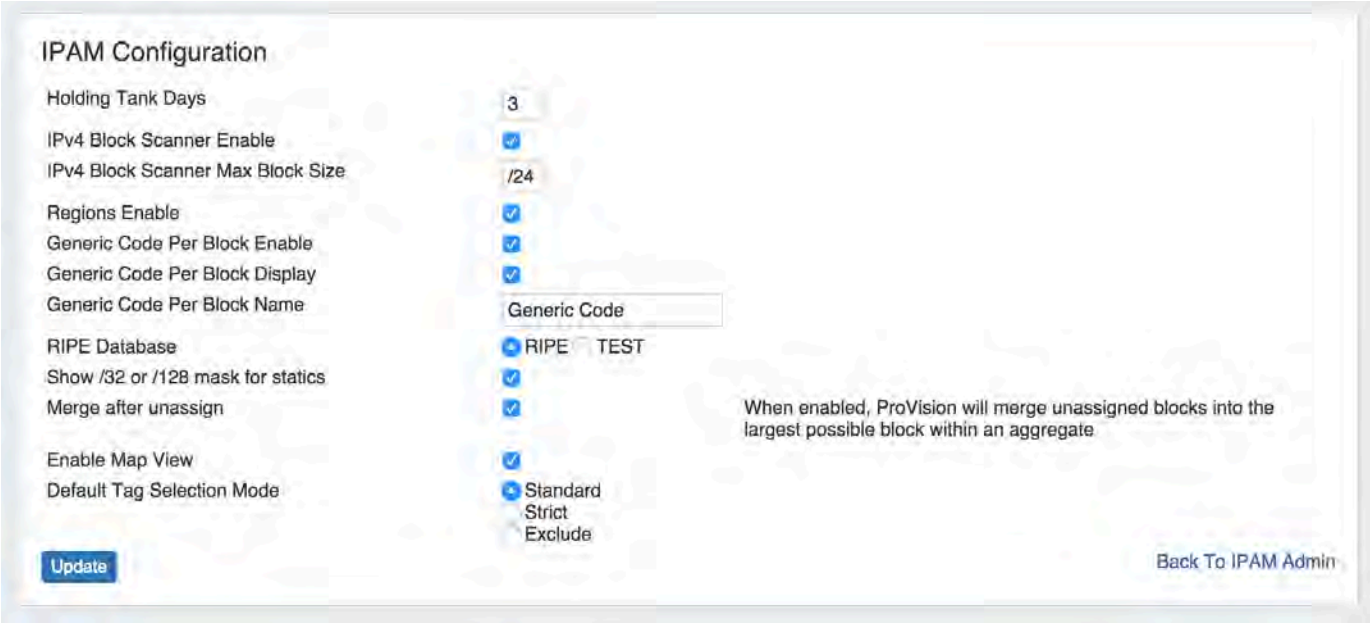


IPAM Administration is accessed through the Admin area of ProVision. It includes sections to manage IPAM Lists, the Holding Tank, LIR, and IPAM Configuration.

IPAM Lists Management

These links are to the respective IPAM Parameters that are available for customization - tags, regions, and subnets. Go to the IPAM Parameters page for more details and examples for customization.

IPAM Configuration



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

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

addresses.

IPv4 Block Scanner Max Block Size: The max size (mask) the IPv4 Block Scanner considers.

Regions Enable: Check the box to enable "Region" tags for IP blocks. This will add a "Region" column to the IPAM Manage screen. Values may be set from "Edit IPAM Regions" under IPAM Admin - IPAM Lists Management.

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

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

Generic Code Per Block Name: This is the label that will be displayed for the Generic Code field.

RIPE Database: Select the desired database

Show /32 or /128 mask for statics: Enable to show /32 or /128 masks.

Merge after unassign: Select to automatically merge adjacent blocks when they become available after unassign.

Enable Map View: Select to enable a map view of assigned aggregates on the IPAM tab, determined by assigned region.

Default Tag Selection Mode: Set which radio button will be selected by default when working with tags in the IPAM Gadget

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.

Additional Information:

- IPAM Parameters
- Holding Tank Management
- LIR Management and Use

IPAM Parameters

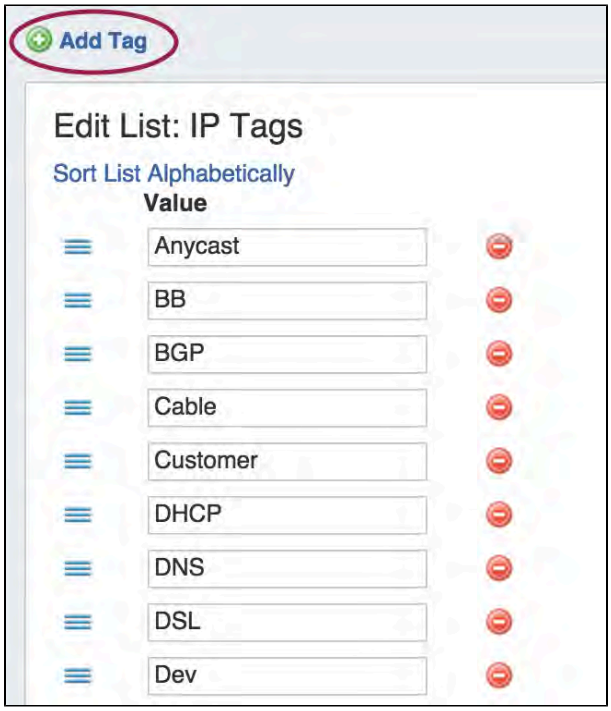
IPAM Parameters - Overview

IPAM Lists Management is access from the Admin section of ProVision, under the **IPAM Admin** tab. Through this area, admin users can update IPAM tags, regions, IPv4 subnets, and IPv6 subnets.

Add / Edit IPAM Tags

When you are applying properties to IP blocks, you have the option to edit tags. IPAM Tags are used in a number of areas in ProVision and can be added or edited from this screen.

To add a new IPAM tag, click on "Add Tag" at the top of the Edit IPAM Tags page.



Then, type in the desired name value for the new tag, and hit "Add Tag".

To **edit** a tag, simply type your changes in to the text box with the tag name. Tags with unsaved changes will be highlighted until saved.

To **delete** a tag, click on the red "delete" symbol to the right of the tag name.

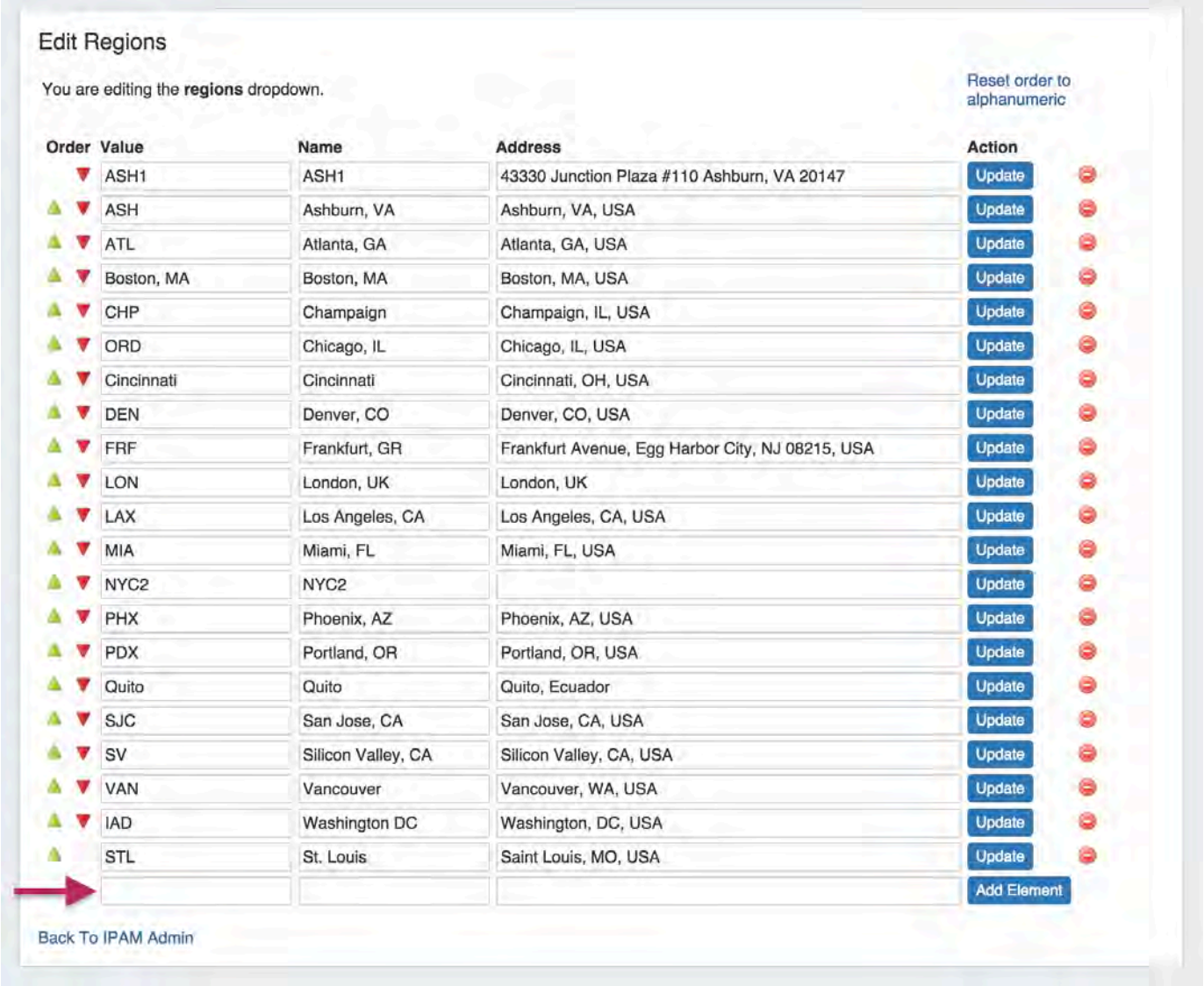
When complete, be sure to click on the "Save" button to save your changes.

Add / Edit Regions

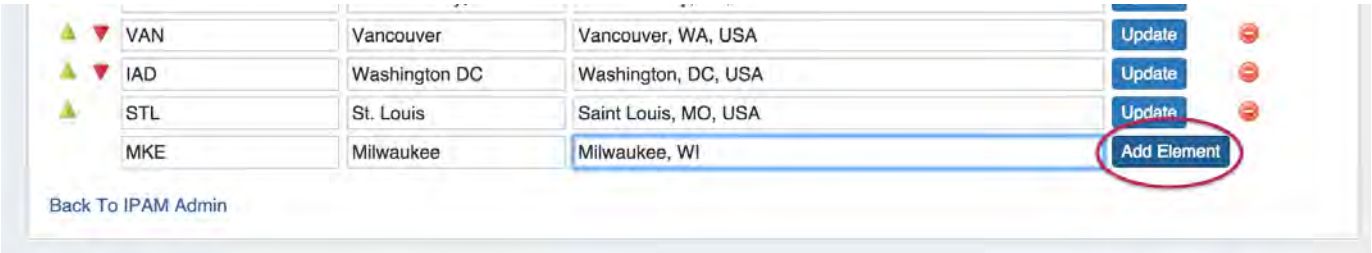
If enabled, Regions can function as a way to further define your network segments (regional tie-downs, etc.). This gives you flexibility for allocations and assignments beyond simply using Tags. Regions are used by the IPAM Gadget, the IPAM Manage UI, and IPAM Map View.

The Regions display includes fields for Value, Name, and Address. 'Name' is the region name that will show in the IPAM Gadget and IPAM Manage screens, whereas 'Value' is the value that will be written to the database, used for API calls, and is also used for some filter selectors. The Address field is used by IPAM map view to geolocate aggregates with that assigned region. Address may be in the form of a City / State / Country or a full street address. The more accurate your address information, the more accurate map view will be with your aggregate locations.

To add a new Region item, type in a new Value and Name into the empty fields at the bottom of the list.

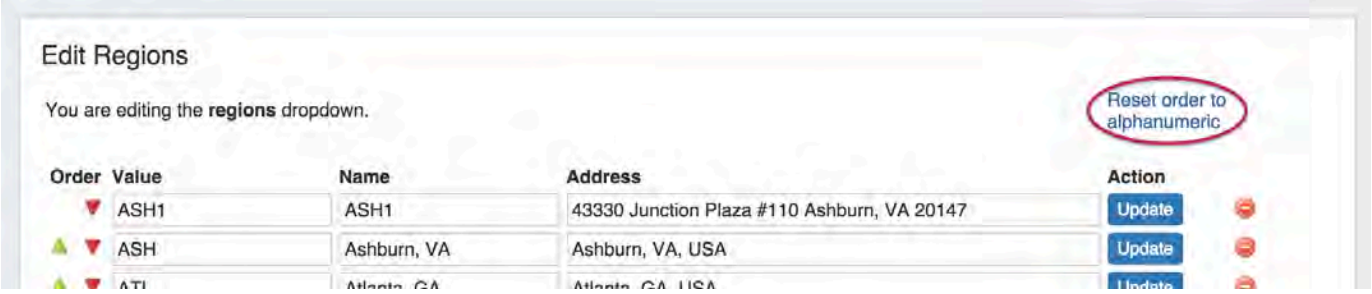


Then, click "Add Element.



The new region entry will be added to the bottom of the region list. From here, you can manually set the entry's position in the list by clicking the up/down arrows to the left of the entry.

You may also automatically sort into alphabetical order by clicking the "Reset Order to Alphanumeric" link at the top of the page.



To **edit** a region, simply type your changes in to the text box with the region name. Then, click on the "Update" Button.

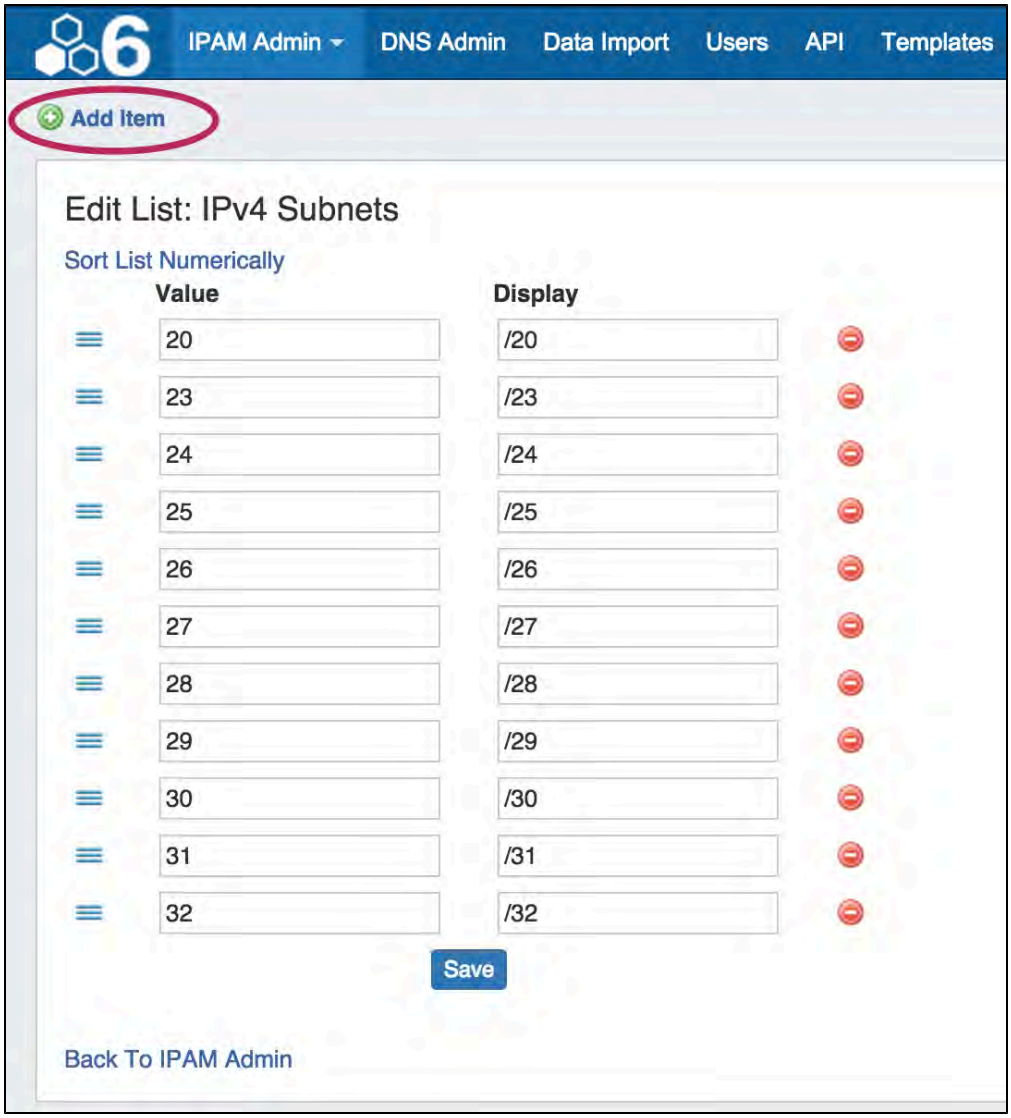
To **delete** a region, click on the red "delete" symbol to the right of the region name.

Add / Edit IPv4 / IPv6 Subnet Dropdowns

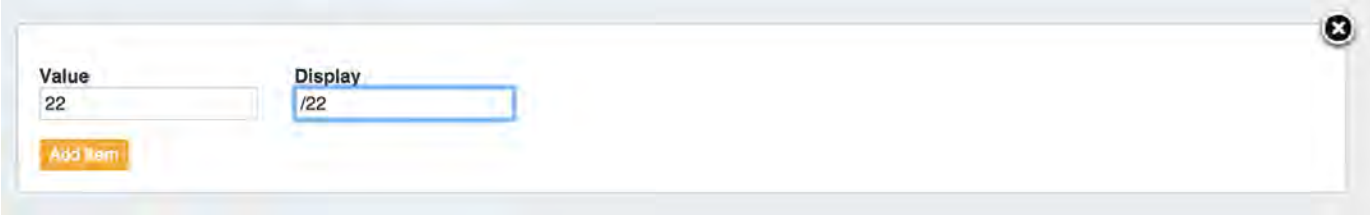
When assigning blocks using the "Smart Assign" function in the IPAM Gadget, the user has an option to assign an IP resource by allocation size. ProVision supports assignments down to a single host level (/32 for IPv4, /128 for IPv6).

> **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

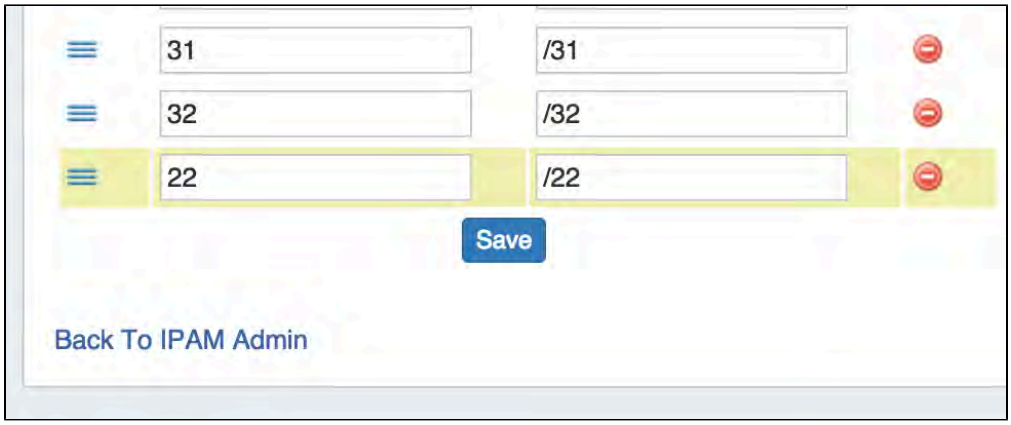
To add a new Subnet item, click on "Add Item" at the top of the Edit List: IPv4 or IPv6 Subnets page.



Then, type in the desired Value and Display value for the Subnet, and hit "Add Item".



After adding a new item, it will show at the bottom of the list highlighted in yellow. Hit "Save" to save your changes.



To **edit** a subnet, simply type your changes in to the text box. Entries with unsaved changes will be highlighted until saved.

After adding or editing a subnet, you may wish to clean up the list order by hitting "Sort List Numerically" at the top of the page to reset the list order including the new entry.

To **delete** a subnet, click on the red "delete" symbol to the right of the subnet entry.

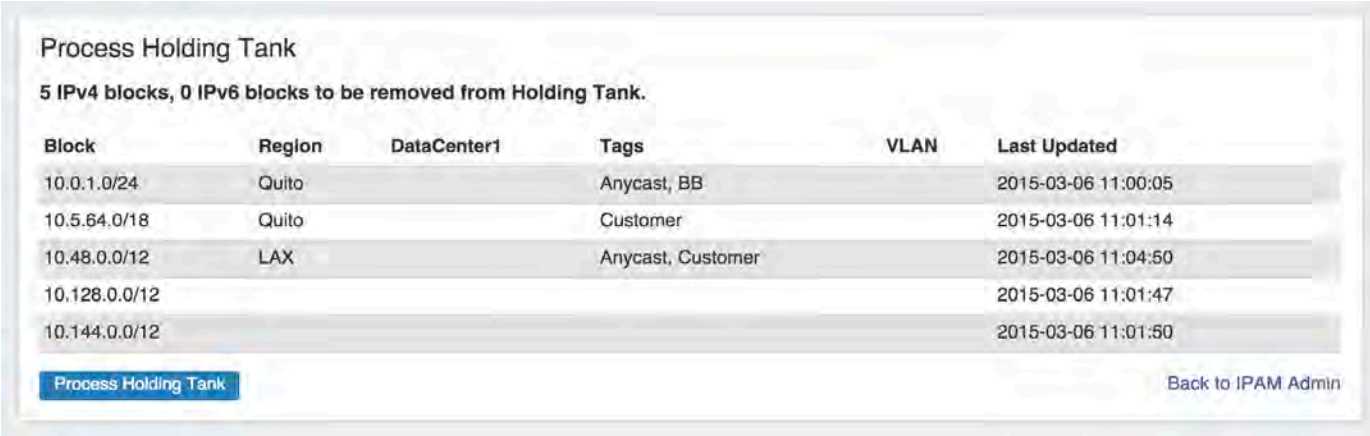
When complete, be sure to click on the "Save" button to save your changes.

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.

Holding Tank Permissions

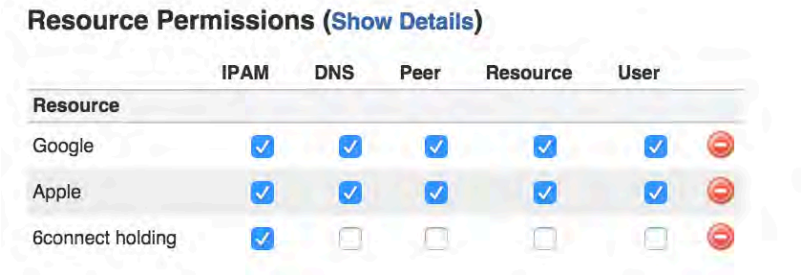
For blocks with subassignments, the Holding Tank can utilize ProVision's permissions structure in order to override holding. This allows the subbassigned block to be set as assigned to the parent resource.

Using this feature requires:

1) A block (block 1) assigned to a Resource (Resource A) that allows subassignments.

2) A block (block 2) subassigned to another Resource (Resource B) from the parent (block 1).

3) A User assigned to a Group with permissions to both Resource A and Resource B, and IPAM permissions for the 6connect Holding Resource. See Users and Groups for additional information on setting up Users and Group permissions.



Steps:

1) While logged in as the user with the above permissions, go to the IPAM manage screen for the subassigned block (block 2).

2) Select "Unassign" from the Action Menu. The 'Assigned To' column will change from showing "Resource B" to "Holding"

3) For the same block, select "Override Holding" from the Action Menu. This is the step that setting the 6connect Holding resource permissions allows, that would otherwise be unaccessible.

4) You will then see the 'Assigned To' field change from "Holding" to "Resource A", as the assignment is reverted to match the parent level.

> Override Holding through the permissions structure is intended only for blocks that are subassigned. It is not intended for blocks that are not subassigned. Setting holding permissions for non-subassigned blocks simply allows viewing of the block(s) in holding, and permissions do -not- extend to allowing overrides by design.

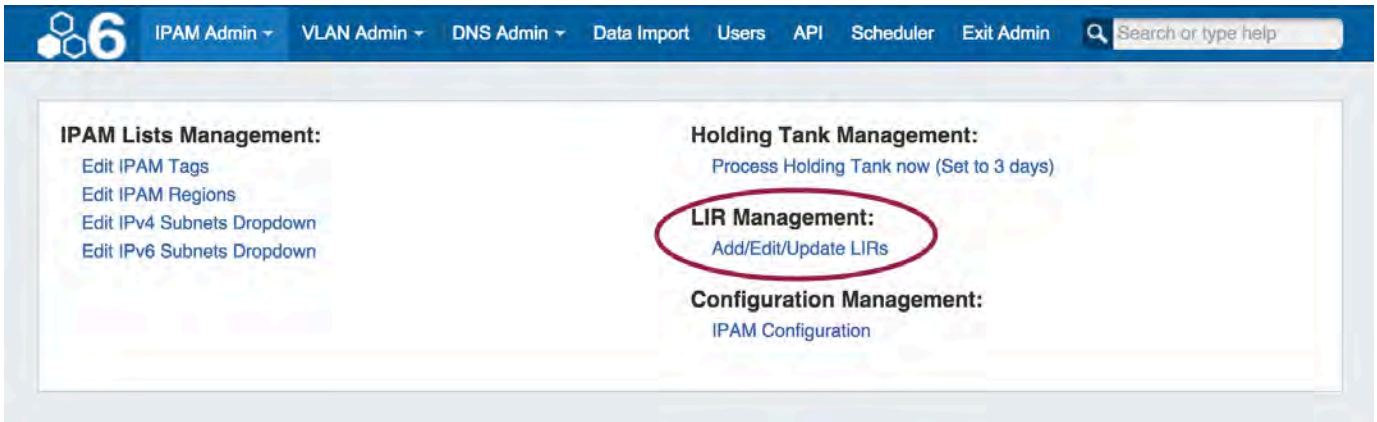
LIR Management and Use

Overview

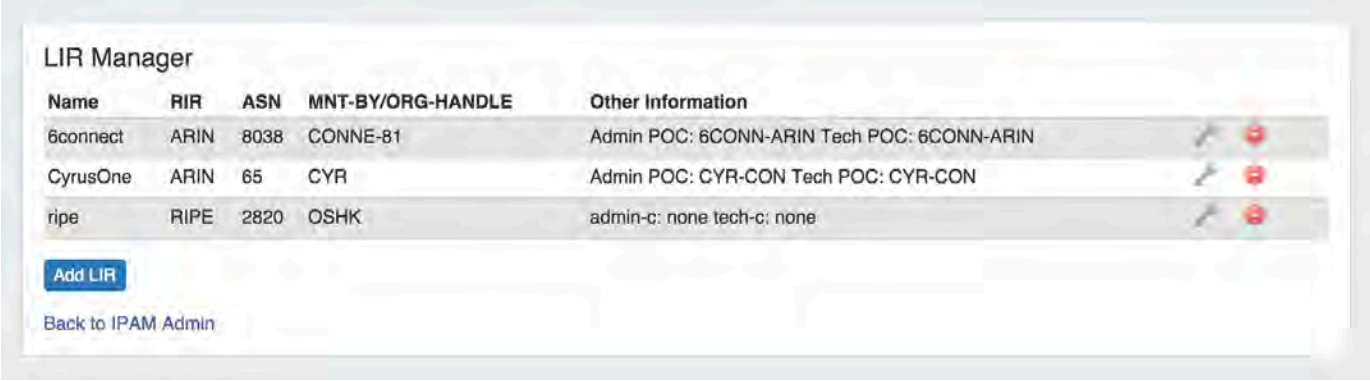
ProVision supports multiple LIRs (Local Internet Registries) in a single instance. This means that you have the ability to update SWIP/RPSL functions for a given allocation with the LIR information that you wish.

LIR Setup and Use

There is an LIR Manager available from the IPAM Admin page. In the Admin section of ProVision, select the **IPAM Admin** tab, then "LIR Management: Add / Edit / Update LIRs".

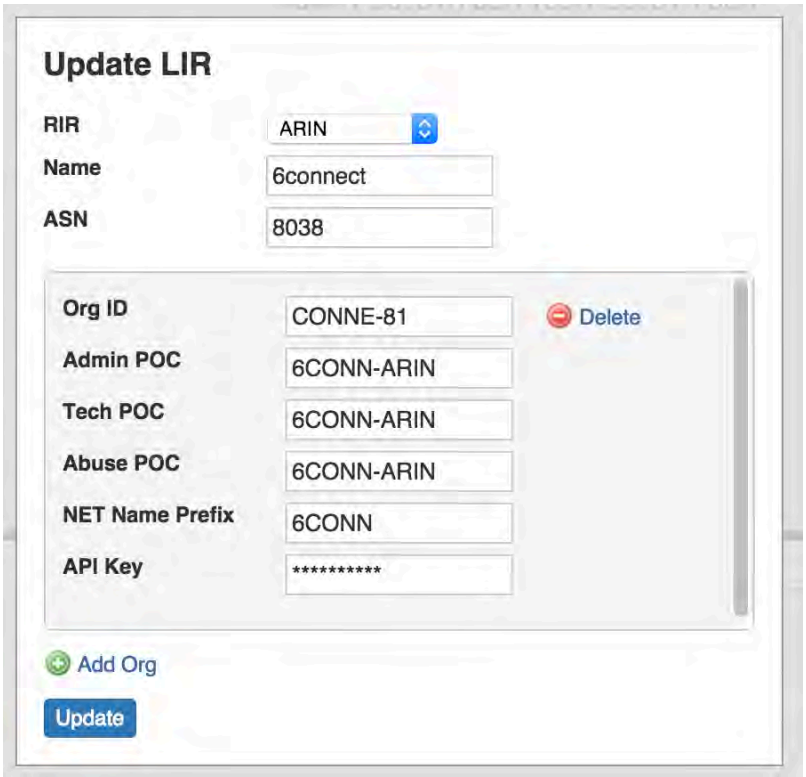


You will be taken to the LIR Manager page, showing your current LIRs. You may Add LIRs, edit the existing LIRS by hitting the Action Menu (wrench icon), or delete LIRs through the red delete icon.



Edit a LIR

Edit a LIR by clicking on the Action Menu (wrench icon) next to the entry in the LIR Manager. The Update LIR dialog will open. From here, you can edit RIR, Name, ASN, and Org information, as well as add / delete Orgs.

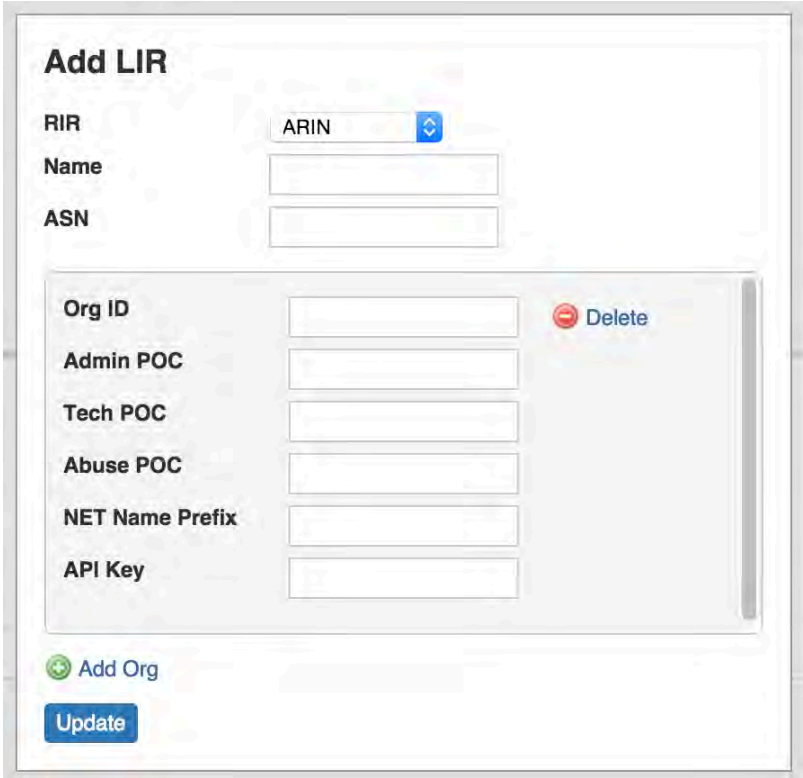


Add a LIR

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

ARIN

Adding a new LIR with ARIN selected for RIR will bring up the following fields.



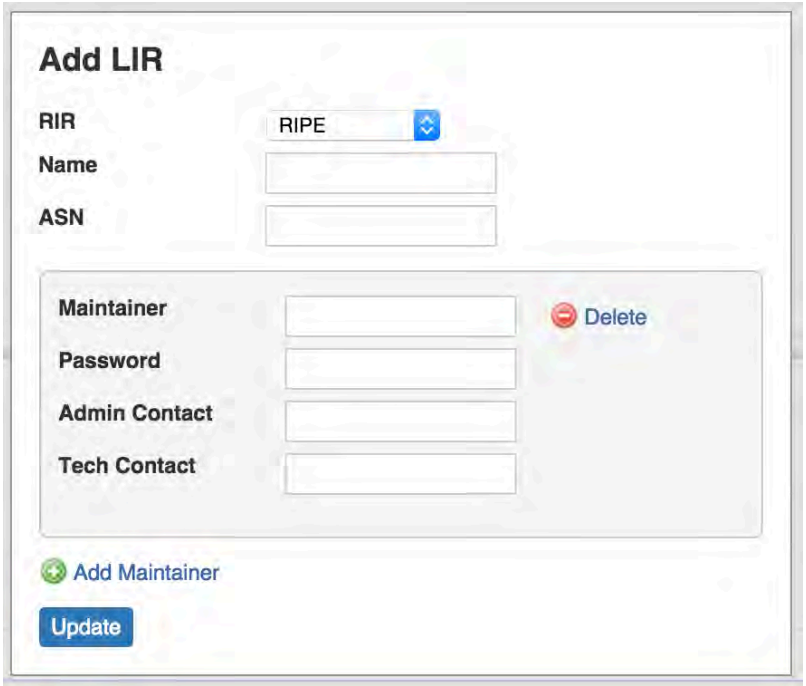
Enter the Name, ASN, ORG ID, POC information, NET Name Prefix, and API Key. Additional Orgs may be added by selecting the "Add Org" button.

Be sure to click the "Update" button when done to save your changes.

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

RIPE

Adding a new LIR with RIPE selected for RIR will bring up the following fields.



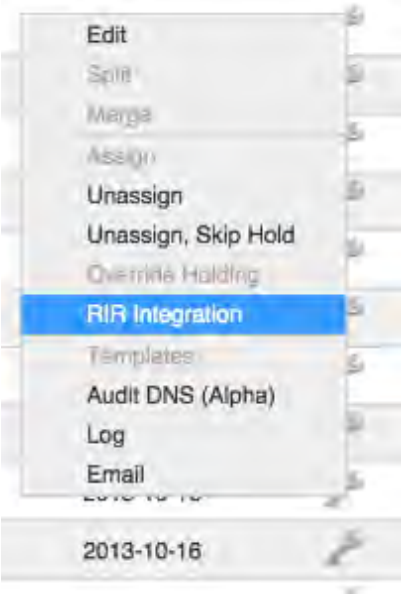
Enter the Name, ASN, Maintainer, Password, and Contact information. Additional Maintainers may be added by selecting the "Add Maintainer" button.

Be sure to click the "Update" button when done to save your changes.

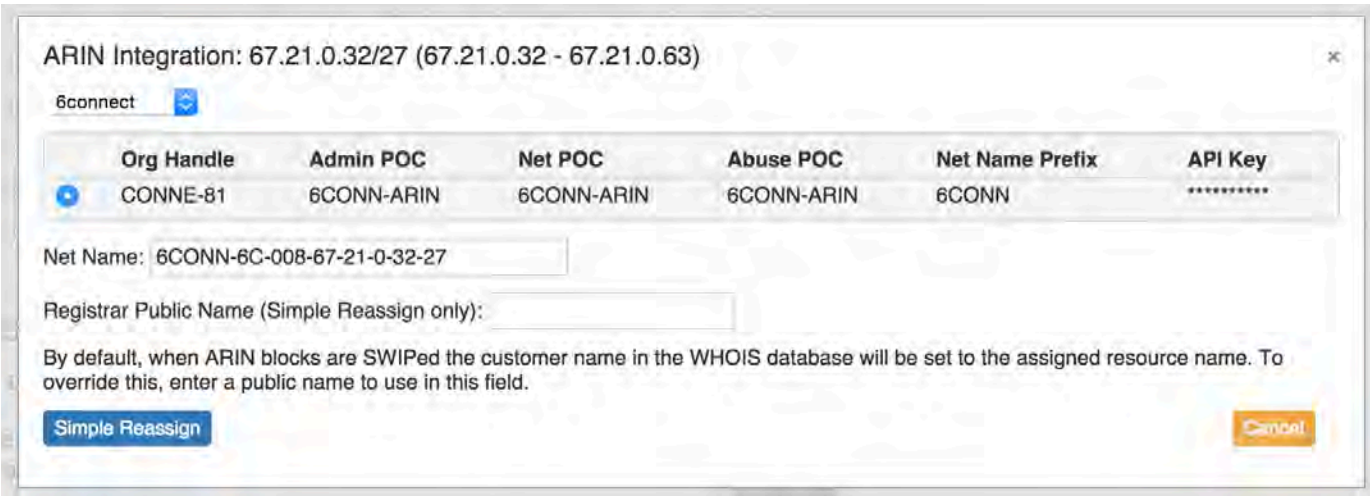
> **Press UPDATE to SAVE!**
> Make sure to press the Update button or else the LIR data will not save.

RIR Integration

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:



RIR specific options will pop up (see ARIN example below) and give the option to either "Simple Reassign" or "Cancel".



For additional detail, continue on to:

- ARIN LIR Setup and Use

- RIPE LIR Setup and Use

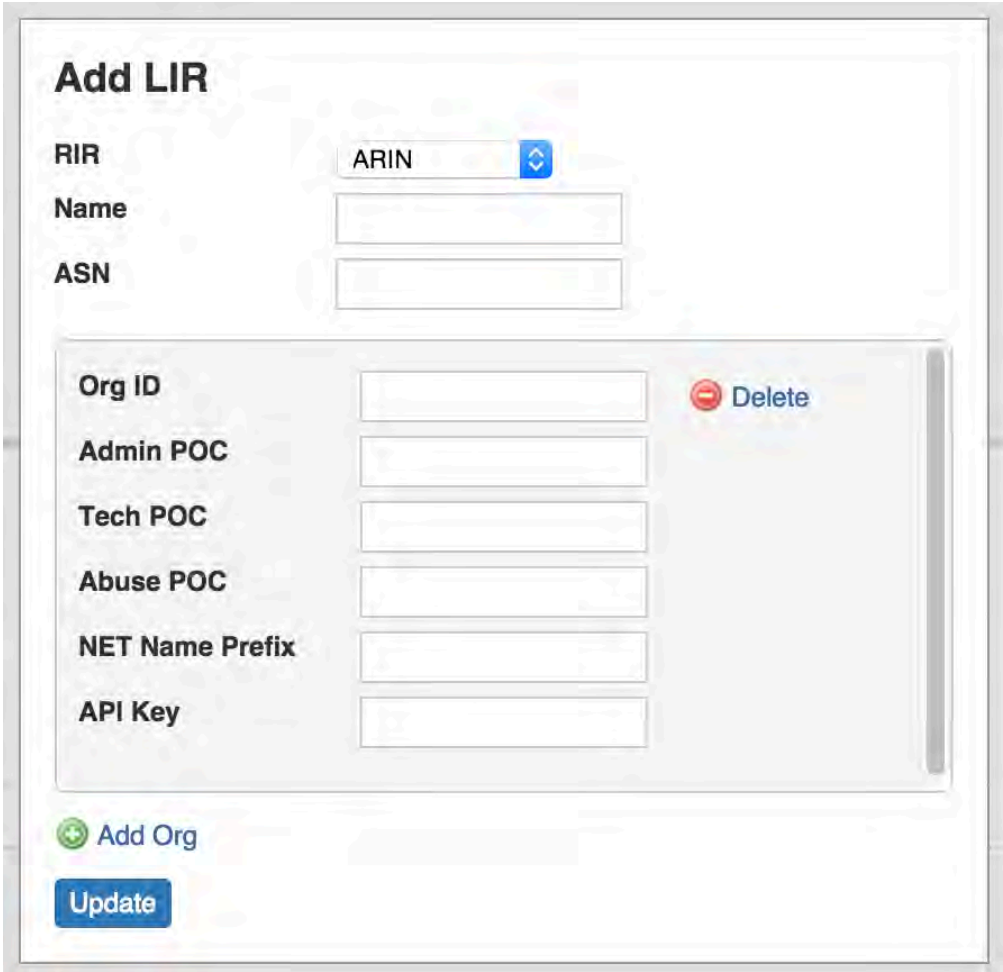
ARIN LIR Setup and Use

LIR Setup

Step 1: Setup the LIR information via the LIR Manager

You will be prompted to the select the RIR

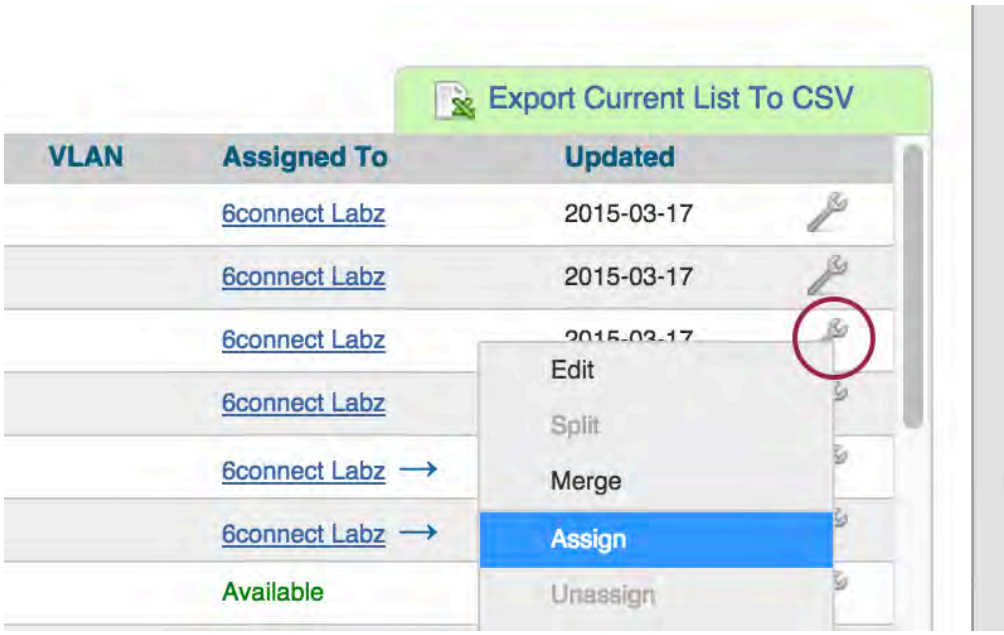
Add in the requisite Org and POC information



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.

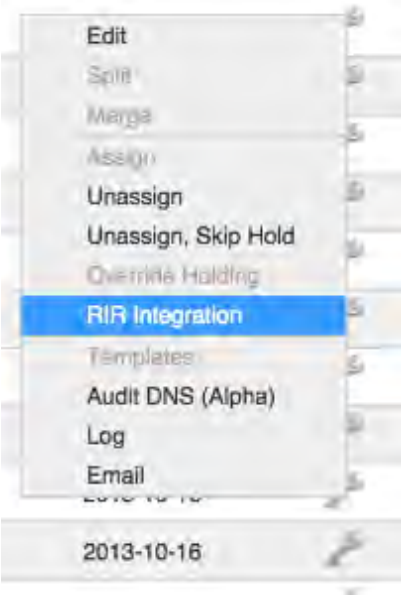
Step 2: Assign an IP block to a Resource using the **IPAM Gadget** or the Assign function from the IPAM Manage screen.

Go into IPAM Manage or the IPAM Gadget, and assign a block. See Working with IP Blocks for additional detail.

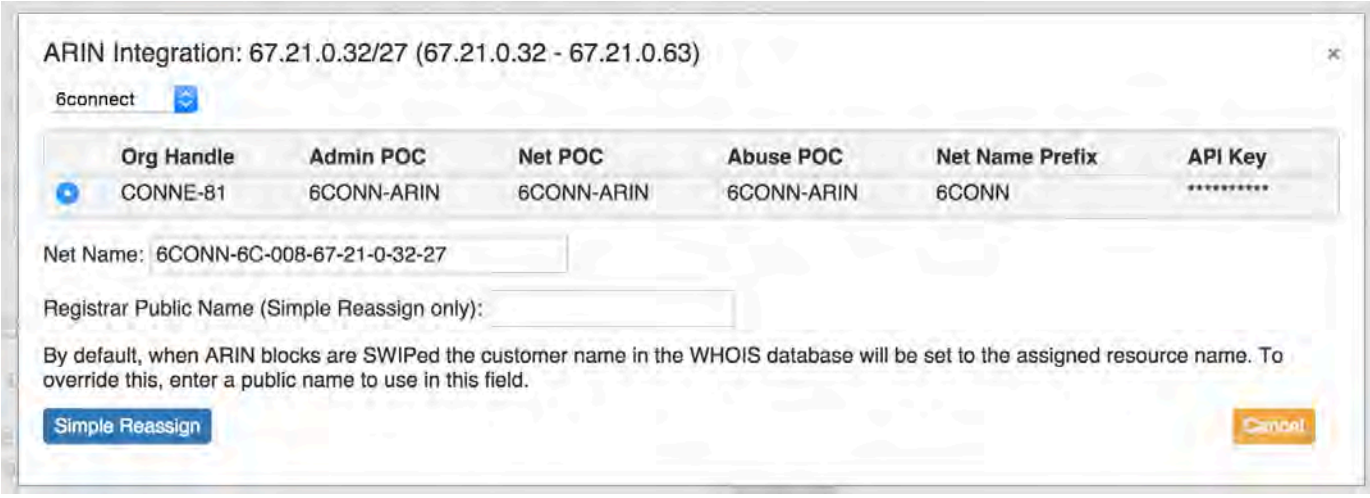


Step 3: Update SWIP information

Select "RIR Integration" from the Action Menu in IPAM Manage.



It will bring up an RIR Integration dialog, dependent on the selected RIR (ARIN example shown):



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.

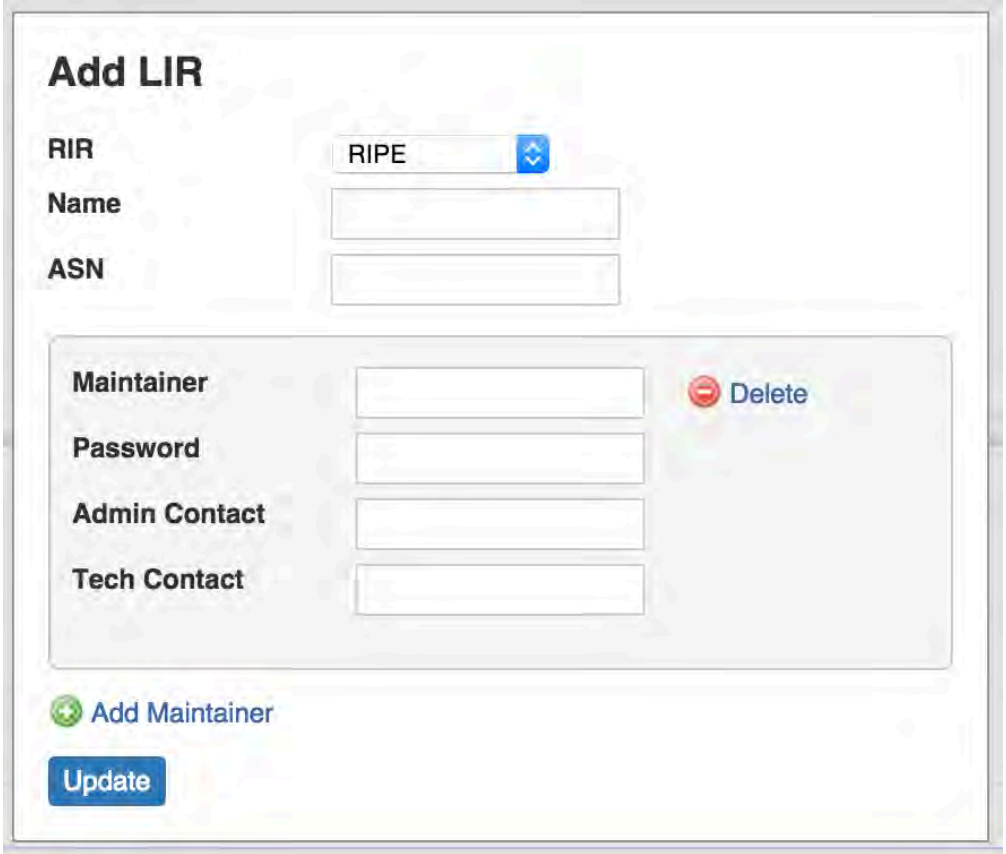
RIPE LIR Setup and Use

LIR Setup - RIPE

Step 1: Setup the LIR information via the LIR Manager

You will be prompted to the select the RIR

Then add in the requisite Maintainer Object related information:

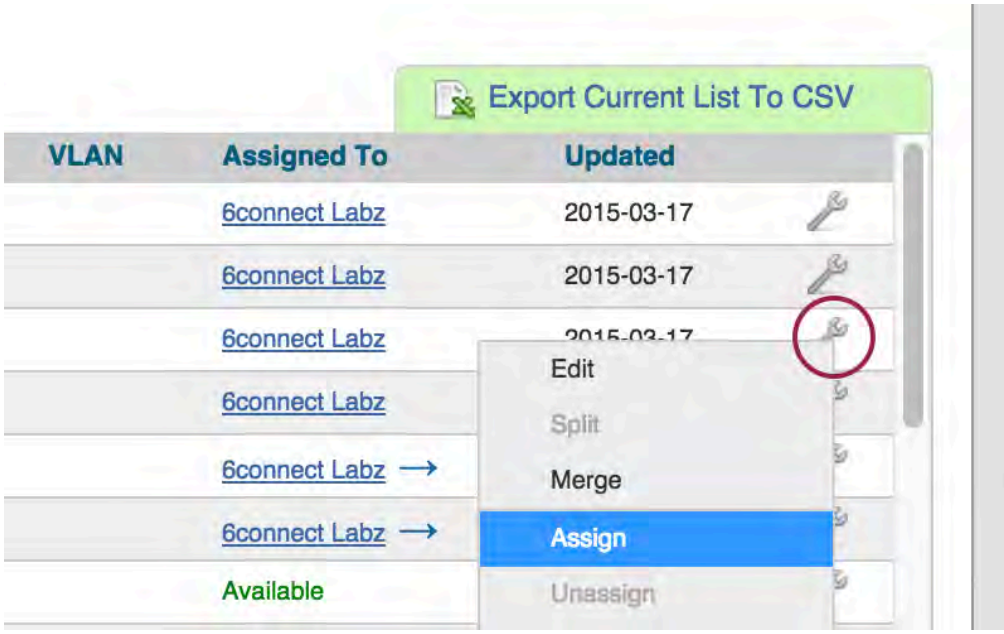


Be sure to hit "Update" when done to save your changes.

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.

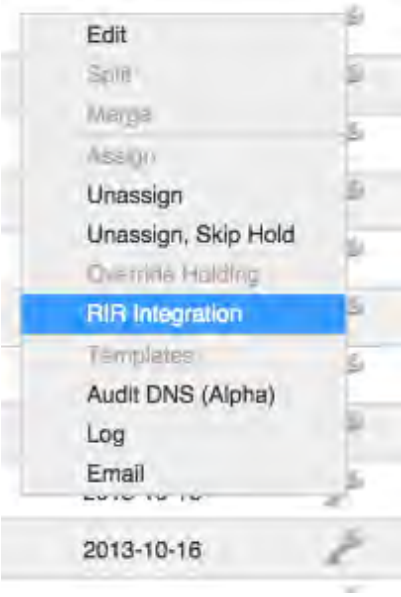
Step 2: Assign an IP block to a Resource using the **IPAM Gadget** or the Assign function from the IPAM Manage screen.

Go into IPAM Manage or the IPAM Gadget, and assign a block under the RIPE RIR. See Working with IP Blocks for additional detail.

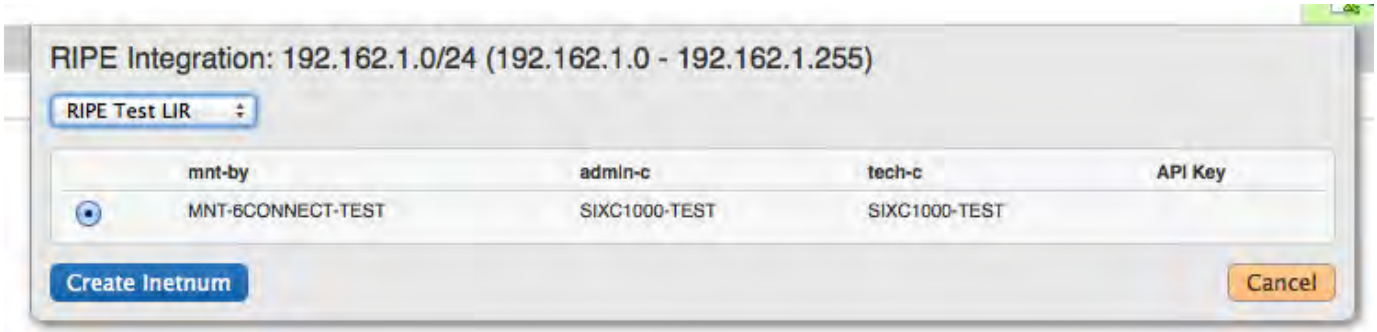


Step 3: Update RPSL information

Select "RIR Integration" from the Action Menu in IPAM Manage.

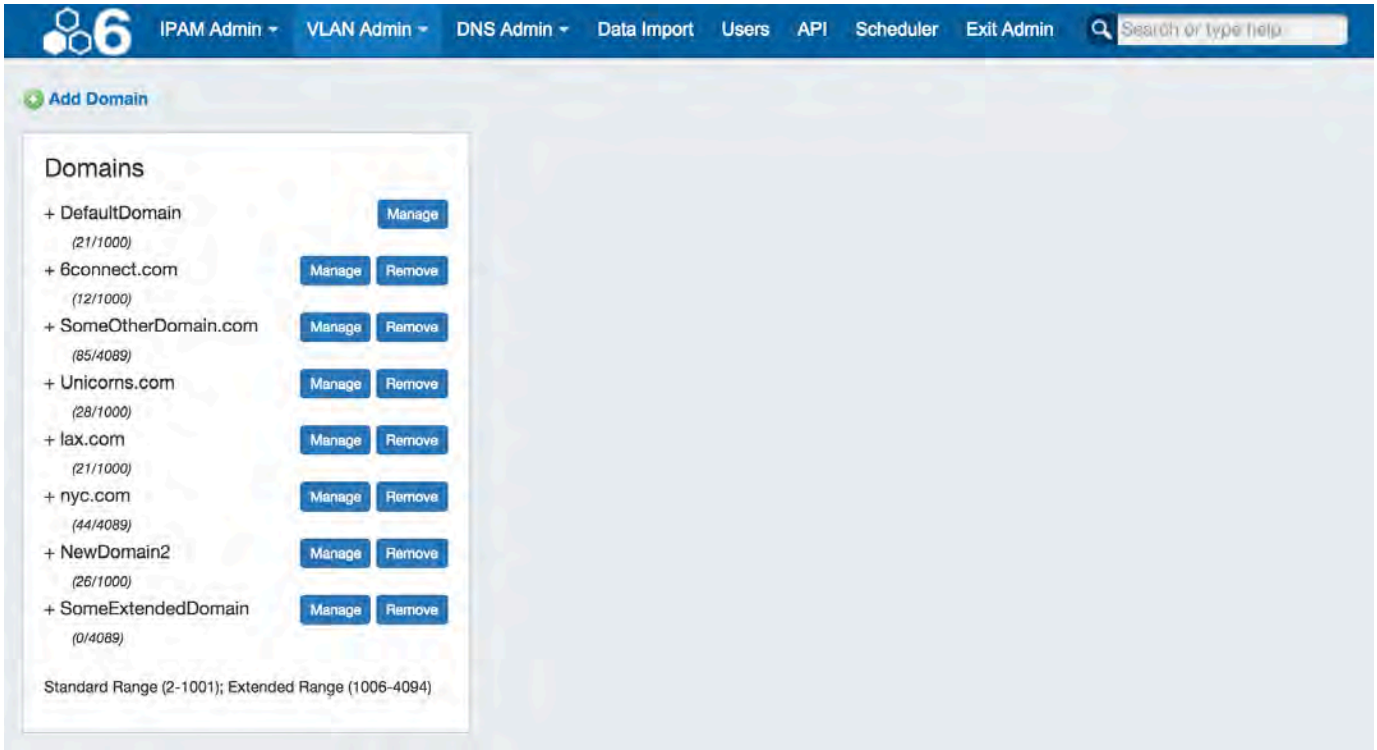


Identify which LIR data you want to use for the netnum update, and select either "Create Inetnum" or "Cancel" to exit.


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
MNT-6CONNECT-TEST
SIXC1000-TEST
SIXC1000-TEST
Create Inetnum
Cancel

VLAN Administration

VLAN Manager Overview



The VLAN Manager allows Admin users to add domains and VLANs to their ProVision instances, and associate them with IP Blocks. Optionally, VLAN tags may be associated with ranges or individual VLANs under a domain, to help limit VLAN search results when enabling VLANs. Although some VLAN functionality occurs under the "standard user" area of ProVision (IPAM Tab), only Admin-level users may access the IPAM -> VLAN page. Therefore, VLAN Manager processes are included under the ProVision Admin Guide.

VLAN Manager Workflow

Most of VLAN Manager workflow is split between the Admin area of ProVision (VLAN Admin Tab), and the IPAM Tab, under the VLAN submenu. There are two types of workflow. One is the "Standard" flow, where the VLAN tag system is not used - the only primary tasks are adding domains, VLANs, and associating the domain/VLAN pairs with IP Blocks. The second workflow is using optional VLAN tags, which requires creating the VLAN tags and associating them with domains / VLANs during the domain / VLAN add processes.

For detailed information on each of these workflows, click below:

Click here for detailed VLAN Manager Workflow information...

Standard flow (without VLAN tags):

The workflow starts with creating a domain in the VLAN Admin section of ProVision. During creation, domains may be selected as "standard" or "extended" domains, declaring the size of the VLAN pool from which VLANs are available to be enabled. Next, VLANs must be enabled and added to the domain. This is done under the IPAM Tab -> VLAN section of ProVision. Add VLANs to the domains by clicking "Add", searching for the desired range of VLANs to enable from the standard / extended pool, and selecting the desired VLANs to enable. Once enabled, VLANs may be edited or have IP blocks associated with that VLAN. Editing VLANs and Direct / Smart Browse / Search IPs functions for adding blocks to VLANs are available from the Domain/ VLAN list under under the IPAM Tab -> VLAN section of ProVision by expanding the desired domain and clicking on the VLAN link. IP blocks may also be edited individually through the IPAM gadget, IPAM Manage, and VLAN Manage areas to add domain and VLAN information to the block.

Optional flow (with VLAN tags):

The workflow starts with creating a domain in the VLAN Admin section of ProVision. During creation, domains may be selected as "standard" or "extended" domains, declaring the size of the VLAN pool from which VLANs are available to be enabled. VLAN tags may be added (from the VLAN Admin Tab -> Edit Tags submenu) before or after domain creation. Tags created in this area are available to all domains and VLANs.

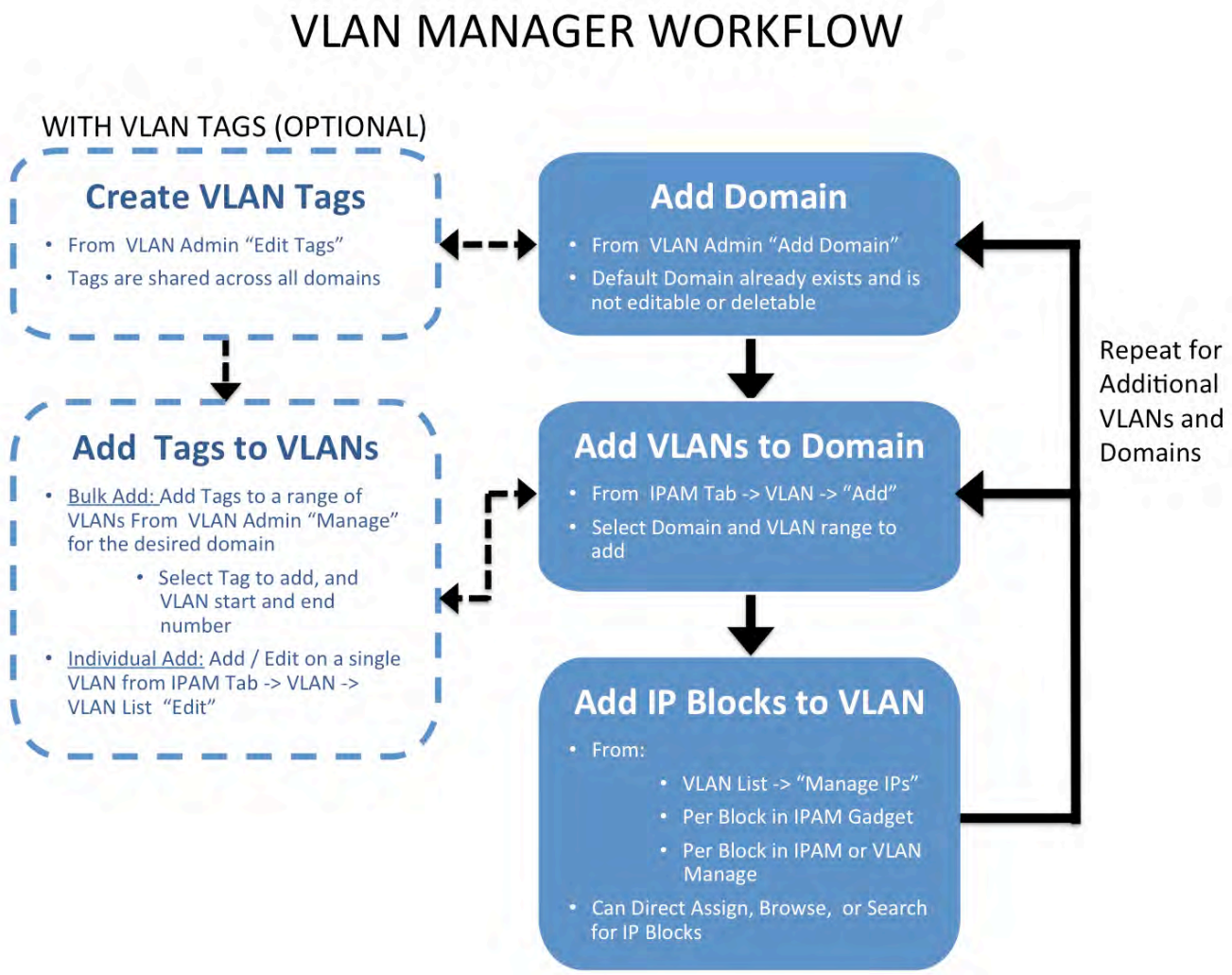
If the tags created need to be added to a large range of VLANs, the next step would be to add tags to the desired range(s) of VLANs from the

VLAN Admin -> Domain "Manage" button. Domain Manage allows you to select an existing tag, type in the numeric VLAN start and end point, and assign that tag to that entire range of VLANs under the current domain.

Once a bulk range of tags is associated with VLANs, you can use the tag as a search criteria when adding VLANs to a domain. This is done under the IPAM Tab -> VLAN section of ProVision. Add VLANs to the domains by clicking "Add", using the tag name as a search field for the desired range of VLANs, and then selecting the desired VLANs to enable.

If tags are only desired for individual VLANs, they may be added when editing a VLAN after the VLAN has already been added/enabled for the domain. Editing VLANs may be accessed from the IPAM Tab -> VLAN section, then expanding the desired domain and clicking on the VLAN link, selecting "Edit".

Next, VLANs must be enabled and added to the domain. This is done under the IPAM Tab -> VLAN section of ProVision. Add VLANs to the domains by clicking "Add", searching for the desired range of VLANs to enable from the standard / extended pool, and selecting the desired VLANs to enable. Once enabled, VLANs may be edited or have IP blocks associated with that VLAN. Editing VLANs and Direct / Smart Browse / Search IPs functions for adding blocks to VLANs are available from the Domain/ VLAN list under under the IPAM Tab -> VLAN section of ProVision. IP blocks may also be edited individually through the IPAM gadget, IPAM Manage, and VLAN Manage areas to add domain and VLAN information to the block.



Additional Information:

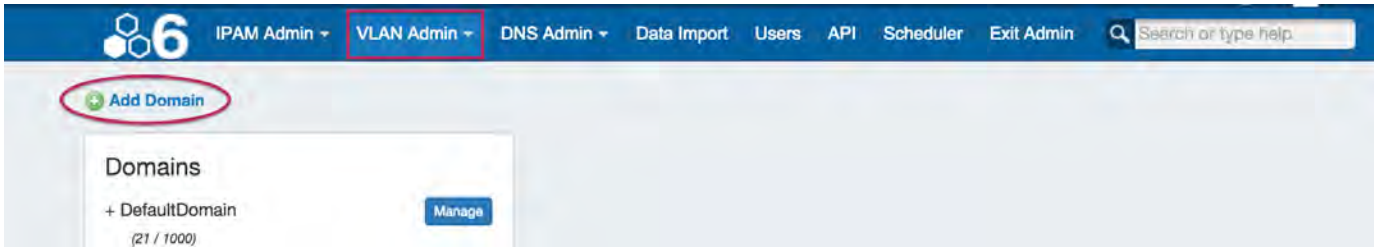
For details on performing specific tasks with the VLAN Manager, proceed to Working with the VLAN Manager.

Working with the VLAN Manager

Working with the VLAN Manager (Standard Flow)

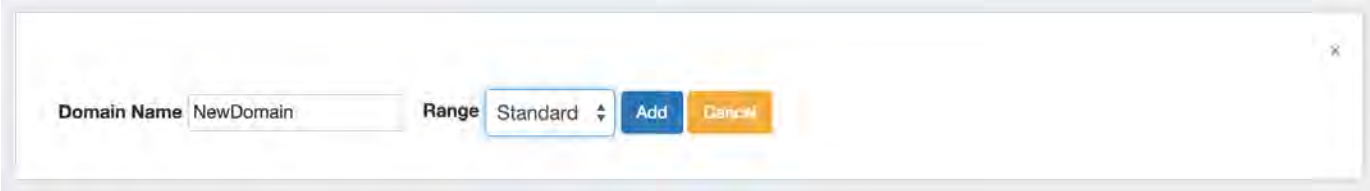
Step 1) Add Domain

Users will already have a non-editable Default Domain created, which may house VLANs entered under the previous VLAN text field, however, you may choose to create new domains. To create a new VLAN domain, navigate to the VLAN Admin Tab, in the Admin area of ProVision, and click on "Add Domain".



Enter the domain name, and select whether it is standard or extended - Standard has a range of VLAN IDs from 1-1005; Extended includes up to 4094.

> NOTE: The ProVision UI and API has restricted usage of VLANs 1 and 1002-1005 in tagging and selecting or searching VLANs.



Click "Add", and your new domain will be added to the Domain list below.

Viewing the Domain List

If a domain already has VLANs enabled for it, clicking on the "+" sign in front of the domain name will expand the domain list to show enabled VLANs.

The numbers in parentheses under the domain name (shown as 21/1000 in the above image for a standard domain; an extended domain would show as 21/4089) indicate the quantity of enabled VLANs over the available pool of VLANs to select from. There are 5 VLANs (1, 1002-1005) which are reserved, and not included in the count.

Edit a Domain

The Default Domain is not editable, however users may edit the name of user-created domains. Once a domain has been created, users may not edit the range (standard / extended) of a domain - if a domain is created with a mistaken range, it must be deleted and re-created.

To edit a domain, click on the "Manage" button next to the domain name.

Then, edit the name as desired and hit "Save".

Remove a Domain

To edit a domain, click on the "Manage" button next to the domain name.

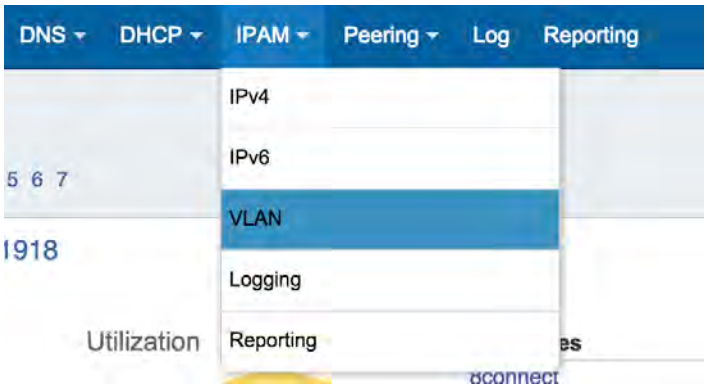
Then, edit the name as desired and hit "Save".

Step 2) Add VLANs to a Domain (Enable)

After creating domains, VLANs need to be added (enabled) for that domain out of the pool of available VLANs.

The IPAM VLAN Page

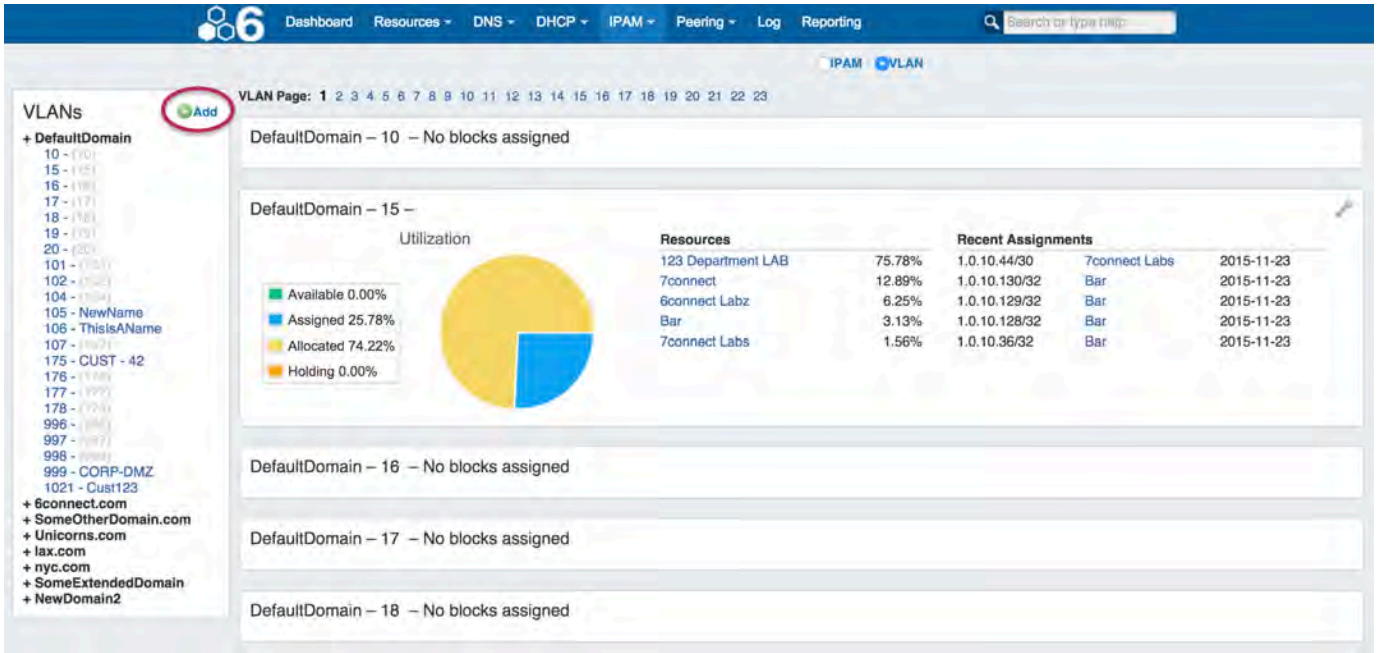
Go to the IPAM Tab, and then either select VLAN from the IPAM dropdown menu, or click "VLAN" from the radio buttons that appear on the IPAM page.



This takes you to the IPAM - VLAN page, which shows an overview of Domains and VLANs on the left sidebar (click the + next to the domain names to see the enabled VLANs under each domain), and similar to the IPAM page, shows IPAM Utilization data for each Domain / VLAN pair. Clicking on the wrench will open "VLAN Manage", which has the same type of functionality for IP blocks as IPAM Manage, but organized under the scope of that VLAN.

Add VLANs to a Domain (Enable)

In the IPAM - VLAN page VLAN sidebar, click "Add".

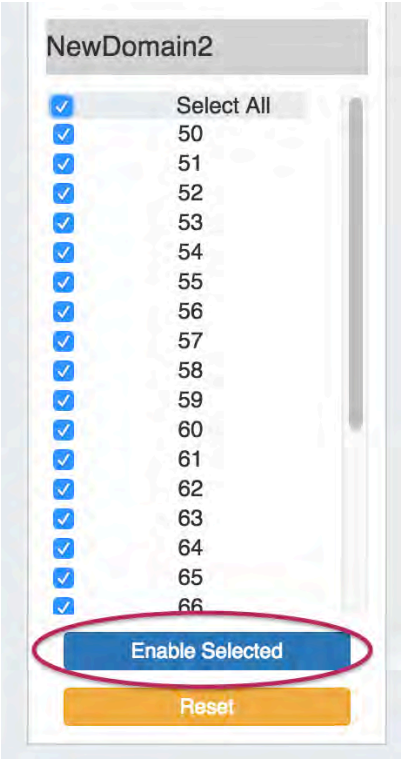


This changes the sidebar into a VLAN search module. Select the desired domain, range of VLANs to view, and click "Search".

> Note: If VLAN Tags have already been applied to a range of VLANs, you may also include VLAN tag criteria in your search. If tags have not been created or applied to a range of VLANs, searching by tags will not return results.

Below the search criteria, the results list will show VLANs meeting your criteria.

Select the desired VLANs to enable for the domain - you may either select all VLANs in your results list, or check VLANs individually.

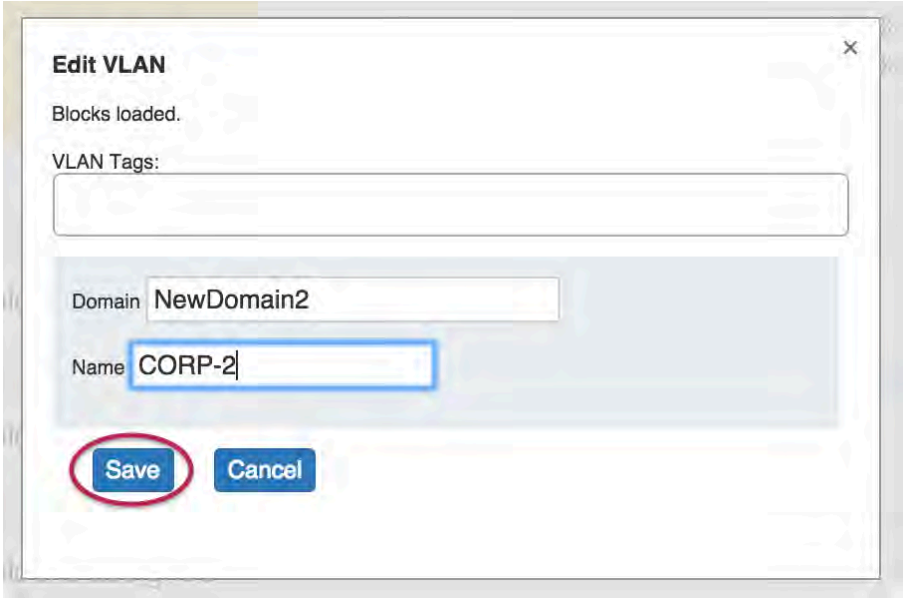


When done selecting the VLANs, click "Enable Selected VLANs". The selected VLANs will now be enabled for the domain, show in Domain / VLAN lists, and be available to edit or add blocks to. To add more VLANs to another domain, click "Reset" and repeat the process. If done, hit the "Cancel" button at the top of the Add VLAN module, and you will return to the main IPAM - VLAN page view.

Edit a VLAN

You can edit a VLAN by clicking on the VLAN in the VLAN List sidebar, then selecting "Edit".

You cannot change the Domain of the VLAN, but you can add a descriptive text name, or apply a VLAN tag (if using the tag system).



After making your changes, click the "Save" button. The named VLAN will now show both VLAN number and name when viewing Domain / VLAN lists or when working with blocks.

Step 3) Add Blocks to VLANs

After VLANs have been enabled for domains, you can associate those VLANs with IP blocks.

Add / Update Blocks to VLAN from VLAN - Manage IPs:

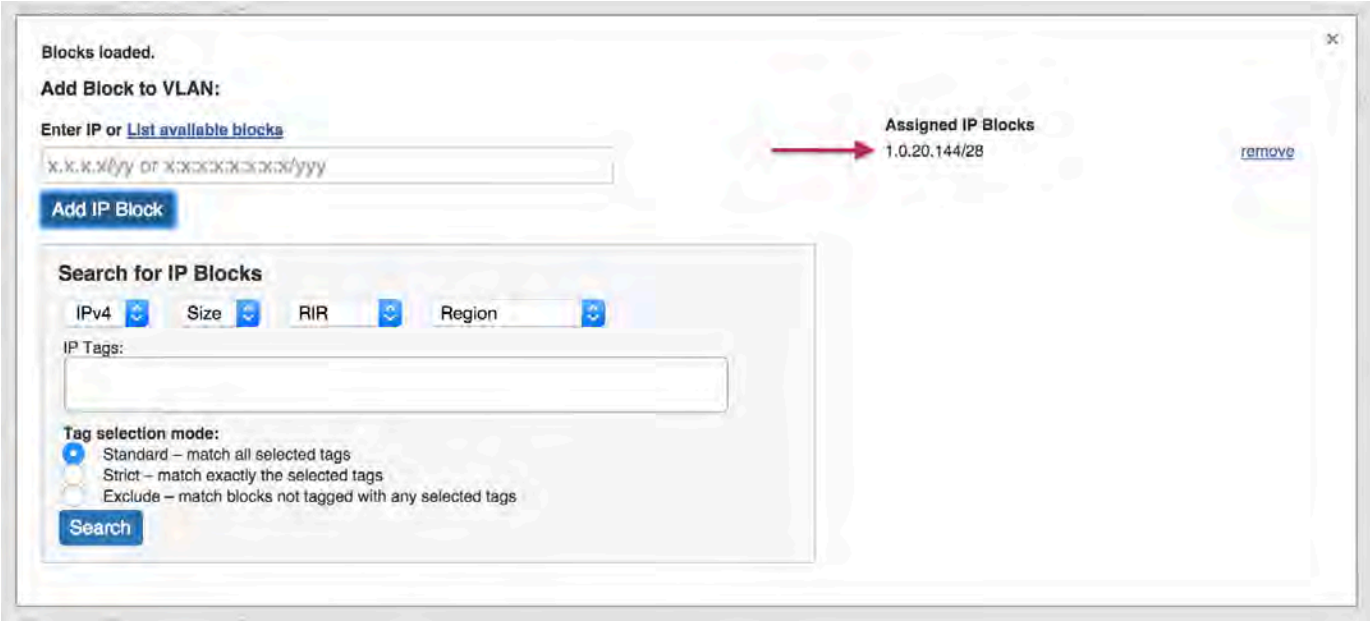
While in the IPAM Tab -> VLAN Page, expand the desired domain and click on a VLAN. Then select "Manage IPs".

This opens up a window where IP Blocks may be associated to the VLAN. From here, you can "Direct Assign" a block to the VLAN, "Browse Assign" blocks, or Search for a block to add to the VLAN.

Direct Add Block to VLAN:

Directly add a known, available IP Block by typing in the CIDR, then hitting "Add IP Block".

When complete, the block will be added to the "Assigned IP Blocks" list on the right of the window, with the option to remove if desired.

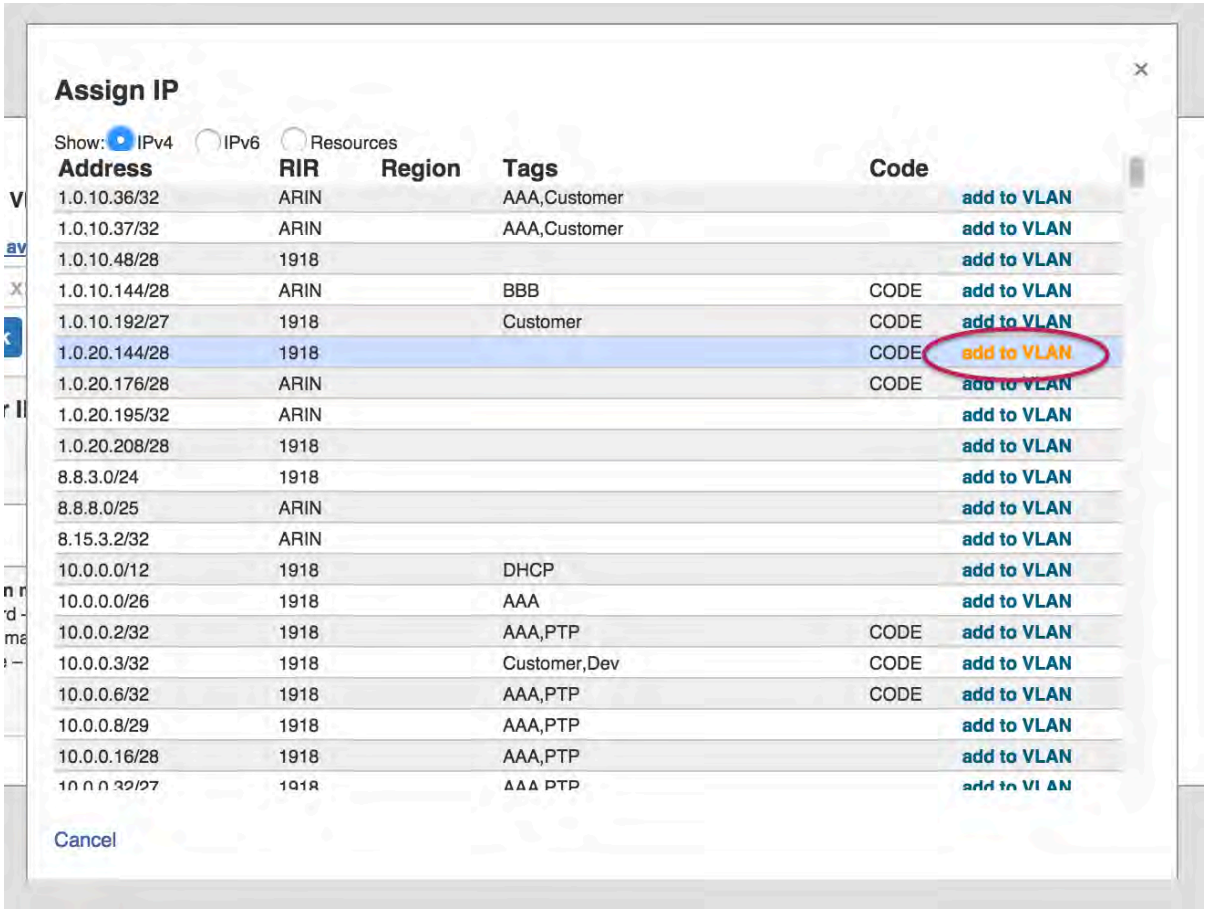


Browse IP Blocks to add to VLAN:

Browse blocks available to add to the VLAN by clicking on the "List available blocks" link above the IP block input box.

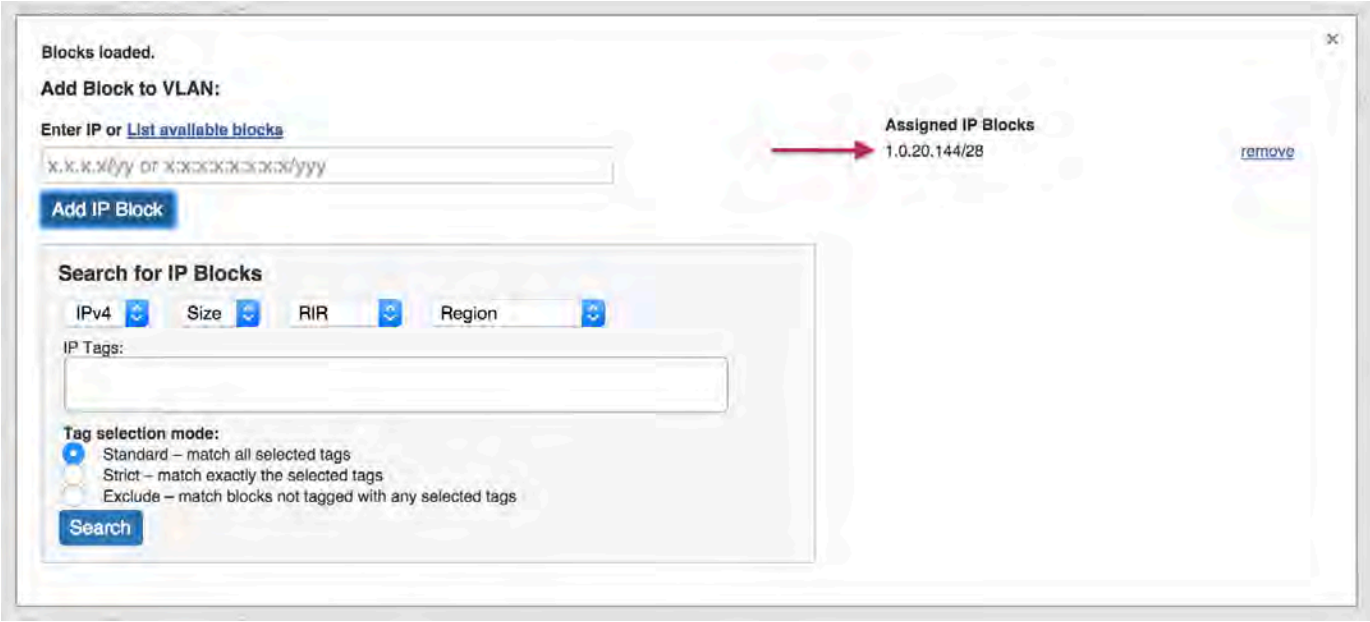
This opens a window similar to the "Smart Browse" used in assigning IPs to a resource - just select IPv4, IPv6 blocks, or Resources to browse from the radio buttons at the top of the list.

When you have selected the block to add, click "add to VLAN".



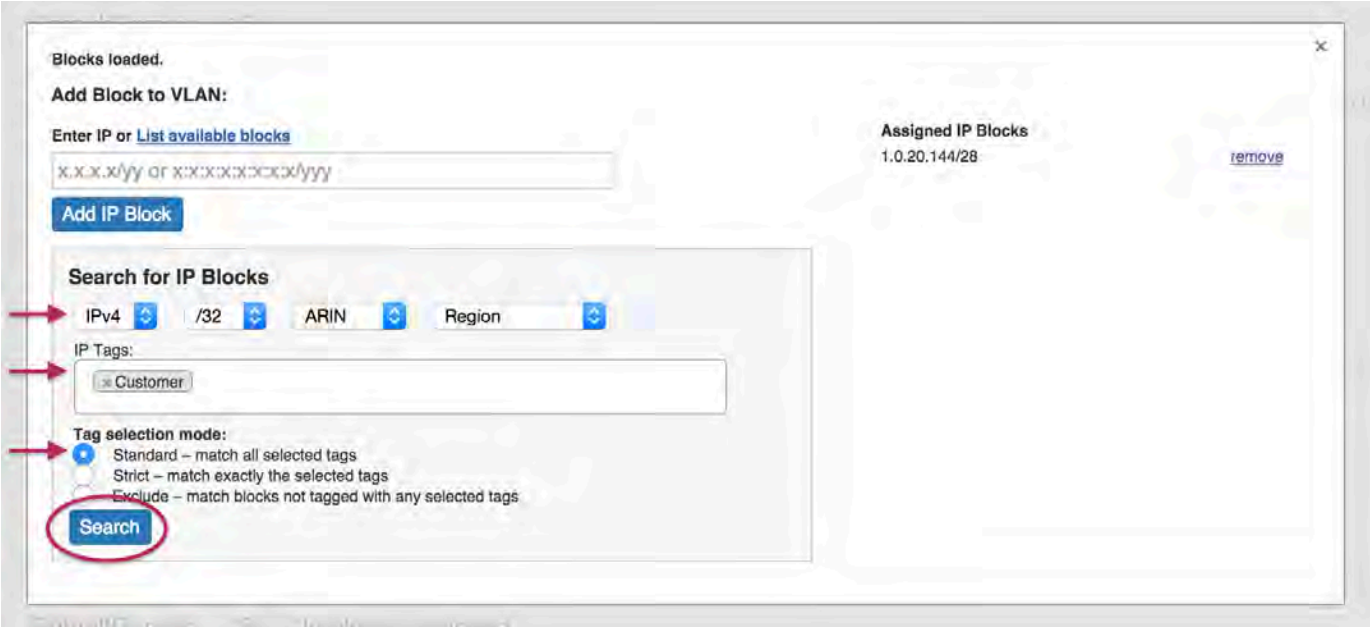
The selected block will automatically fill in the block input box - just hit "Add to IP Block" below the input to finalize the addition.

When complete, the block will be added to the "Assigned IP Blocks" list on the right of the window, with the option to remove if desired.



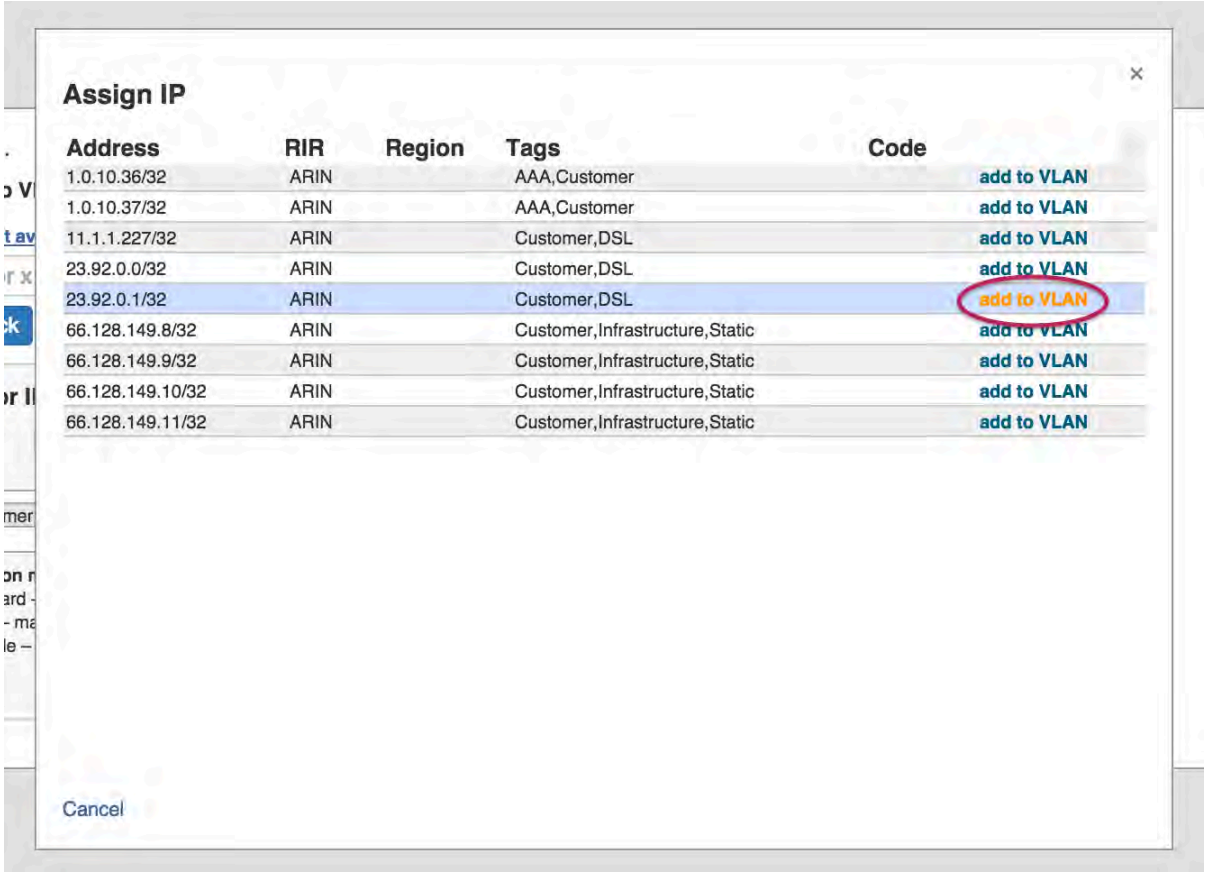
Search IP Blocks to add to VLAN:

Select search criteria of IPv4 / IPv6, size, RIR, Region, or Associated Tags / Tag selection mode to create a filtered list of IP blocks meeting that criteria. Then, click "Search".



This opens a window with filtered results matching your criteria.

When you have selected the block to add, click "add to VLAN".



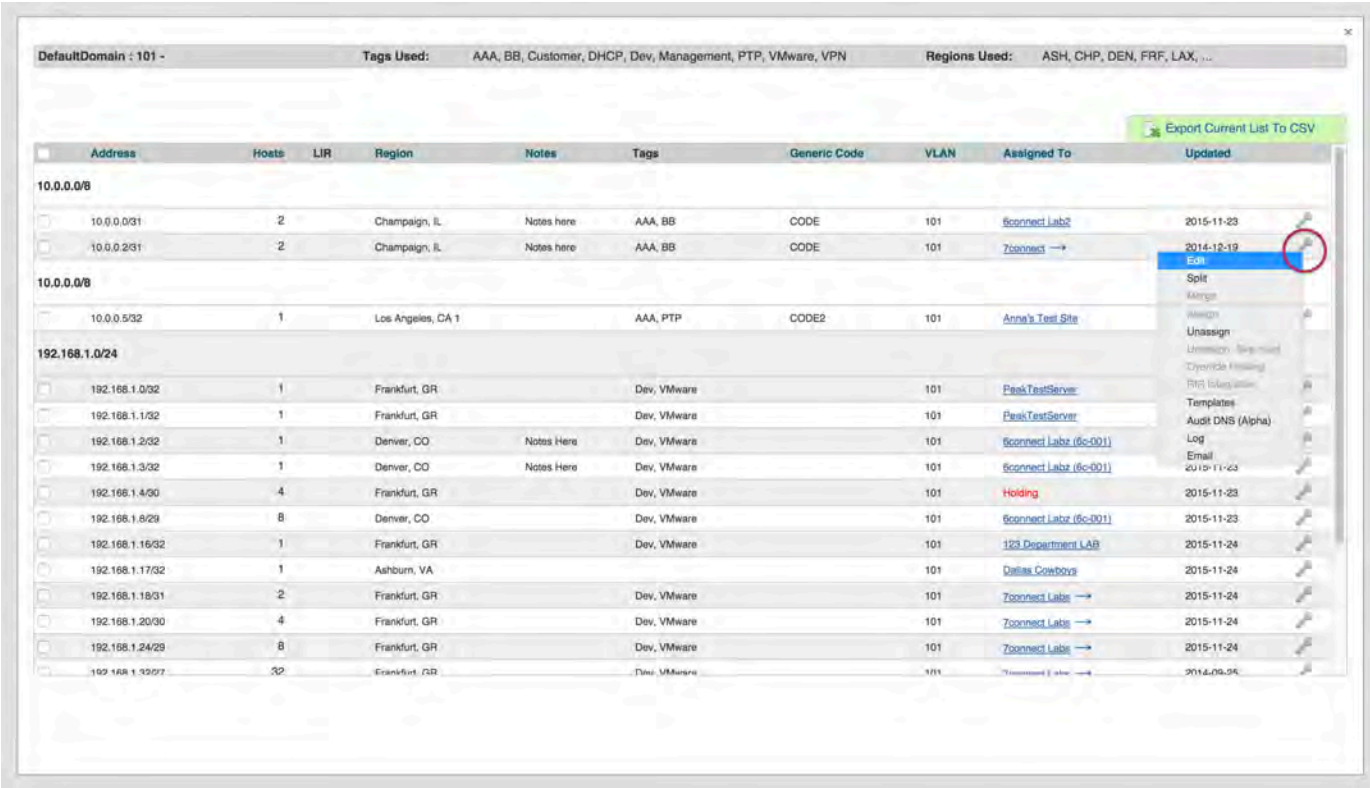
The selected block will automatically fill in the block input box - just hit "Add to IP Block" below the input to finalize the addition.

When complete, the block will be added to the "Assigned IP Blocks" list on the right of the window, with the option to remove if desired.

Add / Update VLAN to Blocks from IP Manage or VLAN Manage:

When editing a block in either IP Manage (IPAM Tab -> Action Menu (wrench) -> Manage), or VLAN Manage (IPAM Tab - VLAN -> Action Menu (wrench)) , you can select an existing domain and VLAN to associate to the block.

From either of the Manage screens, click on the Action Menu (wrench icon) for the desired block, the select "Edit".



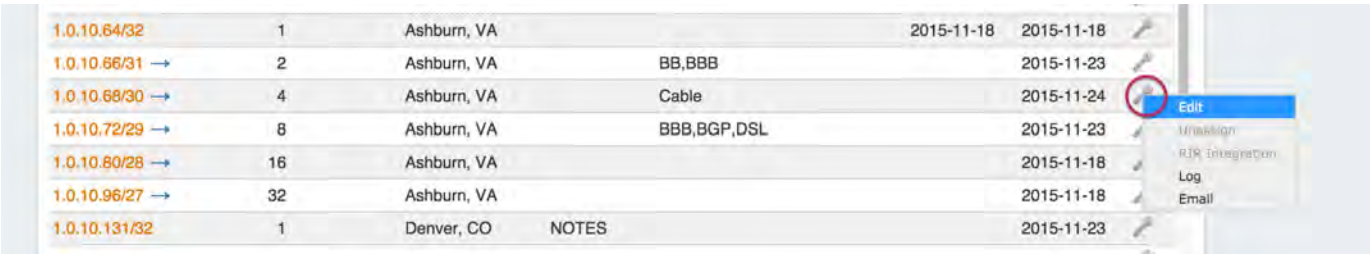
In the Edit Attributes page for the block, change the Domain to the desired item, then select the VLAN. Hit save when done.

VLANs and Child Blocks
NOTE: Child blocks whose Parent blocks already have an assigned Domain / VLAN, may not have domains and VLANs changed to be different than the Parent Block's values. If you attempt to save domain / VLAN changes to a child block under this situation, an error will occur.

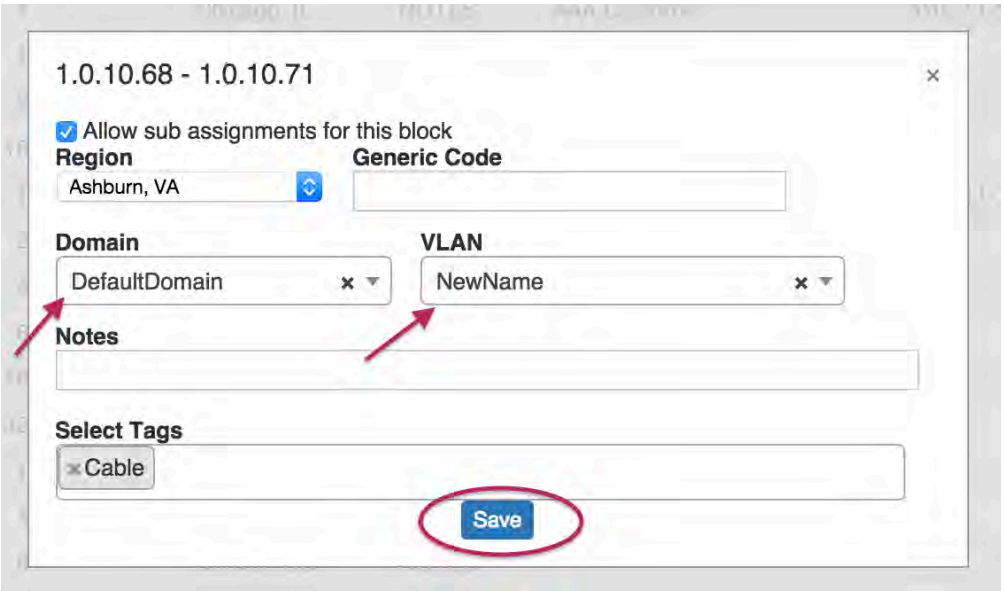
i.e., Parent and Child IP Blocks must have matching Domain / VLAN values.

Add / Update VLAN to Blocks from IPAM Gadget:

You may also add / edit Domain /VLAN information when editing a block in the IPAM Gadget. From the IP block's Action Menu (wrench icon), select "Edit".



Then, in the Edit Attributes menu that pops up, select the desired Domain / VLAN information, and click "save".

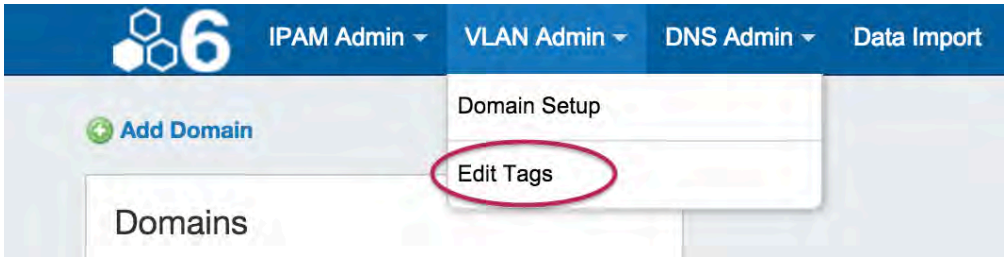


Working with VLAN Tags - OPTIONAL

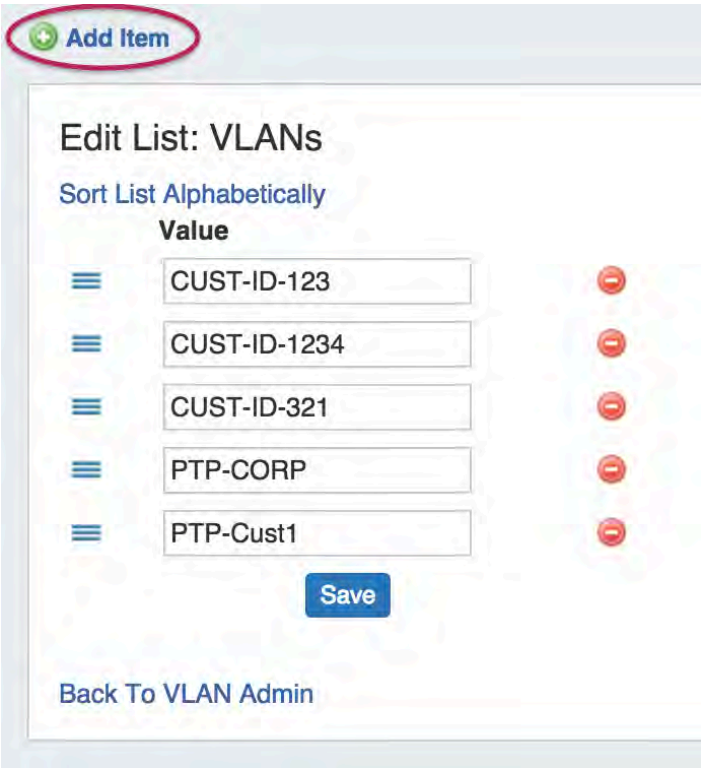
VLAN tags provide further organizational / search criteria when searching for VLANs to enable.

Creating VLAN Tags

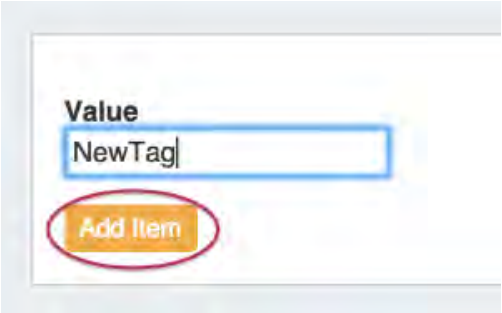
Create VLAN Tags from the VLAN Admin Tab, in the Admin area of ProVision, and select "Edit Tags" from the dropdown.



To add a new VLAN tag, click on "Add Item" at the top of the Edit List: VLAN Tags page.

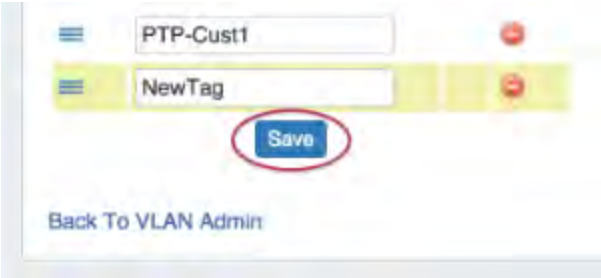


Then, type in the desired name value for the new tag, and hit "Add Item". The new tag will be added to the list below.



To **edit** a tag, simply type your changes in to the text box with the tag name. Tags with unsaved changes will be highlighted until saved.

To **delete** a tag, click on the red "delete" symbol to the right of the tag name.



When complete, be sure to click on the "Save" button to save your changes.

Once VLAN tags have been created, they will be available to add to VLAN ranges and use in the "Add VLAN" search function.

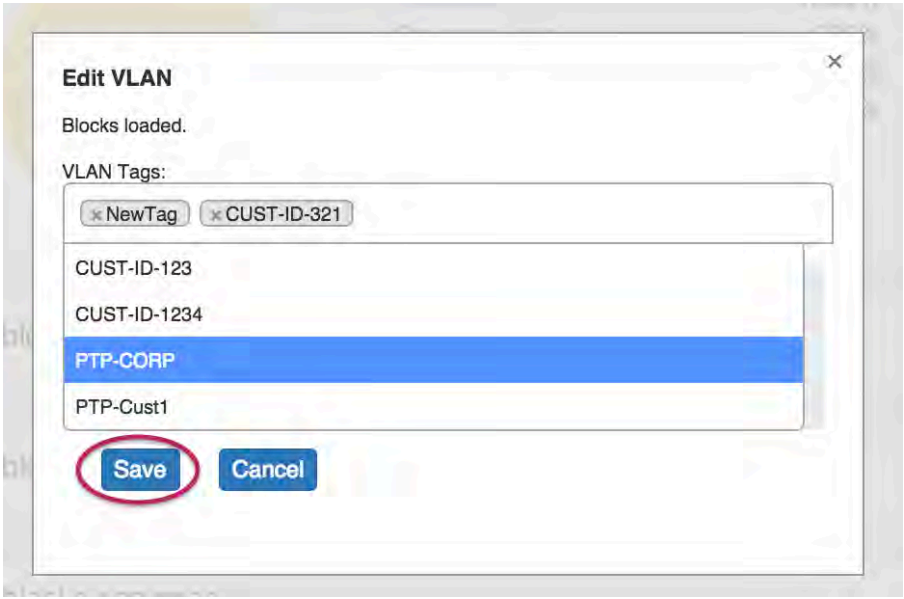
Add VLAN Tags to VLANs

VLAN tags may be added to VLANs in two ways: Editing an already-enabled VLAN (VLAN Edit), or by assigning the tag to a range of VLANs from the Domain VLAN pool (un-enabled VLANs, which are then available to search by VLAN tag to enable).

Add Tag to an Enabled VLAN

Edit an existing, already enabled VLAN by clicking on the VLAN in the VLAN List sidebar, then selecting "Edit".

Apply a VLAN tag from clicking in the "VLAN Tags" box, and selecting one or more VLAN Tags.



After making your changes, click the "Save" button. Tagged VLANs will be viewable in the VLAN Chart a single tag points under VLAN Admin - Domain Manage.

Add Tag to a Range of the VLAN Pool

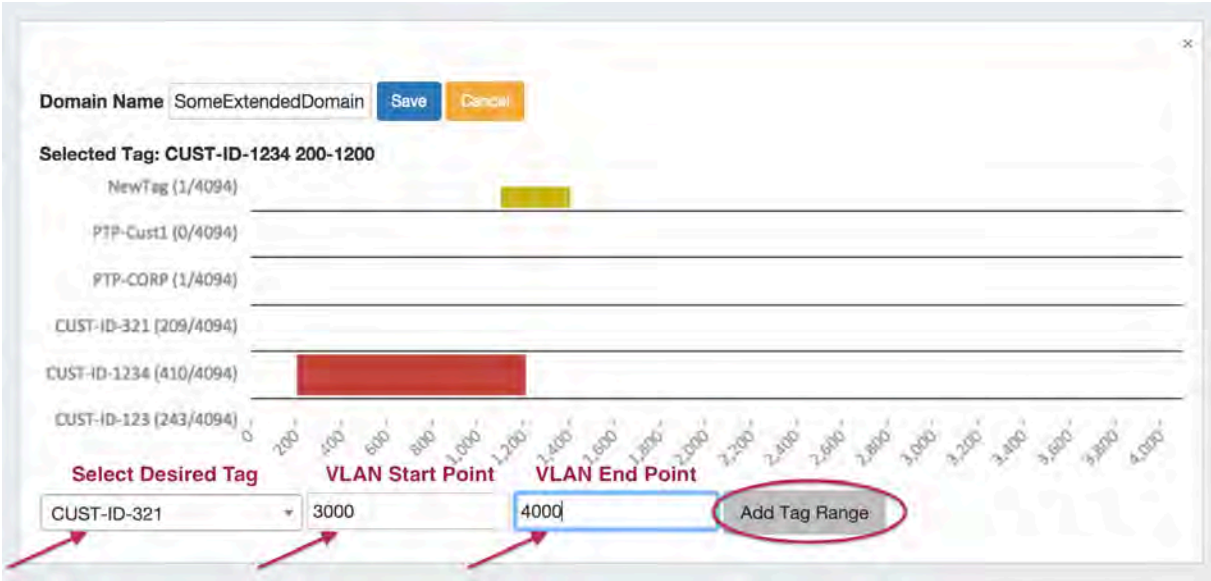
You can also tag a range of the (un-enabled) VLANs in a domain's VLAN pool. Then, when moving on to enable a set of VLANs for a Domain, you can select those VLANs by tag.

Go to the VLAN Admin Tab in the Admin area of ProVision. Then, next to the desired domain, click "Manage".

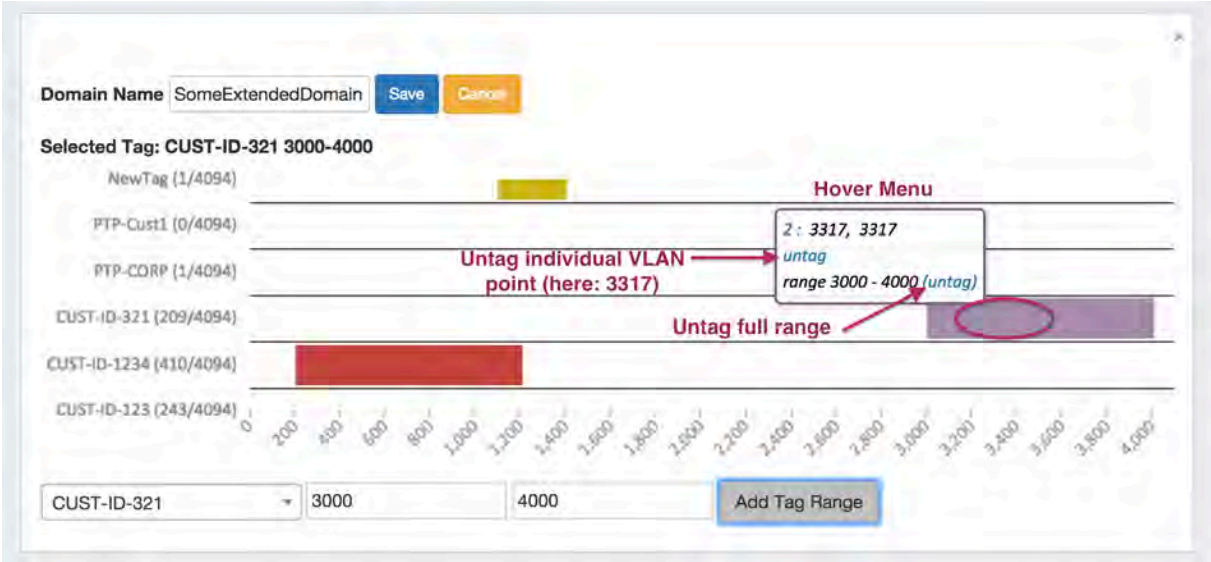
This brings up the page to edit the domain or view a chart of VLAN tags (y axis) to VLANs in the available pool (x axis).

Bulk assign tags to ranges by:

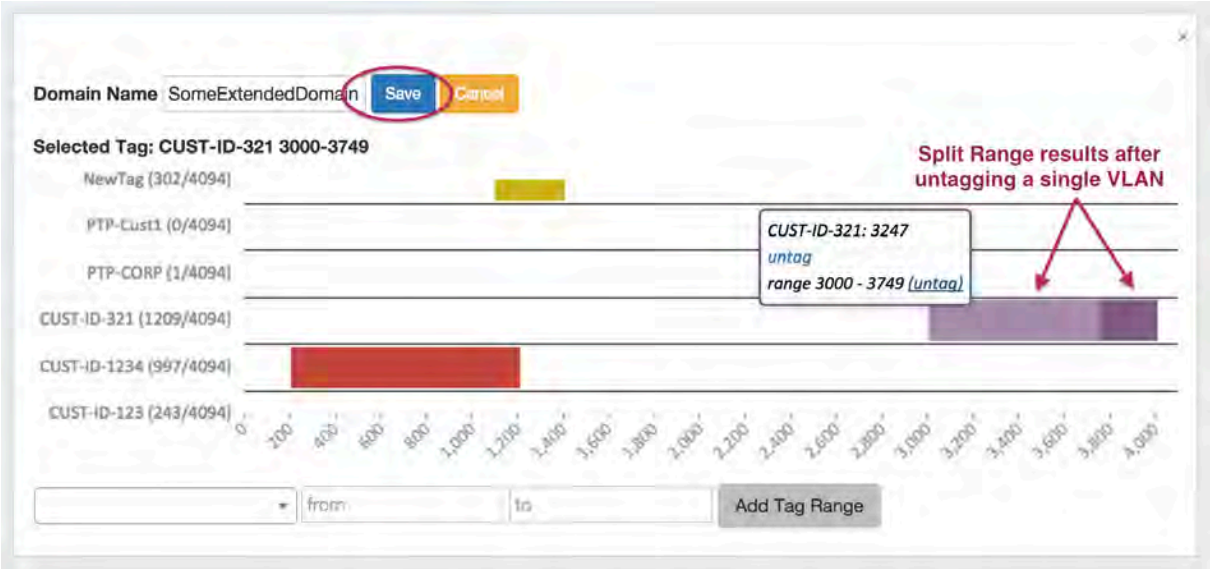
1. Select tag to assign in dropdown below chart
2. Type desired VLAN start and end for range
3. Click "Add Tag Range" to set and save.



4. Hovering over the range in the chart will bring up a menu to allow for untagging individual VLAN points or the full range



5. Untagging a single point will split the tag range into two separate ranges that behave independly.

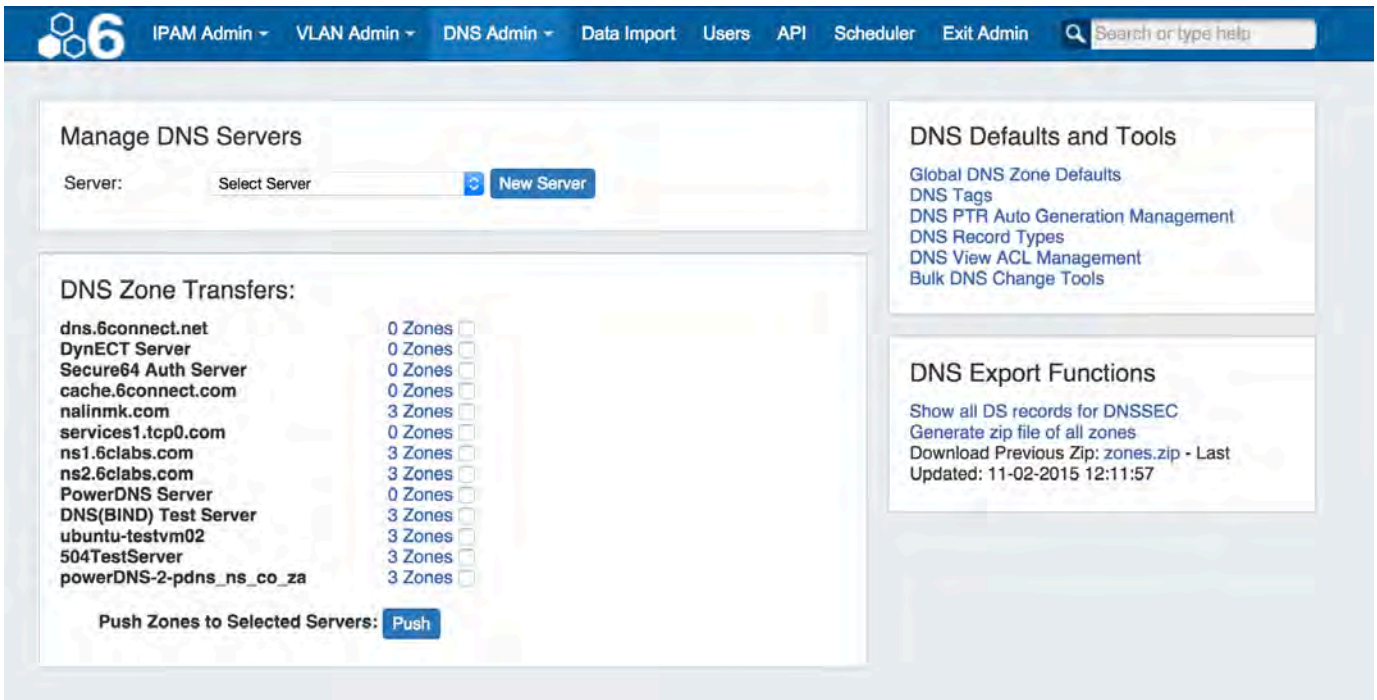


6. After making desired changes, hit the save button to save and exit.

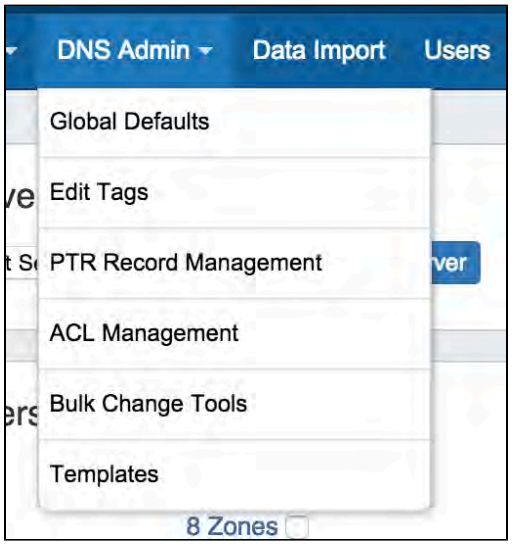
Once assigned, the tags show as a property of the VLAN block in the VLAN edit menu and may be used for VLAN Searches / Assignments.

DNS Administration

DNS Administration

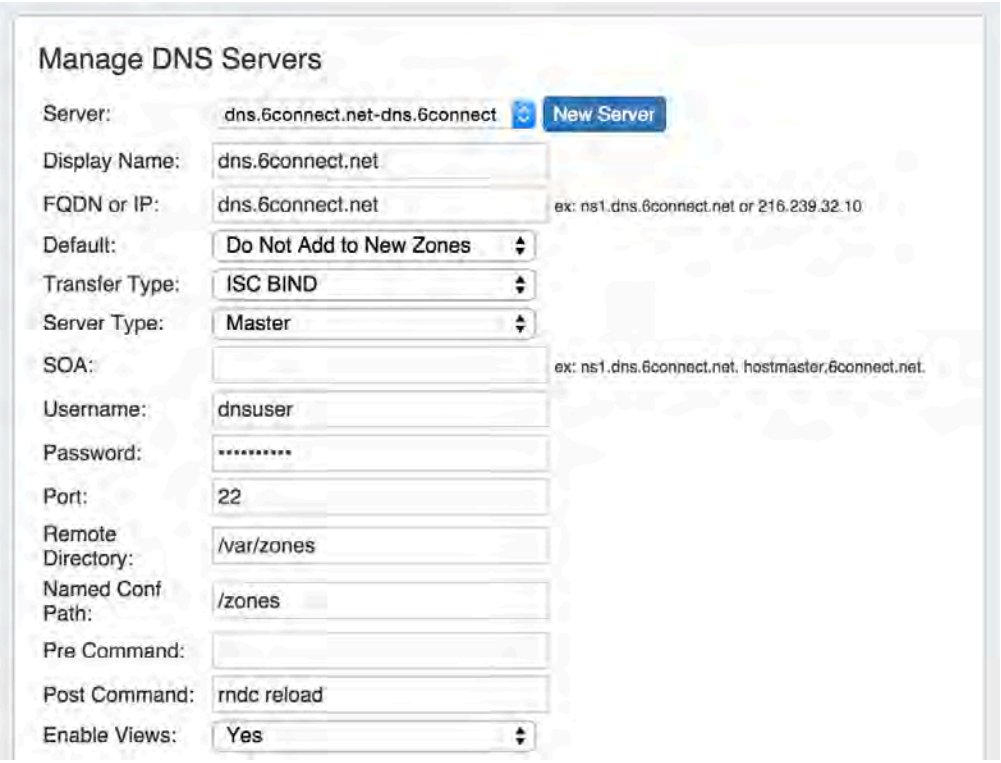


DNS Administration is accessed through the Admin area of ProVision. The **DNS Admin** tab contains four different functional areas: Manage DNS Server, DNS Zone Transfers, DNS Defaults and Tools, and DNS Export Functions. The DNS Admin dropdown menu provides shortcuts to many of the DNS Tools as well as DNS templates.



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 name of the server.

Display Name: Name you want the server to display.

FQDN or IP: The FQDN or ip address of the DNS server.

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.

Enable Views: Select Yes or No to enable / disable views. You must click "Update Server" to show the view options.

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), then click "Add new view". The config files transferred to the server will automatically be built according to the server type.

Adding ACLs to Views

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

"Add Key" and "Val" are fields to provide additional options for DNS Views.

For additional information on working with views, see Configuring Split Horizon / Views.

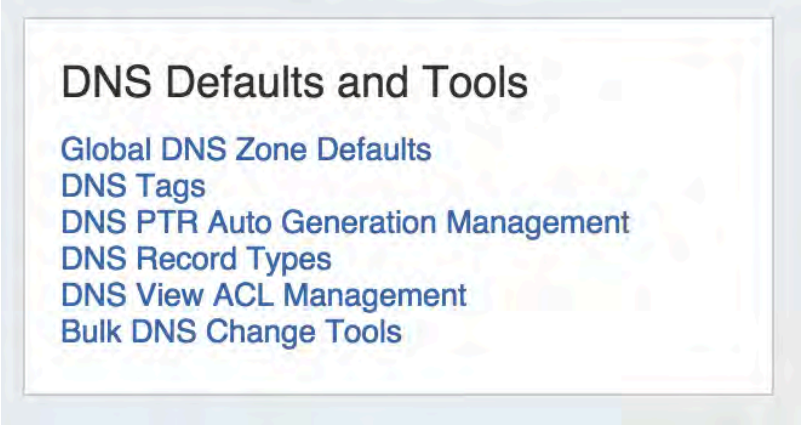
DNS Zone Transfers

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

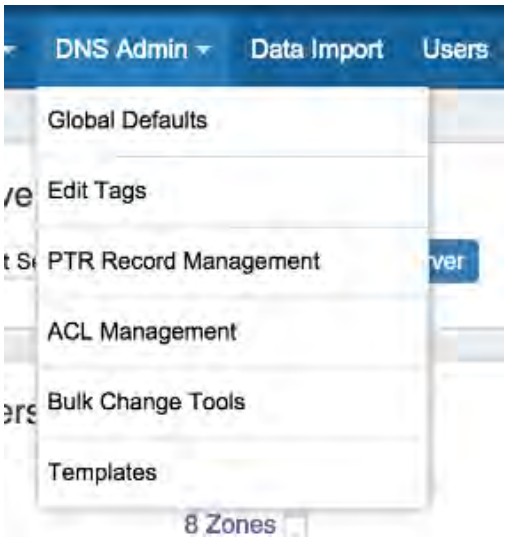
How to transfer zones:

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

DNS Defaults and Tools



This section provides a collection of links for other useful DNS functions including setting Global DNS defaults, DNS Tags, PTR Auto Generation Management, DNS Record Types, DNS View ACL Management, and Bulk DNS Change Tools. Many of the tools are also accessible from the DNS Admin dropdown menu.

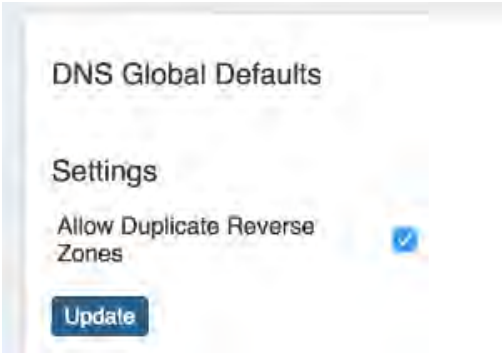


Global DNS Zone Defaults

DNS Settings

Provides DNS settings options.

Allow Duplicate Reverse Zones: Check to enable / disable allowing duplicate reverse DNS zones. If duplicate reverse zones already exist, those zones must be removed before disabling duplicates. If a zone has duplicates, a link appears in the top right corner of that zone's ViewZone page.



DNS Global Defaults / Default SOA Values

Provides default configuration settings 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

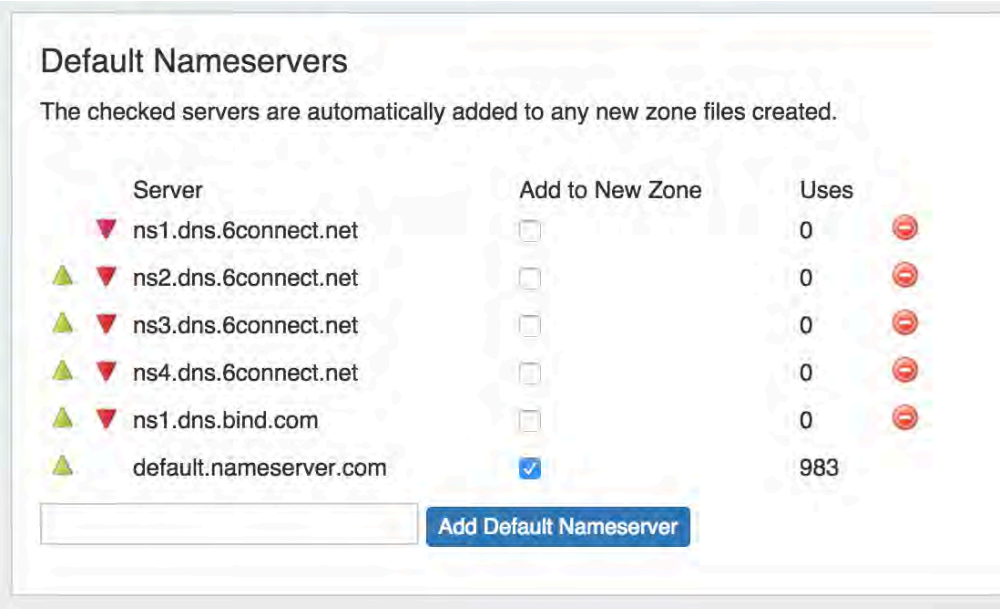
Default Minimum: in seconds, default value is 3600

Default SOA: Server Of Authority and hostmaster contact. E.g. ns1.domain.com. hostmaster.domain.com.

Default Nameservers

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

The checked servers are automatically added to any new zone files created.



To remove a server from default status, uncheck the box under "Add to New Zone". Servers with "0" Uses may be deleted by hitting the red delete icon.

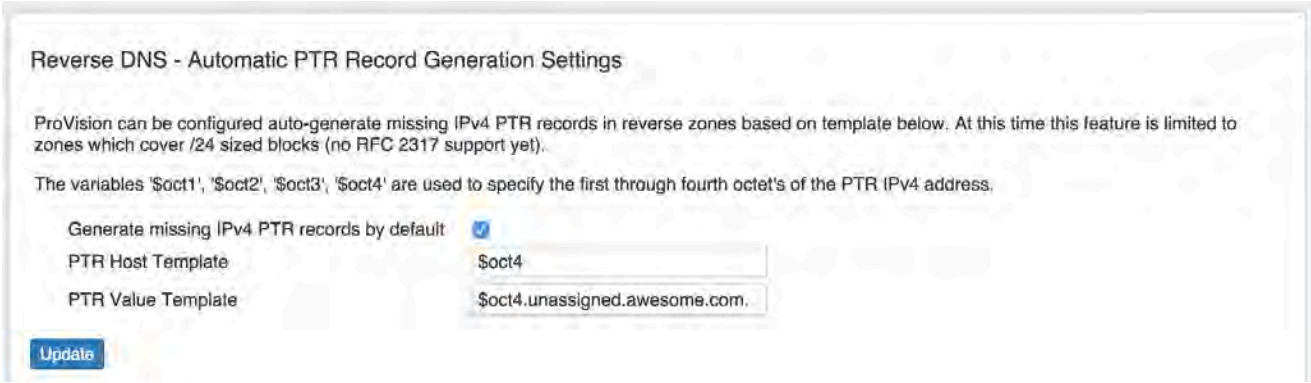
DNS Tags

Under DNS Tags (Or "Edit Tags" from the DNS Admin dropdown), you can manage the tag list that is available to apply to DNS zones. See Working with DNS Zones for detailed information on managing DNS Tags.

DNS PTR Auto Generation Management

ProVision can be configured auto-generate missing IPv4 PTR records in reverse zones based on the template provided on this page. This feature is limited to zones which cover /24 sized blocks (no RFC 2317 support yet).

The variables '$oct1', '$oct2', '$oct3', '$oct4' are used to specify the first through fourth octet's of the PTR IPv4 address.



DNS Record Types

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

See Working with DNS Zones for detailed information on managing DNS Record Types.

DNS View ACL Management

DNS View ACL Management

- Manage ACLs for use in DNS Views.

See Configuring Split Horizon / Views for detailed information on using DNS View ACL Management.

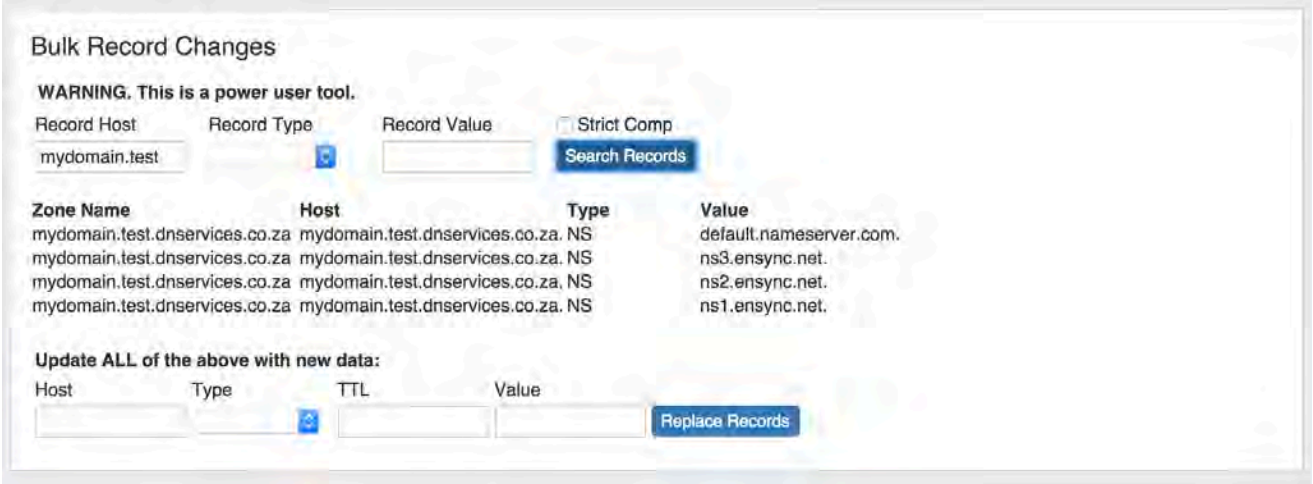
Bulk DNS Change Tools

Bulk Zone Assignment

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.

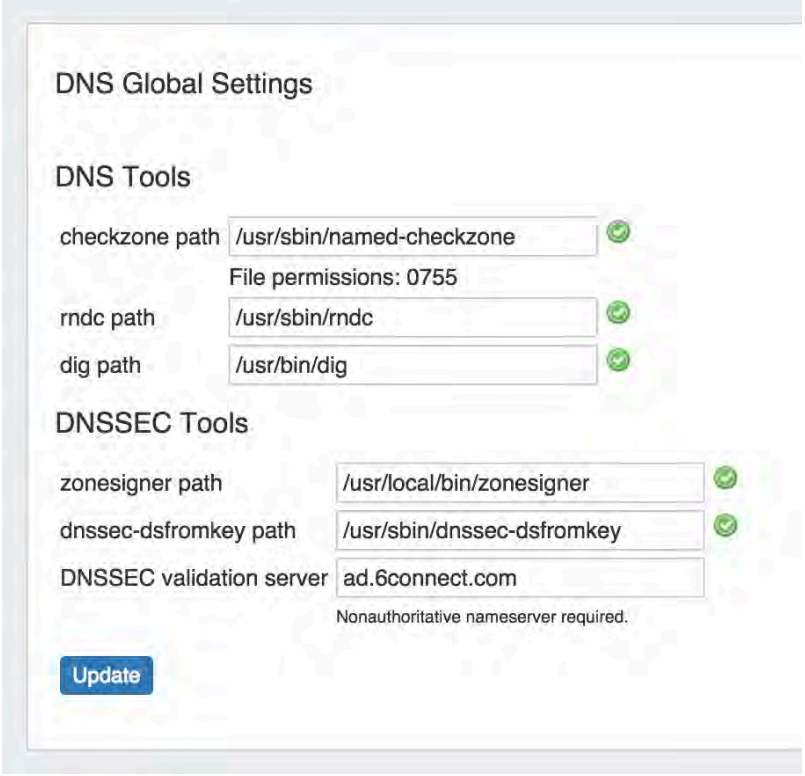
Bulk Record Changes

The Bulk DNS Editor allows an Admin to perform "find and replace" functions across all DNS zones. Enter Record Host, Record Type, and/or Record Value information and select "Search Records". 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. You can then replace the data for the results by using the fields below.



Global DNS Settings (Local Installation Only)

The "Global DNS Settings" link is only viewable with the local installation version of ProVision.



Checkzone path: Path to checkzone

rndc path: Path to rndc

dig path: Path to dig

zonesigner path: Path to zonesigner

dnssec-dsfromkey path: Path to dnssec-dsfromkey

DNSSEC validation server: Address of DNSSEC validation server, required to be a non-authoritative name server.

DNS Export Functions

This section provides links for export functions.

Generate/ Show all DS records for DNSSEC

- This link will generate/show 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.

Additional Information:

Importing DNS Zones

ProVision offers three DNS zone import options, available under the Data Import tab in the Admin section. For more information on importing DNS zones, see Importing your Data and Import DNS Zones.

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 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.

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 Sections:

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 Split Horizon/Views
- Configuring DNS Templates
- Configuring DNSSEC
- DNS Audit Tools (Alpha)

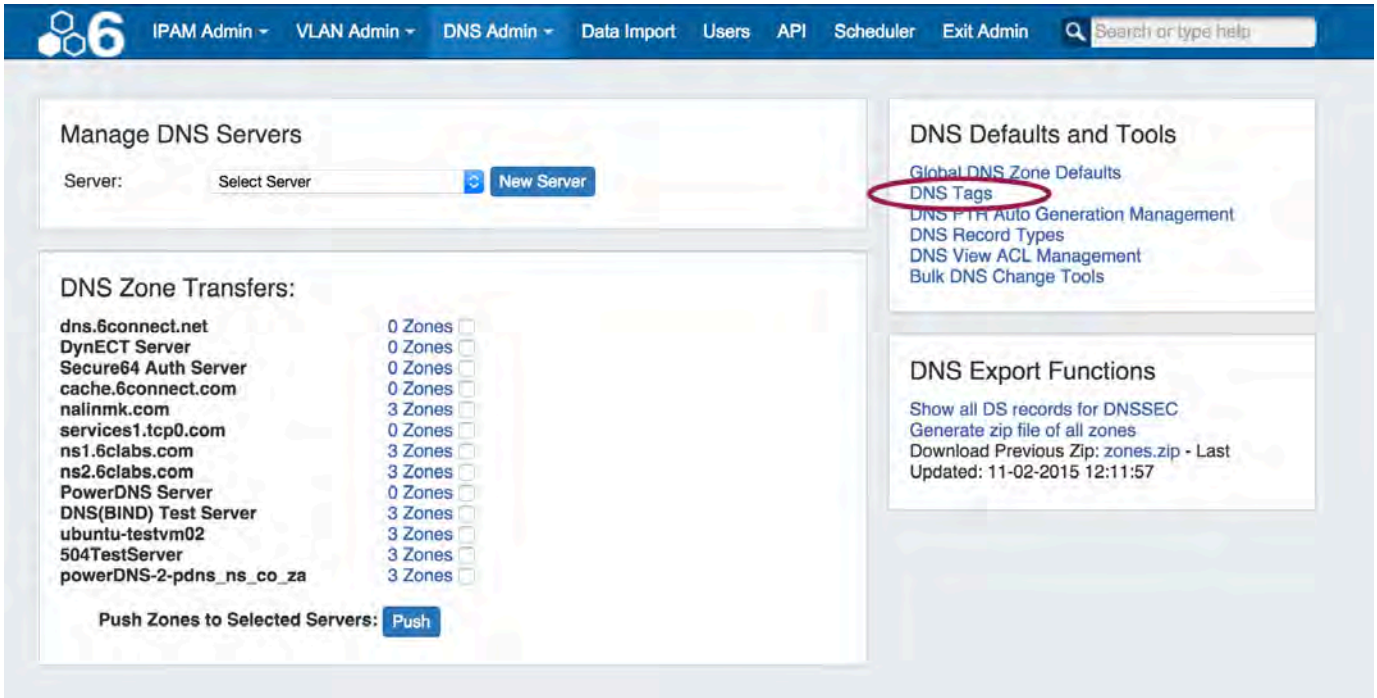
Working with DNS Zones

Working with DNS Zones - List Management

This page describes in detail some of the common list management tasks that may be performed in the DNS Admin section.

Add / Edit DNS Tags

Access the DNS Tags link under **DNS Defaults and Tools** in the **DNS Admin** Tab. It is also available from the DNS Admin dropdown menu, under "Edit Tags".



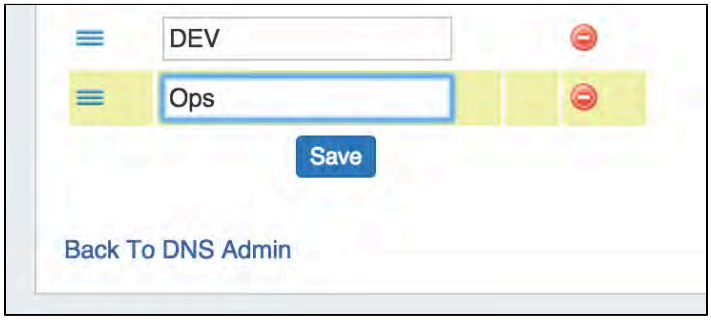
When you are working with DNS Zones in the **DNS** Tab, you have the option to add tags to further categorize the zone. DNS Tags are used in a number of areas in ProVision and can be added or edited from this screen.

To add a new DNS tag, click on "Add Item" at the top of the Edit DNS Tags page.

Then, type in the desired name value for the new tag, and hit "Add Item".

To **edit** a tag, simply type your changes in to the text box with the tag name. Tags with unsaved changes will be highlighted until saved.

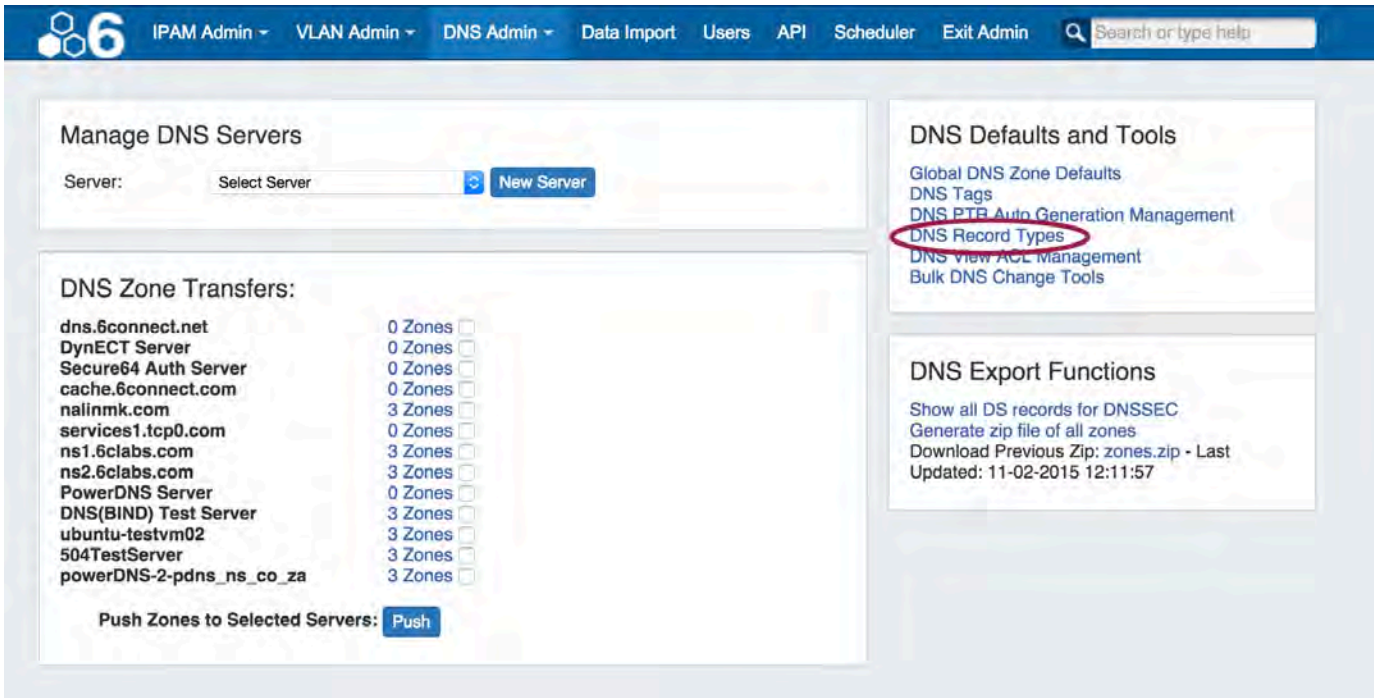
To **delete** a tag, click on the red "delete" symbol to the right of the tag name.



When complete, be sure to click on the "Save" button to save your changes.

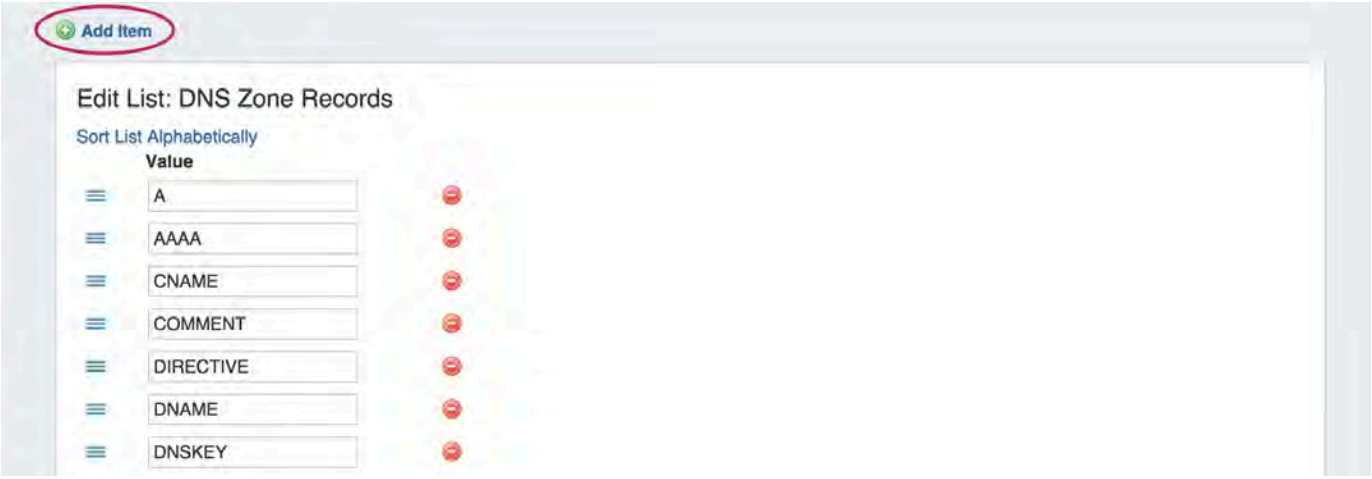
Managing DNS Record Types

Access the DNS Record Types link under **DNS Defaults and Tools** in the **DNS Admin** Tab.



The list of DNS Record types used in DNS Zone areas may be customized on this page.

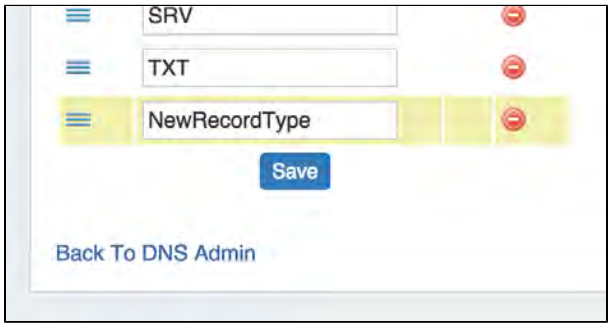
To add a new DNS Record type, click on "Add Tag" at the top of the Edit DNS Zone Records page.



Then, type in the desired name value for the new tag, and hit "Add Item".

To **edit** a record type, simply type your changes in to the text box with the type name. Types with unsaved changes will be highlighted until saved.

To **delete** a record type, click on the red "delete" symbol to the right of the type name.



When complete, be sure to click on the "Save" button to save your changes.

Configuring ISC BIND Support

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:

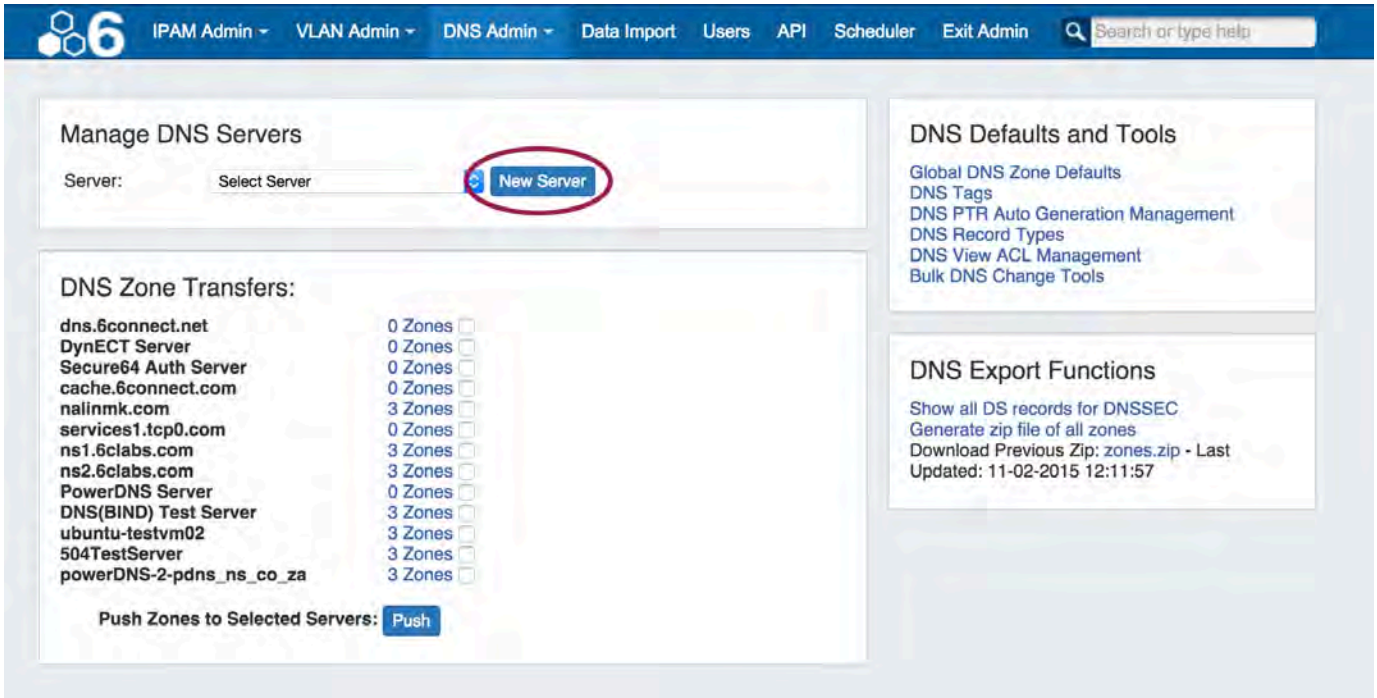
include "/path/to/conf/directory/6connect_named.conf";

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

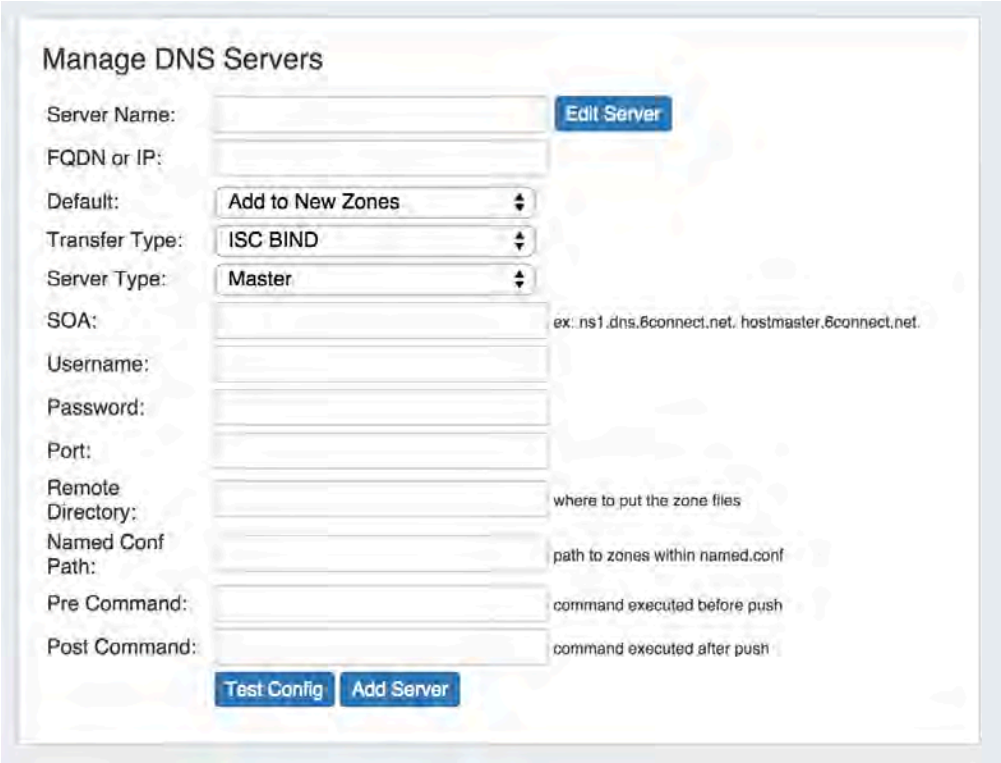
It is also important to note that if your existing named.conf file contains zones within Split Horizon views, then the 6connect-managed zones must also be view-enabled. Likewise, if existing zones are not grouped into views, then views must be disabled on ProVision.

Adding a BIND Server to ProVision

Go to the **DNS Admin** tab and under **Manage DNS Servers**, click "New Server"



Select "ISC BIND" as the Transfer Type, and fill in the server information fields. For detailed descriptions of the fields, see New Server dialogue section on the DNS Admin page.



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.

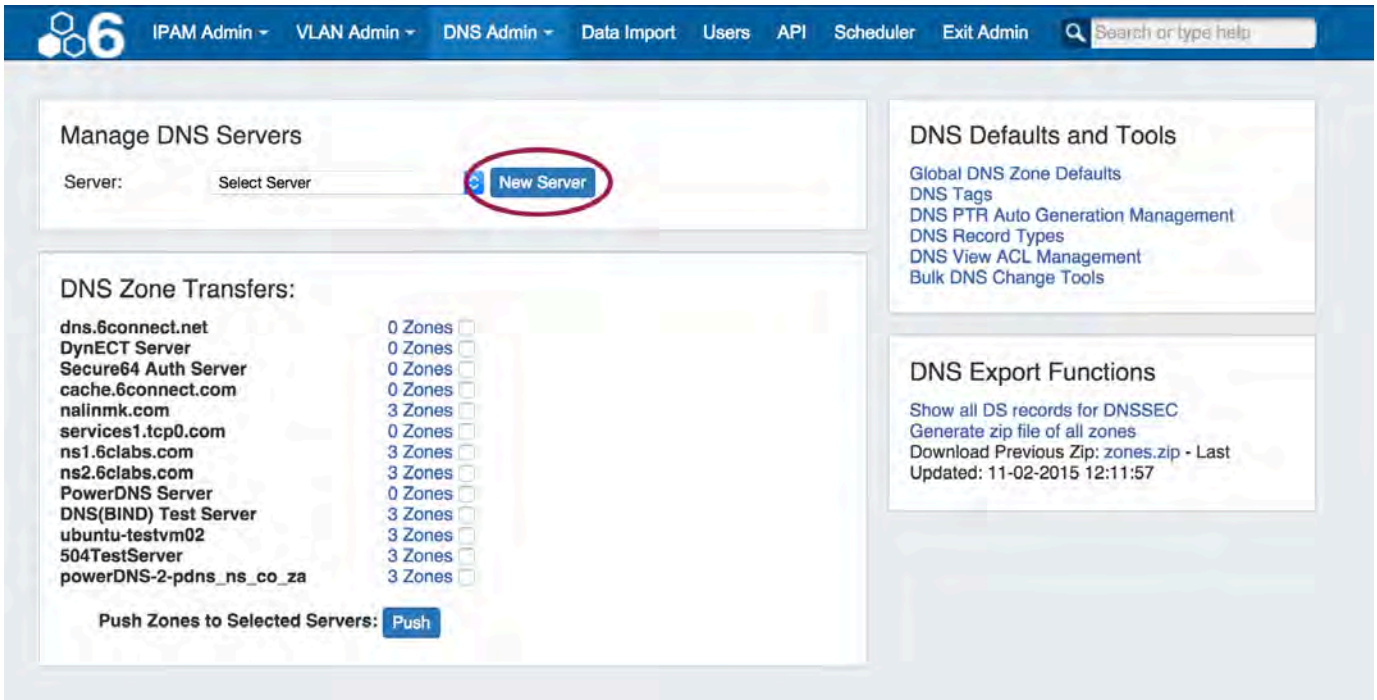
When done, click on the "Add Server" button to save your server information.

Configuring DynECT Support

Configuring DynECT Support

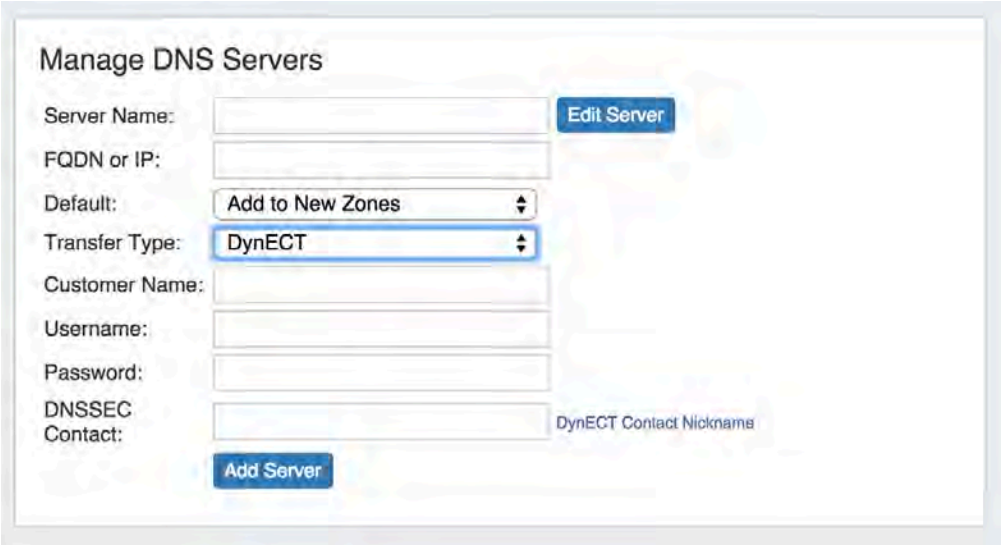
Adding a DynECT Server to ProVision

Go to the **DNS Admin** tab and under **Manage DNS Server**, click "New Server"



Select "DynECT" as the Transfer Type, and enter your Dyn username, password, and customer name along with the server information fields. For detailed descriptions of the fields, see New Server dialogue section 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.



When complete, hit the "Add Server" button.

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

Configuring PowerDNS Support

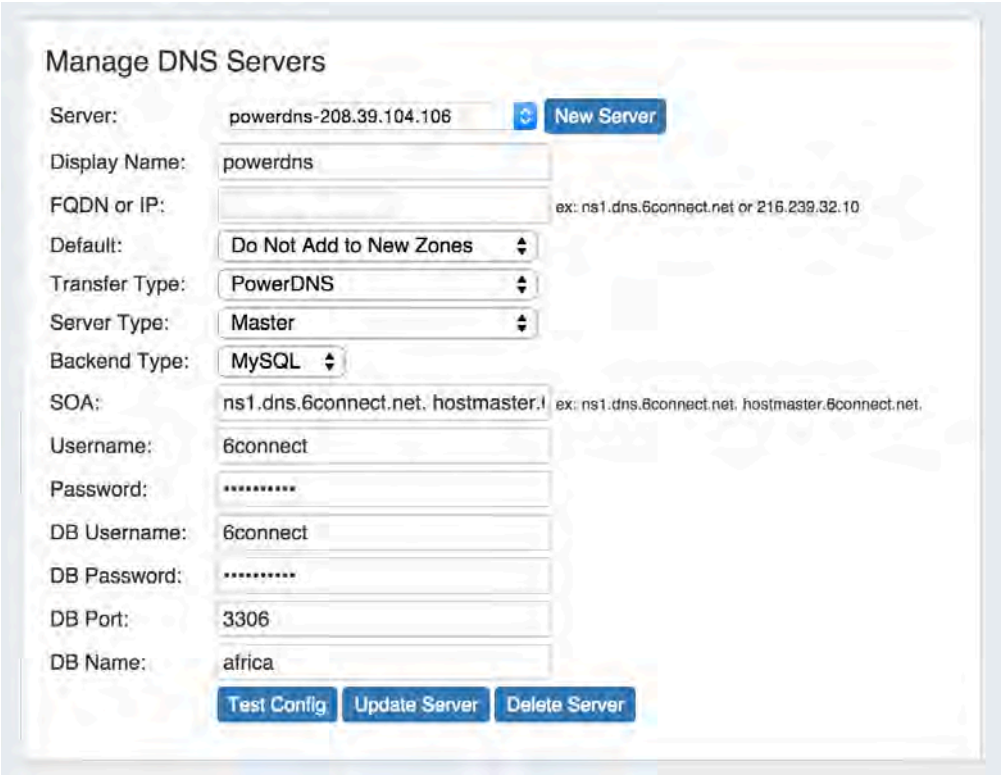
Environments supported

- PowerDNS version 3.0 or above on the target server(s)
- BIND or MySQL backend

Overview

Step 1: Setup your PowerDNS Server

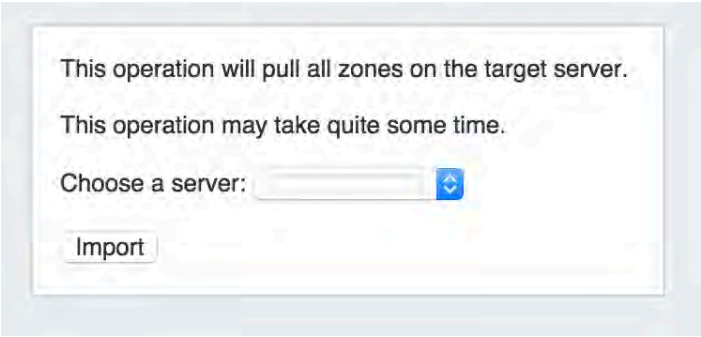
In the **DNS Admin** tab, Under "Manage DNS Servers", select "New Server", then add the information for your PowerDNS server. See Manage DNS Servers for detailed field information.



Step 2: Import your PowerDNS zones

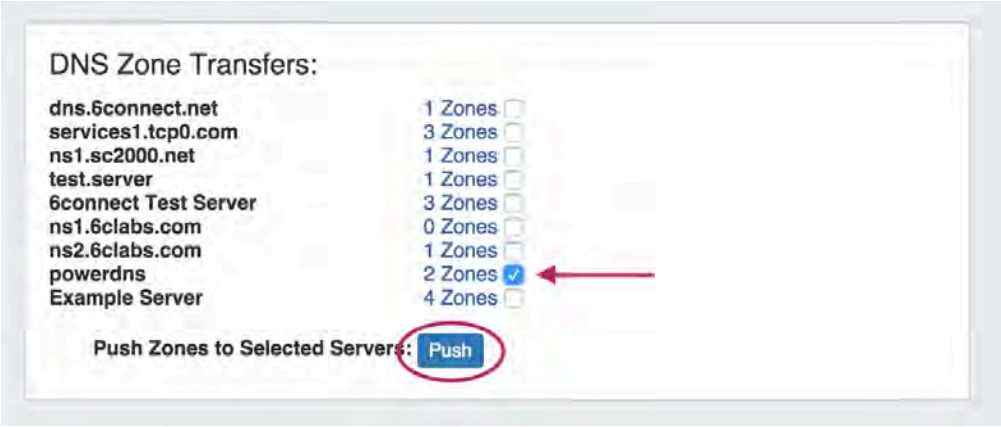
While in the **Admin** section, navigate to the **Data Import** Tab. Select the "Power DNS Zone Import" link.

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



Step 3: Edit/Push your zones to PowerDNS

Navigate back to the In the **DNS Admin** tab, and under "DSN Zone Transfers, select the check box next to the PowerDNS server zones that you want to push, then click "Push".



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

Configuring Secure64 Support

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

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

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

DO THIS
Make sure to add the line include: 6connect_nsd.conf to the nsd.conf file

Output/Input

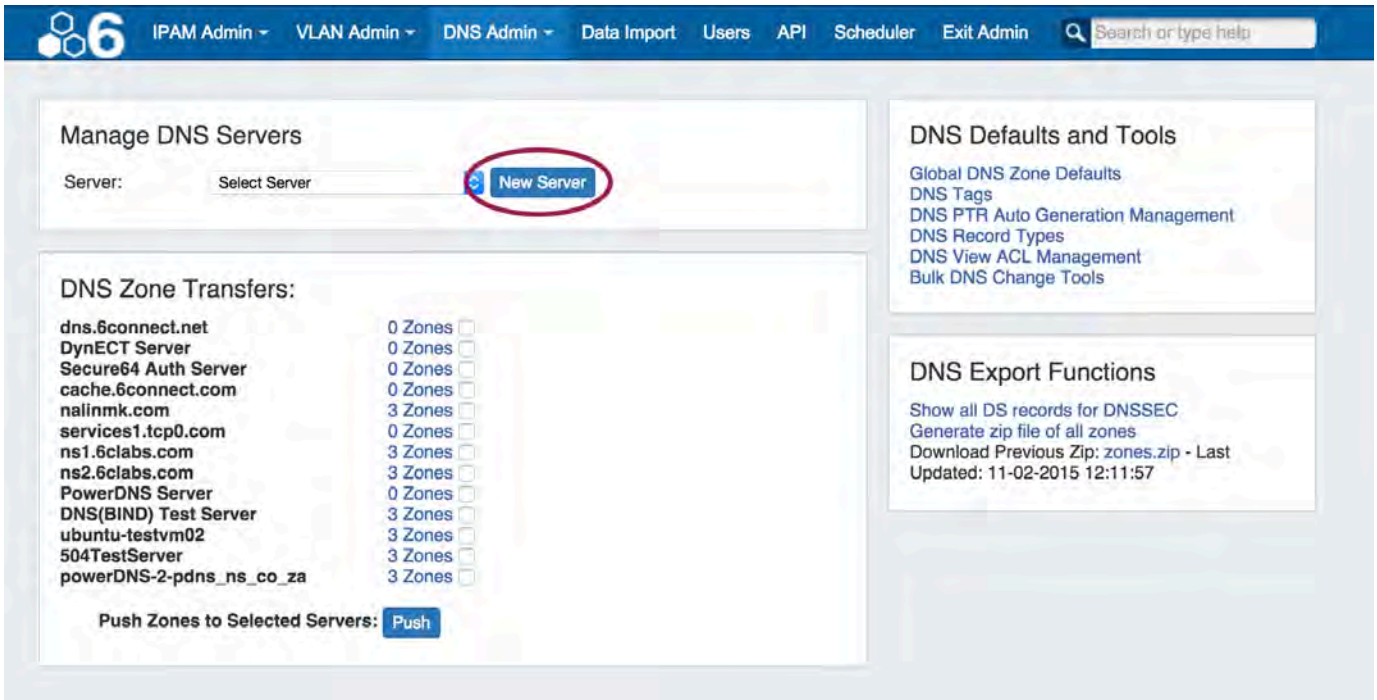
```
[authdnsadmin@Secure64DNS]# cat nsd.conf
server:
ip-address: 50.198.192.141

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

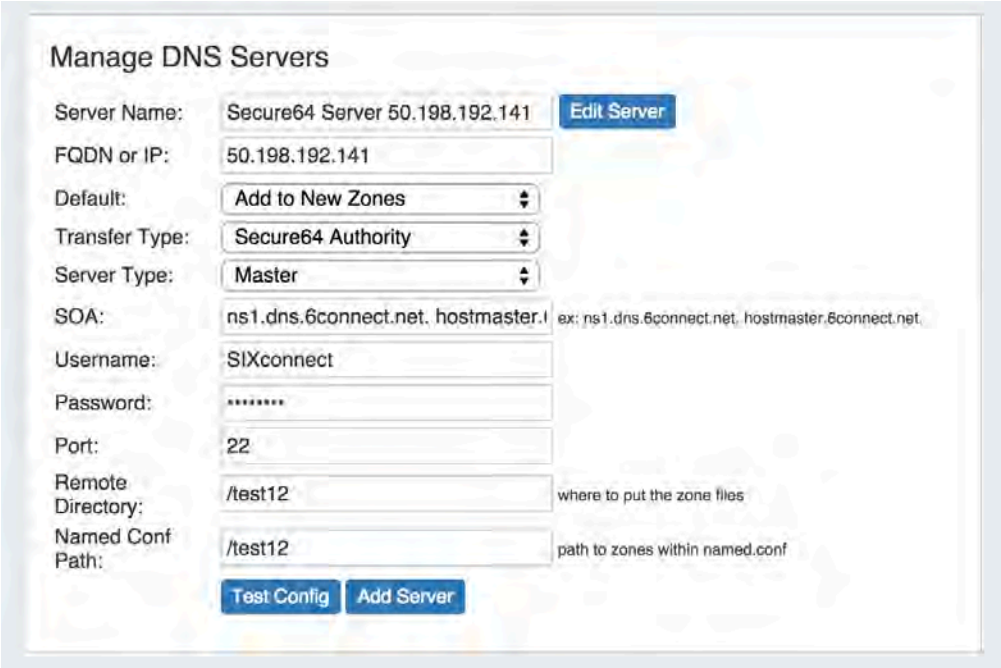
```
[authdnsadmin@Secure64DNS]# mkdir test12
[authdnsadmin@Secure64DNS]# ls
/:
322 2013-08-19 06:07:42 nsd.conf
<DIR> 1024 2013-08-16 17:30:12 test12
```

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.



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

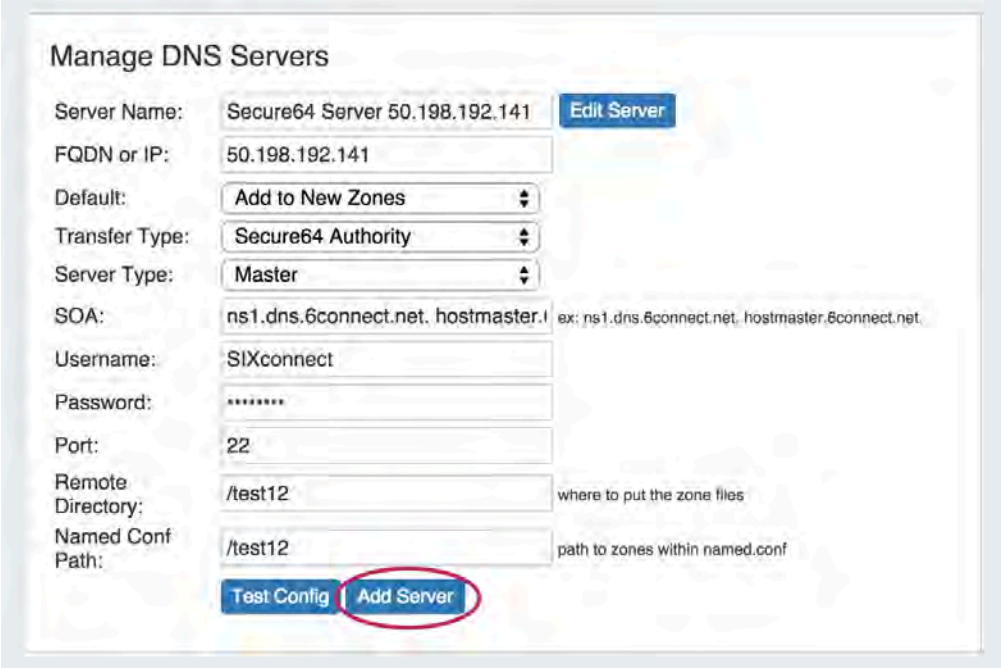


Step 4: Test the Secure64 DNS Server configuration

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

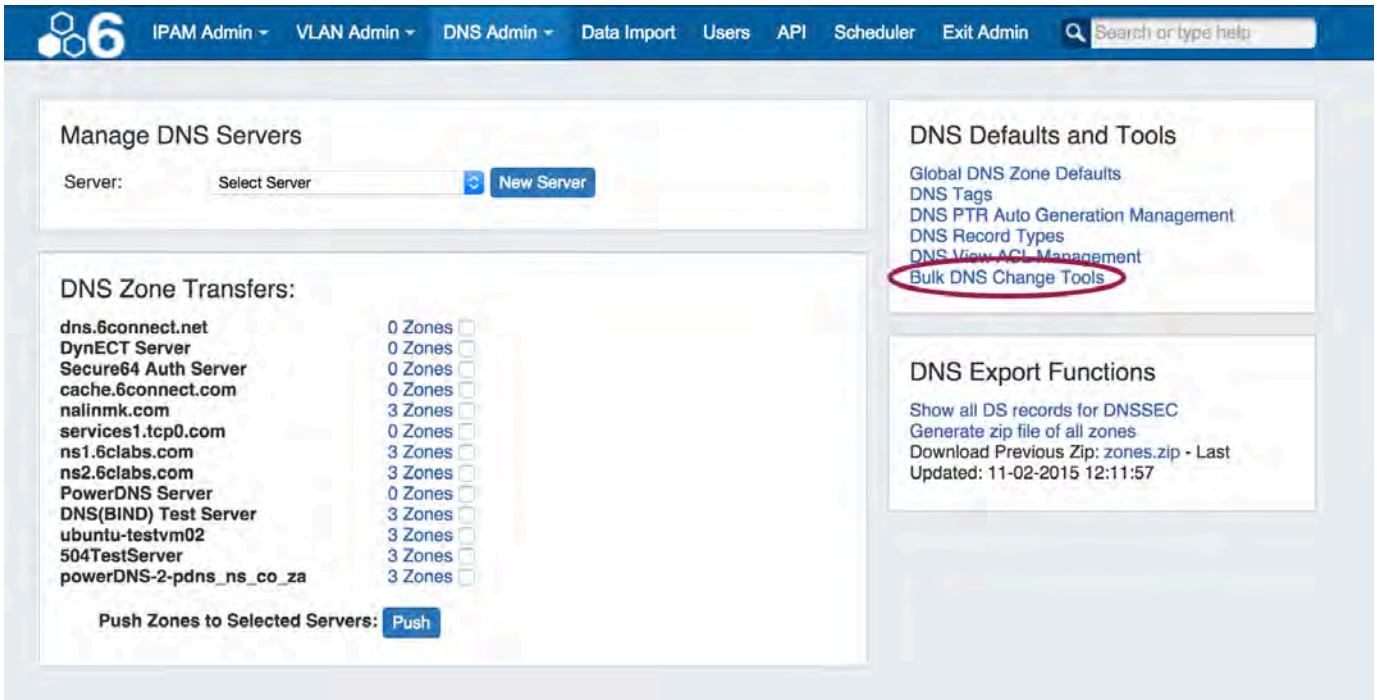
A success or error message will show.

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



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

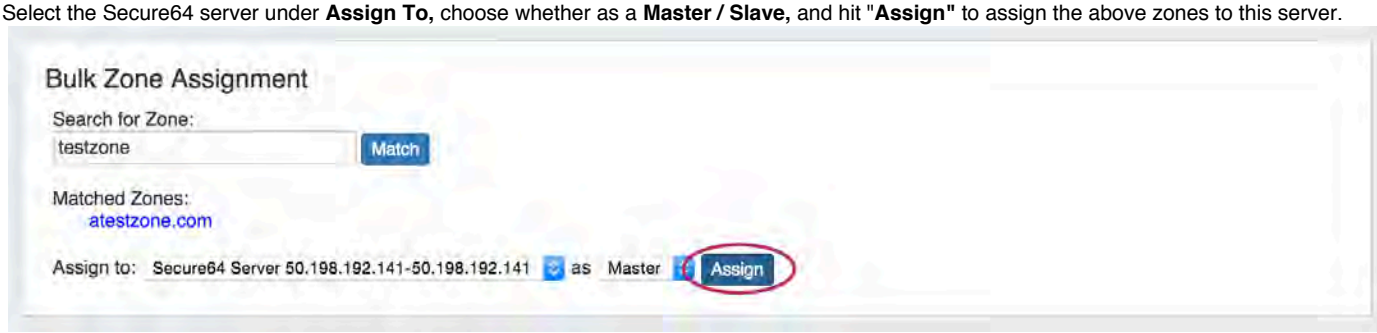
Select the "Bulk DNS Change Tools" link under the DNS Defaults and Tools section of the page. It is also available under the DNS Admin dropdown menu.



Search for all available zones or enter in a value to find specific existing zones in the system. Click the "Match" button to see results.

> **Search Tip**
> No character in the search area indicates a search for all zones

Select the Secure64 server under **Assign To,** choose whether as a **Master / Slave,** and hit "**Assign"** to assign the above zones to this server.

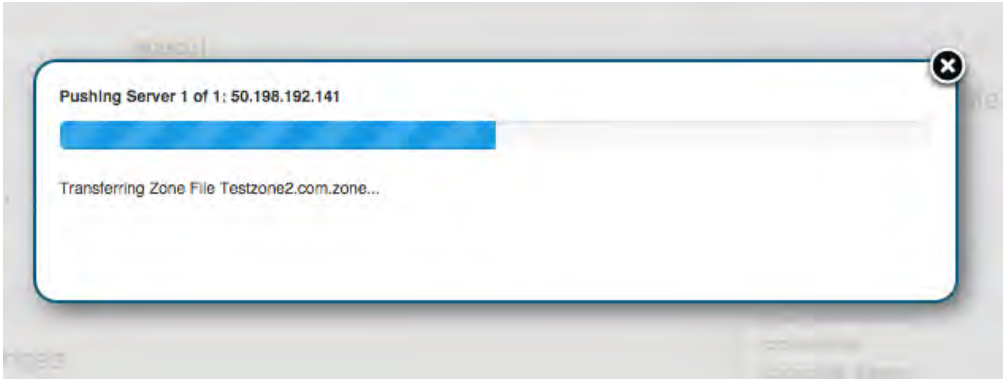


Step 6: Push Zones to Secure64 Server(s)

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 system will present the following live progress bar.



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, or Secure64 at support@secure64.com

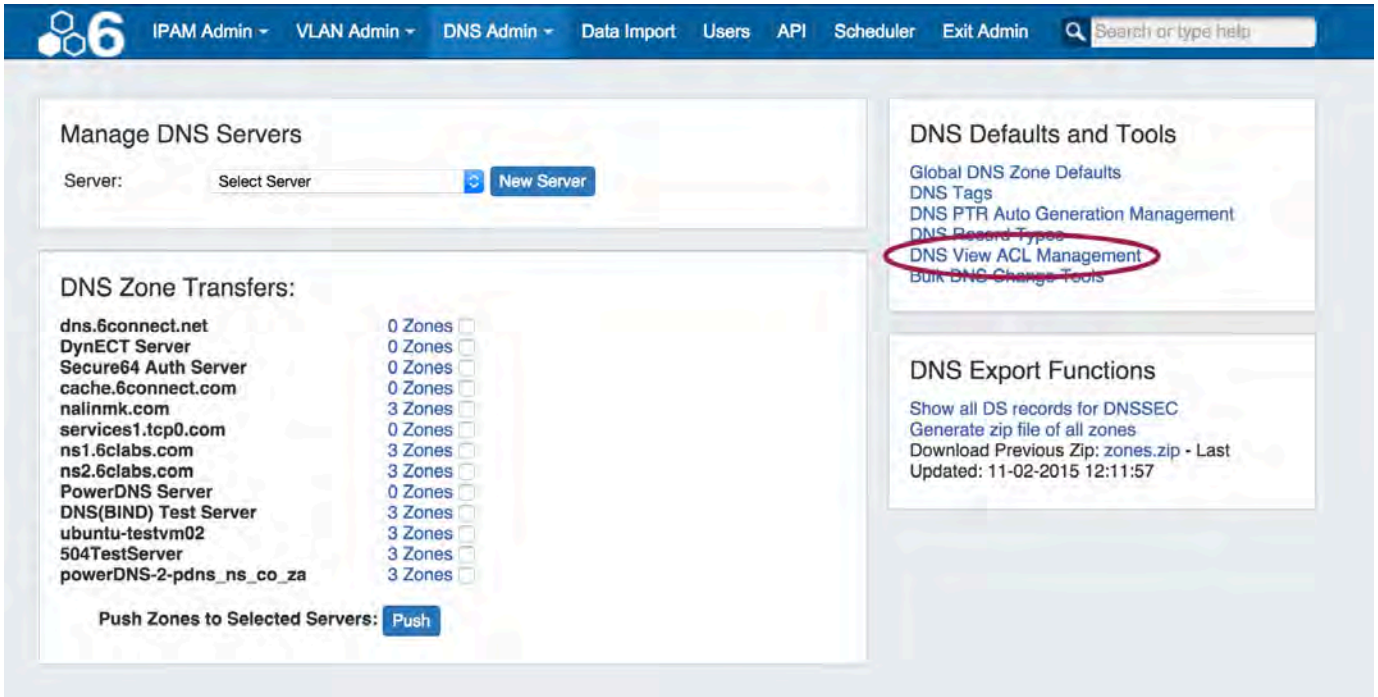
Configuring Split Horizon/Views

Configuring Split Horizon/Views

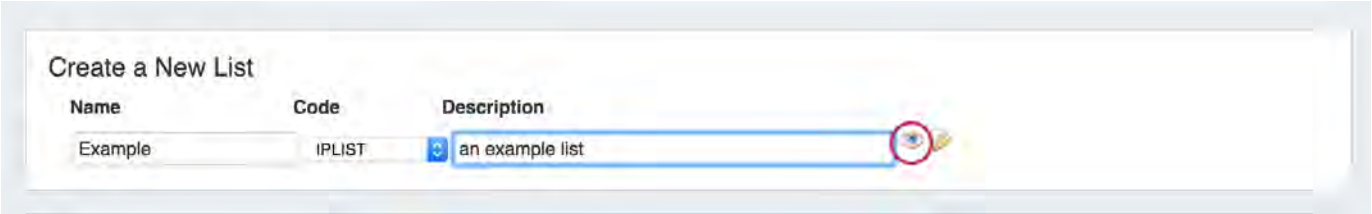
> **WARNING**
> If you see a view named "_6connectDefault" - DO NOT DELETE IT.

Create a List in the List manager

The List manager is accessed from the **DNS Admin** tab. Click on the "DNS View ACL Management" link under DNS Defaults and Tools. It is also available under the DNS Admin dropdown menu.

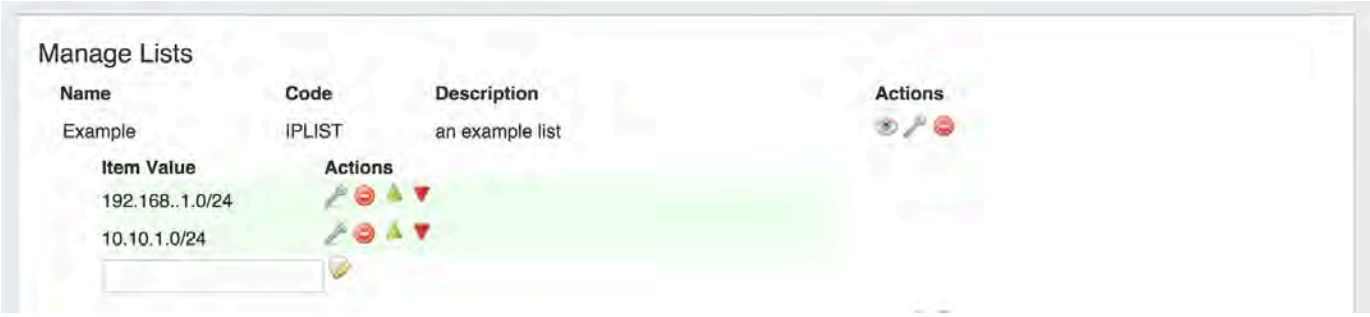


You will be presented with the options to **Create a New List** and also **Manage Lists**. To create a list, enter in the descriptive information and ensure that the **Code** dropdown is marked "**IPLIST**".



Press the **Eye** icon and you will be presented with en 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.

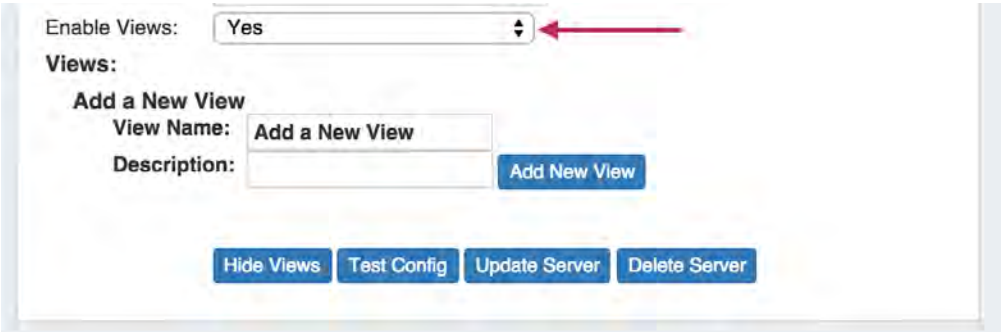
The List will now be available from the **Manage Lists** display area and can now be assigned to a Server View.



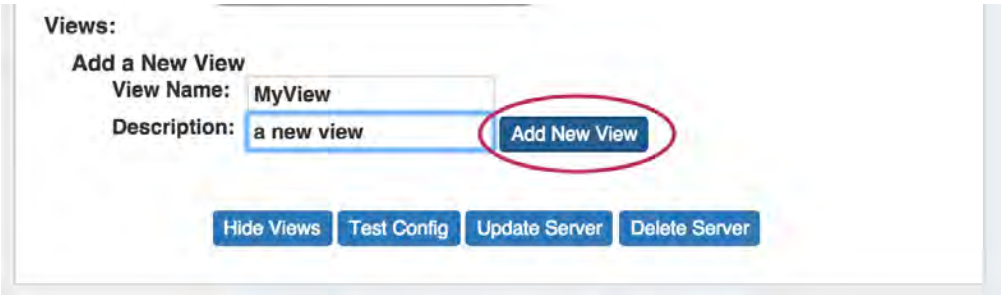
Define and Assign a View to the DNS Server

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

Under "Manage DNS Servers", select a server and check "Enable Views". You will then have the option to define a View.



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



Once the View is created, you can select the IP List that you want to assign to this View from the dropdown menu, and then press the "Add" button. Here, we have selected our "Example" list to add.

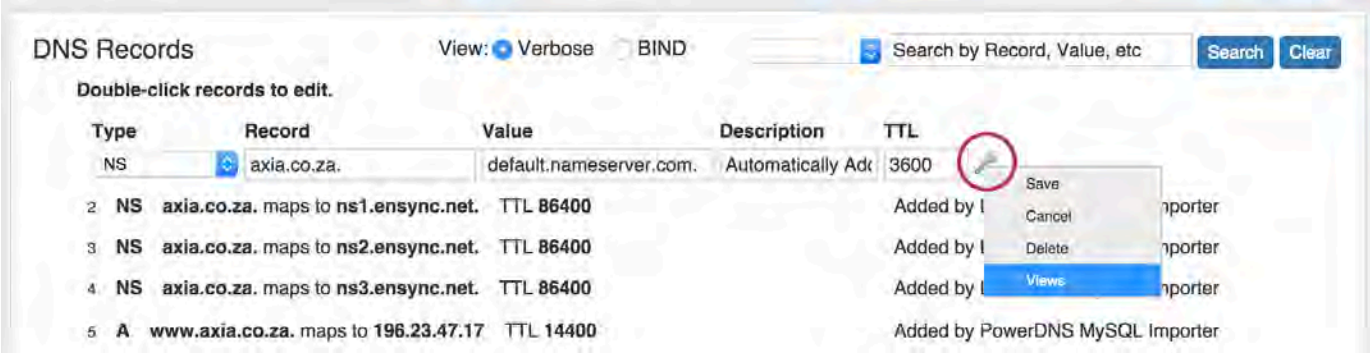
Assigning other Directives

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

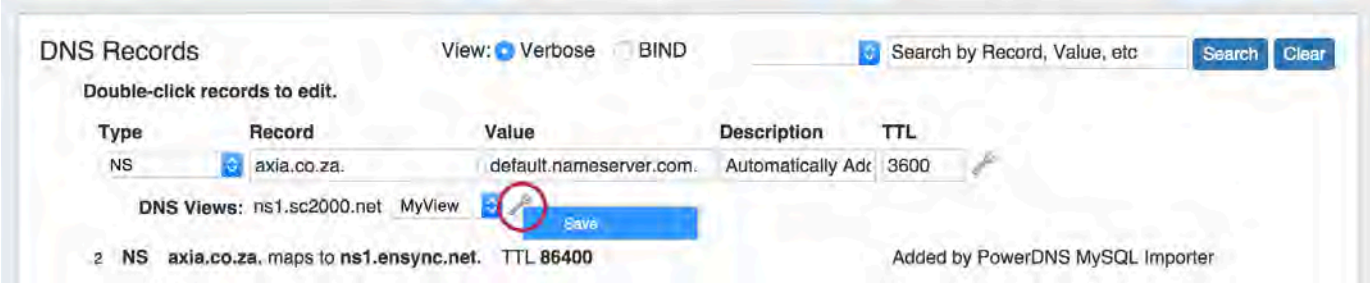
> **A Note on Directives**
> For example, if you wanted to allow recursion, you would simply enter "allow-recursion" as a Key, with a Value of "on".

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. Then, under the "DNS Records" section, double-click on the zone record to edit it. Click on the Action Menu (wrench icon) and select "Views".



It will bring up the DNS Views menu where you can select the View from the dropdown menu to apply to the zone record. Click on the Action Menu (wrench icon) and select "Save" to save your changes.



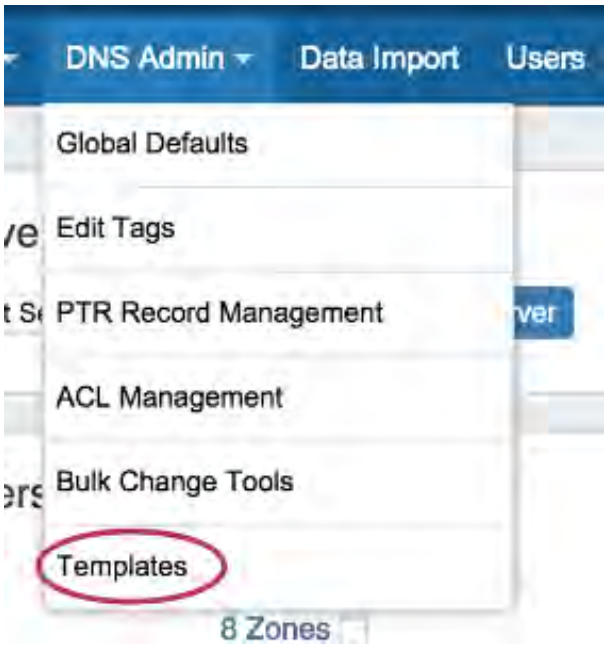
Push the zone out like normal and the View should be applied as expected. You can also preview the zone from the "Show Zone" area of the screen that will be visible once you push the zone out successfully. This will also display the History for the zone if a rollback is necessary.

Configuring DNS Templates

Configuring DNS Templates

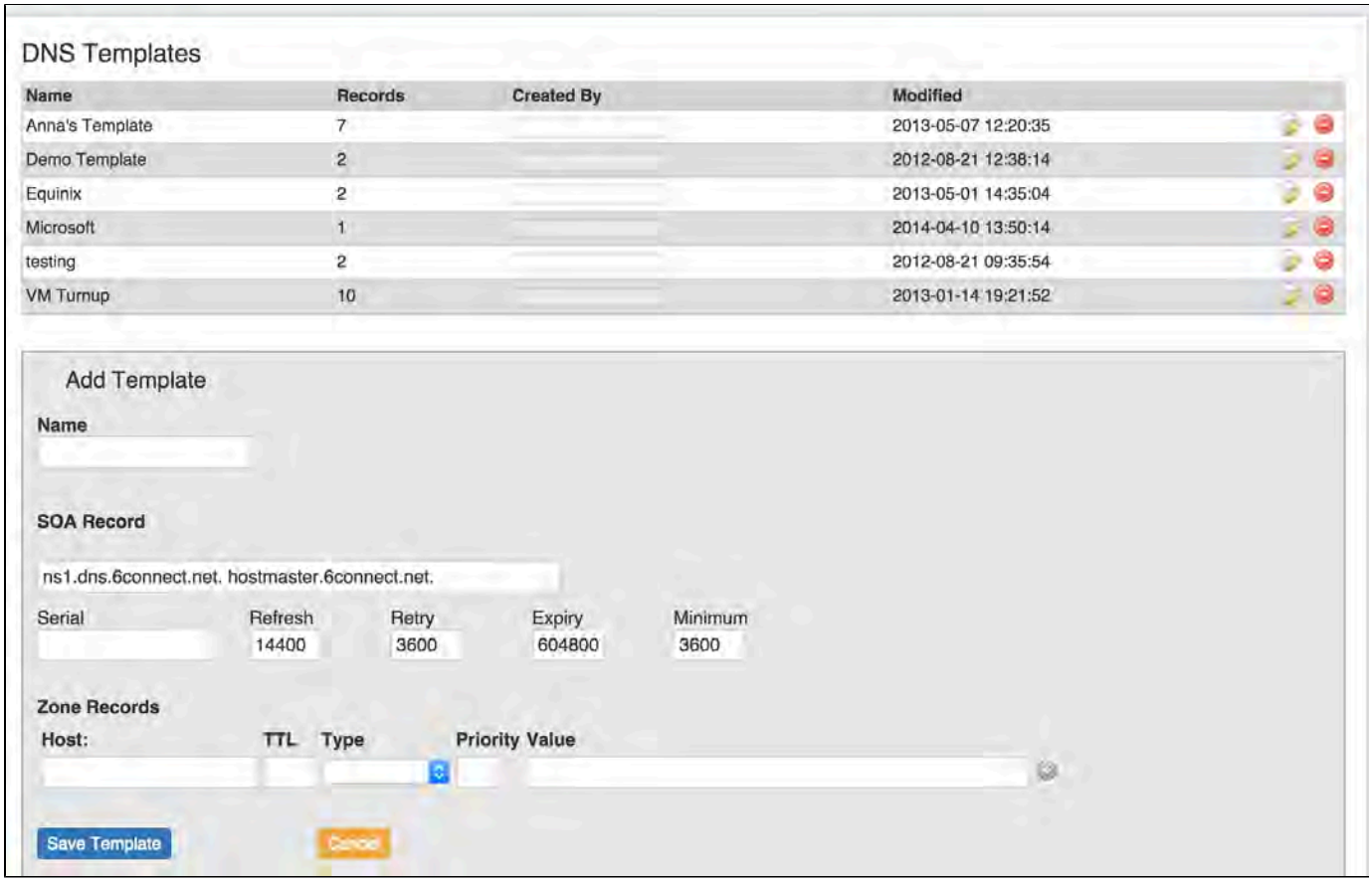
Overview

When creating a new DNS zone, the user can specify a zone template to use. Templates are setup from the DNS Admin dropdown menu, under "Templates".

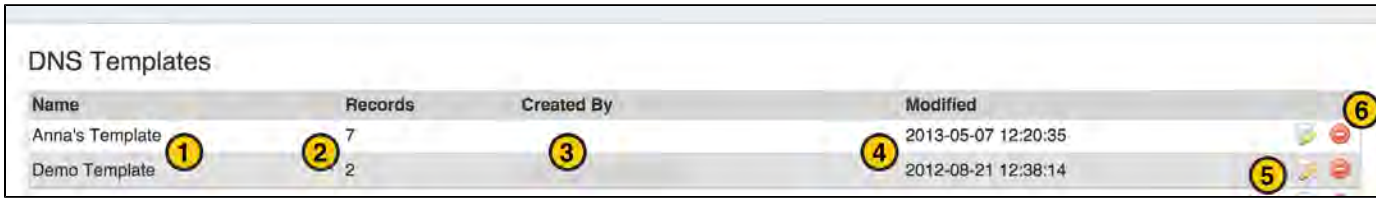


DNS Templates

Once on the DNS Templates page, you will see a list of existing DNS Templates, and below that, an area to Add / Edit a template. Unless an existing template has been selected, the template editing area will default to "Add Template" for creating new templates.



UI Overview



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

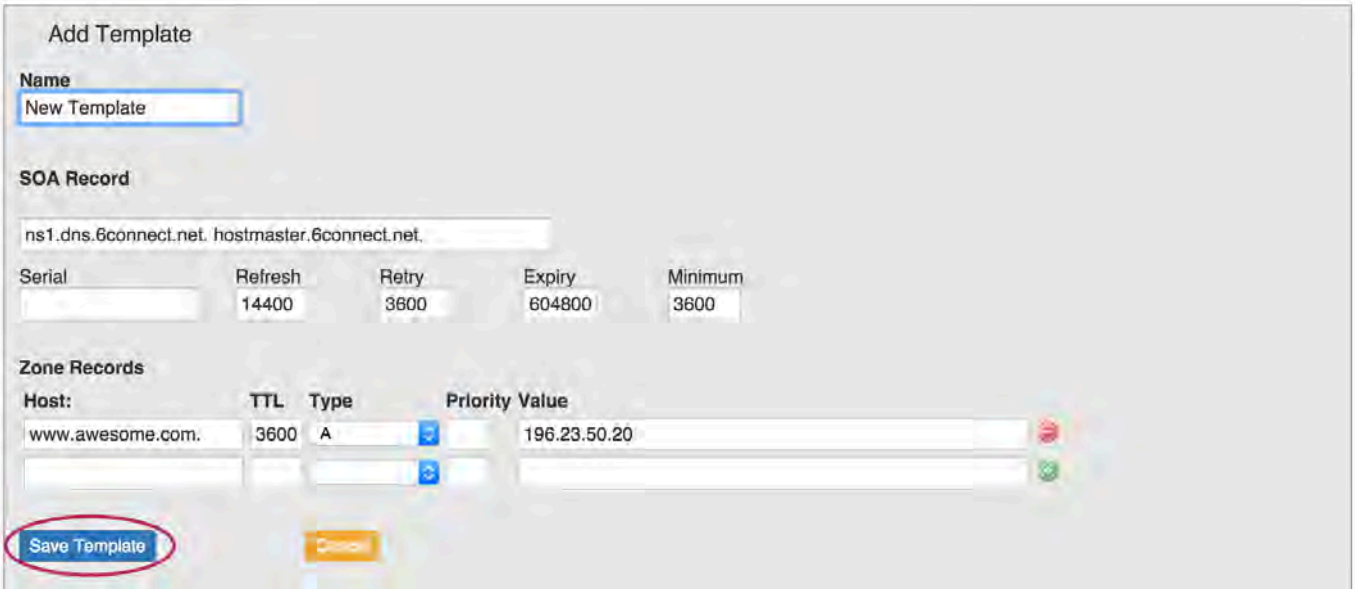
5) **Edit:** Click to bring up the template detail below and edit information

6) **Delete:** Click to delete the DNS templat

Create a New Template

To create a new template, verify that the edit template area header says "Add Template".

Enter in your new template name, SOA information, and default zone records. When adding records, click the green "plus" icon when adding record information to save your record and add another.

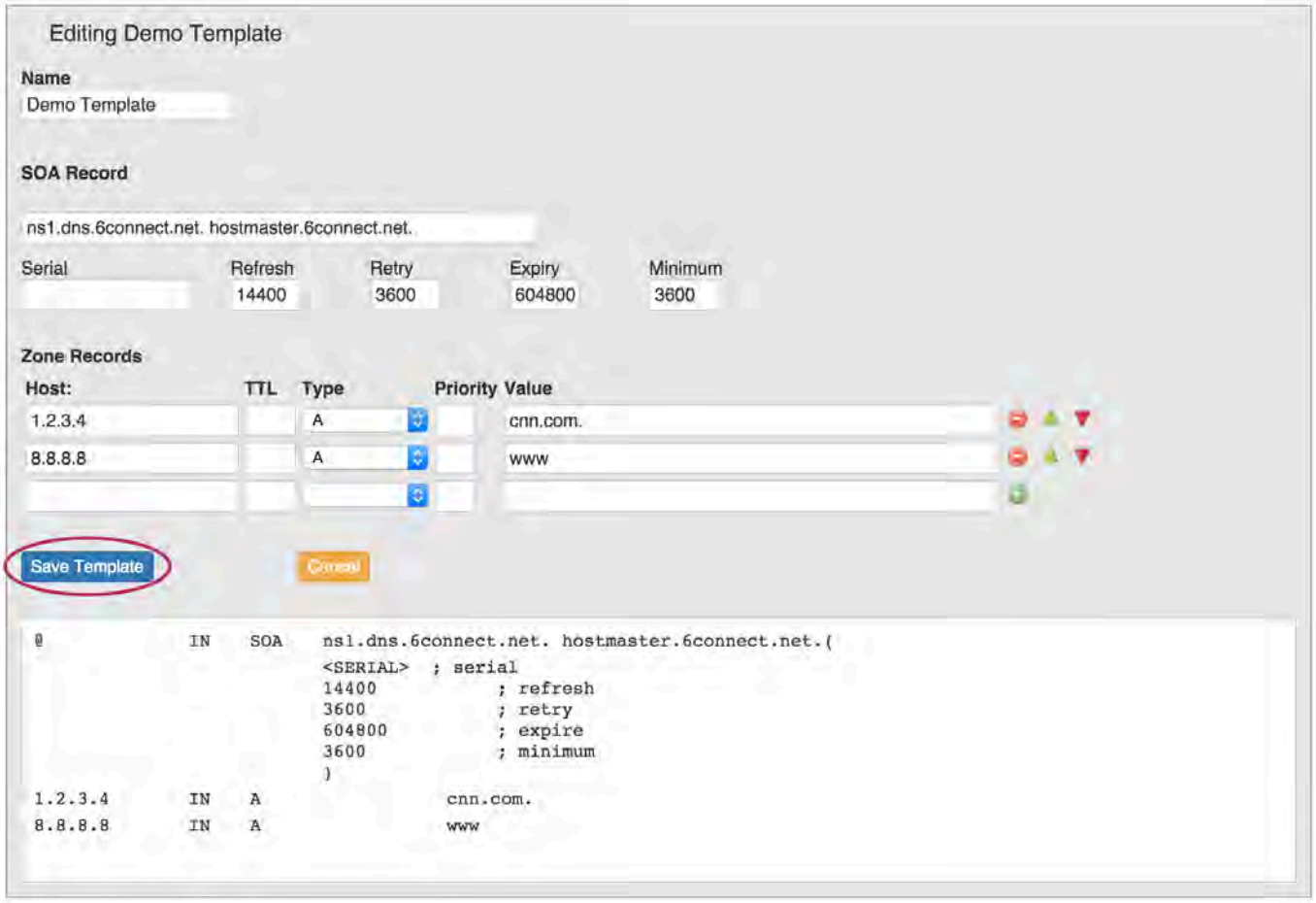


When complete, hit "Save Template". Your template will then be added to the list above.

Edit a Template

To edit a template, select the pencil icon next to the desired template.

When editing a DNS template, the edit template area will change its header to say "Editing (template name)". You can then specify the data in the fields below. After making your changes, click "Save Template".



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

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

Delete a DNS Template

To delete a template, simply click on the red delete icon next to the template name.

Using DNS Templates

From the DNS Gadget, when creating a new DNS zone, select the DNS Template from the dropdown that you would like to use.

Configuring DNSSEC

Configuring DNSSEC

DRAFT - working on this and will be adding some images/visuals...

Enabling DNSSEC for a zone via ProVision GUI
How to enable DNSSEC (per zone) via the ProVision GUI

- Make sure DNSSEC is enabled on the DNS server(s) you will be pushing zones to (see below)
- run **configTest.php** to make sure that your directories/permissions are correct
- Set external server for Authenticated Data verification (DNS Admin setting)
- Create/Edit a zone like usual
- Link the zone to a DNS server(s) as needed
- Enable DNSSEC for the zone (image)
- Push zone successfully
- You will now have a “DS Records” section on the zone page (image)
- Upload these values to your Zone Registrar (image - label fields)
  - DS Record #, Key Tag, Algorithm, Digest Type, Digest
- Confirm values are saved at the Zone Registrar
- Check DNSSEC status of zone
  - ProVision GUI (image)
    - DNSSEC column
      - Means that DNSSEC has been enabled for the zone
    - DS column
      - Red X means DS keys have been generated only
      - Green AD means DS keys have been generated AND the Authenticated Data has been verified by the external server (DNS Admin setting)
  - External sites
    - http://dnssec-debugger.verisignlabs.com/
    - http://dnsviz.net/

For BIND server(s)

To enable DNSSEC on BIND9 you need to modify **named.conf.options** with following parameters in the **options { }** section:

```
dnssec-enable yes:
dnssec-validation yes:
dnssec-lookaside auto;
```

These parameters may already be enabled in some Linux distributions by default, so please confirm before making changes.

Your DNSSEC implementation may need other options for your environment - please contact support@6connect.com if you have ny

Please note that you will need to restart the BIND service after these changes.

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. We are evaluating how to properly integrate DNSSEC functions to ProVision for these platforms. Please contact support@6connect.com if you have feedback or specific questions.

DNS Audit Tools (Alpha)

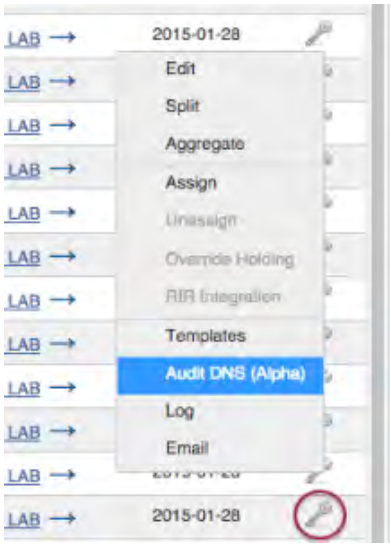
DNS Audit Tools (Alpha):

The DNS audit tools perform a simple audit of both forward and reverse DNS.

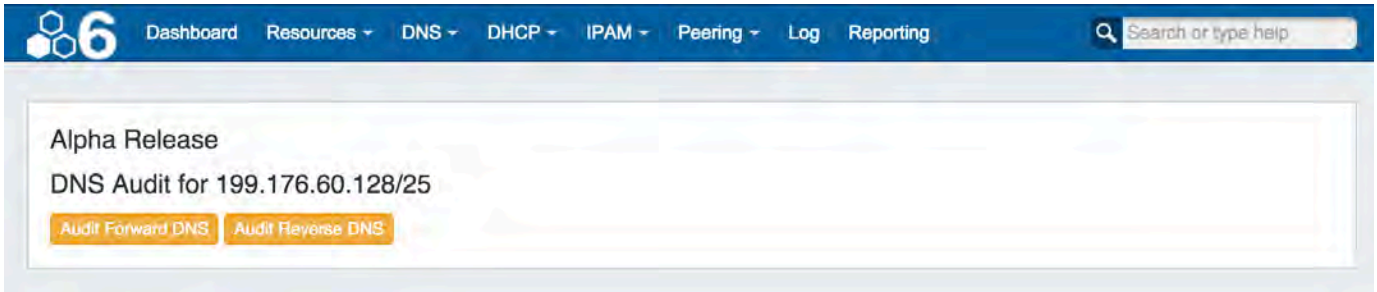
The tool set includes UI, API end points, and a command line interface. The audit results include the DNS as found in the 6connect ProVision database, the results from a resolver, and if there is a conflict in these two pieces of information.

Accessing the Audit Tools: UI

1) 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 for the block.



2) 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).

Accessing the Audit Tools: API

Access the tools via the API through the use of:

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

Audit DNS - Execute

URL	api/v1/api.php?target=auditDNS&action=execute
Description	Audits a DNS CIDR block
Returns	**Examples:** SUCCESSFUL: {"success":1,"message":"DNS Lookups Started."} ERROR: {"success":0, "message":"error message"}
Required Parameters	

Name	Type	Example	Description
type	STRING	forward	Type of DNS lookup. Valid values are "forward" or "reverse".
block	STRING	209.183.188/28	CIDR of the DNS block to audit
jobId	INTEGER	1424218758	Job ID

Example URL	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

Audit DNS - getStatus

URL	api/v1/api.php?target=auditDNS&action=getStatus
Description	Displays the audit results table information
Returns	**Examples:**

SUCCESSFUL:	{"state":"Completed","processName":"audit_reverse_dns.php","message":"Lookup process complete.","percentage":"1","processed":"16","processId":"27483","meta":[{"ip":"209.183.0","reverseDNSRecord
ERROR:	{"success":0, "message":"error message"}

Required Parameters

Name	Type	Example	Description
block	STRING	209.183.188/28	CIDR of the DNS block to audit
jobId	INTEGER	1424218758	Job ID.

Example URL	/api/v1/api.php?target=auditDNS&action=getStatus&block=209.183.188/28&jobId=1424218758

Accessing the Audit Tools: CLI

To access the Audit Tools via the command line:

For Forward DNS: "*php audit_forward_dns.php -b "* plus the CIDR you wish to audit

Example:
php audit_forward_dns.php -b 209.183.188/28

For Reverse DNS: *"php audit_reverse_dns.php -b "* plus the CIDR you wish to audit

Example:
php audit_reverse_dns.php -b 209.183.188/28

Additional Information

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

Importing Your Data

Preparing for Data Import

Before importing your data into ProVision, there are a few steps we recommend in order to make the import process as smooth as possible:

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 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.

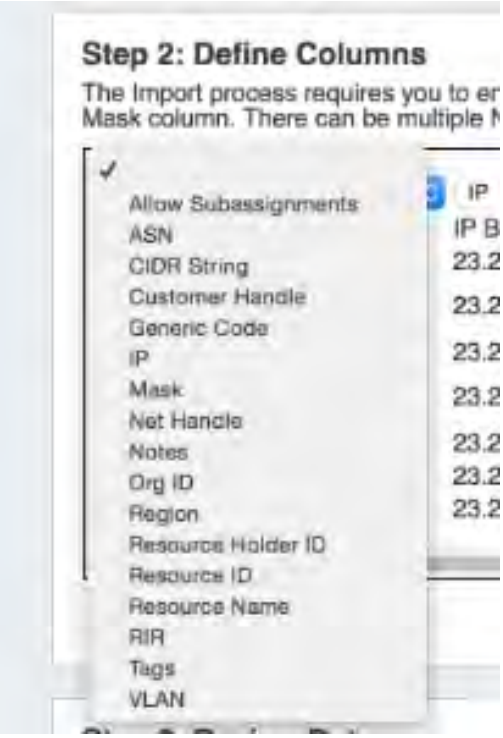
> **Data Encoding Format**
> To ensure correct importing of any special characters, make sure to use UTF-8 encoding for your CSV file!

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:

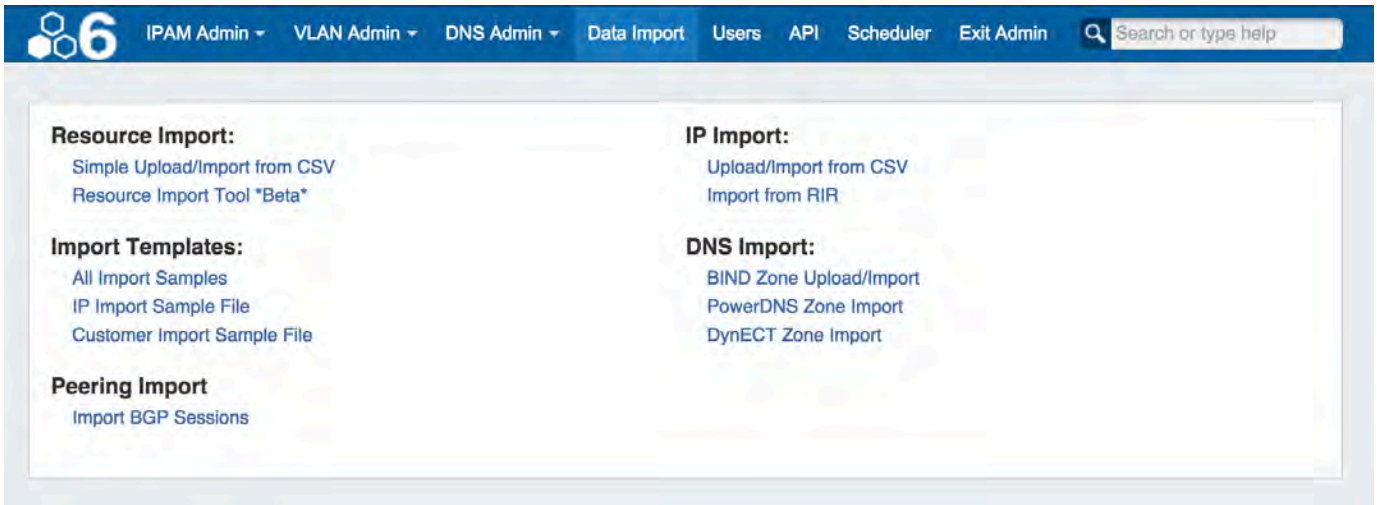
Click here to expand...



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.



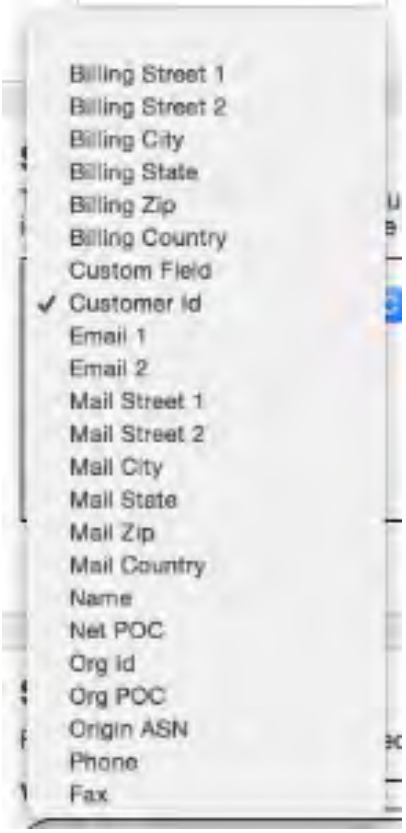
Data Import Overview: Which Import Tool Should I Use?

Resource Import

Simple Upload / Import from CSV: Use this tool if you have a simple .csv file of customer / contact information, such as Name, Address, Billing Address, Phone numbers, POC, etc. See Resource Import from CSV for more

details.

This tool supports the following fields:
Click here to expand...



Resource Import Tool: The Resource Import tool is designed for importing .csv files for any Resource type (company, physical devices, contacts, customers, etc), and allows for customized headers through integrating ProVision's Section fields. It also allows for data editing within the tool. Use the Resource Import Tool if you have resource data that does not fit under the Simple Upload / Import from CSV field options, and can equate that data to an existing or newly created Section.

> **Additional Information**
> A tutorial / walkthrough for this tool is available at the Resource Import Tool page.

Import Templates

This section links sample files for selected ProVision import tools. These may be viewed as an example of .csv setup for their respective importers, or used for tutorials.

They are:

All Import Samples - Links to the CSV import templates page of the documentation, showing all available import template .csv files.

IP Import Sample File - Download the IP-import-sample_v1.csv file, used in the IP Import: Upload / Import from CSV Tool.

Customer Sample Import - Download the customer-import-sample.csv file, used in the Resource Import: Simple Upload / Import from CSV Tool.

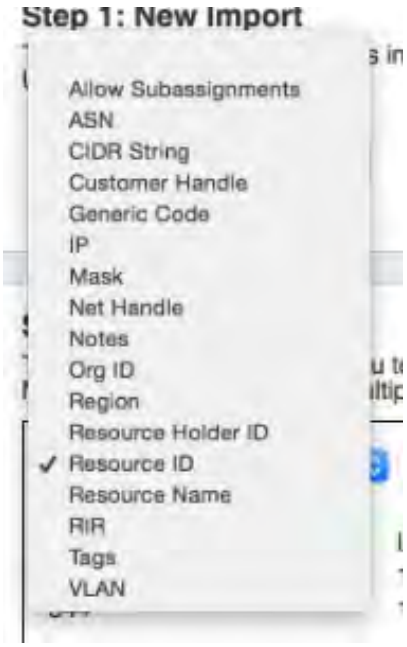
Peering Import

Import BGP Sessions: This section imports peering sessions from a selected exchange and router. Routers must be created prior to using this importer. See Adding Routers and Importing Sessions for additional information.

IP Import

Upload / Import from CSV: Use this tool if you have a .csv file of IP block information, such as CIDR, Mask, ASN, RIR, etc. This tool supports the following fields:

Click here to expand...



Import from RIR: This tool automatically lookups your ARIN or RIPE information based on the IP address you are connected to. Once it identifies the blocks assigned to your company, you can import both 1918 aggregates as well as public IP space from ARIN and RIPE. See Import Aggregate Blocks for more detail on this tool.

DNS Import

BIND Zone Upload / Import: Imports DNS zones 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. This is the simplest and most commonly used import method. Refer to Import DNS Zones for more information.

PowerDNS Zone Import: This tool is available after configuring a PowerDNS server with a MySQL backend. The PowerDNS Import connects to the selected server and imports all zones.

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. See Configuring DynECT Support for information on setting up a DynECT server in ProVision.

Additional Information

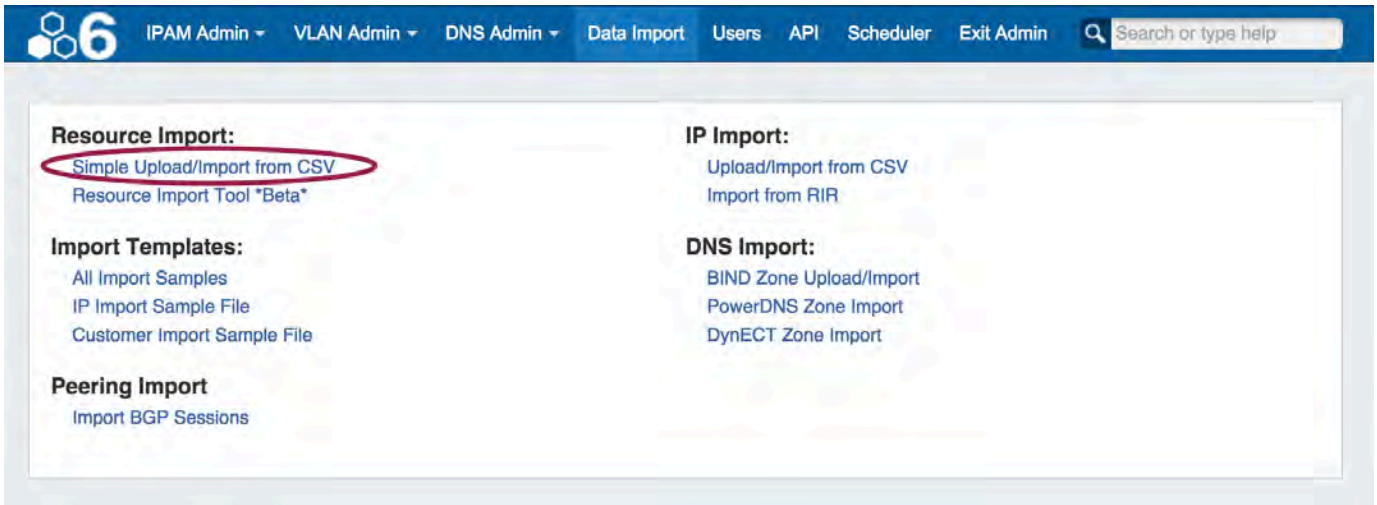
For more details, see:

- Resource Import from CSV
- Resource Import Tool
- Import Sessions
- IP Import from CSV
- Import Aggregate Blocks
- Import DNS Zones

Resource Import from CSV

Simple Upload / Import from CSV

The Simple Upload / Import from CSV tool is used if you have a simple .csv file with customer / contact information, such as Name, Address, Billing Address, Phone numbers, POC, etc. It is accessed from the **Data Import Tab** from the **Admin** section of ProVision.



To import your customer / contact information, follow the following steps:

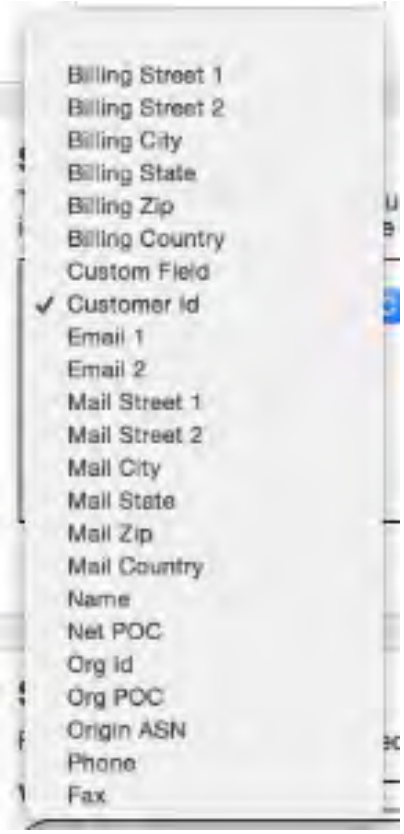
- Before you Begin: Prepare your Data for Import
- Step 1: Create a new Resource / Customer Import Job
- Step 2: Define Columns
- Step 3: Reviewing Data
- Step 4: Execute Import

Before you Begin: Prepare your Data for Import

The Simple Upload / Import from CSV tool requires only a .csv file for importing.

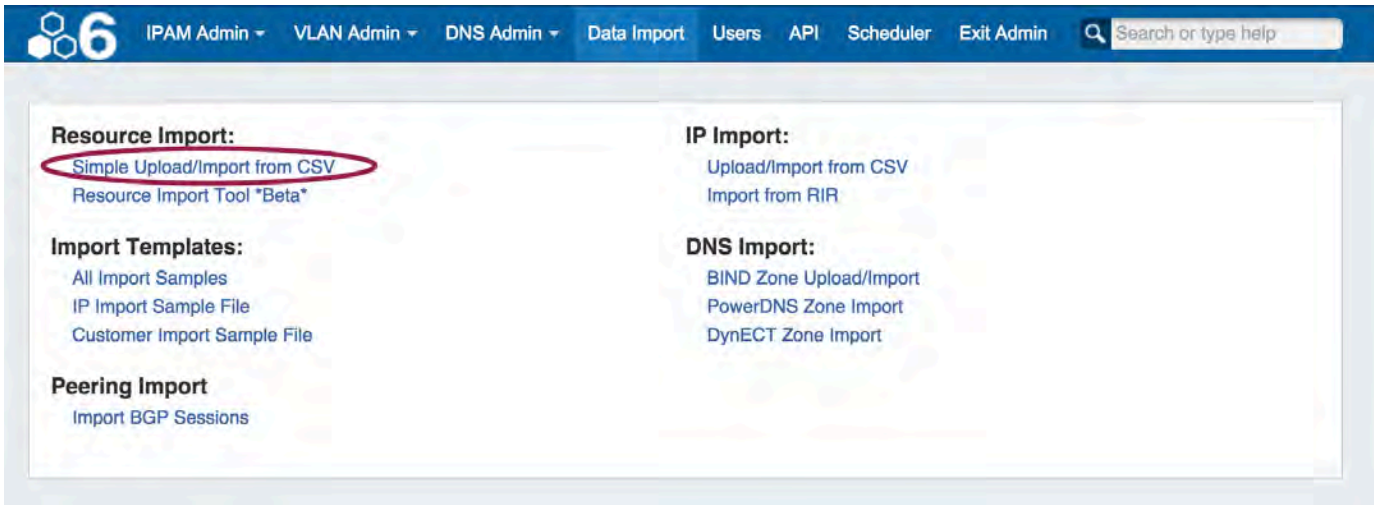
Be sure to review "Preparing Data for Import" on the Importing your Data page before you begin. Verify that your .csv is correctly parsed and cleanly formatted with your customer/contact information (using your choice of the available field list shown below), and is UTF-8 encoded for best results.

This tool supports the following fields:
Click here to expand...



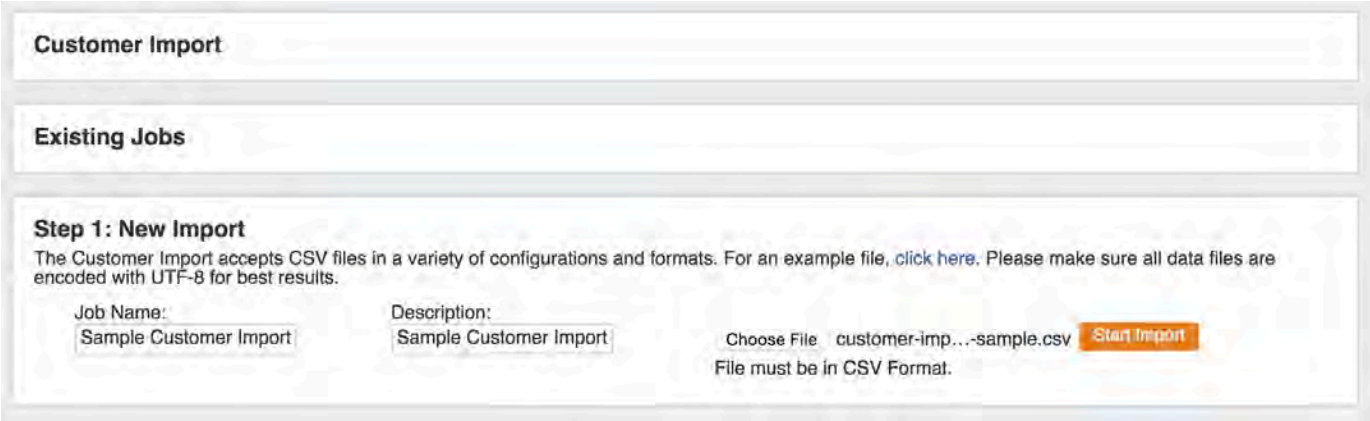
Step 1: Create a new Resource / Customer Import Job

Navigate to the **Data Import** Tab from the **Admin** button. Select "Simple Upload / Import from CSV" under "Resource 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.

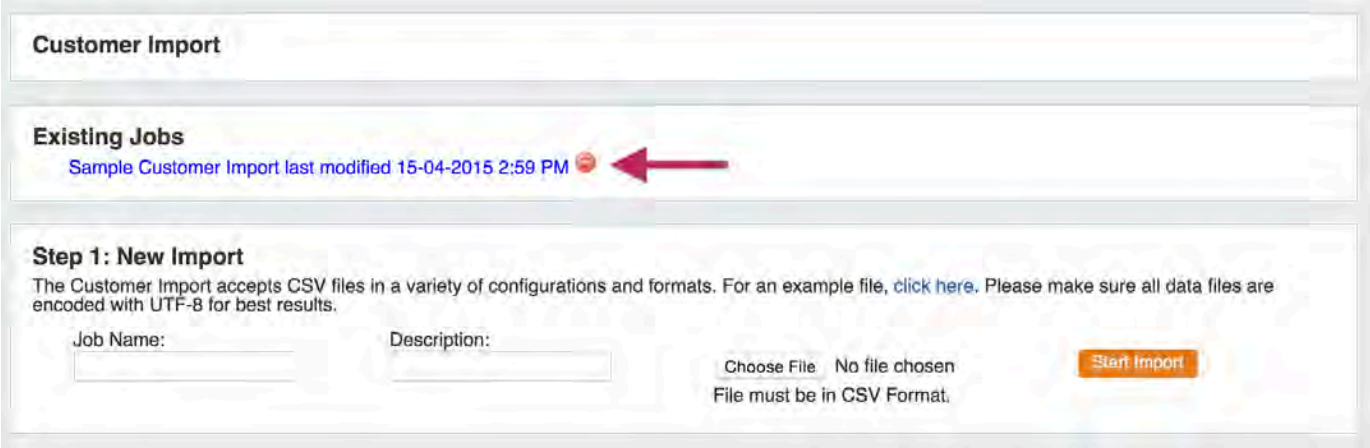
Select the .csv file that you prepared above by selecting the "Choose File" button, and browsing to the correct file location. Then hit "Start Import".



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 (by clicking on the link) and the next step will appear on the page.

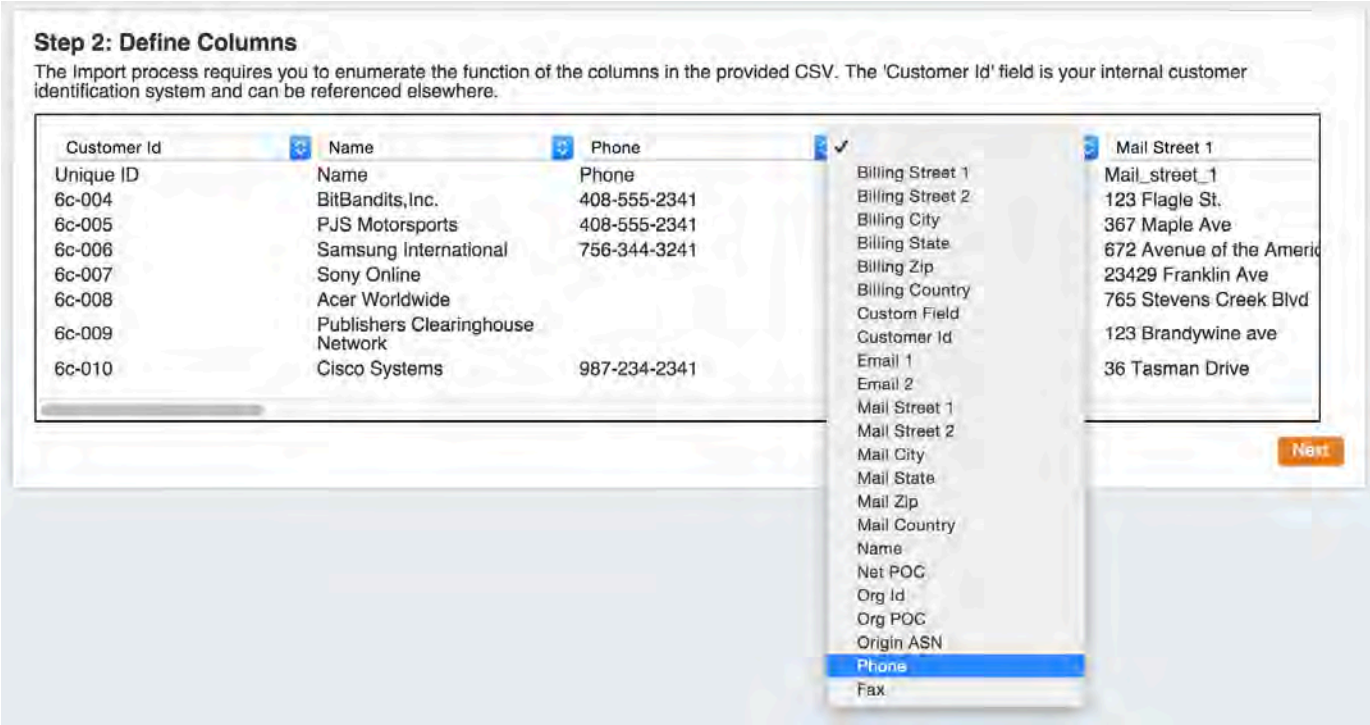


Step 2: Define Columns

Using the dropdown menu above each data field, select the appropriate definition for each of the imported columns. **Phone** and **Fax** may have multiple columns associated with the data. Other columns which do not apply under the available definitions should be left as blank, and will be skipped during the upload process.

Make sure that you have defined all desired fields by using the scroll bar below your data to view additional columns.

When completed, hit "Next".

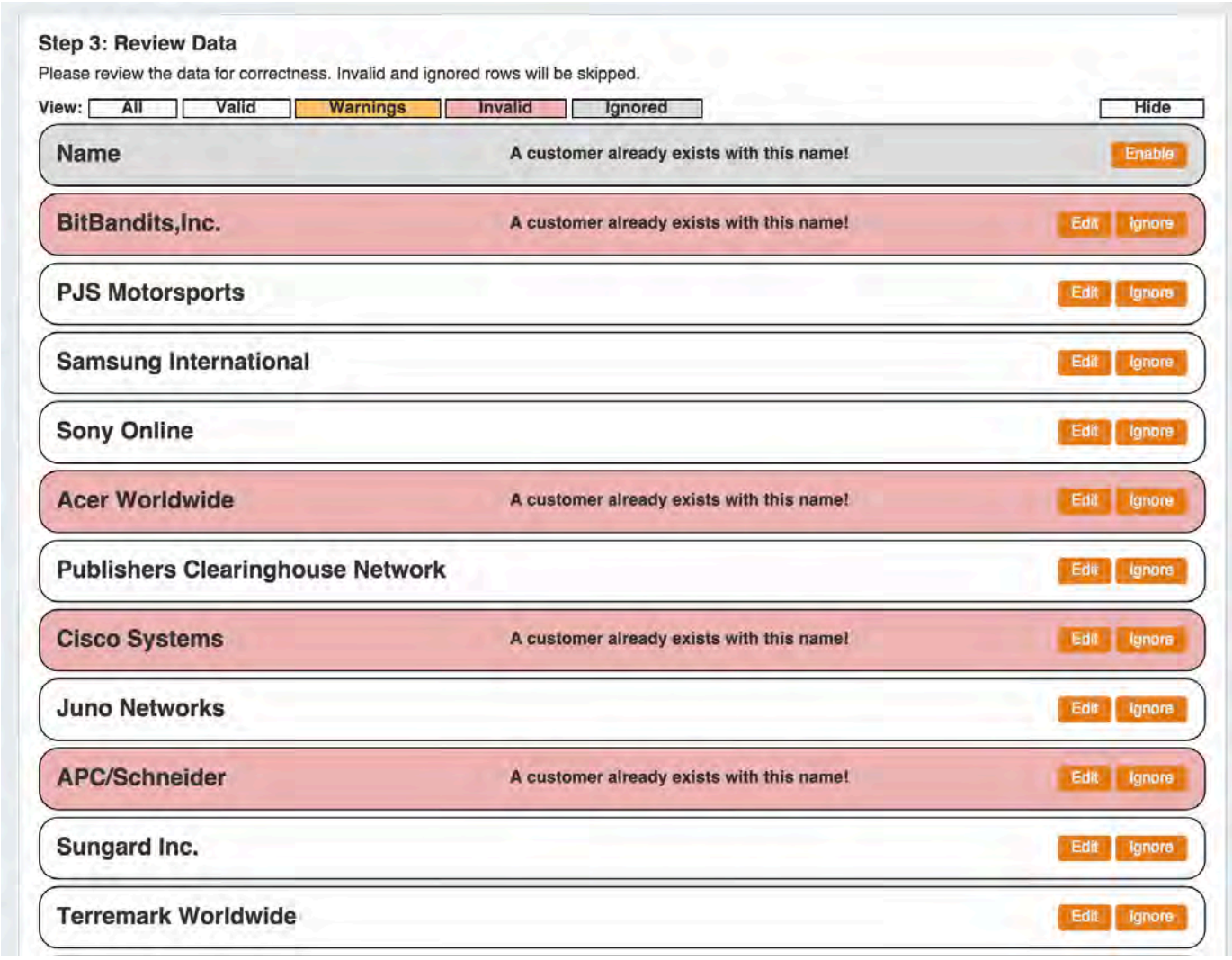


Step 3: Reviewing Data

After supplying the file and defining columns, a review step is provided. Records with errors will show as color coded, and can be filtered to be viewed by All, Valid, Warnings, Invalid, or Ignored.

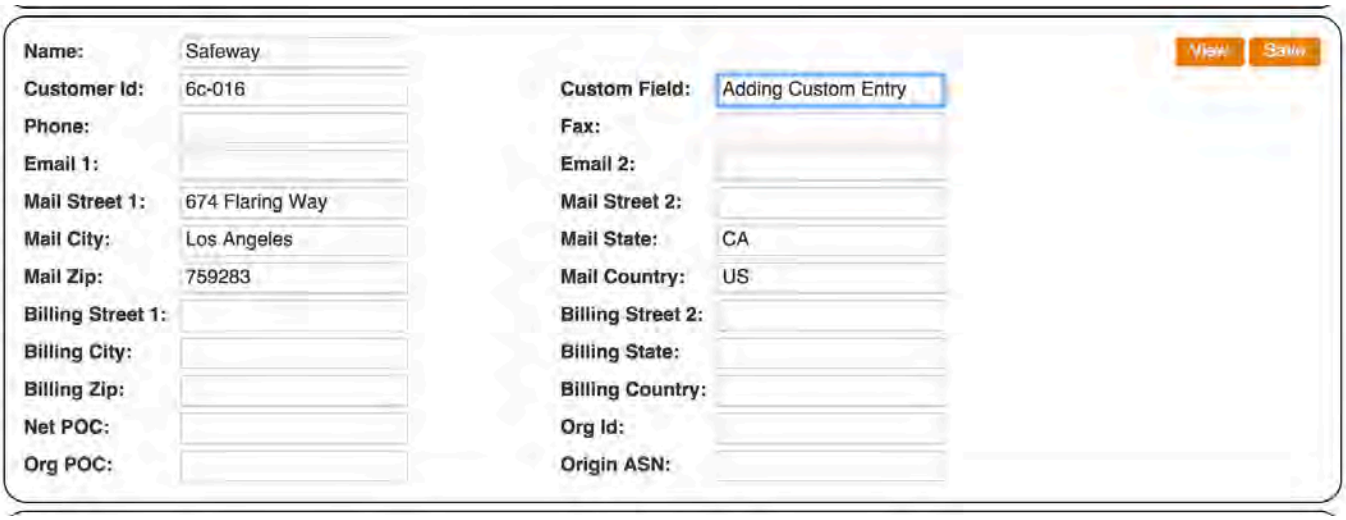
From here, the records can be edited or ignored. Select "Ignore" for records that you do not wish to import at this time. Records that are Ignored or Invalid will automatically be skipped.

> **Header Rows**
> If your .csv has a header row as the first line, that row will be shown as the first record in review data as well. Simply click "Ignore" on the first record to disregard the row.



Hitting the "Edit" button for the record provides options to change or add information for available fields.

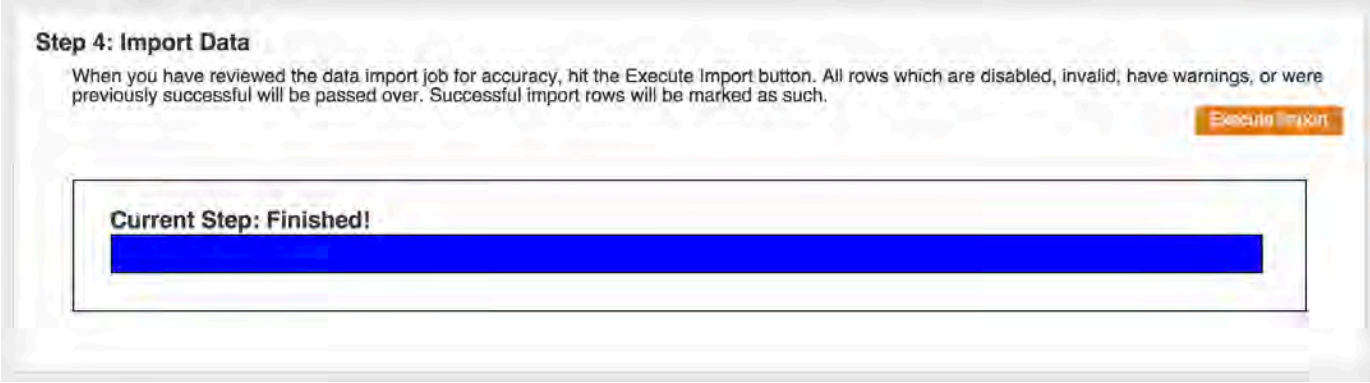
After editing, hit "Save", and continue reviewing / editing data as desired.



Step 4: Execute Import

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.



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.

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:

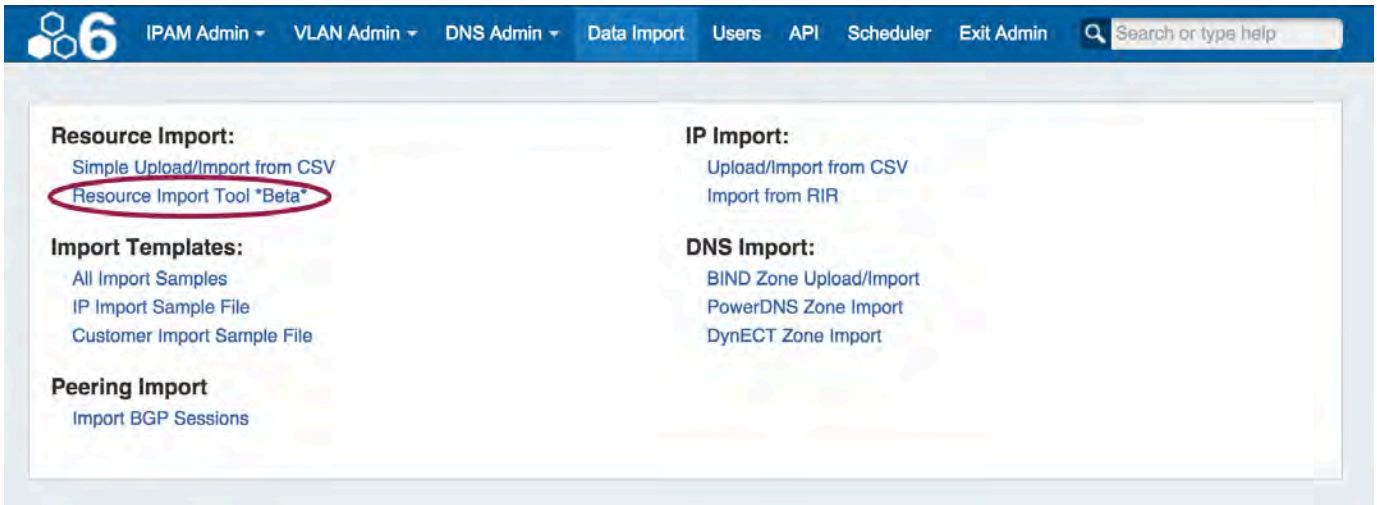
- The .csv document you wish to import saved with UTF-8 encoding. Windows, Mac, and Linux type .csv files are supported.
- A header row for the data in the .csv.
- The .csv file should be "clean", that is, only contain the data to be imported and a header row for that data.
- A Section created in ProVision 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.

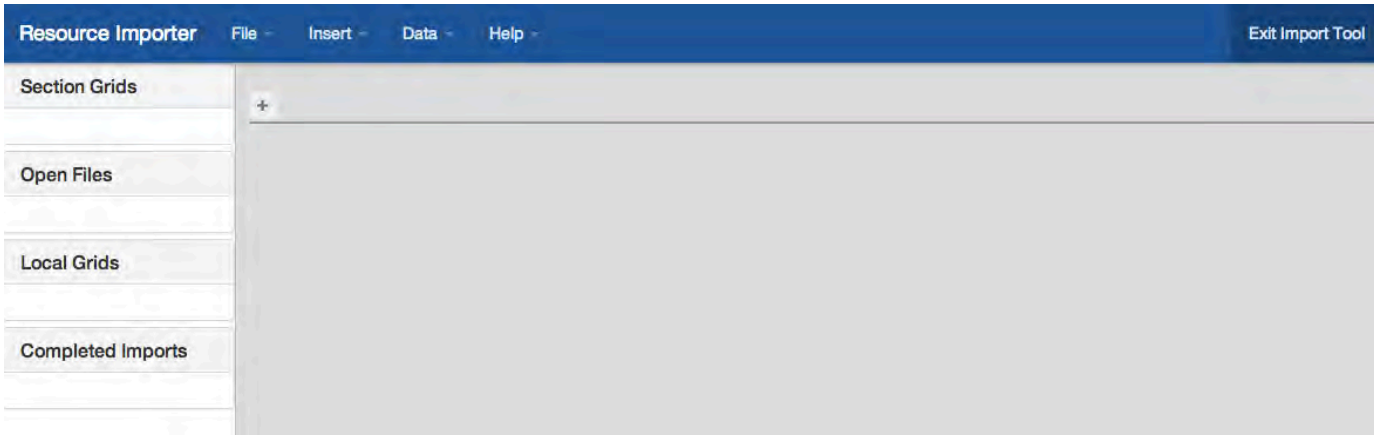
Opening the Resource Import Tool

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

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.

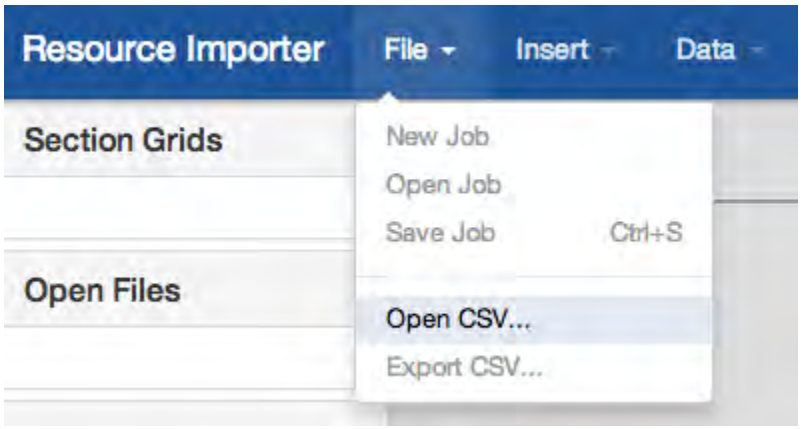
For this tutorial, we will be using the Contact Import Sample .csv available on the Import Templates page as an example, and associating it to an existing Section called "Contact" having the fields: First Name, Last Name, Email, 2nd Email, Phone, 2nd Phone, Mobile Phone, Role, and Time Zone. To create 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.

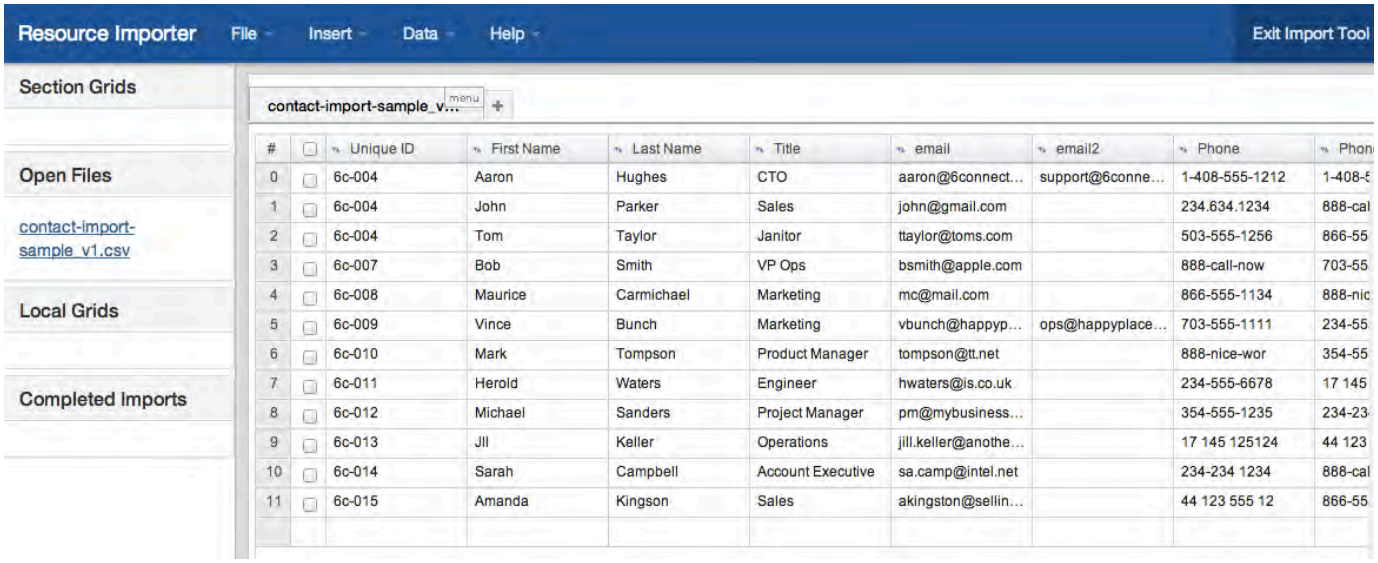
When you are ready to begin, open the Resource Importer and proceed to Step 1.

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:



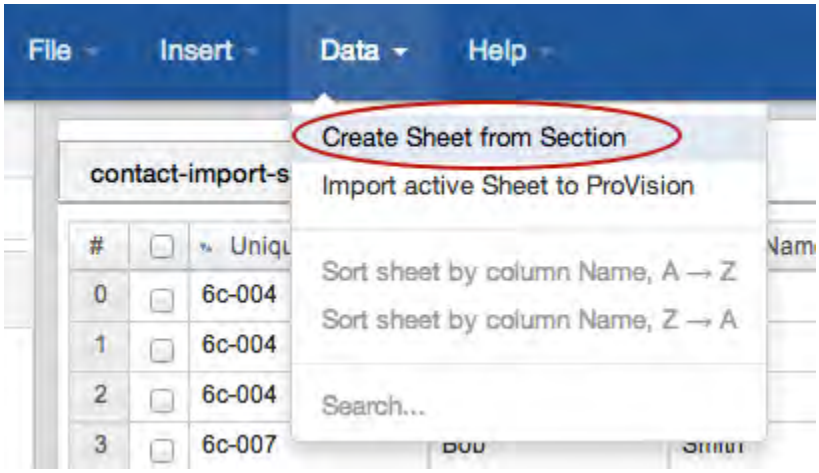
After opening your .csv grid, proceed to Step 2 - Open a template grid from an existing Section

Resource Importer Walkthrough - Step 2

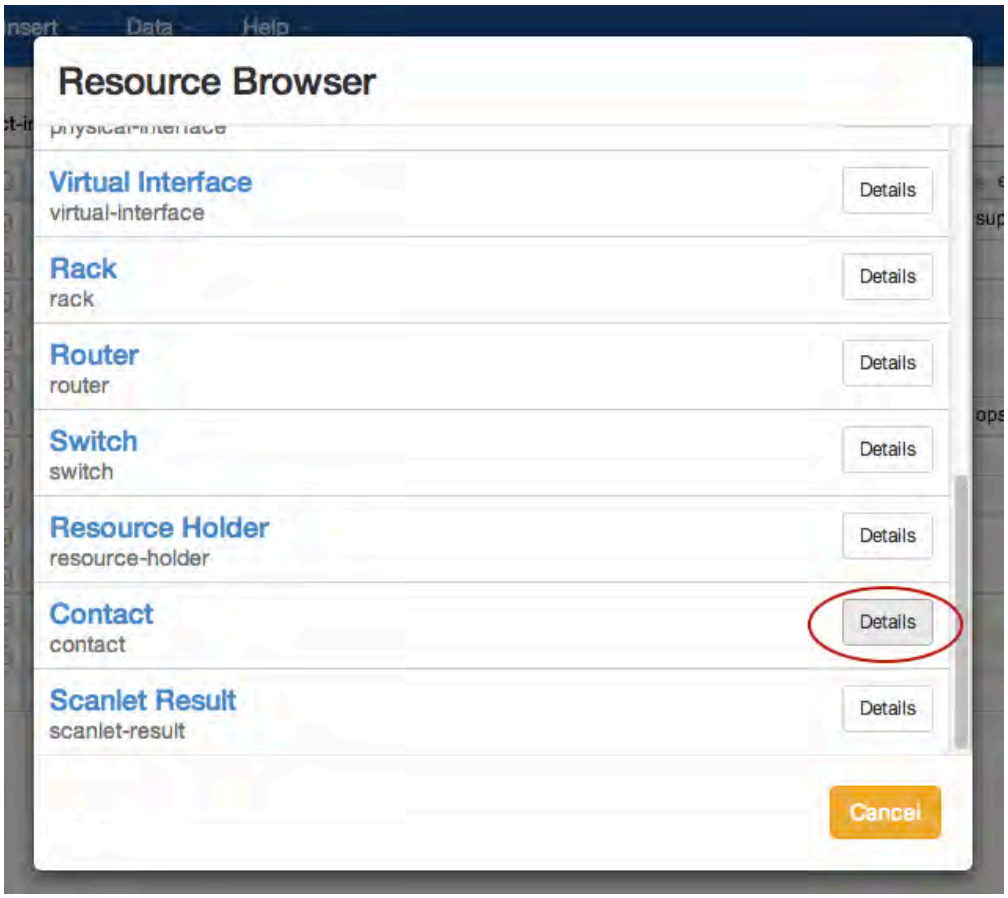
Importing Resources

Step 2: Open a Template Grid from an existing Section

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

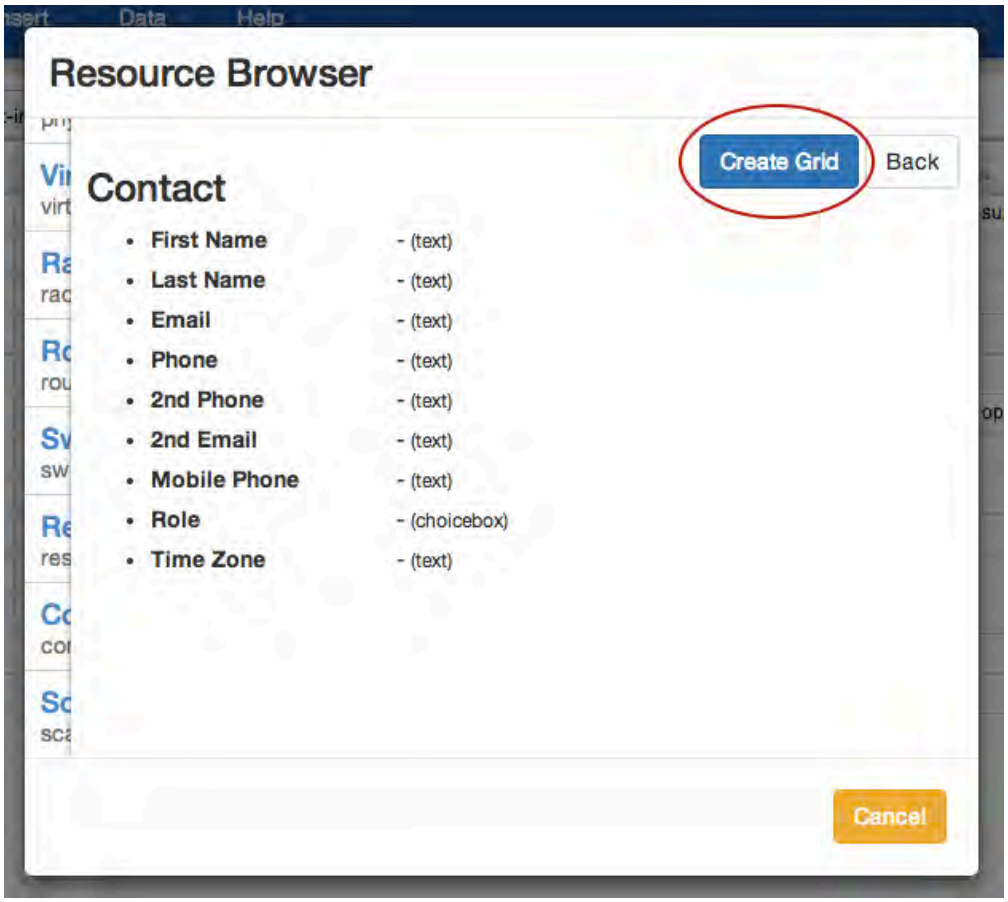


The Resource Browser will pop up, showing the list of Sections currently available in Provision. Clicking on the "Details" button will show the fields for that Section.

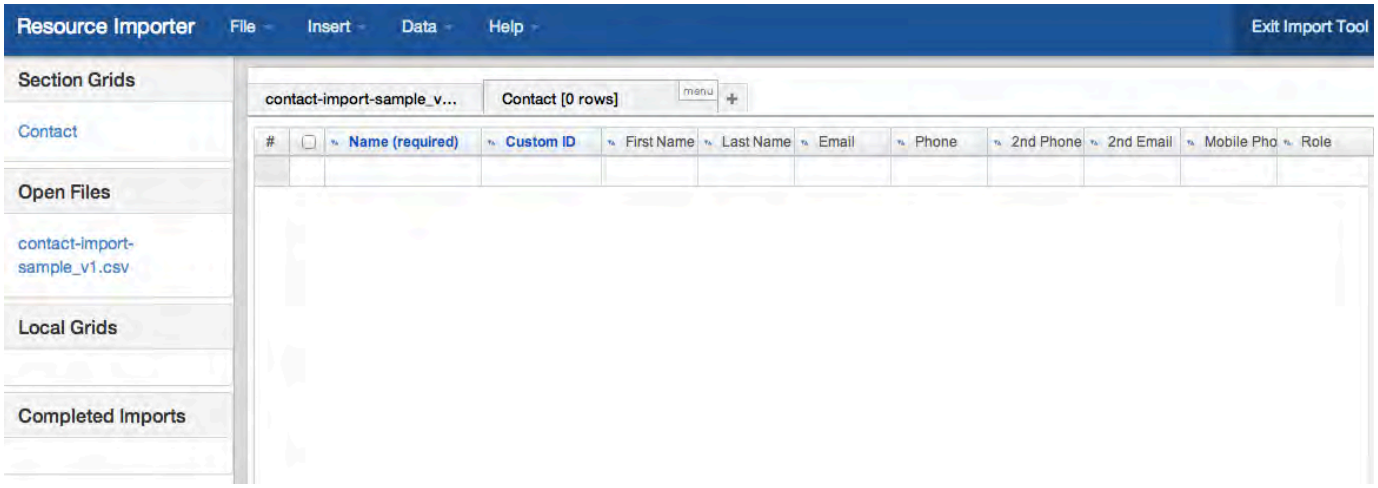


Verify that the Section and available fields match the type of data you are trying to import. In this case, the Section "Contact" has the fields that correlate to our spreadsheet data.

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:

#		Unique ID	Last Name	First Name
0		6c-004	Hughes	Aaron
1		6c-004	Parker	John
2		6c-004	Taylor	Tom
3		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

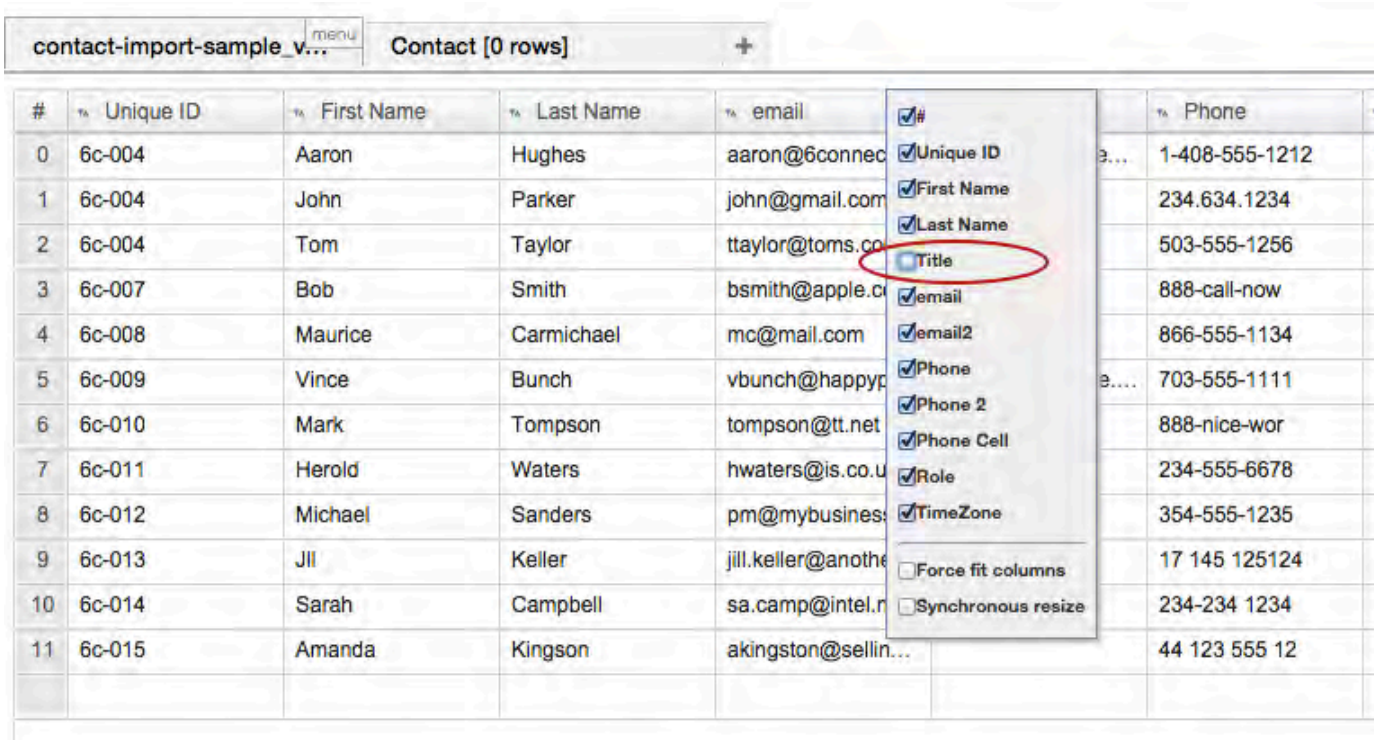
As you may have noticed in Step 3, with this example we have a couple of columns that don't quite match up to the Section Grid. The "Title" column in the .csv data is an additional column we are not tracking in our Section. Also, although we have a "First Name" and "Last Name", we are missing a data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

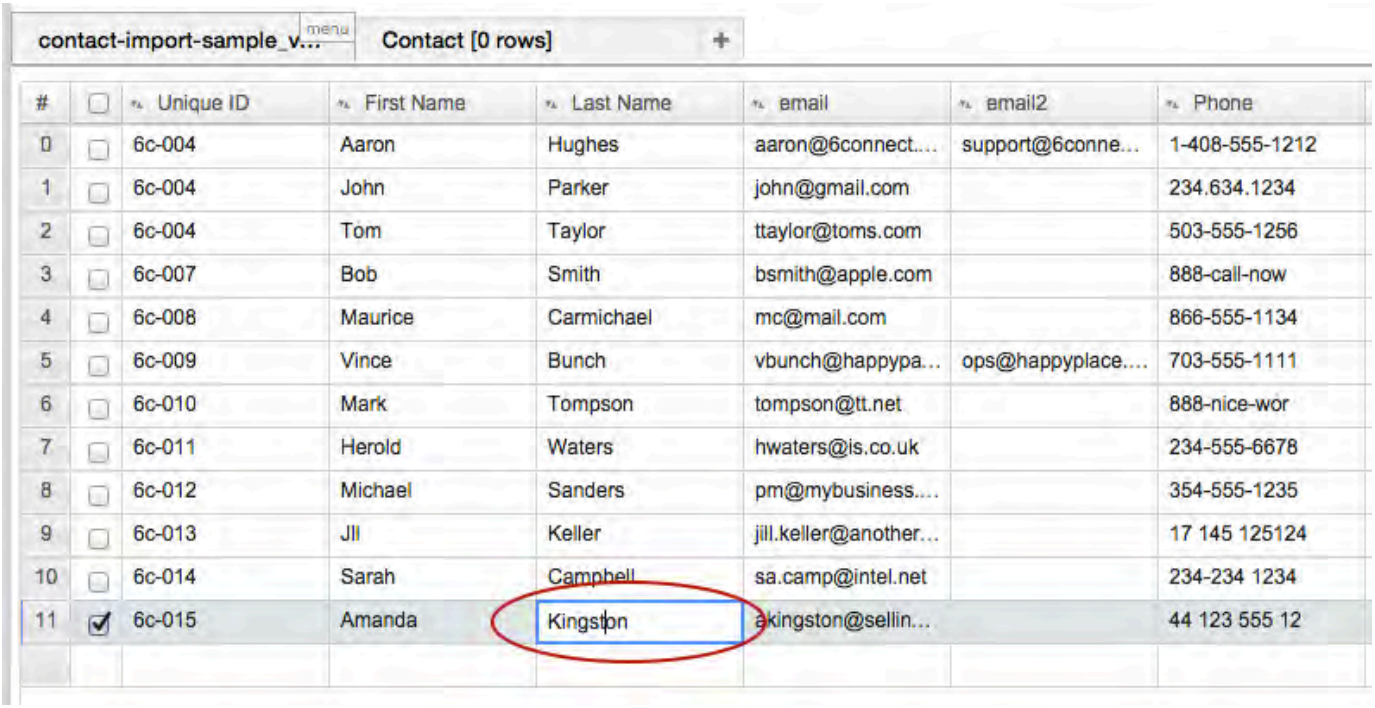
To hide extraneous column information:

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".



To Edit Data in the Resource Importer

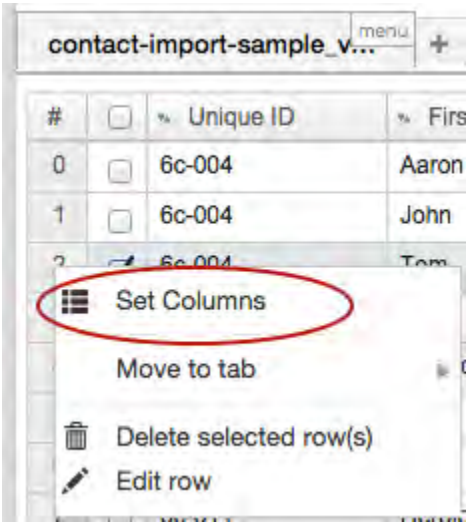
Data in the grids can be edited directly by clicking on the cell(s). In our example, we can see that "Amanda Kingson" should really be "Amanda Kingston". Let's fix that! Click in the cell, type in the edit you wish to make, and then click outside of the cell to exit edit mode. To edit a full row of data,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



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".

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	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	Jll	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

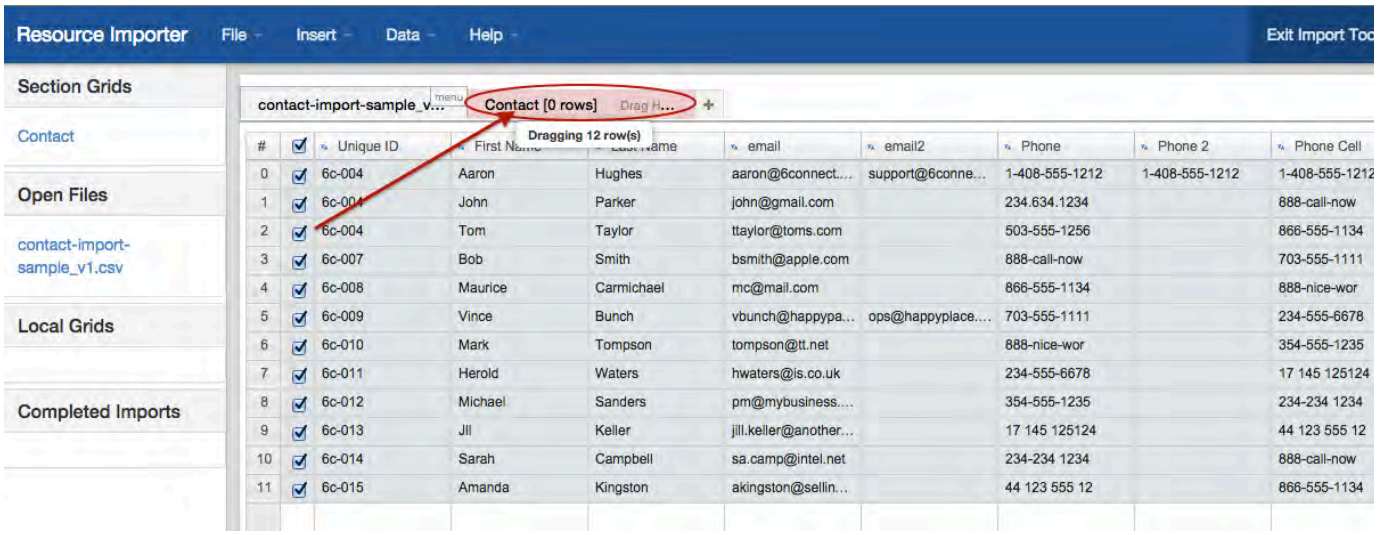
Resource Importer Walkthrough - Step 5

Importing Resources

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

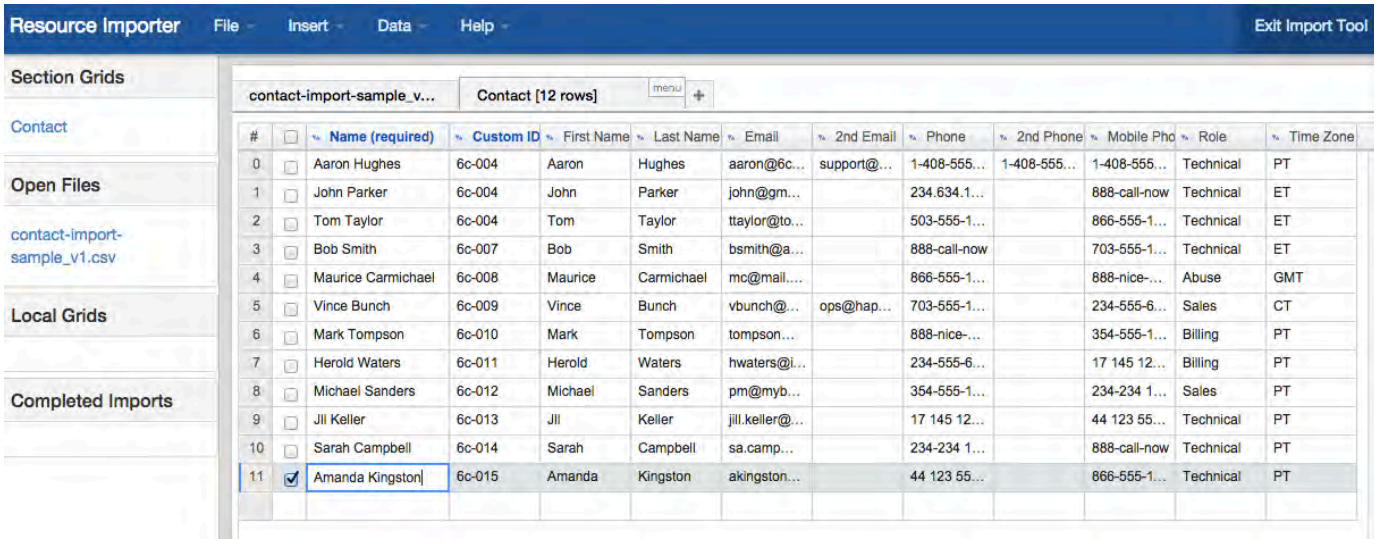
Once you have set the columns to match exactly between the .csv Grid and the Section grid, it's time to pull in the data from one to the other.

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.



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.



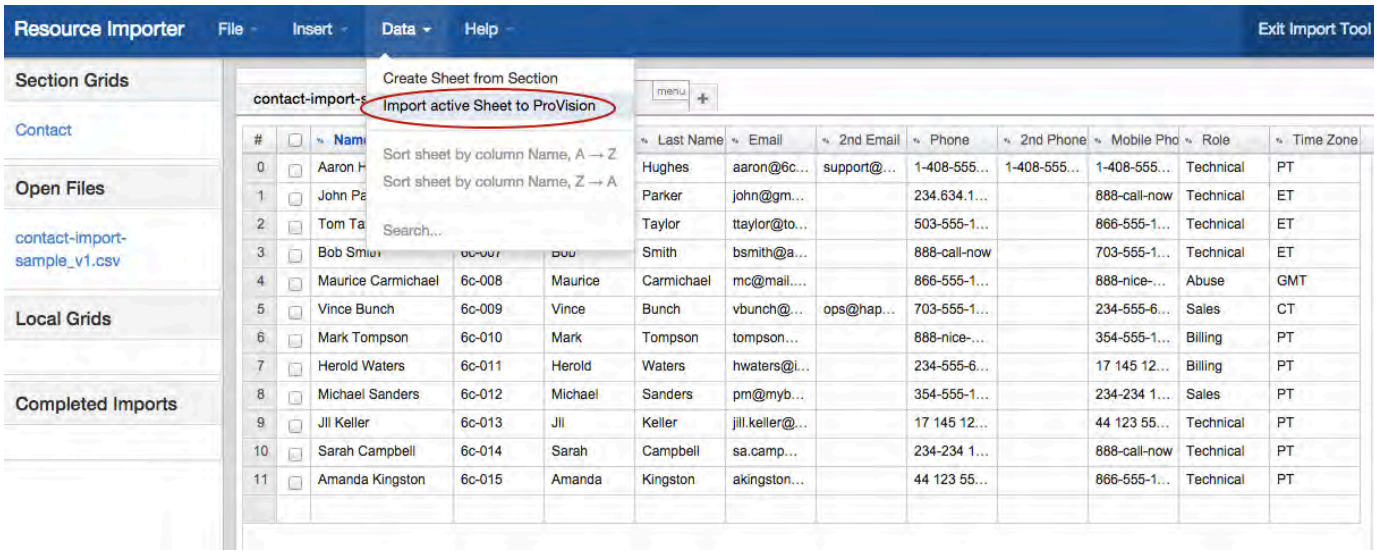
After moving your data into the Section grid, proceed to Step 6 - Importing into ProVision.

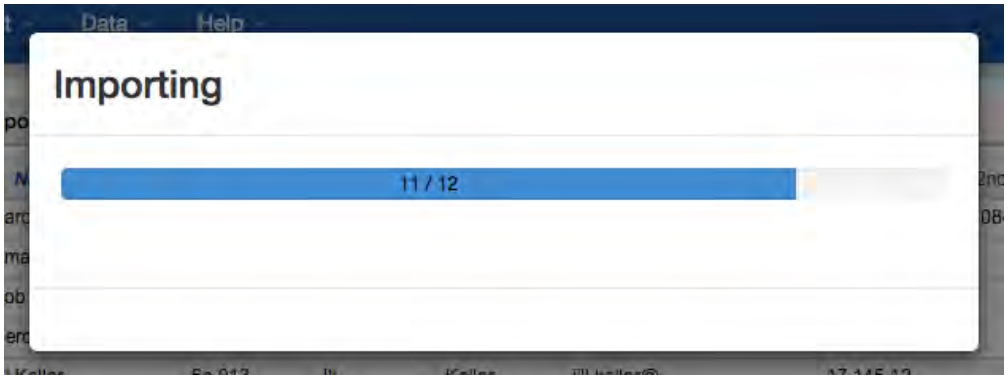
Resource Importer Walkthrough - Step 6

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.





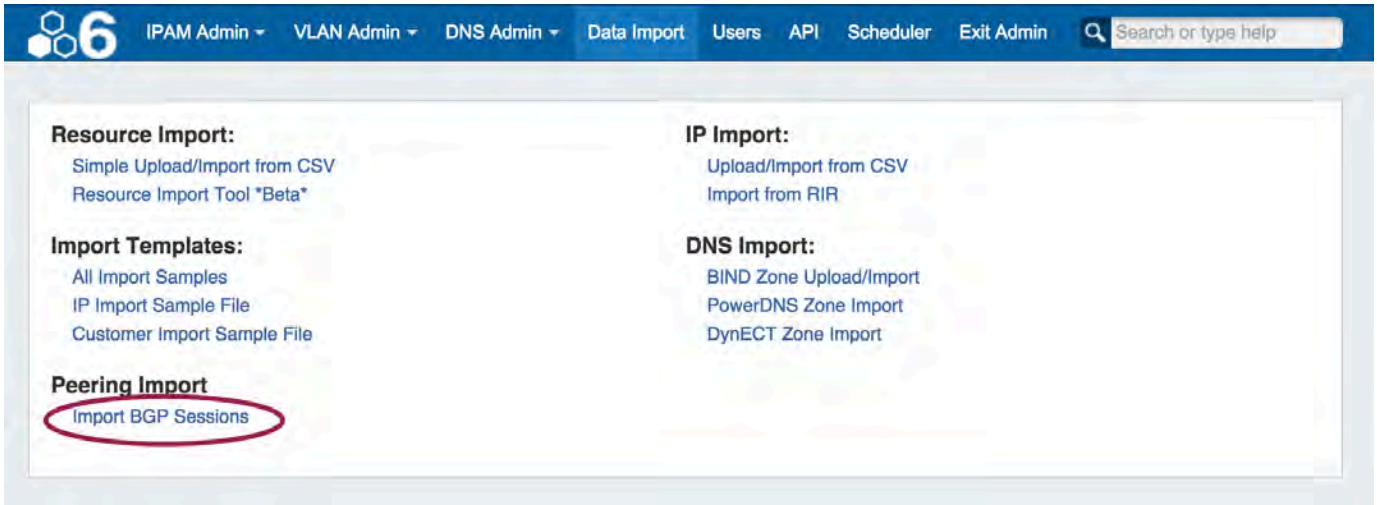
Import Sessions

Importing Sessions

Peering Import

Importing peering sessions requires Admin-level permissions, and is accessible only from the Admin section of ProVision.

From the Admin section of ProVision, navigate to the **Data Import Tab.** Under Peering Import, select **Import BGP Sessions.** This will take you to the Peering Import section of ProVision.



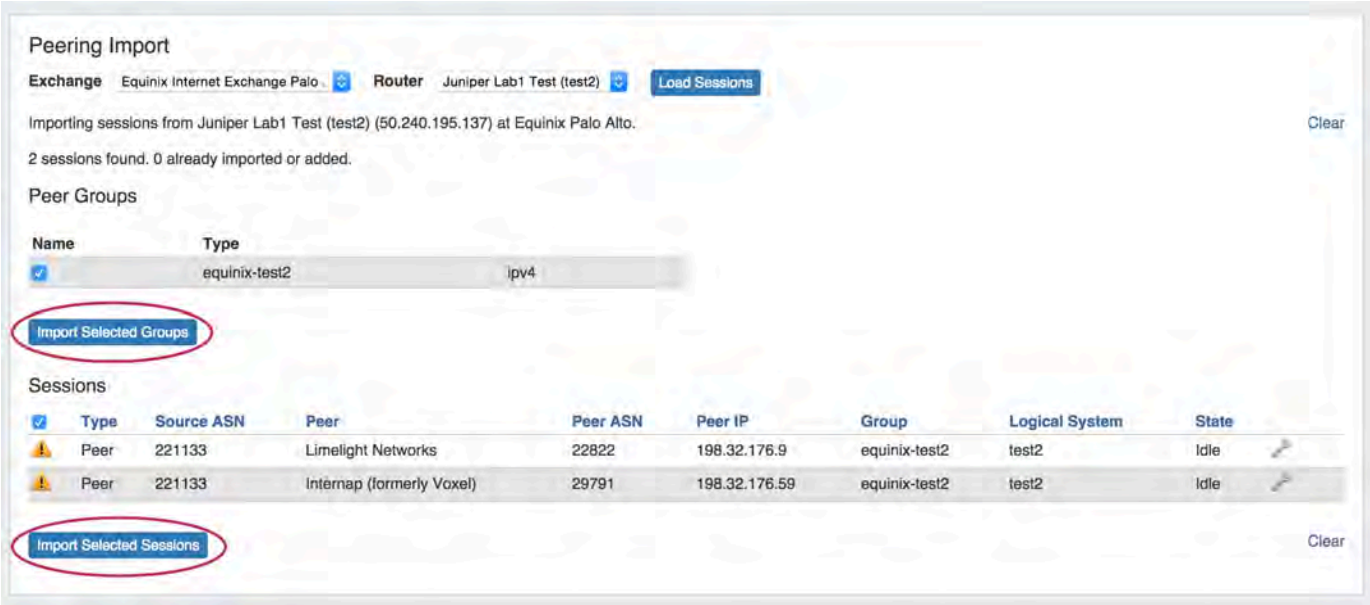
First, select the desired exchange and router. Routers with Logical Systems information will show up as the router name with the Logical System info in parenthesis (e.g. "Juniper (test)"). Then click "Load Sessions".

Peer Group and Sessions will then display below your selections.

Select Groups and Sessions

The available peer Groups and Sessions will display below your selected exchange and router. 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 (or the check box at the top to select all sessions) and click "Import Selected Sessions". Successful imports will then display with a green check mark at the beginning of the row.



The next step is to configure and manage your sessions.

IP Import from CSV

Simple Upload / Import from CSV

The Upload / Import from CSV tool is used if you have a simple .csv file with IP block information, such as CIDR, Mask, ASN, RIR, etc. It is accessed from the **Data Import Tab** from the **Admin** section of ProVision.

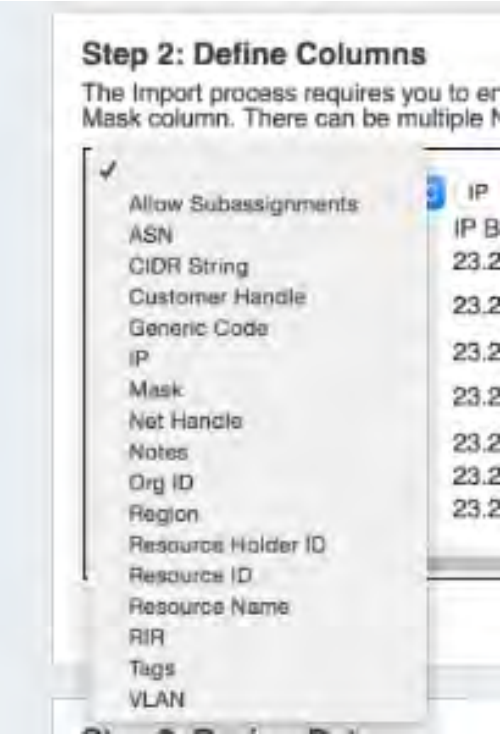
To import your IP block information, follow the following steps:

Before you Begin: Prepare your Data for Import

The Upload / Import from CSV tool requires only a .csv file for importing.

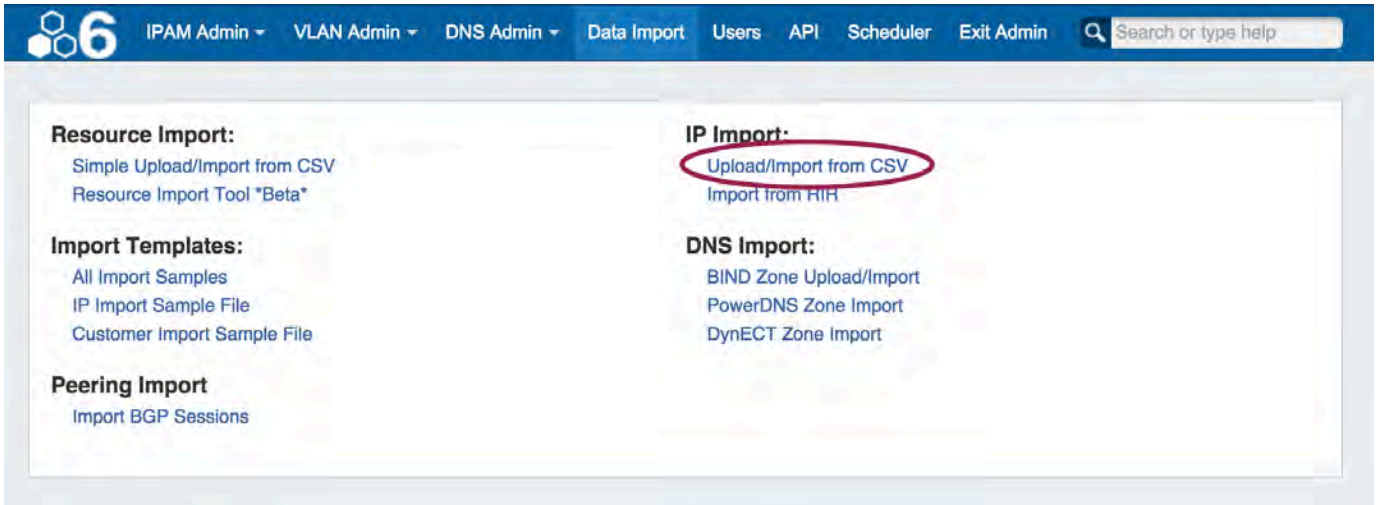
Be sure to review "Preparing Data for Import" on the Importing your Data page before you begin. Verify that your .csv is correctly parsed and cleanly formatted with your IP block information (using your choice of the available field list shown below), and is UTF-8 encoded for best results.

This tool supports the following fields:
Click here to expand...



Step 1: Create a new Resource / Customer Import Job

Navigate to the **Data Import** Tab from the **Admin** button. Select "Upload / Import from CSV" under "IP Import".



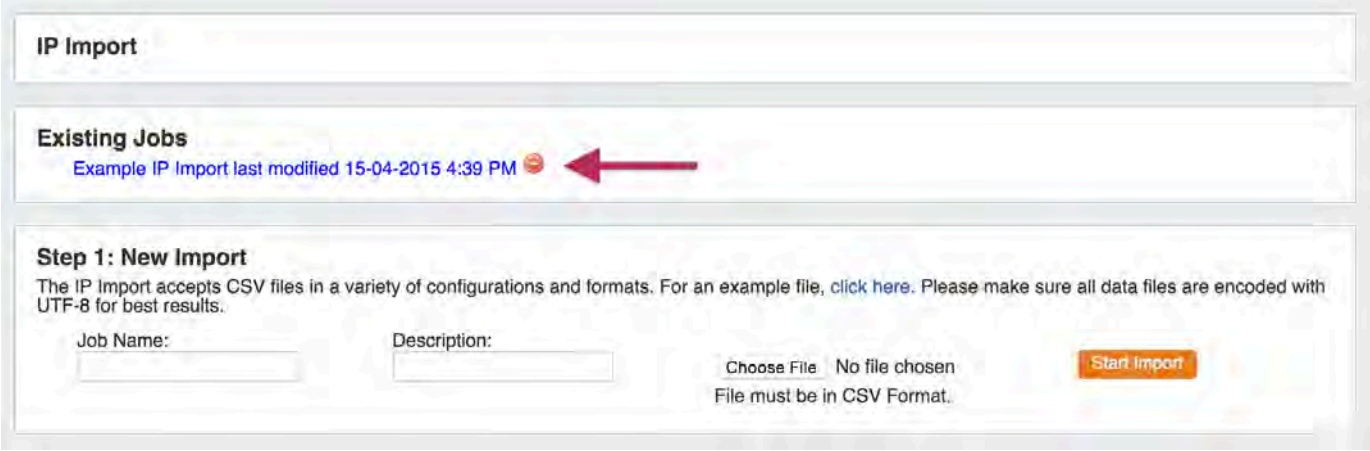
Create a Job Name and Description for the import. This is especially useful to keep track of progress in cases the data arrives from multiple sources, or will require multiple stages of manual review.

Select the .csv file that you prepared above by selecting the "Choose File" button, and browsing to the correct file location. Then hit "Start Import".

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 (by clicking on the link) and the next step will appear on the page.

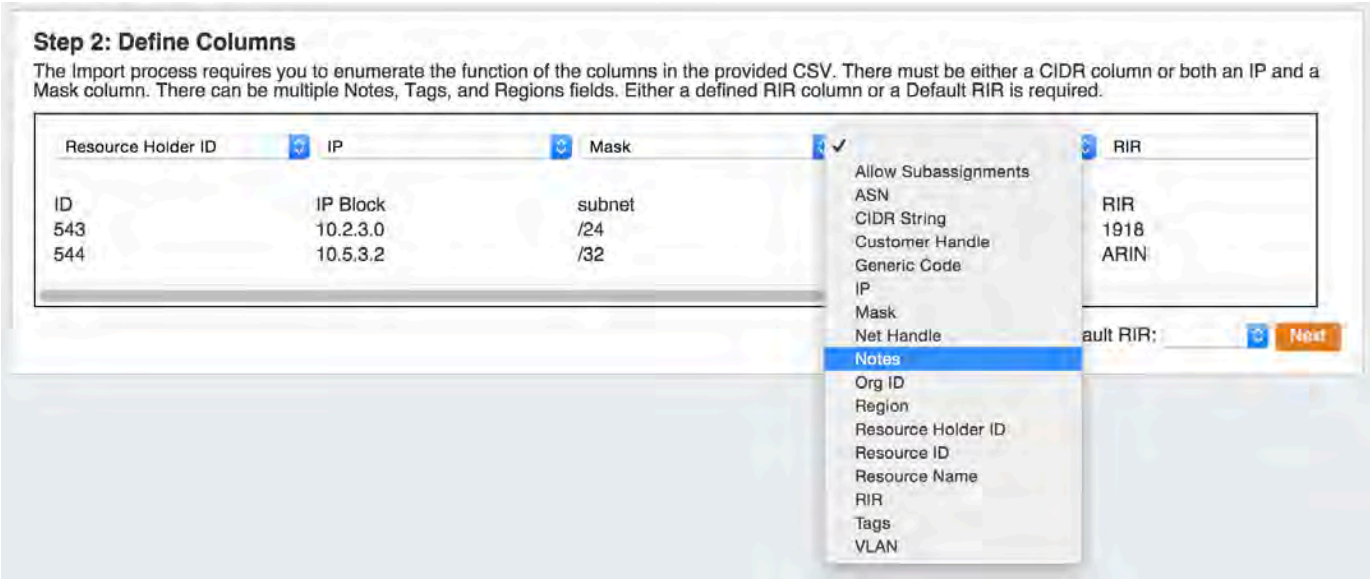


Step 2: Define Columns

Using the dropdown menu above each data field, select the appropriate definition for each of the imported columns. **CIDR** or **IP/Mask** fields must be provided. **Notes, Tags,** and **Regions** may have multiple columns associated with the data. If you do not have a defined **RIR** column, you must select a default RIR to associated the blocks to. Other columns which do not apply under the available definitions should be left as blank, and will be skipped during the upload process.

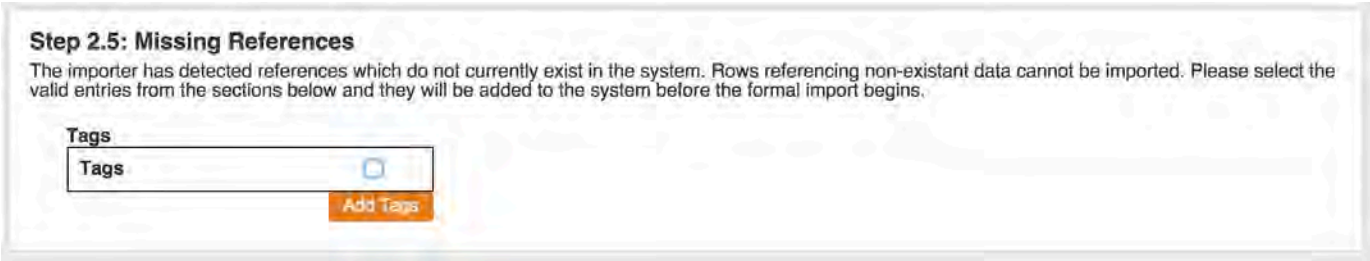
Make sure that you have defined all desired fields by using the scroll bar below your data to view additional columns.

When completed, hit "Next".



Step 2.5: Missing References

If references exist in your data that do not currently exist in ProVision (such as new tags or resources) the import tool will alert you to the missing references, and give you the option to create those references in ProVision. If you wish to add these elements, select the checkboxes next to the element and click the "Add (Element name)" button below.



In this example, we get a notification that we are missing a tag named "Tags". However, we know by looking at our data that is simply an item that was part of the header line in the .csv, so we want to bypass this step.

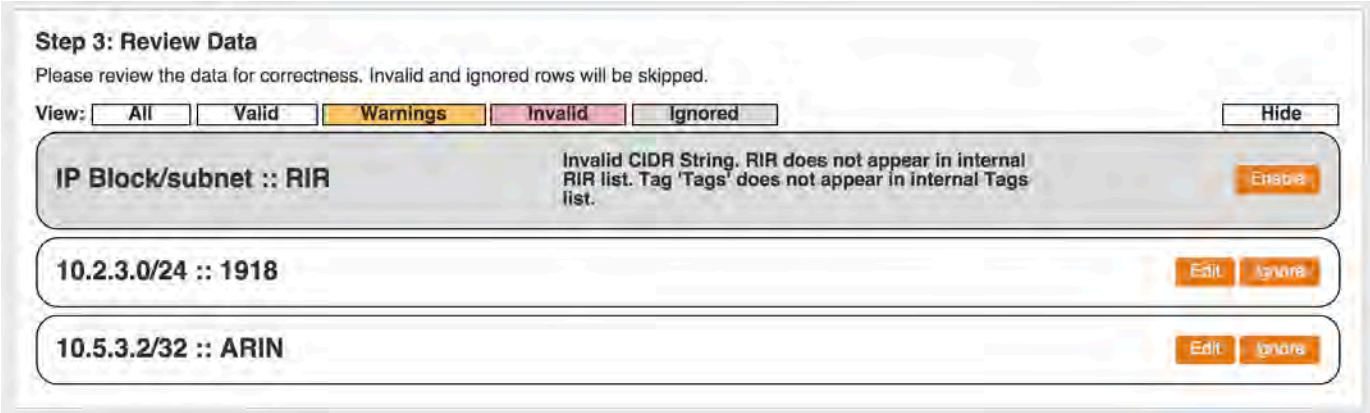
> **Header Rows**
> If your .csv has a header row as the first line, that row may give information that produces a missing reference notice. If this is the case, simply uncheck that item and add only the desired references.

Step 3: Reviewing Data

After supplying the file and defining columns, a review step is provided. Records with errors will show as color coded, and can be filtered to be viewed by All, Valid, Warnings, Invalid, or Ignored.

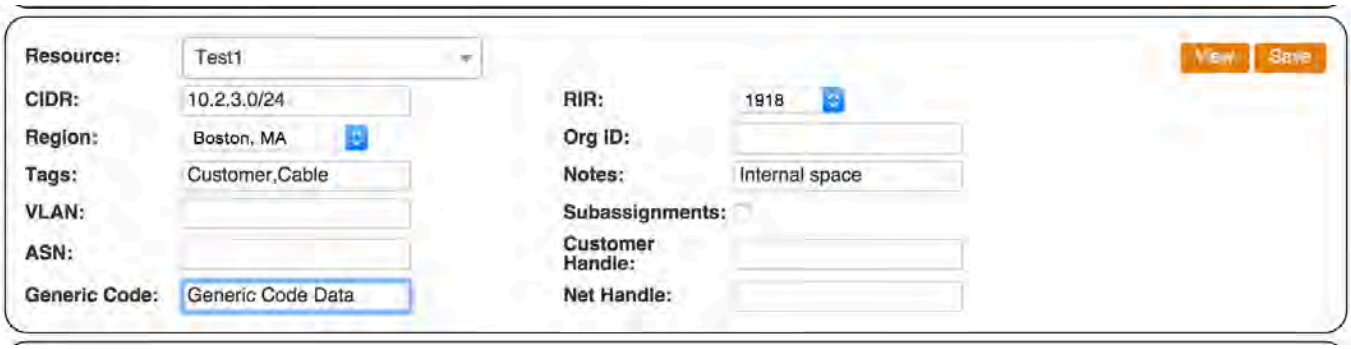
From here, the records can be edited or ignored. Select "Ignore" for records that you do not wish to import at this time. Records that are Ignored or Invalid will automatically be skipped.

> **Header Rows**
> If your .csv has a header row as the first line, that row will be shown as the first record in review data as well. Simply click "Ignore" on the first record to disregard the row.



Hitting the "Edit" button for the record provides options to change or add information for available fields.

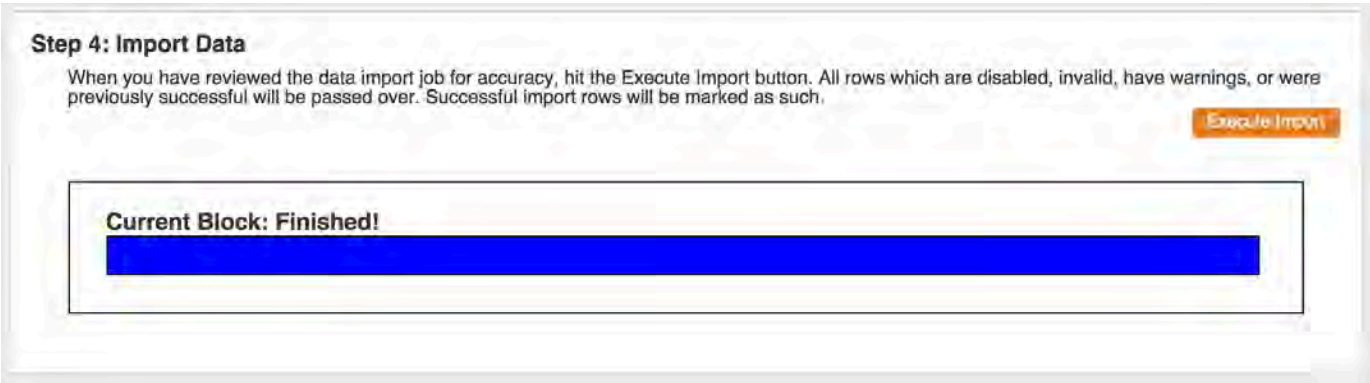
After editing, hit "Save", and continue reviewing / editing data as desired.



Step 4: Execute Import

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 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".

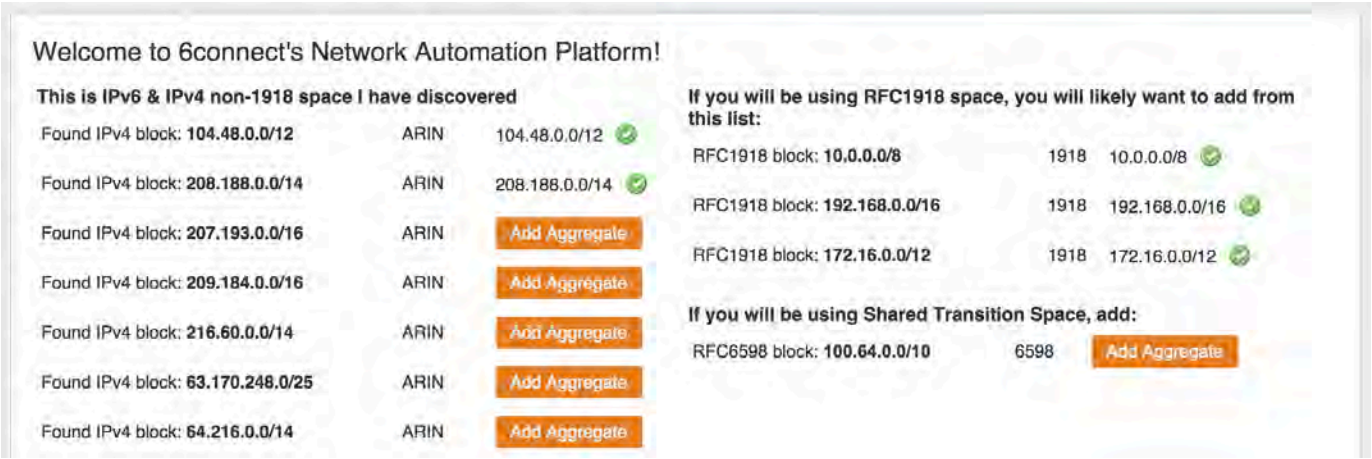
Step 1: Lookup from Source IP

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

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.



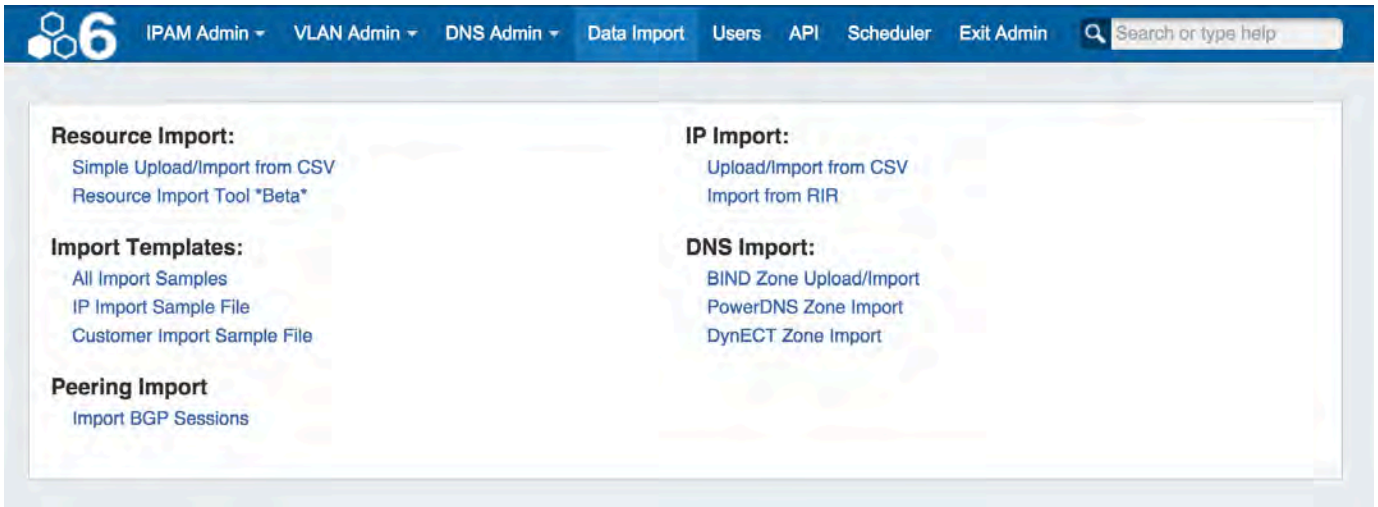
When done adding aggregates, hit the "Continue to ProVision" button at the bottom of the page. You will be redirected to the **IPAM** tab to manage the aggregates as desired.

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 the Admin section of ProVision:

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.

PowerDNS Zone Import

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

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.

Continue to the following sections for details on performing each import method:

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

BIND DNS Zone Upload / Import

BIND DNS Zone Import

The BIND DNS Zone Upload / Import tool uses 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.

The following steps are used when importing BIND DNS zones:

Preparing your DNS Zones for Import

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

There are three components for 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.

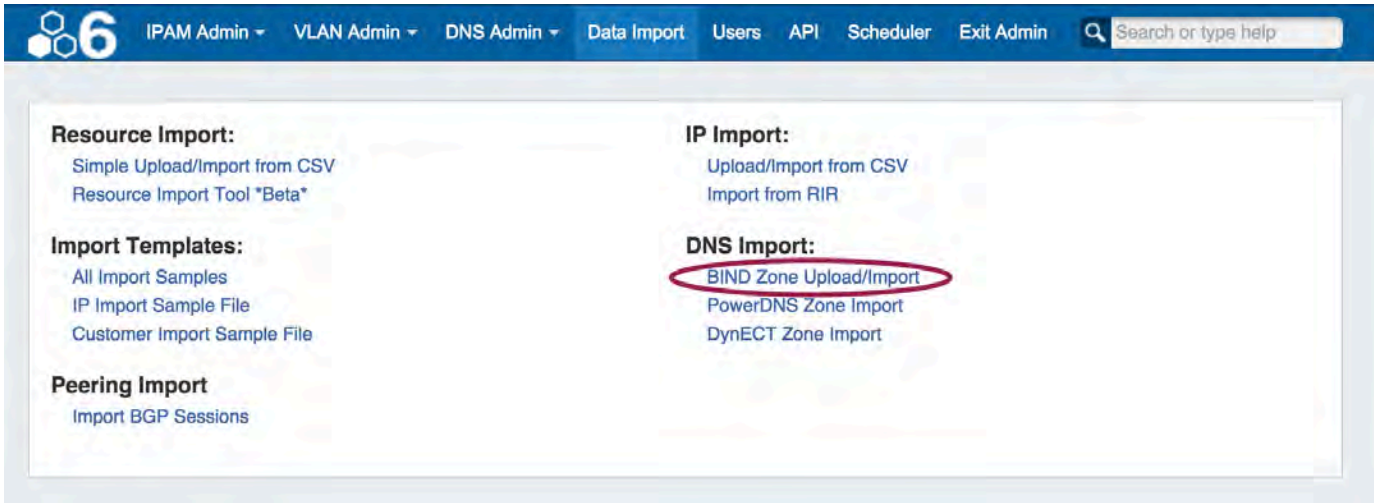
For a sample CSV: config.csv

Importing your DNS Zones (BIND)

Video Walkthrough

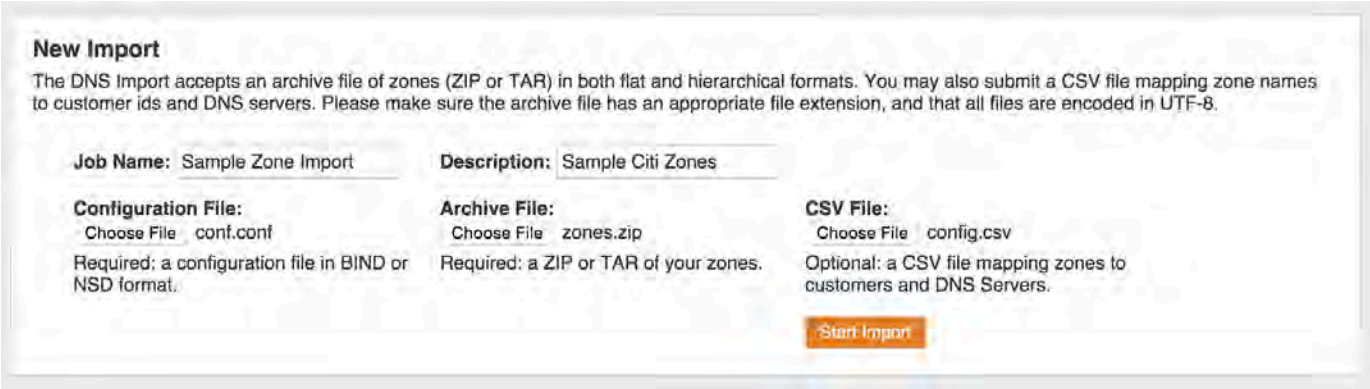
Step 1: Create a new DNS Import Job

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.

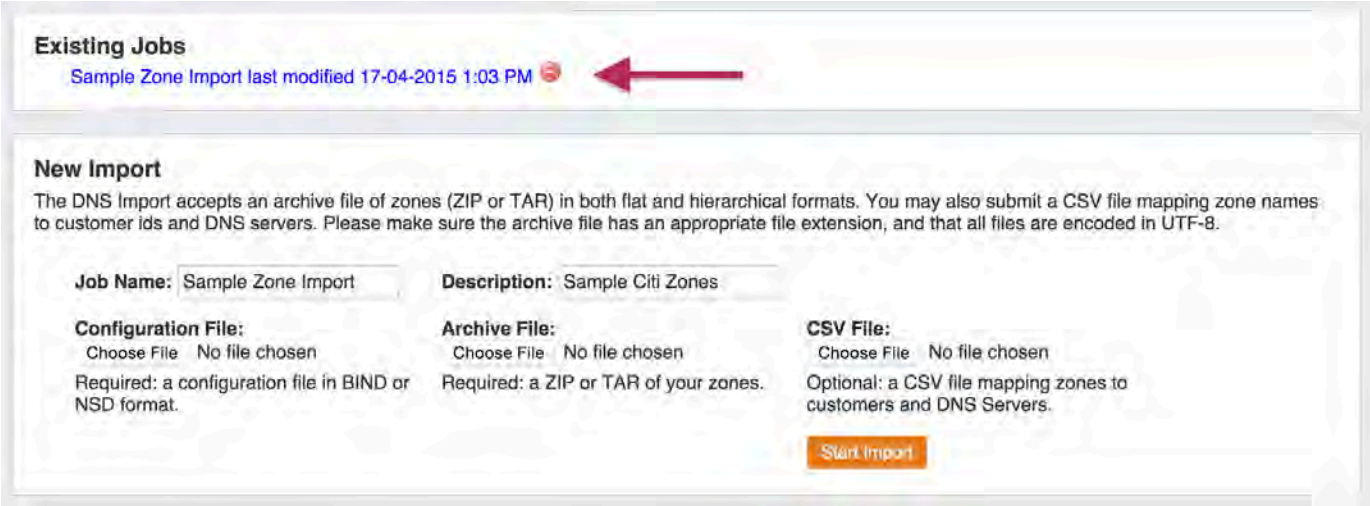
Select the appropriate **Configuration File** (required), **Archive File** (required), and **CSV File** (optional) that you prepared above by selecting the "Choose File" button(s) under each section, and browsing to the correct file location. Then hit "Start Import".



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.

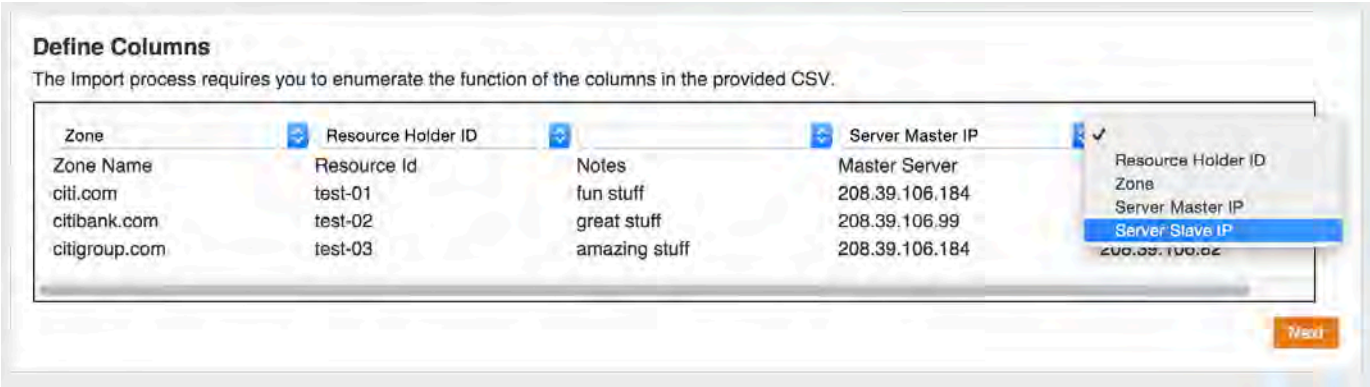


Step 2: Map Data Columns (Optional)

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.

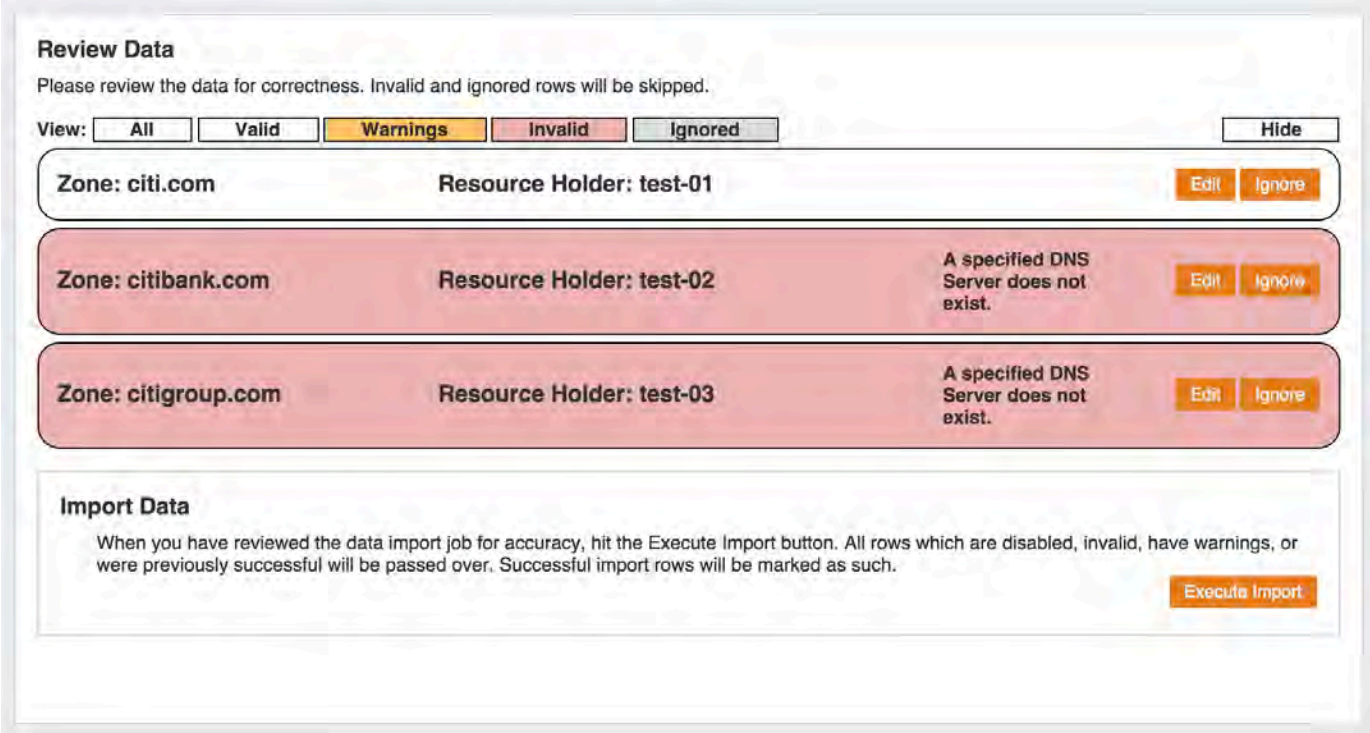
For DNS imports, four column definitions are available: **Zone**, **Resource ID**, **Server Master IP**, and **Server Slave IP**. Using the dropdown menu, select the appropriate definition for each of the imported columns. **Zone** and **Resource Holder ID** should each only have a single column selected, 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".



Step 3: Reviewing Data

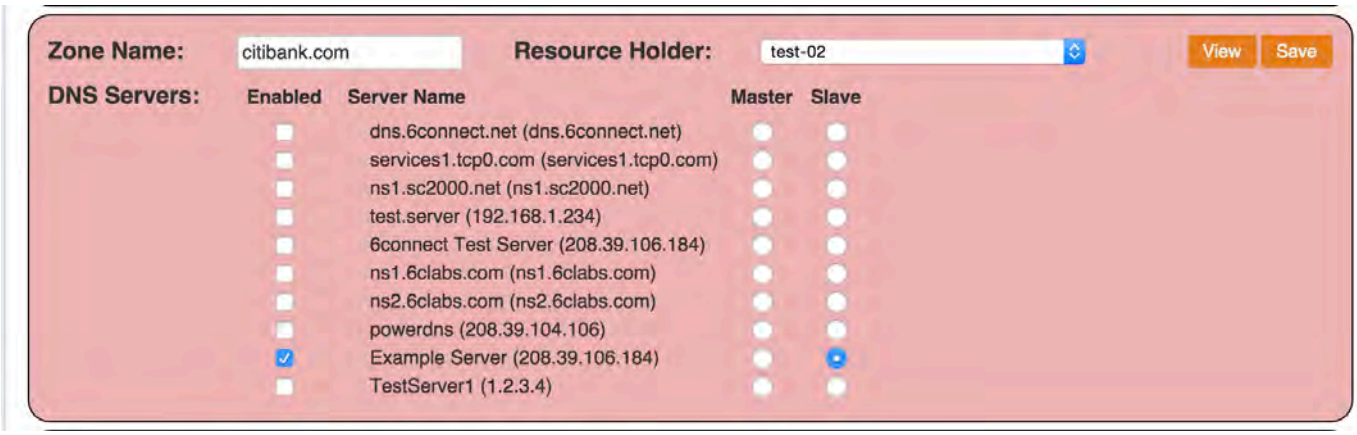
After supplying the file set and defining columns (if applicable), a review step is provided. The configuration file is broken into individual jobs, scanned for errors, and shown by row (in batches of 100) to be reviewed. Zones with errors will show as color coded. and can be filtered to be viewed by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



To fix errors, hit "Edit" on the zone record.

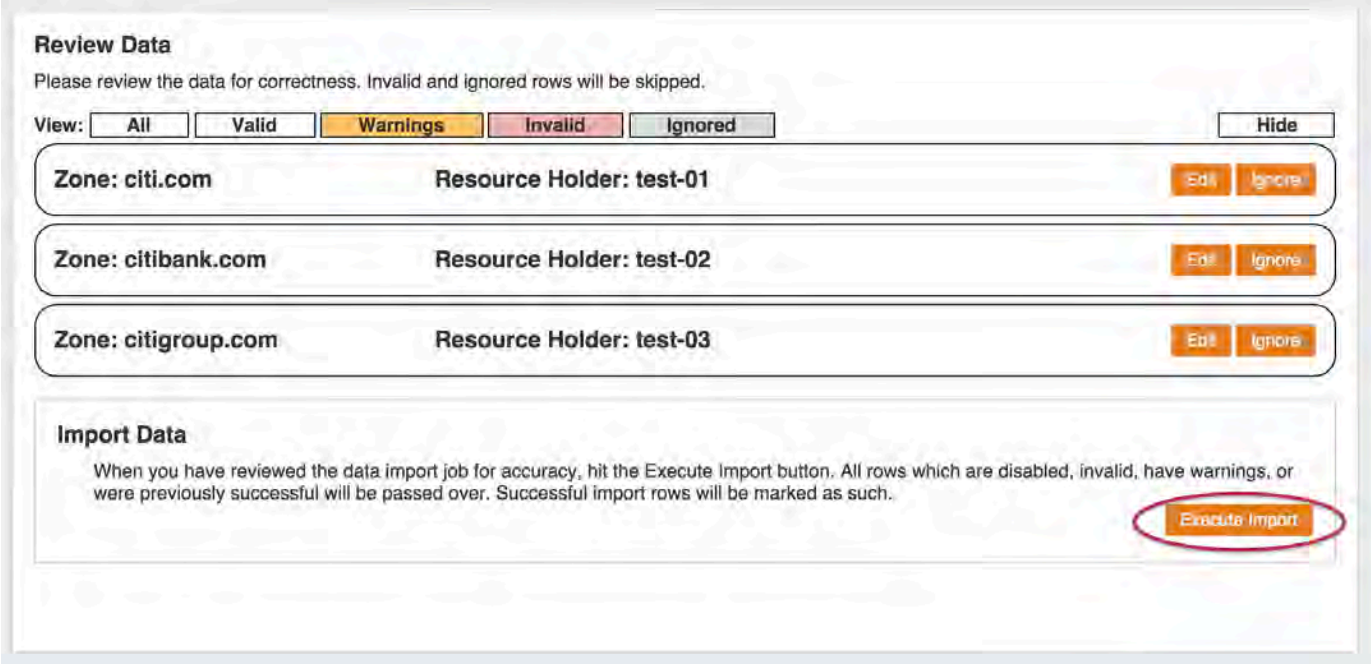
Editing the zone provides options to alter the Resource Holder, enable DNS servers, and redefine Master and Slaves.

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

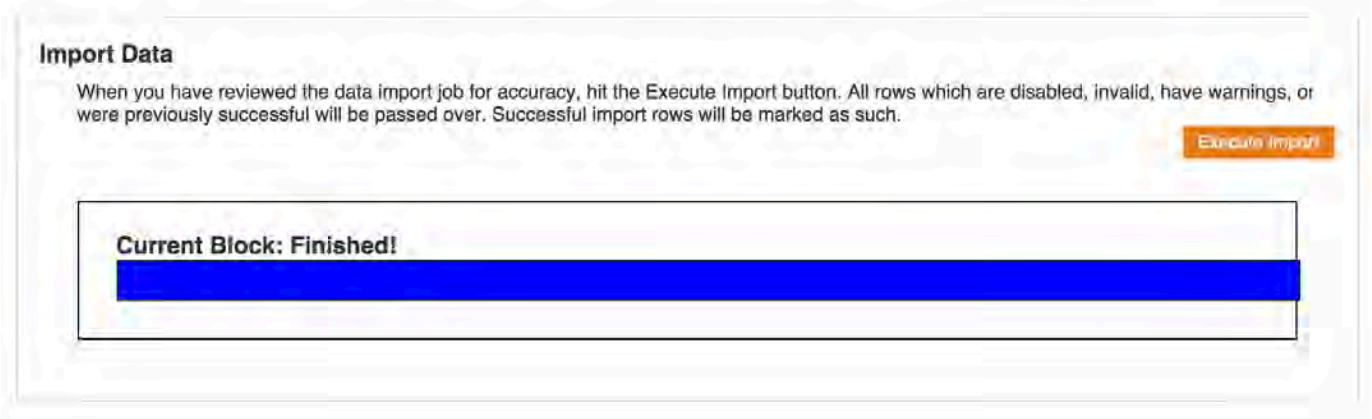


Step 4: Execute Import

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.



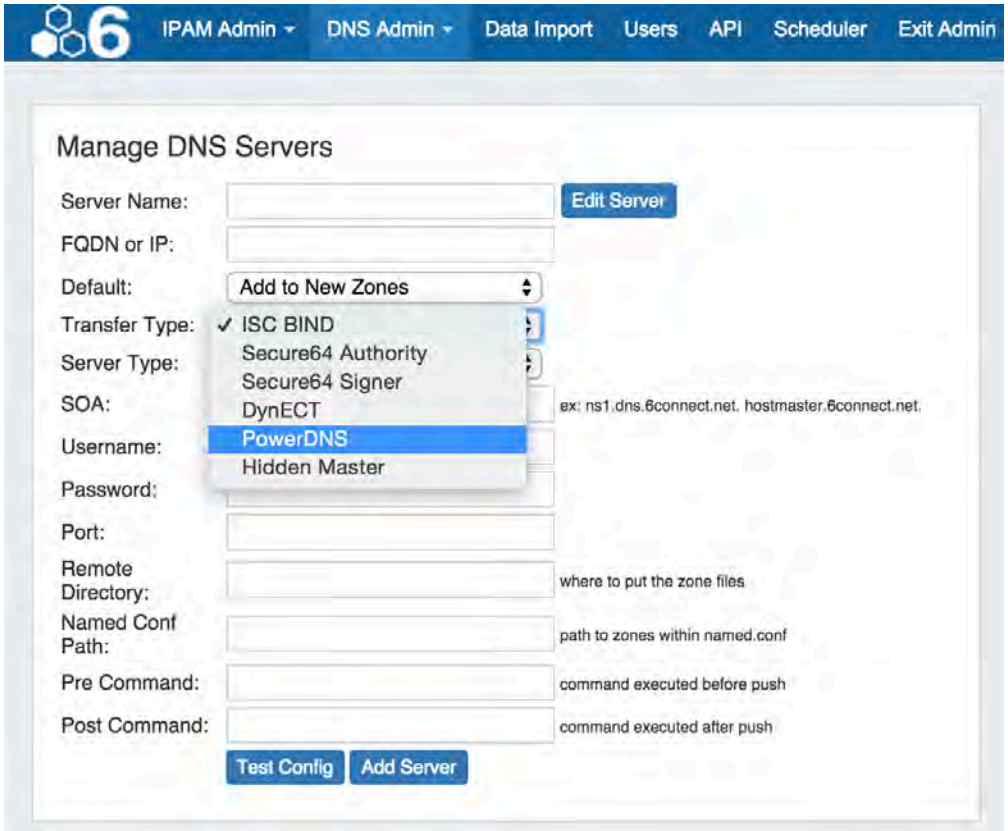
PowerDNS Zone Import

PowerDNS Zone Import

The PowerDNS Zone Import Option is available after configuring a PowerDNS server with a MySQL backend. It connects to the selected server and imports all zones.

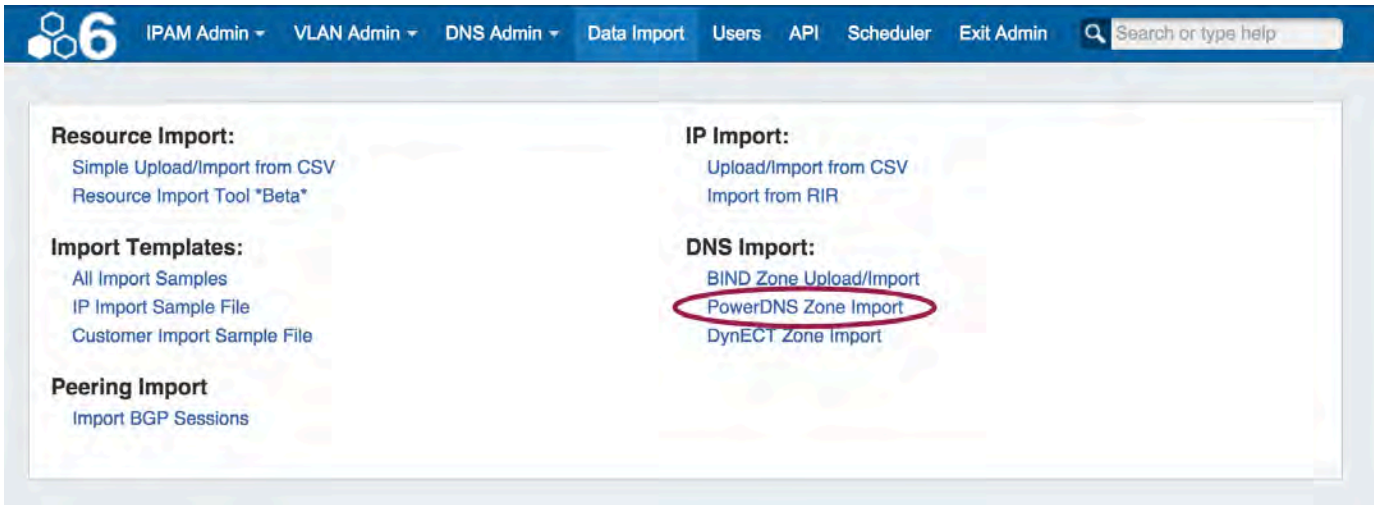
Step 1: Verify PowerDNS server setup

To import PowerDNS zones, first ensure the PowerDNS server has been set up under DNS Admin - Manage DNS Servers and configuring a PowerDNS server.



Step 2: Import your PowerDNS zones

Once server setup has been verified, navigate to the **Data Import** Tab in the Admin section. Select the "Power DNS Zone Import" link.



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

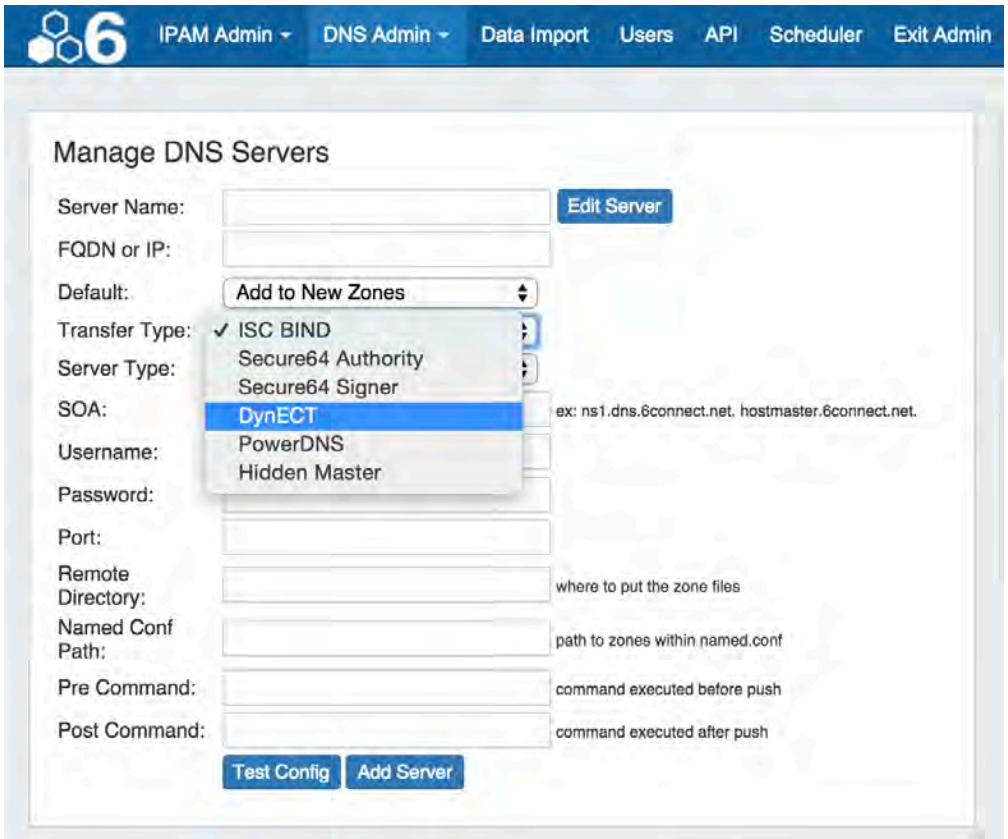
DynECT Zone Import

DynECT Zone Import

The DynECT Zone Import Option 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.

Step 1: Verify DynECT server setup

To import DynECT zones, first ensure the DynECT server has been set up under DNS Admin - Manage DNS Servers and configuring a DynECT server.



Step 2: Import your DynECT zones

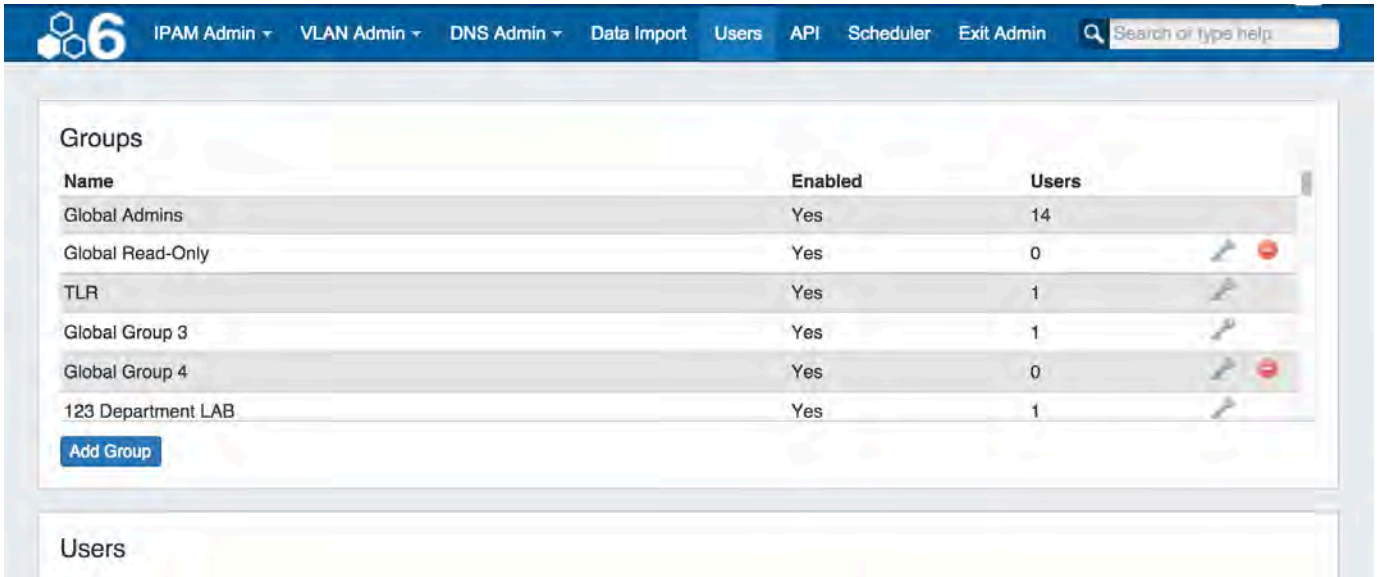
Once server setup has been verified, navigate to the **Data Import** Tab in the Admin section. Select the "DynECT Zone Import" link.

Then, to import your data simply choose your PowerDNS server and click on "Click here to begin the transfer".

Users & Permissions

Overview

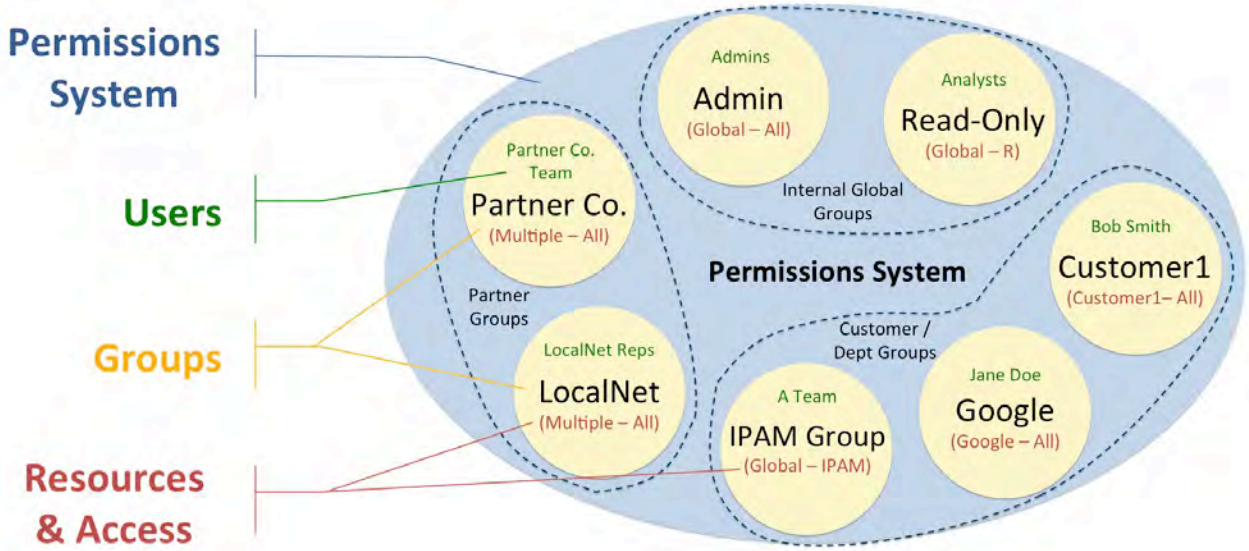
Users & Permissions is accessed from the Admin screen under the **Users** tab. Here, you will find tools for adding and managing permissions groups, users, and running queries for verifying a user's specific permissions.



The Permissions Structure

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



In this diagram, we have created groups for each of those scenarios – we have internal groups, Partner Groups, and Customer groups. Each of these groups has access to different resources, permission levels, and users assigned to them.

The components of the Permissions System include:

> **Users:** A User is a single login account that accesses ProVision. Users are assigned to Groups.
>
> **Groups:** A Group is a set of permission conditions that apply to selected Users. Allowed Resources and access levels (C/R/U/D permissions) are set inside the Group.
>
> **Resources & Access:** Inside a Group, Resource access may be set to Global (applies to all Resources), or to the Resource level (applies to only the selected Resources). For each Resource selected, access permissions can be set with C/R/U/D permissions under each ProVision functional area (IPAM, DNS, Resource, Peering).

As a whole, this makes up the ProVision permissions system. The Permissions system allows you to fine-tune access to resource data to be as detailed as you need.

Video Overview

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

Global Permissions

Global Permissions apply to the "TLR" or "Top Level Resource" within ProVision. By default, ProVision includes two groups with Global Permissions access - Global Admins and Global Read-Only. Initial Users are typically placed in the "Global Admins" group, and have administrative access to the entire platform. Global Read-Only users have full access to the platform, but with only read permissions.

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

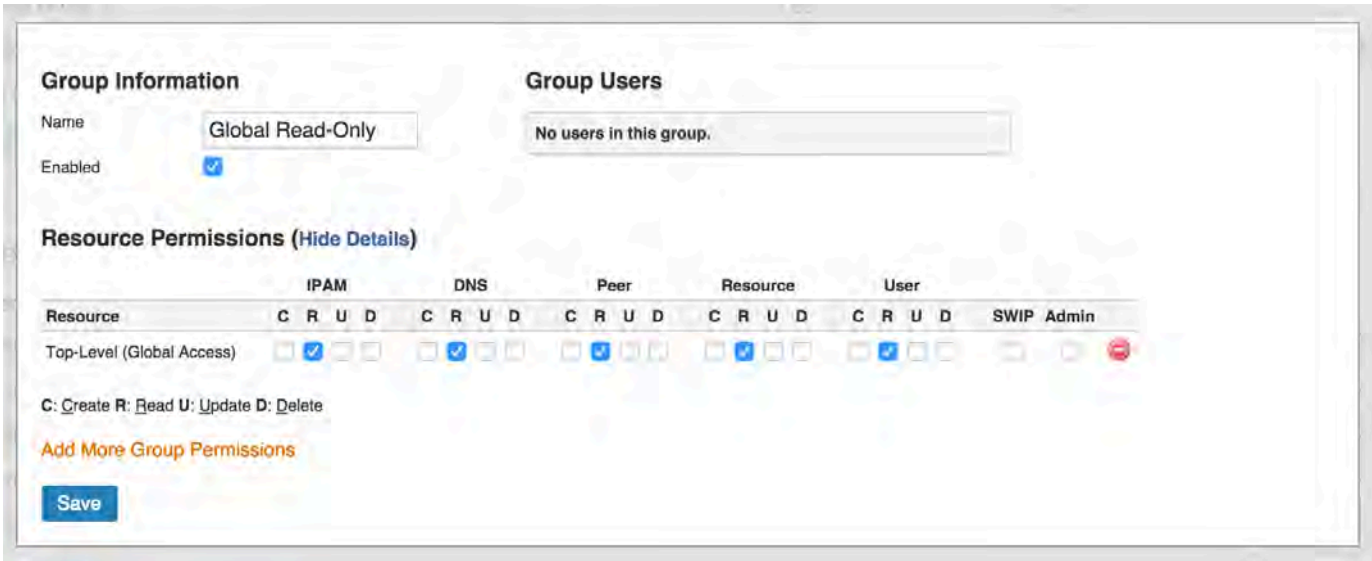
Global Permission Details

Global groups are visible under the "Groups" section of the **Users** tab. In addition to the two default Global groups, new Global groups may be created through the "Add Group" button.

Under the resource selector, chose the "Top-Level (Global Access)" Resource, and then check permissions as desired.

Global groups may also be edited just like standard groups, through selecting the Action Menu (Wrench Icon) to bring up the group information details. Groups may be deleted by selecting the red circle icon.

Group details are the same for Global groups as for non-global groups, excepting that the resource selected is Top-Level (Global Access). You may choose to edit the name, enable/disable the group, show or hide C/R/U/D permission details, and view users assigned to that group. Be sure to save any changes after editing.



Details on each global permission option is as follows:

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

Resource Permissions apply to designated Resources within ProVision. Rather than allowing a user full access to the ProVision platform, you can choose to limit a user's access to only certain resources or functional areas.

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

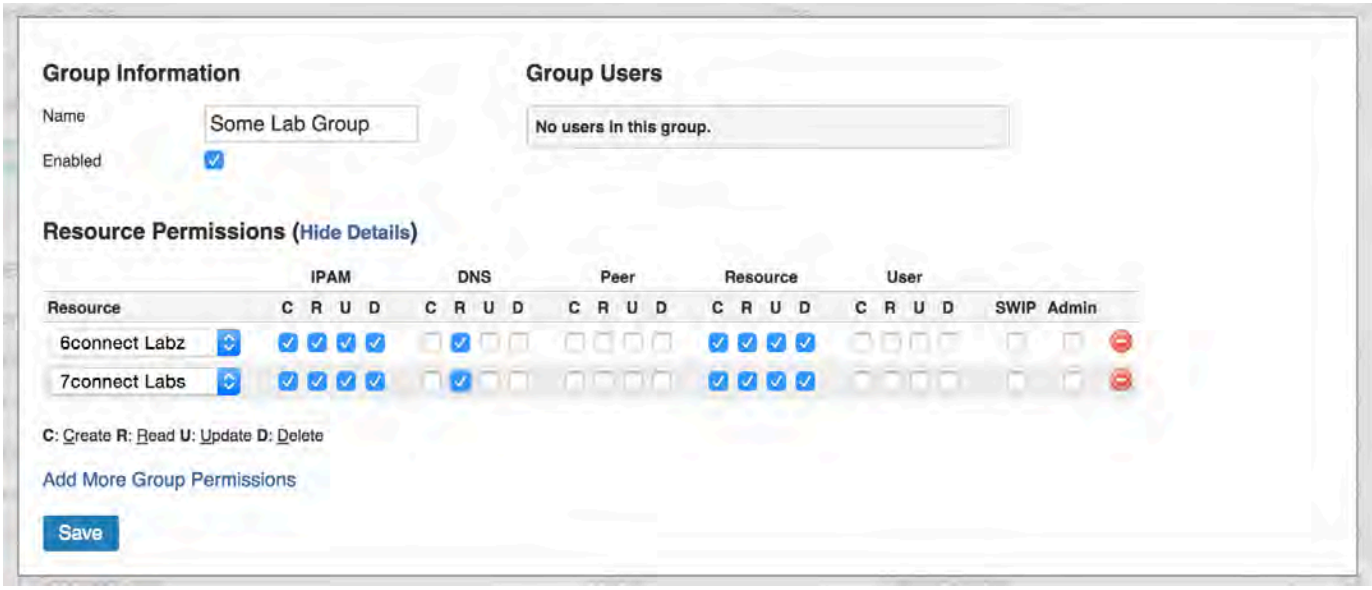
Resource Permission Details

Resource Permission groups are visible under the "Groups" section of the **Users** tab. New groups may be created through the "Add Group" button.

Under the resource selector, chose one or more resources for which you want to define permissions, and then check permissions as desired. Additional Resources may be added to the list by clicking on "Add more Group Permissions".

Here, we show making a group called "Some Lab Group", whose users we want to be able to access two resources: 6connect Labz and 7connect Labs. These users will be working extensively in IPAM and Resources, so we give them full access to those areas of ProVision.

However, we also want them to view DNS information, but not edit it. We click on "Show Details" to fine-tune the permissions, and then check the "R" column under DNS. Lastly, we hit "Save" to save our changes.



Details on each resource permission option is as follows:

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

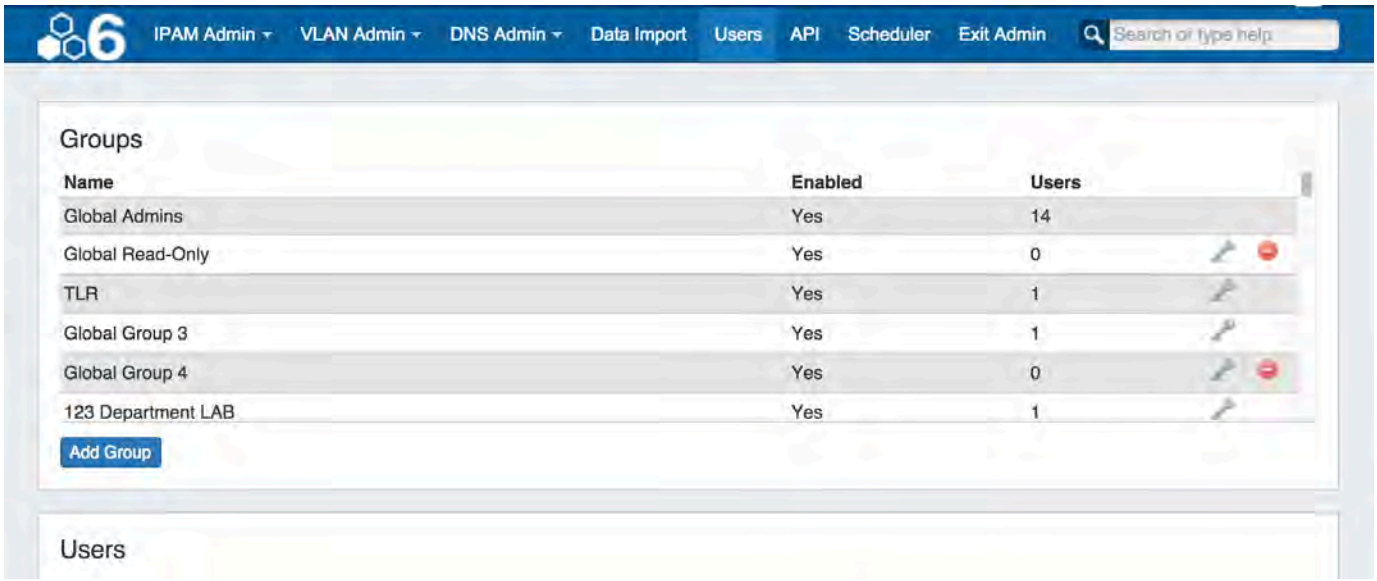
Functional Area	Description
IPAM	IP Address Management functionality - this covers the IPAM Tab in addition to the IPAM "Gadget" that can be present in Resources.
DNS	DNS Zone/Zone Record Management functionality - this covers the DNS Tab in addition to the DNS "Gadget" that can be present in Resources.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

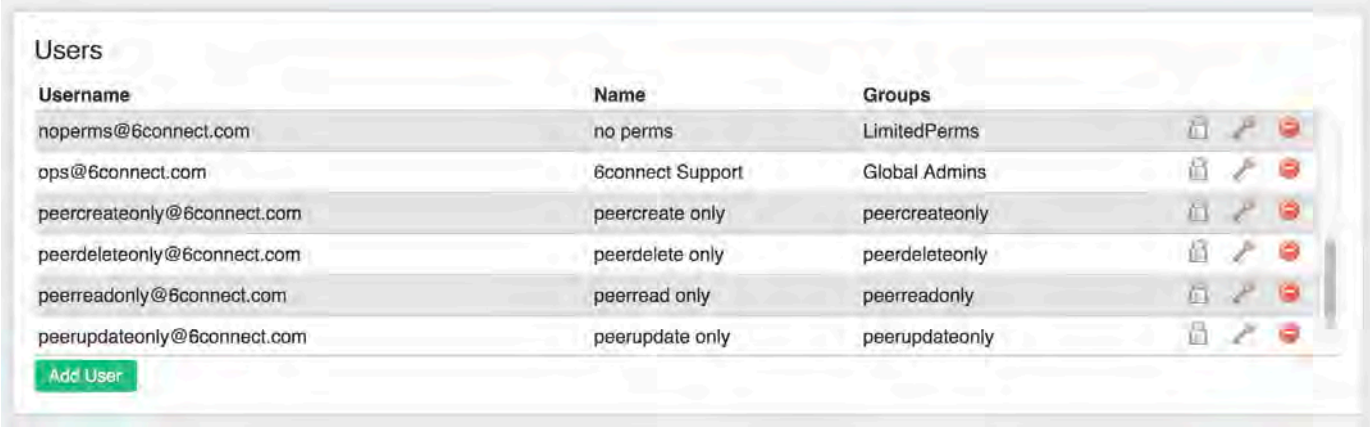
Managing Users and Groups

Users and Groups are managed from the Admin area of ProVision, under the **Users** tab. Both Global and Resource level groups are shown.



Users Overview

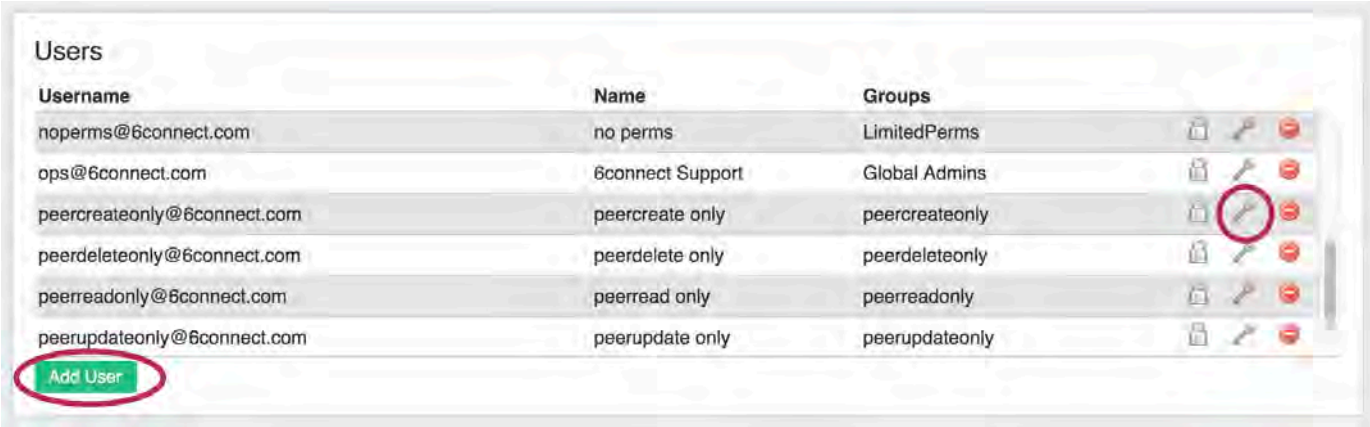
A User is defined as a single login account that accesses ProVision. In the Users section, users may be added, edited, assigned to groups, have password information reset, or be removed.



Creating/Editing User Accounts

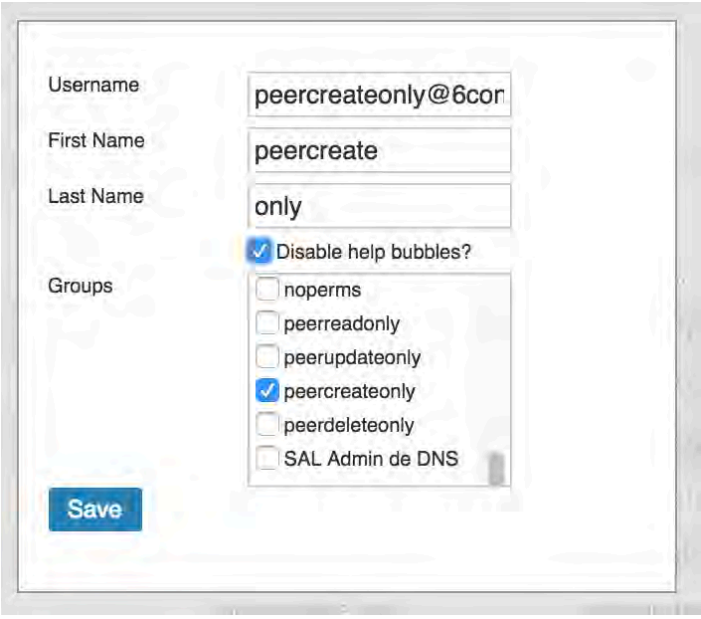
Add Users by click on the "Add User" button at the bottom of the "Users" section of the page. The Edit User dialog will pop up.

You can bring up the same dialog to edit a user's information by clicking on the Action Menu (wrench icon) at the end of the row for the user.



When creating or editing User accounts, you will be presented with the following options.

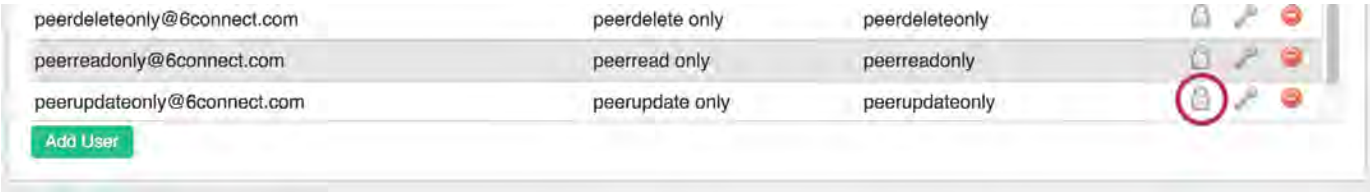
You may enter or edit the Username, First Name, Last Name, whether to enable help bubbles, and select one or more permissions groups to which to assign the user.



> When assigning users to multiple permissions groups, be sure to review the group permission details to ensure that they do not conflict with eachother!

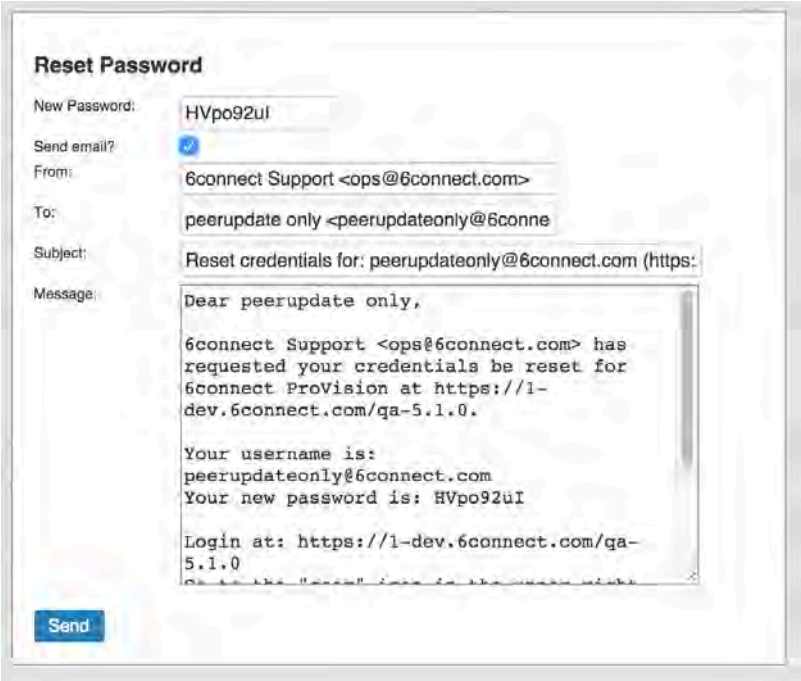
Setting/Resetting User Passwords

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.



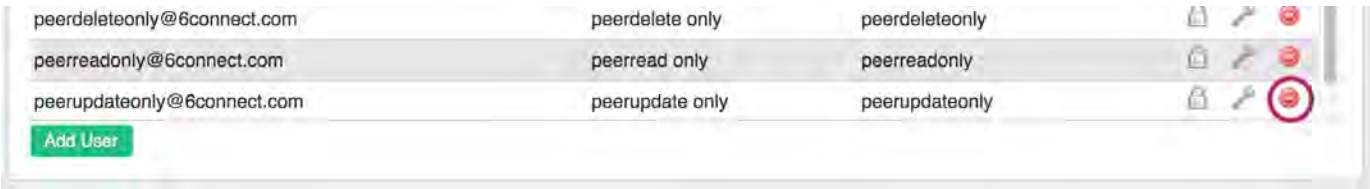
From here, you can choose a new password, or use the automatically generated random password.

If you select the checkbox next to "Send email?", email fields will appear that may be edited as desired. When complete, hit "Send".



Removing a User

To remove a user from the Users list, click on the red circle "delete" icon next to that user.



Groups

ProVision administrators can also create permission groups to assign users to. This allows more control over user roles.

A Group has four elements associated with it:

Users: The Users that are assigned to the group, and will be limited by the selected permissions.

Resources: Resources access may be set to Global (applies to all resources), or to the level of individually selected resources.

Functional Area: The ProVision functional area (IPAM, DNS, Resources, Peering, etc) for the selected resource(s) that permissions are set under.

Permission Level: Create, Read, Update, and Delete (C/R/U/D) permissions may be set under each functional area for each resource selected.

In the example above, the group “MyGroup” allows the users Bob Smith and Jane Doe to only read IPAM and DNS data for the Resource “Awesome Co”.

Through the use of these four elements, Groups can create permissions structures that are as flexible as you need.

Default Groups

Two default groups are available initially in ProVision:

- Global Admin
- Global Read-Only

These Global groups allow for access to the entire ProVision platform. You may add additional Global groups by creating a new group using the Top Level (Global Access) Resource. You may also add detailed resource level group permissions by selected individual resources.

For more detail on top-level and resource permissions, see Global Permissions and Resource Permissions.

> **Overlapping group and user permissions**
> Permissions are inherited based on the hierarchy of the objects, unless you specify a different permission!

Add or Edit a Group

New Groups can be created by ProVision administrators by pressing the green "Add Group" button. To edit a group, you may click on the Action Menu (wrench icon) for the group.

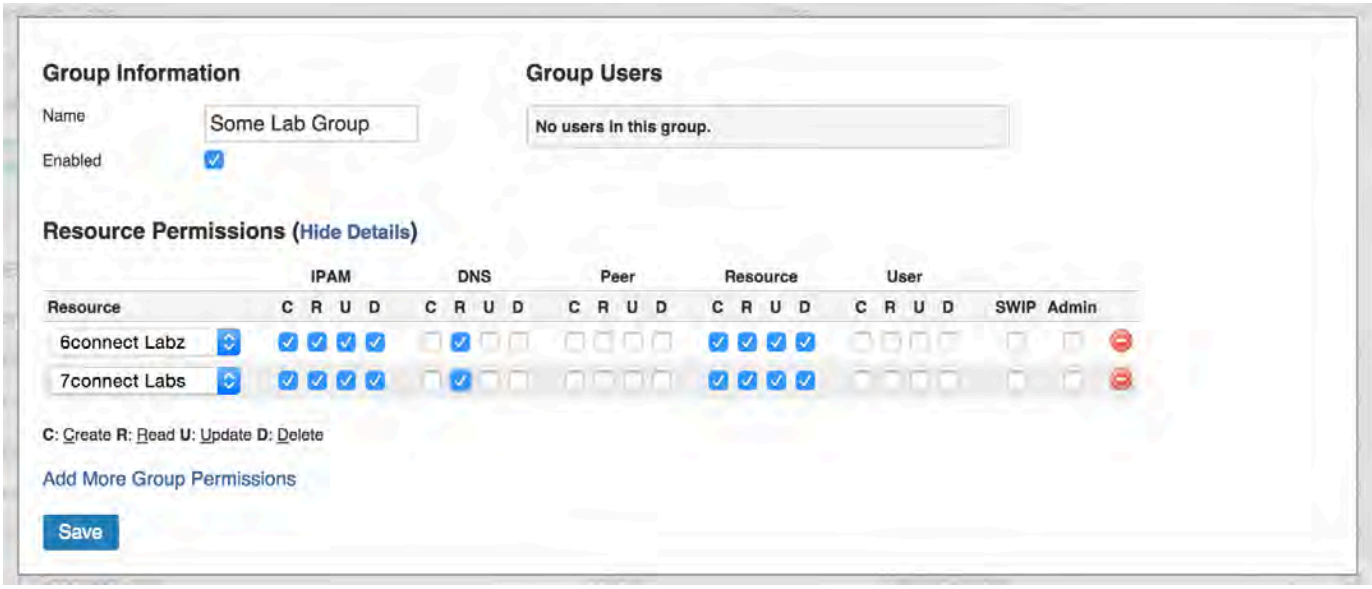
After hitting the "Add Group" button or wrench icon, the Group Information screen will pop up.

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, we make a group called "Some Lab Group", whose users we want to be able to access two resources: 6connect Labz and 7connect Labs. These users will be working extensively in IPAM and Resources, so we give them full access to those areas of ProVision. However, we also want them to view DNS information, but not edit it. We click on "Show Details" to fine-tune the permissions, and then check the "R" column under DNS.



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

Removing a Group

To remove a user from the Users list, click on the red circle "delete" icon next to that user.


Groups
Global Admins
Yes
11
Global Read-Only
Yes
0
TLR
Yes
2
Global Group 3
Yes
2
Global Group 4
Yes
0
123 Department LAB
Yes
1
Add Group

Verifying Permissions

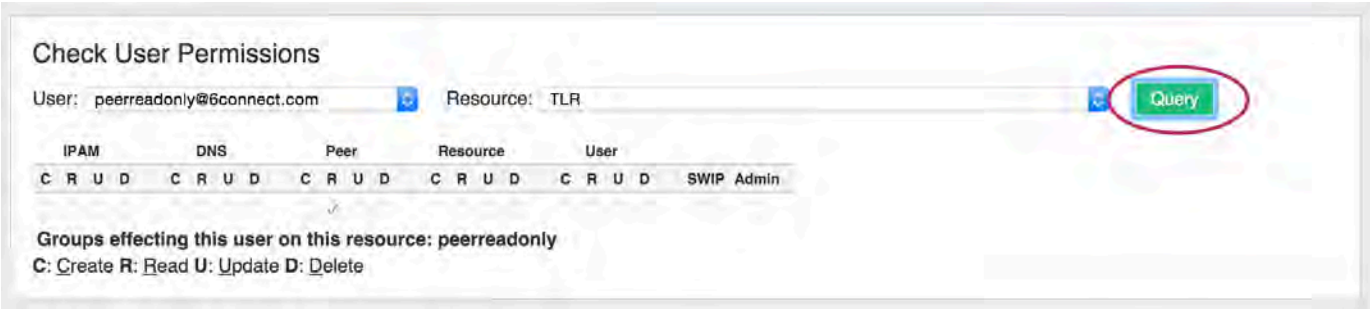
Verifying Permissions

Check User Permissions

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.

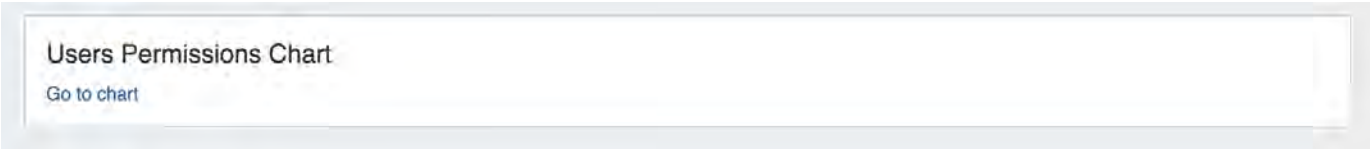
To verify the permissions of a certain user, simply select their user account from the dropdown menu, the resource you are checking against, and click on the green "Query" button.

The resulting output will display the detailed permissions for that user and resource combination, as well as list groups effecting the user.

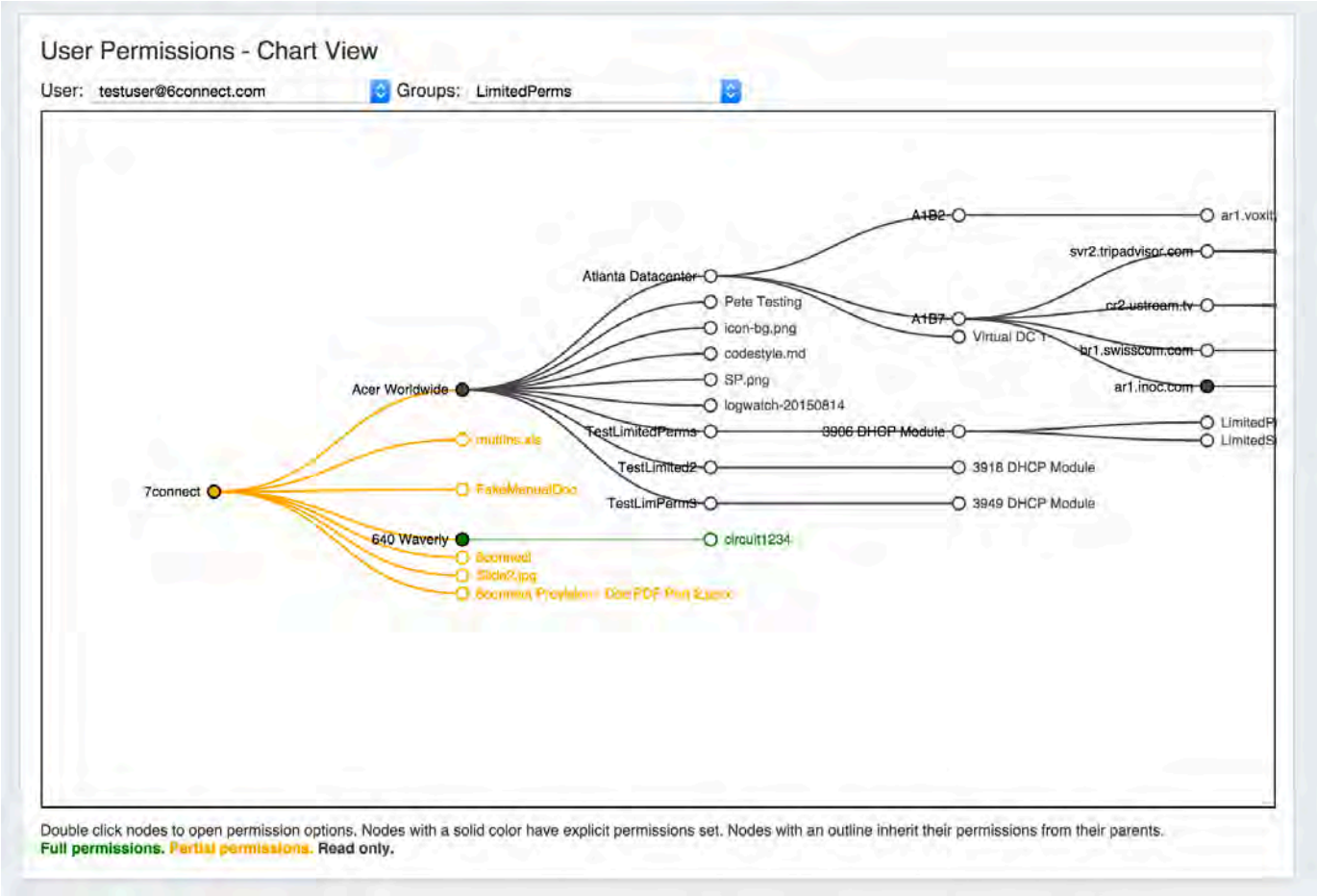


User Permissions Chart

The User Permissions Chart is accessed from the bottom of the **Users** tab page. Click "Go to Chart" and the chart page will open.

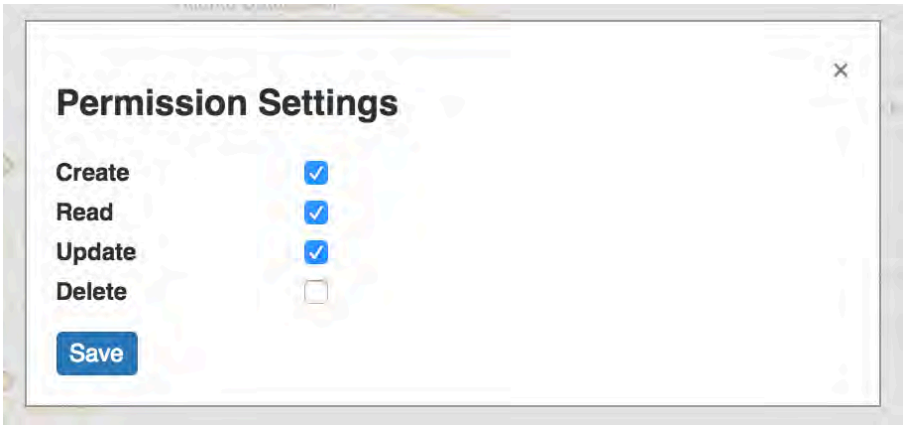


You can graphically view the permissions level of a user by selecting their user name and group from the dropdowns at the top of the chart. The resulting chart will be color coded depending on permissions level.



Single clicking on a filled circle resource node will expand and collapse the node.

Double clicking on a resource will open up the Edit Permissions box, where CRUD permissions may be updated for allowed Resources. Check / Uncheck the desired permissions level, and click save.

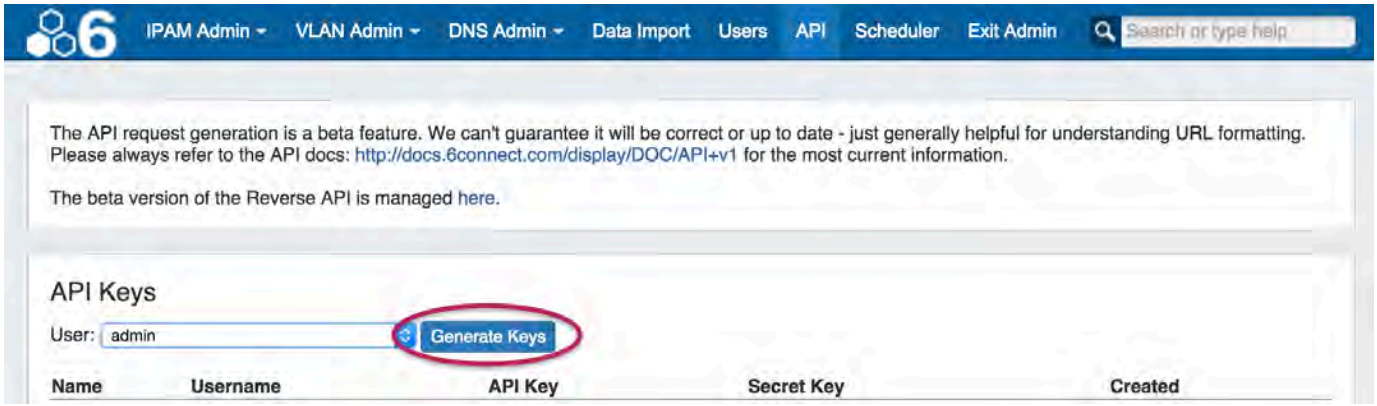


API Tab

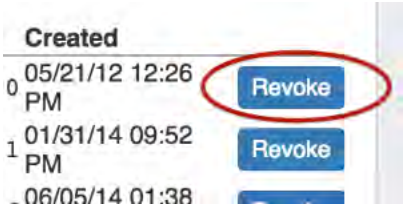
API Tab

The **API** tab allows you to create and manager API keys for users. It also provides links to API documentation and beta Reverse API management.

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.

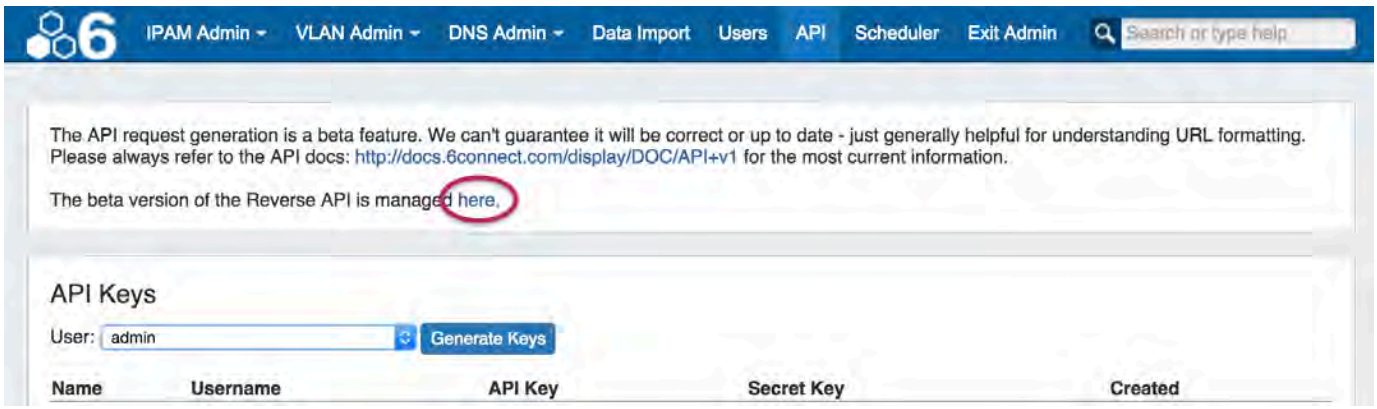
Table of Contents:

Reverse API 1

Reverse API User Interface

This is a beta feature

The Reverse API endpoint builder is accessed from the Admin section of ProVision, under the **API** tab.



Clicking on the link circled above will take you to the Reverse API Endpoint Builder interface.

The Revese API system is a beta feature.

Reverse API Endpoint Builder

Endpoint: Add a New Endpoint

Name:

Call:

Presentation Javascript:

```
/*
    The Reverse API system will perform the system call described above and supply its raw data
    to the function below, along with a jQuery reference to the output div.

    The presentation function should parse the return data according to its particular format,
    interpret and arrange that data, make any necessary secondary calls, and construct the final
    output in the given Div.

    The language used is Javascript.  A full jQuery install is available.
*/

endpoint = function(data, outputDiv) {

    // presentation code goes here

};
```

Test Call on: Select a Resource Test

Delete Endpoint Save Endpoint

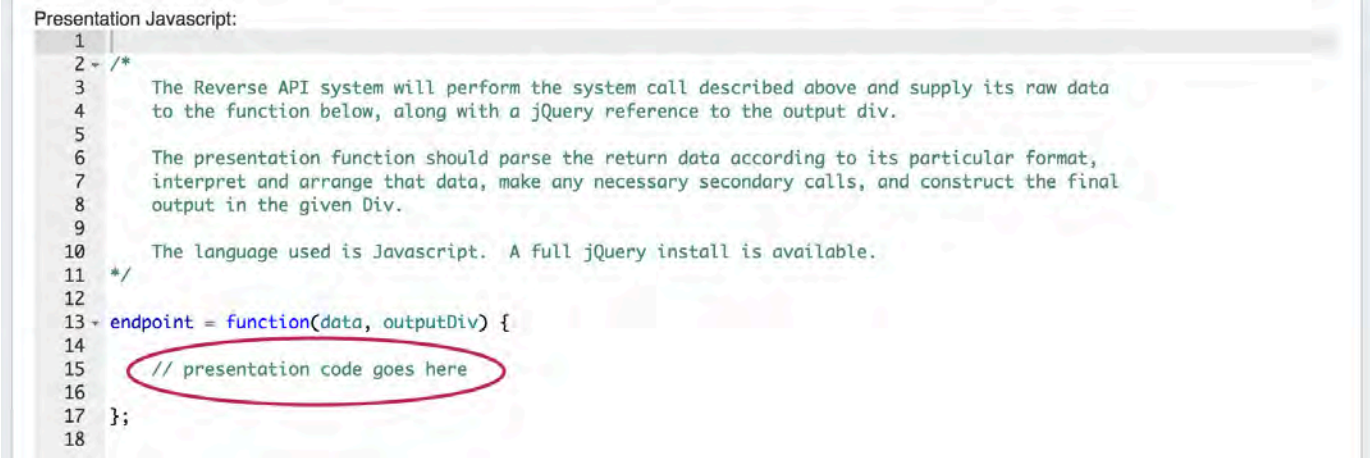
Add a New Endpoint

To add a new endpoint

1) Select "Add a New Endpoint" next to "Endpoint".

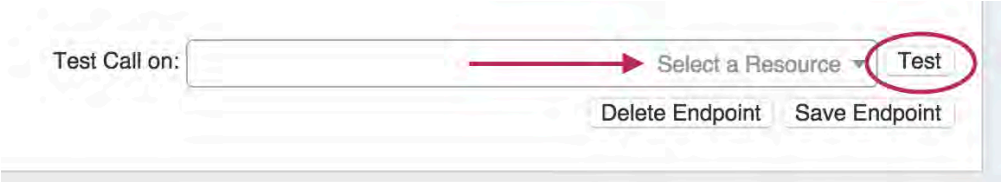
2) Type in the desired endpoint name and call

3) Review / Edit the presentation Javascript as desired for your output



4) Save your new endpoint

5) Test the call on a selected Resource by selecting a Resource from the dropdown, then hitting the "Test" button.

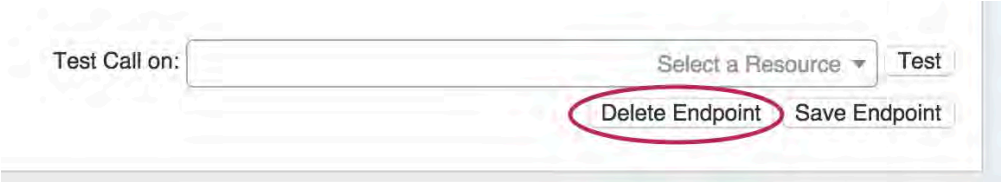


> Note: The selected Resource must have a Reverse API gadget enabled section! See Customizing Sections and Gadgets for information on enabling gadgets for a section.

Delete an Endpoint

1) Select the endpoint name next to "Endpoint".

2) At the bottom of the screen, click "Delete Endpoint"



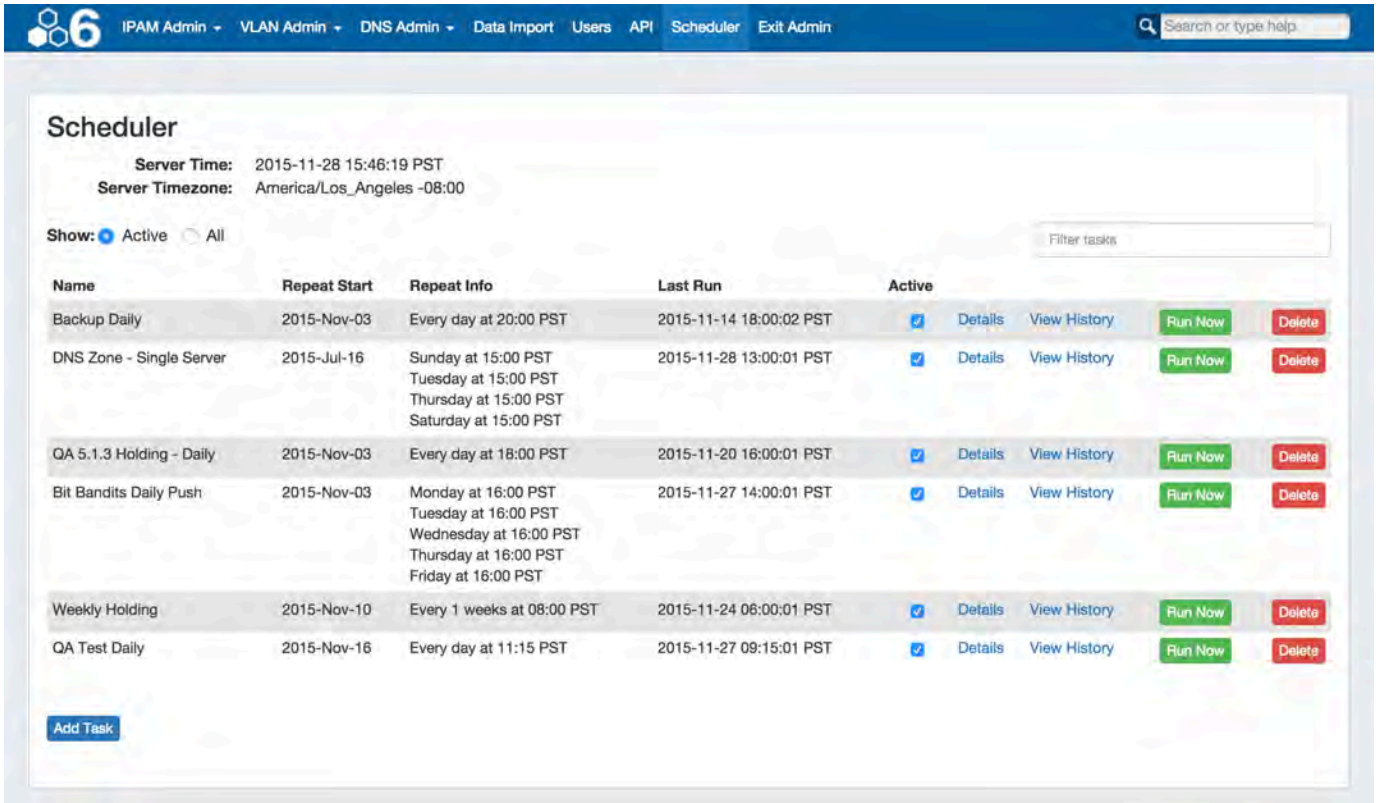
Edit an Endpoint

1) Select the endpoint name next to "Endpoint".

4) Save the endpoint

Scheduler

Scheduler



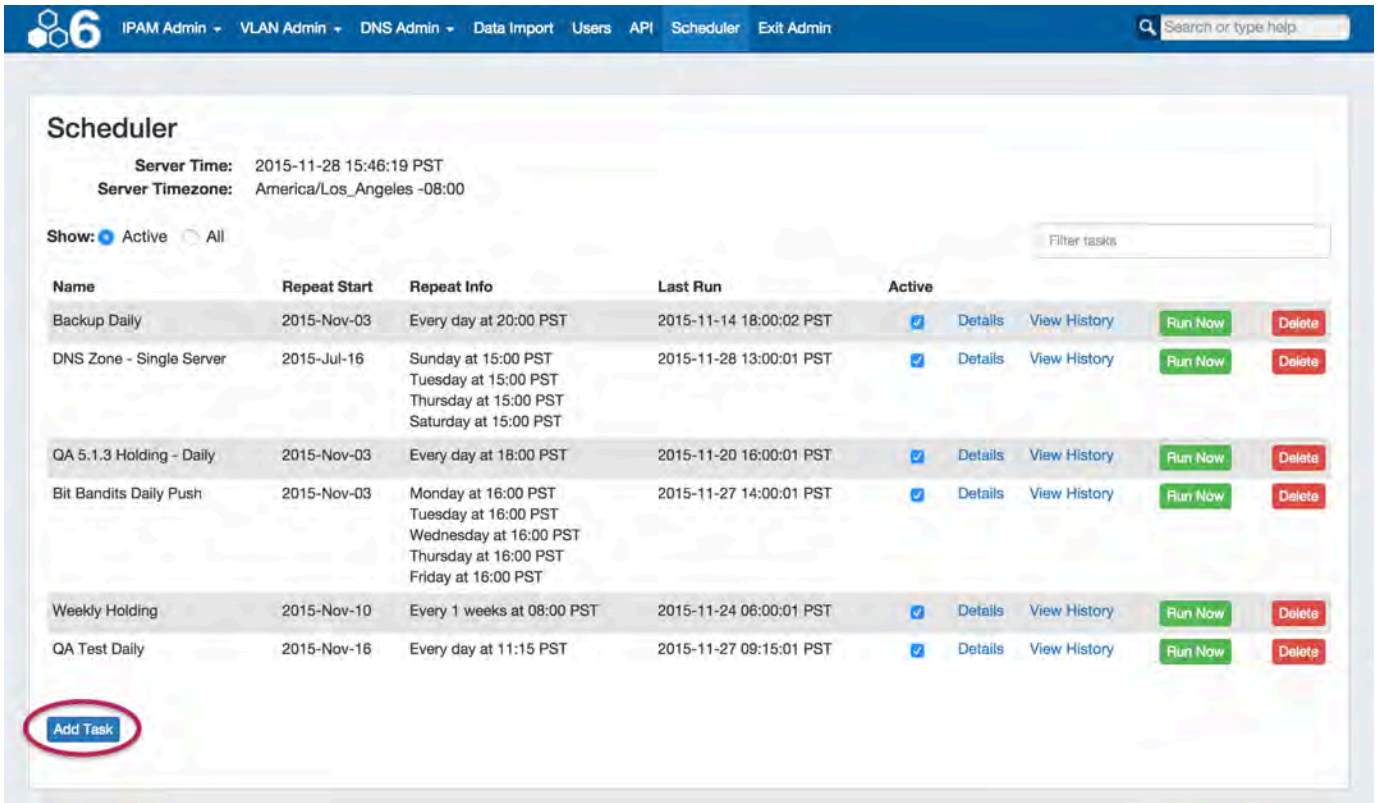
The Scheduler tab allows you to manage and schedule repeating tasks in ProVision.

Available Task Types

Four predefined task types are available through the scheduler:
Process Holding Tank - Removes IP Blocks from the Holding Tank and returns them to the available pool.
DNS Zone Transfer - Pushes zone updates to the DNS servers. Select pushes to apply to all zones on all servers, all zones on one server, or one particular zone.
Backup - Performs a data backup to the 6connect cloud servers, or to a selected resource server existing in ProVision.
IPAM Alerts - Set a time / frequency for IPAM aggregate alert emails to be sent to alert recipients. Alert recipient email and available space threshold percentage is set through the action menu for individual aggregates under the IPAM Tab.

Add Task

To add a new scheduled task, hit the "Add Task" button below the Scheduler task list. You will then see the Task Detail Settings area.



Enter Task Details

Under "Task Detail", fill in the following fields:

Name: Create a name for your task

Task: Select a pre-created task from the dropdown menu.

Process Holding Tank Settings: When selecting the Process Holding Tank task, you will have the following setting options:

> **Email From / Email To**: Enter an email address to send task notifications from, and the desired recipient. This is an optional setting.

DNS Zone Transfer Settings: When selecting the DNS Zone Transfer task, you will have the following setting options:

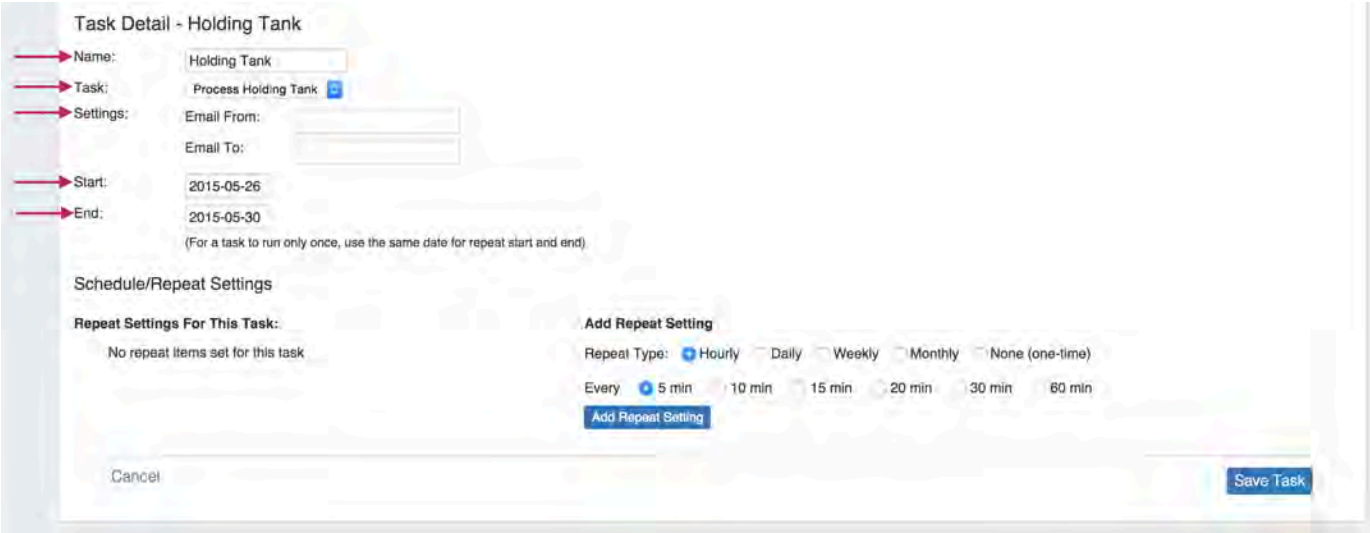
> **Select Action Type / Select Server:** When selecting the DNS zone transfer task, you can choose the action type to a) push all zones to all servers b) push all zones on a single server (then select server) or c) push one zone (then select zone).

Backup the Database Settings: When selecting the Backup task, you will have the following setting options:

> **Email From / Email To**: Enter an email address to send task notifications from, and the desired recipient. This is an optional setting.
>
> **Location:** Choose "Cloud" to save the backup to the 6connect cloud, or "Server" to save to a specified server resource already defined in ProVision. If "Server" is selected, choose the server resource from the dropdown list.

Start: The start date you wish the task to begin repeatedly occurring. By default, the current day is entered.

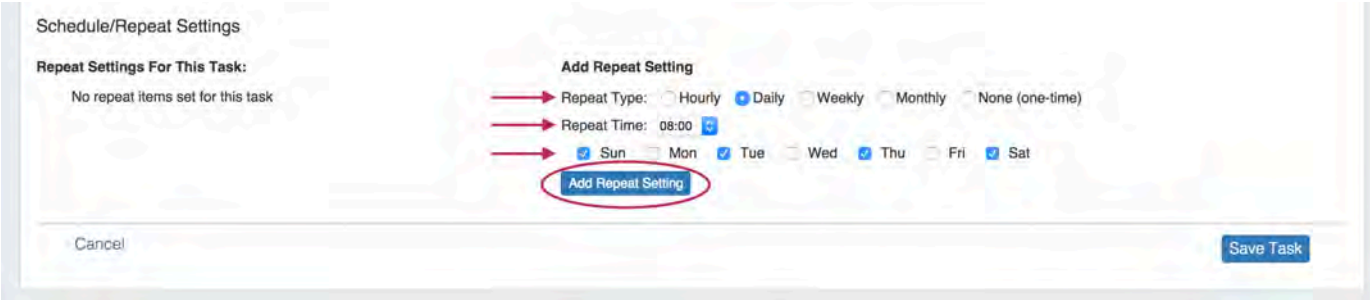
End: The date you wish repeat settings to end.



> **Tip**
> To create a task that only runs one day, use the same date for start and end.

Add Repeat Settings

Add repeat settings for your tasks. First, select your repeat type - hourly, daily, weekly, monthly, or none (one time). The type that you select will prompt different additional setting options.



Repeat Type options and details:

Hourly: Selecting the hourly repeat type will then prompt for the number of repeat times per hour - every 5,10, 15, 20, 30, or 60 minutes. All minute selections start from the top of the hour. For example, if every 20 minutes is selected, the task will run at :00, :20, :40 during each hour.

Daily: Selecting the daily repeat type will prompt for the specific time to repeat the task, and which days you wish the task to occur. The Repeat Time is based on a 24 hour clock set at UTC time. To assist in any time zone conversions, the current UTC time is shown at the top of the scheduler page.

Weekly: Selecting the weekly repeat type prompts for the specific time to repeat the task as well as frequency by week. Weekly tasks occur on the day selected as the start date. Using the weekly option, you may choose from a range of weekly (every 1 week) to once every two months (every 8 weeks).

Monthly: Selecting the monthly repeat type prompts for a repeat time and a day. For the day option, you can either choose a specific day (ie, the 15th of each month) or a relative day (the 1st Friday of each month).

> For Monthly repeat settings, verify that the calendar day you select the task to occur exists! For example, if you select the 5th Saturday of each month, the task will repeat only in months which have 5 Saturdays and skip all other months.

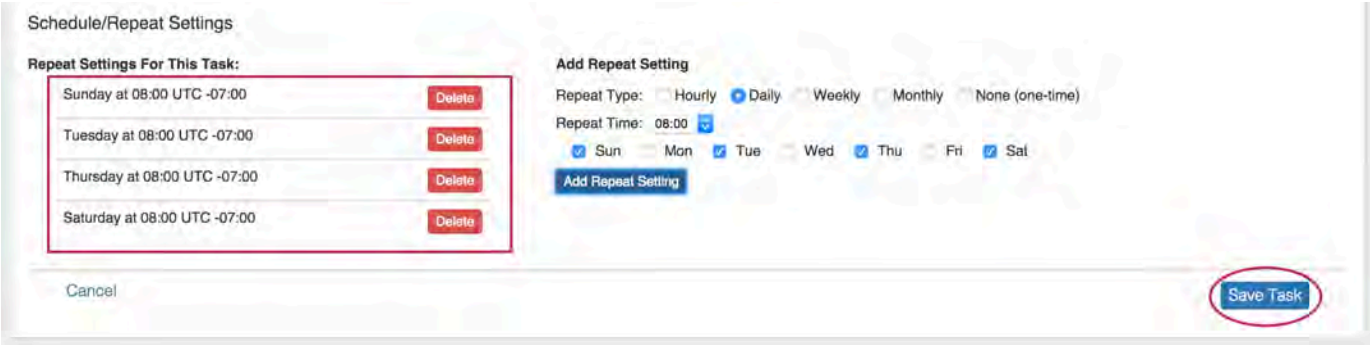
None (one-time): Select the time and day that you wish the single occurrence task to run.

After you have selected your Repeat Setting options, click the "Add Repeat Setting" button.

Verify Repeat Settings and Save

When you add repeat settings, they will appear under the "Repeat Settings For This Task" area. Verify your settings, and if desired, you may delete individual repeat settings by hitting the "Delete" button.

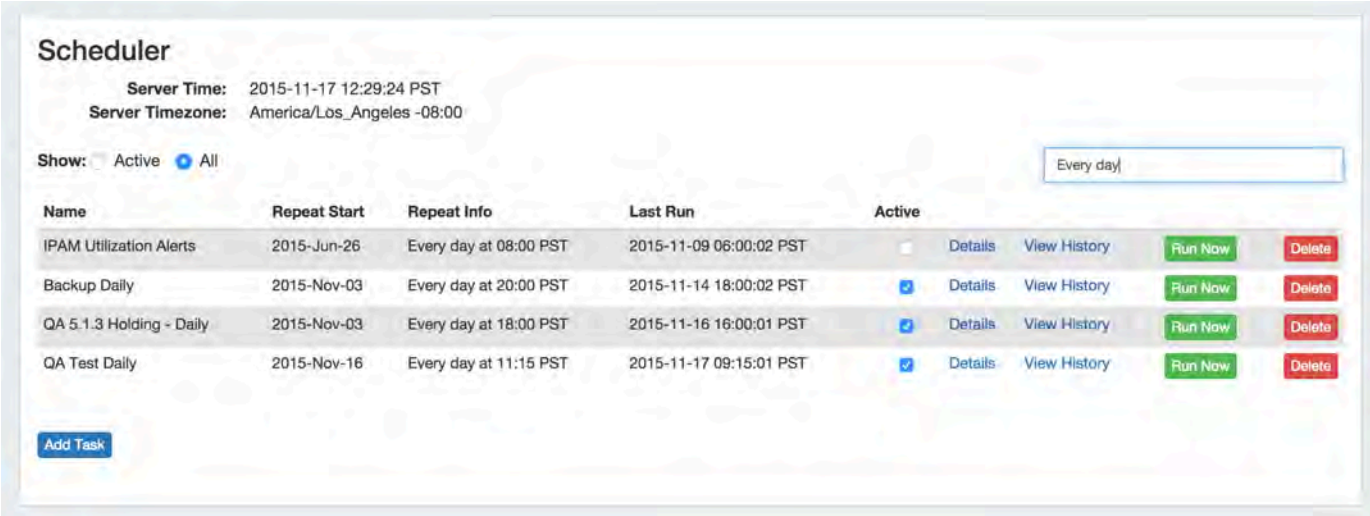
When complete, hit the "Save Task" button to finalize your changes and add your new task to the task list.



Managing Tasks

After tasks have been added, they will show in the task list along with their basic settings. The list may be filtered to show all tasks or active tasks only by selecting the radio buttons next to "Show:". Further filtering may be done by typing a Name or Repeat Info keyword into the "Filter Tasks" input box above the task list.

From the task list itself, you can enable/disable tasks, view details, view history, run tasks, and delete tasks.



Filters:

Show: Select "Active" to show only active tasks (checkbox enabled under "Active" for the task), or "All" to show all tasks

Filter Tasks (Text box): To further filter the task list, type a Name or Repeat Info keyword (ie, "Backup" or "Every day") into the Filter Tasks text box.

Task List:

Name: The task name assigned during the "Add Task" creation process.

Repeat Start: The date selected for the repeat settings to start

Repeat Info: The repeat settings chosen for the task.

Last Run: The date and time the task was last run, if applicable.

Active: To activate tasks, simply click the check box under "Active" in the task list. To deactivate a task, uncheck it. By default, tasks are checked as active once created.

Details: This link will bring up the task details and repeat settings, which then may be edited and re-saved if needed.

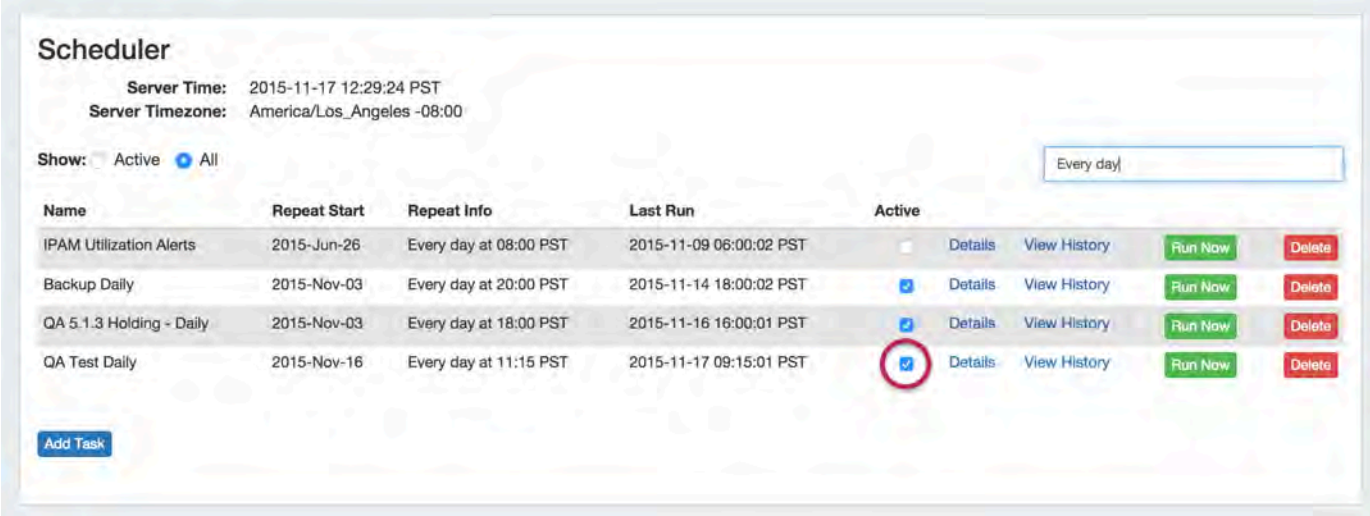
View History: This link will show the log for task actions.

Run Now: The "Run Now" button will run the scheduled task when pressed, regardless of the scheduled repeat settings.

Delete: Deletes the task.

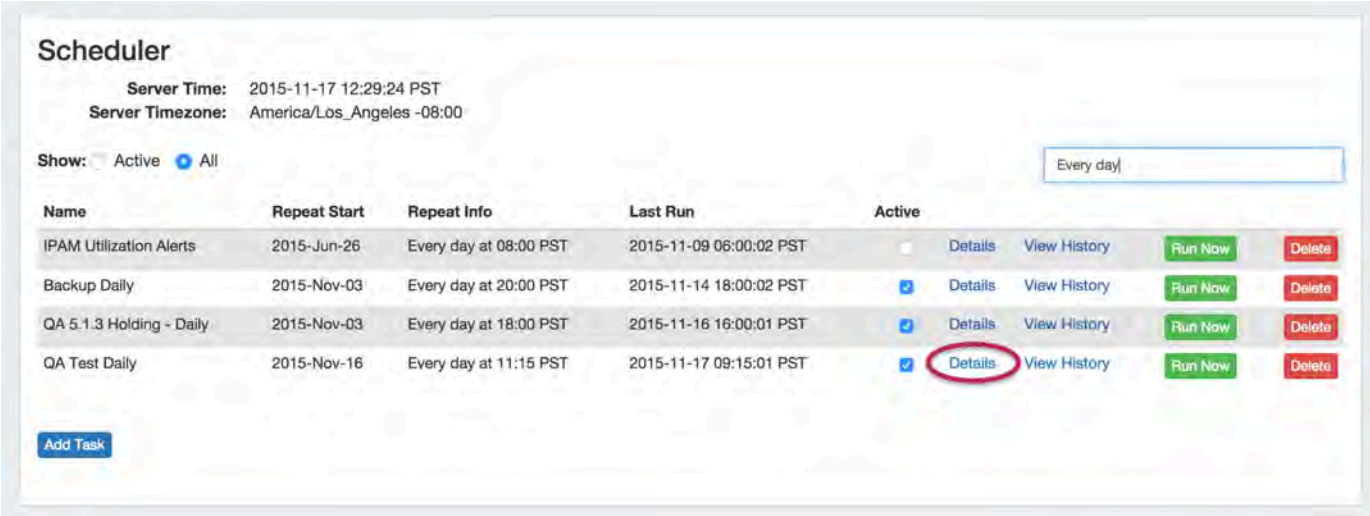
Activate / Deactivate Tasks

Click on the check box under "Active" for the task

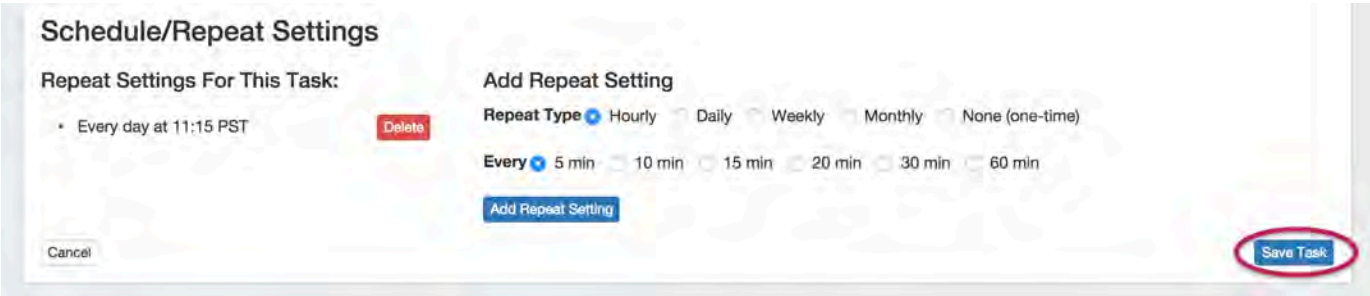


Edit a Task

Click on the "Details" link for the task, then edit the Task Detail and Repeat Settings that appear below as needed.

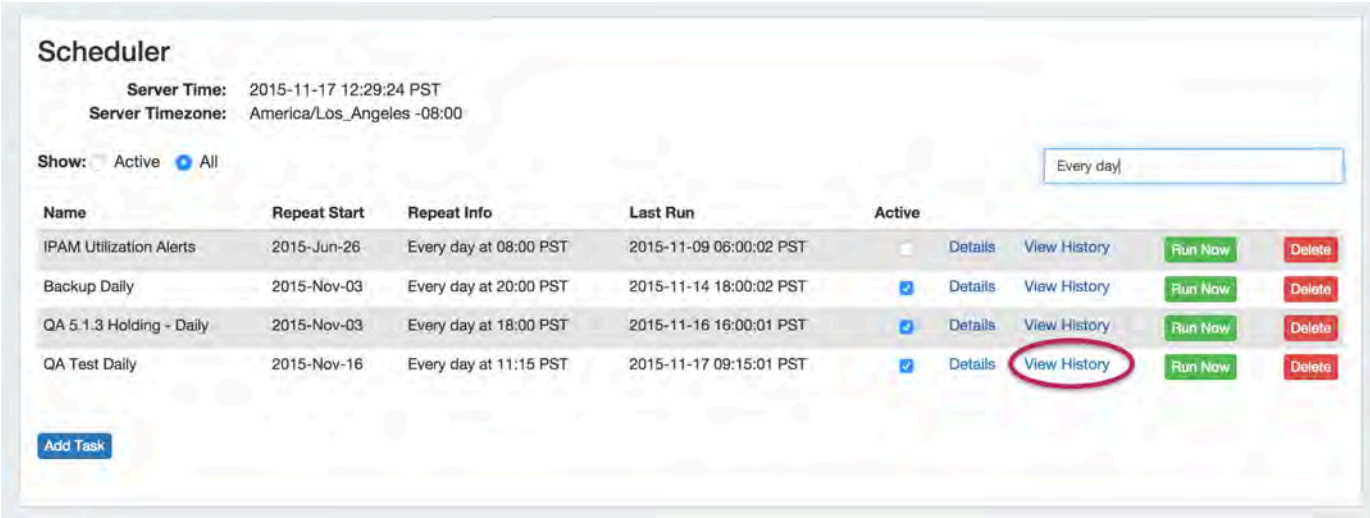


Hit "Save Task" when finished to save your changes, or select "Cancel" to close the view without saving.



View Task Log

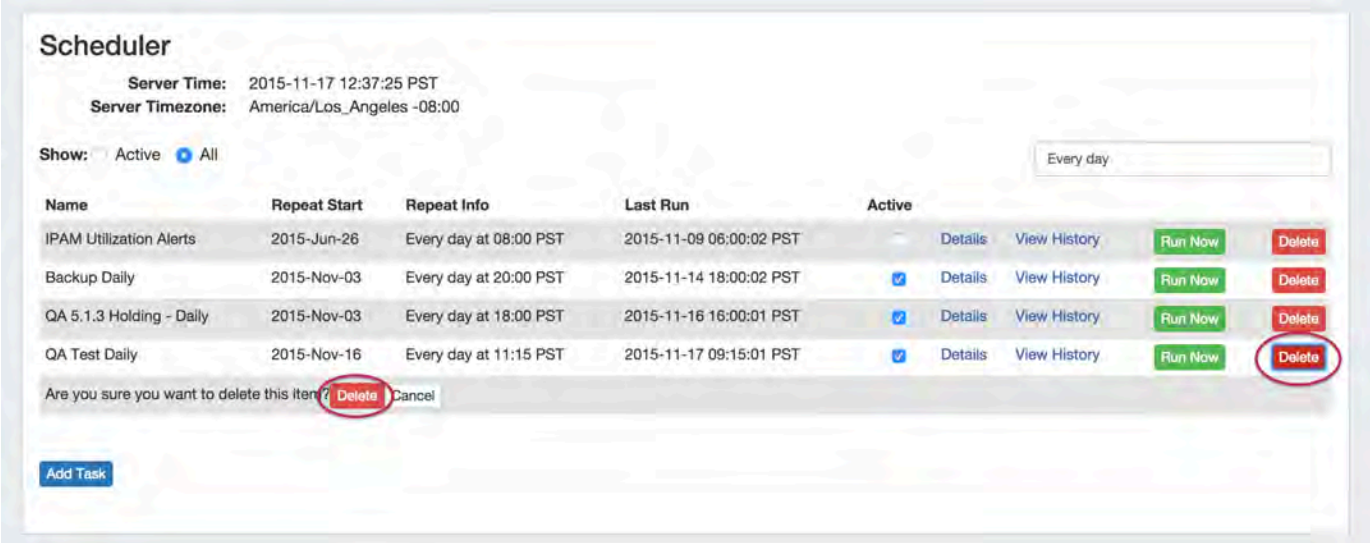
Click on the "History" link for the task.



The log details for the task will be shown below. When done, click on the "Close" button to exit the view.

Delete a Task

To delete a task, hit the "Delete" button at the end of the row in the Task List. You will be presented with message asking if you are sure you wish to delete the task. Click on the "Delete" button next to the message to verify the deletion, or hit "Cancel" to exit without deleting.



Enabling the Scheduler
To enable the scheduler in a local instance, add the following to cron:
* * * * * /path/to/php /path/to/ProVision/scheduler/task-runner.php

SWIP / RIR Integration

Overview

ProVision supports updating SWIP/RPSL functions for ARIN and RIPE blocks through simple reassigns using the RIR Integration action.

RIR Integration

When do I use RIR Integration and Why?

From ARIN.net:

Organizations that receive space allocations from ARIN, either directly or as a downstream customer, must provide reassignment information back to ARIN. This information must be sent within seven days of the reassignment so that the WHOIS database may be maintained. ARIN also uses utilization history, projected requirements, and other information in order to make future space allocations.

From RIPE.net:

All RIPE assignments and allocations must be registered in the RIPE Database. This is necessary to ensure uniqueness and to support network operations.
Only allocations and assignments registered in the RIPE Database are considered valid. Registration of objects in the database is the final step in making an allocation or assignment. Registration data (range, contact information, status etc.) must be correct at all times (i.e. they have to be maintained).The RIPE community's policies require LIRs to register an inetnum object in the RIPE Database for their own infrastructure and customers' networks. LIRs must ensure registration information is correct and up to date at all times.

What is 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.

Workflow

1) LIR Setup

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. LIRs are set up and managed from the IPAM Admin area of ProVision, and thus require Admin level permissions to set up.

While setting up the LIR, the POCs / Contact fields that are filled in will later be used for the RIR Integration.

For detailed step by step instructions, see:

LIR Management and Use

ARIN LIR Setup and Use

RIPE LIR Setup and Use

2) Assign IP blocks to the ARIN or RIPE RIRs

Assign an IP block to a Resource using the IPAM Gadget or the Assign function from the IPAM Manage screen. See Working with IP Blocks for additional detail.

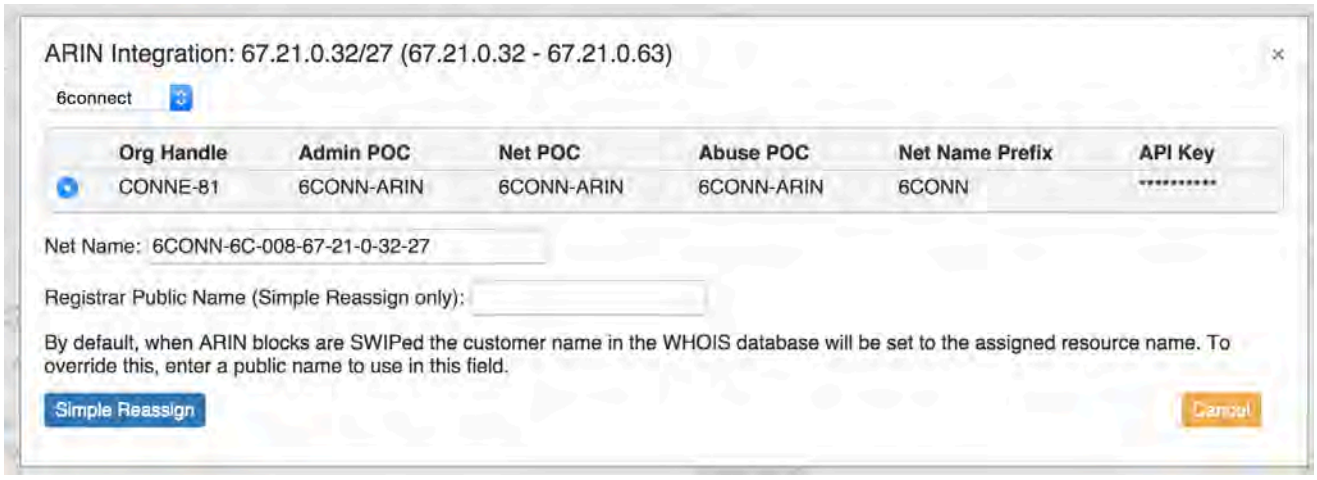
3) RIR Integration

Once LIRs have been configured, and blocks assigned under the applicable RIR, you will be able to use the **RIR integration** feature from the Action Menu on the IPAM Manage screen or IPAM Gadget.

Depending on the RIR associated with the block, either an ARIN or RIPE Integration box will pop up.

ARIN / Update SWIP

Select the desired LIR, verify the Net Name / Public Name if desired, and hit "Simple Reassign" or "Cancel".

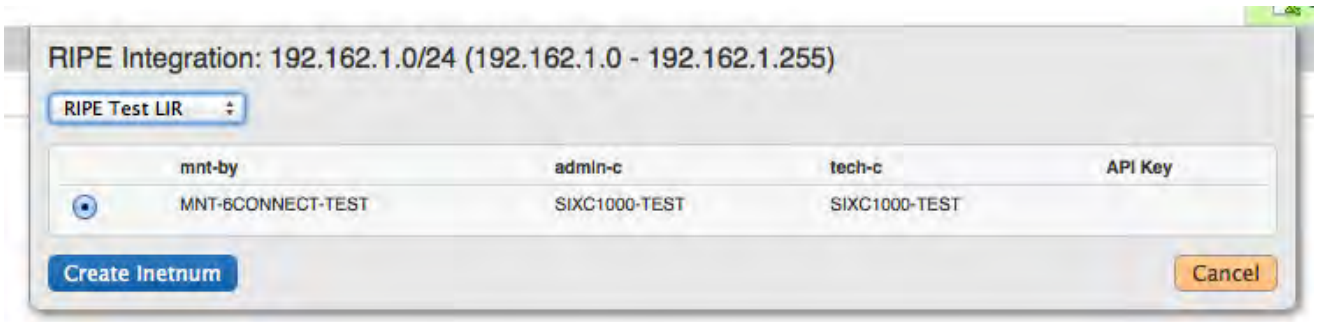


For detailed step by step instructions, see:

ARIN LIR Setup and Use

RIPE / Update RPSL

Identify which LIR data you want to use for the netnum update, and select either "Create Inetnum" or "Cancel" to exit.



For detailed step by step instructions, see:

RIPE LIR Setup and Use

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

API v1

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 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.
- 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

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

The next 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

6connect API - Getting Started with the SDK

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

SDK for PHP

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.

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.

SDK for Python

> **In Progress**
> This section is in progress. More information will be added as we improve and refine our new Python SDK!

Install the SDK

Download the file 6connect-Python-SDK.zip

Configure the SDK Security Credentials

- Extract the zipped tar file to a directory.
- Open the apiclient.py 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.

Table of Contents

Authentication Testing

Authentication Testing

testSSH

URL	/api/v1/api.php?target=auth&action=testSSH
Description	Returns success or failure of a connection to an external server via SSH.
Returns	**Examples:** SUCCESSFUL: {"success":1,"message":"Success!"} ERROR: *{'success':0, 'message':'error message'}*

Required Parameters

Name	Type	Example	Description
SSHServer	STRING	totally.awesome.com	IP or FQDN of server.
SSHPort	NUMBER	22	Port ssh is running on.

Optional Parameters

Name	Type	Example	Description
username	STRING	jsmith	Username on target server.
password	STRING	password123	Password for user.
directory	STRING	/tmp	Directory to attempt to access after successful login.

Example URL	/api/v1/api.php?target=auth&action=testSSH&username=jsmith&password=password123&directory=%2Fvar%2Fnamed%2F6conn

testLDAP

URL	/api/v1/api.php?target=auth&action=testLDAP
Description	Test basic connectivity to an LDAP server. Does not test actual authenticaion against server.
Returns	**Examples:** SUCCESSFUL: *{'success':1, 'id':'12345'}* ERROR: *{'success':0, 'message':'unable to add block'}*>

Required Parameters	

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=ldap.awesome.com&ldapMode=None

testSecure64

URL	/api/v1/api.php?target=auth&action=testSecure64
Description	Returns success or failure of a connection to an Secure64 DNS appliance.
Returns	**Examples:**

SUCCESSFUL	{"success":1,"message":"Success!"}
ERROR	*{'success':0, 'message':'error message'}*

Required Parameters

Name	Type	Example	Description
SSHServer	STRING	totally.awesome.com	IP or FQDN of server.
SSHPort	NUMBER	22	Port ssh is running on.

Optional Parameters

Name	Type	Example	Description
username	STRING	jsmith	Username on target server.
password	STRING	password123	Password for user.
directory	STRING	/tmp	Directory to attempt to access after successful login.

Example URL	/api/v1/api.php?target=auth&action=testSecure64&username=jsmith&password=password123&directory=%2Fvar%2Fnamed%2F6

Backup

Backup

backup now	
URL	/api/v1/api.php?target=backup&action=now
Description	Performs a manual backup to the designated location.
Returns	**Examples:** SUCCESSFUL: {"success":1,"message":"Backup via curl complete: "} ERROR: *{"success":0, "message":"Error Message"}>*
Required Parameters	**Name** | **Type** | **Example** | **Description** type | STRING | curl | File transfer type. Acceptable values are 'curl' or 'scp'
Optional Parameters	**Name** | **Type** | **Example** | **Description** resource | INTEGER | 2213 | The server resource id to backup to. If left blank, backs up to the 6connect cloud.
Example URL	**Backup to designated server:** /api/v1/api.php?target=backup&action=now&type=scp&resource_id=2213 **Backup to 6connect cloud:** /api/v1/api.php?target=backup&action=now&type=curl

When specifying a backup server resource ID, that server must have the Hostname, Username, and Password fields correctly provided in ProVision.

Log Management

Log Management

Get

URL	/api/v1/api.php?target=log&action=get
Description	Returns a list of log entries. Use optional parameters to filter the list.
Returns	**Examples:**

SUCCESSFUL	*{"success":1,"message":"Search Successful.","data":[{"logId":"31568","time": "2012-05-07 17:44:43", "logLevel":"INFO","userId":"39","userName": "user@6connect.com","logCategory": "User","message": "User Doe (user@6connect.com) logged in via local authentication","ip":"107.111.0.228"}]}*
ERROR	*{'success':0, 'message':'error message'}*

Data Detail

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

Required Parameters	None*

*There are no required parameters, but at least one optional parameter must be provided for the call to succeed.

Optional Parameters	

Name	Type	Example	Description
block_id	INTEGER	310	Id of the IPAM netblocks to which the logs belong
get_attributes	BOOLEAN	1	Display the log attributes along with the log into the client's response. (The attributes for each log record). Valid values are 1 (true) and 0 (false).
log_id	INTEGER	24	Unique log entry id.
time_min	DATETIME	2015-05-07 [21:00:00]	Retrieve logs starting at this Date and optional time year to second.
time_max	DATETIME	2015-05-07 [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.
orderby	STRING	log_id	Order results by log_id, time, log_level
order	STRING	ASC	Order by ascending / descending (ASC / DESC).
offset	INTEGER	50	Offset from 0 to retrieve log entries
username	STRING	user@6connect.com	The unique user name associated with the log entry.
log_category	STRING	IPAM	The 6connect category for the log entry (User, IPAM, Resource Holder, DNS, Peering, Assistant, NTP, Reporting).
log_level	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
search	STRING	Aggregate Added	Search for a string in the logs. It searches in 'message', 'username', 'time', 'ip' and 'log_category'
time	DATETIME	2015-05-07 [21:00:00]	Search logs from a specific time.

Example URL	/api/v1/api.php?target=log&action=get&block_id=310&order_by=log_id&order=DESC

Scheduler 1

This API is in beta and subject to change.

Scheduler

addTask

URL	/api/v1/api.php?target=scheduler&action=addTask
Description	Add a new scheduled task. Request is POST-only. Params are a single JSON object:
Returns	**Examples:** SUCCESSFUL: {"success":1,"message":"Message "} ERROR: *{"success":0, "message":"Error Message"}>*
Required Parameters	

Name	Type	Example	Description
user_id	INTEGER	100	User ID
name	STRING	Process Holding Tank	Name given to the new
class_name	STRING	\scheduler\Tasks\ProcessHoldingTask	PHP-namespaced name for the task class. Currently available: \scheduler\Tasks\ProcessHoldingTask \scheduler\Tasks\BackupTask \scheduler\Tasks\DNSPushTask
data		{}	
active	BOOL	true	
repeat_items			

Optional Parameters

Name	Type	Example	Description
repeat_start	STRING	2015-05-06	Date to start repeating task. Date must be in YYYY-MM-DD format.
repeat_end	STRING	null	Date to end repeating task or null. Date must be in YYYY-MM-DD format.
repeat_hour	INTEGER	8	Hour of day to repeat, from 0-23
repeat_minute	INTEGER	0	Minute of that hour to repeat, from 0-59
repeat_day	INTEGER	1	Day of month, from 1-31
repeat_week	INTEGER	1	Week of month, from 1-5
repeat_weekday	INTEGER	1	Days to repeat. Sun-Sat are from 0-6 respectively

updateTask

URL	/api/v1/api.php?target=scheduler&action=updateTask
Description	Update a scheduler task
Returns	**Examples:** SUCCESSFUL: {"success":1,"message":"*Message*"} ERROR: *{"success":0, "message":"Error Message"}>*
Required Parameters	

Name	Type	Example	Description
id	INTEGER	4	ID of the task to be updated
user_id	INTEGER	100	User ID
name	STRING	Process Holding Tank	Name given to the new
class_name	STRING	\scheduler\Tasks\ProcessHoldingTask	PHP-namespaced name for the task class. Currently available: \scheduler\Tasks\ProcessHoldingTask \scheduler\Tasks\BackupTask \scheduler\Tasks\DNSPushTask
data		{}	
active	BOOL	true	
repeat_items			

Optional Parameters

Name	Type	Example	Description
repeat_start	STRING	2015-05-06	Date to start repeating task. Date must be in YYYY-MM-DD format.
repeat_end	STRING	null	Date to end repeating task or null. Date must be in YYYY-MM-DD format.
repeat_hour	INTEGER	8	Hour of day to repeat, from 0-23
repeat_minute	INTEGER	0	Minute of that hour to repeat, from 0-59
repeat_day	INTEGER	1	Day of month, from 1-31
repeat_week	INTEGER	1	Week of month, from 1-5
repeat_weekday	INTEGER	1	Days to repeat. Sun-Sat are from 0-6 respectively

deleteTask

URL	/api/v1/api.php?target=scheduler&action=deleteTask
Description	Delete a scheduler task
Returns	**Examples:** SUCCESSFUL: {"success":1,"message":"Task \"New Task\" (2) deleted","data":null} ERROR: *{"success":0, "message":"Error Message"}>*
Required Parameters	

Name	Type	Example	Description
taskId	INTEGER	4	ID of the task to delete

Optional Parameters	None
Example URL	/api/v1/api.php?target=scheduler&action=deleteTask&taskId=2

executeTask

URL	/api/v1/api.php?target=scheduler&action=executeTask
Description	Execute a task immediately
Returns	**Examples:** SUCCESSFUL: {"success":1,"message":"Holding tank processed. 0 IPv4 and 0 IPv6 blocks moved to the available pool. ","data":null} ERROR: *{"success":0, "message":"Error Message"}>*
Required Parameters	**Name** | **Type** | **Example** | **Description** taskId | INTEGER | 4 | ID of the task to run
Optional Parameters	None
Example URL	/api/v1/api.php?target=scheduler&action=executeTask&taskId=1

getTaskHistory

URL	/api/v1/api.php?target=scheduler&action=getTaskHistory
Description	Get history for a specific task
Returns	**Examples:** SUCCESSFUL: { "success": 1, "data": [{ "log_id": "1005037", "time": "2015-05-07 12:29:45", "log_level": "6", "user_id": "31", "username": "usernamehere", "log_category": "System", "message": "task_id=1 task=Process Holding Tank action=finished message=Holding tank processed. 0 IPv4 and 0 IPv6 blocks moved to the available pool.", "ip": null }, { "log_id": "1005033", "time": "2015-05-07 12:28:50", "log_level": "6", "user_id": "31", "username": "usernamehere", "log_category": "System", "message": "task_id=1 task=Process Holding Tank action=finished message=Holding tank processed. 1 IPv4 and 0 IPv6 blocks moved to the available pool.", "ip": null }, { "log_id": "1005030", "time": "2015-05-07 12:28:14", "log_level": "6", "user_id": "31", "username": "usernamehere", "log_category": "System", "message": "Task \"Process Holding Tank\" (1) added", "ip": null }] } ERROR: *{"success":0, "message":"Error Message"}>*
Required Parameters	**Name** | **Type** | **Example** | **Description** taskId | INTEGER | 1 | ID of the task to view
Optional Parameters	None
Example URL	/api/v1/api.php?target=scheduler&action=getTaskHistory&taskId=1

getTasks

URL	/api/v1/api.php?target=scheduler&action=getTasks

Description	Gets a list of scheduled tasks and their repeat settings
Returns	**Examples:** SUCCESSFUL: { "success": 1, "data": [{ "id": "1", "user_id": "31", "name": "Process Holding Tank", "class_name": "\\scheduler\\Tasks\\ProcessHoldingTask", "active": true, "data": null, "last_run": "2015-05-07 12:29:45", "repeat_items": [{ "id": "1", "task_id": "1", "repeat_start": "2015-05-06", "repeat_end": null, "repeat_interval": null, "repeat_year": "*", "repeat_month": "*", "repeat_day": "*", "repeat_week": "*", "repeat_weekday": "4", "repeat_hour": "20", "repeat_minute": "0" }, { "id": "2", "task_id": "1", "repeat_start": "2015-05-06", "repeat_end": null, "repeat_interval": null, "repeat_year": "*", "repeat_month": "*", "repeat_day": "*", "repeat_week": "*", "repeat_weekday": "5", "repeat_hour": "20", "repeat_minute": "0" }, { "id": "3", "task_id": "1", "repeat_start": "2015-05-06", "repeat_end": null, "repeat_interval": null, "repeat_year": "*", "repeat_month": "*", "repeat_day": "*", "repeat_week": "*", "repeat_weekday": "6", "repeat_hour": "20", "repeat_minute": "0" }, { "id": "4", "task_id": "1", "repeat_start": "2015-05-06", "repeat_end": null, "repeat_interval": null, "repeat_year": "*", "repeat_month": "*", "repeat_day": "*", "repeat_week": "*", "repeat_weekday": "0", "repeat_hour": "20", "repeat_minute": "0" }] }, { "id": "2", "user_id": "31", "name": "New Task", "class_name": "\\scheduler\\Tasks\\TestTask", "active": true, "data": null, "last_run": "2015-05-07 12:35:41", "repeat_items": [{ "id": "5", "task_id": "2", "repeat_start": "2015-05-06", "repeat_end": null, "repeat_interval": null, "repeat_year": "*", "repeat_month": "*", "repeat_day": "*", "repeat_week": "*", "repeat_weekday": "1", "repeat_hour": "21", "repeat_minute": "0" }, { "id": "6", "task_id": "2", "repeat_start": "2015-05-06", "repeat_end": null, "repeat_interval": null, "repeat_year": "*", "repeat_month": "*", "repeat_day": "*", "repeat_week": "*", "repeat_weekday": "2", "repeat_hour": "21", "repeat_minute": "0" }, { "id": "7", "task_id": "2", "repeat_start": "2015-05-06", "repeat_end": null, "repeat_interval": null, "repeat_year": "*", "repeat_month": "*", "repeat_day": "*", "repeat_week": "*", "repeat_weekday": "3", "repeat_hour": "21", "repeat_minute": "0" }, { "id": "8", "task_id": "2", "repeat_start": "2015-05-06", "repeat_end": null, "repeat_interval": null, "repeat_year": "*", "repeat_month": "*", "repeat_day": "*", "repeat_week": "*", "repeat_weekday": "4", "repeat_hour": "21", "repeat_minute": "0" }, { "id": "9", "task_id": "2", "repeat_start": "2015-05-06", "repeat_end": null, "repeat_interval": null, "repeat_year": "*", "repeat_month": "*", "repeat_day": "*", "repeat_week": "*", "repeat_weekday": "5", "repeat_hour": "21", "repeat_minute": "0" }] }] } ERROR: *{"success":0, "message":"Error Message"}>*
Required Parameters	None
Optional Parameters	None
Example URL	/api/v1/api.php?target=scheduler&action=getTasks

Zone Templates

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	**Examples:** SUCCESSFUL: {"success":1,"message":"Found 1 records for template \"Awesome Template\".","data":{"templateId":"1011","nam 14:01:24","userId":"112","soa":null,"refresh":"14400","retry":"3600","expire":"604800","minimum":null,"ttl":null,"use ERROR: *{'success':0, 'message':'error message'}*
Required Parameters	None
Optional Parameters	templateId (NUMBER, Example: 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	**Examples:** SUCCESSFUL: {"success":1,"message":"Template updated","data":{"templateId":"1011","name":"Awesome Template","created":"2 23:15:52","modified":"2013-08-05 23:15:52","userId":"112","soa":"ns1.test.net hostmaster.ns1.test.net","refresh":"14400","retry":"3600","expire":"604800","minimum":null,"ttl":false,"userName":"anna@6connect.c ERROR: *{'success':0, 'message':'Error updating template: error details'}>*
Required Parameters	name (STRING, Example: Test Template) — The name of the template to be created or updated.

Optional Parameters	

Name	Type	Example	Description
soa	STRING	ns1.test.net hostmaster.ns1.test.net	A valid SOA for the template in for format
ttl	INTEGER	86400	The TTL for the zone template, which is the default expiration time for all r 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 parame followed with their position in the count. In this example, the first record wo for the first record followed by _1, the second record _2, and so on. This wi the template follow.
host_1	STRING		The DNS record value.
ttl_1	INTEGER	3600	TTL of the specific host record.
type_1	STRING	A	A valid DNS record type.
value_1	IP	1.2.3.4	A valid IPv4 or IPv6 address.

Example URL	api/v1/api.php?target=zoneTemplate&action=update&templateId=1011&count_records=1&name=Awesome+Template&soa=ns1.te &refresh=14400&retry=3600&expire=604800&minimum=3600&value_0=undefined+undefined&host_1=www&ttl_1=3600&type_1=A

Delete

URL	/api/v1/api.php?target=zoneTemplate&action=delete
Description	Deletes a DNS template.
Returns	**Examples:**

SUCCESSFUL	{"success":1,"message":"Template \"Test Template\" delete."}
ERROR	{"success":0,"message":"No template found for templateId \"1005\"."}

Required Parameters

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 Management Version 2

Overview

DHCP Management Version 2 integrates DHCP management with ProVision's resource and permissions hierarchy, as well as the IP Management system. Individual DHCP servers can be assigned via Resource Permissions to different internal user groups, to be managed by only the appropriate parties.

Under DHCPv2 there is no distinct “DHCP Server” type or section – instead there is a “DHCP Module” which, when attached as a child to an existing resource, transforms it into a DHCP-enabled device. The most common use is to take the generic “Server” Section and turn it into a DHCP Server by attaching the DHCP Module as a child. This configuration allows users to add functionality to a basic resource and provides a cleaner management interface.

API Updates

The DHCPv1 API operated via calls to the DHCPServer and the DHCPEntry endpoint families. However, now that DHCPv2 is contained entirely within the resource system, most of the API calls to manipulate DHCP data do so using the Resource family of API endpoints to modify specific Resource attributes reserved for DHCP functionality.

DHCP API How-To

Relate with Resources

The DHCPv2 system builds upon the ProVision Resource API. With the exception of a few configuration commands all DHCPv2 API commands use the Resource family of API endpoints.

How to attach the DHCP Module as a child

As described above, DHCPv2 functionality is enabled on a particular resource by attaching a DHCP Module as a child. A command to do this is as follows:

```
            [ProVision root]/api/v1/api.php?target=resource&action=add

            data:
meta[type]: dhcp_module
meta[name]: [parent resource id] DHCP Module
meta[parent_id]: [parent resource id]
```

The special resource type “dhcp_module” indicates to ProVision that the DHCP system is enabled for the parent object. The attributes associated with the “dhcp_module” resource govern the DHCP system's behavior.

Updating the attributes of a DHCP Server uses a Resource Update command:

```
[ProVision root]/api/v1/api.php?target=resource&action=update&meta[id]=2178
&meta[type]=dhcp_module&fields[_dhcp_attributes][]={"type":"ISC","notes":"notes go
here","username":"username","port":"port","config_test":"/etc/init.d/dhcpd
configtest","server_stop":"/etc/init.d/dhcpd stop","server_start":"/etc/init.d/dhcpd
start","config_path":"/tmp/dhcpd.conf","option_routers":"192.168.0.0","option_domain_na
line 1","freeLine2":"free line 2","freeLine3":"free line 3"}
```

This command appears rather complicated, but can be broken apart into reasonable pieces. The first section:

```
target=resource&action=update&meta[id]=2178&meta[type]=dhcp_module
```

is familiar from other parts of ProVision. We are updating a resource of type “dhcp_module” whose resource id is 2178. The second section of the command details the update values, starting with

```
fields[_dhcp_attributes][]=
```

which contains a JSON-encoded string of all the fields specific to a DHCP server's function. When expanded into its full object form it is substantially easier to digest:

```
{
            "type":"ISC",
            "notes":"notes go here",
            "username":"username",
            "port":"port",
            "config_test":"/etc/init.d/dhcpd configtest",
            "server_stop":"/etc/init.d/dhcpd stop",
            "server_start":"/etc/init.d/dhcpd start",
            "config_path":"/tmp/dhcpd.conf",
            "option_routers":"192.168.0.0",
            "option_domain_name_servers":"ns1.6connect.com",
            "option_domain_name":"6connect.com",
            "authoritative":"1",
            "default_lease_time":"600",
            "max_lease_time":"7200",
            "local_port":"67",
            "log_facility":"local7",
            "password":"password",
            "server_ip":"192.168.0.1",
            "freeLines":3,
            "freeLine1":"free line 1",
            "freeLine2":"free line 2",
            "freeLine3":"free line 3"
}
```

This object describes all the most common DHCP server configuration options. For a full explanation of each of the fields, see the Detailed API Specification later in this document.

Please note that the object above must be passed to the DHCP system as a JSON-encoded string. It must be passed into the special

"_dhcp_attributes" attribute for it to be functional, as in the example URL.

Create DHCP IP Aggregates

For details on how to manage IP aggregates using ProVision's IPAM API, see API Module - IPAM.

Of particular interest to DHCP management is the addition of DHCP aggregates, which are sections of IP space marked as available for use by the DHCPv2 system.

How to add a DHCP Aggregate

An example command to add a DHCP Aggregate is:

```
[ProVision root]/api/v1/api.php?target=ipam&action=add&block=192.168.0.0/24&rir=
1918&vlan=&tags=&region=&resourceId=1282&allowSubAssignments=true
```

The important part to note is that the IP block is being assigned to resourceId 1282, which corresponds to the DHCP Available resource. The DHCP Available resource is a system-level resource which is used to hold all unassigned DHCP IP addresses. Every instance has its own DHCP Available resource, whose id can be found with the following command:

```
[ProVision root]/api/v1/api.php?target=resource&action=get&slug=dhcp-available
```

New DHCP subnets and hosts draw their IPs from this pool. If there are no IPs in the DHCP Available pool new subnets and hosts will not be able to be created.

DHCP IP aggregates are fetched, updated, split, and deleted using the standard IPAM management API endpoints. Please see the IPAM API Documentation. for details.

Subnets and Hosts

Every DHCP configuration file consists primarily of Subnet and Host declarations, mapping out what IP addresses are available for what purpose. In DHCPv2, DHCP Pools are reusable components that can be attached to several DHCP Servers in order to build flexible, responsive DHCP configurations.

In ProVision DHCPv2 all DHCP Pools regardless of whether they span Subnets or individual Hosts require that a "dhcp_pool" resource be created to govern them.

How to create DHCP Pools

Similar to how the "dhcp_module" resource was created above, the command to create a DHCP Pool is as follows:

```
[ProVision root]/api/v1/api.php?target=resource&action=add&meta[type]=dhcp_pool
&meta[name]=New
Subnet&fields[_dhcp_type][]=subnet&fields[_dhcp_pool_attributes][]={"mac":"","rangeStar
Line 1","freeLine2":"Free Line 2","freeLine3":"Free Line 3"}
```

The first half of this command is relatively straightforward:

```
target=resource&action=add&meta[type]=dhcp_pool&meta[name]=New Subnet
```

This section informs the API that we wish to create a new, empty "dhcp_pool" resource whose name is "New Subnet."

```
fields[_dhcp_type][]=subnet&fields[_dhcp_pool_attributes][]={"mac":"","rangeStart":"",
"rangeEnd":"","freeLines":3,"freeLine1":"Free Line 1","freeLine2":"Free Line
2","freeLine3":"Free Line 3"}
```

The second half of the command behaves in a similar manner to the “dhcp_module.” The “_dhcp_pool_attributes” field holds a JSON-encoded string which describes the dhcp_pool resource. When expanded, the JSON string becomes the following object:

```
{
            "mac":"",
            "rangeStart":"",
            "rangeEnd":"",
            "freeLines":3,
            "freeLine1":"Free Line 1",
            "freeLine2":"Free Line 2",
            "freeLine3":"Free Line 3"
}
```

For a full explanation of each of the fields, see the Detailed API Specification.

> Please note that the object above must be passed to the DHCP system as a JSON-encoded string. It must be passed into the “_dhcp_pool_attributes” attribute for it to be functional, as in the example URL.

Once a dhcp_pool resource is in the system it can be updated with IP data obtained from the IP Management system. Under DHCPv2, the DHCP system uses all the standard IPAM API endpoints and can make use of both the smartAssign and the directAssign methods. Please see the IPAM API documentation for details.

How to smart-assign a DHCP IP range from the DHCP Available resource to a dhcp_pool resource

An example command for smart-assigning a DHCP IP range from the DHCP Available resource to a newly-created dhcp_pool resource is as follows:

```
[ProVision root]/api/v1/api.php?target=ipam&action=smartAssign&resourceId=2180&
type=ipv4&mask=31&rir=1918&assignedResourceId=1282
```

In this example we are using the IPAM API endpoint to smart-assign an IPv4 /31 from the DHCP Available resource (resource id 1282) to the newly-created dhcp_pool object (resource id 2180). This action removes this IP range from the available pool and prevents it from being used by other parts of ProVision.

Once an IP block is assigned to a dhcp_pool it should be updated with the proper range start and range end. A Resource Update command is used for this.

```
[ProVision root]/api/v1/api.php?target=resource&action=update&meta[type]=dhcp_pool&
meta[name]=Another
Test&fields[_dhcp_type][]=subnet&fields[_dhcp_pool_attributes][]={"mac":"","rangeStart"
```

The key information here is that the “rangeStart” and the “rangeEnd” fields in the JSON-encoded '_dhcp_pool_attributes' attribute have been populated with the beginning and end of the IP range assigned by smart-assign. Also note that a new field is being populated as '_dhcp_ip_id', which contains the IPAM id of the newly-assigned IP block.

When assigning dhcp_pools covering a single host the steps are much the same, but the 'mac' field in the '_dhcp_pool_attributes' object must be populated with the MAC address of the host in question.

Linking Subnets and Hosts with DHCP Servers

DHCP Pools exist as re-usable components which can be individually assigned to any number of DHCP Servers in order to assemble flexible DHCP Configurations. Once created, a DHCP Pool is not attached to any DHCP Server in the system. DHCP Pools must be linked to a server for the pool to be included in DHCP configuration pushes.

How to link a dhcp_pool and a DHCP Server

An example of building a link between a dhcp_pool and a DHCP Server is:

```
[ProVision root]/api/v1/api.php?target=resource&action=addLink&resource_id1=2178&
resource_id2=1452&relation=dhcpPoolLink
```

The Resource Linkage system controls which DHCP Pools are associated with a given DHCP Server. In the case of linking a DHCP Pool to a DHCP Server, the relation used is “dhcpPoolLink”. This is a directional link, so it is important that resource_id1 and resource_id2 do not get confused.

```
relation:  "dhcpPoolLink"
resource_id1:  the id of the dhcp_module this pool is being linked to
resource_id2:  the id of the dhcp_pool being linked
```

> It is very important that resource_id1 not be confused with resource_id2. The link will not function with the values reversed.

To undo the above and break a DHCP Pool link, use the same command but substitute “deleteLink” for the action “addLink”.

```
[ProVision root]/api/v1/api.php?target=resource&action=deleteLink&resource_id1=2178&
resource_id2=2179&relation=dhcpPoolLink
```

Pushing Configurations

Pushing configuration files and restarting a DHCP server is a fairly straightforward process.

How to push configuration files

Once the server has been configured according to the previous sections, hitting the following API endpoint will trigger a DHCP push:

```
[ProVision root]/api/v1/api.php?target=dhcp&action=push&id=2178
```

The “id” in the above string is the id of the dhcp_module resource attached to the server you whose configuration is to be pushed. The API return payload will contain success or failure codes, as well as a description of any errors which might have occurred.

When a DHCP configuration file is pushed an SSH connection is opened to the configured server using the user, password, and port supplied to the '_dhcp_attributes' attribute on the dhcp_module resource. If the system successfully connects, it will assemble a DHCP configuration from the information given to the dhcp_module's '_dhcp_attribute' attribute and then parse and add in all linked dhcp_pool resources.

After the assembled file has been transferred to the DHCP server it will be placed in the location given by 'config_path' on the dhcp_module, and then the command described in 'config_test' will be run to determine whether or not this new file parses correctly. If 'config_test' is blank or omitted, this step is skipped.

If the file parses correctly the DHCP will be stopped and restarted according to the 'server_stop' and 'server_start' commands on the DHCP module. If there are errors at any point the system backs out, replaces old config files, and reports the errors via the 'message' return field of the API call.

Detailed API Specification

A detailed listing of API endpoints related to DHCP Servers, Pools, and Links can be found here:

- API Module - DHCPv2

API Module - DHCPv2

DHCPv2 Module

The DHCPv2 system is built upon the Resource API, so actions relating to DHCP tasks are largely expressed in terms of Resource actions.

This section describes common DHCP tasks and how they are accomplished via the DHCPv2 system.

get all DHCP-enabled resources	
Description	Finds all resources from section 'dhcp_module,' which indicates that their parents are DHCP-enabled. Adding in other Resource-Get API parameters can filter this list further.
URL	/api/v1/api.php?target=resource&action=get&type=dhcp_module

Returns	**Examples:**

SUCCESSFUL:	{"success":1,"message":"Search successful","data":[{"id":"1432", "name":"1392 DHCP Module" ,"slug":"1392-dhcp-module", "type":"dhcp_module", "parent_id":"1392", "category_id":null, "attr":{"_dhcp_attributes":"{\"type\":\"ISC\", \"notes\":\"\", \"username\":\"\", \"port\":\"\", \"config_test\":\"/etc/init.d/dhcpd configtest\", \"server_stop\":\"/etc/init.d/dhcpd stop\", \"server_start\":\"/etc/init.d/dhcpd start\", \"config_path\":\"\", \"option_routers\":\"\", \"option_domain_name_servers\":\"\", \"option_domain_name\":\"\", \"authoritative\":\"1\", \"default_lease_time\":\"600\", \"max_lease_time\":\"7200\", \"local_port\":\"67\", \"log_facility\":\"local7\", \"password\":\"\", \"server_ip\":\"10.0.0.0\", \"freeLines\":0}", "_dhcp_config_id":"33"}}], "result_count":1, "found_count":1}
ERROR:	{"success":0, "message":"error message"}

Return Detail:

Name	Type	Description
id	INTEGER	ID of the dhcp_module resource
name	STRING	The name of the dhcp_module
slug	STRING	The unique reference string for this resource
type	STRING	Always 'dhcp_module
parent_id	INTEGER	The resource to which the dhcp_module is attached
category_id	INTEGER	The category to which this dhcp_module is associated
result_count	INTEGER	How many dhcp_modules are returned in this search.
found_count	INTEGER	How many dhcp_modules were found in this query, without pagination.

Attributes:

Key	Type	Description
_dhcp_attributes	JSON	A JSON-encoded string containing all the specific configuration parameters which govern this DHCP server. An expansion of the JSON object is given below in the Data Attributes section.
_dhcp_config_id	INTEGER	A reference to the DHCP Config file written within the system. This field is maintained by the DHCPv2 system itself and should not be set externally.

create a new DHCP-enabled resource

Description	A resource becomes a DHCP-enabled by adding a special "dhcp_module" resource as a child. This action is identical to a normal Resource Create command.
URL	/api/v1/api.php?target=resource&action=add&meta[type]=dhcp_module&meta[name]=2163 DHCP Module&meta[parent_id]=2163

Returns	**Examples:**

SUCCESSFUL:	{"success":1,"message":"Resource added","data":{"id": 2165,"name" :"2163 DHCP Module","slug": "2163-dhcp-module-2","type" :"dhcp_module","parent_id": 2163,"category_id": null,"attr":[]}}
ERROR:	{"success":0, "message":"error message"}

Return Detail:

Name	Type	Description
id	INTEGER	ID of the newly created dhcp_module
name	STRING	The name of the dhcp_module
slug	STRING	The unique reference string for this resource
type	STRING	Always 'dhcp_module'
parent_id	INTEGER	The resource to which the dhcp_module is attached
category_id	INTEGER	The category to which this dhcp_module is associated

update a DHCP-enabled resource with new configuration info

Description	Modifying an existing dhcp_module uses the identical commands as all other Resource-Update actions. An example of configuring
URL	/api/v1/api.php?target=resource&action=update&meta[id]=2178 &meta[type]=dhcp_module&fields[_dhcp_attributes][]={"type":"ISC" start","config_path":"/tmp/dhcpd.conf","option_routers":"192.168.0.0","option_domain_name_servers":"ns1.6connect.com","option_ line 1","freeLine2":"free line 2","freeLine3":"free line 3"}
Returns	**Examples:**

SUCCESSFUL:	{"success": 1,"message": "Resource Updated","data": {"id":"2166" ,"name":"2163 DHCP Module", "slug":"2163-c \"config_test\":\"\\\/etc\\\/init.d\\\/dhcpd configtest\" ,\"server_stop\":\"\\\/etc\\\/init.d\\\/dhcpd stop\" ,\"server_start\": \"authoritative\":\"1\", \"default_lease_time\":\"600\", \"max_lease_time\":\"7200\", \"local_port\":\"67\", \"log_facilit
ERROR:	{"success":0, "message":"error message"}

Return Detail:

Name	Type	Description
id	INTEGER	ID of the newly created dhcp_module
name	STRING	The name of the dhcp_module
slug	STRING	The unique reference string for this resource
type	STRING	Always 'dhcp_module'
parent_id	INTEGER	The resource to which the dhcp_module is attached
category_id	INTEGER	The category to which this dhcp_module is associated

Attributes:

Key	Type	Description
_dhcp_attributes	JSON	A JSON-encoded string containing all the specific configuration parameters which govern this DHCP s

remove DHCP functionality from a resource

Description	To remove DHCP functionality, delete the dhcp_module child resource. This operation uses general Resource->Delete functionality.
URL	/api/v1/api.php?target=resource&action=delete&id=2166
Returns	**Examples:**

SUCCESSFUL:	{"success":1,"message":"Deleted 2163-dhcp-module-3 (#2166)"}
ERROR:	{"success":0, "message":"error message"}

get all DHCP Pools

Description	As with the dhcp_module commands, the API endpoints governing DHCP IP Pools use the general Resource system. All the modifi
URL	/api/v1/api.php?target=resource&action=get&type=dhcp_pool
Returns	**Examples:**

SUCCESSFUL:	{"success":1,"message":"Search successful","data":[{"id":"1482","name":"Blah","slug":"blah","type":"dhcp_pool","parent_id":"1","category_id":null,'],"result_count":1,"found_count":1}
ERROR:	{"success":0, "message":"error message"}

Return Detail:

Name	Type	Description
id	INTEGER	ID of the dhcp_pool resource
name	STRING	The name of the dhcp_pool
slug	STRING	The unique reference string for this resource
type	STRING	Always 'dhcp_pool'
parent_id	INTEGER	The resource to which the dhcp_pool is attached
category_id	INTEGER	The category to which this dhcp_pool is associated
result_count	INTEGER	How many dhcp_pools are returned in this search.
found_count	INTEGER	How many dhcp_pools were found in this query, without pagination.

Attributes:

Key	Type	Description
_dhcp_type	STRING	Either 'subnet' or 'host'. Determines whether this DHCP Pool is describing a Subnet or a Hos
_dhcp_pool_attributes	JSON	A JSON-encoded string containing all the specific configuration parameters which govern this
_dhcp_ip_id	INTEGER	The id of the IPAM subnet or host which is assigned to this DHCP Pool

create a new DHCP Pool resource

Description	Uses the general Resource-Add endpoint to create a DHCP Pool resource.
URL	/api/v1/api.php?target=resource& action=add& meta[type]=dhcp_pool& meta[name]=New Subnet& fields[_dhcp_type][]=host& fields[_dhcp_pool_attributes][]={"mac":"aa:bb:cc:dd:ee:ff", "rangeStart":"", "rangeEnd":"", "freeLines":3, "freeLine1":"Free Line 1", "freeLine2":"Free Line 2", "freeLine3":"Free Line 3"}
Returns	**Examples:**

SUCCESSFUL:	{"success":1,"message":"Resource added","data":{"id":2167,"name":"New Subnet","slug":"new-subnet","type":"dhcp_pool ","parent_id":1,"category_id":null,"attr":[]}}
ERROR:	{"success":0, "message":"error message"}

Return Detail:

Name	Type	Description
id	INTEGER	ID of the newly created dhcp_pool
name	STRING	The name of the dhcp_pool
slug	STRING	The unique reference string for this resource
type	STRING	Always 'dhcp_pool'
parent_id	INTEGER	The parent resource; by default the TLR.
category_id	INTEGER	The category to which this dhcp_pool is associated

update a DHCP Pool

Description	Modifying an existing dhcp_pool uses the identical commands as all other Resource-Update actions.
URL	/api/v1/api.php?target=resource& action=update& meta[type]=dhcp_pool& meta[name]=Another Test& fields[_dhcp_type][]=subnet& fields[_dhcp_pool_attributes][]={"mac":"", "rangeStart":"10.10.10.4", "rangeEnd":"10.10.10.5", "freeLines":3, "freeLine1":"example1", "freeLine2":"example2", "freeLine3":"example3"}&fields[_dhcp_ip_id][]=92430&meta[id]=2165

Returns	**Examples:**

SUCCESSFUL:	{"success":1, "message":"Resource Updated", "data":{"id":"2165", "name":"Another Test", "slug":"2163-dhcp-module-2", "type":"dhcp_module", "parent_id":"2163", "category_id":null,"attr":{"_dhcp_type":"subnet" ,"_dhcp_pool_attributes":"{\"mac\":\"\", \"rangeStart\":\"10.10.10.4\", \"rangeEnd\":\"10.10.10.5\", \"freeLines\":3, \"freeLine1\":\"example1\", \"freeLine2\":\"example2\", \"freeLine3\":\"example3\"}", "_dhcp_ip_id":"92430"}}}
ERROR:	{"success":0, "message":"error message"}

Return Detail:

Name	Type	Description
id	INTEGER	ID of the newly created dhcp_module
name	STRING	The name of the dhcp_module
slug	STRING	The unique reference string for this resource
type	STRING	Always 'dhcp_module'
parent_id	INTEGER	The resource to which the dhcp_module is attached
category_id	INTEGER	The category to which this dhcp_module is associated

Attributes:

Key	Type	Description
_dhcp_type	STRING	Either 'subnet' or 'host'. Determines whether this DHCP Pool is describing a Subnet or a Host.
_dhcp_pool_attributes	JSON	A JSON-encoded string containing all the specific configuration parameters which govern this DHCP Pool. An expansion of the JSON object is given below in the Data Attributes section.
_dhcp_ip_id	INTEGER	The id of the IPAM subnet or host which is assigned to this DHCP Pool

delete a DHCP Pool

Description	To delete a DHCP Pool, use the standard Resource-Delete functionality
URL	/api/v1/api.php?target=resource&action=delete&id=2165
Returns	**Examples:**

SUCCESSFUL:	{"success":1,"message":"Deleted 2165-another-subnet-3 (#2165)"}
ERROR:	{"success":0, "message":"error message"}

assigning an IP address or blocks to a DHCP Pool

Description	Assigning IP addresses or blocks to a DHCP Pool resource removes them from the available pool so they cannot be assigned out a
URL	/api/v1/api.php?target=ipam&action=smartAssign&resourceId=2162&type=ipv4&mask=31&rir=1918&assignedResourceId=1282

Returns	**Examples:**

SUCCESSFUL:	{"success":1,"message":"Assigned 10.8.1.4√31 to 208.39.104.106 (2162) via Smart Assign","id":94468,"data":{"id":94468,"type":"ipv4","top_aggregate":44616,"cidr":"10.8.1.4√31","formatted_ip":"1(20:17:32","description":null,"parent":80882,"rir":"1918","lir_id":null,"notes":null,"generic_code":null,"code":null,"re
ERROR:	{"success":0, "message":"error message"}

Return Detail:

For a detailed breakdown of this endpoint's return data, please see the IPAM documentation.

get all DHCP Pool linkages

Description	The association between DHCP Pools and DHCP Modules belongs to the Resource Linkage family of endpoints. The 'relation' field should be set to the 'dhcpPoolLink' type to pull only DHCP Pool linkage information.
URL	/api/v1/api.php?target=resource&action=getLink&relation=dhcpPoolLink
Returns	**Examples:**

SUCCESSFUL:	{"success":1 ,"message":"Search successful", "data":{"meta":{"totalRecords":"3", "retrieved":3}, "0":{"id":"22", "resource_id1":"1292", "resource_id2":"1302", "relation":"dhcpPoolLink"}, "1":{"id":"2", "resource_id1":"1292", "resource_id2":"1452", "relation":"dhcpPoolLink"}, "2":{"id":"12", "resource_id1":"1422", "resource_id2":"1482", "relation":"dhcpPoolLink"}}}
ERROR:	{"success":0, "message":"error message"}

Return Detail:

Name	Type	Description
id	INTEGER	Id of the pool-module linkage
resource_id1	INTEGER	The id of the dhcp_module resource
resource_id2	INTEGER	The id of the dhcp_pool resource
relation	STRING	The relation type. Always 'dhcpPoolLink'

Meta Attributes:

Name	Type	Description
totalRecords	INTEGER	How many records were found by this query, without pagination.
retrieved	INTEGER	How many records were returned by this query, with pagination.

add a new DHCP Pool linkage

Description	Adds a new link between a DHCP Pool and a dhcp_module resource. A single pool can be linked to many dhcp_modules, and a single dhcp_module can have any number of linked pools.
URL	/api/v1/api.php?target=resource&action=addLink&resource_id1=1292&resource_id2=2162&relation=dhcpPoolLink

Returns	**Examples:**

SUCCESSFUL:	{"success":1,"message":"Resource link added"}
ERROR:	{"success":0, "message":"error message"}

Data Detail:

Name	Type	Description
resource_id1	INTEGER	The id of the dhcp_module resource
resource_id2	INTEGER	The id of the dhcp_pool resource
relation	STRING	The relation type being added. Always 'dhcpPoolLink'

delete DHCP Pool linkages

Description	Deletes a link between a dhcp_module and a dhcp_pool. Uses the standard Resource Linkage endpoints.
URL	/api/v1/api.php?target=resource&action=deleteLink&id=22
Returns	**Examples:**

SUCCESSFUL:	{"success":1,"message":"Resource link(s) deleted."}
ERROR:	{"success":0, "message":"error message"}

push a DHCP config

Description	Builds a DHCP configuration from the attributes assigned to a dhcp_module and all of the linked dhcp_pools. Pushes that config to the configured DHCP server, tests it against the config parsing function, then restarts the server with the new configuration.
URL	/api/v1/api.php?target=dhcp&action=push&id=1292
Returns	**Examples:**

SUCCESSFUL:	{"success":1,"message":"Pushes Attempted.","data":[[1,"1292","381 DHCP Module","Configuration successfully pushed."]]}
ERROR:	{"success":0, "message":"error message"}

Data Detail

Name	Type	Description
id	INTEGER	The id of the dhcp_module resource whose configuration is to be pushed.

DHCP search

Description	Searches DHCP information by name, mac, or IP.

URL	/api/v1/api.php?target=dhcp&action=search&searchType=name&searchValue=Blah /api/v1/api.php?target=dhcp&action=search&searchType=mac&searchValue=22: /api/v1/api.php?target=dhcp&action=search&searchType=ip&searchValue=13.0.0.255

Returns

Examples:

SUCCESSFUL:	{"success":1,"message":"Search Successful","data":[{"id":"1482","name":"BlahBlah","slug":"blah","type":"dhcp_pool","parent_id":"1","category_id"
ERROR:	{"success":0, "message":"error message"}

Return Detail

Name	Type	Description
id	INTEGER	ID of the dhcp_module resource
name	STRING	The name of the dhcp_module
slug	STRING	The unique reference string for this resource
type	STRING	Always 'dhcp_pool'
parent_id	INTEGER	The resource to which the dhcp_module is attached
category_id	INTEGER	The category to which this dhcp_module is associated

Attributes

Key	Type	Description
_dhcp_type	STRING	Either 'subnet' or 'host'. Determines whether this DHCP Pool is describing a Subnet or a Hos
_dhcp_pool_attributes	JSON	A JSON-encoded string containing all the specific configuration parameters which govern thi
_dhcp_ip_id	INTEGER	The id of the IPAM subnet or host which is assigned to this DHCP Pool.
_dhcp_links	INTEGER	The id of dhcp links.

Required Parameters

Name	Type	Example	Description
searchType	STRING	name	Type of search performing. Valid values are 'name', 'mac', and 'ip'.
searchValue	STRING	Blah	If searchType is 'IP': searchValue must be a valid IP address or IP Block. if searchType"is 'mac': searchValue must be a full or partial mac address. Not required if searchType is 'name', but is recommended to filter results.

Optional Parameters

Name	Type	Example	Description
selectOffset	INTEGER	10	Start number for the first result to show on the page to set pagination. For example, a select
selectCount	INTEGER	5	The number of results to return on the page, if return results exceed the number of selectCo
linkedTo	INTEGER	1422	The ID of the resource's DHCP link(s)

Data Attributes

_dhcp_attributes

Description	The _dhcp_attributes data attribute holds the specific settings used to generate a DHCP configuration file, place it on a server via SCP, and restart that server via a SSH session.
Example:	{"type":"ISC", "notes":"notes here", "username":"username", "port":"22", "config_test":"/etc/init.d/dhcpd configtest", "server_stop":"/etc/init.d/dhcpd stop", "server_start":"/etc/init.d/dhcpd start", "config_path":"/tmp/dhcpd.conf", "option_routers":"", "option_domain_name_servers":"", "option_domain_name":"", "authoritative":"1", "default_lease_time":"600", "max_lease_time":"7200", "local_port":"67", "log_facility":"local7", "password":"", "server_ip":"10.0.0.0", "freeLines":0}

Data Description

Name	Type	Description
type	STRING	The type of DHCP server being administered. Currently only 'ISC' is supported.
notes	STRING	Notes associated with this DHCP server
server_ip	STRING	The IP address of the DHCP server
username	STRING	The SSH username employed when transferring the DHCP configuration file to the server.
password	STRING	The SSH password employed when transferring the DHCP configuration file to the server.
port	INTEGER	The SSH port employed when transferring the DHCP configuration file to the server.
config_test	STRING	The command to test if a configuration file parses correctly. ex: /etc/init.d/dhcpd configtest
server_stop	STRING	The command to stop the DHCP server. ex: /etc/init.d/dhcpd stop
server_start	STRING	The command to start the DHCP server. ex: /etc/init.d/dhcpd start
config_path	STRING	Where to place the configuration file on the server.
authoritative	BOOL	Whether or not this DHCP server is authoritative.
default_lease_time	INTEGER	The default lease time for IPs distributed by this DHCP server.
max_lease_time	INTEGER	The max lease time for IPs distributed by this DHCP server.
local_port	INTEGER	The port on which this DHCP server listens
option_routers	STRING	The information which populates the "routers" option in the DHCP configuration
option_domain_name_servers	STRING	The information which populates the "domain_name_servers" option in the DHCP configuration
option_domain_name	STRING	The information which populates the "domain_name" option in the DHCP configuration
log_facility	STRING	The log facility to which this DHCP Server sends its logging information
freeLines	INTEGER	As this system cannot hope to support all the thousands of different DHCP configurations, ProVision's DHCPv2 system includes a mechanism for adding "free lines" to the end of certain DHCP config sections so that administrators can customize their DHCP config file to their needs. The "freeLines" field indicates how many of these lines exist to be inserted after the general server definition section but before the subnets and hosts are enumerated.
freeLine#	STRING	Free line data to be inserted after the general server definition section but before the subnets and hosts are enumerated. There can be multiple instances of this attribute, numbered appropriately. ex: "freeLine1", "freeLine2", "freeLine3", etc. The number of freeLine# entries must match the number in the "freeLines" attribute.

_dhcp_pool_attributes

Description	A JSON-encoded string containing all the specific configuration parameters which govern this DHCP Pool.
Example:	{"mac":"ab:cc:de:ff:aa:bc","rangeStart":"13.0.0.0","rangeEnd":"13.0.0.255","freeLines":1,"freeLines1":"free line"}

Data Description

Name	Type	Description
mac	STRING	Only used when setting up a DHCP Host-type Pool. Holds the MAC address of the system to which the IP will be associated.
rangeStart	STRING	Only used when setting up a DHCP Subnet-type Pool. Holds the beginning of the Subnet range being allocated.
rangeEnd	STRING	Only used when setting up a DHCP Subnet-type Pool. Holds the end of the Subnet range being allocated.
freeLines	INTEGER	As this system cannot hope to support all the thousands of different DHCP configurations, ProVision's DHCPv2 system includes a mechanism for adding "free lines" to the end of certain DHCP config sections so that administrators can customize their DHCP config file to their needs. The "freeLines" field indicates how many of these lines exist to be inserted within the DHCP Pool declaration.
freeLine#	STRING	Free line data to be inserted after the general server definition section but before the subnets and hosts are enumerated. There can be multiple instances of this attribute, numbered appropriately. ex: "freeLine1", "freeLine2", "freeLine3", etc. The number of freeLine# entries must match the number in the "freeLines" attribute.

API Module - DNS

DNS Server Control

get	
URL	/api/v1/api.php?target=dnsServer&action=get
Description	If provided with an id, fetches that DNS Server from the database. If not, fetches a list of all stored DNS Servers

Returns

Examples:

SUCCESSFUL:	{"success":1,"message":"Fetch Sucessful.","data":[{"id":"10","server":"mrbomasm-dns-4.onnet.net", "username": "SCP","remote_directory":"zones", "named_conf_path":"\Vetc\Vzones","active":"1","post_command":null,"pre_com "powerdns_backend":"Bind","db_username":null,"db_password":"(","db_port":null,"db_name":null, "server_type" ,\"SOA\":\"\",\"remote_directory\":\"zones\", \"named_conf_path\":\"\\\Vetc\\\Vzones\", \"dyn_DNSSEC_contact\" :\" \\"db_username\":\"\",\"db_password\":\"(\",\"db_port\":\"\",\"db_name\":\"\",\"enable_views\":\"1\"}", "testID":"963", \"server_id\":\"10\",\"name\":\"_6connectDefault\",\"extras\":\"[]\",\"description\":\"\",\"timestamp\":\"1371789181\"}]}
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	The options entry is a JSON-encoded string containing a variety of server-specific configuration c This string will vary widely by server type and configuration. The following are a selection of comr
transfer_type	STRING	Protocol used for transfer of DNS zones and records. Valid settings include SCP, PowerDNS, Se
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 attache
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.
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

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

Example URL	/api/v1/api.php?target=dnsServer&action=get&id=15

add

URL	/api/v1/api.php?target=dnsServer&action=add
Description	Adds a new DNS Server
Returns	**Examples:**

SUCCESSFUL:	{"success":1,"message":"Add Successful."}
ERROR:	{"success":0, "message":"error message"}

Required Parameters

Name	Type	Example	Description
server	STRING	dns.yourdomain.com	IP or 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	**Examples:** SUCCESSFUL: {"success":1,"message":"Delete Successful."} ERROR: {"success":0, "message":"error message"}
Required Parameters	

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	**Examples:** 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	IP or 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
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&server=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	**Examples:**

SUCCESSFUL:	{"success":1,"message":"Transfer Successful."}
ERROR:	{"success":0, "message":"error message"}

Required Parameters

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	Transfers a single Zone file to all its associated DNS Servers, along with updated server configurations. Performs pre and post commands on the target servers, transfers the zone file(s), and restarts services.
Returns	**Examples:** SUCCESSFUL: {"success":1,"message":"Updated Zone: $name.zone on $server via SCP"} ERROR: {"success":0, "message":"error message"}
Required Parameters	Name: zoneId; Type: INTEGER; Example: 35; Description: 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	Accepts search criteria to retrieve a list of all matching DNS Zones and associated Records. Search can be performed on any combination of Zone and Record attributes.

Returns	**Examples:**

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

Data Detail:

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	BOOl	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 bee

Required Parameters	None

Optional Parameters

Name	Type	Example	Description
likeFlag	BOOL	1	When 1, string searches are done via LIKE with wildcards at b strict comparison is used.
generalFlag	BOOL	1	When 1, searches over the provided parameters using OR. If
selectCount	INTEGER	30	When supplied only returns the first X entries
selectOffset	INTEGER	10	When supplied, only returns entries after record X
sortArray	JSON	{"zoneName":"desc","zoneMask":"asc"}	A JSON-encoded object containing a list of columns to sort on which to sort. Any API variable may be used for sorting. Valid 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 c

Returns	**Examples:**

SUCCESSFUL:	{"success":1,"message":"Search Successful.","data":[{"zoneId":"123","zoneName":"foobs.net","zoneResourceId":"483","zoneIpver":null,"zoneMas
ERROR:	{"success":0, "message":"error message"}

Data Detail:

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	BOOl	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 bee

Required Parameters	None

Optional Parameters

Name	Type	Example	Description
likeFlag	BOOL	1	When 1, string searches are done via LIKE with wildcards at b
generalFlag	BOOL	1	When 1, searches over the provided paramenters using OR. I
selectCount	INTEGER	30	When supplied only returns the first X entries
selectOffset	INTEGER	10	When supplied, only returns entries after record X
sortArray	JSON	{"zoneName":"desc","zoneMask":"asc"}	A JSON-encoded object containing a list of columns to sort on

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	**Examples:**

SUCCESSFUL:	{"success":1,"message":"Update Successful."}
ERROR:	{"success":0, "message":"error message"}

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.

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.
updateZoneResourceId	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	INTEGER	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.

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

add

URL	/api/v1/api.php?target=zone&action=add
Description	Adds a new DNS Zone.
Returns	**Examples:**

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

Data Detail:

Name	Type	Description
data	INTEGER	The Id of the new Zone entry.

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	**Examples:**

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	

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	INTEGER	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	**Examples:**

SUCCESSFUL:	{"success":1,"message":"Search Successful.","data":[{"recordType":"A"},{"recordType":"AAAA"},{"recordType":"MX"},{"recordType":"CNAME"},{"r
ERROR:	{"success":0, "message":"error message"}

Data Detail:

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	

Name	Type	Example	Description
zoneId	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&zoneId=50

checkZone

URL	/api/v1/api.php?target=zone&action=checkZone
Description	Runs a zone file through Named checkzone
Returns	**Examples:**

SUCCESSFUL:	{"success":1,"message":"No errors found."}
ERROR:	{"success":0,"message":"21: ignoring out-of-zone data (veggie.com) 22: ignoring out-of-zone data (veggie.com) dns_rdata_fromtext: 23: near '2001:db8:': bad IPv6 address dns_rdata_fromtext: 24: near '1.2.3.': bad dotted quad dns_rdata_fromtext: 25: near '2001::db8::\/32': bad IPv6 address "}

Required Parameters

Name	Type	Example	Description
zoneId	INTEGER	50	The Id of the zone to check.

Optional Parameters	None
Example URL	/api/v1/api.php?target=zone&action=checkZone&zoneId=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	**Examples:**

SUCCESSFUL:	{"success":1,"message":"Search Successful.","data":[{"zoneArchiveId" :"2768","zoneId":"1227", "zoneArchiveTimestamp":"1375298692", "zoneArchiveFingerprint":"d060e59d69606326d80b2e55b50f0bc9","zoneName":"6connect.com","zoneIpver":nu "zoneMask":null,"zoneSerial": "2013073105","zoneRefresh":"14400","zoneRetry":"2000","zoneExpire":"604800", "zoneMinimum":"3600", "zoneSOA":null,"zoneTags":null,"zoneTTL":"3600","zoneEnableDNSSEC":"1", "zoneResourceId":"1013","zonePreviousViewLinkage":"[]"}]}
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	

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

get

URL	/api/v1/api.php?target=record&action=get
Description	Accepts search criteria to retrieve a list of all matching DNS Records. Search can be performed on any combination of Zone and Re
Returns	**Examples:**

SUCCESSFUL:	{"success":1,"message":"Search Successful.","data":[{"recordId":"30894","recordZoneId":"229","recordHost":"@","recordType":"NS","recordValue
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 comp
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	**Examples:**

SUCCESSFUL:	{"success":1,"message":"Update Successful."}
ERROR:	{"success":0, "message":"error message"}

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.

Name	Type	Example	Description
searchZoneId	INTEGER	123	The Zone ID to search for.
searchZoneName	STRING	foo	The Zone Name to search for.
searchZoneCustId	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	INTEGER	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.
updateZoneResourceId	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	**Examples:**

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

Data Detail:

Name	Type	Description
data	INTEGER	The ID of the new Record entry.

Required Parameters	

Name	Type	Example	Description
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

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.
newRecordDescription	STRING	Description.	Notes for the Record.
newRecordTTL	INTEGER	3600	Record TTL.

Example URL	/api/v1/api.php?target=record&action=add&newRecordZoneId=123&newRecordHost=@host&newRecordType=PTR&newRecordTTL=3600

delete

URL	/api/v1/api.php?target=record&action=delete
Description	Performs a search over the Zones and Records dataset and deletes all found Records, but leaves their parent Zones intact.
Returns	**Examples:**

SUCCESSFUL:	{"success":1,"message":"Deletion Successful."}
ERROR:	{"success":0, "message":"error message"}

Required Parameters	None

Optional Parameters	

Name	Type	Example	Description
deleteZoneId	INTEGER	123	The Zone ID to search for.
deleteZoneName	STRING	foo	The Zone Name to search for.
deleteZoneCustId	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	INTEGER	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	**Examples:**

SUCCESSFUL:	{"success":1,"message":"Record Moved."}
ERROR:	{"success":0, "message":"error message"}

Required Parameters

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

get

URL	/api/v1/api.php?target=zoneLinkage&action=get
Description	Searches for Server-Zone Linkages. If no search parameters are supplied, all linkages are returned.
Returns	**Examples:**

SUCCESSFUL:	{"success":1,"message":"2 rows retrieved.","data":[{"id":"285","zoneId":"64","serverId":"1","serverName":"173.164.182.169","serverType":"SCP","
ERROR:	{"success":0, "message":"error message"}

Data Detail:

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

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	**Examples:**

SUCCESSFUL:	{"success":1,"message":"Link Added."}
ERROR:	{"success":0, "message":"error message"}

Required Parameters	

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	**Examples:**

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	**Examples:**

SUCCESSFUL:	*{"success":1,"message":"Fetch Sucessful.","data":[{"id":"1","nameserver":"ns1.dns.6connect.net","add_to_zones_default":"1","ordering":"10","us*
ERROR:	*{"success":0, "message":"error message"}*

Data Detail:

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	

Name	Type	Example	Description
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	**Examples:**

SUCCESSFUL:	*{"success":1,"message":"Add Successful."}*
ERROR:	*{"success":0, "message":"error message"}*

Required Parameters

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	**Examples:** SUCCESSFUL: *{"success":1,"message":"Server Deleted."}* ERROR: *{"success":0, "message":"error message"}*
Required Parameters	**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	**Examples:** SUCCESSFUL: *{"success":1, "message":"Success."}* ERROR: *{"success":0, "message":"error message"}*
Required Parameters	**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	**Examples:** SUCCESSFUL: *{"success":1,"message":"Reordering Successful."}* ERROR: *{"success":0, "message":"error message"}*
Required Parameters	**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	**Examples:** SUCCESSFUL: *{"success":1,"message":"Reordering Successful."}* ERROR: *{"success":0, "message":"error message"}*
Required Parameters	**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	
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": "192.168.0.0\/24", "formatted_ip": "192.168.0.0\/24", "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
allowSubAssignments	BOOL	true	Filters blocks based on wether 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"
lastUpdateTime	DATETIME	=2015-8-19 21:08:54	SQL Datetime format, prefaced by an "=" for exact time updated, "=>" for blocks updated after the given time, or "=<" for blocks updated before the given time.
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`	(**Deprecated**: Use resourceQuery instead) A custom ID which can be used to link resources in the 6Connect database back to your organization.
resourceId	INTEGER	1234	The ID of the resource the block is assigned to
resourceQuery	JSON	`{"parent_id":15}`	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.
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. Accepable 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	**Examples:**

SUCCESSFUL	*{"success":1,"message":"Block 192.168.0.0/24 (12345) added", "id":12345, "data":{ "id":12345, "cidr":192.168.0.0/24", ...} }*
ERROR	*{ "success":0, "message":"error message" }*

Required Parameters

Name	Type	Example	Description
block	STRING	213.37.29.0/24	CIDR block description
rir	STRING	ARIN	Acceptable values: ARIN, RIPE, APNIC, AfriNIC, LACNIC, 1918

Optional Parameters

Name	Type	Example	Description
allowDuplicate	BOOL	true	Allow the creation of duplicate blocks. The default behavior is to reject duplicates.
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	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
resourceId	INTEGER	1234	The ID of the resource the block is assigned to
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	**Examples:**

SUCCESSFUL	SINGLE BLOCK	*{"success":1,"message":"Block 192.168.0.0/24 (12345) updated", "data":{ "id":12345, "cidr":192.168.0.0/24", ...} }*
SUCCESSFUL	MULTIPLE BLOCKS	*{"success":1,"message":"3 blocks updated", "data":[{ "id":12345, "cidr":192.168.0.0/24", ...}, {"id":12346, "cidr", "192.168.0.1/32", ...}] }*
ERROR		*{ "success":0, "message":"error message" }*

Required Parameters

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.
*Either block or id can be used, but only one must be provided				

Optional Parameters

Name	Type	Example	Description
allowSubAssignments	BOOL	true	Does the block allow sub-assignments? If the block is assigned and allowSubAssignments is "true", children split from this block will be able to be assigned to different resources. Acceptable values: "true" or "false"
asn	INTEGER	1000	ASN for the block
code	STRING	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. ***Available in version 5.1.0***
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	Examples

SUCCESSFUL	*{"success":1,"message":"Aggregate deleted: 192.168.0.0/24", "data":{ "id":12345, "cidr":192.168.0.0/24", ...} }*
ERROR	*{ "success":0, "message":"error message" }*

Required Parameters

Name	Type	Example	Description
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

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=addTag
Description	Adds a tag to an IP block.
Returns	**Examples:**

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	**Examples:**

SUCCESSFUL	*{ "success":1,"message":"Tag Removed.", "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 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, mask size, etc.) and assigns it to a Resource Holder.
Returns	**Examples:**

SUCCESSFUL	*{ "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", ...} }*
ERROR	*{ "success":0, "message":"error message" }*

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	`{"custom_id":"cust-001"}`	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`	(**Deprecated**: 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&action=smartAssign&mask=24&type=IPv4&resourceId=250&rir=ARIN

Direct Assign

URL	/api/v1/api.php?target=ipam&action=directAssign
Description	Assigns a block to an Resource Holder
Returns	**Examples:**

SUCCESSFUL	SINGLE BLOCK	*{ "success":1,"message":"Assigned 192.168.0.0/24 to Resource (1234)", "id":12345, "data":{ "id":12345, "cidr":192.168.0.0/24", ...} }*
SUCCESSFUL	MULTIPLE BLOCKS	*{ "success":1,"message":"Assigned 5 blocks to Resource (1234) via Direct Assign", "data":{ "ids":[12345, 12346, 12347, ...] } }*
ERROR		*{ "success":0, "message":"error message" }*

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`	(**Deprecated**: 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	`{"custom_id":"cust-001"}`	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&resourceId=1234

Unassign

URL

/api/v1/api.php?target=ipam&action=unassign

Description

Reclaims the specified block to be reassigned in the future

Returns

Examples:

SUCCESSFUL	*{ "success":1,"message":"192.168.0.0/24 unassigned", "id":12345, "data":{ "id":12345, "cidr":192.168.0.0/24", ...} }*
ERROR	*{ "success":0, "message":"error message" }*

Required Parameters	

Name	Type	Example	Description
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

Name	Type	Example	Description
skipHolding	BOOL	true	If set to true (skipHolding=true) then the holding tank is skipped. If set to false, or not included, normal holding tank rules apply. Acceptable values: "true" or "false"

Example URL: /api/v1/api.php?target=ipam&action=unassign&block=213.37.29.0/24

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	**Examples:**

SUCCESSFUL	*{"success":1,"message":"Tags Retrieved.","data":[{"value":"IT","name":"IT"},{"value":"LTE","name":"LTE Mobile"},{"value":"PTP","name":"Point to Point"},{"value":"Prod","name":"Production"},{"value":"VM","name":"Virtual Machines"},{"value":"VOIP","name":"VOIP"},{"value":"ANY","name":"ANY"}]}*
ERROR	*{'success':0, 'message':'error message'}*

Add Tag To List

URL	/api/v1/api.php?target=ipam&action=addTagToList
Description	Adds a tag to the IPAM tag list
Returns	**Examples:**

SUCCESSFUL	*{"success":1,"message":"Tag Added."}*
ERROR	*{'success':0, 'message':'error message'}*

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 C

Get Regions List

URL	/api/v1/api.php?target=ipam&action=getRegionList
Description	Returns a list of all valid Regions in the database.

Returns	**Examples:**

SUCCESSFUL	*{"success":1,"message":"Regions Retrieved.","data":[{"value":"ANY","name":"Any Region"},{"value":"ASH1","name":"Ashburn, VA"},{"value":"BOS","name":"Boston, MA"},{"value":"CHI","name":"Chicago, IL"},{"value":"DAL","name":"Dallas, TX"},{"value":"DEN","name":"Denver, CO"},{"value":"FRKT","name":"Frankfurt, DE"},{"value":"LON1","name":"London, UK"},{"value":"MIA","name":"Mia FL"},{"value":"PAR","name":"Paris, FR"},{"value":"SFO","name":"San Francisco, CA"},{"value":"SEA","name":"Se WA"},{"value":"Tokyo","name":"Tokyo"},{"value":"Singapore","name":"Singapore"},{"value":"Jakarta","name":"Jak*
ERROR	*{'success':0, 'message':'error message'}*

Add Region To List

URL	/api/v1/api.php?target=ipam&action=addRegionToList
Description	Adds a region to the IPAM region list
Returns	**Examples:**

SUCCESSFUL	*{"success":1,"message":"Region Added."}*
ERROR	*{'success':0, 'message':'error message'}*

Required Parameters

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

Get Utilization

URL	/api/v1/api.php?target=ipam&action=utilization
Description	Gets the utilization percentages for a specific ip block or ip block and mask combination.

Returns	**Examples:**

SUCCESSFUL	{
	"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"
	}

ERROR	*{'success':0, 'message':'error message'}*

Required Parameters

Name	Type	Example	Description
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	

Name	Type	Example	Description
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

Get Host Utilization

URL	/api/v1/api.php?target=ipam&action=getHostUtilization
Description	Gets the host utilization statistics with support for filters.
Returns	**Examples:**

SUCCESSFUL	{
	"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"
	}]
	}

ERROR	*{'success':0, 'message':'error message'}*

Required Parameters	

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

NOTE: to filter using multiple values, pass the values as a JSON-encoded string representation of an array.

For example, to get utilization data for multiple tags, you could use the following URL:

/api/v1/api.php?target=ipam&action=getHostUtilization&type=ipv4&tags=["Customer","PTP"]

Example URL	/api/v1/api.php?target=ipam&action=getHostUtilization&type=ipv4&tags=["Customer","PTP"]&region=SMF

Aggregate

URL	/api/v1/api.php?target=ipam&action=aggregate
Description	Aggregates a selected block to the mask specified. If no mask specified, re-aggregates blocks to next parent. IE. calling aggregate on a /25 will aggregate both children back to the parent /24. All child blocks must be Available for aggregation to succeed.

Returns

Examples:

SUCCESSFUL	`{"success":1,"message":"10.2.0.128\/25 aggregated into 10.2.0.0\/24","id":16326}`
ERROR	*{'success':0, 'message':'error message'}*

Required Parameters

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

Name	Type	Example	Description
autoAggregateToMask	INTEGER	24	All blocks and IPs smaller than this netmask will be aggregated.
ignoreAssignments	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	/api/v1/api.php?target=ipam&action=split
Description	Splits a selected block to the mask specified. If no mask specified, it split blocks to next child. IE. calling aggregate on a /24 will split both parent to the child /25s. All parent blocks must be Available, or have Allow Sub Assignments on for a split to succeed.
Returns	**Examples:**

SUCCESSFUL	`{"success":1,"message":"10.1.0.0\/24 split into 10.1.0.0\/25 and 10.1.0.128\/25","data":{"child1":23441 ,"child2":23451}}`
ERROR	*{'success':0, 'message':'error message'}*

Required Parameters

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

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	**Examples:**

SUCCESSFUL	`{"success":1,"message":"Ping scan started for 8.8.8.0\/27"}`
ERROR	*{'success':0, 'message':'error message'}*

Required Parameters

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	**Examples:**

SUCCESSFUL	`{"success":1,"data":{"block":"8.8.8.0\/27","date":"07\/14\/2014 11:07:10",` `"data":[{"address":"8.8.8.8","host":"google-public-dns-a.google.com","status":"Up"}`
ERROR	*{'success':0, 'message':'error message'}*

Required Parameters

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	**Examples:**

SUCCESSFUL	{"success":1,"message":"Options for 14.0.0.0\/25 (125)","options":{"actions":["aggregate"],"templates":[{"name":"Auto Split","masks":[26,27,28,29,30,31,32]}]}}
ERROR	*{'success':0, 'message':'error message'}*

Required Parameters

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 VLAN

URL	/api/v1/api.php?target=ipam&action=getVlan
Description	Returns the VLAN for the block
Returns	**Examples:**

SUCCESSFUL	{"success":1,"message":"Found VLAN 1002 (14.0.0.0\/25)","data":{"id":125,"type":"ipv4","top_aggregate":81532,"cidr":"14.0.0.0\/25","formatted_ip":"14.0.0.0\/ Labz","last_updated_time":"2015-01-22 12:30:37","description":null,"parent":81532,"rir":"ARIN","lir_id":"1062","nc 10:30:31","asn":"143","allowSubAssignments":true,"permissions":{"permissionIPAMRead":"1","permissionIPAMU
ERROR	*{'success':0, 'message':'error message'}*

Required Parameters	

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	**Examples:**

SUCCESSFUL	{"success":1,"message":"1 IPv4 and 0 IPv6 blocks would be moved to the available pool. ","data":[{"id":77712,"type":"ipv4","top_aggregate":77552,"cidr":"23.92.0.64\/26","formatted_ip":"23.92.0.64\/26",": holding","last_updated_time":"2014-10-29 11:25:41","description":null,"parent":77682,"rir":"ARIN","lir_id":"451","r 11:20:34","asn":null,"allowSubAssignments":false,"permissions":{"permissionIPAMRead":"1","permissionIPAMUp
ERROR	*{'success':0, 'message':'error message'}*

Required Parameters	None
Optional Parameters	

Name	Type	Example	Description
preview	BOOL	true	Acceptable values: "true" or "false" If set to "true", returns a list of blocks that would be removed from the holding tank, but does not con If set to "false", processes the holding tank and returns a list of blocks returned to available status.

Example URL	/api/v1/api.php?target=ipam&action=processHoldingTank&preview=true

IPAM SWIP Calls:

Deassign

URL	/api/v1/api.php?target=ipam&action=deassign
Description	Performs a SWIP deassignment for the indicated IPAM block.
Returns	**Examples:**

SUCCESSFUL	*{ "success":1,"message":"success message" }*
ERROR	*{ "success":0, "message":"error message" }*

Required Parameters	

Name	Type	Example	Description
blockId	INTEGER	1234	ID of the block to deassign
block	STRING	67.221.241.0/24	the CIDR of the block being de-assigned
resourceId	INTEGER	1234	ID of resource representing the customer to deassign
lirId	INTEGER	1234	The ProVision id of the LIR which is performing the de-assignment
entityHandle	STRING	CONNE-81	The Org ID for the LIR.

Optional Parameters

Name	Type	Example	Description
netName	STRING	6CONN-67-221-241-0-24	Optional name for the network to override the default. The default net name will address for the block.

Example URL

/api/v1/api.php?target=ipam&action=deassign&resourceId=1234&blockId=1234&lirId=1234&entityHandle=CONNE-81&block=67.22

Get RIR List

URL

/api/v1/api.php?target=ipam&action=getRIRList

Description

Returns a list of all valid RIRs in the database.

Returns

Examples:

SUCCESSFUL	*{"success":1,"message":"RIRs Retrieved.","data":[{"value":"ARIN","name":"ARIN"},{"value":"1918","name":"1918"},{"value":"AfriNIC","name":"Afi*
ERROR	*{'success':0, 'message':'error message'}*

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 cu

Returns

Examples:

SUCCESSFUL	*{ "success":1,"message":"Sent ARIN SWIP with action simpleReassign for 67.221.244.0/28 for Acme, Message:*
ERROR	*{ "success":0, "message":"error message" }*

Required Parameters

Name	Type	Example	Description
blockId	INTEGER	1234	ID of the block to reassign
resourceId	INTEGER	1234	ID of resource representing the customer to reassign to
lirId	INTEGER	1234	The ProVision id of the LIR to use for reassignment
entityHandle	STRING	CONNE-81	The Org ID for the LIR.

Optional Parameters	

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 Name Prefix and IP address for the block.

Example URL	/api/v1/api.php?target=ipam&action=simpleReassign&resourceId=121&blockId=31559&lirId=95&entityHandle=CONNE-81&&netNa

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	**Examples:**

SUCCESSFUL	{"asns":[],"masks":["24"],"rirs":["1918"],"lirs":[],"tags":["DHCP"],"codes":[],"vlans":[],"regions":["Quito"],"resources": 1","slug":"quito-lab-1","type":"dhcp_pool","parent_id":"1","category_id":null,"attr":{"_dhcp_type":"subnet","_dhcp_
ERROR	*{'success':0, 'message':'error message'}*

Required Parameters	

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	
URL	/api/v1/api.php?target=lir&action=get
Description	Returns a list of LIRs
Returns	**Examples:**

SUCCESSFUL	{
	"success": 1,
	"message": "2 objects found",
	"data": [
	{
	"id": "100",
	"name": "RIPE Test LIR",
	"slug": "ripe-test-lir",
	"entities": [
	{
	"mnt_by": "mntner@email.com"
	"mnt_by_password": "password",
	"admin_c": "test-admin-c",
	"tech_c": "test-tech-c",
	"api_key": null
	}
	],
	"rir": "RIPE"
	},
	{
	"id": "101",
	"name": "ARIN Test LIR",
	"slug": "arin-test-lir",
	"entities": [
	{
	"org_handle": "TEST-10",
	"admin_poc": "TEST-ARIN",
	"net_poc": "TEST-ARIN",
	"abuse_poc": "",
	"net_name_prefix": "PRFX",
	"api_key": "API-XXXX-YYYY-ZZZZ-1234"
	}
	],
	"rir": "ARIN",
	"asn": "1000"
	}
	]
	}

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

Example URL	/api/v1/api.php?target=lir&action=get

Delete	
URL	/api/v1/api.php?target=lir&action=delete&id=<ID>
Description	Deletes an LIR

Returns	**Examples:**
	
	SUCCESSFUL
	`{ "success": 1, "message": "LIR deleted." }`
	
	ERROR
	`{ "success":0, "message":"error message" }`

Example URL	/api/v1/api.php?target=lir&action=delete&id=100

API Module - Peering

Peering

getCommunications	
Base URL	/api/v1/api.php?target=peering&action=getCommunications
Description	Returns all communication data on peers at a particular exchange.
Returns	**Examples:** SUCCESSFUL: {"success":1,"message":"8 records found.","data":[{"name":"1&1 Internet","asn":"8560","request_status":null,"qua Networks","asn":"5580","request_status":null,"qualified":null,"is_peer":false,"id":"2283","log_data":[]},{"name":"OV Communications","asn":"7029","request_status":null,"qualified":null,"is_peer":false,"id":"1820","log_data":[]},{"nar Inc.)","asn":"6461","request_status":null,"qualified":null,"is_peer":false,"id":"541","log_data":[]},{"name":"tw teleco ERROR: *{'success':0, 'message':'error message'}*
Required Parameters	**Name** / **Type** / **Example** / **Description** public_id / INTEGER / 1 / The unique numerical identifier of the exchange to retrieve peering communicaiton 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	**Examples:** SUCCESSFUL: {"success":1,"message":"184 peers found.","data":[{"id":"262","public_id":"1","asn":"8560","name":"1&1 Internet","qu House","qualified":true,"is_peer":0,"request_status":null,"info_prefixes":"100","public_ips":[],"contacts":[],"log_data":[]},{"id":"890","pu Hosting","qualified":true,"is_peer":0,"request_status":null,"info_prefixes":"3000","public_ips":[],"contacts":[],"log_data":[]},{"id":"1676" Communications, Inc.","qualified":true,"is_peer":0,"request_status":null,"info_prefixes":"200","public_ips":[],"contacts":[],"log_data":[]},{"id":"576","publi Communications","qualified":true,"is_peer":0,"request_status":null,"info_prefixes":"3000","public_ips":[],"contacts":[],"log_data":[]},{"i LLC","qualified":true,"is_peer":0,"request_status":null,"info_prefixes":"60","public_ips":[],"contacts":[],"log_data":[]},{"id":"4078","publi Communications Inc.)","qualified":true,"is_peer":0,"request_status":null,"info_prefixes":"20000","public_ips":[],"contacts":[],"log_data ERROR: {"success":1,"message":"No peers found."}
Required Parameters	None

Optional Parameters	

Name	Type	Example	Description
public_id	INTEGER	1	The unique numerical identifier of the exchange to re
id	INT	1	The unique numerical identifier of the peer in peering
asn	INT	4436	
name	STRING	GTT	
aka	STRING	nLayer	
website	STRING	http://www.gt-t.net	
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','100
info_ratio	ENUM	Mostly Outbound	enum('Not Disclosed','Heavy Outbound','Mostly Outb
info_scope	ENUM	Global	enum('Not Disclosed','Regional','North America','Asia
info_type	ENUM	NSP	enum('Not Disclosed','NSP','Content','Cable/DSL/ISP
info_prefixes	INT	10000	
info_lookingglass	STRING	http://lg.nlayer.net/	
info_routeserver	STRING	telnet://route-server.nlayer.net	
info_unicast	CHAR	1	
info_multicast	CHAR		
info_ipv6	CHAR	1	
policy_url	STRING	http://www.gt-t.net/Peering_policies.aspx	
policy_general	ENUM	Selective	enum('Open','Selective','Restrictive','No') DEFAULT
policy_locations	ENUM	Required - International	enum('Not Required','Preferred','Required - US','Req
policy_ratio	ENUM	No	enum('Yes','No') DEFAULT NULL
policy_contracts	ENUM	Not Required	enum('Not Required','Private Only','Required') DEFA
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

Example URL	/api/v1/api.php?target=peering&action=getPeers&public_id=1

getRequests

URL	/api/v1/api.php?target=peering&action=getRequests
Description	Returns a list of all peering requests issued

Returns	**Examples:** SUCCESSFUL: {"success":1,"message":"1 request found.","data":[{"id":"131","public_id":"5","source_participant_id":"2335","source_asn":"8038","peer_participant_id":"1418","peer_asr ops@6connect.com","email_to":"nalinmk@gmail.com","subject":"Peering request from 6connect, Inc.","body":"Peering,\r\n\r\n6conr locations.\r\n\r\nFacility, IP Address\r\nEquinix Ashburn - 206.126.236.68\r\nEquinix Palo Alto - 198.32.176.36\r\nEquinix Ashburn - LAN - 195.66.225.175\r\n\r\nSincerely,\r\nOperations\r\nops@6connect.com\r\n\r\n\r\n6connect, Inc. information:\r\nEquinix Palo Al http:\/\/as8038.peeringdb.com\/\r\n","status":null,"created":"2014-04-23 10:31:33","modified":"2014-04-23 10:31:33"}]} ERROR: {"success":1,"message":"No request found.","data":[]}
Required Parameters	None
Optional Parameters	

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	**Examples:** SUCCESSFUL: {"success":1,"message":"1 sessions found.","data":[{"id":"51","source_asn":"32787","source_ipaddr":"1.2.3.4","resource_id":"422","resource_name":"ar1.inoc.com","peer Technologies","peer_participant_id":"2","peer_ipaddr":"206.126.236.102","peer_hostname":null,"peer_group":"group b","public_id":" Ashburn","ip_type":"ipv4","type":"Peer","state":"not configured","prfx_max":"20","prfx_received":null,"password":"0","note":null},]} ERROR: {"success":1,"message":"No peers found."}
Required Parameters	

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&action=getPeers&public_id=1

addSession

URL	/api/v1/api.php?target=peering&action=addSession
Description	Adds a bgp session
Returns	**Examples:** SUCCESSFUL: {"success":1,"message":"Session added: Amazon.com (AS8038\/1.2.3.5) - (AS16509\/206.126.236.68)","data":{"id":111,"source_asn":"8038","source_ipaddr":"1.2.3.5","resource_id":"422","resource_name":n configured","prfx_max":"200","prfx_received":null,"password":"ace12345","note":"I'm a fancy note."}} ERROR: {"success":1,"message":"No request found.","data":[]}
Required Parameters	None

Optional Parameters	

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=1

configureSession

URL	/api/v1/api.php?target=peering&action=configureSession
Description	Configure a BGP session on the router
Returns	**Examples:** SUCCESSFUL: ERROR: {"success":0,"message":"Unable to authenticate "}
Required Parameters	

Name	Type	Example	Description
session_id	INTEGER	1	

Optional Parameters	

Name	Type	Example	Description
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=51&username=&config=conf+t%0A%0Arouter+bgp+32787%

deleteSession	
URL	/api/v1/api.php?target=peering&action=deleteSession
Description	Delete sessions matching criteria
Returns	**Examples:** 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&action=deleteSession&id=171

updateSession

URL	/api/v1/api.php?target=peering&action=updateSession
Description	Updates session values with any new values specified
Returns	**Examples:** SUCCESSFUL:{"success":1,"message":"Session updated: 123.net (AS32787\/1.2.3.4) - (AS12129\/206.126.236.70)","data":{"id":"41","source_asn":"32787","source_ipaddr":"1.2.3.4","resource_id":"422","resource_name" a","public_id":"1","public_name":"Equinix Ashburn","ip_type":"ipv4","type":"Peer","state":"not configured","prfx_max":"10","prfx_recei ERROR:
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&action=updateSession&note=Adding+an+awesome+note.&peer_group=group+a

resestPeerStatus

URL	/api/v1/api.php?target=peering&action=resetPeerStatus
Description	
Returns	**Examples:** SUCCESSFUL: {"success":1,"message":"1&1 Internet status reset","data":{"id":"262","public_id":"1","asn":"8560","name":"1&1 Internet","qualified":true,"is_peer":0,"request_status":"none","info_prefixes":null,"public_ips":[],"contacts":[],"log_data":[{"message":"I status reset","time":"2014-05-22 23:14:54","request_id":null,"session_id":null,"public_id":"1"},{"message":"Peer status reset","time":"2014-05-22 23:14:18","request_id":null,"session_id":null,"public_id":"1"},{"message":"Session deleted: 1&1 Internet (AS32787\/1.2.3.4) - (AS8560\/206.126.236.200)","time":"2014-05-22 22:39:43","request_id":null,"session_id":"71","public_id":"1"},{"message":"Request sent: ","time":"2014-04-12 13:24:43","request_id":"121","session_id":null,"public_id":"1"},{"message":"Session added: 1&1 Internet (AS32787\/1.2.3.4) - (AS8560\/206.126.236.200)","time":"2014-04-07 11:32:37","request_id":null,"session_id":"71","public_id":"1"}]}} ERROR: {"success":0,"message":"Could not find peer matching parameters"}

Required Parameters	

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=262&public_id=1

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
Returns	**Examples:** SUCCESSFUL: {"success":1,"message":"Request sent","data":{"id":"922","public_id":"1","asn":"10933","name":"ATX Communicatio Inc.","qualified":true,"is_peer":0,"request_status":"sent","info_prefixes":null,"public_ips":[],"contacts":[],"log_data":[{"message":"Requ 16:59:01","request_id":"181","session_id":null,"public_id":"1"},{"message":"Request sent to ","time":"2014-05-27 16:49:30","request_ ERROR: {"success":0,"message":"Internal error"}
Required Parameters	

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','accepted','rejected')

Optional Parameters	None
Example URL	https://ops.6connect.com/peering-demo/api/v1/api.php?target=peering&action=sendRequest&peer_participant_id=1909&peer_asn= &public_id=1&type=text&email_from=ops%406connect.com&email_to=operations%40as28929.net&cc=&bcc=&subject=Peering+re &body=%0D%0APeering%2C%0D%0A%0D%0A6connect%2C+Inc.%2C+8038%2C+would+like+to+peer+with+ASDASD+srl+at+o

sendEmail

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	**Examples:** SUCCESSFUL: ERROR:

Required Parameters	

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	**Examples:** SUCCESSFUL: ERROR:
Required Parameters	

Name	Type	Example	Description
participant_id	INTEGER	262	
public_id	INTEGER	1	

Optional Parameters	None
Example URL	

API Module - Resource

Resources

get	
URL	/api/v1/api.php?target=resource&action=get
Description	Get a resource or resources
Returns	**Examples:** SUCCESSFUL: *{"success":1,"message":"Search successful","data":[{"id":"57","name":"2nd Email","slug":"6c-contact-email2","type":"field","parent_id":"1","category_id":null,"attr":[]}]}* ERROR: *{"success":0,"message":"Search failed"}*
Optional Parameters	

Name	Type	Notes/Example
name	STRING	Name of the resource. Example: 6Connect, Inc.
slug	STRING	The unique URL friendly name of the resource. Example: 6connect-inc
type	STRING	Type of resource (eg. *entry*, *field*, *category*)

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	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: • = *(default)* • != • > • >= • < • <= • LIKE • NOT LIKE • IN • NOT IN • BETWEEN • NOT BETWEEN

> 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	`var data = {`
		`"type: "entry",`
		`"attributes": [`
		`{`
		`"attr_key": "_section",`
		`"attr_value": "105",`
		`},`
		`{`
		`"attr_key": "address-mail-state",`
		`"attr_value": "CA",`
		`}`
		`],`
		`"resources_per_page: 10`
		`}`

You can restrict the range of the resources returned.

Name	Type	Notes/Example
resources_per_page	INTEGER	How many resources to return.
offset	INTEGER	How many resources to offset (the initial resource is 0, not 1).
paged	INTEGER	The page to return (starts at 1, not 0). This parameter is provided for convenience and is used to calculate the offset where: offset=(paged-1)*resources_per_page

Example URL	/api/v1/api.php?target=resource&action=get&id=7

add

URL	/api/v1/api.php?target=resource&action=add
Description	Add a resource.
Returns	**Examples:** */api/v1/api.php?target=resource&action=add&meta[name]=apitest&meta[type]=entry&meta[section]=firewall&fields[network-fqdn][]=* SUCCESSFUL: {"success":1,"message":"Resource added","data":{"id":1077,"name":"apitest","slug":"apitest","type":"entry","parent_id":1,"category_id":"NULL","attr":{"_section":"70","ne */api/v1/api.php?target=resource&action=add&meta[name]=apitest&meta[type]=entry&fields[network-fqdn][]=www.example.com* ERROR:{"success":0,"message":"Entries must be assigned to a section"}
Required Parameters	

Name	Type	Notes/Example
meta[name]	STRING	Name of the resource
meta[type]	STRING	Type of resource (entry, section, field, ect)

Optional Parameters

Name	Type	Notes/Example
meta[parent_id]	INTEGER	ID of the parent resource
meta[category_id]	INTEGER	ID of the category

Required Parameters

(meta[type] = entry)

One of the following:

Name	Type	Notes/Example
meta[section_id]	INTEGER	ID of the section that the entry will be assigned to
meta[section]	STRING	Slug of the section that the entry will be assigned to

Optional Parameters

(meta[type] = entry)

Name	Type	Notes/Example

fields[]	ARRAY	Entry field values (for fields that have already been assigned to the section) can be populated when
		
		The format is field[field-slug][field-instance]. If the field instance is left blank, it will simply be the nex
		
		fields[network-fqdn][]=example.com&fields[network-fqdn][]=test.com
		
		would be written in JSON as
		
		`var fields = {`
		
		`"network-fqdn": [`
		
		`"example.com",`
		
		`"test.com"`
		
		`]`
		
		`}`
		
		Please note that the Field Slug might differ from the Field Name! To find the correct slug to use populating. An Edit Field box will pop up which displays the Field's slug.
		
		A field can be added to a section multiple times. The field instance is used to keep track of which fie for it.

meta[custom_id]	STRING	A custom ID for the entry. In the past this has been called the Resource Holder ID or Customer ID. I

Required Parameters

(meta[type] = field)

Name	Type	Notes/Example
meta[field_type]	STRING	Type of field • text • textarea • radios • checkboxes • choicebox

Optional Parameters (meta[type] = field)	

Name	Type	Notes/Example
meta[help_block]	STRING	Fields can have a line of text under them with instructions
meta[options]	ARRAY	Fields of type radios, checkboxes, or choicebox can have multiple options. This could be multiple ra meta[type]=field&meta[name]=Colors&meta[field_type]=choicebox&meta[options][]=Blue&meta[opt Will create a choicebox with dropdown options of Blue and Green.

update

URL	/api/v1/api.php?target=resource&action=update
Description	Update a resource.
Returns	**Examples:** SUCCESSFUL: *{"success":1,"message":"Resource Updated","data":{"id":"1055","name":"87-child-1","slug":"87-child-1","type":"entry","parent_id":"87","category_id":"65","attr":{"_sectio* ERROR: *{"success":0,"message":"No resource found with ID: 1079"}*
Required Parameters	

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	The fields value should be all the fields that are assigned to the section. Giving an empty array as the fields va The format is: fields[position][key] The position value is the position that the field will appear in (0 is first). The position value must always be inclu fields[0][id]=2 fields[0][slug]=asset-serial-number fields[0][help_block]=something fields[0][new]=false ▪ Either the id or the slug is required, not both. ▪ When the "new" parameter is not included, FALSE is assumed If you want to create a new field and assign it to the section, use a format like this: fields[10][name]=TextArea fields[10][field_type]=textarea fields[10][new]=true

delete

URL	/api/v1/api.php?target=resource&action=delete

Description	Delete a resource.
Returns	**Examples:** SUCCESSFUL: *{"success":1,"message":"Resource deleted."}* ERROR: *{"success":0,"message":"No resource found with ID: 57"}*
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>

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>
	A recursive delete will delete all resources, which are permitted to be deleted, from the bottom up.
	Imagine the following hierarchy:
	<pre>A
	  B1 B2
	C11 C12 C21 C22</pre>
	If a recursive delete is performed on A, but C21 is not deletable, the following
	resources would still be deleted: (B1, C11, C12, C22).
	B2 would not be deleted because it depends on C21 and A would not be deleted because it depends on B2.

Example URL	/api/v1/api.php?target=resource&action=delete&id=57

API Module - VLAN

getById

URL	api/v1/api.php?target=vlan&action=getById
Description	get information of a vlan
Returns	Examples

SUCCESSFUL	{"success":1,"message":"1 VLAN found. ","data":[{"id":"11190","vlan":"176","domain_id":"1","resource_id":null,"name":"","domain_name":"DefaultDomain"
ERROR	*{ "success":0, "message":"error message" }*

Required Parameters

Name	Type	Example	Description
id	INTEGER	11190	ID of the VLAN

Optional Parameters	None
Example URL	api/v1/api.php?target=vlan&action=getById&id=11190

getDomains

URL	api/v1/api.php?target=vlan&action=getDomains
Description	get list and info of all domains or a specific domain. When specifying the domain the results will include tag info
Returns	Examples

SUCCESSFUL	{"success":1,"message":"6 Domains found. ","data":[{"id":"1","domain":"DefaultDomain","attr_6c":{"range":"S","restricted":[1,1002,1003,1004,1005]},"vlans":[{ 6connect.com","attr_6c":{"range":"S","restricted":[1,1002,1003,1004,1005]},"vlans":[{"id":"19","vlan":"10","name":" ... {"id":"12755","vlan":"70","name":""},{"id":"12756","vlan":"71","name":""},{"id":"12757","vlan":"72","name":""},{"id":"1
ERROR	*{ "success":0, "message":"error message" }*

Required Parameters	None

Optional Parameters	

Name	Type	Example	Description
id	INTEGER	2	ID of the domain

Example URL	api/v1/api.php?target=vlan&action=getDomains&id=2

enable

URL	api/v1/api.php?target=vlan&action=enable
Description	enable vlans in a domain
Returns	Examples

SUCCESSFUL	{"success":1,"message":"Success. ","data":"3 in request (0 were found already namedVavailable and 3 found as unavailable). 3 VLANs have been updated to unnamed available entries."}
ERROR	*{ "success":0, "message":"error message" }*

Required Parameters

Name	Type	Example	Description
domainId	INTEGER	1	the domain ID in the domain table
vlan	INTEGER or STRING	18,19,20	vlan for this domain that we want enabled

Optional Parameters	None
Example URL	api/v1/api.php?target=vlan&action=enable&domainId=1&vlan=18,19,20

addDomain

URL	api/v1/api.php?target=vlan&action=addDomain
Description	add a domain
Returns	Examples

SUCCESSFUL	{"success":1,"message":"domain 10 added","id":10,"data":[{"id":"10","domain":"sdfasfaf","attr_6c":{"range":"S","restricted":[1,1002,1003,1004,1005]},""
ERROR	*{ "success":0, "message":"error message" }*

Required Parameters

Name	Type	Example	Description
domain	STRING	6connect.com	Name to call new domain
range	STRING	S	'S' or 'E', depending on VLAN range: extended or standard

Optional Parameters	None
Example URL	api/v1/api.php?target=vlan&action=addDomain&domain=6connect.com&range=S

getAvailable

URL	api/v1/api.php?target=vlan&action=getAvailable

Description	get VLANs in a domain that have not been named yet (even names that are '', or just uses their number as a name)
Returns	Examples

SUCCESSFUL	{"success":1,"message":"16 VLANs found. ","data":{"1":{"2":null,"3":null,"4":null,"5":null,"6":null,"7":null,"8":null,"9":null,"12":null,"13":null,"15":null,"16":null,"1
ERROR	*{ "success":0, "message":"error message" }*

Required Parameters

Name	Type	Example	Description
domainId	INTEGER or STRING	'1,2'	ID of the domains to search for

Optional Parameters

Name	Type	Example	Description
tags	INTEGER or STRING	'111,112'	ID of the tags to search for
min	INTEGER	24	VLAN to start search
max	INTEGER	200	VLAN to end search

Example URL: api/v1/api.php?target=vlan&action=getAvailable&domainId=1&tags=111,112&min=2&max=23

get

URL	api/v1/api.php?target=vlan&action=get
Description	Searches enabled VLANs. When not given options, returns all enabled VLANs. Note: an optional parameter, unavailable, is used to
Returns	Examples

SUCCESSFUL	{"success":1,"message":"2 domains found. ","data":{"1":{"id":"1","domain":"DefaultDomain","attr_6c":null,"attr_custom":null,"vlans":[{"id":"4","vlan":"105","nam
ERROR	*{ "success":0, "message":"error message" }*

Required Parameters: None

Optional Parameters

Name	Type	Example	Description
unavailable	STRING	true	Will return available, instead of enabled, VLANs
domainId	INTEGER	1	domain ID
id	INTEGER	21	VLAN ID in the vlan table

Example URL: api/v1/api.php?target=vlan&action=get&id=4&unavailable=true

deleteDomain

URL	api/v1/api.php?target=vlan&action=deleteDomain
Description	Deletes a VLAN domain.

Returns	Examples SUCCESSFUL: *{"success":1,"message":"Domain #7 successfully deleted."}* ERROR: *{ "success":0, "message":"error message" }*
Required Parameters	

Name	Type	Example	Description
domainId	INTEGER	7	domain ID

Optional Parameters	None
Example URL	api/v1/api.php?target=vlan&action=deleteDomain&domainId=7

updateDomain

URL	api/v1/api.php?target=vlan&action=updateDomain
Description	Updates a VLAN domain.
Returns	Examples

SUCCESSFUL	{"success":1,"message":"Domain 2 updated","id":"2","data":[{"id":"2","domain":"44.6connect.com","attr_6c":{"range":"S","restricted":[1,1002,1003,100 .... "1002":null,"1003":null,"1004":null,"1005":null},"type":"domain","util":{"S":{"used":11,"total":1000}}}]}
ERROR	*{ "success":0, "message":"error message" }*

Required Parameters

Name	Type	Example	Description
domain	STRING	6connect.com	Name of VLAN Domain
range	STRING	E	VLAN range. E for Extended or S for Standard

Optional Parameters	None.
Example URL	api/v1/api.php?target=vlan&action=updateDomain&domain=DefaultDomainnn&range=S&id=1

update

URL	api/v1/api.php?target=vlan&action=update
Description	Update properties of a VLAN: name, tags, or both.
Returns	Examples

SUCCESSFUL	*{"success":1,"message":"VLAN (table ID 11735) updated","id":"11735","data":[{"id":"11735","vlan":"50","domain_id":"7","resource_id":null,"name":"","domain_nam lax.com","tags":[],"tagsString":[]}]}*
ERROR	*{ "success":0, "message":"error message" }*

Required Parameters

Name	Type	Example	Description
id	INTEGER	11735	VLAN record ID (which is “id” of the VLANs in “get” action results)

Optional Parameters	

Name	Type	Example	Description
name	STRING or empty string	HQ	name of the VLAN
tags	INTEGER or STRING or empty string	111,112	ID of tags to be assigned to this VLAN. Multiple tags IDs are assignable with comma delimiter. Empty parameter will result in no tags being assigned.

Example URL	api/v1/api.php?target=vlan&action=update&id=11735&name=OReilly&tags=156,159 api/v1/api.php?target=vlan&action=update&id=11735&name=OReilly api/v1/api.php?target=vlan&action=update&id=11735&tags=156,159 api/v1/api.php?target=vlan&action=update&id=11735&name= api/v1/api.php?target=vlan&action=update&id=11735&tags=

removeTags

URL	api/v1/api.php?target=vlan&action=removeTags
Description	Remove tags from a domain's VLAN(s).
Returns	Examples

SUCCESSFUL	*{"success":1,"message":"VLAN tags for VLANs updated","id":"1","data":[]}*
ERROR	*{ "success":0, "message":"error message" }*

Required Parameters	

Name	Type	Example	Description
domainId	INTEGER	1	domain ID for domain whose VLANs will be un-tagged
vlan	INTEGER or STRING	500 500-510	a single VLAN (not the record ID from the get action results, but actual VLAN ID), OR range of VLANs (using hyphen for minimum through maximum), OR comma-delimited list of VLANs and/or VLAN ranges
tagId	INTEGER or STRING	150,99	a single tag ID or comma-delimited list of tag ID's

Optional Parameters	None
Example URL	api/v1/api.php?target=vlan&action=removeTags&domainId=1&vlan=500-510&tagId=159 api/v1/api.php?target=vlan&action=removeTags&domainId=1&vlan=400,406,500-510&tagId=159,160

addTags

URL	api/v1/api.php?target=vlan&action=addTags
Description	Add tags to a domain's VLAN(s).
Returns	Examples

SUCCESSFUL	*{"success":1,"message":"VLAN tags for VLANs updated","id":"1","data":[]}*
ERROR	*{ "success":0, "message":"error message" }*

Required Parameters	

Name	Type	Example	Description
domainId	INTEGER	1	domain ID for domain whose VLANs will be un-tagged
vlan	INTEGER or STRING	515 500-510	a single VLAN (not the record ID from the get action results, but actual VLAN ID), OR range of VLANs (using hyphen for minimum through maximum), OR comma-delimited list of VLANs and/or VLAN ranges
tagId	INTEGER or STRING	150,99	a single tag ID or comma-delimited list of tag ID's

Optional Parameters	None
Example URL	api/v1/api.php?target=vlan&action=addTags&domainId=1&vlan=500-510&tagId=159 api/v1/api.php?target=vlan&action=addTags&domainId=1&vlan=400,406,500-510&tagId=159,160

smartAssign

URL	api.php?target=vlan&action=smartAssign
Description	Finds and enables an unassigned VLAN matching the provided search parameters and optionally renames it.
Returns	Examples

SUCCESSFUL	Return Data: the full domain description, plus the chosen VLAN with its new name (optional) *{"success":1,"message":"VLAN enabled. ","data":{"21":{"id":"21","domain":"Swisscom Demo","attr_6c":"{\"range\":\"E\"}","attr_custom":null,"vlans":[{"id":"46183","vlan":"1808","name":" RESERVED_DEV"}]}}}*
ERROR	*{ "success":0, "message":"error message" }*

Required Parameters

Name	Type	Example	Description
domainId	INTEGER	15	The Id number of the domain you would like to Smart Assign from.

Optional Parameters

Name	Type	Example	Description
tags	STRING	5,7,12	A comma-separated list of the Tag Ids being used for the Smart Assign.
name	STRING	CustomerName_13 , RESERVED_DEV	The name that will be assigned to the VLAN matching the given search parameters.

Example URL	api.php?target=vlan&action=smartAssign&domainId=21&name=RESERVED_DEV&tags=5,7,12

How Do I...

How Do I... (Use Cases)

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...

IP Blocks - Update Fields

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";
```

IP Blocks - Assign / Subassign

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 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]=N
+= '&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))
```

DNS

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
$apiResponse = $apiClient->sendRequest('zone', 'add', $params);
if ($apiResponse->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 = $apiResponse->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();
}
?>
```

DHCP

Context: I need to attach the DHCP module as a child

Click here to expand...

DHCPv2 functionality is enabled on a particular resource by attaching a DHCP Module as a child. A command to do this is as follows:

```
            [ProVision root]/api/v1/api.php?target=resource&action=add

            data:
meta[type]: dhcp_module
meta[name]: [parent resource id] DHCP Module
meta[parent_id]: [parent resource id]
```

The special resource type “dhcp_module” indicates to ProVision that the DHCP system is enabled for the parent object. The attributes associated with the “dhcp_module” resource govern the DHCP system's behavior.

Updating the attributes of a DHCP Server uses a Resource Update command:

```
[ProVision root]/api/v1/api.php?target=resource&action=update&meta[id]=2178
&meta[type]=dhcp_module&fields[_dhcp_attributes][]={"type":"ISC","notes":"notes go
here","username":"username","port":"port","config_test":"/etc/init.d/dhcpd
configtest","server_stop":"/etc/init.d/dhcpd stop","server_start":"/etc/init.d/dhcpd
start","config_path":"/tmp/dhcpd.conf","option_routers":"192.168.0.0","option_domain_na
line 1","freeLine2":"free line 2","freeLine3":"free line 3"}
```

This command appears rather complicated, but can be broken apart into reasonable pieces. The first section:

```
target=resource&action=update&meta[id]=2178&meta[type]=dhcp_module
```

is familiar from other parts of ProVision. We are updating a resource of type “dhcp_module” whose resource id is 2178. The second section of the command details the update values, starting with

```
fields[_dhcp_attributes][]=
```

which contains a JSON-encoded string of all the fields specific to a DHCP server's function. When expanded into its full object form it is substantially easier to digest:

```
{
            "type":"ISC",
            "notes":"notes go here",
            "username":"username",
            "port":"port",
            "config_test":"/etc/init.d/dhcpd configtest",
            "server_stop":"/etc/init.d/dhcpd stop",
            "server_start":"/etc/init.d/dhcpd start",
            "config_path":"/tmp/dhcpd.conf",
            "option_routers":"192.168.0.0",
            "option_domain_name_servers":"ns1.6connect.com",
            "option_domain_name":"6connect.com",
            "authoritative":"1",
            "default_lease_time":"600",
            "max_lease_time":"7200",
            "local_port":"67",
            "log_facility":"local7",
            "password":"password",
            "server_ip":"192.168.0.1",
            "freeLines":3,
            "freeLine1":"free line 1",
            "freeLine2":"free line 2",
            "freeLine3":"free line 3"
}
```

This object describes all the most common DHCP server configuration options. For a full explanation of each of the fields, see the Detailed API Specification later in this document.

Please note that the object above must be passed to the DHCP system as a JSON-encoded string. It must be passed into the special “_dhcp_attributes” attribute for it to be functional, as in the example URL.

Context: I need to add a DHCP aggregate

Click here to expand...

An example command to add a DHCP Aggregate is:

```
[ProVision root]/api/v1/api.php?target=ipam&action=add&block=192.168.0.0/24&rir=
1918&vlan=&tags=&region=&resourceId=1282&allowSubAssignments=true
```

The important part to note is that the IP block is being assigned to resourceId 1282, which corresponds to the DHCP Available resource. The DHCP Available resource is a system-level resource which is used to hold all unassigned DHCP IP addresses. Every instance has its own DHCP Available resource, whose id can be found with the following command:

```
[ProVision root]/api/v1/api.php?target=resource&action=get&slug=dhcp-available
```

New DHCP subnets and hosts draw their IPs from this pool. If there are no IPs in the DHCP Available pool new subnets and hosts will not be able to be created.

DHCP IP aggregates are fetched, updated, split, and deleted using the standard IPAM management API endpoints. Please see the IPAM API Documentation. for details.

Context: I need to add a DHCP Pool

Click here to expand...

Similar to how the “dhcp_module” resource was created above, the command to create a DHCP Pool is as follows:

```
[ProVision root]/api/v1/api.php?target=resource&action=add&meta[type]=dhcp_pool
&meta[name]=New
Subnet&fields[_dhcp_type][]=subnet&fields[_dhcp_pool_attributes][]={"mac":"","rangeStar
Line 1","freeLine2":"Free Line 2","freeLine3":"Free Line 3"}
```

The first half of this command is relatively straightforward:

```
target=resource&action=add&meta[type]=dhcp_pool&meta[name]=New Subnet
```

This section informs the API that we wish to create a new, empty “dhcp_pool” resource whose name is “New Subnet.”

```
fields[_dhcp_type][]=subnet&fields[_dhcp_pool_attributes][]={"mac":"","rangeStart":"",
"rangeEnd":"","freeLines":3,"freeLine1":"Free Line 1","freeLine2":"Free Line
2","freeLine3":"Free Line 3"}
```

The second half of the command behaves in a similar manner to the “dhcp_module.” The “_dhcp_pool_attributes” field holds a JSON-encoded string which describes the dhcp_pool resource. When expanded, the JSON string becomes the following object:

```
{
            "mac":"",
            "rangeStart":"",
            "rangeEnd":"",
            "freeLines":3,
            "freeLine1":"Free Line 1",
            "freeLine2":"Free Line 2",
            "freeLine3":"Free Line 3"
}
```

For a full explanation of each of the fields, see the Detailed API Specification.

> Please note that the object above must be passed to the DHCP system as a JSON-encoded string. It must be passed into the “_dhcp_pool_attributes” attribute for it to be functional, as in the example URL.

Once a dhcp_pool resource is in the system it can be updated with IP data obtained from the IP Management system. Under DHCPv2, the DHCP system uses all the standard IPAM API endpoints and can make use of both the smartAssign and the directAssign methods. Please see the IPAM API documentation for details.

Context: I need to link a DHCP pool to a DHCP server

Click here to expand...

An example of building a link between a dhcp_pool and a DHCP Server is:

```
[ProVision root]/api/v1/api.php?target=resource&action=addLink&resource_id1=2178&
resource_id2=1452&relation=dhcpPoolLink
```

The Resource Linkage system controls which DHCP Pools are associated with a given DHCP Server. In the case of linking a DHCP Pool to a DHCP Server, the relation used is “dhcpPoolLink”. This is a directional link, so it is important that resource_id1 and resource_id2 do not get confused.

```
relation:  "dhcpPoolLink"
resource_id1:  the id of the dhcp_module this pool is being linked to
resource_id2:  the id of the dhcp_pool being linked
```

It is very important that resource_id1 not be confused with resource_id2. The link will not function with the values reversed.

To undo the above and break a DHCP Pool link, use the same command but substitute “deleteLink” for the action “addLink”.

```
[ProVision root]/api/v1/api.php?target=resource&action=deleteLink&resource_id1=2178&
resource_id2=2179&relation=dhcpPoolLink
```

Context: I need to push a DHCP config file

Click here to expand...

Once the server has been configured according to the previous sections, hitting the following API endpoint will trigger a DHCP push:

```
[ProVision root]/api/v1/api.php?target=dhcp&action=push&id=2178
```

The “id” in the above string is the id of the dhcp_module resource attached to the server you whose configuration is to be pushed. The API return payload will contain success or failure codes, as well as a description of any errors which might have occurred.

When a DHCP configuration file is pushed an SSH connection is opened to the configured server using the user, password, and port supplied to the '_dhcp_attributes' attribute on the dhcp_module resource. If the system successfully connects, it will assemble a DHCP configuration from the information given to the dhcp_module's '_dhcp_attribute' attribute and then parse and add in all linked dhcp_pool resources.

After the assembled file has been transferred to the DHCP server it will be placed in the location given by 'config_path' on the dhcp_module, and then the command described in 'config_test' will be run to determine whether or not this new file parses correctly. If 'config_test' is blank or omitted, this step is skipped.

If the file parses correctly the DHCP will be stopped and restarted according to the 'server_stop' and 'server_start' commands on the DHCP module. If there are errors at any point the system backs out, replaces old config files, and reports the errors via the 'message' return field of the API call.

Python SDK:

IP Blocks

Context: How do I create aggregates, get block information, and delete aggregates using the API / python SDK?

Click here to expand...

```
#!/usr/bin/python
from apiclient import APIClient, APIResponse

# REPLACE WITH CORRECT VALUES FOR YOUR INSTANCE

base_url  = 'https://<ProVision Instance URL>'

api_key   = '00-ABCDEFGHIJ123456'

api_secret_key = '0123456789abcdef0123456789abcdef'


# create the APIClient

client = APIClient(base_url, api_key, api_secret_key)



# create aggregate 1.2.3.0/24

target = 'ipam'

action = 'add'

params = {'block': '1.2.3.0/24', 'rir':'ARIN'}

url = client.get_request_url(target, action, params)

print url

response = client.make_api_call(target, action, params)

print response



# get block 1.2.3.0/24

target = 'ipam'

action = 'get'

params = {'block': '1.2.3.0/24'}

url = client.get_request_url(target, action, params)

print url

response = client.make_api_call(target, action, params)

print response
```

```
# delete aggregate 1.2.3.0/24

target = 'ipam'

action = 'delete'

params = {'block': '1.2.3.0/24'}

url = client.get_request_url(target, action, params)

print url
```

```
response = client.make_api_call(target, action, params)

print response
```

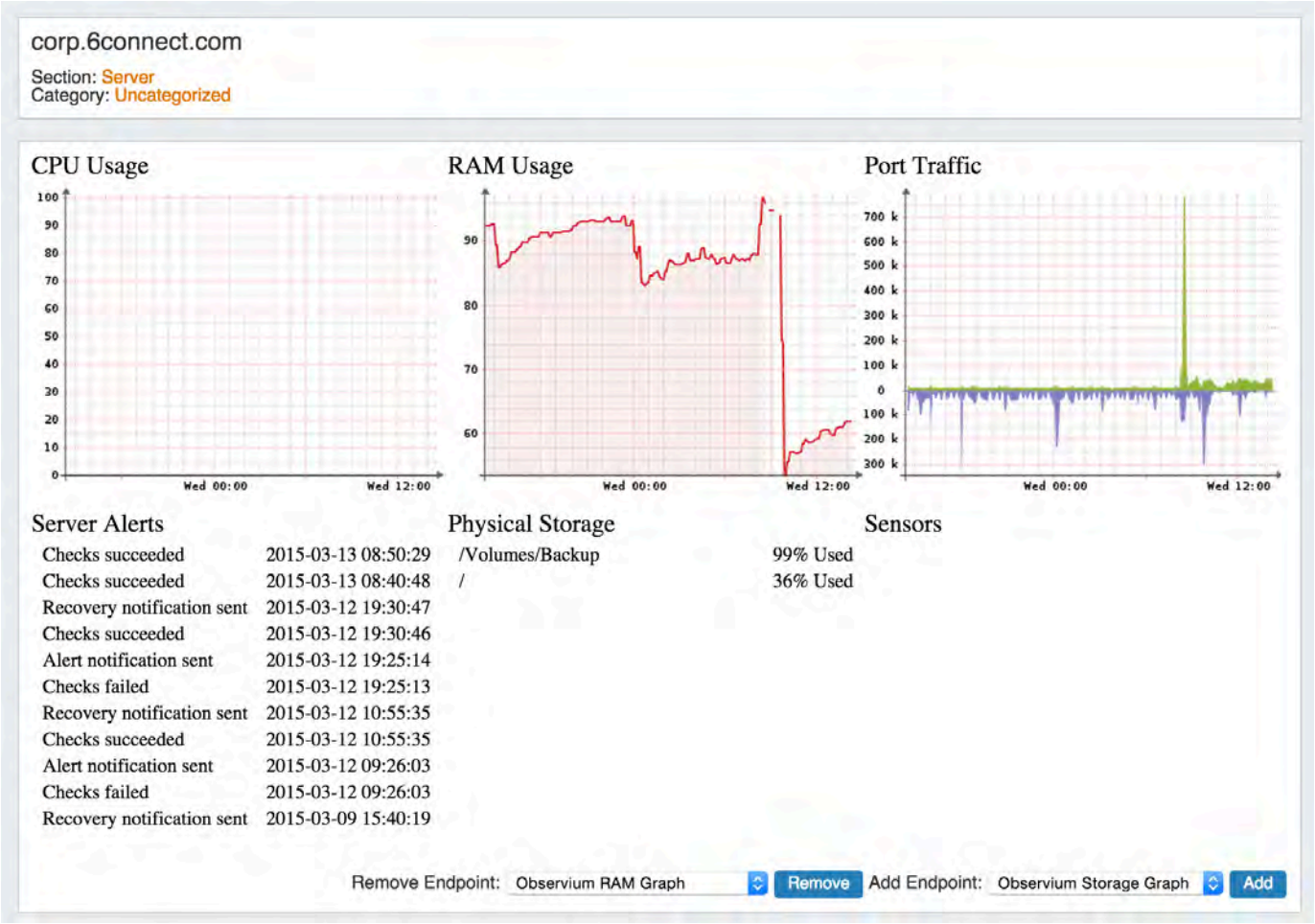
Reverse API

Reverse API

Overview

Reverse API Tools - Beta

ProVision's Reverse API calls and UI elements allow for integration with outside APIs to improve workflow and create custom display content. In the ProVision user interface, the Reverse API page allows for endpoints to be built and provides a text editor to create presentation JavaScript commands. This JavaScript presentation code is then displayed in the Reverse API Gadget.



Using the ProVision Reverse API (rAPI), you can perform these same actions and customize to meet your specific needs outside of the ProVision UI.

API Call Formatting

Reverse API (rAPI) calls are made to hit a user-defined URL or command line program. If the rAPI type is set to 'external,' the system makes a HTTP request. If the rAPI type is set to 'local', it executes the call on the command line of the local machine. In both cases the call is first customized with the attributes of the resource supplied to the execute function. This allows a single rAPI endpoint to serve a wide array of individual resources, fetching only information relevant to that particular resource without having to store anything locally.

rAPI calls are formatted thusly:

```
http://observium.tcp0.com/graph.php?height=200&width=265&type=device_bits&legend=no&userna
```

The interesting part about this URL is the bit in curly-braces: {observium-id}. When this call is made, the system decodes the URL by searching for everything within curly-braces and replaces it with data pulled from a resource. A rAPI call may have as many or as a few curly-brace replacement targets as is needed. Each curly-brace target will be replaced with exactly one resource attribute. All targets must be successfully replace for the command to succeed.

For example, when this rAPI call is decoded with information from the 1-dev resource, the decoded call is as follows:

```
http://observium.tcp0.com/graph.php?height=200&width=265&type=device_bits&legend=no&userna
```

The rAPI service then pulls the data from that URL, pairs it with its presentation code, and returns it via the execute endpoint. If a user attempted to execute the above call on a resource which did not have the “observium-id” property, the execute endpoint would return the following error: "Reverse API Call references token 'observium-id', which does not exist in resource."

> All curly-brace calls reference a resource attribute by their unique resource slugs. The list of available resource attributes can be customized using the existing resource system endpoints.

> Curly-brace targets may take the form {parent.observium-id}. The “parent.” prefix indicates that in decoding this particular brace, the attributes of the resource's parent are to be used.

Reverse API Detail:

For detail on the Reverse API calls and parameters, proceed to Reverse API - Detail.

Reverse API - Detail

Reverse API Module

reverseAPI_add

URL	/api/v1/api.php?target=reverseAPI&action=add
Description	Add a Reverse API Endpoint
Returns	**Examples:**

SUCCESSFUL	performs ResourceAPI->Get operation to return the newly-created entry, thus will appear similar to a "Get" return: {"success":1,"message":"Reverse API endpoint added","data":{"id":"1","name":"TestPoint2","type":"local","created_by":"user","last_modified":"2015-04-01 12:24:57","call":"http:\/\/www.test.com?api.php","presentation":null}}
ERROR	{"success":0, "message":"error message"}

Required Parameters

Name	Type	Example	Description
name	STRING	TestPoint2	The name of the rAPI object
type	STRING	local	The type of the rAPI object. Valid settings are 'local' and 'external'.
call	STRING	http://www.test.com?api.php&server={serverId}	The URL or the system path which is first decoded against a resource and then hit when this rAPI endpoint is invoked.

Optional Parameters

Name	Type	Example	Description
presentation	STRING	endpoint = function(data, outputDiv) {console.log(outputDiv);};	The Javascript code used to display this endpoint on a resource page.

Example URL	/api/v1/api.php?target=reverseAPI&action=add&name=TestPoint2&type=local&call=http://www.test.com?api.php&server={serverId}

reverseAPI_get

URL	/api/v1/api.php?target=reverseAPI&action=get
Description	Fetches one or more Reverse API endpoints

Returns	**Examples:**

SUCCESSFUL	An array of ReverseAPI objects: {"id":"535","name":"Observium RAM Graph","type":"external","created_by":"user","last_modified":"2015-03-18 09:16:49","call":"http:\/\/observium.tcp0.com\/graph.php?height=200&width=265&type=device_ucd_memory&leg = function(data, outputDiv) {\n\n outputDiv.css('font-family','Helvetica, Arial, sans-serif;');\n\n\toutputDiv.append {\n\t\toutputDiv.append(\" No data returned<\/div>\");\n\t\treturn;\n\t}\n\n\toutputDiv.append('<div style=\"ma \/><\/div>');\n};"}
ERROR	{"success":0, "message":"error message"}

Return Detail:

Name	Type	Description
id	INTEGER	The id of the rAPI objectSt
name	STRING	The name of the rAPI object
type	STRING	The type of the rAPI object
created_by	STRING	What user created the rAPI object
last_modified	DATETIME	When this rAPI object was last changed
call	STRING	The URL or the system path which is first decoded against a resource and then hit when this rAPI er
presentation	STRING	The Javascript code used to display this endpoint on a resource page.

Required Parameters	None
Optional Parameters	

Name	Type	Example	Description
id	INTEGER	535	The id of the rAPI objectSt
name	STRING	TestPoint2	The name of the rAPI object
type	STRING	external	The type of the rAPI object. Valid settings are 'local' and 'external'.
created_by	STRING	user	The user who created the rAPI object

Example URL	/api/v1/api.php?target=reverseAPI&action=get&name=TestPoint2

reverseAPI_delete

URL	/api/v1/api.php?target=reverseAPI&action=delete
Description	Delete a rAPI endpoint
Returns	**Examples:**

SUCCESSFUL	{"success":1,"message":"Reverse API endpoint deleted"}
ERROR	{"success":0, "message":"error message"}

Required Parameters	

Name	Type	Example	Description
id	INTEGER	535	The id of the rAPI objectSt
name	STRING	TestPoint2	The name of the rAPI object

Optional Parameters	None
Example URL	/api/v1/api.php?target=reverseAPI&action=delete&id=535&name=TestPoint2

reverseAPI_update

URL	/api/v1/api.php?target=reverseAPI&action=update
Description	Update an existing rAPI endpoint
Returns	**Examples:**

SUCCESSFUL	ResourceAPI->Get updated object, thus will appear the same as a "Get" return: {"id":"535","name":"Observium RAM Graph","type":"external","created_by":"user","last_modified":"2015-03-18 09:16:49","call":"http:\/\/observium.tcp0.com\/graph.php?height=200&width=265&type=device_ucd_memory&leg = function(data, outputDiv) {\n\n outputDiv.css('font-family','Helvetica, Arial, sans-serif;');\n\n\toutputDiv.append {\n\t\toutputDiv.append(\" No data returned<\/div>\");\n\t\treturn;\n\t}\n\n\toutputDiv.append('<div style=\"ma \/><\/div>');\n};"}
ERROR	{"success":0, "message":"error message"}

Required Parameters	

Name	Type	Example	Description
id	INTEGER	535	The id of the rAPI objectSt

Optional Parameters	

Name	Type	Example	Description
name	STRING	TestPoint2	The name of the rAPI object
type	STRING	external	The type of the rAPI object. Valid setting
call	STRING	http://www.test.com?api.php&server={serverId}&action=12	The URL or the system path which is firs
presentation	STRING	endpoint = function(data, outputDiv) {console.log(outputDiv);};	The Javascript code used to display this

Example URL	/api/v1/api.php?target=reverseAPI&action=update&id=535&name=TestPoint2

reverseAPI_execute

URL	/api/v1/api.php?target=reverseAPI&action=execute
Description	Decodes a rAPI call against a Resource, executes the resulting string, returns the result along with the presentation code

Returns	**Examples:**

SUCCESSFUL	The result of a rAPI call customized for the supplied resource {“data”:”\u0017\u00fbD","presentation":"endpoint = function(data, outputDiv) {\n\n\toutputDiv.css('font-family','Helvetica, Arial, sans-serif;');\n\n\toutputDiv.append(\" Port Traffic<\/div>\");\n\n\tif (data == \"\") {\n\t\toutputDiv.append(\" No data returned<\/div>\");\n\t\treturn;\n\t}\n\n\toutputDiv.append('<div style=\"margin-left:-15px;margin-top:-3px;\"><\/div>');\n};"}}
ERROR	{"success":0, "message":"error message"}

Return Detail:

Name	Type	Description
data	MIXED	The result of the reverse API call. Could be binary data, could be JSON.
presentation	STRING	The presentation javascript which pairs with this rAPI call

Required Parameters

Name	Type	Example	Description
id	INTEGER	537	The id of the rAPI objectSt
resource_id	INTEGER	2245	ID of the resource

Optional Parameters

None

Example URL

/api/v1/api.php?target=reverseAPI&action=execute&id=537&resource_id=2245

CLI (Alpha)

Command Line Interface - 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 an makes it available.
```

Toolkit

Toolkit Modules

6connect provides customers with additional CLI modules for ProVision power users in the /tools directory. These tools are outlined below.

Users / Permissions

Add User

File: add-user.php

Description: Adds a new user to 6connect Provision. Especially useful if an admin has locked themselves out and has access to the local server, or for adding additional credentials to a new local installation. The tool walks through the required information.

Command: “php add-user.php"

Rebuild Permissions

File: rebuild-permissions.php

Description: Will rebuild the cached permissions for the specified resource id. Should not be used outside of a db reset, or a corruption of the permissions cache.

Command: “php rebuild-permissions.php"

DNS

Audit Forward DNS

File: audit_forward_dns.php

Description: Audits the forward DNS for a specified IP block, comparing the DNS records in 6connect to a publicly facing server. Will provide detail output showing resolved records, if records conflict and why.

Command: “php audit_forward_dns.php”

Options:

-v Sets verbose mode to print debug information

-a <message key> run as asycronous request (note: you should not use this, it is for the GUI)

-b <ip block> IP block in cird notation to audit dns for.

-n <name server> FQDN or IP address of specific name server to resolve records against. Defaults to 8.8.8.8.

-h print help

Audit Reverse DNS

File: audit_reverse_dns.php

Description: Audits the reverse DNS for a specified IP block, comparing the DNS records in 6connect to a publicly facing server. Will provide detail output showing resolved records, if records conflict and why.

Command: "php audit_reverse_dns.php"

Options:

-v Sets verbose mode to print debug information

-a <message key> run as asycronous request (note: you should not use this, it is for the GUI)

-b <ip block> IP block in cird notation to audit dns for.

-n <name server> FQDN or IP address of specific name server to resolve records against. Defaults to 8.8.8.8.

-h print help

Database

Database Analyzer

File: databaseAnalyzer.php

Description: Takes two MySQL dump files for input, one thought to be the “good” database, and one the “bad” to compare against each other for differences. Is used internally to audit database changes after an upgrade to ensure all changes for the upgrade were committed by comparing against the template database for that release. Can be useful for diagnosing missed upgrades, or other database issues. Could be used on any set of databases, and is not 6connect specific.

Command: "php databaseAnalyzer.php"

Options:

-v Sets verbose mode to print debug information

-a <database name> Required - the 'good' database to comapre against.

-b <database name> Required - the 'bad' database to check for incorrect information.

-g <globals file> Optional - The script will automatically look to the globals.php file for database connection information. This option can be used to specify and alternate file, or file location.

-h print help

Database Reset

File: dbReset.php

Description: This tool resets portions or all of the database to factory defaults. Can reset only IP, DNS, or Resource information or all all three. If erasing resource information, you need to enter a user to assume global admin permission to log back in with. Even if this isn’t specified, the add-user tool can be used to add a user back after. We recommend creating a backup prior to using this tool, and only use if you are sure you know what you are doing!

Command:"php dbReset.php"

Options:

-v Sets verbose mode to print debug information

-i Erases any and all IP information. Tags, regions, and other IPAM preference settings are not changed.

-d Erases any and all DNS information including zones, records, and archive data. Does not affect DNS perference settings, or configured DNS servers.

-r Resets all resource information back to the factory defaults, and removes all associated information (user groups, files, etc).

-u <user name> User name that will have Global Admin permissions restored on the default resources

-h print help

Import

IPPlan Importer:

File: ipplan_import.php

Description: The IPPlan Importer is a command-line tool to import IPs from an IPPlan database into ProVision. Details on how to use this tool are available at IPPlan Importer.

Command: "php ipplan_import.php"

Options:

For options, refer to IPPlan Importer.

Internal 6connect Tools

The following tools contained in the /tools folder are for 6connect internal use only:

File: scp_wrapper.php - Internal 6connect tool, not meant for command line use.

File: s64_restart_master - Internal 6connect tool, not meant for command line use.

File: s64_restart_slave - Internal 6connect tool, not meant for command line use.

File: async_import.php - Internal 6connect tool, not meant for command line use.

File: importBigDump.php - Internal 6connect tool supporting IPPlan Importer, not meant for command line use.

File: ipplan_functions.php - Internal 6connect tool supporting IPPlan Importer, not meant for command line use.

File: dns_transfer_server.php - Internal 6connect tool, not meant for command line use.

File: observium_import.php - Internal 6connect tool, not meant for command line use.

File: digAndLookup.php

File: pullLAMPInfo.php

Additional Information:

- IPPlan Importer

IPPlan Importer

IPPlan Importer

Overview

The IPPlan Importer is a command-line tool to import IPs from an IPPlan database into ProVision. This tool can be used via two approaches: generating .csv files via the tool only, then using the ProVision IP Import UI to import the csv files (Connector), or as a full command-line import solution, bypassing the ProVision UI entirely (Importer).

Options:

1) Generate .csv files to use with the IP Import - Upload/Import from CSV tool through the ProVision UI.

2) Import IPs directly into ProVision without accessing the UI.

Before you begin:

You will need:

- A MySQL export of IPPlan created from mysqldump, located on the same server / accessible from the IPPlan Importer tool.
- Administrative access to your ProVision instance folder

Accessing the Tool and Showing Help Instructions:

The tool is located in tools/ipplan_import.php. A help document is provided listing example commands and detailed situation-specific options.

To execute it, you must preface it with "php" program. From from your instance's root folder, the command would be:

"php tools/ipplan_import.php --help"

If you are in the tools folder, the command would be:

"php ipplan_import.php --help"

This will bring up the help / instruction document. It is also provided below.

Click here to see the IPPLan Importer help text...

Help File Text

```
usage: php ipplan_import.php [options] <SQL dump file>

Examples:

For creating CSV's for use in ProVision IP importer:
    php ipplan_import.php filename.SQL --csv

For creating CSV's, but do not load SQL file if one already has been:

    php ipplan_import.php filename.SQL --csv --no-overwrite

For importing data from SQL dump. Note that you must specify RIR:
    php ipplan_import.php filename.SQL --rir=1918

For importing data from SQL dump, if one hasn't been loaded already:
    php ipplan_import.php filename.SQL --rir=1918 --no-overwrite

Caution: By default, SQL dump will be loaded into database 'provision_import_ipplan'.
If database already exists it will be dropped, unless --no-overwrite option is
specified.

options:

  --only-import - Skip loading SQL file. Instead, import existing data in database
'provision_import_ipplan'.

  --only-sql - Load SQL dump into database 'provision_import_ipplan'. --only-import
can be run at a later time to use loaded database.

  --rir=RIR - (required for import) RIR for IP's

  OPTIONS IF LOADING SQL FILE:

  --no-overwrite - Load SQL dump only if database 'provision_import_ipplan' is
currently non-existent. Otherwise, drop existing database.

  OPTIONS IF IMPORTING DATA:

  --csv - generate CSV's instead of importing directly to ProVision.
```

Connector Method (Results in .csv files only):

This method creates .csv files that you can use with the IP Import - Upload/Import from CSV tool through the ProVision UI.

1. From the tools/ folder, run the tool with the --csv option. Be sure to reference your sql export file name and location:

   "php ipplan_import.php ../ipplanv6.sql --csv"

   (where '../ipplanv6.sql' is the path to your SQL file)
   This will generate two files: ipplan_aggregates.csv and ipplan_hosts.csv. Both can be used to import in the IP import section of ProVision.
2. Copy / Download the generated .csv files to a browsable directory on your local machine, to be accessed by the ProVision UI.
3. Through the ProVision UI, import the ipplan_aggregates.csv file as described in IP Import - Upload/Import from CSV.
4. Through the ProVision UI, import the ipplan_hosts.csv file as described in IP Import - Upload/Import from CSV.

Note that you must import ipplan_aggregates first. into IP Import - Upload/Import from CSV to create the aggregates ipplan_hosts.csv

will use.

Then, import ipplan_hosts.csv as described in IP Import - Upload/Import from CSV.

Importer Method (Results in full import):

This method has the tool process the import task. Using this case, you must set the RIR in the command line for all of the IP's.

Run & Set the RIR:

1. From the tools/ folder, run the tool with the --rir option, referencing your SQL file location:

   "php ipplan_import.php ../ipplanv6.sql --rir=1918"
   (where '../ipplanv6.sql' is the path to your SQL file)

This will load your IPPlan database file into your mysql server and then import the hosts into ProVision. They will each be given the RIR you specified, as well as this text in the Notes field: 'IPPlan import'

The import may need to run for a number of minutes, depending on the size of your data. For reference, an import of 2100 hosts inside of 150 aggregates took approximately 10 minutes to complete during our testing.

Additional Run Options:

Additional run options for various combination of conditions are detailed below. The command is the text within the quotes only.

Load SQL Without Importing

You can load the SQL file, but NOT run an import, with the --only-sql option (feel free to include --no-overwrite option as well in case you don't want to overwrite a prior loaded SQL file):

"ipplan_import.php ../ipplanv6.sql --only-sql"

"ipplan_import.php ../ipplanv6.sql --only-sql --no-overwrite"

Generate CSV Without Loading SQL

If you have already loaded the sql file previously, you can generate CSV files without having to load the SQL file:

"ipplan_import.php --only-import --csv"

Generate CSV No Overwrite

If you want to just generate CSV files without re-loading the database file, you can as well:

"ipplan_import.php ../ipplanv6.sql --no-overwrite --csv"

Generate CSV With RIR

You can also generate csv files with the rir option:

"ipplan_import.php ../ipplanv6.sql --no-overwrite --csv --rir=1918"

Run Only Import

If you have already loaded your IPPlan database with the tool before, you can do --only-import:

"php ipplan_import.php --rir=1918 --only-import"

Run Import No Overwrite

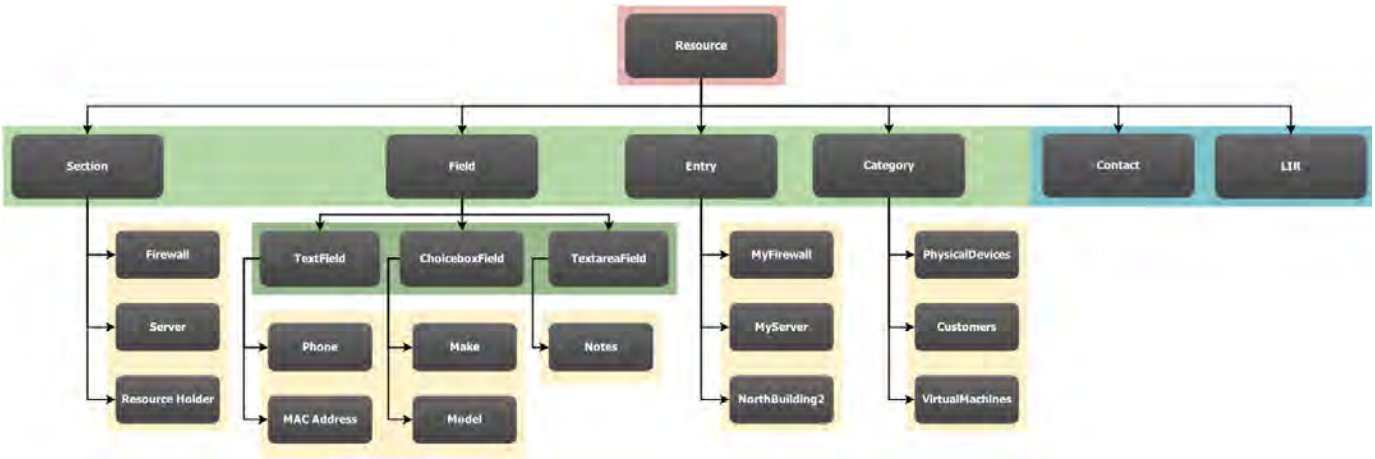
If you are not sure whether you have already loaded your IPPlan database with the tool before, you can specify --no-overwrite to NOT load the sql file if it was loaded already, or do load if it wasn't:

"ipplan_import.php ../ipplanv6.sql --rir=1918 --no-overwrite"

Resource Concepts 1

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:

- Resource Classes Doc
- Database Layout 1
- Asset System

Resource Classes Doc

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)*

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

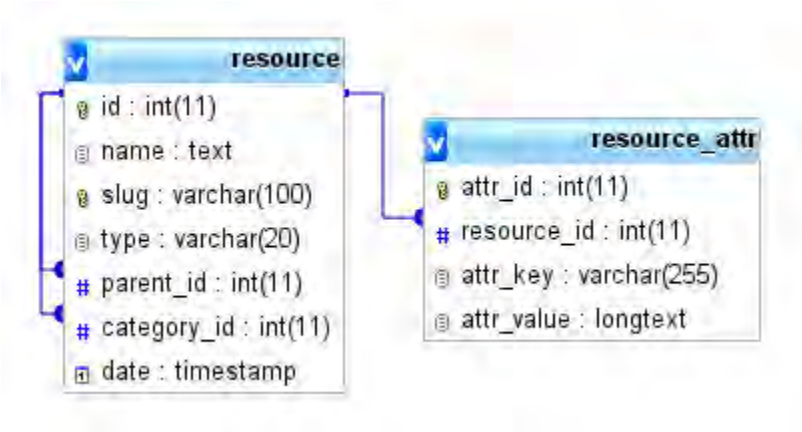
```
{
  "success": 1,
  "message": "Search successful",
  "data": [
    {
      "id": "1",
      "name": "TLR",
      "slug": "tlr",
      "type": "resource",
      "parent_id": null,
      "category_id": null,
      "attr": {}
    }
  ]
}
```

Database Layout 1

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;
```

Asset System

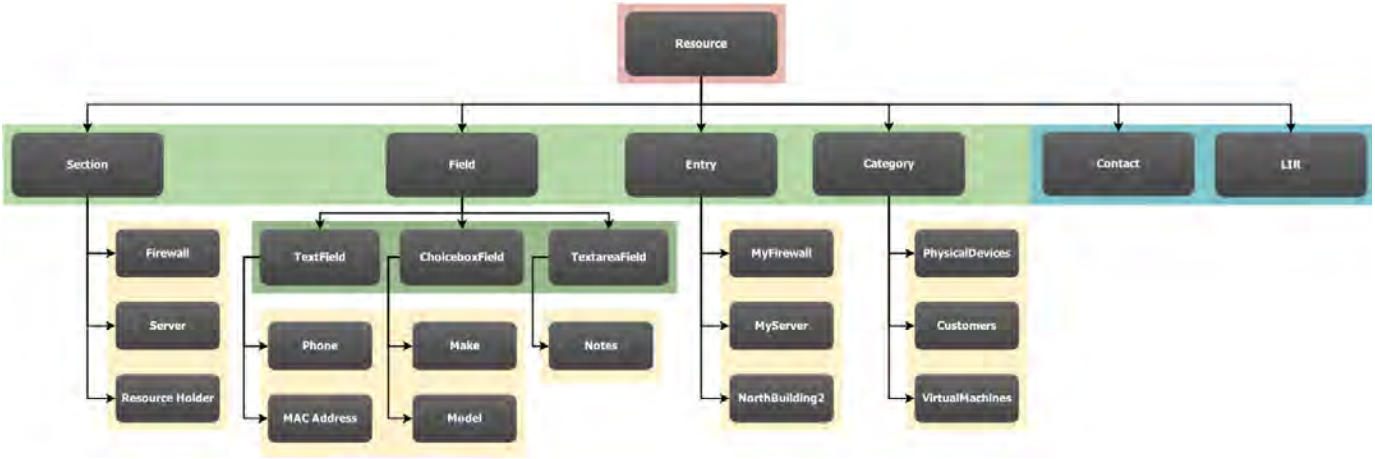
The Asset System

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) 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. Field has its 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 organizational 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 careful plan exactly how you want the resources in your system to be organized. Or you can ignore the whole thing completely and just let every resource have the default category of "uncategorized." Many user find that the ability to create hierarchal 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 it's 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, it would be recommended to email us first with an outline of what you're trying to do. Then the 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 recommend 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.

Portable Gadgets

Portable Gadgets

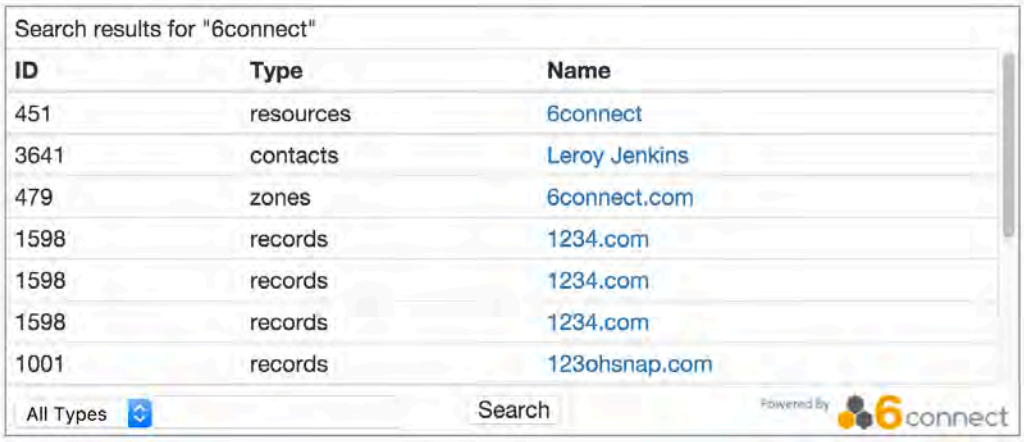
Overview

ProVision's Portable Gadgets are drop-in code snippets that use the ProVision API to bring in data to other systems or web pages. Portable Gadgets allow for quick data access and increased integration.

Currently, three Gadgets are available: Global Search, IPAM Search, and DHCP Search. Each Gadget comes in various default option types for display style, number of records return, or behavior. Additional client-side style customizations may be made to further integrate the gadget with company styles.

Global Search:

The Global Search Gadget utilizes ProVision's global search API to searching by name string and return ID, Type, and the Name. The search may be limited to specific types (IPAM, Resources, Zones, Records, Contacts, or All Types) if desired.



Global Search Option Types:

Type I.

Click here to expand...

Search results for "6connect"

ID	Type	Name
451	resources	6connect
3641	contacts	Leroy Jenkins
479	zones	6connect.com
1598	records	1234.com
1598	records	1234.com
1598	records	1234.com
1001	records	123ohsnap.com

All Types Search Powered By 6connect

- Returns 5 records per Type
- Auto Width, height set to 300px

Type II.

Click here to expand...

Search results for "6connect"

ID	Type
451	resources
3641	contacts
479	zones
1598	records
1598	records
1598	records
1001	records

Powered By 6connect

- Returns 5 records per Type
- No search input provided
- Fixed size (400px wide x 300px high)

Type III.

Click here to expand...

Search results for "6connect"

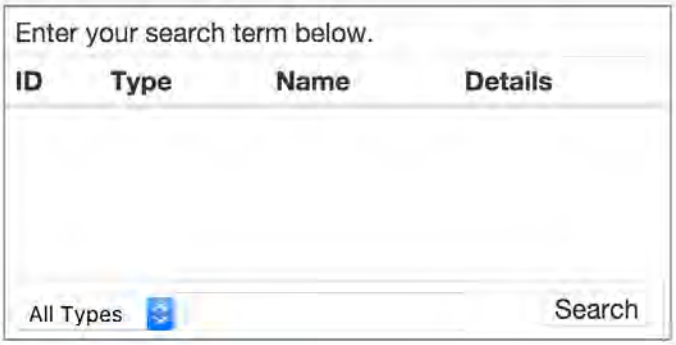
ID	Type	Name
451	resources	6connect
3641	contacts	Leroy Jenkins
479	zones	6connect.com
1598	records	1234.com

All Types Search Powered By 6connect

- Returns 3 records per Type
- Auto Width, height set to 200px

Type IV.

Click here to expand...



- No search on load
- Links stay in current window
- Fixed size (400px wide x 200px high)

Type V.

Click here to expand...

- Type I functionality restyled with CSS

IPAM Search:

The IPAM Search Gadget utilizes ProVision's IPAM API to return ID, CIDR, and Resource Name results searching by an IPAM Resource name string.

IPAM Option Types:

Type I.

Click here to expand...

IPAM query results for "6connect"

ID	CIDR	Resource
233414	10.0.0.0/16	6connect Available
233709	10.0.0.0/16	6connect Available
233824	10.0.0.0/16	6connect Available
234708	10.0.0.0/16	6connect Available

Search

Powered By 6connect

- Returns 5 records per Type
- Auto Width, height set to 200px

Type II.

Click here to expand...

IPAM query results for "6connect"

ID	CIDR
233414	10.0.0.0/16
233709	10.0.0.0/16
233824	10.0.0.0/16

Powered By 6connect

- Returns 3 records per Type
- No search input provided
- Fixed size (400px wide x 200px high)

Type III.

Click here to expand...

Enter your search term below.

ID	CIDR	Resource

Search

Powered By 6connect

- No search on load
- Links stay in current window
- Auto Width, height set to 300px

Type IV.

Click here to expand...

IPAM query results for "6connect"

ID	CIDR	Resource
233414	10.0.0.0/16	6connect Available
233709	10.0.0.0/16	6connect Available
233824	10.0.0.0/16	6connect Available
234708	10.0.0.0/16	6connect Available
234950	10.0.0.0/16	6connect Available

Search

- Type I functionality restyled with CSS
- Auto Width, height set to 400px

Type V.

Click here to expand...

IPAM query results for "6connect"

CIDR	updated	RIR
10.0.0.0/16	2016-02-29 14:57:20	1918
10.0.0.0/16	2016-03-09 14:17:26	1918
10.0.0.0/16	2015-10-01 09:26:54	1918
10.0.0.0/16	2016-01-28 09:21:03	1918
10.0.0.0/16	2016-03-09 13:57:40	ARIN

Search

- Customizable Fields
- Auto Width, height set to 300px

DHCP Search:

The DHCP Search Gadget utilizes ProVision's DHCP API to return results of varying types when searching for a name string. The search may be limited to specific types if desired.

DHCP query results for "test"

Name	DHCP Type	IP Assigned
RegionTest1	subnet	IP block ID 230785 (record not found)
RegionTest2	subnet	no IP
RegionTest4	subnet	IP block ID 230250 (record not found)
6connectTest	subnet	IP block ID 231193 (record not found)
QATest	subnet	IP block ID 231194 (record not found)

All Types test Search Powered by 6connect

DHCP Option Types:

Type I.

Click here to expand...

DHCP query results for "test2"

Name	DHCP Type	IP Assigned
Test2	subnet	198.0.0.0/32

All Types Search Powered by 6connect

- Returns 5 records per Type
- Auto Width, height set to 300px

Type II.

Click here to expand...

No results found for "6connect"

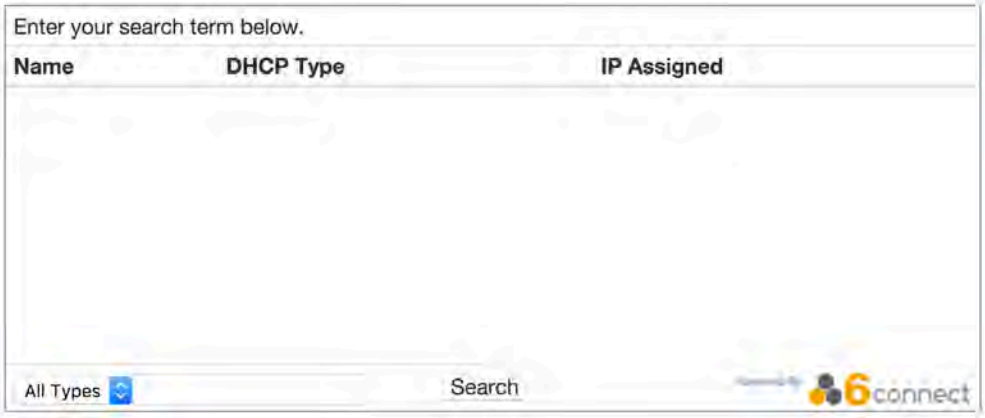
Name	DHCP Type	IP Assigned

Powered by 6connect

- Returns 3 records per Type
- No search input provided
- Fixed size (400px wide x 200px high)

Type III.

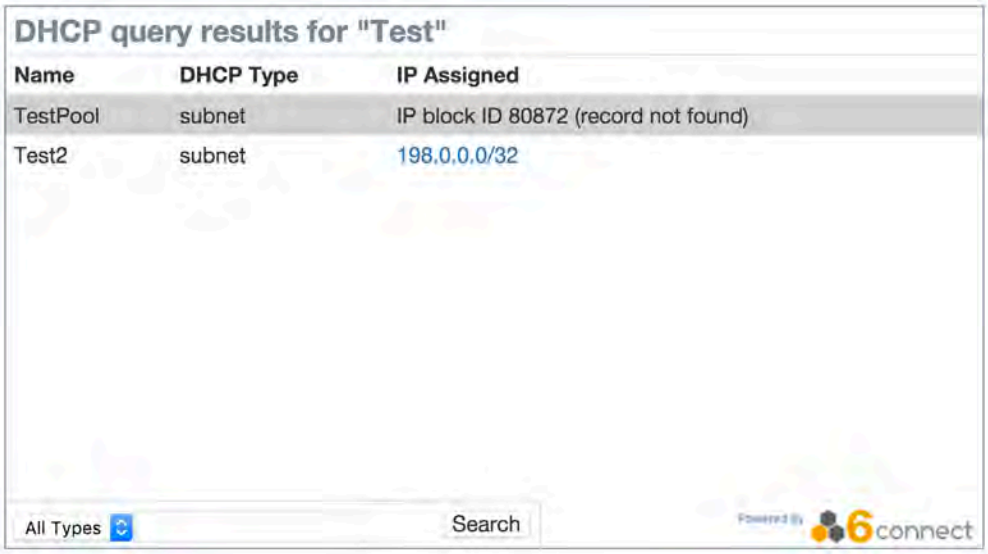
Click here to expand...



- No search on load
- Links stay in current window
- Returns 5 records per Type
- Auto Width, height set to 300px

Type IV.

Click here to expand...



- Type I functionality restyled with CSS
- Returns 5 records per Type
- Auto Width, height set to 300px

Additional Information:

- Getting Started With Portable Gadgets
- Portable Gadget Customization

Getting Started With Portable Gadgets

Getting Started

Adding a Portable Gadget to a Web Page

Requirements

Setting up Portable Gadgets requires a ProVision instance URL and valid API credentials (API key and secret key), as well as some familiarity with HTML, CSS, and Javascript.

Before You Begin

Before you begin, make sure you have access to the source code for the web page on which you want the portable gadget to appear, or create a simple new web page for testing purposes with minimal tags like this (with the location of steps included as comments):

```
<html>
 <head>
  <!-- STEP 1 WILL GO HERE -->

  <!-- STEP 3a WILL GO HERE -->
  <!-- STEP 3b WILL GO HERE -->
 </head>
 <body>
  <!-- STEP 2 WILL GO HERE -->
 </body>
</html>
```

Step 1: Reference dependency files

Include the following two scripts the <HEAD> block. They should reference your provision installation.

In this example, "https://cloud.6connect.com/6c_123" is the root folder of a 6connect installation, so replace that with your own 6connect instance URL.

```
<script type='text/javascript'
src="https://cloud.6connect.com/6c_123/portable/init.js"></script>
<link rel="stylesheet" type="text/css"
href="https://cloud.6connect.com/6c_123/portable/css/provision-portable.css">
```

Step 2: Create Gadget

Place the html tag for your gadget where you want it to appear in the page <body>, and give it a unique id.

Your gadget should be an html element in your page body (any contents inside the element will be removed). This example has a div referred to as "gadget" (as the ID).

```
<!-- provision gadget -->
<div id="gadget"></div>
```

Step 3: Initialize the Gadget

a) Preparing the settings

First, a settings variable must be prepared with the details of the gadget.

It must have the following keys for the provision url and api keys:

- provision_location
- provision_api_key
- provision_secret_key

...as well as the name of one of the gadget to load:

- search -- does a search with the global search API
- dhcp -- does a search with the DHCP API
- ipam -- does a search with the IPAM API

Here is an example for a search gadget, loading with the search term "6connect". By default it will load up to five results per type.

```
<script type='text/javascript'>
var settings = {
provision_location: "https://cloud.6connect.com/6c_123",
provision_api_key: "00-ABCDEFGHIJKLMN00",
provision_secret_key: "00abcd11ef22ghij3300klmno123",

search: "6connect",
};
</script>
```

Here is an example to load as an IPAM gadget, with an option to limit results to 10 records:

```
<script type='text/javascript'>
var settings = {
provision_location: "https://cloud.6connect.com/6c_123",
provision_api_key: "00-ABCDEFGHIJKLMN00",
provision_secret_key: "00abcd11ef22ghij3300klmno123",

ipam: "6connect",
limit: 10
};
</script>
```

An example with the DHCP gadget:

```
<script type='text/javascript'>
var settings = {
provision_location: "https://cloud.6connect.com/6c_123",
provision_api_key: "00-ABCDEFGHIJKLMN00",
provision_secret_key: "00abcd11ef22ghij3300klmno123",

dhcp: "Test",
};
</script>
```

b) Initialize the gadget HTML element

The gadget will initialize when scripted to do so with the provision() command. For the example html in section II, we can instantiate this gadget by running the following (as long as settings was prepared):

```
<script type='text/javascript'>
provision('#gadget', settings)
</script>
```

The first parameter targets the html element that is the gadget, using jQuery's selector format. In this case, it is an element with the ID "gadget".

Basic Page Example

Here is a simple example html page with a search gadget. In this example, a default search term of "6connect" is included, and the return limited 10 records.

```
<html>
 <head>
  <script type='text/javascript'
src="https://cloud.6connect.com/6c_123/portable/init.js"></script>
  <link rel="stylesheet" type="text/css"
href="https://cloud.6connect.com/6c_123/portable/css/provision-portable.css">
  <script type='text/javascript'>
   var settings = {
    provision_location: "https://cloud.6connect.com/6c_123",
    provision_api_key: "00-ABCDEFGHIJKLMN00",
    provision_secret_key: "00abcd11ef22ghij3300klmno123",

    search: "6connect",
    limit: 10
   };
   provision('#search_gadget', settings);
  </script>
 </head>
 <body>
  <div id="search_gadget"></div>
 </body>
</html>
```

Portable Gadget Customization

Customizing Portable Gadgets

Customization Options

Portable Gadgets come with both built-in customization options as well as the ability to edit styles locally through CSS.

Required Settings:

The required fields are:

provision_location: the location of the provision installation
provision_api_key: available from the Admin API Tab in ProVision
provision_secret_key: available from the Admin API Tab in ProVision

Other required fields (by gadget)

Search:
  search: "string"
IPAM:
  ipam: "string"
DHCP:
  dhcp: "string"

Where, in each case, "string" is the search term used on the initial loading of the gadget. If it is an empty string (literally, ""), then gadget will load without a pre-defined search.

Optional Settings:

These settings must be included as parameters when initializing the gadget. See Getting Started With Portable Gadgets.

limit: positive integer, or -1

The record return limit of the gadget. In some gadgets (search and DHCP) there are multiple types, so it limits per type. The default limit, if this option is not provided, is 5. If you specify limit a -1 (or basically any negative number), it will not enforce a limit.

links_change_window: true

setting this to true (not string "true" but actual true type in javascript) will make links change the current window. The default behavior (not having this option or false), would have links open a new tab.

interact: false

By default, there will be a search box on all these gadgets. But with this option set to false (again, using false not string "false") will remove that box so search cannot be changed.

Sample CSS Customization

Alternating Row Formatting:


```
<style>
.pg-subtitle {
    color: lightslategrey;
    font-size:1.4em;
    font-weight: bold;
}

.pg-field-name a:hover {
    text-decoration: none;
    color: white;
    background-color: gray;
}

.pg-results tr:nth-child(2n+1) {
    background: white;
}
.pg-results tr:nth-child(2n+0) {
    background: lightgray;
}
</style>
```

Help & Support

Help & Support

For setup assistance or additional information, you can contact our support team at 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

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.

Table of Contents

Common Tasks

Here is a list of shortcuts to walkthroughs and information on performing common tasks in ProVision:

IPAM

IP Blocks

Adding/Editing blocks

Aggregating/Splitting blocks

SWIP configuration and use

RPSL configuration and use

IP Import

Import Aggregate Blocks

IP Import from CSV

DNS

Importing DNS Data

Import DNS Zones:

BIND DNS Zone Upload / Import

PowerDNS Zone Import

DynECT Zone Import

Peering

Peering Common Tasks

Adding routers

Adding sessions

Importing sessions

Resources

Importing Resource Data

Resource Import from CSV

Resource Importer Walkthrough

UI Tours

Administration

Managing Group and User Permissions

DNS

PowerDNS w/ MySQL Support

Broken macro

FAQ

FAQ

How can take a manual backup of ProVision?

From the command line in your ProVision instance, you can get a manual SQL backup using the following way:

```
mysqldump -uUSER -pPASSWORD DATABASE > bkup.sql
```

You can get the values for USER, PASSWORD, and DATABASE by reading them from the following file:

```
[ProVision Root]/data/globals.php
```

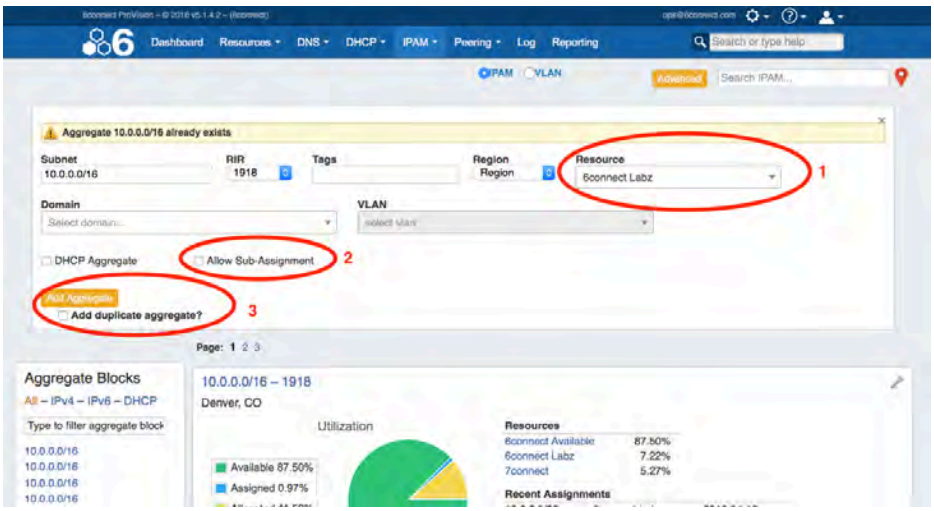
An example of a completed command:

```
mysqldump -uroot -p6connect_DBPSWD qa_5_1_5 > bkup.sql
```

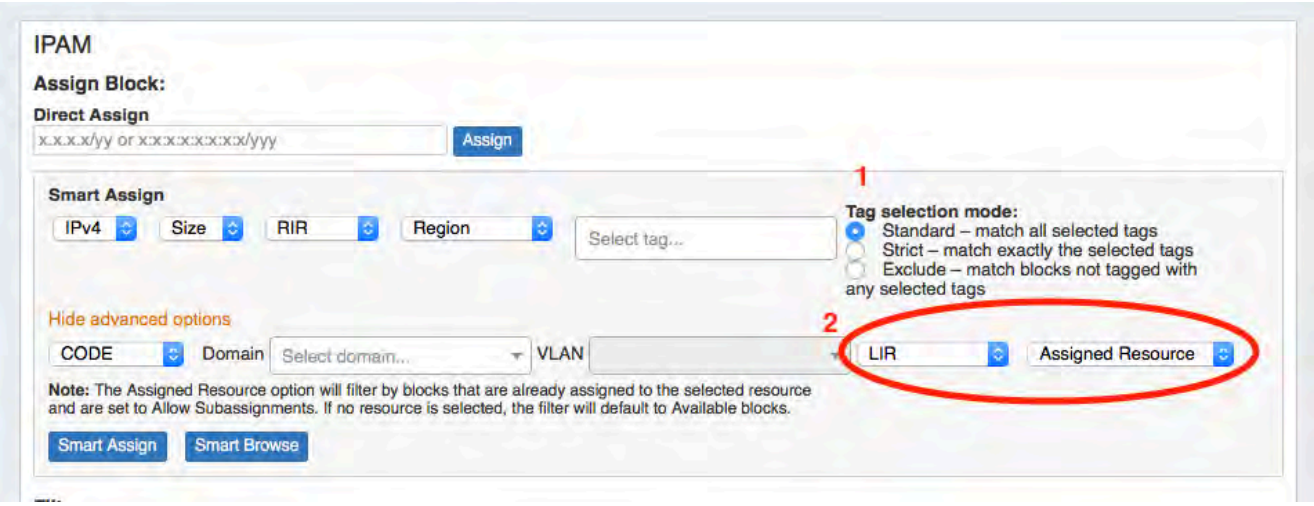
You can then use this .sql file to restore the old version should you need to revert the installation.

How can I manage overlapping/duplicate IP blocks?

When managing duplicate/overlapping IP ranges withing ProVision, there are several methods you can use to ensure that blocks are "unique" and still leverage functions like Smart Assign. The easiest approach is to use metadata for the aggregate (LIR, Region, Resource, etc.). From the ProVision GUI, you can use the "Add Aggregate" function on the IPAM tab (image below) to add a duplicate aggregate. By assigning the new Aggregate to a Resource (#1) you will tie the block and a Resource together. You most likely wish to make assignments out of this new aggregate, so you can enable subassignments (#2) before adding the aggregate (#3).



Now the block will be assigned to a Resource, and you can still provision smaller blocks out of these duplicate parent blocks. When using the IPAM gadget from the Resource screen, just go the Advanced section and you will have more options available for assigning IP data. If you have "strict" enabled, your Smart Assign function should work as expected when you use the LIR or Assigned Resource fields for blocks.



Does ProVision support configuring a DNS server SOA that is different from the actual server name?

Yes - in DNS Admin, simple specify the DNS Servers like normal and then attach the zones as needed with the desired SOA.

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

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	Modified
IP-import-sample_v1.csv	Jul 30, 2013 by Christina Force
customer-import-sample.csv	Aug 07, 2013 by Christina Force
import-zone-assign.csv	Nov 08, 2013 by Christina Force
contact-import-sample_v1.csv	Dec 13, 2013 by Christina Force

Drag and drop to upload or browse for files

Download All

List of Abbreviations

List of Abbreviations:

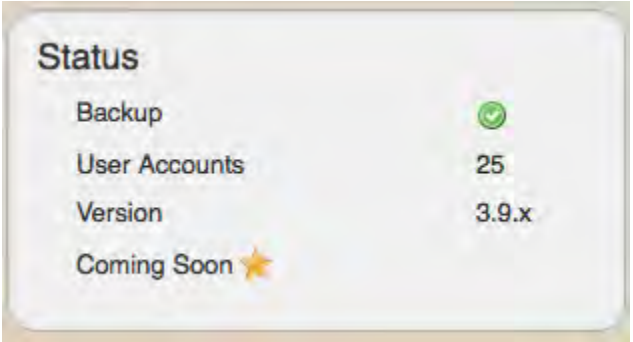
Edit Document

API	Application Program Interface
CLI	Command-line interface
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DNSSec	Domain Name System Security Extensions
IP address	Internet Protocol address
IPAM	Internet Protocol address management
LDAP	Lightweight Directory Access Protocol)
SDK	software development kit
SSH	Secure Shell

Abbreviation List.xlsx

Feedback and Feature Requests

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



You can also submit product feedback and feature requests to support@6connect.com