



Platform Documentation

ProVision

Application Version 4.2.x

Peering Manager

Application Version 2.0.x

Covering:

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

1. 6connect ProVision	4
1.1 Installation & System Requirements	4
1.1.1 Backup and Redundancy	5
1.1.2 CentOS Configuration Guide	6
1.1.3 General 6connect Installation Instructions	10
1.2 VM Install Guide - VMware Fusion	12
1.3 Getting Started	18
1.3.1 Concepts	19
1.3.1.1 Resources	19
1.3.1.1.1 Classes	19
1.3.1.1.2 Database Layout	21
1.3.1.1.3 The Asset System	24
1.3.1.2 IP Workflow	26
1.4 User Guide	26
1.4.1 Working with Resources	27
1.4.1.1 Customizing Sections	28
1.4.1.2 Customizing Fields	30
1.4.1.3 Gadgets	31
1.4.1.3.1 XML Specifications	34
1.4.2 DHCP Administration	35
1.4.3 6connect Peering v2	38
1.4.3.1 Peering Workflow	39
1.4.3.2 Peering - Common Tasks	39
1.4.3.2.1 Add Routers	39
1.4.3.2.2 Add Sessions	39
1.4.3.2.3 Import Sessions	40
1.4.3.3 Managing Peer Communications	41
1.4.3.4 Managing Peer Sessions	42
1.5 Admin Guide	43
1.5.1 Admin Preferences	43
1.5.2 Authentication Options	45
1.5.2.1 LDAP Authentication	46
1.5.2.2 RADIUS Authentication	47
1.5.3 IPAM Administration	51
1.5.3.1 Working with IP Blocks	52
1.5.3.2 IPAM Parameters	57
1.5.3.3 Holding Tank Management	59
1.5.3.4 LIR Management and Use	60
1.5.3.4.1 ARIN LIR Setup and Use	63
1.5.3.4.2 RIPE LIR Setup and Use	65
1.5.4 DNS Administration	67
1.5.4.1 Working with DNS Zones	70
1.5.4.1.1 Editing DNS Zones	71
1.5.4.2 Configuring ISC BIND Support	74
1.5.4.3 Configuring DynECT Support	74
1.5.4.4 Configuring PowerDNS Support	75
1.5.4.5 Configuring Secure64 Support	77
1.5.4.6 Configuring DNSSEC	82
1.5.4.7 Configuring Split Horizon/Views	83
1.5.4.8 Configuring DNS Templates	85
1.5.5 Importing Your Data	86
1.5.5.1 Import Aggregate Blocks	87
1.5.5.2 Import DNS Zones	89
1.5.5.3 Resource Import Tool	93
1.5.5.3.1 Resource Importer Walkthrough - Step 1	94
1.5.5.3.2 Resource Importer Walkthrough - Step 2	95
1.5.5.3.3 Resource Importer Walkthrough - Step 3	97
1.5.5.3.4 Resource Importer Walkthrough - Step 4	98
1.5.5.3.5 Resource Importer Walkthrough - Step 5	101
1.5.5.3.6 Resource Importer Walkthrough - Step 6	102
1.5.6 Permissions Overview	103

1.5.6.1 Global Permissions	103
1.5.6.2 Resource Permissions	104
1.5.6.3 Users and Groups	105
1.5.6.4 Verifying Permissions	107
1.6 Tutorials	108
1.6.1 Common Tasks	108
1.6.2 UI Tours	108
1.7 FAQ	109
1.8 List of Abbreviations	110
1.9 Feedback and Feature Requests	111
2. CLI (Alpha)	111
3. API v1	112
3.1 1 - Overview	113
3.2 2 - Making API Requests	113
3.3 3 - SDK - PHP	115
3.4 API Module - Admin and Audit	116
3.4.1 Authentication Testing	116
3.4.2 Log Management	118
3.4.3 Zone Templates	122
3.5 API Module - DHCP	126
3.6 API Module - DNS	153
3.7 API Module - IPAM	196
3.8 API Module - LIR	218
3.9 API Module - Peering	221
3.10 API Module - Resource	237
3.11 How Do I...	248
4. Release Notes	251
4.1 ProVision 4.2.0	251
4.1.1 ProVision 4.2.1	251
4.2 ProVision 4.1.0	253
4.2.1 ProVision 4.1.1	257
4.2.2 ProVision 4.1.2	257
4.2.3 ProVision 4.1.3	257
4.2.4 ProVision 4.1.4	258
4.2.5 ProVision 4.1.5	258
4.2.6 ProVision 4.1.6	258
4.2.7 ProVision 4.1.7	259
4.2.8 ProVision 4.1.8	259
4.2.9 ProVision 4.1.9	260
4.2.10 ProVision 4.1.10	261
4.2.11 ProVision 4.1.11	262
4.2.12 ProVision 4.1.12	262
4.2.13 ProVision 4.1.13	263
4.2.14 ProVision 4.1.14	264
4.2.15 ProVision 4.1.15	264
4.2.16 ProVision 4.1.16	264
4.2.17 ProVision 4.1.17	265
4.2.18 ProVision 4.1.18	266
4.2.19 ProVision 4.1.19	267
4.2.20 ProVision 4.1.20	269
4.2.21 ProVision 4.1.21	269
4.2.22 ProVision 4.1.22	271
4.2.23 ProVision 4.1.23	271
4.3 ProVision 4.0.0	271
4.4 ProVision 3.9.3	273
4.5 ProVision 3.9.2	273
4.6 ProVision 3.9.1	275
4.7 ProVision 3.9.0	276
4.8 ProVision 3.8.0	277
4.9 ProVision 3.7.0	277
4.10 ProVision 3.6.0	278

4.11 ProVision 3.5.0 278
4.12 ProVision 3.4.0 279
4.13 ProVision 3.3.0 279
4.14 ProVision 3.2.0 280
4.15 ProVision 3.0.0 281
4.16 ProVision 2.5 281

6connect ProVision

6connect - ProVision

Table of contents

- [Installation & System Requirements](#)
- [VM Install Guide - VMware Fusion](#)
- [Getting Started](#)
- [User Guide](#)
- [Admin Guide](#)
- [Tutorials](#)
- [FAQ](#)
- [List of Abbreviations](#)
- [Feedback and Feature Requests](#)

Installation & System Requirements

6connect Cloud Hosted Instance

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

Web Browsers Supported:

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

6connect Virtual Machine

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

Host Environment:

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

The minimum recommended virtual environment is:

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

Software Environment:

Operating System: FreeBSD

Port Requirements:

Open outbound ports 443 and port 80

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

6connect Locally Hosted Instance

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

Hardware Requirements:

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

The minimum recommended hardware is:

- Dual-core Xeon class processor or equivalent (Quad-core Xeon Recommended)
- 2GB RAM (4GB Recommended)

- Local SATA storage (local SAS/SSD or iSCSI/FC LUN optional)
- Rack mount server chassis with redundant power supplies

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

Software Requirements:

Operating System: Linux/BSD/OSX

Base Software Needed:

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



MySQL Triggers

6connect does not support custom MySQL triggers at this time - please email support@6connect.com if you have any questions.

Port Requirements:

Open outbound ports 443 and port 80

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

Backup and Redundancy

Backup and Redundancy

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

6connect Hosted Instance

Backup Schedule

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

Restoration

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

Local/VM Instance

Backup Schedule

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

Restoration

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



Backup your Data

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

mysqldump

And system folders off the 6connect root:

/scans

/zones

```
/keys
/archive
/data
```

CentOS Configuration Guide

Install and Configure MySQL

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

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

If its not installed:

```
yum install mysql-server
```



Service Startup

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

Set the mysql root password.

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

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

Install and Configure PHP

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

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

You should see something like php53.x86_64, php53-mysql.x86_64, php53-cli.x86_64 listed. If not:

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



PLEASE INSTALL

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

Install PCNTL

```
yum install php-pcntl
```

Install and Configure Apache and SSL



mod_rewrite REQUIRED

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

If SSL support is not already installed, install it:

```
yum install mod_ssl openssl
```

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

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

Copy the files to the correct locations

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



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

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

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

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

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

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

Add 443 virtual hosts as needed in httpd.conf.

Install and Configure Source Guardian

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

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

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

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

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

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

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

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

Configure SELinux



RE-IP WARNING

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

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

To view the SELinux configuration for http:

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

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

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

Configure IPTables

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

To list all current IPTable rules:

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

To add a rule for 443:

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



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

To save the new config:

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

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

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

With the file open for editing, add:

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

Once complete - restart the iptables service:

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



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

Install Radius

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

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

Install 6connect

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

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

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

Follow the instructions for 6connect installation.

Suggested follow up items for all customers

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

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

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

General 6connect Installation Instructions

Apache Configuration Requirements

 **mod_rewrite and mod_ssl are required**

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

 **ssl.conf**

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

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

Apache 2.2:

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

Apache 2.4:

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



AllowOverride

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

MySQL Configuration

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

PHP Configuration



PHP Compatibility

Please note that as of version 5.0.0 of ProVision, PHP versions >5.5 are required

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

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

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

```
session.bug_compat_warn=0
```

SMTP = localhost

smtp_port = 25



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

```
extension=radius.so
```

```
extension=ssh2.so
```

Source Guardian

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

extension=ixed.5.x.xxx

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

Additional PHP Extensions

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

Additional System Packages

memcache
memcached
openssl
cURL
nmap
sendmail (Or any mail software. The correct binary should be specified in php.ini)

DNS Tools and Packages

named-checkzone
rndc
zonesigner
dnssec-dsfromkey

NEW INSTALLATION

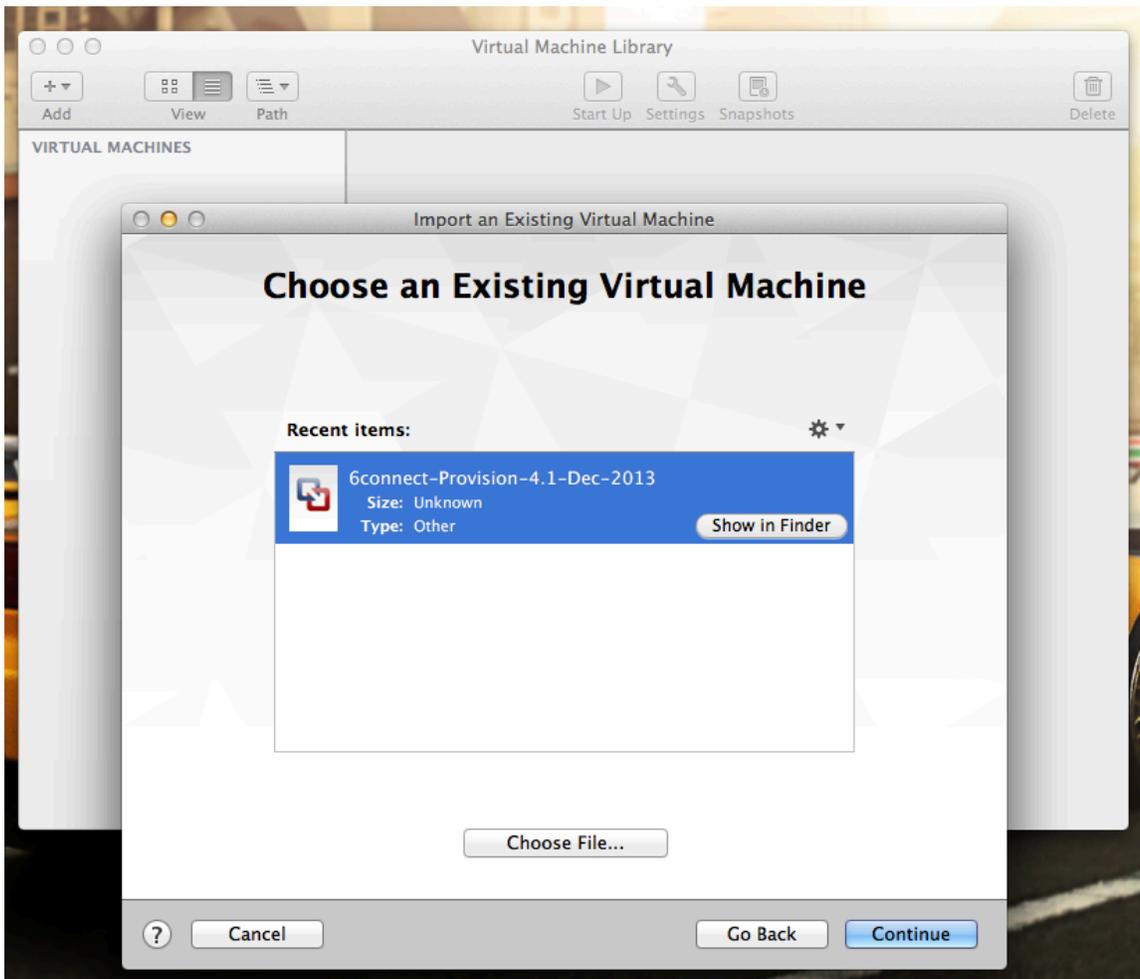
-
1. Install all the packages, extensions, and perform configuration listed above and the Source Guardian extension. To install the Source Guardian extension:
Download the correct Source Guardian loader for your OS/php version from: <http://www.sourceguardian.com/ixeds/>
Place the file in your php extension directory as specified in your php.ini
Add extension=ixed.x.x.y.y to your php.ini
 2. Move the tar file in 6connect web root.
tar -xof productionBuild-4.0.3.tar
This will place all the new files into your web root directory.
 3. Go to <http://<web root>/configTest.php>. If there are any configuration errors listed in red, other than in the Database and Configuration Files section, they must be corrected.
 4. Click on Setup Wizard or go to <http://<web root>/configBootstrap.php>. Fill in the requested information. Permissions on your web directory may cause the automatic setup of directories to fail. If there are still permissions issues, run configDir.sh from the command line. If there are any other errors listed, follow the instructions to complete these items manually.
 5. Carefully note the login credentials provided before continuing.
 6. Log in and use!

VM Install Guide - VMware Fusion

Step 1 - Download the OVF* tar file

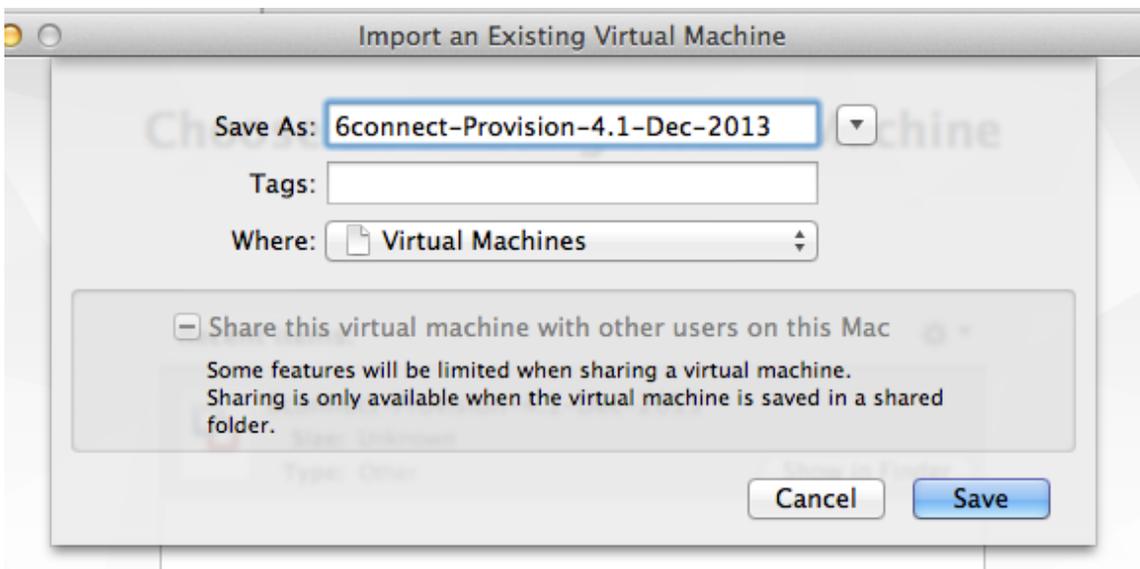
(VMDK is also available)

Select the option to Import an Existing Virtual Machine



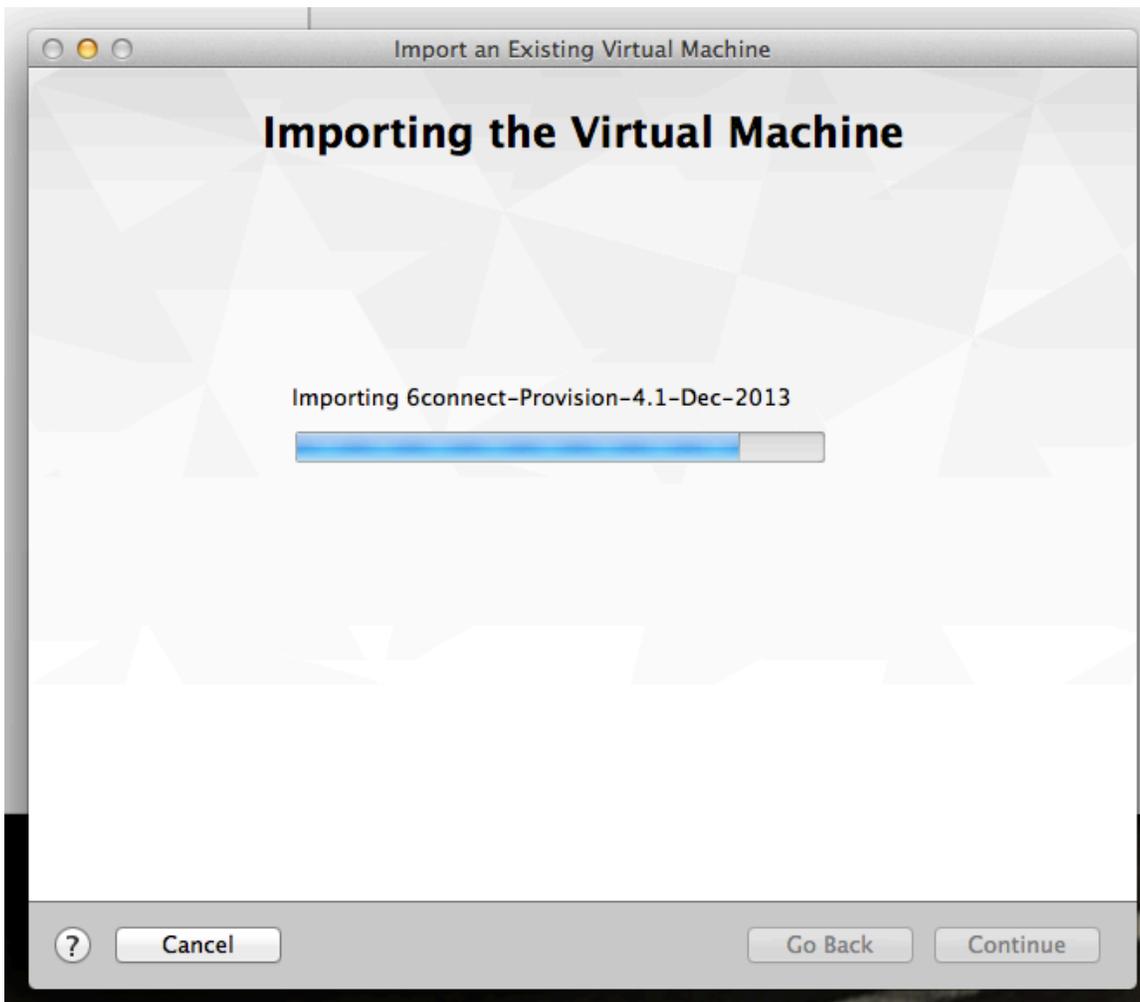
Step 2 - Select the VM you wish to import

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



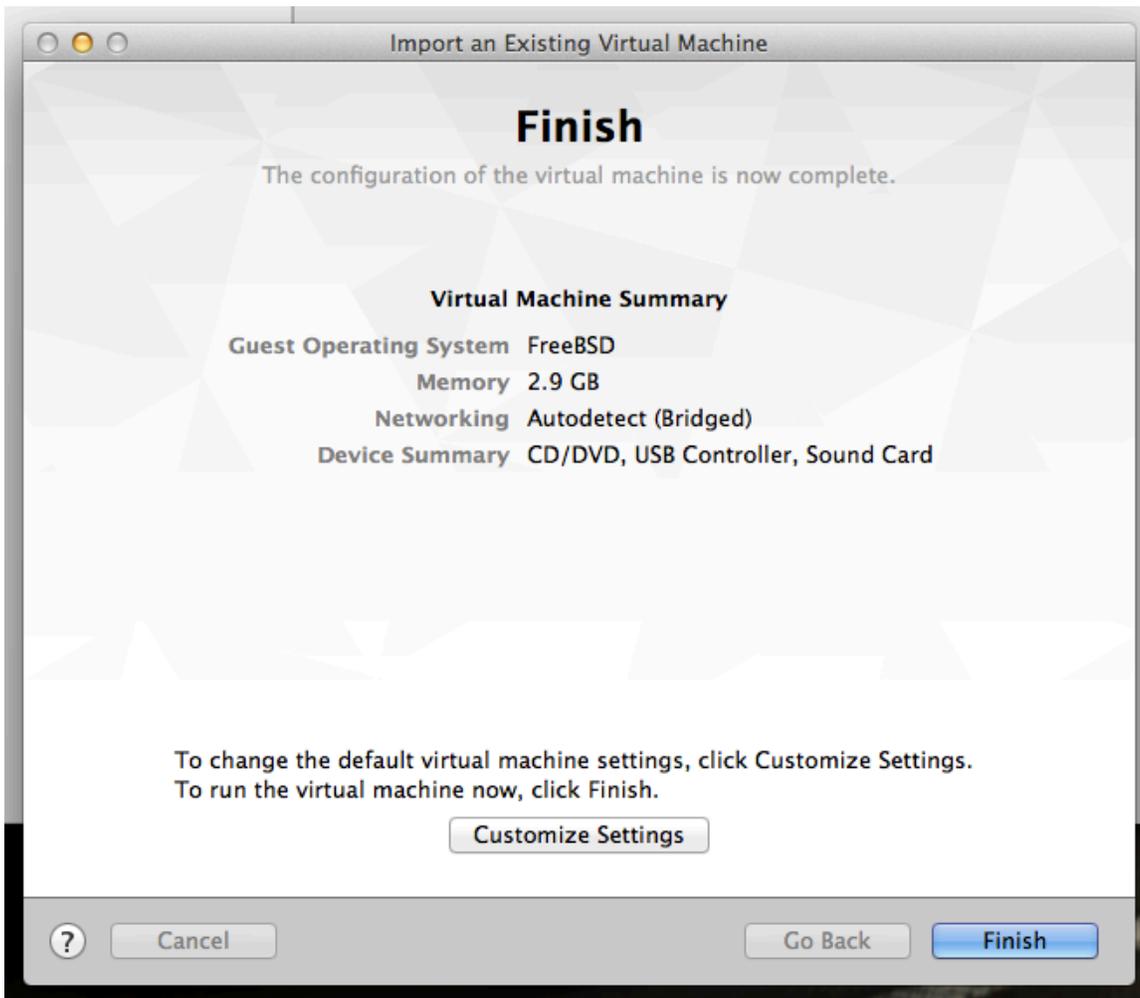
Step 3 - Import the VM image

This will take a few minutes.



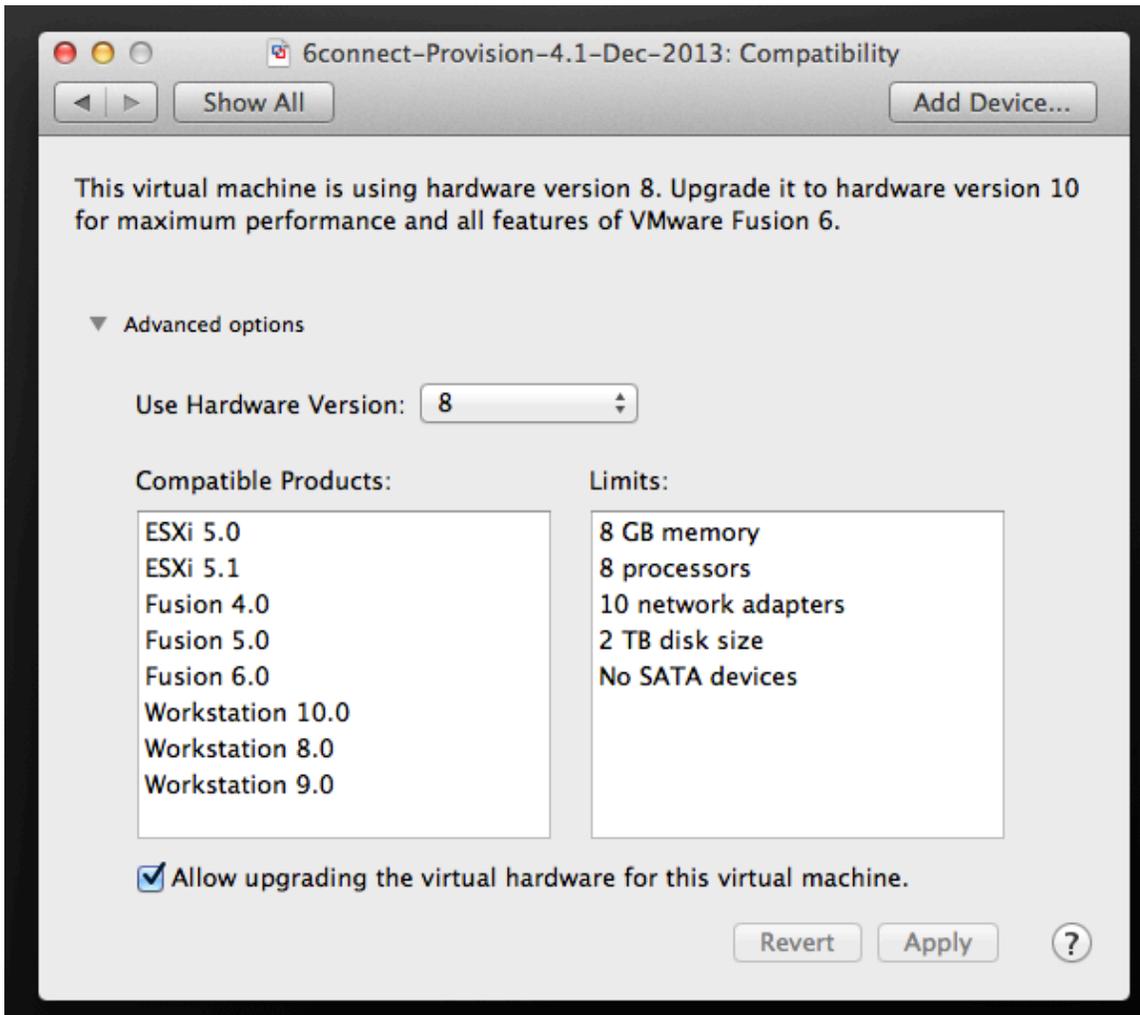
Step 4 - Confirm the VM import configuration

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



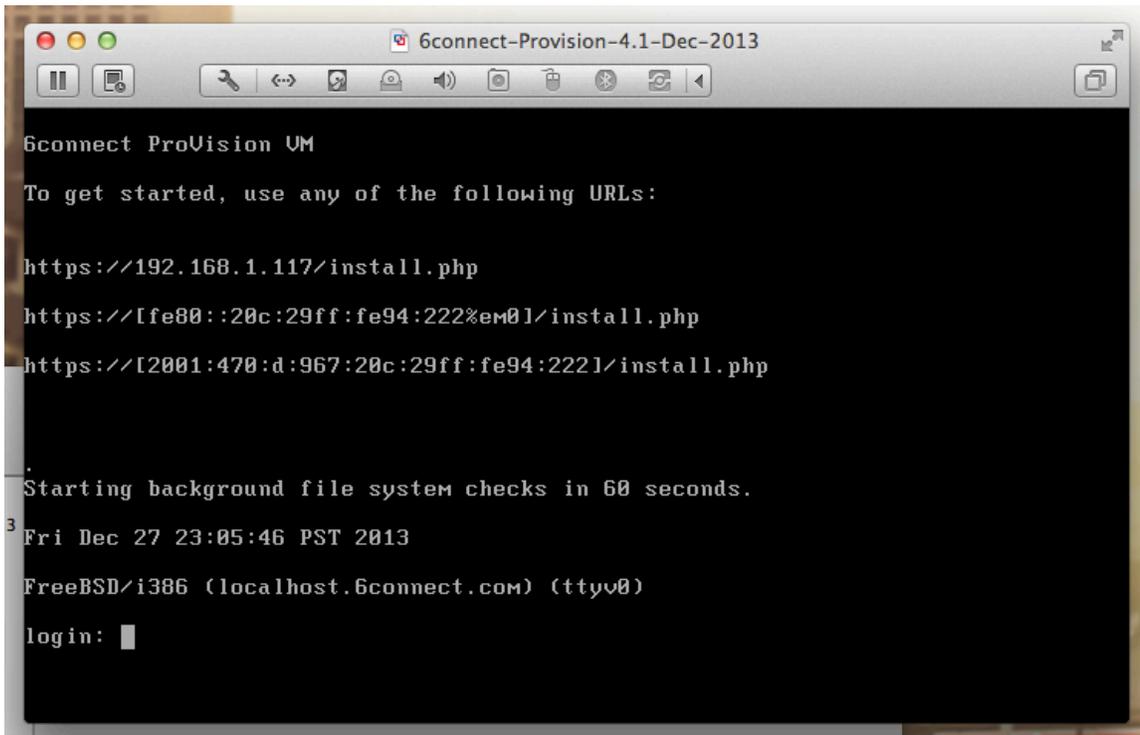
Step 5 - Optional - Modify VM configuration properties

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



Step 6 - Setup ProVision environment

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



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

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

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

Step 7 - Run ProVision Setup Wizard

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

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

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

Setup Wizard - 4.1.8

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

Database and Configuration Files

MySQL Configuration

MySQL Found
MySQL Version Version requirements met.

Required Information

Database Information

Database User
Database Password
Database Name
Database Host

General Information

Company Name
Registration Email
Install URL
Install Directory

[Create Defaults](#)

Directory/File Ownersip and Permissions

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

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

Step 8 - Use ProVision

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

Setup Complete!

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

Getting Started

Getting Started

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

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

[Step 1 - Resources Overview](#)

[Step 2 - Admin Preferences](#)

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

Step 4 - IPAM Administration

Step 5 - DNS Administration

Step 6 - DHCP Administration

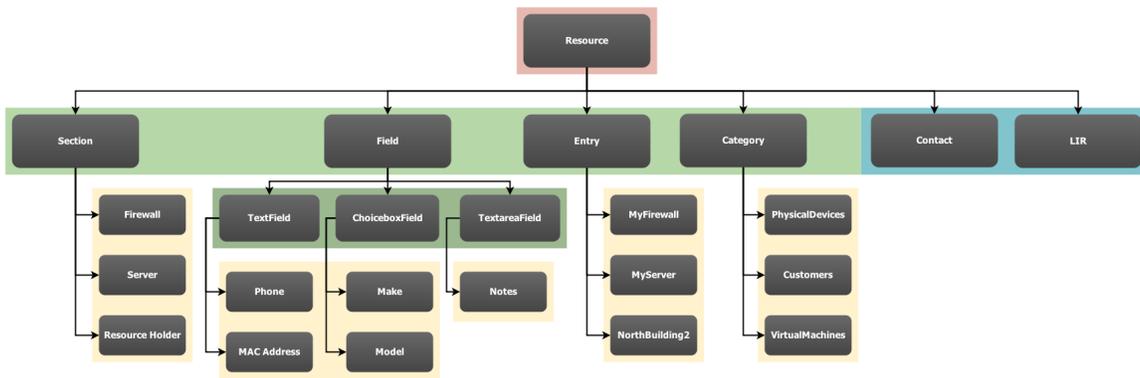
Step 7 - Importing Data

Concepts

Resources

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.

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
 - Properties
- Examples
 - 1 - PHP
 - 2 - API request

class Resource

```

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

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

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

```

Properties

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

```

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

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

```

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

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

Examples

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

1 - PHP



Internal code example

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

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

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

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

2 - API request

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

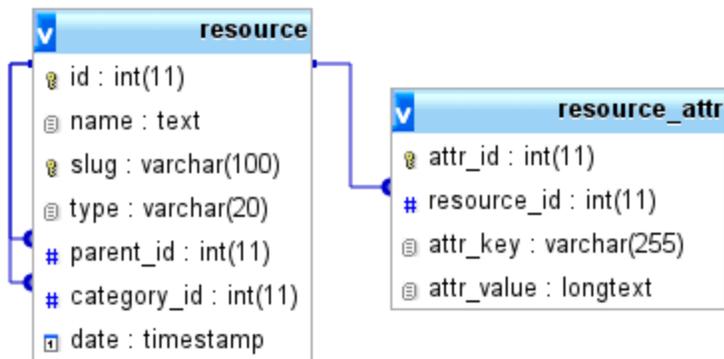
/api/v1/api.php?target=resource&action=get&slug=TLR

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

Database Layout

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

Figure



Relations

``resource`.`category_id` -> `resource`.`id``

``resource`.`parent_id` -> `resource`.`id``

``resource_attr`.`resource_id` -> `resource`.`id``

Structure in SQL

resource

[Expand source](#)

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

resource_attr

› Expand source

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

The Asset System

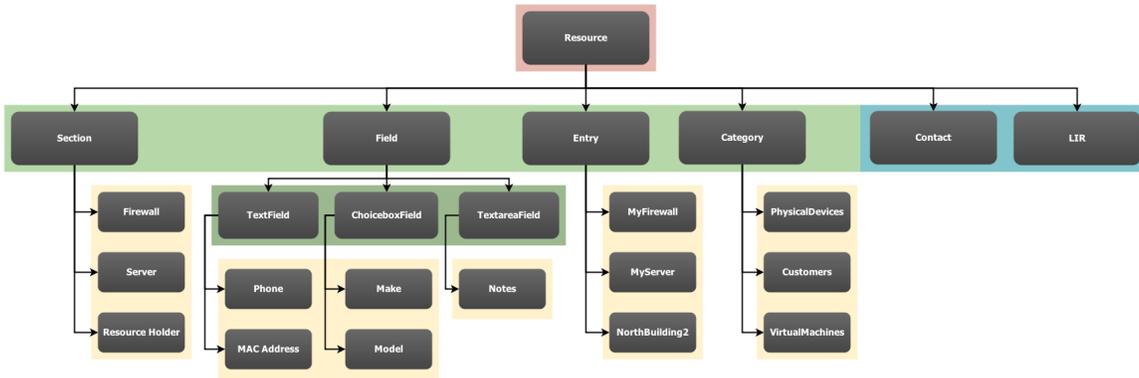
Prerequisites

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

Overview

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

have been instantiated from their Class. And the items in blue are examples of resource child-classes (resource types) that exist outside of the asset system.



Introduction

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

Components

Section

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

Entry

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

Field

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

Fields also have a use beyond acting as properties for classes. The field object (in this case *color*) is a resource object in it's own right. This means it can be modified independently of the sections that have assigned it and the entries that are using it. For example, a field which shows a dropdown box of several options could be modified to include more options; any entry which is using that field would automatically receive those new changes. Or consider a simple textfield object called "MAC Address" that is used by several sections and entries. If that field was modified to include a filter that checks the input for a valid MAC string, any entry using that field would get those improved validation checks.

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

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

Category

Categories are just an organisational tool. There is a clearly defined relationship between Sections, Entries, and Fields, but Categories exist on their own. If you look on the [Classes page](#), you'll see that every Resource has the same 6 fundamental properties and 3 of them are ID values. The first is the ID that belongs to the resource itself, the second is the ID of the resource's parent, and the third is the ID of the Category that the

resource belongs to (if any). There isn't a strict hierarchy here, how you use categories is entirely up to you. You can create categories, child categories, and careful plan exactly how you want the resources in your system to be organised. Or you can ignore the whole thing completely and just let every resource have the default category of "uncategorized." Many 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.

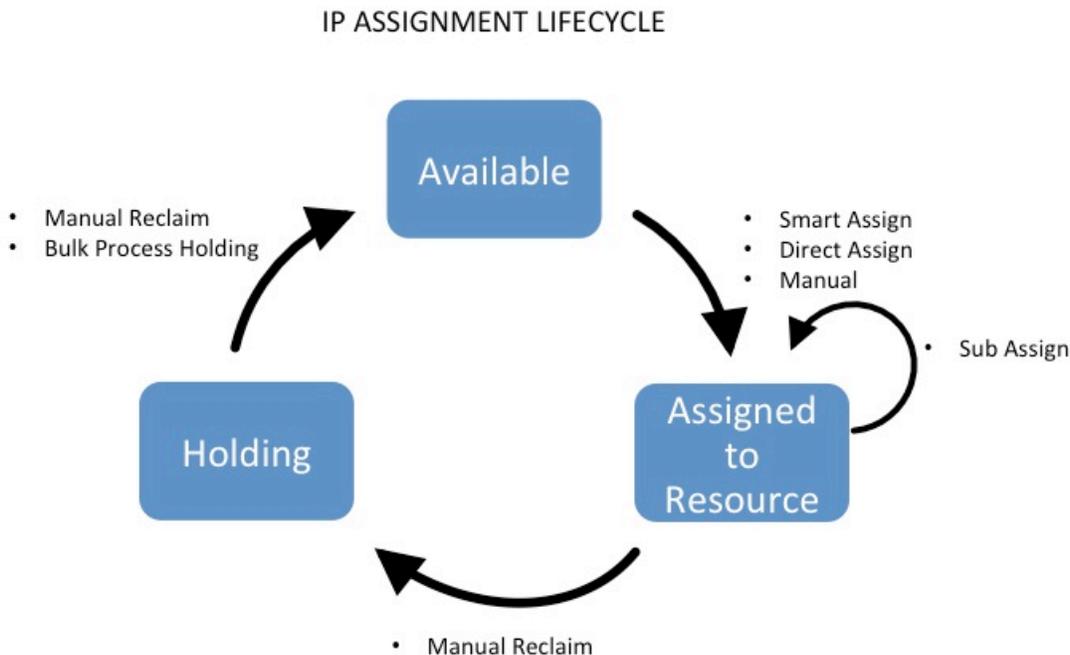
Gadgets

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, I would recommend emailing us first with an outline of what you're trying to do. Then my recommended procedure would be to first create it as a standalone HTML/Javascript webpage that connects to our API (you may need to disable cross domain request security in your browser to make the AJAX connections work). Once you have your standalone page working, the process to turn that into an embeddable gadget is trivial.

Note: Gadgets are initialized as AngularJS applications. Both the AngularJS and jQuery libraries will be loaded on the page and available to use, but it is highly 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.

IP Workflow



User Guide

Step 1 - Resources Overview

Step 2 - Admin Preferences

Step 3 - User & Group Permissions

Step 4 - IPAM Administration

Step 5 - DNS Administration

Step 6 - DHCP Administration

Step 7 - Importing Data

Working with Resources

What is a Resource?

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



WARNING!

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

How to Work with Resources?

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

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

Resources / Entries / Add Entry

Fundamentals

Name (required)
Some Resource

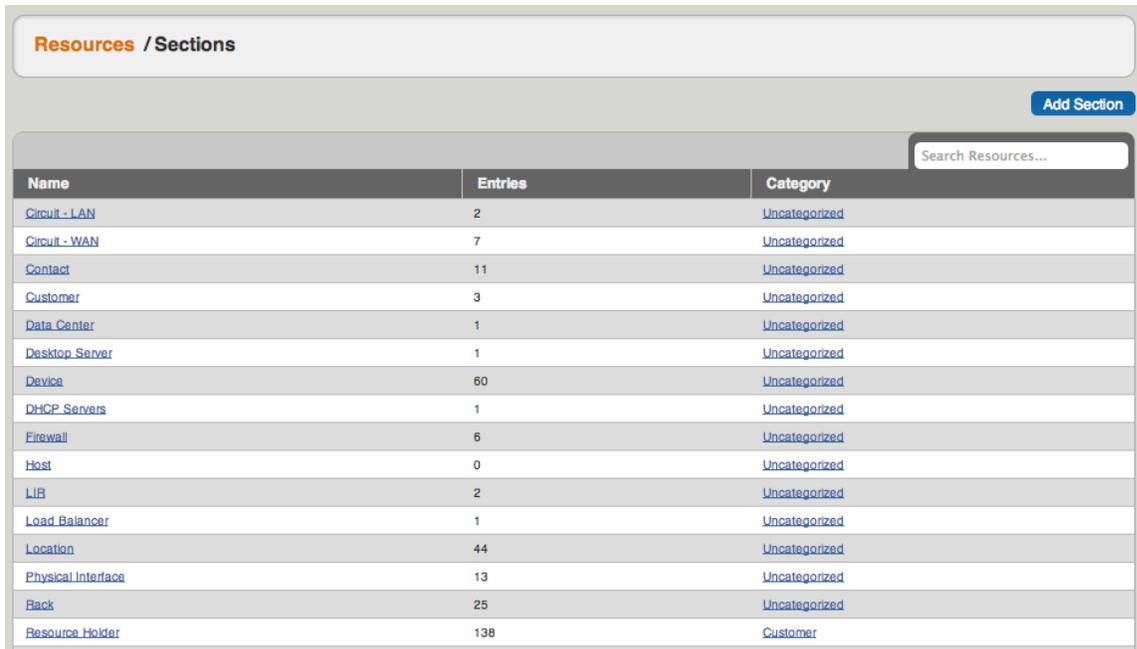
Parent
TLR

Type
Resource Holder

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

- **Resource 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 [Cu](#)

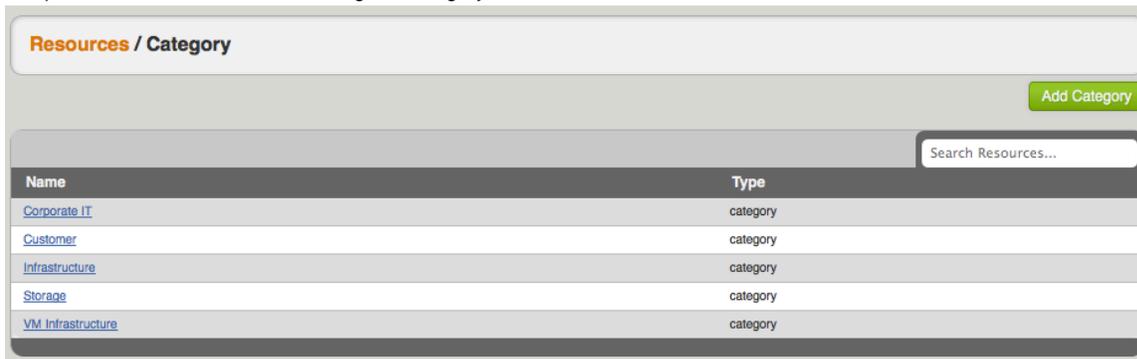
stomizing Fields for more details on how to fit these elements to your business.



The screenshot shows a web interface titled "Resources / Sections". It features a search bar labeled "Search Resources..." and an "Add Section" button. Below is a table with three columns: Name, Entries, and Category. The table lists various resource types and their counts, with most categories set to "Uncategorized".

Name	Entries	Category
Circuit - LAN	2	Uncategorized
Circuit - WAN	7	Uncategorized
Contact	11	Uncategorized
Customer	3	Uncategorized
Data Center	1	Uncategorized
Desktop Server	1	Uncategorized
Device	60	Uncategorized
DHCP Servers	1	Uncategorized
Firewall	6	Uncategorized
Host	0	Uncategorized
LIB	2	Uncategorized
Load Balancer	1	Uncategorized
Location	44	Uncategorized
Physical Interface	13	Uncategorized
Rack	25	Uncategorized
Resource Holder	138	Customer

- **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".



The screenshot shows a web interface titled "Resources / Category". It features a search bar labeled "Search Resources..." and an "Add Category" button. Below is a table with two columns: Name and Type. The table lists several categories, all of which are of type "category".

Name	Type
Corporate IT	category
Customer	category
Infrastructure	category
Storage	category
VM Infrastructure	category

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

Some examples:

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

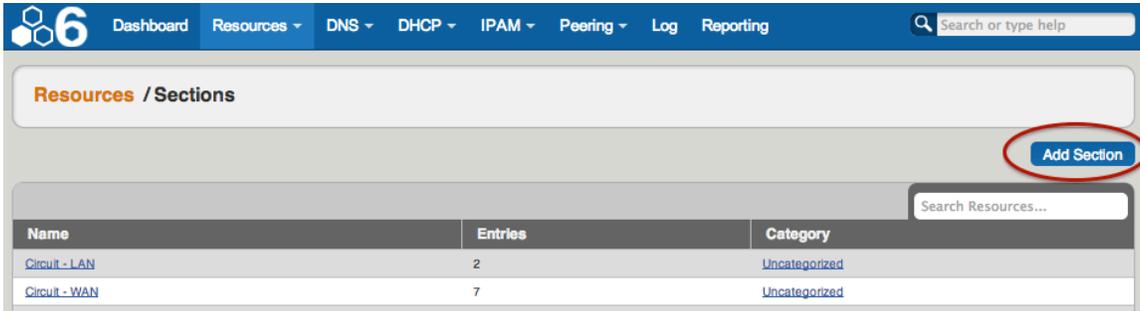
Customizing Sections

(video overview to be developed)

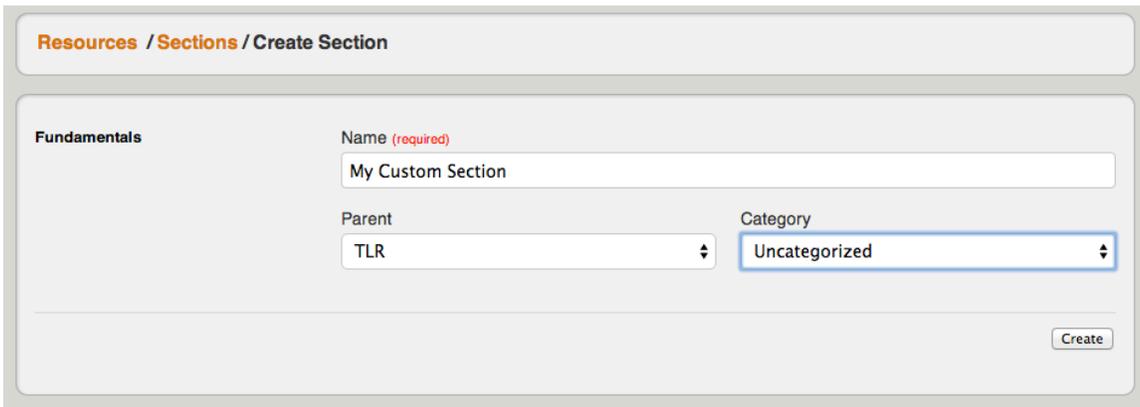
You can create as many Sections as you wish (Firewall, Server, VM, Virtual Interface, etc.) and customize the fields that you care about for each Section. For example, you may not need to track the console port for your virtual firewall, so you would simply not use that field for the "Virtual Firewall"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

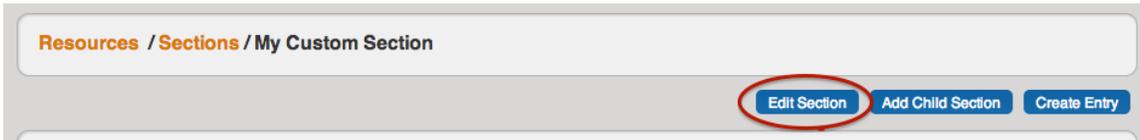


Create a new Section by specifying a Name, Parent, and Category

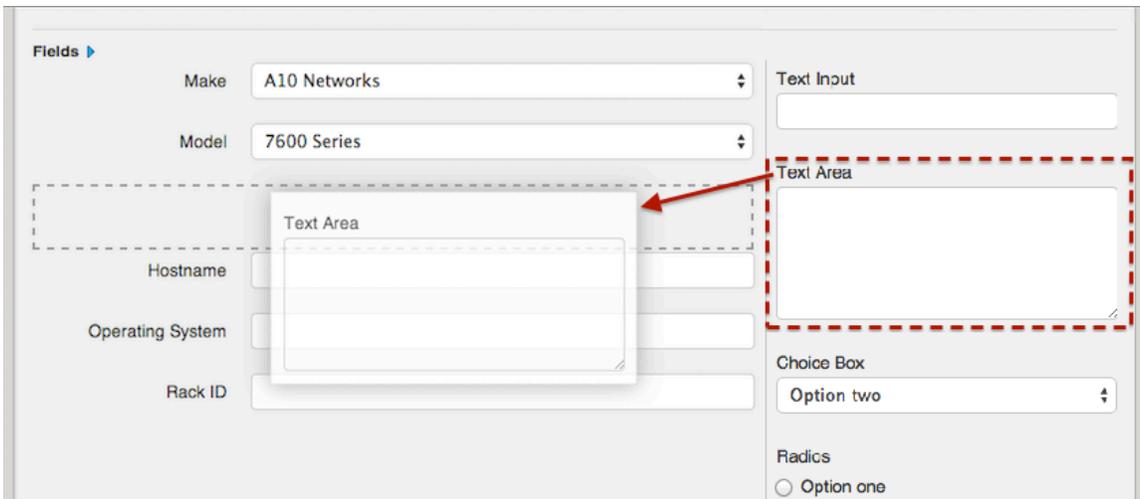


Step 2: Add a Custom Field to a Section

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



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



Step 3: Edit Customize Field Data

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

Parent: TLR

Category: Ur

Fields

- Asset Tag
- Make: A10 Networks
- Model: 7600 Series
- Hostname
- Operating System
- Rack ID

Name: Model (Open in new tab)

Help Text

Options

- 7600 Series
- ASR 9000 Series
- ASR 1000 Series
- 6500 Series
- T Series

Slug: 6c-model

This is a protected System Field. Only 'options' and 'help block' can be changed.

Save Cancel

Step 4: Add Gadgets to your Section

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

Manage Gadgets

Gadgets are mini applications that can add extra features to your resources.

Delete

Gadgets

- Contact Info
- DNS
- IPAM
- Resource View
- Tech Info
- Test

Add Save

Customizing 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.

Add Existing Fields

It is preferable to reuse existing fields whenever possible. This improves pattern matching between resources.

Existing Fields

- 2nd Email (6c-contact-email)
- 2nd Email (6c-contact-email2)
- 2nd Phone (6c-contact-phone2)
- A New Field (some-text)

Add field

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

The screenshot shows a form editor interface. On the left, under the 'Fields' section, there are several fields: 'Make' (A10 Networks), 'Model' (7600 Series), 'Hostname', 'Operating System', and 'Rack ID'. A 'Text Area' field is highlighted with a red dashed box, and a red arrow points to it from the right. On the right side of the editor, there are other field types: 'Text Input', another 'Text Area', 'Choice Box' (Option two), and 'Radios' (Option one).

Editing/Removing Fields

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

The screenshot shows a form editor interface. On the left, under the 'Fields' section, there are several fields: 'Asset Tag', 'Make' (A10 Networks), 'Model' (7600 Series), 'Hostname', 'Operating System', and 'Rack ID'. The 'Model' field is highlighted with a blue border. On the right side, there is a 'Category' panel with a dropdown menu set to 'Ur'. Below the dropdown, there are fields for 'Name' (Model), 'Help Text', and 'Options' (7600 Series, ASR 9000 Series, ASR 1000 Series, 6500 Series, T Series). There is also a 'Slug' field (6c-model) and a note: 'This is a protected System Field. Only 'options' and 'help block' can be changed'. At the bottom of the panel are 'Save' and 'Cancel' buttons.

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

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

Gadgets

Gadgets

- What are Gadgets
- Available Gadgets
 - Resource View
 - Contact Info
 - Tech Info
 - IPAM
 - DNS
 - DHCP
 - Peering Session
 - Peer Groups

What are Gadgets

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

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

Available Gadgets

Resource View

This visual element is used on the Resource Holder Section type.

Some Customer (cust1)

Type: **Resource Holder**
Category: **Customer**

Contact Info

This visual element is used on the Resource Holder Section type.

Contact Info [edit](#)

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

Tech Info

This visual element is used on the Resource Holder Section type.

Tech Info [edit](#)

DNS Servers

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

ARIN Info

Org ID:	ARIN-ORGNAME	Org POC:	ARIN-POC1
Net POC:	ARIN-POC2	Abuse POC:	
Origin AS:	23456		

Residential Customer Privacy: Disabled

IPAM

This visual element is used on the Resource Holder Section type.

IPAM IPv4 IPv6

Assign Block:

Browse To Assign Direct Assign
List available blocks: x.x.x.x/yy

Smart Assign
RIR Region Size Tags...

Filter:
Notes/CIDR... RIR Region Size Some Customer Tags... **Filter** **Clear**

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

DNS

This visual element is used on the Resource Holder Section type.

DNS

New DNS Zone **Create Zone**

Zone Delegation

Delegated Zone	Slave IP	Customer
Zone name	IPv4 or IPv6	84

Add Slave

Zone Records Tags Entries

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

DHCP

This visual element is used on the Server Section type.

The DHCP Management Gadget in the "Off" configuration:

DHCP Management

DHCP Services: Off On **Update**

The DHCP Management Gadget in the "On" configuration:

DHCP Management

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

Save Configuration **Push Configuration**

Peering Session

This visual element is used on the Router Section type.

Peering Sessions

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

Peer Groups

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

Peer Groups

Add Peer Group: IPv4 IPv6

Exchange	Peer Group	Type

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

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

Peer Groups

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

Add Peer Group: IPv4 IPv6

Exchange	Peer Group	Type	
Equinix Palo Alto	Peer Group 2	ipv4	<input type="button" value="delete"/>



Note

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

XML Specifications

XML Specifications

THIS IS AN EXPERIMENTAL FEATURE

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

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

XML Specification

The XML gadget specification is based on the Atlassian Gadgets.

Implemented Tags

The implemented tags and corresponding attributes are:

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

Example

```

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

```

Fields

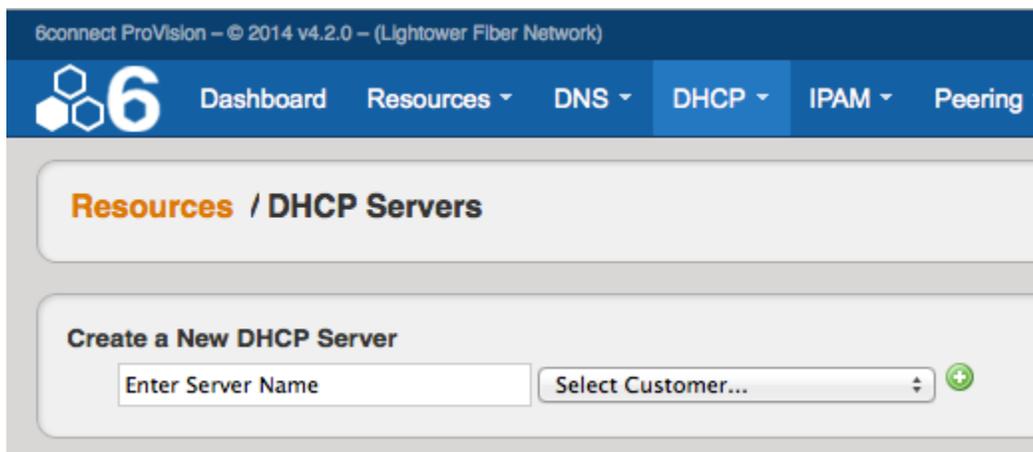
If a gadget uses fields, you can optionally add the slug of the field in this section to hide it from the main field list.

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

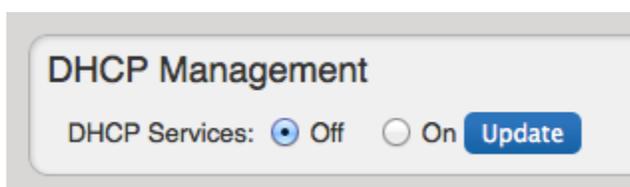
DHCP Administration

Adding DHCP Servers

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



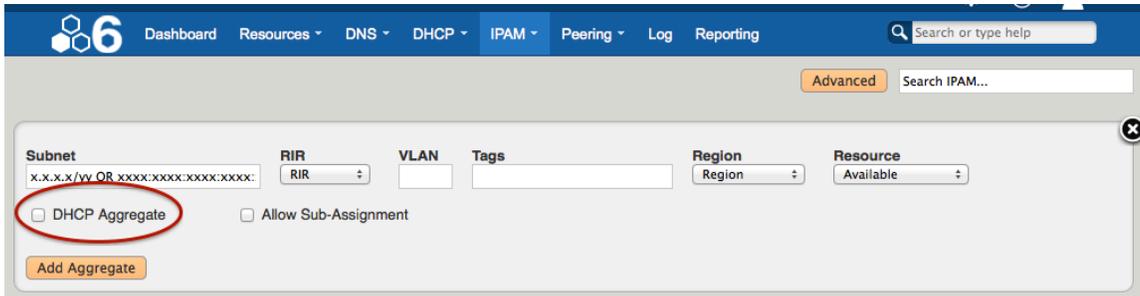
If DHCP is enabled for a Resource Section, you can also use the DHCP toggle function to enable this functionality.



Defining DHCP Scopes

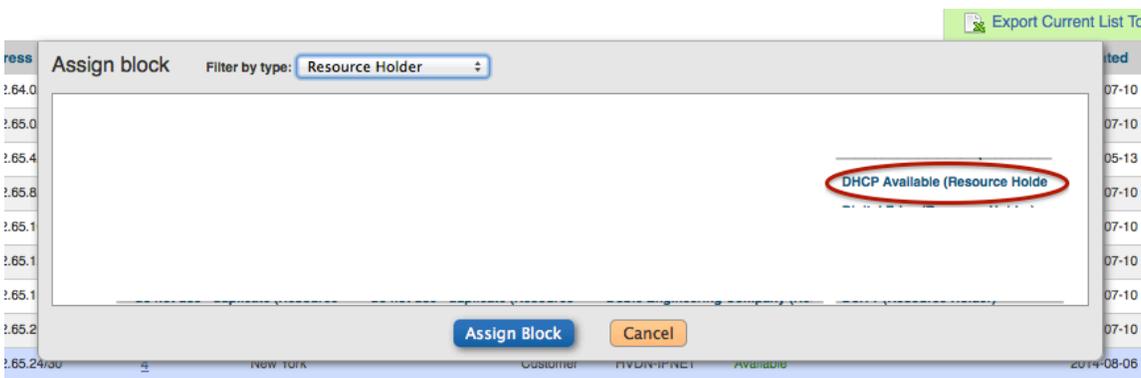
In order to use DHCP functions and add DHCP Pools, the IP blocks need to be defined in the IPAM section.

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



The screenshot shows the 'Add Aggregate' form in the IPAM section. The 'Subnet' field is highlighted with a red circle. Below it, the 'DHCP Aggregate' checkbox is also highlighted with a red circle. Other fields include RIR, VLAN, Tags, Region, and Resource.

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

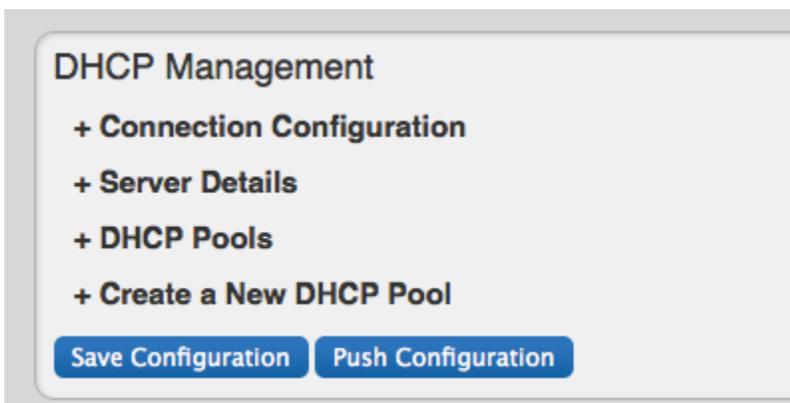


The screenshot shows the 'Assign block' dialog box. The 'Filter by type' dropdown is set to 'Resource Holder'. The 'DHCP Available (Resource Holder)' option is highlighted with a red circle. The 'Assign Block' button is visible at the bottom.

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

Managing DHCP Server Configurations

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



The screenshot shows the DHCP Management gadget. It shows a list of options: '+ Connection Configuration', '+ Server Details', '+ DHCP Pools', and '+ Create a New DHCP Pool'. At the bottom, there are two buttons: 'Save Configuration' and 'Push Configuration'.

Connection Configuration

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

+ Connection Configuration

Manual IP:

Notes:

SSH

Username:

Password:

Port:

Server Details

+ Server Details

DHCP Vendor:

DHCP Config File Path:

Server Options

Routers:

Domain Name Servers:

Domain Name:

Free Lines (appended to DHCP Server Config):

No lines saved.

Add a new Line:

Add

Server Commands

Config Test:

Server Stop:

Server Start:

Advanced Options

Authoritative:

Default Lease Time:

Max Lease Time:

Local Port:

Log Facility:

DHCP Pools

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

+ DHCP Pools

Linked Pools

Internal Lab - VM [Subnet]

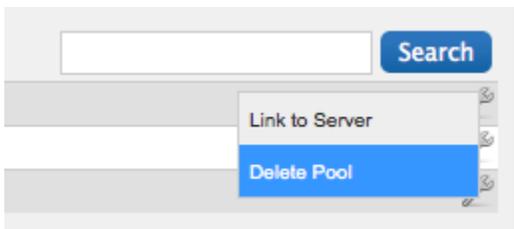
Existing Pools

Internal Lab - VM [Subnet]

LargerTest [Subnet]

TsetNet [Subnet]

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



Create a New DHCP Pool - Subnets

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

+ Create a New DHCP Pool

Create a new **Subnet** ▾

Subnet Name: (ex: Lab #1)

New IP Assignment: Smart ▾ IPv4 ▾ Mask ▾ RIR ▾ Region ▾

Free Lines:
No lines saved.

Add a New Line: **Add**

Add Pool

Create a New DHCP Pool - Host

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

+ Create a New DHCP Pool

Create a new **Host** ▾

Hostname: (ex: 6connect.com)

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

New IP Assignment: Smart ▾ IPv4 ▾ RIR ▾ Region ▾

Free Lines:
No lines saved.

Add a New Line: **Add**

Add Pool

Saving/Pushing DHCP Server Configurations

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



6connect Peering v2

6connect Peering

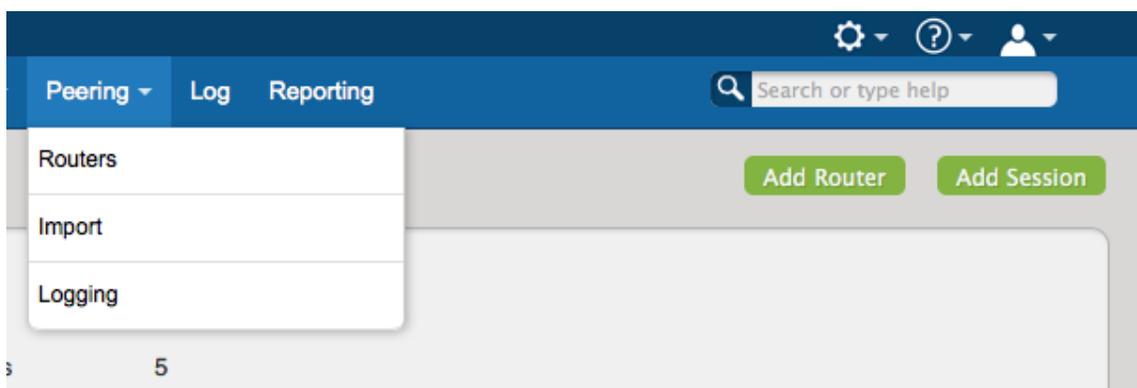
The **Peering** tab displays peering stats, allows you to add routers and sessions, and to manage communications and sessions for each exchange.

Three other sections are available via the drop down menu:

Routers - Links to the resource list of routers

Import - Import Sessions by exchange and router

Logging - View peering related logs



Peering Workflow

Peering Workflow

(Typical Workflow order in progress)

Peering - Common Tasks

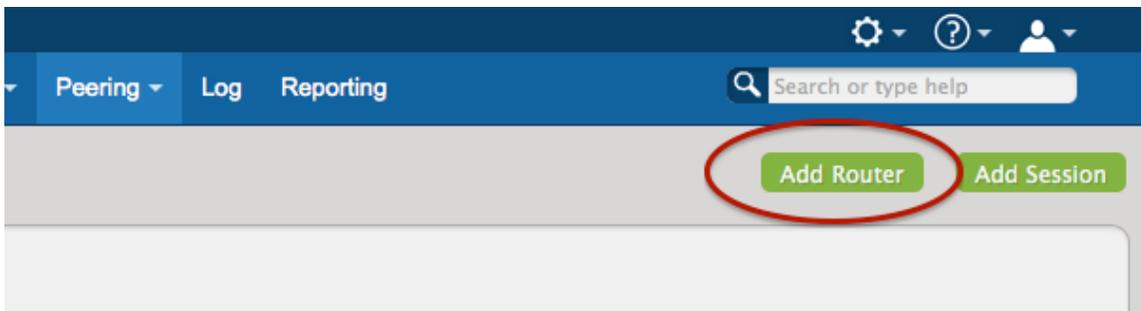
Peering Common Tasks

- Add Routers
- Add Sessions
- Import Sessions

Add Routers

Adding Routers

Navigate to the [Peering](#) tab. Select "Add Router".



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

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

A screenshot of a modal window titled "Add Router". The form contains the following fields:

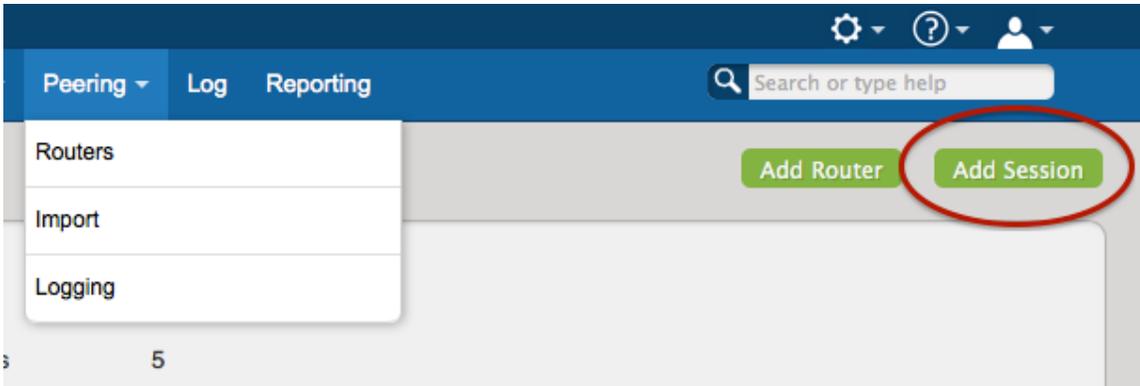
- Parent Resource: TLR (dropdown)
- Exchange: Equinix Internet Exchange (dropdown)
- Name: (text input)
- Make: A10 Networks (dropdown)
- Model: 7600 Series (dropdown)
- Hostname: (text input)
- IPv4 Address: (text input)
- IPv6 Address: (text input)
- Username: (text input)
- Password: (text input)
- Peer Group: (text input)
- Type: IPv4 IPv6

Below the form, there is a table with columns "Exchange", "Peer Group", and "Type". The table is currently empty, with the text "No groups specified" below it. An "Add Group" link is visible to the right of the "Type" field. At the bottom right of the modal, there is a green "Add Router" button.

Add Sessions

Adding a Peering Session

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



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

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.

A screenshot of the 'Add Session' form. The form is titled 'Add Session' and contains several sections:

- Type:** A dropdown menu set to 'Peer'.
- Exchange:** A dropdown menu set to 'Equinix Internet Exchange'.
- Note:** A text area for additional information.
- Peer Group:** A dropdown menu set to 'Select Peer Group...'.
- MD5:** A text input field.
- Max Prefixes:** A text input field.
- Source:**
 - ASN:** A dropdown menu set to 'AS8038'.
 - Router:** A dropdown menu set to 'Select Router...'.
 - IP Address:** A text input field.
- Destination:**
 - A sub-section titled 'Select peer and public IP data PeeringDB or specify custom data for the session.' containing:
 - Peer:** A dropdown menu set to 'Amazon.com'.
 - Public IP:** A dropdown menu set to 'AS16509 - 198.32.176.36'.
 - Peer:** A text input field containing 'Amazon.com'.
 - ASN:** A text input field containing '16509'.
 - IP Address:** A text input field containing '198.32.176.36'.

At the bottom left, there is a checkbox labeled 'Configure router after saving?'. At the bottom right, there is a green 'Save' button.

Import Sessions

Importing Sessions

Navigate to the **Peering Tab**. In the dropdown menu, select **Import**.

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

Exchange Router

Type	Source ASN	Source IP Address	Peer	Peer ASN	Peer IP Address	Group
No router selected						

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

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

Exchange Router

Importing sessions from lab2-cisco (50.240.195.132) at Equinix Palo Alto.

13 sessions found. 0 already imported or added.

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

The next step is to configure and manage your sessions.

Managing Peer Communications

Communications Manager

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

Equinix Ashburn - Ashburn, US (206.223.115.0/24)

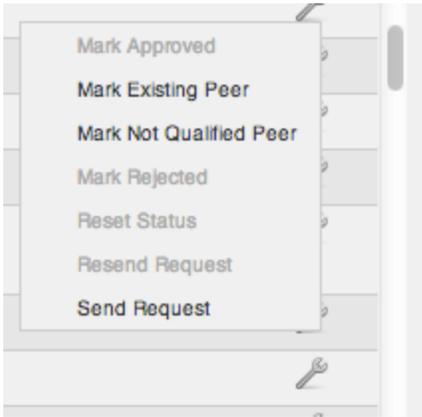
Current Peers: 6	Rejected Requests: 1	Sessions Tracked: 18
Qualified Peers: 183	Pending Requests: 0	Sessions Up:
Not Qualified Peers: 0	Most Recent Request:	Peers Without Sessions: 177
Most Recent Peer: Akamai Technologies - 07/25/2014		

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**.

Communications - Equinix Palo Alto

Peer	ASN	Name	Request	Notes	
<input checked="" type="checkbox"/>	AS15145			2014-09-24 – Session updated: (AS8038/50.240.195.135) - (AS15145/1.2.3.15) 2014-09-22 – Session added: (AS8038/50.240.195.135) - (AS15145/1.2.3.15)	
	AS7575	AARNet			
	AS9264	Academia Sinica Network(ASNet)			
<input checked="" type="checkbox"/>	AS20940	Akamai		2014-09-22 – Session added: (AS8038/50.240.195.135) - (AS20940/206.126.236.102) 2014-09-22 – Session deleted: (AS8038/50.240.195.135) - (AS20940/206.126.236.102)	
	AS20940	Akamai Technologies			

Select the wrench icon to manage the communication status:



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

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

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

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

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

Resend Request: Resends the peering request.

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

Managing Peer Sessions

Managing Peering Sessions

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

Equinix Ashburn - Ashburn, US (206.223.115.0/24)

Current Peers: 6	Rejected Requests: 1	Sessions Tracked: 18
Qualified Peers: 183	Pending Requests: 0	Sessions Up:
Not Qualified Peers: 0	Most Recent Request:	Peers Without Sessions: 177
Most Recent Peer: Akamai Technologies - 07/25/2014		

Communications
Sessions

The Peering Manager UI:

BGP Sessions - Equinix Palo Alto ①

Filter by: Peer ▾ Source ASN ▾ Destination ASN ▾ IP Type ▾ Session Type ▾ State ▾ Filter Clear Filters ② Add Session

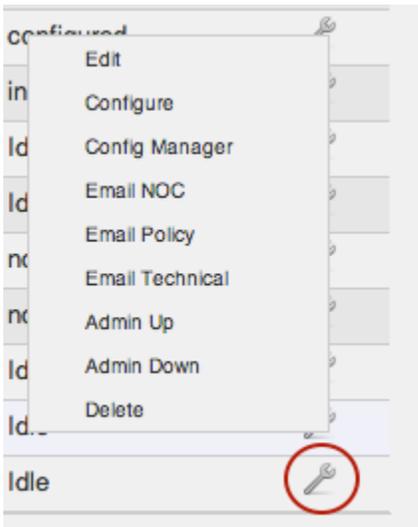
Source	Router	Peer	Destination	Peer Group	Type	Prfx Rcvd/Max	State	Notes
AS8038 - 50.240.195.135	lab1-cisco		AS15141 - 1.2.3.10		Peer	0/0	Idle	⚙️
AS8038 - 50.240.195.135	lab1-cisco		AS15142 - 1.2.3.11		Peer	0/0	Idle	⚙️ ④
AS8038 - 50.240.195.135	lab1-cisco		AS15144 - 1.2.3.14		Peer	0/0	Idle	⚙️
AS8038 - 50.240.195.135	lab1-cisco		AS15145 - 1.2.3.15		Peer	0/0	configured	⚙️
AS8038 - 50.240.195.137	lab1-juniper	Amazon.com	AS16509 - 198.32.176.36	group-a	Peer	0/0	inactive	⚙️
AS8038 - 50.240.195.135	lab1-cisco	Akamai	AS20940 - 206.126.236.102	group-a	Peer	0/0	Idle	⚙️

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

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

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

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



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.

Admin Guide

[Step 1 - Resources Overview](#)

[Step 2 - Admin Preferences](#)

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

[Step 4 - IPAM Administration](#)

[Step 5 - DNS Administration](#)

[Step 6 - DHCP Administration](#)

[Step 7 - Importing Data](#)

Admin Preferences

Overview



Details

dnsconfig

License Info

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

Application Settings

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

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

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

Peering Parameters

ASN : Enter the ASN that will be used

Backup Parameters (local install only)

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

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

Logging Options

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

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

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

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

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

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

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

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

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

Remote Log Type: Select SysLog format or JSON output

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

Authentication Options

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

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

Radius Enable: Check this box to enable RADIUS functionality.

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

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

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

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

LDAP authentication - for implementation details, go here

LDAP Enable: check the box to enable LDAP functionality.

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

LDAP Port: Set the port for your LDAP server

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

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

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

Templates

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

Authentication Options



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

i 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 [Permissions Overview](#) section for more detail on the local authentication configuration.

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

LDAP Authentication

LDAP Authentication

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

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

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 ) )
```

Configure the LDAP Server

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

```
dn: cn=JoeSmith,ou=people,dc=6connect,dc=com
cn: JoeSmith
sn: JoeSmith
objectclass: top
objectclass: person
objectclass: sixConnectPermissionsV2
sixConnGroup: "Global Admins"
sixConnGroup: "bonk"
sixConnGroup: "poof"
sixConnGroup: "groom"
userPassword: testpass
```

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

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

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

Example:

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

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

Test the LDAP Server

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

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

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

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

Example:

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

Configure ProVision for LDAP Authentication

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

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

LDAP Server Address: 52.240.195.12

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

LDAP Security: None

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

LDAP Fetch DN: cn=%LOGIN%



Setting default login authentication options

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

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

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

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

RADIUS Authentication

RADIUS Authentication

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

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

- Add the 6connect VSA to the Radius installation
- Configure Radius accounts
- Test the Radius account
- Configure ProVision for Radius Authentication

Add the 6connect VSA to the Radius Installation

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

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

ProVision 3.9.3 and prior:

▼ [Click here to expand...](#)

3.9.3 VSA text file

```
VENDOR          6connect          36009

BEGIN-VENDOR    6connect

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

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

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

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

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

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

END-VENDOR      6connect
```

ProVision 4.0 and greater:

▼ [Click here to expand...](#)

```

VENDOR          6connect          36009

BEGIN-VENDOR    6connect

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

END-VENDOR      6connect

```



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

Configure Radius Accounts

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

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

```

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

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

```

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

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

Setting up a RADIUS account

```

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

```



Note on RADIUS attributes

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

Test Radius Accounts

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

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

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

<insert example>

Configure ProVision for Radius Authentication

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

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



Setting default login options

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

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

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

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

IPAM Administration

Overview

IPAM Lists Management: <ul style="list-style-type: none">Edit IPAM TagsEdit IPAM RegionsEdit IPv4 Subnets DropdownEdit IPv6 Subnets Dropdown	Holding Tank Management: <ul style="list-style-type: none">Process Holding Tank now (Set to 60 days) LIR Management: <ul style="list-style-type: none">Add/Edit/Update LIRs Configuration Management: <ul style="list-style-type: none">IPAM Configuration
--	---

IPAM Lists Management

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

IPAM Configuration

IPAM Configuration

Holding Tank Days	<input type="text" value="60"/>
IPv4 Block Scanner Enable	<input checked="" type="checkbox"/>
IPv4 Block Scanner Max Block Size	<input type="text" value="/20"/>
Regions Enable	<input checked="" type="checkbox"/>
Generic Code Per Block Enable	<input checked="" type="checkbox"/>
Generic Code Per Block Display	<input checked="" type="checkbox"/>
Generic Code Per Block Name	<input type="text" value="DataCenter1"/>
Enable VLAN per Block	<input checked="" type="checkbox"/>
RIPE Database	<input checked="" type="radio"/> RIPE <input type="radio"/> TEST
Show /32 or /128 mask for statics	<input checked="" type="checkbox"/>

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

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

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

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

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

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

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

Holding Tank Management

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

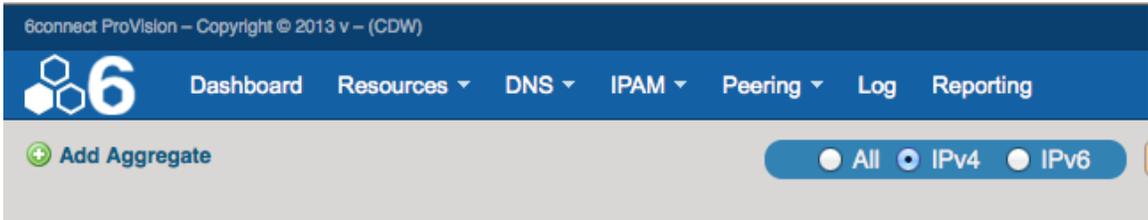
LIR Management and Use

ProVision supports multiple LIRs from the UI. This allows users to select from various LIRs when they want to update SWIP/RPSL information for a subnet allocation. Go to the [LIR Management and Use](#) page for more details.

Working with IP Blocks

Adding/Deleting IP Address Aggregates

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



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

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

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

Requirements to Delete an IP Aggregate

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

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

Architecting IP Address Blocks

Splitting/Aggregating blocks manually

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

Splitting/Aggregating blocks with Templates

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

2607:FEE0::/32 – ARIN

2013-08-09 13:17:49 Aggregate added: 2607:FEE0::/32 ops@6connect.com

Apply Template... ▾

- Apply Template...
- Auto Split

Manage

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

Regions Used: No regions used

Apply Clear Make Default

Export

ng	VLAN	Assigned To	U
		Available	20
		Available	20

- Edit
- Split
- Aggregate
- Assign
- Unassign
- Override Holding
- RIR Integration
- Templates
- Log
- Email

and then select the auto split parameters from there

Apply Template: 2607:fee0:8000::/33 (2607:fee0:8000:: - 2607:fee0:fff:fff:fff:fff:fff:fff)

Auto Split ▾ Select mask... ▾ Limit Apply Template Done

- Select mask...
- 34
- 35
- 36
- 37
- 38
- 39
- 40
- 41
- 42

IP Block parameters

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

ddi
07:
07:

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

RIR: ARIN | LIR: Select LIR... | Something: | VLAN: | Region: Region | Notes:

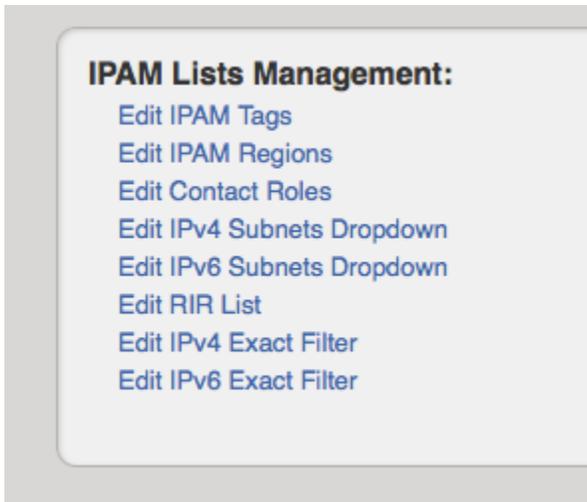
ASN:

Select tags...

<input type="checkbox"/> Backbone	<input type="checkbox"/> Customer	<input type="checkbox"/> Development	<input type="checkbox"/> DO NOT USE
<input type="checkbox"/> Infrastructure	<input type="checkbox"/> Interface	<input type="checkbox"/> IT	<input type="checkbox"/> LTE Mobile
<input type="checkbox"/> Point to Point	<input type="checkbox"/> Production	<input type="checkbox"/> Virtual Machines	<input type="checkbox"/> VOIP
<input type="checkbox"/> Legacy Verosity	<input type="checkbox"/> Servers	<input type="checkbox"/> BGP	<input type="checkbox"/> Anycast

Save Cancel

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



Assigning IP Space

To assign space, you do it using the IPAM Gadget from the particular Resource.

IPAM IPv4 IPv6

Assign Block:

Browse To Assign
List available blocks:

Smart Assign
RIR: Region: Size: Tags...

Filter:
Notes/CIDR... RIR: Region: Size: Acer Worldwide: Tags... Filter Clear

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

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

Browse to Assign

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

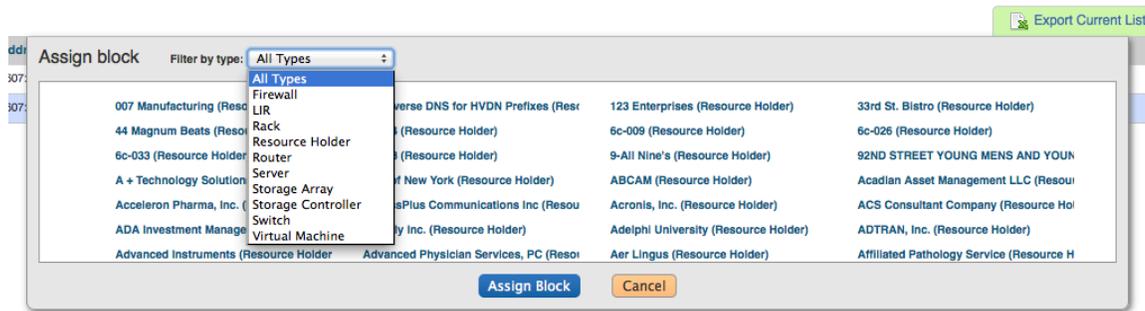
Direct Assign

This field allows you to manually enter an IP block to assign.

Smart Assign

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

Or you can assign blocks manually using the "Assign" function from the IPAM Manager screen (accessible from the IPAM Tab)



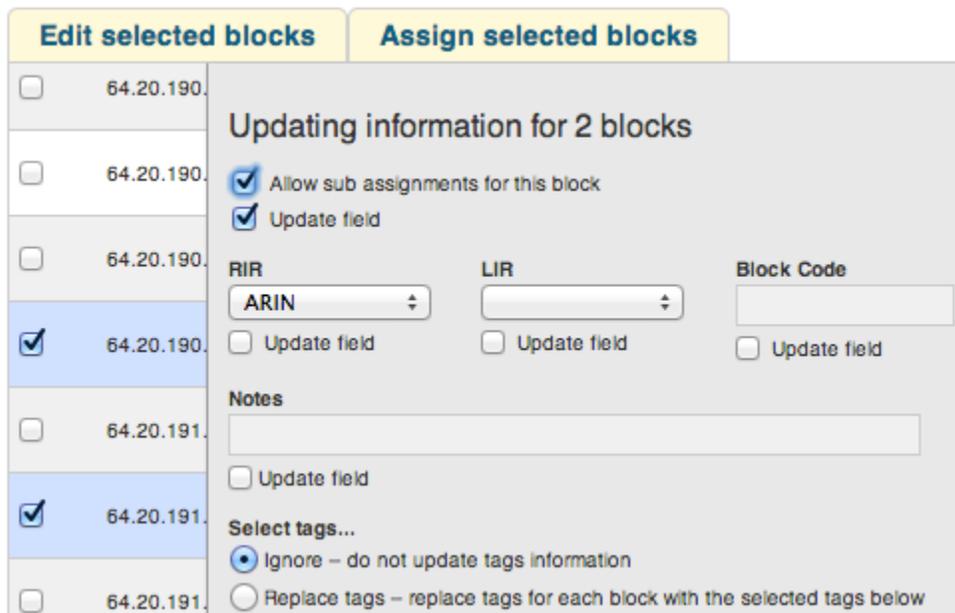
Please note that once a block is assigned, you will also have other options available, including reverse DNS, and IP subassignments



Sub Assigning IP Space

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

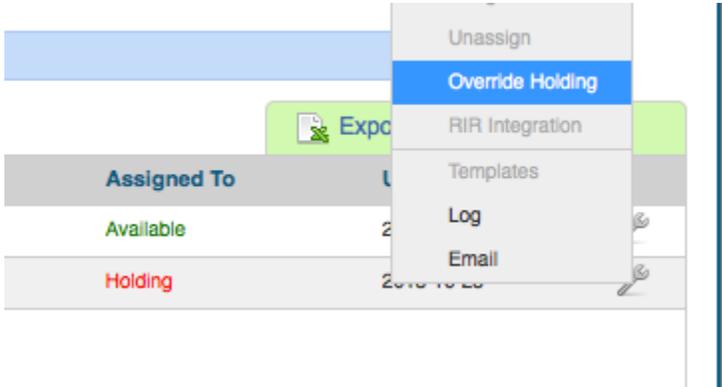
To allow sub assignments for multiple blocks at once, open the Manage screen for the aggregate. Then, select the desired blocks and click "Edit Selected Blocks". The Multi-block edit interface will open. In that interface, select the check boxes next to "Allow sub assignments for this block" and the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



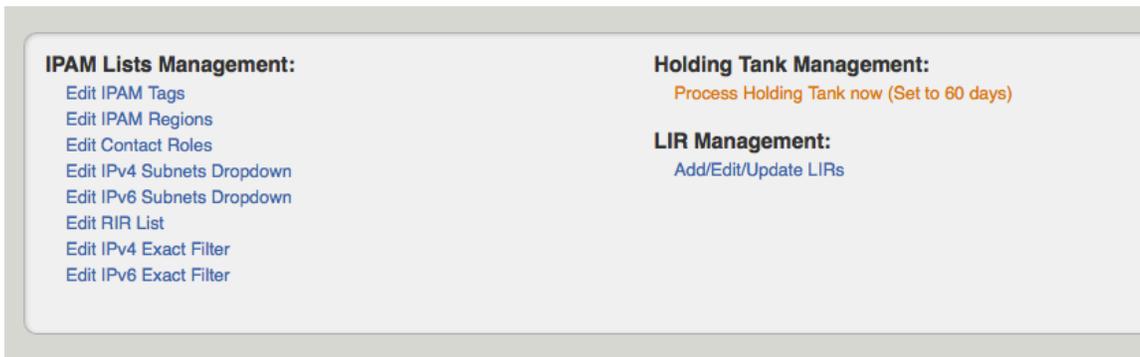
Unassigning IP Space

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

1) Manually override the holding tank



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



IPAM Parameters

Overview

The elements



Editing Tags

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

Editing Regions

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

Editing Subnet Dropdowns (used by the IPAM Gadget)

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



Note on Editing the Subnet Dropdown

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

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

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



Tracking Overlapping IP Space

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



RIR and SWIP/RPSL functions

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

Edit Exact Filter Dropdowns for Filter by Netmask

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

The screenshot shows the IPAM Manage interface. At the top, there's a search bar and filters for "Tags Used: No tags used" and "Regions Used: No regions used". Below the search bar, there's a "Filter By:" section with a dropdown menu for "All Mask" which is currently open, showing options: /24, /25, /26, /27, /28, /29, /30, /31, /32. The main table displays IP blocks with columns: Hosts, RIR, Region, Notes, Tags, Something, Assigned To, and Updated. The table is filtered by the selected subnet mask. The first row is highlighted in green, indicating it is selected.

Hosts	RIR	Region	Notes	Tags	Something	Assigned To	Updated
12.11.10.0/24	2	1918				brl_awesome sauce.com	2013-07-24 19:54:13
12.11.10.0/25	2	1918				brl_awesome sauce.com	2013-07-24 19:54:21
12.11.10.0/26	1	1918				brl_totallyrad.com	2013-07-25 16:41:40
12.11.10.0/27	1	1918				brl_totallyrad.com	2013-07-25 16:41:48
12.11.10.5/32	2	1918				Available	2013-07-25 16:41:03
12.11.10.6/31	8	1918				Some Customer	2013-08-05 19:46:21
12.11.10.8/29	8	1918				Some Customer	2013-08-05 19:46:37
12.11.10.16/29	8	1918				Available	2013-08-05 19:46:25

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

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

Search By: Search

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

Export Current List To CSV

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

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

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

Search By: Search

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

Export Current List To CSV

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

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

Dashboard Resources DNS IPAM Peering Log Reporting

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

Filtered by: Mask: 36, 48, 64 Search

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

Export Current List To CSV

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

Holding Tank Management

How it Works

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

Process Holding Tank

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

[Back to IPAM Admin](#)

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

i Pro-Tip!

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

LIR Management and Use

Overview

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

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

LIR Setup and Use

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

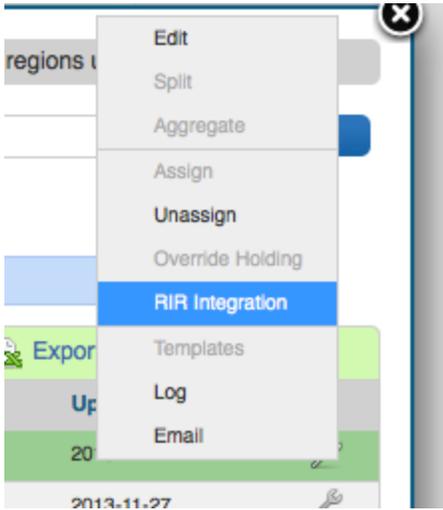
LIR Manager

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

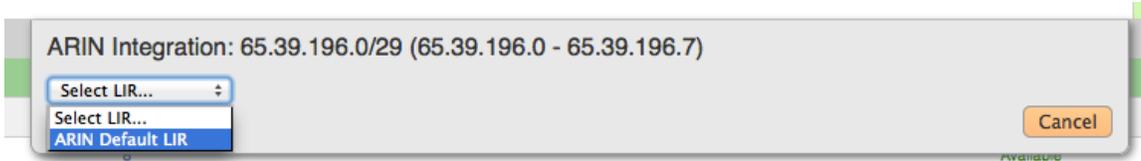
[Add LIR](#)

[Back to IPAM Admin](#)

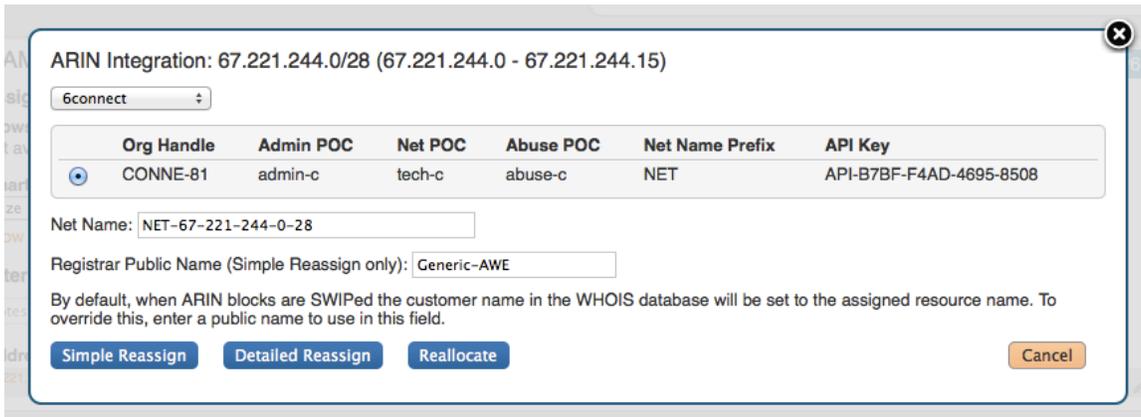
Once these have been configured, you will be able to use the **RIR integration** feature from the **Action** menu on the IPAM Manage screen or IPAM Gadget:



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



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



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

ARIN

Update LIR

RIR

Name

ASN

Org Handle ⊖ Delete

Admin POC

Tech POC

Abuse POC

NET Name Prefix

API Key

⊕ Add Org

Update

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

RIPE

Add LIR

RIR

Name

ASN

Maintainer Delete

Password

Admin Contact

Tech Contact

+ Add Maintainer

Update

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

ARIN LIR Setup and Use

Step 1: Setup the LIR information via the LIR Manager

You will be prompted to the select the RIR

Add LIR

RIR

Name

ASN

Delete

Update

Add in the requisite Org and POC information

Add LIR

RIR ARIN

Name

ASN

Org Handle Delete

Admin POC

Tech POC

Abuse POC

NET Name Prefix

API Key

+ [Add Org](#)

Update



Multiple Org Support

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

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

Step 3: Update SWIP information

Functions supported:



SWIP Update Functionality Details

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

Simple Re-assign

From ARIN.net:

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

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

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

Detailed Re-assign

From [ARIN.net](#):

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

Re-allocate

From [ARIN.net](#):

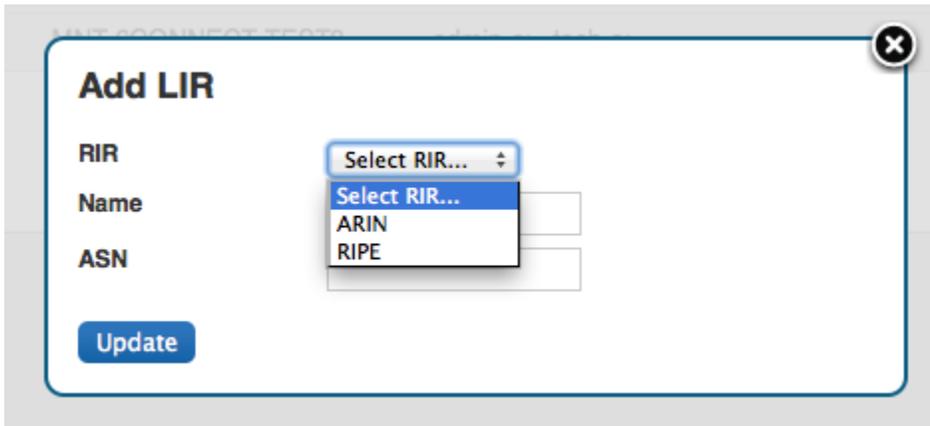
Used to subdelegate IP addresses to a downstream organization that will further subdelegate the IP addresses to their own customers. These requests must be submitted by an ARIN Online user account linked to the parent organization's Admin or Tech POC, or the Tech POC for the resource.

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

RIPE LIR Setup and Use

Step 1: Setup the LIR information via the LIR Manager

You will be prompted to the select the RIR:



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

Then add in the requisite Maintainer Object related information:

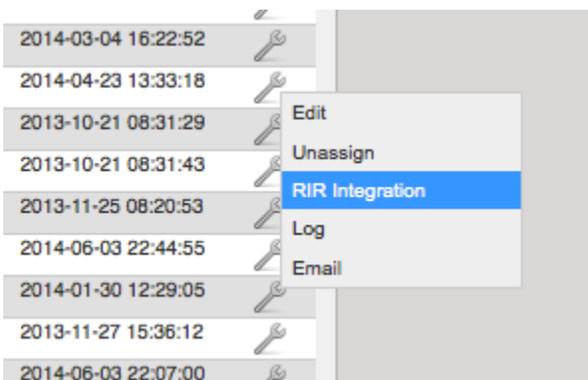
i Multiple Maintainer Object Support

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

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

Step 3: Update RPSL information

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



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

RIPE Integration: 192.162.1.0/24 (192.162.1.0 - 192.162.1.255)

RIPE Test LIR

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

Create Inetnum Cancel

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

193.0.0.0 - 193.0.0.31
(NET-193-0-0-0-27)
MNT-BY: MNT-6CONNECT-TEST

Hosts	RIR
32	RIPE 
32	RIPE

DNS Administration

DNS Admin

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

DNS Functions

Defaults and Options

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

Edit DNS Record Types

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

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

Edit DNS Delegations

Generate all DS records for DNSSEC

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

Generate zip file of all zones

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

Increment All Serials

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

DynECT Zone Import

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

PowerDNS Zone Import

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

DNS View ACL Management

- Manage ACLs for use in DNS Views.

DNS Zone Transfers

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

How to transfer zones:

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

Manage DNS Servers

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

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

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

- Server:** A dropdown menu with "ns1-ns1.6clabs.com" selected and a "New Server" button to its right.
- Nick Name:** A text input field containing "ns1".
- FQDN or IP:** A text input field containing "ns1.6clabs.com". To the right, there is an example: "ex: ns1.dns.6connect.net or 216.239.32.10".
- Default:** A dropdown menu with "Do Not Add to New Zones" selected.
- Transfer Type:** A dropdown menu with "ISC BIND" selected.
- Server Type:** A dropdown menu with "Master" selected.
- SOA:** A text input field containing "ns1.dns.6connect.net. hostmaster.6c". To the right, there is an example: "ex: ns1.dns.6connect.net. hostmaster.6connect.net."

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

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

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

Views

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

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

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

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

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

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

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

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

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

Adding a View

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

<insert image>

Adding Options to a View

Adding ACLs to Views

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

Bulk Zone Assignments

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

Bulk DNS Changes

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

Nameserver Management

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

Notes

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

System Information for Local Installations

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

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

Working with DNS Zones

Using the DNS Gadget

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



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

Navigating the DNS Tab

Clicking on the main DNS Tab, then on "Add Zone" will bring up the following UI.

Creating/Adding Zones

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

Create a DNS Zone

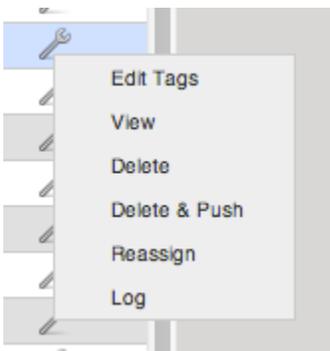
+

DNS Tab User Interface

Zone	Customer	Tags	DNSSEC	DS	Records	Actions
6clabs.com	6connect Labs				12	
6connect.com	6connect Available				7	
aaron.com	123 Department LAB		DNSSEC	X	11	
anna.com	123 Department LAB				12	
awesome.com	Anna's Test Site		DNSSEC	X	7	

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

DNS Zone Action Menu



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

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

Editing DNS Zones

Editing a Zone Record

There are two ways to edit a DNS zone:

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

1) Zone Management:

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

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

The screenshot shows the DNS zone management interface for the domain **awesome.com**. The DNSSEC status is **Enabled**. The zone parameters are: TTL: 3600, Serial: 2013071001, Refresh: 14400, Retry: 3600, Expire: 604800, and Minimum: 3600. The "Link Zone to Server" section shows the IP address 173.164.182.169 as a Master server. The "Current Masters" list includes ns1.6clabs.com, nalinmk.com, and cache.6connect.com. The "Current Slaves" list is empty (None!). The "Check Zone" button is highlighted with a green [OK!] status. Other buttons include "Edit SOA", "Disable Auto Check", and "Push Zone".

Figure 1: Normal zone with no errors

The screenshot shows the DNS zone management interface for the domain **awesome.com**. The DNSSEC status is **Enabled**. The zone parameters are: TTL: 3600, Serial: 2013071001, Refresh: 14400, Retry: 3600, Expire: 604800, and Minimum: 3600. The "Link Zone to Server" section shows the IP address 173.164.182.169 as a Master server. The "Current Masters" list includes ns1.6clabs.com, nalinmk.com, and cache.6connect.com. The "Current Slaves" list is empty (None!). The "Check Zone" button is highlighted with a red [ERRORS!] status. Below the zone details, there are three error messages: "View 6connectGeneric on cache.6connect.com NS 'ns2.dns.6connect.net.awesome.com' has no address records (A or AAAA)", "View 6connectGeneric on nalinmk.com NS 'ns2.dns.6connect.net.awesome.com' has no address records (A or AAAA)", and "View 6connectGeneric on ns1.6clabs.com NS 'ns2.dns.6connect.net.awesome.com' has no address records (A or AAAA)". Other buttons include "Edit SOA", "Disable Auto Check", and "Push Zone".

Figure 2: Zone with Errors

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

2) DNS Zone Record Data:

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

DNS Records View: Verbose BIND

Search by Record, Value, etc

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

Add a New Record

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

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

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

Save

Delete

Cancel

Configuring Views per DNS Zone

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

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

All Views

All Views

Local Only

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

3) Show DNS Zone:

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

```

Hide Zone File

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

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

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

```

4) Show DS Records:

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

5) Show Zone History:

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



Configuring ISC BIND Support

Getting Started

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

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

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

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

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

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

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

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

Configuring DynECT Support

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

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

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

Configuring PowerDNS Support

Environments supported

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

Either BIND or MySQL backend

Overview



Step 1: Setup your PowerDNS Server

Manage DNS Servers

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

Manage DNS Servers

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

Step 2: Import your PowerDNS zones

This operation will pull all zones on the target server.
 This operation may take quite some time.
 Choose a server:

Step 3: Edit/Push your zones to PowerDNS

DNS Zone Transfers:

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

Push Zones to Checked Servers:

BIND Backend

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

MySQL Backend



Note on SSH

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

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

Please note that Views are not supported with the MySQL backend



Database

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

Configuring Secure64 Support



A note on Ports

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

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

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



DO THIS

Make sure to add the line include: 6connect_nsd.conf to the nsd.conf file

Output/Input

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

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

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

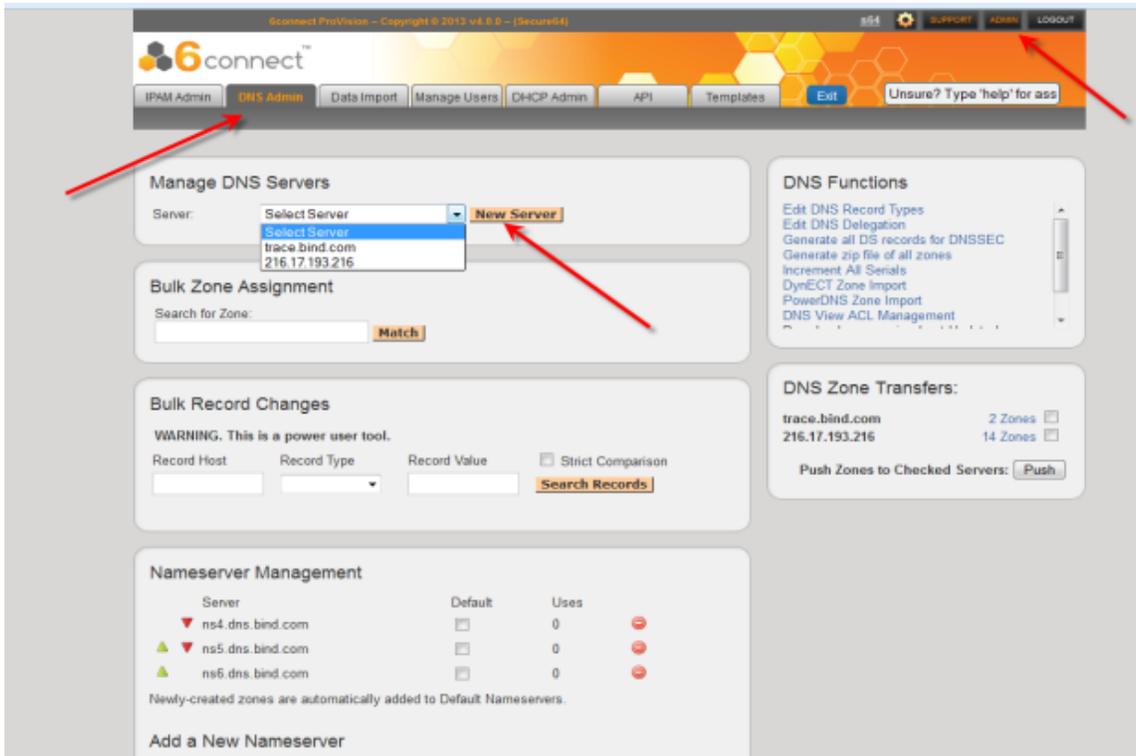
include: 6connect_nsd.conf
```

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

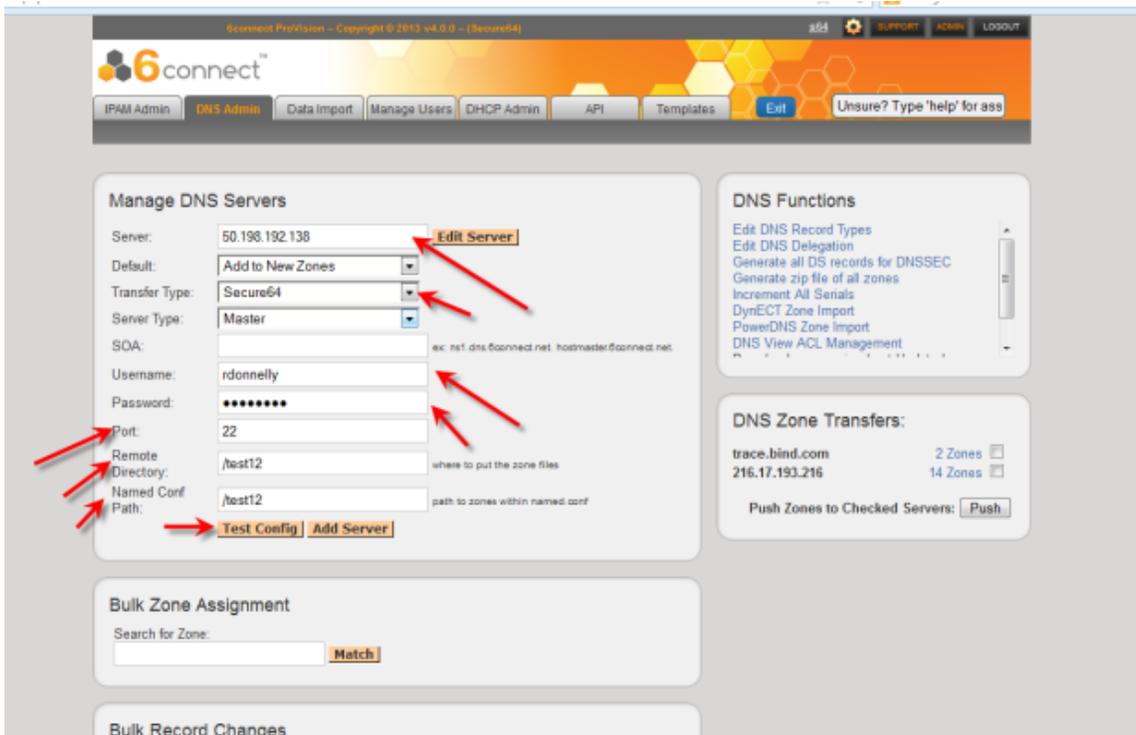
```
[authdnsadmin@Secure64DNS]# mkdir test12
[authdnsadmin@Secure64DNS]# ls
/:
322 2013-08-19 06:07:42 nsd.conf
<DIR> 1024 2013-08-16 17:30:12 test12
```

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

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



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



Step 4: Test the Secure64 DNS Server configuration

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

6connect™

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

Manage DNS Servers

Server: [Edit Server](#)

Default:

Transfer Type:

Server Type:

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

Username:

Password:

Port:

Remote Directory: where to put the zone files

Named Conf Path: path to zones within named.conf

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

Success!

DNS Functions

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

DNS Zone Transfers:

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

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

Success! Will show as depicted above.

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

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

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

6connect™

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

Manage DNS Servers

Server: [New Server](#)

Bulk Zone Assignment

Search for Zone: [Match](#)

Matched Zones:

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

Assign to: as [Assign](#)

Bulk Record Changes

WARNING. This is a power user tool.

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

[Search Records](#)

DNS Functions

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

DNS Zone Transfers:

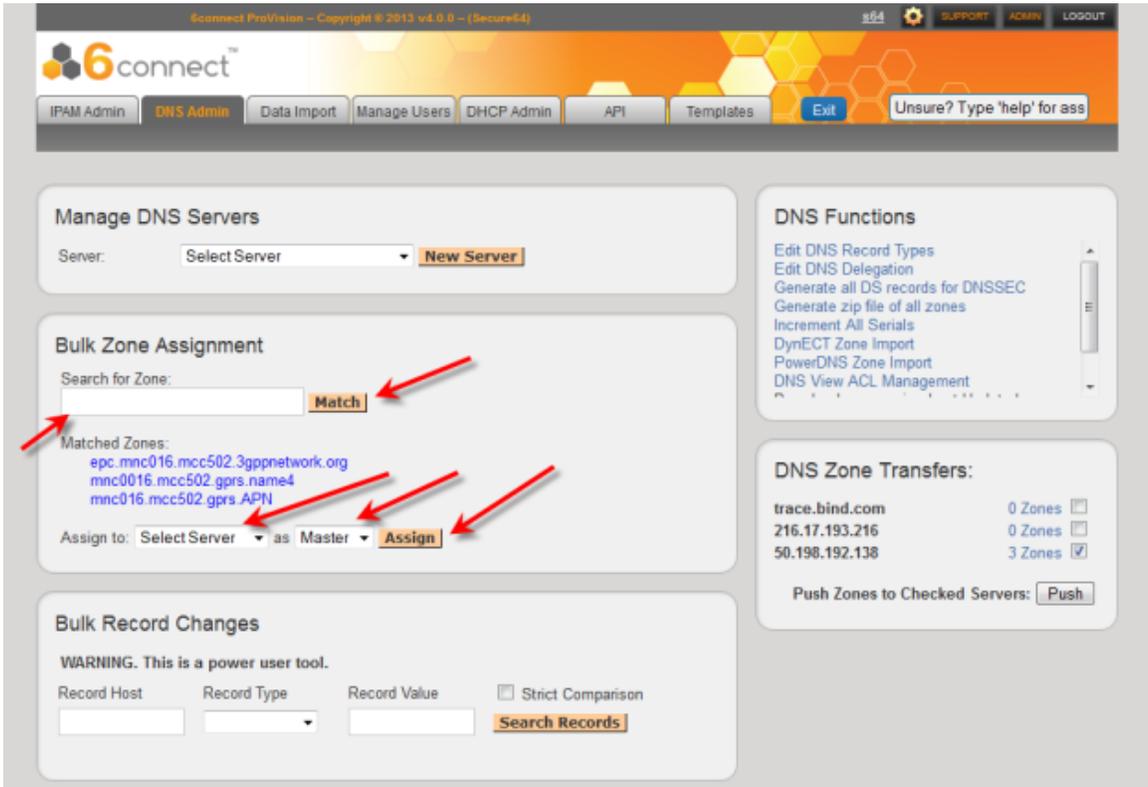
trace.bind.com	0 Zones	<input type="checkbox"/>
216.17.193.216	0 Zones	<input type="checkbox"/>
50.198.192.138	3 Zones	<input checked="" type="checkbox"/>

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



Search Tip

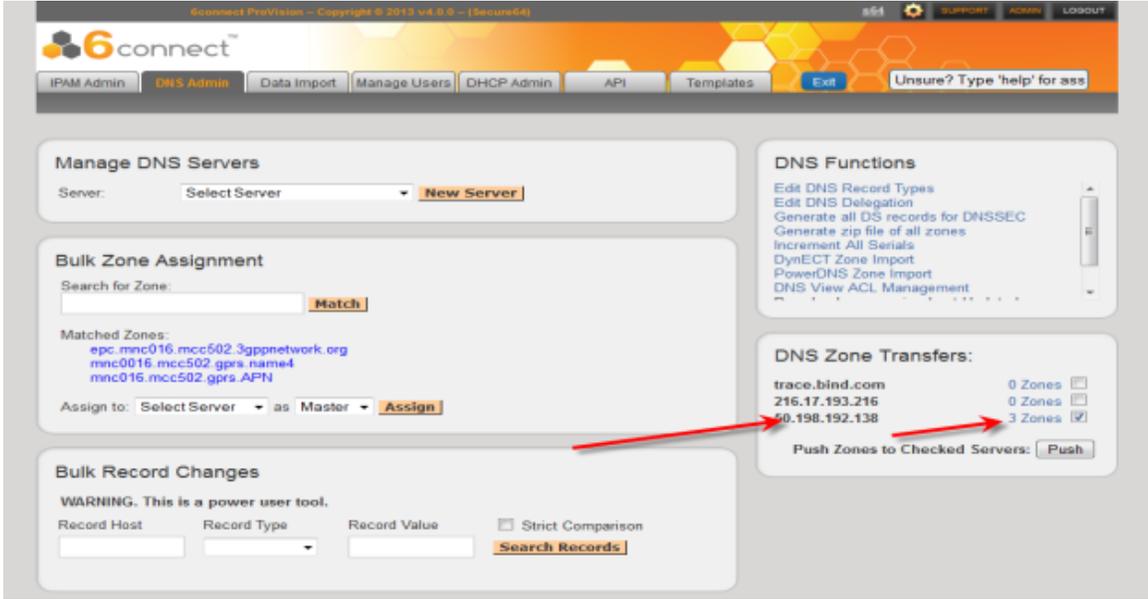
No character in the search area indicates a search for all zones as shown below



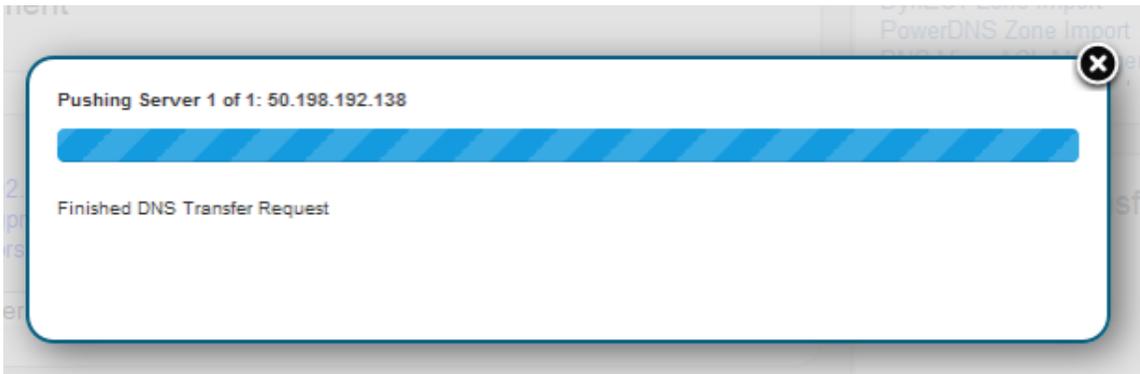
Select the **Select Server** and as **Master** dropdowns and **Assign** the above zones to this server. Verify the DNS Zones Transfers area indicates your server and the # of zones to transfer.

Step 6: Push Zones to Secure64 Server(s)

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



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



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

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

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



Verifying Zone pushes

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

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

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

You can verify the Push contents by doing a cat of the 6connect_nsd.conf



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

zone:
name: mnc016.mcc502.gprs.APN
zonefile: /test12/6connectGeneric/m/mnc016.mcc502.gprs.APN.zone
zone:
name: mnc0016.mcc502.gprs.name4
zonefile: /test12/6connectGeneric/m/mnc0016.mcc502.gprs.name4.zone
zone:
name: epc.mnc016.mcc502.3gppnetwork.org
zonefile: /test12/6connectGeneric/e/epc.mnc016.mcc502.3gppnetwork.org.zone
```

In the example above, three Zones have transferred.

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

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

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

Step 8: Validate Zone data in Your Infrastructure

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

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

For any questions regarding the integration of Secure64 products into 6connect ProVision, please email 6connect at support@6connect.com, or Secure64 at support@secure64.com

Configuring DNSSEC



DNSSEC Implementation

How enable DNSSEC (per zone)

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

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

For BIND

Coming soon

For DynECT

Coming soon

For Secure64 and PowerDNS



DNSSEC Signatures

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

Configuring Split Horizon/Views

video coming soon



WARNING

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

Create a List in the List manager

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

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

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

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

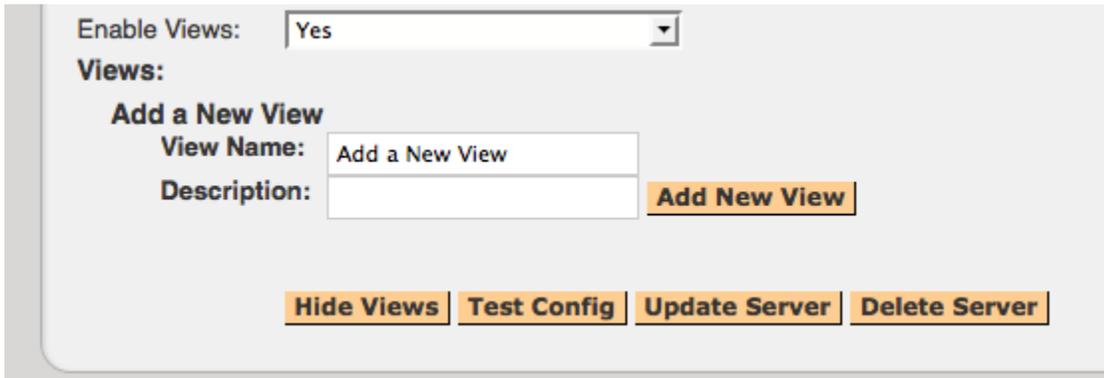
The List will now be available from the **Manage Lists** display area and can now be assigned to a Server View.

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

Define and Assign a View to the DNS Server

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

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



Enable Views: Yes

Views:

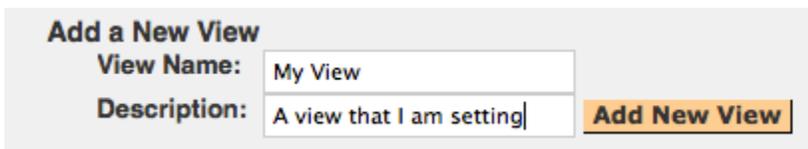
Add a New View

View Name: Add a New View

Description: **Add New View**

Hide Views **Test Config** **Update Server** **Delete Server**

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

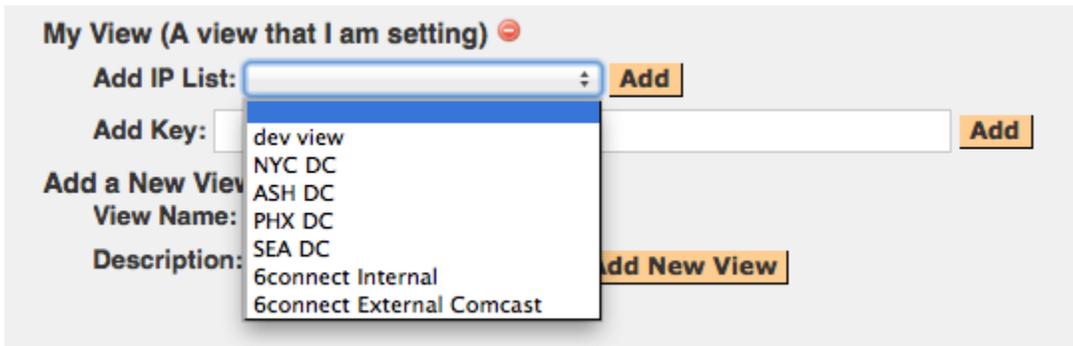


Add a New View

View Name: My View

Description: A view that I am setting **Add New View**

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



My View (A view that I am setting)

Add IP List: **Add**

Add Key: **Add**

Add a New View

View Name: **Add New View**

Description: **Add New View**

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

Assigning other Directives

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



My View (A view that I am setting)

Included IPs: 6connect Internal

Add IP List: **Add**

Add Key: Val: **Add**



A Note on Directives

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

Assign a View to a DNS Zone Record

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

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

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

Type	Record	Value	Description	TTL
A	www	12.12.12.12		

DNS Views: 173.164.182.169

- All Views
- All Views
- My View

6 A mail maps to 11.11.11.11

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

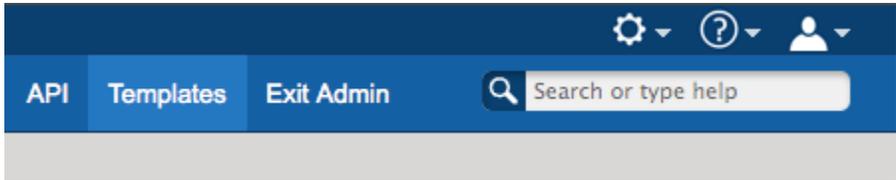
Configuring DNS Templates

Overview

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

Configuring DNS Templates

Go to the Templates Tab in the Admin Menu



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

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

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

Editing Demo Template

Name: Demo Template

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

Serial	Refresh	Retry	Expiry	Minimum
	14400	3600	604800	3600

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

Zone Records

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

Buttons: Save Template, Cancel

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

```

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

```

Using DNS Templates

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

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

Importing Your Data

Step 1: Normalize your Data

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



Data Encoding Format

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

Step 2: Prep your Data

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

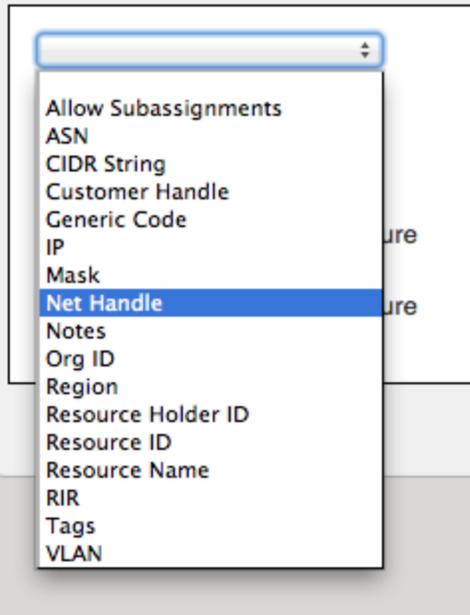
[For Company information](#) you can import relevant data including mailing/billing address information as well as ARIN specific SWIP fields, and specific DNS servers.

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

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

Step 2: Define Columns

The Import process requires you to enumerate the columns to be imported. The Mask column. There can be multiple Note

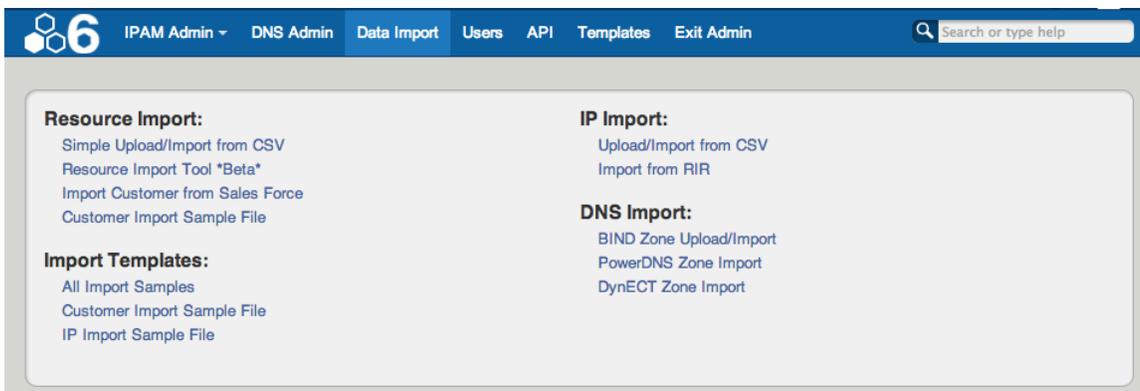


Allow Subassignments option

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.



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 "DNS Import".

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

Resource Import:

- Simple Upload/Import from CSV
- Resource Import Tool *Beta*
- Import Customer from Sales Force
- Customer Import Sample File

Import Templates:

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

IP Import:

- Upload/Import from CSV
- Import from RIR

DNS Import:

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

Step 1: Lookup from Source IP

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

Welcome to 6connect's Network Automation Platform!

I believe your organization name is: **AT&T Internet Services**

I believe your ARIN ORGID is: **SIS-80**

If you know your organizations ARIN ORG ID, enter it below:

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

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.

6 Dashboard Resources ▾ DNS ▾ DHCP ▾ IPAM ▾ Peering ▾ Log Reporting

Welcome to 6connect's Network Automation Platform!

<p>This is IPv6 & IPv4 non-1918 space I have discovered</p> <table border="0" style="width: 100%;"> <tr> <td>Found IPv4 block: 104.48.0.0/12</td> <td>ARIN</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>Found IPv4 block: 208.188.0.0/14</td> <td>ARIN</td> <td><input type="button" value="Add Aggregate"/></td> </tr> <tr> <td>Found IPv4 block: 207.193.0.0/16</td> <td>ARIN</td> <td><input type="button" value="Add Aggregate"/></td> </tr> <tr> <td>Found IPv4 block: 209.184.0.0/16</td> <td>ARIN</td> <td><input type="button" value="Add Aggregate"/></td> </tr> <tr> <td>Found IPv4 block: 216.60.0.0/14</td> <td>ARIN</td> <td><input type="button" value="Add Aggregate"/></td> </tr> <tr> <td>Found IPv4 block: 63.170.248.0/25</td> <td>ARIN</td> <td><input type="button" value="Add Aggregate"/></td> </tr> <tr> <td>Found IPv4 block: 64.216.0.0/14</td> <td>ARIN</td> <td><input type="button" value="Add Aggregate"/></td> </tr> </table>	Found IPv4 block: 104.48.0.0/12	ARIN	<input checked="" type="checkbox"/>	Found IPv4 block: 208.188.0.0/14	ARIN	<input type="button" value="Add Aggregate"/>	Found IPv4 block: 207.193.0.0/16	ARIN	<input type="button" value="Add Aggregate"/>	Found IPv4 block: 209.184.0.0/16	ARIN	<input type="button" value="Add Aggregate"/>	Found IPv4 block: 216.60.0.0/14	ARIN	<input type="button" value="Add Aggregate"/>	Found IPv4 block: 63.170.248.0/25	ARIN	<input type="button" value="Add Aggregate"/>	Found IPv4 block: 64.216.0.0/14	ARIN	<input type="button" value="Add Aggregate"/>	<p>If you will be using RFC1918 space, you will likely want to add from this list:</p> <table border="0" style="width: 100%;"> <tr> <td>RFC1918 block: 10.0.0.0/8</td> <td>1918</td> <td>10.0.0.0/8</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>RFC1918 block: 192.168.0.0/16</td> <td>1918</td> <td>192.168.0.0/16</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>RFC1918 block: 172.16.0.0/12</td> <td>1918</td> <td></td> <td><input type="button" value="Add Aggregate"/></td> </tr> </table> <p>If you will be using Shared Transition Space, add:</p> <table border="0" style="width: 100%;"> <tr> <td>RFC6598 block: 100.64.0.0/10</td> <td>6598</td> <td></td> <td><input type="button" value="Add Aggregate"/></td> </tr> </table>	RFC1918 block: 10.0.0.0/8	1918	10.0.0.0/8	<input checked="" type="checkbox"/>	RFC1918 block: 192.168.0.0/16	1918	192.168.0.0/16	<input checked="" type="checkbox"/>	RFC1918 block: 172.16.0.0/12	1918		<input type="button" value="Add Aggregate"/>	RFC6598 block: 100.64.0.0/10	6598		<input type="button" value="Add Aggregate"/>
Found IPv4 block: 104.48.0.0/12	ARIN	<input checked="" type="checkbox"/>																																				
Found IPv4 block: 208.188.0.0/14	ARIN	<input type="button" value="Add Aggregate"/>																																				
Found IPv4 block: 207.193.0.0/16	ARIN	<input type="button" value="Add Aggregate"/>																																				
Found IPv4 block: 209.184.0.0/16	ARIN	<input type="button" value="Add Aggregate"/>																																				
Found IPv4 block: 216.60.0.0/14	ARIN	<input type="button" value="Add Aggregate"/>																																				
Found IPv4 block: 63.170.248.0/25	ARIN	<input type="button" value="Add Aggregate"/>																																				
Found IPv4 block: 64.216.0.0/14	ARIN	<input type="button" value="Add Aggregate"/>																																				
RFC1918 block: 10.0.0.0/8	1918	10.0.0.0/8	<input checked="" type="checkbox"/>																																			
RFC1918 block: 192.168.0.0/16	1918	192.168.0.0/16	<input checked="" type="checkbox"/>																																			
RFC1918 block: 172.16.0.0/12	1918		<input type="button" value="Add Aggregate"/>																																			
RFC6598 block: 100.64.0.0/10	6598		<input type="button" value="Add Aggregate"/>																																			

Step 3: Customizing

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

Import DNS Zones

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

Preparing your DNS Zones for Import

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

There are three components to the upload process:

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

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

Simple DNS Config File

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

or could be more complex like this file structure directory:

Complex DNS Config File

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

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

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".

Resource Import:
Simple Upload/Import from CSV
Resource Import Tool *Beta*
Import Customer from Sales Force
Customer Import Sample File

IP Import:
Upload/Import from CSV
Import from RIR

DNS Import:
BIND Zone Upload/Import
PowerDNS Zone Import
DynECT Zone Import

Import Templates:
All Import Samples
Customer Import Sample File
IP Import Sample File

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

DNS Zone Import

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

Job Name: **Description:**

Configuration File: conf.conf **Archive File:** zones.zip **CSV File:** config.csv

Required: a configuration file in BIND or NSD format. Required: a ZIP or TAR of your zones. Optional: a CSV file mapping zones to customers and DNS Servers.



Working with Large or Multiple Data Sets

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

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

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

DNS Zone Import

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

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

Job Name: **Description:**

Step 2: Map Data Columns (Optional)

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

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

When completed, hit "Next".

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

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



Step 3: Reviewing Data

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

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

View:

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

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

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

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

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

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

View:

Zone Name: **Resource Holder:**

DNS Servers:

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

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

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

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

Current Block: Finished!

Resource Import Tool

Importing Resources

The Resource Import Tool

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

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

Step 0: Before You Begin

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

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



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

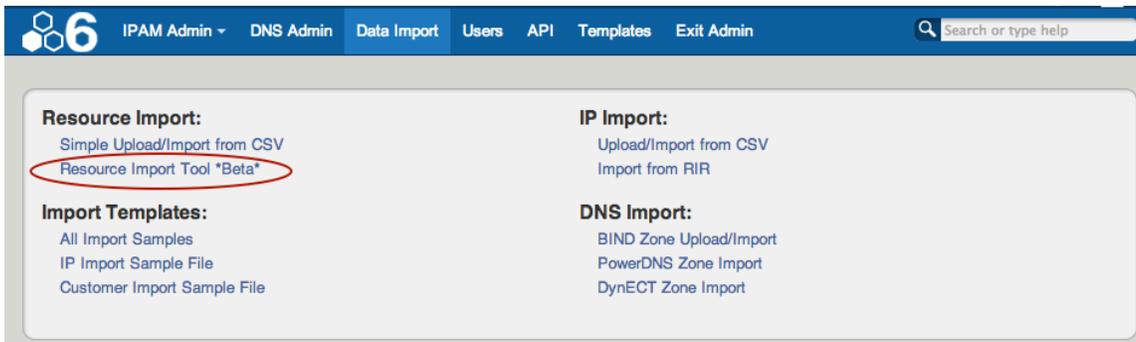


Best Practice

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

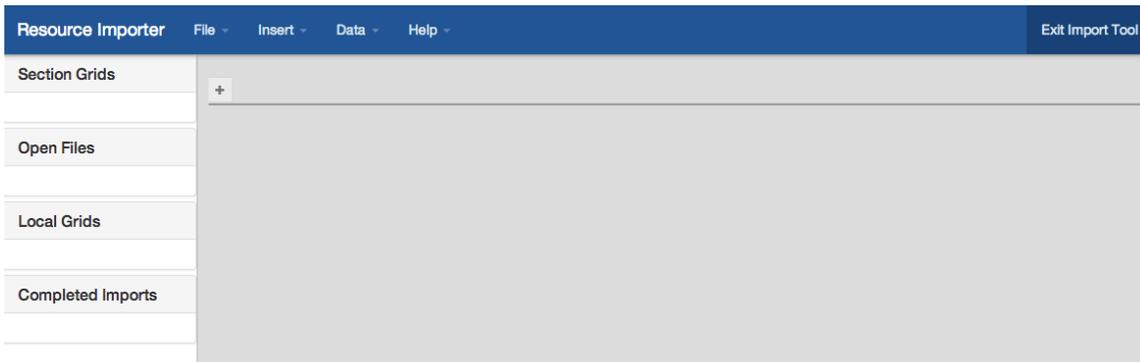
Opening the Resource Import Tool

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



The Resource Import Tool UI

When you first open the Resource Importer, you will be given the option to view a short on-screen guide to using the tool. After stepping through the guide and/or exiting out of it, the tool will look like this:



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

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

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

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

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

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

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

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

Resource Importer Walkthrough

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

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

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

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

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

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

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

Resource Importer Walkthrough - Step 1

Importing Resources

Before You Begin

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

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

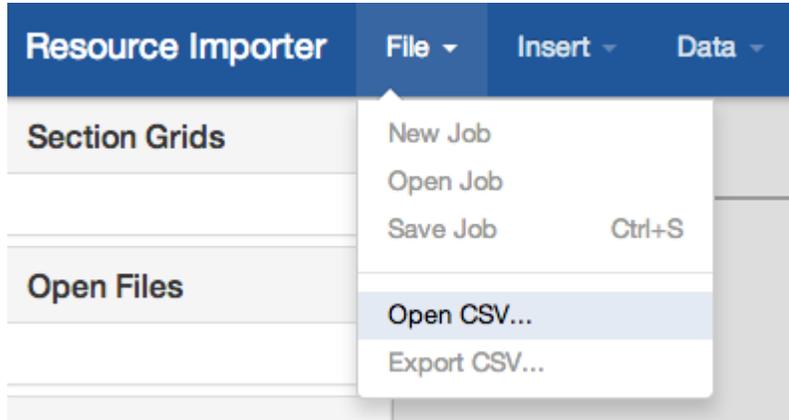
In order to illustrate the abilities of the Resource Importer to edit data and adjust for formatting issues, the Contact Import Sample .csv is used intentionally leaving a few less-than-ideal conditions (much like you may encounter in real life) such as leaving typos, having an extra data

column, and missing a needed column. If you follow the "Before you Begin" requirements and "Best Practice" notes, however, you may be able to skip any editing or column adjustment steps.

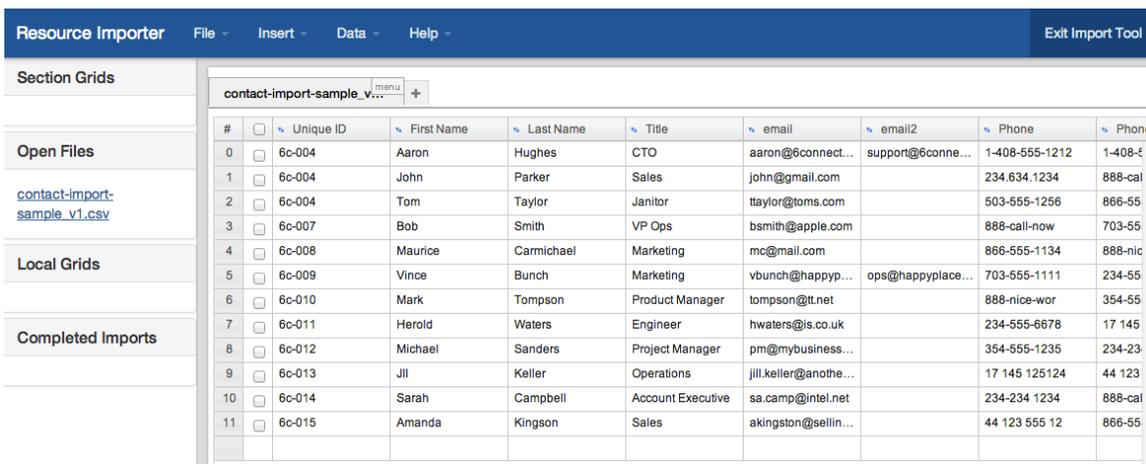
When you are ready to begin, open the Resource Importer and proceed to Step 1.

Step 1: Upload your .csv data file

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



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



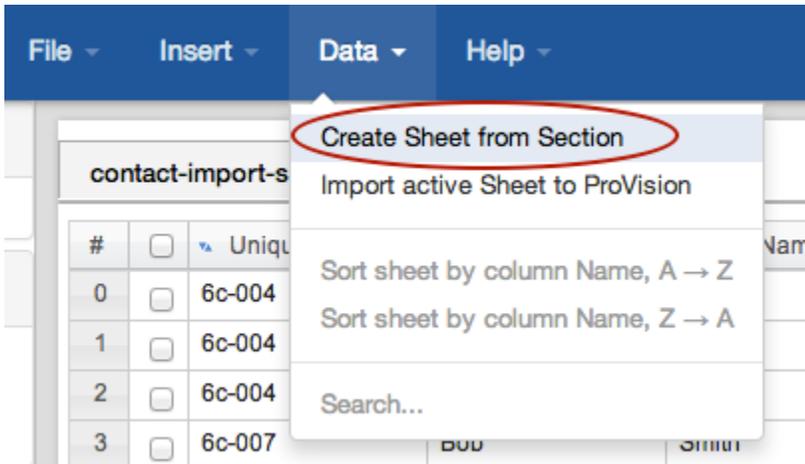
After opening your .csv grid, proceed to Step 2 - Open a template grid from an existing Section

Resource Importer Walkthrough - Step 2

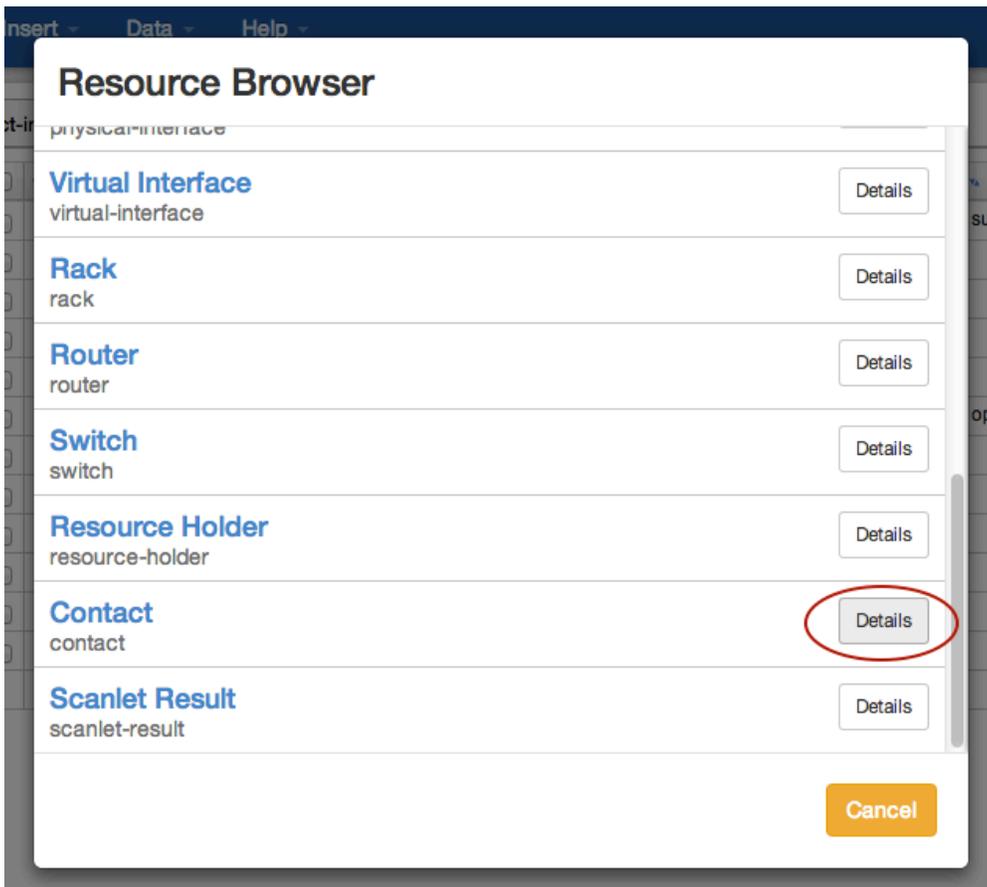
Importing Resources

Step 2: Open a Template Grid from an existing Section

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

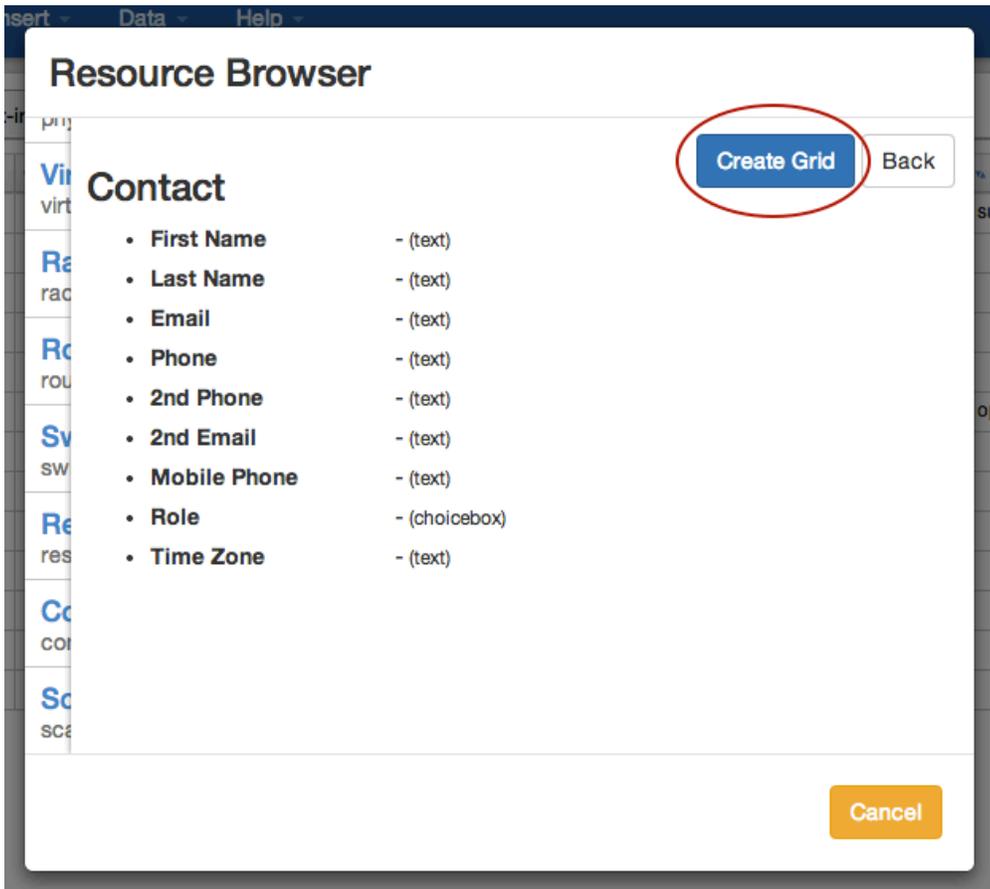


The Resource Browser will pop up, showing the list of Sections currently available in Provision. Clicking on the "Details" button will show the fields for that Section.

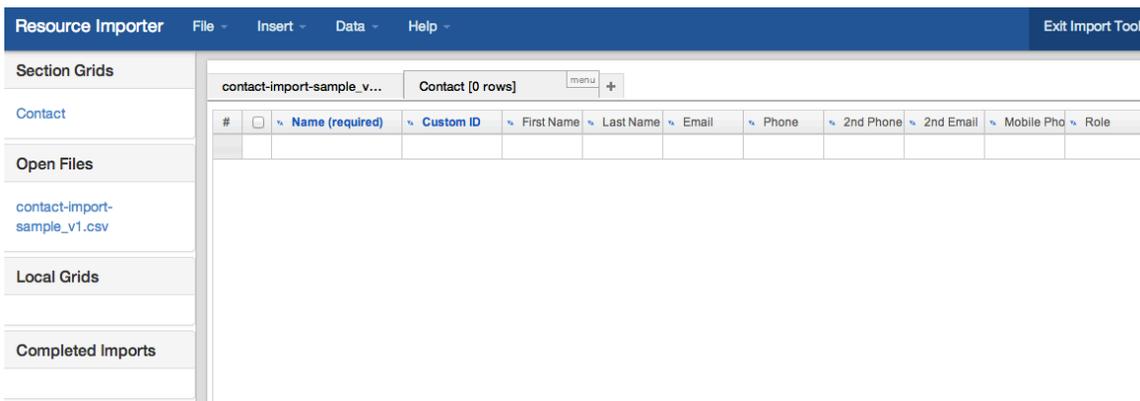


Verify that the Section and available fields match the type of data you are trying to import. In this case, the Section "Contact" has the fields that correlate to our spreadsheet data.

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



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



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

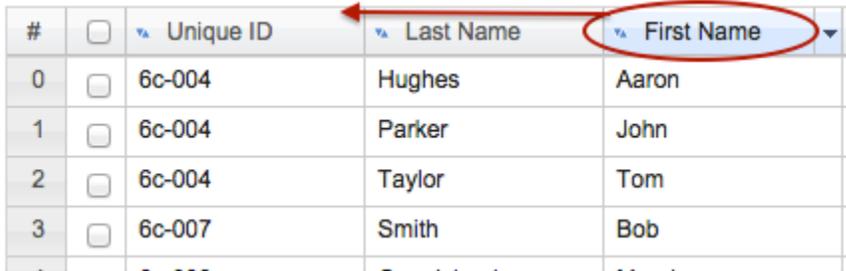
Resource Importer Walkthrough - Step 3

Importing Resources

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

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

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



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

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

Common Column Editing Questions

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

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

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

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

Resource Importer Walkthrough - Step 4

Importing Resources

Step 4: Edit data as needed

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

Common Editing Questions:

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

To hide extraneous column information:

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

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

- #
- Unique ID
- First Name
- Last Name
- Title
- email
- email2
- Phone
- Phone 2
- Phone Cell
- Role
- TimeZone
- Force fit columns
- Synchronous resize

To Edit Data in the Resource Importer

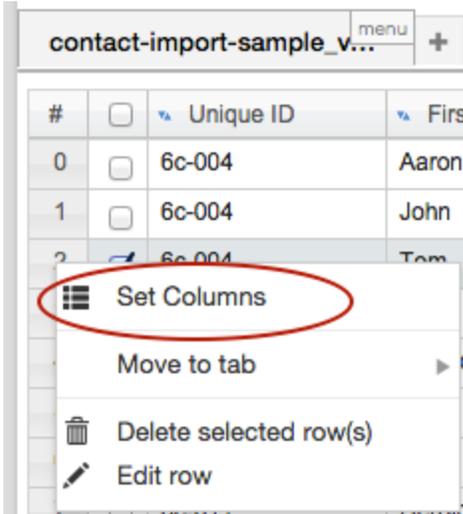
Data in the grids can be edited directly by clicking on the cell(s). In our example, we can see that "Amanda Kingston" should really be "Amanda Kingston". Let's fix that! Click in the cell, type in the edit you wish to make, and then click outside of the cell to exit edit mode. To edit a full row of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

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

If a column header is over the wrong data:

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

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



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

Change Column Header

- #
-
- Unique ID
- First Name
- Last Name
- Title
- email
- email2
- Phone
- Phone Cell
- Phone 2
- TimeZone
- Role

OK Cancel

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

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

Currently, our options to fix this are:

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

Or:

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

#	Name (required)	Custom ID	First Name	Last Name	Email	2nd Email	Phone	
0	Aaron Hughes	6c-004	Aaron	Hughes	aaron@6c...	support@...	1-408-555...	1-
1	Amanda Kingston	6c-015	Amanda	Kingston	akingston...		44 123 55...	
2	Bob Smith	6c-007	Bob	Smith	bsmith@a...		888-call-now	
3	Herold Waters	6c-011	Herold	Waters	hwaters@i...		234-555-6...	
4	Jill Keller	6c-013	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

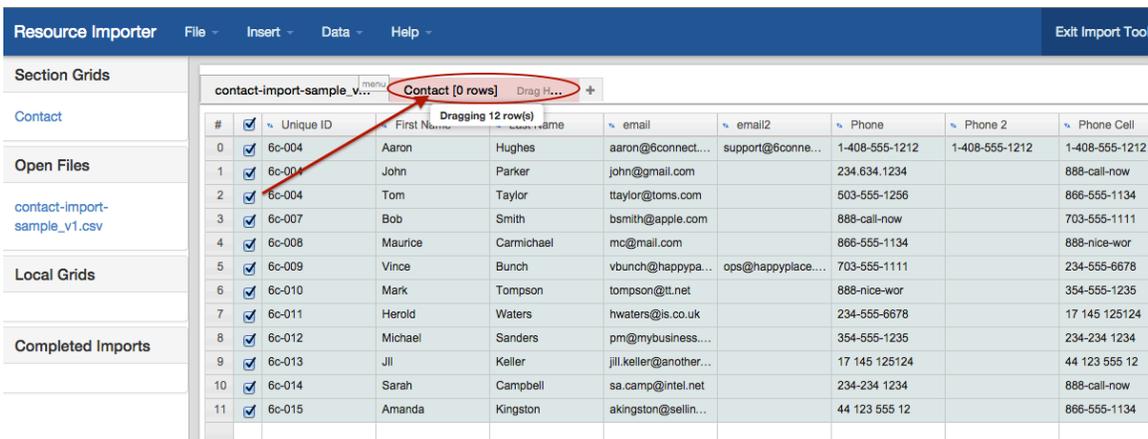
Resource Importer Walkthrough - Step 5

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.

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

After moving your data into the Section grid, proceed to [Step 6 - Importing into ProVision](#).

Resource Importer Walkthrough - Step 6

Importing Resources

Step 6: Import into ProVision

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

The screenshot shows the 'Resource Importer' application with the 'Data' menu open. The 'Import active Sheet to ProVision' option is circled in red. The background shows the same data grid as in the previous image.

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

Permissions Overview

Overview

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

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



Permission Levels

Global Permissions

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

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

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

Resource Permissions

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

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

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

Global Permissions

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

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

Global Permission Details

Group Information

Name:

Enabled:

Group Users

Resource Permissions (Hide Details)

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

[Add More Group Permissions](#)

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

Functional Area	Description
IPAM	IP Address Management functionality - this covers the IPAM Tab in addition to the IPAM "Gadget" that can be present in Resources.
DNS	DNS Zone/Zone Record Management functionality - this covers the DNS Tab in addition to the DNS "Gadget" that can be present in Resources.
Peering	Peering functionality - covers the Peering Tab, both the Communication Manager and the Session Manager.
Resources	Resource functionality - this controls access for Resources depending on either the TLR or the individual Resource.
User	User/Group management - this controls access for User and Group functions within the administrative area for ProVision.
SWIP*	This affects the SWIP/RPSL integration for ARIN/RIPE. This way a user can either be enabled to have this capability or not.
Admin*	This controls whether a user is a administrator for the global ProVision application.

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

Resource Permissions

Resource Permissions apply to designated Resources within ProVision.

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

Resource Permission Details

Users

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

Add User

Creating/Editing Accounts

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

Username:

First Name:

Last Name:

Disable help bubbles?

Groups:

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

Save

Setting/Resetting User Passwords

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

yes

Reset Password

New Password:

Send email?

From:

To:

Subject:

Message:

Dear test mcTest,

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

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

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

Regards,
6connect Automated Admin

User Groups

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

- Global Admin
- Global Read-Only

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

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

Add Group

Overlapping group and user permissions

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

Verifying Permissions

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

Check User Permissions

User: Resource:

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

Groups effecting this user on this resource: test group 1

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.

Common Tasks

UI Tours

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

Common Tasks

IPAM

[Adding/Editing blocks](#)

[Reserve IP space](#)

[Aggregating/Splitting blocks](#)

[SWIP configuration and use](#)

[RPSL configuration and use](#)

DNS

[Importing DNS Zones](#)

Peering

[Adding routers](#)

[Adding sessions](#)

[Importing sessions](#)

Importing Data

[Resource Importer Walkthrough](#)

[Import Aggregate Blocks](#)

[Import DNS Zones](#)

UI Tours

Administration

Managing Group and User Permissions



DNS

PowerDNS w/ MySQL Support



FAQ



- ✓ [How can I manage overlapping/duplicate IP blocks?](#)
When breaking apart blocks - use the LIR functions (Admin->IPAM Admin->RIR/LIR Manager) to differentiate blocks for 1918 space.
- ✓ [On the dashboard, I see "n+1" users - why?](#)
The users list includes a "system user" that is only used by ProVision internally in the application.
- ✓ [I have already SWIPed subnets to ARIN. What happens if I try to SWIP from ProVision, but the block is already SWIPed?](#)
In the case when a user already has SWIPped blocks to ARIN, 6connect checks prior to actually performing a SWIP. In the process, if the IP block is already SWIPped, it will check for existing ARIN customer data and update the 6connect data to reflect what ARIN has on file. Once that is complete, the user can then perform a de-SWIP function using ProVision.
- ✓ [How does 6connect avoid duplicate assignments or resolve conflicts?](#)
When you make an API request to assign a block, if the block is already assigned to another resource, you will receive an error. If your process is to search for and then assign blocks, the Smart Assign API call may be very helpful. That call combines the search and assignment into one action.
- ✓ [My VM works, but I am getting a "URL Not Found" error when using ProVision](#)
Please make sure that URL rewriting is enabled in your instance (apache mod_rewrite)
- ✓ [My DNS zone views aren't working as they should!](#)
In some legacy instances we have seen zone record-view linkages come out of alignment and result in unexpected behavior.



BACKUP YOUR DATABASE

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

First, verify the error with the following mysql commands:

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

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

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

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

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

▼ How can I 'reserve' IP space?

To create a reserved pool of IP space, you can create a Section called "Reserved", add the IPAM gadget to it, then create an Entry with that Section to be the address group. From there, use the IPAM gadget and the IPAM Manage page to assign and unassign IP space from that pool.

The workflow for this would be:

1. Assign IP space to the "Reserved" Section.
2. When you are ready to pull space from "Reserved", unassign the desired block. This moves it to the holding tank.
3. Override the holding tank to make the space "available". This can be done in the IPAM manager via the "Override Holding" wrench option, or a manual 'pull out of holding' API call.
4. Assign the block to the desired Resource.

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.

Status	
Backup	
User Accounts	25
Version	3.9.x
Coming Soon	

You can also submit product feedback and feature requests to support@6connect.com

CLI (Alpha)

Command Line Interface - ALPHA

- Overview
- CLI Commands (ALPHA)

Overview

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



How to Access the CLI from your browser

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

CLI Commands (ALPHA)



CLI Help

When in the CLI, type:

```
ipam man
```

for sample commands and syntax

Currently, the CLI supports the following commands:

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

show: show details for a block. Examples:

- "ipam show 10.0.0.0/8" will show details for the block 10.0.0.0/8
- "ipam show holding" will show details for all blocks in the Holding Tank
- "ipam show "<resource name>" will show details for all blocks assigned to <resource name>

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

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

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

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

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

API v1

- [1 - Overview](#)
- [2 - Making API Requests](#)
- [3 - SDK - PHP](#)

- [API Module - Admin and Audit](#)
- [API Module - DHCP](#)
- [API Module - DNS](#)
- [API Module - IPAM](#)
- [API Module - LIR](#)
- [API Module - Peering](#)
- [API Module - Resource](#)
- [How Do I...](#)

1 - Overview

6connect API - Overview

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

To learn more about request formatting, making requests, and the tools available, visit [Making API Requests](#). You can also get the [PHP SDK](#) for PHP libraries and sample code.

Here are some important details about our ReST implementation:

- The API only comes with the full 6Connect IPAM product. If you would like to upgrade to the full version, contact sales@6Connect.com.
- 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,"child2":null,"is_assigned":"0","is_swipped":"0",
  "is_aggregate":"1","custid":"holding","last_updated_time":"2012-03-20
  09:49:00","description":null,"parent":null,"rir":"ARIN","notes":"2012-03-20
  09:49:00","generic_code":null,"region":null,"vlan":null,"arin_net_id":null,"arin_cust_
  id":null,"arin_swip_time":"0000-00-00 00:00:00","assigned_time":"2012-03-20
  09:45:12"}]}
```

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=8jxj4IApYmgb5IZ0wBY4tFv+WiIXb5JuVpjrwpuyXQo=
```

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

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

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

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

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

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

Calculating it out yields the completed hash:

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

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

```
https://cloud.6connect.com/6c_375/api/v1/api.php?target=ipam&action=get&type=IP&mask=24&apiKey=32-5DAYTJQY2TZHOFOB&hash=yneSFMyxPPE+3W4IOkVp50K3VStatBcRRak+2ygDUWQ=
```



A Note on False Positives

ProVision utilizes several possible authentication schemes of which key-based API authentication is only one. Session-based, username/password authentication is used for the majority of user interaction with the ProVision front end. Because session

information is stored in browsers cookies, a browser can be authenticated to execute API commands as long as the session is active.

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

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

Other Languages

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

3 - SDK - PHP

6connect API - Getting Started with the SDK for PHP

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

Prerequisites

The API only comes with a licensed 6connect ProVision application. If you would like access to a ProVision license please contact sales@6connect.com.

Create Your API Credentials

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

To create your API Key and Secret Key:

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

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



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

Important!



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

Requirements

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

Minimum Requirements

- PHP 5.5 or newer.
- PHP JSON and PCRE extensions (XML will be coming soon).
- Curl PHP extension compiled with OpenSSL libraries. [Click here for more information on curl.](#)

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

Install the SDK

Download the file [6connect-PHP-SDKv2.tar.gz](#)

Configure the SDK Security Credentials

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

Important!

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

Need More Information?

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

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

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

API Module - Admin and Audit

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

Authentication Testing

Authentication Testing



testSSH															
URL	/api/v1/api.php?target=auth&action=testSSH														
Description	Returns success or failure of a connection to an external server via SSH.														
Returns	Examples: <table border="1"><tbody><tr><td>SUCCESSFUL</td><td colspan="2">{"success":1,"message":"Success!"}</td></tr><tr><td>ERROR</td><td colspan="2">{"success":0,"message":"error message"}</td></tr></tbody></table>			SUCCESSFUL	{"success":1,"message":"Success!"}		ERROR	{"success":0,"message":"error message"}							
SUCCESSFUL	{"success":1,"message":"Success!"}														
ERROR	{"success":0,"message":"error message"}														
Required Parameters	<table border="1"><thead><tr><th>Name</th><th>Type</th><th>Example</th><th>Description</th></tr></thead><tbody><tr><td>SSHServer</td><td>STRING</td><td>totally.awesome.com</td><td>IP or FQDN of server.</td></tr><tr><td>SSHPort</td><td>NUMBER</td><td>22</td><td>Port ssh is running on.</td></tr></tbody></table>			Name	Type	Example	Description	SSHServer	STRING	totally.awesome.com	IP or FQDN of server.	SSHPort	NUMBER	22	Port ssh is running on.
Name	Type	Example	Description												
SSHServer	STRING	totally.awesome.com	IP or FQDN of server.												
SSHPort	NUMBER	22	Port ssh is running on.												

Optional Parameters	Name	Type	Example	Description
	username	STRING	jsmith	Username on target server.
	password	STRING	password123	Password for user.
	directory	STRING	/tmp	Directory to attempt to access after successful login.
Example URL	/api/v1/api.php?target=auth&action=testSSH&username=jsmith&password=password123&directory=%2Fvar%2Fnamed%2F6connect%2Fq4&SSHPort=22			

testLDAP				
URL	/api/v1/api.php?target=auth&action=testSSH			
Description	Test basic connectivity to an LDAP server. Does not test actual authentication against server.			
Returns	Examples: SUCCESSFUL: <code>{'success':1, 'id':'12345'}</code> ERROR: <code>{'success':0, 'message':'unable to add block'}</code> >			
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	<p>Examples:</p> <table border="1"> <tr> <td>SUCCESSFUL</td> <td>{"success":1,"message":"Success!"}</td> </tr> <tr> <td>ERROR</td> <td>{'success':0, 'message':'error message'}</td> </tr> </table>	SUCCESSFUL	{"success":1,"message":"Success!"}	ERROR	{'success':0, 'message':'error message'}																
SUCCESSFUL	{"success":1,"message":"Success!"}																				
ERROR	{'success':0, 'message':'error message'}																				
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>SSHServer</td> <td>STRING</td> <td>totally.awesome.com</td> <td>IP or FQDN of server.</td> </tr> <tr> <td>SSHPort</td> <td>NUMBER</td> <td>22</td> <td>Port ssh is running on.</td> </tr> </tbody> </table>	Name	Type	Example	Description	SSHServer	STRING	totally.awesome.com	IP or FQDN of server.	SSHPort	NUMBER	22	Port ssh is running on.								
Name	Type	Example	Description																		
SSHServer	STRING	totally.awesome.com	IP or FQDN of server.																		
SSHPort	NUMBER	22	Port ssh is running on.																		
Optional Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>username</td> <td>STRING</td> <td>jsmith</td> <td>Username on target server.</td> </tr> <tr> <td>password</td> <td>STRING</td> <td>password123</td> <td>Password for user.</td> </tr> <tr> <td>directory</td> <td>STRING</td> <td>/tmp</td> <td>Directory to attempt to access after successful login.</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Example	Description	username	STRING	jsmith	Username on target server.	password	STRING	password123	Password for user.	directory	STRING	/tmp	Directory to attempt to access after successful login.				
Name	Type	Example	Description																		
username	STRING	jsmith	Username on target server.																		
password	STRING	password123	Password for user.																		
directory	STRING	/tmp	Directory to attempt to access after successful login.																		
Example URL	/api/v1/api.php?target=auth&action=testSecure64&username=jsmith&password=password123&directory=%2Fvar%2Fnamed%2Fconnect%2Fqa4&SSHPort=22																				

Log Management

Log Management



Get	
URL	/api/v1/api.php?target=log&action=get

Description

Returns a list of log entries. Use optional parameters to filter the list.

Returns

Examples:

SUCCESSFUL	<pre>{ "success": 1, "message": "Search Successful.", "data": { "logId": "31568", "time": "2012-05-07 17:44:43", "logLevel": "INFO", "userId": "39", "userName": "anna@6connect.com", "logCategory": "User", "message": "Anna Claiborne (anna@6connect.com) logged in via local authentication", "ip": "107.111.0.228" } }</pre>
ERROR	<pre>{ "success": 0, "message": "error message" }</pre>

Data Detail

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

Required Parameters

None

Optional Parameters

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

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

Zone Templates

Zone Templates



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

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

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

Required Parameters

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

Optional Parameters

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

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

Example URL

```
api/v1/api.php?target=zoneTemplate&action=update&templateId=1011&count_records=1&name=Awesome+Template&soa=ns1.test.net+hostmaster.ns1.test.net&refresh=14400&retry=3600&expire=604800&minimum=3600&value_0=undefined+undefined&host_1=www&t1_1=3600&type_1=A&value_1=1.2.3.4
```

Delete

URL

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

Description

Deletes a DNS template.

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

API Module - DHCP

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

DHCP Server Control

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

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

Data Detail:

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

Optional Parameters

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

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

update

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

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

Returns

Examples:

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

Optional Parameters

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

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

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

add

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

Description Adds a new DHCP Server.

Returns

Examples:

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

Data Detail:

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

Required Parameters

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

Optional Parameters

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

delete

URL

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

Description

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

Returns

Examples:

SUCCESSFUL:

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

ERROR:

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

Optional Parameters

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

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

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

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

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

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

DHCP Entry Control

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

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

Optional Parameters

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

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

update

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

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

Returns

Examples:

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

Optional Parameters

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

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

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

updateOption

URL

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

Description

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

Returns

Examples:

SUCCESSFUL:

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

ERROR:

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

Optional Parameters

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

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

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

add

URL

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

Description

Adds a new DHCP Entry and returns the new ID.

Returns	<p>Examples:</p> <table border="1" data-bbox="824 186 1481 352"> <tr> <td>SUCCESSFUL:</td> <td><code>{"success":1,"message":"Add Successful. ","data":123}</code></td> </tr> <tr> <td>ERROR:</td> <td><code>{"success":0, "message":"error message"}</code></td> </tr> </table> <p>Data Detail:</p> <table border="1" data-bbox="824 451 1481 585"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>data</td> <td>INT</td> <td>The ID of the new DHCP Entry.</td> </tr> </tbody> </table>	SUCCESSFUL:	<code>{"success":1,"message":"Add Successful. ","data":123}</code>	ERROR:	<code>{"success":0, "message":"error message"}</code>	Name	Type	Description	data	INT	The ID of the new DHCP Entry.						
SUCCESSFUL:	<code>{"success":1,"message":"Add Successful. ","data":123}</code>																
ERROR:	<code>{"success":0, "message":"error message"}</code>																
Name	Type	Description															
data	INT	The ID of the new DHCP Entry.															
Required Parameters	<table border="1" data-bbox="824 646 1481 1066"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>EntryServerId</td> <td>INT</td> <td>123</td> <td>The DHCP Server this new Entry belongs to.</td> </tr> <tr> <td>EntryType</td> <td>STRING</td> <td>subnet</td> <td>The Entry type of this new Entry.</td> </tr> <tr> <td>EntryName</td> <td>STRING</td> <td>30.20.10.1</td> <td>The name of this new Entry.</td> </tr> </tbody> </table>	Name	Type	Example	Description	EntryServerId	INT	123	The DHCP Server this new Entry belongs to.	EntryType	STRING	subnet	The Entry type of this new Entry.	EntryName	STRING	30.20.10.1	The name of this new Entry.
Name	Type	Example	Description														
EntryServerId	INT	123	The DHCP Server this new Entry belongs to.														
EntryType	STRING	subnet	The Entry type of this new Entry.														
EntryName	STRING	30.20.10.1	The name of this new Entry.														
Optional Parameters	<table border="1" data-bbox="824 1127 1481 1404"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>EntryParent</td> <td>INT</td> <td>123</td> <td>The parent Entry to search for.</td> </tr> <tr> <td>EntryNetmask</td> <td>STRING</td> <td>255.255.255.0</td> <td>The subnet mask of this new Entry.</td> </tr> </tbody> </table>	Name	Type	Example	Description	EntryParent	INT	123	The parent Entry to search for.	EntryNetmask	STRING	255.255.255.0	The subnet mask of this new Entry.				
Name	Type	Example	Description														
EntryParent	INT	123	The parent Entry to search for.														
EntryNetmask	STRING	255.255.255.0	The subnet mask of this new Entry.														

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

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

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

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

Optional Parameters

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

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

deleteOption

URL

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

Description

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

Returns

Examples:

SUCCESSFUL:

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

ERROR:

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

Optional Parameters

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

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

API Module - DNS

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

- Name Server Control

DNS Server Control

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

Name	Type	Description
id	INT	Server ID
server	STRING	Server Name
username	STRING	Login Name
password	CRYPT	Login Password
port	INT	Port the Server listens on
zoneCount	INT	The number of zones attached to this server.
options	JSON	<p>The options entry is a JSON-encoded string containing a variety of server-specific configuration options.</p> <p>This string will vary widely by server type and configuration. The following are a selection of common settings.</p>
transfer_type	STRING	The type of server this is. Valid settings include SCP, PowerDNS, Secure64, Secure64Signer
server_type	STRING	Whether this server is a master or a slave server
SOA	STRING	The SOA entry to be used for zones on this server
remote_directory	STRING	The directory where SCP will place the zone files.
named_conf_path	STRING	The 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	INT	Whether or not Views are enabled
views	JSON	The views entry is a JSON-encoded string containing all the information about the Views attached to this server, if any exist.
id	INT	The View ID
server_id	INT	The ID of the server the View is attached to
name	STRING	The name of the View
description	STRING	A description of the View
timestamp	INT	The UNIX timestamp of when the view was created.
extras	JSON	A JSON-encoded array of the extra attributes printed out in the view definition in the config file.

Optional Parameters

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

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
username	STRING	bobuser	Login name for Server
password	STRING	password1	Login password for Server
transferType	STRING	SCP	Protocol used for transfer of DNS zones and records
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

Optional Parameters

There 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	INT	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		For use with Dyn dns service
SOA	STRING	ns1.6connect.com. hostmaster.6connect.com.	Server of Authority record for DNS server
active	INT	0	Values 0 or 1 only, sets the server to inactive on 0 value
powerDNSBackend	STRING	Bind or MySQL	pDNS server backend type
dbDatabaseName	STRING	pdns_1	DB name for pDNS servers with MySQL powerDNSBackend type
dbPort	INT	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

delete									
URL	/api/v1/api.php?target=dnsServer&action=delete								
Description	Deletes a DNS Server								
Returns	<p>Examples:</p> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td>{"success":1,"message":"Delete Successful."}</td> </tr> <tr> <td>ERROR:</td> <td>{"success":0, "message":"error message"}</td> </tr> </table>	SUCCESSFUL:	{"success":1,"message":"Delete Successful."}	ERROR:	{"success":0, "message":"error message"}				
SUCCESSFUL:	{"success":1,"message":"Delete Successful."}								
ERROR:	{"success":0, "message":"error message"}								
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>NUMERIC</td> <td>5</td> <td>ID of server to delete.</td> </tr> </tbody> </table>	Name	Type	Example	Description	id	NUMERIC	5	ID of server to delete.
Name	Type	Example	Description						
id	NUMERIC	5	ID of server to delete.						
Example URL	/api/v1/api.php?target=dnsServer&action=delete&id=5								

update																	
URL	/api/v1/api.php?target=dnsServer&action=update																
Description	Updates an existing DNS Server with new information.																
Returns	<p>Examples:</p> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td>{"success":1,"message":"Update Successful."}</td> </tr> <tr> <td>ERROR:</td> <td>{"success":0, "message":"error message"}</td> </tr> </table>	SUCCESSFUL:	{"success":1,"message":"Update Successful."}	ERROR:	{"success":0, "message":"error message"}												
SUCCESSFUL:	{"success":1,"message":"Update Successful."}																
ERROR:	{"success":0, "message":"error message"}																
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>username</td> <td>STRING</td> <td>bobuser</td> <td>Login name for Server</td> </tr> <tr> <td>password</td> <td>STRING</td> <td>password1</td> <td>Login password for Server</td> </tr> <tr> <td>transferType</td> <td>STRING</td> <td>SCP</td> <td>Protocol</td> </tr> </tbody> </table>	Name	Type	Example	Description	username	STRING	bobuser	Login name for Server	password	STRING	password1	Login password for Server	transferType	STRING	SCP	Protocol
Name	Type	Example	Description														
username	STRING	bobuser	Login name for Server														
password	STRING	password1	Login password for Server														
transferType	STRING	SCP	Protocol														

Optional Parameters	<p>There optional parameters vary according to what type of server is being configured.</p> <table border="1" data-bbox="824 218 1479 833"> <thead> <tr> <th data-bbox="829 218 987 268">Name</th> <th data-bbox="992 218 1149 268">Type</th> <th data-bbox="1154 218 1312 268">Example</th> <th data-bbox="1317 218 1474 268">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="829 275 987 352">customerName</td> <td data-bbox="992 275 1149 352">STRING</td> <td data-bbox="1154 275 1312 352">/tmp/zones</td> <td data-bbox="1317 275 1474 352">Customer Name</td> </tr> <tr> <td data-bbox="829 359 987 464">remoteDirectory</td> <td data-bbox="992 359 1149 464">STRING</td> <td data-bbox="1154 359 1312 464">/tmp/zones</td> <td data-bbox="1317 359 1474 464">Zone Directory on Server</td> </tr> <tr> <td data-bbox="829 470 987 667">namedConfPath</td> <td data-bbox="992 470 1149 667">STRING</td> <td data-bbox="1154 470 1312 667">/tmp</td> <td data-bbox="1317 470 1474 667">The path to the zone files used within the named.conf file.</td> </tr> <tr> <td data-bbox="829 674 987 751">preCommand</td> <td data-bbox="992 674 1149 751">STRING</td> <td data-bbox="1154 674 1312 751">/path/to/stuff/precommand</td> <td data-bbox="1317 674 1474 751"></td> </tr> <tr> <td data-bbox="829 758 987 833">postCommand</td> <td data-bbox="992 758 1149 833">STRING</td> <td data-bbox="1154 758 1312 833">/path/to/stuff/postcommand</td> <td data-bbox="1317 758 1474 833"></td> </tr> </tbody> </table>	Name	Type	Example	Description	customerName	STRING	/tmp/zones	Customer Name	remoteDirectory	STRING	/tmp/zones	Zone Directory on Server	namedConfPath	STRING	/tmp	The path to the zone files used within the named.conf file.	preCommand	STRING	/path/to/stuff/precommand		postCommand	STRING	/path/to/stuff/postcommand	
Name	Type	Example	Description																						
customerName	STRING	/tmp/zones	Customer Name																						
remoteDirectory	STRING	/tmp/zones	Zone Directory on Server																						
namedConfPath	STRING	/tmp	The path to the zone files used within the named.conf file.																						
preCommand	STRING	/path/to/stuff/precommand																							
postCommand	STRING	/path/to/stuff/postcommand																							

transferAll									
URL	/api/v1/api.php?target=dnsServer&action=transferServer								
Description	Performs a full zone push on a DNS Server, executing pre and post commands, transferring files, and restarting services.								
Returns	<p>Examples:</p> <table border="1" data-bbox="824 1129 1479 1297"> <tbody> <tr> <td data-bbox="829 1136 1149 1213">SUCCESSFUL:</td> <td data-bbox="1154 1136 1474 1213">{"success":1,"message":"Transfer Successful."}</td> </tr> <tr> <td data-bbox="829 1220 1149 1297">ERROR:</td> <td data-bbox="1154 1220 1474 1297">{"success":0,"message":"error message"}</td> </tr> </tbody> </table>	SUCCESSFUL:	{"success":1,"message":"Transfer Successful."}	ERROR:	{"success":0,"message":"error message"}				
SUCCESSFUL:	{"success":1,"message":"Transfer Successful."}								
ERROR:	{"success":0,"message":"error message"}								
Required Parameters	<table border="1" data-bbox="824 1360 1479 1528"> <thead> <tr> <th data-bbox="829 1360 987 1411">Name</th> <th data-bbox="992 1360 1149 1411">Type</th> <th data-bbox="1154 1360 1312 1411">Example</th> <th data-bbox="1317 1360 1474 1411">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="829 1417 987 1522">push</td> <td data-bbox="992 1417 1149 1522">INT</td> <td data-bbox="1154 1417 1312 1522">1</td> <td data-bbox="1317 1417 1474 1522">The ID of the server to push zones to</td> </tr> </tbody> </table>	Name	Type	Example	Description	push	INT	1	The ID of the server to push zones to
Name	Type	Example	Description						
push	INT	1	The ID of the server to push zones to						

transferSingle	
URL	/api/v1/api.php?target=dnsServer&action=transferSingle
Description	<p>Transfers a single Zone file to all its associated DNS Servers, along with updated server configurations.</p> <p>Performs pre and post commands on the target servers, transfers the zone file(s), and restarts services.</p>

Returns	<p>Examples:</p> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td>{ "success":1, "message": "Updated Zone: \$name.zone on \$server via SCP" }</td> </tr> <tr> <td>ERROR:</td> <td>{ "success":0, "message": "error message" }</td> </tr> </table>	SUCCESSFUL:	{ "success":1, "message": "Updated Zone: \$name.zone on \$server via SCP" }	ERROR:	{ "success":0, "message": "error message" }				
SUCCESSFUL:	{ "success":1, "message": "Updated Zone: \$name.zone on \$server via SCP" }								
ERROR:	{ "success":0, "message": "error message" }								
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>zoneID</td> <td>NUMERIC</td> <td>35</td> <td>The ID of the zone to push</td> </tr> </tbody> </table>	Name	Type	Example	Description	zoneID	NUMERIC	35	The ID of the zone to push
Name	Type	Example	Description						
zoneID	NUMERIC	35	The ID of the zone to push						

DNS Zone Control

get					
URL	/api/v1/api.php?target=zone&action=get				
Description	<p>Accepts search criteria to retrieve a list of all matching DNS Zones and associated Records.</p> <p>Search can be performed on any combination of Zone and Record attributes.</p>				
Returns	<p>Examples:</p> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td>{ "success":1, "message": "Search Successful.", "data": { "zoneID": "932", "zoneName": "185.160.209.in-addr.arpa", "zoneResourceID": "81", "zoneSerial": "2013040302", "zoneRefresh": "28800", "zoneRetry": "7200", "zoneExpire": "604800", "zoneMinimum": "86400", "zoneSOA": null, "zoneTags": null, "zoneTTL": "28800", "zoneAutoCheck": "1", "zoneEnableDNSSEC": null, "recordID": "154110", "recordZoneID": "932", "recordHost": "185.160.209.inaddr.arpa.", "recordType": "NS", "recordValue": "auth01.verosity.net.", "recordDescription": null, "recordTTL": "28800", "recordOrdering": "1", "recordErrors": null, "assetID": "0", "userCanCreate": 0, "userCanDelete": 1, "userCanUpdate": 1 } }</td> </tr> <tr> <td>ERROR:</td> <td>{ "success":0, "message": "error message" }</td> </tr> </table>	SUCCESSFUL:	{ "success":1, "message": "Search Successful.", "data": { "zoneID": "932", "zoneName": "185.160.209.in-addr.arpa", "zoneResourceID": "81", "zoneSerial": "2013040302", "zoneRefresh": "28800", "zoneRetry": "7200", "zoneExpire": "604800", "zoneMinimum": "86400", "zoneSOA": null, "zoneTags": null, "zoneTTL": "28800", "zoneAutoCheck": "1", "zoneEnableDNSSEC": null, "recordID": "154110", "recordZoneID": "932", "recordHost": "185.160.209.inaddr.arpa.", "recordType": "NS", "recordValue": "auth01.verosity.net.", "recordDescription": null, "recordTTL": "28800", "recordOrdering": "1", "recordErrors": null, "assetID": "0", "userCanCreate": 0, "userCanDelete": 1, "userCanUpdate": 1 } }	ERROR:	{ "success":0, "message": "error message" }
SUCCESSFUL:	{ "success":1, "message": "Search Successful.", "data": { "zoneID": "932", "zoneName": "185.160.209.in-addr.arpa", "zoneResourceID": "81", "zoneSerial": "2013040302", "zoneRefresh": "28800", "zoneRetry": "7200", "zoneExpire": "604800", "zoneMinimum": "86400", "zoneSOA": null, "zoneTags": null, "zoneTTL": "28800", "zoneAutoCheck": "1", "zoneEnableDNSSEC": null, "recordID": "154110", "recordZoneID": "932", "recordHost": "185.160.209.inaddr.arpa.", "recordType": "NS", "recordValue": "auth01.verosity.net.", "recordDescription": null, "recordTTL": "28800", "recordOrdering": "1", "recordErrors": null, "assetID": "0", "userCanCreate": 0, "userCanDelete": 1, "userCanUpdate": 1 } }				
ERROR:	{ "success":0, "message": "error message" }				

Data Detail:

Name	Type	Description
zoneld	INT	The Id of the Zone entry. A single Zone entry might have multiple Records.
zoneName	STRING	The Zone name.
zoneResourceId	INT	The resource Id associated with this Zone.
zoneSerial	INT	Zone Serial.
zoneRefresh	INT	Zone Refresh.
zoneRetry	INT	Zone Retry.
zoneExpire	INT	Zone Expire.
zoneMinimum	INT	Zone Minimum.
zoneSOA	STRING	Zone SOA.
zoneTags	STRING	All the tags associated with this Zone.
zoneTTL	STRING	Zone TTL.
zoneEnableDNSSEC	BOOL	Whether or not DNSSEC is enabled for this Zone.
zoneAutoCheck	BOOL	Whether or not this zone is configured to be automatically validated on load/edit.
recordId	INT	The Id of this Record Entry. It is always included with its parent Zone.
recordHost	STRING	The Hostname of this Record.
recordType	STRING	The Record Type (MX,NS,A,PTR,etc)
recordValue	STRING	The Value of this Record.
recordDescription	STRING	A short description of this Record.

recordTTL	STRING	The TTL of this Record.
recordOrdering	INT	The numerical order in which the record appears in the zone.
recordErrors	STRING	A string containing any detected problems with this record
userCanCreate	BOOL	Whether or not the user has DNS CREATE permissions on this zone's resource
userCanUpdate	BOOL	Whether or not the user has DNS UPDATE permissions on this zone's resource
userCanDelete	BOOL	Whether or not the user has DNS DELETE permissions on this zone's resource
unpagedRows	INT	If pagination is used, this value will contain a total count of records had the pagination not been used.

Optional Parameters

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

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

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

search

URL

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

Description

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

Returns

Examples:

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

Data Detail:

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

zoneEnableDNSSEC	BOOL	Whether or not DNSSEC is enabled for this Zone.
zoneAutoCheck	BOOL	Whether or not this zone is configured to be automatically validated on load/edit.
recordCount	int	How many records are associated with this zone.
userCanCreate	BOOL	Whether or not the user has DNS CREATE permissions on this zone's resource
userCanUpdate	BOOL	Whether or not the user has DNS UPDATE permissions on this zone's resource
userCanDelete	BOOL	Whether or not the user has DNS DELETE permissions on this zone's resource
unpagedRows	INT	If pagination is used, this value will contain a total count of records had the pagination not been used.

Optional Parameters

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

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

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

update

URL

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

Description

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

Returns

Examples:

SUCCESSFUL:

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

ERROR:

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

Optional Parameters

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

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

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

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

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

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

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

Optional Parameters

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

delete

URL

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

Description

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

Returns

Examples:

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

Optional Parameters

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

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

getRecordTypes

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

getFile

URL	/api/v1/api.php?target=zone&action=getFile&zoneId=50								
Description	Returns a fully written zone file. If one does not exist, returns false.								
Returns	A Zone File								
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>zoneId</td> <td>INT</td> <td>50</td> <td>The Id of the zone to retrieve.</td> </tr> </tbody> </table>	Name	Type	Example	Description	zoneId	INT	50	The Id of the zone to retrieve.
Name	Type	Example	Description						
zoneId	INT	50	The Id of the zone to retrieve.						

getDSFile

URL	/api/v1/api.php?target=zone&action=getDSFile&zoneId=50
-----	--

Description	Returns a fully written zone DS key file. If one does not exist, returns false.								
Returns	A Zone DS Key File								
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>zoneld</td> <td>INT</td> <td>50</td> <td>The Id of the zone whose DS keys wer are to retrieve.</td> </tr> </tbody> </table>	Name	Type	Example	Description	zoneld	INT	50	The Id of the zone whose DS keys wer are to retrieve.
Name	Type	Example	Description						
zoneld	INT	50	The Id of the zone whose DS keys wer are to retrieve.						

checkZone

URL	/api/v1/api.php?target=zone&action=checkZone&zoneld=50								
Description	Runs a zone file through Named checkzone								
Returns	<p>Examples:</p> <table border="1"> <tbody> <tr> <td>SUCCESSFUL:</td> <td>{"success":1,"message":"No errors found."}</td> </tr> <tr> <td>ERROR:</td> <td>{"success":0,"message":"21: ignoring out-of-zone data (veggie.com) 22: ignoring out-of-zone data (veggie.com) dns_rdata_fromtext: 23: near '2001::db8:': bad IPv6 address dns_rdata_fromtext: 24: near '1.2.3.': bad dotted quad dns_rdata_fromtext: 25: near '2001::db8::\32': bad IPv6 address "}</td> </tr> </tbody> </table>	SUCCESSFUL:	{"success":1,"message":"No errors found."}	ERROR:	{"success":0,"message":"21: ignoring out-of-zone data (veggie.com) 22: ignoring out-of-zone data (veggie.com) dns_rdata_fromtext: 23: near '2001::db8:': bad IPv6 address dns_rdata_fromtext: 24: near '1.2.3.': bad dotted quad dns_rdata_fromtext: 25: near '2001::db8::\32': bad IPv6 address "}				
SUCCESSFUL:	{"success":1,"message":"No errors found."}								
ERROR:	{"success":0,"message":"21: ignoring out-of-zone data (veggie.com) 22: ignoring out-of-zone data (veggie.com) dns_rdata_fromtext: 23: near '2001::db8:': bad IPv6 address dns_rdata_fromtext: 24: near '1.2.3.': bad dotted quad dns_rdata_fromtext: 25: near '2001::db8::\32': bad IPv6 address "}								
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>zoneld</td> <td>INT</td> <td>50</td> <td>The Id of the zone to check.</td> </tr> </tbody> </table>	Name	Type	Example	Description	zoneld	INT	50	The Id of the zone to check.
Name	Type	Example	Description						
zoneld	INT	50	The Id of the zone to check.						

getArchivedZone

URL	/api/v1/api.php?target=zone&action=getArchivedZone
Description	Searches for all archived versions of the a zone. Zones are archived every time changes are pushed to their DNS Server.
Returns	<p>Examples:</p>

SUCCESSFUL:	<pre>{ "success": 1, "message": "Search Successful.", "data": { "zoneArchiveId": "2768", "zoneId": "1227", "zoneArchiveTimestamp": "1375298692", "zoneArchiveFingerprint": "d060e59d69606326d80b2e55b50f0bc9", "zoneName": "6connect.com", "zoneIpv": null, "zoneMask": null, "zoneSerial": "2013073105", "zoneRefresh": "14400", "zoneRetry": "2000", "zoneExpire": "604800", "zoneMinimum": "3600", "zoneSOA": null, "zoneTags": null, "zoneTTL": "3600", "zoneEnableDNSSEC": "1", "zoneResourceId": "1013", "zonePreviousViewLinkage": [] } }</pre>
ERROR:	<pre>{ "success": 0, "message": "error message" }</pre>

Data Detail:

Name	Type	Description
zoneId	INT	The Id of the Zone entry to find archived versions of.
zoneArchiveId	INT	The ID of the Archive Entry
zoneArchiveTimestamp	INT	A timestamp marking when this zone was archived.
zoneArchiveFingerprint	STRING	A hash value identifying this zone. Used for comparing versions.
zoneName	INT	Zone Name.
zoneMask	INT	Zone Mask.
zoneSerial	INT	Zone Serial.
zoneRefresh	INT	Zone Refresh.
zoneRetry	INT	Zone Retry.
zoneExpire	INT	Zone Expire.
zoneMinimum	INT	Zone Minimum.
zoneSOA	STRING	Zone SOA.
zoneTags	STRING	Zone Tags.
zoneTTL	INT	Zone TTL.
zoneEnableDNSSEC	STRING	Whether or not this version had DNSSEC enabled.
zoneResourceId	STRING	Zone Resource ID
zonePreviousViewLinkage	JSON	A JSON-encoded array of views this zone was linked to.

Optional Parameters

Name	Type	Example	Description
zoneId	INT	123	The Zone Id to search for.
zoneArchiveId	INT	123	The Zone Archive Id
zoneArchiveTimestamp	INT	2012033001	The Zone Archive Timestamp
fetchArchiveFile	BOOL	1	Whether or not to return the full Zone file with the result set..

DNS Record Control

get

URL

/api/v1/api.php?target=record&action=get

Description

Accepts search criteria to retrieve a list of all matching DNS Records. Search can be performed on any combination of Zone and Record attributes.

Returns

Examples:

SUCCESSFUL:	<pre>{"success":1,"message":"Search Successful. ","data":{"recordId":"30894","recordZoneld":"229","recordHost":"@", "recordType":"NS","recordValue":"ns1.domain.com.", "recordDescription":"","recordTTL":""}}</pre>
ERROR:	<pre>{"success":0, "message":"error message"}</pre>

Data Detail:

Name	Type	Description
recordId	INT	The ID of this Record Entry. It is always included with its parent Zone.
recordZoneld	INT	The ID of this Record's parent Zone.
recordHost	STRING	The Hostname of this Record.
recordType	STRING	The Record Type (MX,NS,A,PTR,etc)
recordValue	STRING	The Value of this Record.
recordDescription	STRING	A short description of this Record.
recordTTL	STRING	The TTL of this Record.

Optional Parameters

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

Name	Type	Example	Description
recordId	INT	123	The Record ID to search for.
recordZoneId	INT	123	The parent Zone to search for.
recordHost	STRING	@	The Record Host to search for.
recordType	STRING	NS	The Record Type to search for.
recordValue	STRING	ns1.dns.6connect.com.	The Record Value to search for.
recordDescription	STRING	Description	Search based on Record Description.
recordTTL	STRING	3600	The Record TTL to search for.

update

URL

/api/v1/api.php?target=record&action=update

Description

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

Returns

Examples:

SUCCESSFUL:

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

ERROR:

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

Optional Parameters

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

Name	Type	Example	Description
searchZoneId	INT	123	The Zone ID to search for.
searchZoneName	STRING	foo	The Zone Name to search for.
searchZoneCustomerId	INT	5	The Customer ID to search for.
searchZoneIpsVersion	STRING	IPv6	The IP Version to search for.
searchZoneMask	STRING		The Zone Mask to search for.
searchZoneSerial	INT	2012033001	The Zone Serial to search for.
searchZoneRefresh	INT	36000	The Zone Refresh to search for.
searchZoneRetry	INT	800	The Zone Retry to search for.
searchZoneExpire	INT	6090000	The Zone Expire to search for.
searchZoneMinimum	INT	10	The Zone Minimum to search for.

searchZoneSOA	STRING	200	The Zone SOA to search for.
searchZoneTags	STRING	client,production	Zone Tags to search for.
searchZoneTTL	INT	3600	The Zone TTL to search for.
searchZoneEnableDNSSEC	INT	1	Search based on DNSSEC settings.
searchRecordId	INT	123	The Record ID to search for.
searchRecordHost	STRING	@	The Record Host to search for.
searchRecordType	STRING	NS	The Record Type to search for.
searchRecordValue	STRING	ns1.dns.6connect.com.	The Record Value to search for.
searchRecordDescription	STRING	Description	Search based on Record Description.
searchRecordTTL	STRING	3600	The Record TTL to search for.

Name	Type	Example	Description
updateZoneName	STRING	foo	The Zone name to replace into the searched rows.
updateZoneCustId	INT	5	The Customer ID to replace into the searched rows.
updateZoneIpver	STRING	IPv6	The IP Version to replace into the searched rows.

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

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

add

URL `/api/v1/api.php?target=record&action=add`

Description Adds a new Record to a supplied Zone.

Returns

Examples:

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

Data Detail:

Name	Type	Description
data	INT	The ID of the new Record entry.

Required Parameters

Name	Type	Example	Description
newRecordZoneId	INT	123	The Zone ID of the new Record.
newRecordHost	STRING	@	New Host Name.
newRecordType	STRING	PTR	New Record Type.
newRecordValue	STRING	123	New Record Value.

Optional Parameters	Name	Type	Example	Description
	newRecordDescription	STRING	Description.	Notes for the Record.
	newRecordTTL	INT	foo	Record TTL.

delete

URL: /api/v1/api.php?target=record&action=delete

Description: Performs a search over the Zones and Records dataset and deletes all found Records, but leaves their parent Zones intact.

Returns

Examples:

SUCCESSFUL:	{"success":1,"message":"Deletion Successful."}
ERROR:	{"success":0,"message":"error message"}

Optional Parameters

Name	Type	Example	Description
deleteZoneId	INT	123	The Zone ID to search for.
deleteZoneName	STRING	foo	The Zone Name to search for.
deleteZoneCustomerId	INT	5	The Customer ID to search for.
deleteZoneIpVersion	STRING	IPv6	The IP Version to search for.
deleteZoneMask	STRING		The Zone Mask to search for.
deleteZoneSerial	INT	2012033001	The Zone Serial to search for.
deleteZoneRefresh	INT	36000	The Zone Refresh to search for.
deleteZoneRetry	INT	800	The Zone Retry to search for.
deleteZoneExpire	INT	6090000	The Zone Expire to search for.

deleteZoneMinimum	INT	10	The Zone Minimum to search for.
deleteZoneSOA	STRING	200	The Zone SOA to search for.
deleteZoneTags	STRING	client,production	Zone Tags to search for.
deleteZoneTTL	INT	3600	The Zone TTL to search for.
deleteZoneEnableDNSSEC	INT	1	Search based on DNSSEC settings.
deleteRecordID	INT	123	The Record ID to search for.
deleteRecordHost	STRING	@	The Record Host to search for.
deleteRecordType	STRING	NS	The Record Type to search for.
deleteRecordValue	STRING	ns1.dns.6connect.com.	The Record Value to search for.
deleteRecordDescription	STRING	Description	Search based on Record Description.
deleteRecordTTL	STRING	3600	The Record TTL to search for.

switch

URL /api/v1/api.php?target=record&action=switch

Description Switches the order of two record entries.

Returns

Examples:

SUCCESSFUL:	{"success":1,"message":"Record Moved."}
ERROR:	{"success":0, "message":"error message"}

Required Parameters

Name	Type	Example	Description
moveWhichId	INT	123	The Record Id to be moved.
moveAfterId	INT	@	The Id of the Record the first Record is to be moved after.

Server-Zone Linkage

get

URL

/api/v1/api.php?target=zoneLinkage&action=get

Description

Searches for Server-Zone Linkages. If no search parameters are supplied, all linkages are returned.

Returns

Examples:

SUCCESSFUL:	<pre>{"success":1,"message":"2 rows retrieved.", "data":[{"id":"285", "zoneld":"64", "serverId":"1", "serverName":"173.164.182.169", "serverType":"SCP", "serverMasterType":"master", "zoneName":"bind.com", "resourceId":"483"}, {"id":"287", "zoneld":"371", "serverId":"1", "serverName":"173.164.182.169", "serverType":"SCP", "serverMasterType":"master", "zoneName":"132.235.198.in-addr.arpa", "resourceId":"577"}]}</pre>
ERROR:	<pre>{"success":0, "message":"error message"}</pre>

Data Detail:

Name	Type	Description
id	INT	The Linkage Id.
zoneld	INT	The Zoneld involved in this link.
serverId	INT	The ServerId involved in this link.
serverName	STRING	The server name
serverType	STRING	The server transfer type
serverMasterType	STRING	Whether this server is a master or a slave.
zoneName	STRING	The zone name
resourceId	INT	The Resource Id the Zone is attached to.

Optional Parameters

Name	Type	Example	Description
id	INTEGER	15	Fetches the linkage with the matching id.
serverId	INTEGER	15	Fetches all linkages with the matching serverId.
zoneId	INTEGER	15	Fetches all linkages with the matching zoneId.

add

URL

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

Description

Adds a new link between a DNS Server and a Zone

Returns

Examples:

SUCCESSFUL:	{"success":1,"message":"Link Added."}
ERROR:	{"success":0, "message":"error message"}

Required Parameters

Name	Type	Example	Description
serverId	INT	16	The DNS Server Id.
zoneId	INT	105	The Zone Id.

delete

URL

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

Description

Deletes a link between a DNS Server and a Zone

Returns

Examples:

SUCCESSFUL:	{"success":1,"message":"Link Deleted."}
ERROR:	{"success":0, "message":"error message"}

Optional Parameters

Name	Type	Example	Description
id	INTEGER	15	Fetches the linkage with the matching id.
serverId	INTEGER	15	Fetches all linkages with the matching serverId.
zoneId	INTEGER	15	Fetches all linkages with the matching zoneId.

Name Server Control

get

URL

/api/v1/api.php?target=nameServer&action=get

Description

Fetches a list of all stored Name Servers

Returns

Examples:

SUCCESSFUL:	<pre>{ "success": 1, "message": "Fetch Successful.", "data": [{"id": "1", "nameserver": "ns1.dns.6connect.net", "add_to_zones_default": "1", "ordering": "10", "uses": "34"}, {"id": "10", "nameserver": "ns2.dns.6connect.net", "add_to_zones_default": "1", "ordering": "11", "uses": "46"}, {"id": "9", "nameserver": "ns4.dns.6connect.net", "add_to_zones_default": "1", "ordering": "14", "uses": "12"}, {"id": "3", "nameserver": "ns3.dns.6connect.net", "add_to_zones_default": "1", "ordering": "15", "uses": "46"}]}</pre>
ERROR:	<pre>{ "success": 0, "message": "error message" }</pre>

Data Detail:

Name	Type	Description
id	INT	Server ID
nameserver	STRING	Server Name
add_to_zones_default	BOOL	Whether or not this is a default server.
ordering	INT	Display order
uses	INT	How many zones have been assigned to this NameServer

add

URL

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

Description

Adds a new DNS Server

Returns

Examples:

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

Required Parameters

Name	Type	Example	Description
newServer	STRING	ns.yourdomain.com	Name of the NameServer

delete									
URL	/api/v1/api.php?target=nameServer&action=delete								
Description	Deletes a NameServer								
Returns	<p>Examples:</p> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td><i>{"success":1,"message":"Server Deleted."}</i></td> </tr> <tr> <td>ERROR:</td> <td><i>{"success":0, "message":"error message"}</i></td> </tr> </table>	SUCCESSFUL:	<i>{"success":1,"message":"Server Deleted."}</i>	ERROR:	<i>{"success":0, "message":"error message"}</i>				
SUCCESSFUL:	<i>{"success":1,"message":"Server Deleted."}</i>								
ERROR:	<i>{"success":0, "message":"error message"}</i>								
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>NUMERIC</td> <td>5</td> <td>ID of server to delete.</td> </tr> </tbody> </table>	Name	Type	Example	Description	id	NUMERIC	5	ID of server to delete.
Name	Type	Example	Description						
id	NUMERIC	5	ID of server to delete.						
Example URL	/api/v1/api.php?target=nameServer&action=delete&id=5								

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	<p>Examples:</p> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td><i>{"success":1, "message":"Success."}</i></td> </tr> <tr> <td>ERROR:</td> <td><i>{"success":0, "message":"error message"}</i></td> </tr> </table>	SUCCESSFUL:	<i>{"success":1, "message":"Success."}</i>	ERROR:	<i>{"success":0, "message":"error message"}</i>								
SUCCESSFUL:	<i>{"success":1, "message":"Success."}</i>												
ERROR:	<i>{"success":0, "message":"error message"}</i>												
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>NUMERIC</td> <td>5</td> <td>ID of server to modify.</td> </tr> <tr> <td>value</td> <td>1 or 0</td> <td>1</td> <td>1 = Default, 0 = Normal</td> </tr> </tbody> </table>	Name	Type	Example	Description	id	NUMERIC	5	ID of server to modify.	value	1 or 0	1	1 = Default, 0 = Normal
Name	Type	Example	Description										
id	NUMERIC	5	ID of server to modify.										
value	1 or 0	1	1 = Default, 0 = Normal										
Example URL	/api/v1/api.php?target=nameServer&action=setDefault&id=3&value=1												

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	<p>Examples:</p> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td><i>{"success":1,"message":"Reordering Successful."}</i></td> </tr> <tr> <td>ERROR:</td> <td><i>{"success":0, "message":"error message"}</i></td> </tr> </table>	SUCCESSFUL:	<i>{"success":1,"message":"Reordering Successful."}</i>	ERROR:	<i>{"success":0, "message":"error message"}</i>
SUCCESSFUL:	<i>{"success":1,"message":"Reordering Successful."}</i>				
ERROR:	<i>{"success":0, "message":"error message"}</i>				
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 targetted NameServer with that of the one below it.								
Returns	<p>Examples:</p> <table border="1"> <tr> <td>SUCCESSFUL:</td> <td><i>{"success":1,"message":"Reordering Successful."}</i></td> </tr> <tr> <td>ERROR:</td> <td><i>{"success":0, "message":"error message"}</i></td> </tr> </table>	SUCCESSFUL:	<i>{"success":1,"message":"Reordering Successful."}</i>	ERROR:	<i>{"success":0, "message":"error message"}</i>				
SUCCESSFUL:	<i>{"success":1,"message":"Reordering Successful."}</i>								
ERROR:	<i>{"success":0, "message":"error message"}</i>								
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>NUMERIC</td> <td>5</td> <td>ID of server to activate.</td> </tr> </tbody> </table>	Name	Type	Example	Description	id	NUMERIC	5	ID of server to activate.
Name	Type	Example	Description						
id	NUMERIC	5	ID of server to activate.						
Example URL	/api/v1/api.php?target=nameServer&action=orderDown&id=5								

API Module - IPAM

IP Address Management (IPv4 and IPv6)

- Get
- Add
- Update
- Delete
- Add Tag
- Delete Tag
- Smart Assign
- Direct Assign
- Unassign
- Get Tags List
- Get RIRs List
- Get Regions List
- Get Utilization
- Aggregate
- Split
- Scan Block
- Get Scan Results

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	<p>Examples:</p> <table border="1"> <tr> <td>SUCCESSFUL</td> <td> <pre>{ "success": 1, "message": "1 blocks found. ", "data": [{ "id": 5890, "type": "ipv4", "top_aggregate": null, "cidr": "192.168.0.0V24", "formatted_ip": "192.168.0.0V24", "address": "3232235520", "end_address": "3232235775", "mask": 24, "child1": null, "child2": null, "is_assigned": 0, "is_swipped": 0, "is_aggregate": 1, "custid": 81, "resource_id": 81, "resource_name": "Available", "last_updated_time": null, "description": null, "parent": null, "rir": "1918", "lir_id": null, "notes": null, "generic_code": null, "code": null, "region": "SFO", "vlan": 100, "arin_net_id": null, "arin_cust_id": null, "org_id": null, "arin_swip_time": null, "assigned_time": null, "asn": null, "allowSubAssignments": false, "permissions": { "permissionIPAMRead": "1", "permissionIPAMUpdate": "1", "permissionIPAMCreate": "1", "permissionSWIP": "1", "permissionAdmin": "1" }, "range": "192.168.0.0 - 192.168.0.255", "tags": ["Customer", "PTP"] }] }</pre> </td> </tr> <tr> <td>ERROR</td> <td> <pre>{'success':0, 'message':'error message'}</pre> </td> </tr> </table>	SUCCESSFUL	<pre>{ "success": 1, "message": "1 blocks found. ", "data": [{ "id": 5890, "type": "ipv4", "top_aggregate": null, "cidr": "192.168.0.0V24", "formatted_ip": "192.168.0.0V24", "address": "3232235520", "end_address": "3232235775", "mask": 24, "child1": null, "child2": null, "is_assigned": 0, "is_swipped": 0, "is_aggregate": 1, "custid": 81, "resource_id": 81, "resource_name": "Available", "last_updated_time": null, "description": null, "parent": null, "rir": "1918", "lir_id": null, "notes": null, "generic_code": null, "code": null, "region": "SFO", "vlan": 100, "arin_net_id": null, "arin_cust_id": null, "org_id": null, "arin_swip_time": null, "assigned_time": null, "asn": null, "allowSubAssignments": false, "permissions": { "permissionIPAMRead": "1", "permissionIPAMUpdate": "1", "permissionIPAMCreate": "1", "permissionSWIP": "1", "permissionAdmin": "1" }, "range": "192.168.0.0 - 192.168.0.255", "tags": ["Customer", "PTP"] }] }</pre>	ERROR	<pre>{'success':0, 'message':'error message'}</pre>				
SUCCESSFUL	<pre>{ "success": 1, "message": "1 blocks found. ", "data": [{ "id": 5890, "type": "ipv4", "top_aggregate": null, "cidr": "192.168.0.0V24", "formatted_ip": "192.168.0.0V24", "address": "3232235520", "end_address": "3232235775", "mask": 24, "child1": null, "child2": null, "is_assigned": 0, "is_swipped": 0, "is_aggregate": 1, "custid": 81, "resource_id": 81, "resource_name": "Available", "last_updated_time": null, "description": null, "parent": null, "rir": "1918", "lir_id": null, "notes": null, "generic_code": null, "code": null, "region": "SFO", "vlan": 100, "arin_net_id": null, "arin_cust_id": null, "org_id": null, "arin_swip_time": null, "assigned_time": null, "asn": null, "allowSubAssignments": false, "permissions": { "permissionIPAMRead": "1", "permissionIPAMUpdate": "1", "permissionIPAMCreate": "1", "permissionSWIP": "1", "permissionAdmin": "1" }, "range": "192.168.0.0 - 192.168.0.255", "tags": ["Customer", "PTP"] }] }</pre>								
ERROR	<pre>{'success':0, 'message':'error message'}</pre>								
Optional Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>asn</td> <td>INTEGER</td> <td>1000</td> <td>Filters blocks based on their ASN</td> </tr> </tbody> </table>	Name	Type	Example	Description	asn	INTEGER	1000	Filters blocks based on their ASN
Name	Type	Example	Description						
asn	INTEGER	1000	Filters blocks based on their ASN						

allowSubAssignments	BOOL	true	Filters blocks based on whether they allow sub-assignments or not. Acceptable values: "true" or "false"
block	STRING	213.37.29.0/24	CIDR block description
code	STRING		User-defined block code as defined in Admin-IPAM settings: Generic Code Per Block Name
id	INTEGER	1234	The ID of the block
isAggregate	BOOL	true	Indicates if the block has been split into children or not. A value of 'true' will return blocks with no children.
isAssigned	BOOL	true	Acceptable values: "true" or "false"
isSwapped	BOOL	true	Acceptable values: "true" or "false"
mask	INTEGER	24	Integer bitmask
region	STRING	SFO	The value from the list of name/value pairs which make up the list of available regions

resourceHolderId	STRING	6c-1234	The 20 character resource holder ID. This is an internal short identifier for the customer, and should be used to link resource holder details in the 6Connect database back to your organization. Consider using internal customer numbers or department numbers for this field depending on your use case. Example: Customer-001 or 000213.
resourceId	INTEGER		The ID of the resource the block is assigned to
resourceQuery	JSON	<pre>{"custom_id": "cust-001"}</pre>	A JSON object representing a valid resource query. Any parameters that can be used for a Resource GET API call can be used. Use of the resourceQuery parameter will return blocks assigned to any of the resources returned by that query.

rir	STRING	ARIN	Acceptable values: ARIN, RIPE, APNIC, AfriNIC
search	STRING	192.168	If a search term is provided, all IPAM fields including assigned Resource Holder name will be checked with a LIKE comparison to find matching blocks
selectCount	INTEGER	50	# of blocks to get
selectOffset	INTEGER	25	Offset for results set; useful for paging (e.g. selectCount = 50, selectOffset = 100 would return the 3rd page of 50 results)
sortField	STRING	cidr	Attribute to sort blocks by. Acceptable values: cidr, mask, rir, vlan, code, updateTime
sortOrder	INTEGER	ASC	ASC or DESC
tags	STRING	customer,vpn	Comma-separated list of tags
topAggregateId	INTEGER	1234	The ID of the aggregate block to which the block belongs
vlan	INTEGER	123	VLAN for the block

Example URL

/api/v1/api.php?target=ipam&action=get&rir=ARIN&tags=customer,vpn

Add													
URL	/api/v1/api.php?target=ipam&action=add												
Description	Adds an IPv4 or IPv6 block												
Returns	<p>Examples:</p> <table border="1"> <tr> <td>SUCCESSFUL</td> <td><pre>{"success":1,"message":"Block 192.168.0.0/24 (12345) added", "id":12345, "data":{"id":12345, "cidr":"192.168.0.0/24", ...}}</pre></td> </tr> <tr> <td>ERROR</td> <td><pre>{"success":0, "message":"error message" }</pre></td> </tr> </table>	SUCCESSFUL	<pre>{"success":1,"message":"Block 192.168.0.0/24 (12345) added", "id":12345, "data":{"id":12345, "cidr":"192.168.0.0/24", ...}}</pre>	ERROR	<pre>{"success":0, "message":"error message" }</pre>								
SUCCESSFUL	<pre>{"success":1,"message":"Block 192.168.0.0/24 (12345) added", "id":12345, "data":{"id":12345, "cidr":"192.168.0.0/24", ...}}</pre>												
ERROR	<pre>{"success":0, "message":"error message" }</pre>												
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>block</td> <td>STRING</td> <td>213.37.29.0/24</td> <td>CIDR block description</td> </tr> <tr> <td>rir</td> <td>STRING</td> <td>ARIN</td> <td>Acceptable values: ARIN, RIPE, APNIC, AfriNIC</td> </tr> </tbody> </table>	Name	Type	Example	Description	block	STRING	213.37.29.0/24	CIDR block description	rir	STRING	ARIN	Acceptable values: ARIN, RIPE, APNIC, AfriNIC
Name	Type	Example	Description										
block	STRING	213.37.29.0/24	CIDR block description										
rir	STRING	ARIN	Acceptable values: ARIN, RIPE, APNIC, AfriNIC										
Optional Parameters													

Name	Type	Example	Description
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		User-defined block code as defined in Admin-IPAM settings: Generic Code Per Block Name
notes	STRING		Notes or description for the block
region	STRING	SFO	The value from the list of name/value pairs which make up the list of available regions
tags	STRING	customer,vpn	Comma-separated list of tags
vlan	INTEGER	123	VLAN for the block

Example URL	/api/v1/api.php?target=ipam&action=add&rir=ARIN
-------------	---

Update

URL	/api/v1/api.php?target=ipam&action=update&type=IP&ipid=13420&rir=RIPE
-----	---

Description	Updates detail data about an IP block.
-------------	--

Returns	<p>Examples:</p> <table border="1"> <tr> <td>SUCCESSFUL</td> <td>SINGLE BLOCK</td> <td><pre>{ "success":1, "message": "Block 192.168.0.0/24 (12345) updated", "data": { "id":12345, "cidr":192.168.0.0/24", ... } }</pre></td> </tr> <tr> <td>SUCCESSFUL</td> <td>MULTIPLE BLOCKS</td> <td><pre>{ "success":1, "message": "3 blocks updated", "data": [{ "id":12345, "cidr":192.168.0.0/24", ... }, { "id":12346, "cidr": "192.168.0.1/32", ... }] }</pre></td> </tr> <tr> <td>ERROR</td> <td></td> <td><pre>{ "success":0, "message": "error message" }</pre></td> </tr> </table>	SUCCESSFUL	SINGLE BLOCK	<pre>{ "success":1, "message": "Block 192.168.0.0/24 (12345) updated", "data": { "id":12345, "cidr":192.168.0.0/24", ... } }</pre>	SUCCESSFUL	MULTIPLE BLOCKS	<pre>{ "success":1, "message": "3 blocks updated", "data": [{ "id":12345, "cidr":192.168.0.0/24", ... }, { "id":12346, "cidr": "192.168.0.1/32", ... }] }</pre>	ERROR		<pre>{ "success":0, "message": "error message" }</pre>
SUCCESSFUL	SINGLE BLOCK	<pre>{ "success":1, "message": "Block 192.168.0.0/24 (12345) updated", "data": { "id":12345, "cidr":192.168.0.0/24", ... } }</pre>								
SUCCESSFUL	MULTIPLE BLOCKS	<pre>{ "success":1, "message": "3 blocks updated", "data": [{ "id":12345, "cidr":192.168.0.0/24", ... }, { "id":12346, "cidr": "192.168.0.1/32", ... }] }</pre>								
ERROR		<pre>{ "success":0, "message": "error message" }</pre>								

Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Allow Multiple</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>INTEGER</td> <td>125</td> <td>Yes</td> <td>ID of the IP block. Multiple block IDs can be specified in a comma-separated list.</td> </tr> <tr> <td>block</td> <td>STRING</td> <td>192.0.0.0/24</td> <td>Yes</td> <td>CIDR or the block. Multiple CIDRs can be specified in a comma-separated list.</td> </tr> </tbody> </table>	Name	Type	Example	Allow Multiple	Description	id	INTEGER	125	Yes	ID of the IP block. Multiple block IDs can be specified in a comma-separated list.	block	STRING	192.0.0.0/24	Yes	CIDR or the block. Multiple CIDRs can be specified in a comma-separated list.
Name	Type	Example	Allow Multiple	Description												
id	INTEGER	125	Yes	ID of the IP block. Multiple block IDs can be specified in a comma-separated list.												
block	STRING	192.0.0.0/24	Yes	CIDR or the block. Multiple CIDRs can be specified in a comma-separated list.												

Optional Parameters

Name	Type	Example	Description
allowSubAssignments	BOOL	true	Does the block allow sub-assignments? If the block is assigned and allowSubAssignments is "true", children split from this block will be able to be assigned to different resources. Acceptable values: "true" or "false"
asn	INTEGER	1000	ASN for the block
code	STRING		Arbitrary user-defined block code
lirId	INTEGER		The numeric ID of an LIR resource the block should be linked to
notes	STRING	Words	Misc. Notes
region	STRING	Chicago, IL	The region this IP block is assigned to.
rir	STRING	ARIN	The RIR
vlan	NUMERIC	50000	

Delete

URL	/api/v1/api.php?target=ipam&action=delete&id=12345
Description	Deletes an aggregate block

Returns	<p>Examples</p> <table border="1"> <tr> <td>SUCCESSFUL</td> <td><code>{"success":1,"message":"Aggregate deleted: 192.168.0.0/24", "data":{"id":12345, "cidr":"192.168.0.0/24", ...}}</code></td> </tr> <tr> <td>ERROR</td> <td><code>{"success":0, "message":"error message" }</code></td> </tr> </table>	SUCCESSFUL	<code>{"success":1,"message":"Aggregate deleted: 192.168.0.0/24", "data":{"id":12345, "cidr":"192.168.0.0/24", ...}}</code>	ERROR	<code>{"success":0, "message":"error message" }</code>								
SUCCESSFUL	<code>{"success":1,"message":"Aggregate deleted: 192.168.0.0/24", "data":{"id":12345, "cidr":"192.168.0.0/24", ...}}</code>												
ERROR	<code>{"success":0, "message":"error message" }</code>												
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>block*</td> <td>STRING</td> <td>213.37.29.0/24</td> <td>CIDR block description</td> </tr> <tr> <td>id*</td> <td>INTEGER</td> <td>125</td> <td>ID of the IP block</td> </tr> </tbody> </table> <p>*Either block or id can be used, but only one must be provided</p>	Name	Type	Example	Description	block*	STRING	213.37.29.0/24	CIDR block description	id*	INTEGER	125	ID of the IP block
Name	Type	Example	Description										
block*	STRING	213.37.29.0/24	CIDR block description										
id*	INTEGER	125	ID of the IP block										
Optional Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>force</td> <td>BOOL</td> <td>true</td> <td>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.</td> </tr> </tbody> </table>	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.				
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.										

Add Tag					
URL	/api/v1/api.php?target=ipam&action=addTag&id=13420&tag=Infra				
Description	Adds a tag to an IP block.				
Returns	<p>Examples:</p> <table border="1"> <tr> <td>SUCCESSFUL</td> <td><code>{"success":1,"message":"Tag Added.", "data":{"id":12345, "cidr":"192.168.0.0/24", ...}}</code></td> </tr> <tr> <td>ERROR</td> <td><code>{"success":0, "message":"error message" }</code></td> </tr> </table>	SUCCESSFUL	<code>{"success":1,"message":"Tag Added.", "data":{"id":12345, "cidr":"192.168.0.0/24", ...}}</code>	ERROR	<code>{"success":0, "message":"error message" }</code>
SUCCESSFUL	<code>{"success":1,"message":"Tag Added.", "data":{"id":12345, "cidr":"192.168.0.0/24", ...}}</code>				
ERROR	<code>{"success":0, "message":"error message" }</code>				

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

Delete Tag					
URL	/api/v1/api.php?target=ipam&action=deleteTag&id=13420&tag=Prod				
Description	Removes a tag from an IP block.				
Returns	<p>Examples:</p> <table border="1"> <tr> <td>SUCCESSFUL</td> <td><pre>{ "success":1,"message":"Tag Removed.", "data":{"id":12345, "cidr":192.168.0.0/24, ...} }</pre></td> </tr> <tr> <td>ERROR</td> <td><pre>{ "success":0, "message":"error message" }</pre></td> </tr> </table>	SUCCESSFUL	<pre>{ "success":1,"message":"Tag Removed.", "data":{"id":12345, "cidr":192.168.0.0/24, ...} }</pre>	ERROR	<pre>{ "success":0, "message":"error message" }</pre>
SUCCESSFUL	<pre>{ "success":1,"message":"Tag Removed.", "data":{"id":12345, "cidr":192.168.0.0/24, ...} }</pre>				
ERROR	<pre>{ "success":0, "message":"error message" }</pre>				
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	

Smart Assign	
URL	/api/v1/api.php?target=ipam&action=smartAssign
Description	Selects a block based on supplied parameters (rir, tags, etc.) and assigns to an Resource Holder.

Returns

Examples:

SUCCESSFUL	<pre>{ "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", ...} }</pre>
ERROR	<pre>{ "success":0, "message":"error message" }</pre>

Required Parameters

Name	Type	Example	Description
code	STRING	Code X	Arbitrary user-defined block code
mask	INTEGER		The size of the block to be assigned
lirId	INTEGER		The ID of an LIR resource
rir	STRING	ARIN	Acceptable values: ARIN, RIPE, APNIC, AfriNIC
resourceId*	INTEGER		Integer ID of the resource to assign the block to

resourceHolderId	STRING	6c-1234	The 20 character resource holder ID. This is an internal short identifier for the customer, and should be used to link resource holder details in the 6Connect database back to your organization. Consider using internal customer numbers or department numbers for this field depending on your use case. Example: Customer-001 or 000213.
resourceQuery	JSON	<pre>{ "custom_id": "cust-001" }</pre>	A JSON object representing a valid resource query. Any parameters that can be used for a Resource GET API call can be used. Use of the resourceQuery parameter will return blocks assigned to any of the resources returned by that query.
type	STRING	"IPv4" or "IPv6"	The type of block to assign

Optional Parameters

Name	Type	Example	Description
tags	STRING	customer,vpn	Comma separated string of tags
region	STRING	Ashburn	Region to assign from
assignedToResource	INTEGER	1234	Used to Smart Assign out of a block which is already assigned to another resource and allows sub assignment

Example URL

/api/v1/api.php?target=ipam&action=smartAssign&mask=24&resourceHolderId=SJS-0031&rir=ARIN&tags=customer,vpn

Direct Assign

URL

/api/v1/api.php?target=ipam&action=directAssign

Description

Assigns a block to an Resource Holder

Returns

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		The 20 character resource holder ID. This is an internal short identifier for the customer, and should be used to link resource holder details in the 6Connect database back to your organization. Consider using internal customer numbers or department numbers for this field depending on your use case. Example: Customer-001 or 000213.
resourceId*	INTEGER		Integer ID of the resource to assign the block to

	resourceQuery	JSON	<pre>{ "custom_id": "cust-001" }</pre>	<p>A JSON object representing a valid resource query. Any parameters that can be used for a 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.</p>
Example URL	/api/v1/api.php?target=ipam&action=directAssign&block=213.37.29.0/24&resourceHolderId=SJS-0031			

Unassign													
URL	/api/v1/api.php?target=ipam&action=unassign												
Description	Reclaims the specified block to be reassigned in the future												
Returns	<p>Examples:</p> <table border="1" data-bbox="812 1197 1494 1449"> <tr> <td data-bbox="812 1197 1153 1365">SUCCESSFUL</td> <td data-bbox="1153 1197 1494 1365"> <pre>{ "success":1,"message":"192.168.0.0/24 unassigned", "id":12345, "data":{"id":12345, "cidr":192.168.0.0/24, ...} }</pre> </td> </tr> <tr> <td data-bbox="812 1365 1153 1449">ERROR</td> <td data-bbox="1153 1365 1494 1449"> <pre>{ "success":0, "message":"error message" }</pre> </td> </tr> </table>	SUCCESSFUL	<pre>{ "success":1,"message":"192.168.0.0/24 unassigned", "id":12345, "data":{"id":12345, "cidr":192.168.0.0/24, ...} }</pre>	ERROR	<pre>{ "success":0, "message":"error message" }</pre>								
SUCCESSFUL	<pre>{ "success":1,"message":"192.168.0.0/24 unassigned", "id":12345, "data":{"id":12345, "cidr":192.168.0.0/24, ...} }</pre>												
ERROR	<pre>{ "success":0, "message":"error message" }</pre>												
Required Parameters	<table border="1" data-bbox="812 1533 1494 1764"> <thead> <tr> <th data-bbox="812 1533 974 1596">Name</th> <th data-bbox="974 1533 1153 1596">Type</th> <th data-bbox="1153 1533 1315 1596">Example</th> <th data-bbox="1315 1533 1494 1596">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="812 1596 974 1680">block*</td> <td data-bbox="974 1596 1153 1680">STRING</td> <td data-bbox="1153 1596 1315 1680">213.37.29.0/24</td> <td data-bbox="1315 1596 1494 1680">CIDR block description</td> </tr> <tr> <td data-bbox="812 1680 974 1764">id*</td> <td data-bbox="974 1680 1153 1764">INTEGER</td> <td data-bbox="1153 1680 1315 1764">125</td> <td data-bbox="1315 1680 1494 1764">ID of the IP block</td> </tr> </tbody> </table> <p data-bbox="812 1764 1494 1806">*Either block or id can be used, but only one must be provided</p>	Name	Type	Example	Description	block*	STRING	213.37.29.0/24	CIDR block description	id*	INTEGER	125	ID of the IP block
Name	Type	Example	Description										
block*	STRING	213.37.29.0/24	CIDR block description										
id*	INTEGER	125	ID of the IP block										
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	<p>Examples:</p> <table border="1"> <tbody> <tr> <td>SUCCESSFUL</td> <td><pre>{ "success": 1, "message": "Tags Retrieved.", "data": [{"value": "IT", "name": "IT"}, {"value": "LTE", "name": "LTE"}, {"value": "PTP", "name": "Point to Point"}, {"value": "Prod", "name": "Production"}, {"value": "VM", "name": "Virtual Machines"}, {"value": "VOIP", "name": "VOIP"}, {"value": "ANY", "name": "ANY"}]}</pre></td> </tr> <tr> <td>ERROR</td> <td><pre>{ 'success': 0, 'message': 'error message' }</pre></td> </tr> </tbody> </table>	SUCCESSFUL	<pre>{ "success": 1, "message": "Tags Retrieved.", "data": [{"value": "IT", "name": "IT"}, {"value": "LTE", "name": "LTE"}, {"value": "PTP", "name": "Point to Point"}, {"value": "Prod", "name": "Production"}, {"value": "VM", "name": "Virtual Machines"}, {"value": "VOIP", "name": "VOIP"}, {"value": "ANY", "name": "ANY"}]}</pre>	ERROR	<pre>{ 'success': 0, 'message': 'error message' }</pre>
SUCCESSFUL	<pre>{ "success": 1, "message": "Tags Retrieved.", "data": [{"value": "IT", "name": "IT"}, {"value": "LTE", "name": "LTE"}, {"value": "PTP", "name": "Point to Point"}, {"value": "Prod", "name": "Production"}, {"value": "VM", "name": "Virtual Machines"}, {"value": "VOIP", "name": "VOIP"}, {"value": "ANY", "name": "ANY"}]}</pre>				
ERROR	<pre>{ 'success': 0, 'message': 'error message' }</pre>				

Get RIRs List					
URL	/api/v1/api.php?target=ipam&action=getRIRList				
Description	Returns a list of all valid RIRs in the database.				
Returns	<p>Examples:</p> <table border="1"> <tbody> <tr> <td>SUCCESSFUL</td> <td><pre>{ "success": 1, "message": "RIRs Retrieved.", "data": [{"value": "ARIN", "name": "ARIN"}, {"value": "1918", "name": "1918"}, {"value": "AfrinIC", "name": "AfrinIC"}, {"value": "APNIC", "name": "APNIC"}, {"value": "LACNIC", "name": "LACNIC"}, {"value": "RIPE", "name": "RIPE"}, {"value": "1918-SJC", "name": "1918-SJC"}]}</pre></td> </tr> <tr> <td>ERROR</td> <td><pre>{ 'success': 0, 'message': 'error message' }</pre></td> </tr> </tbody> </table>	SUCCESSFUL	<pre>{ "success": 1, "message": "RIRs Retrieved.", "data": [{"value": "ARIN", "name": "ARIN"}, {"value": "1918", "name": "1918"}, {"value": "AfrinIC", "name": "AfrinIC"}, {"value": "APNIC", "name": "APNIC"}, {"value": "LACNIC", "name": "LACNIC"}, {"value": "RIPE", "name": "RIPE"}, {"value": "1918-SJC", "name": "1918-SJC"}]}</pre>	ERROR	<pre>{ 'success': 0, 'message': 'error message' }</pre>
SUCCESSFUL	<pre>{ "success": 1, "message": "RIRs Retrieved.", "data": [{"value": "ARIN", "name": "ARIN"}, {"value": "1918", "name": "1918"}, {"value": "AfrinIC", "name": "AfrinIC"}, {"value": "APNIC", "name": "APNIC"}, {"value": "LACNIC", "name": "LACNIC"}, {"value": "RIPE", "name": "RIPE"}, {"value": "1918-SJC", "name": "1918-SJC"}]}</pre>				
ERROR	<pre>{ 'success': 0, 'message': 'error message' }</pre>				

Get Regions List	
URL	/api/v1/api.php?target=ipam&action=getRegionList
Description	Returns a list of all valid Regions in the database.

Returns	<p>Examples:</p> <table border="1"> <tr> <td data-bbox="824 184 1149 1115">SUCCESSFUL</td> <td data-bbox="1149 184 1479 1115"> <pre>{ "success": 1, "message": "Regions Retrieved.", "data": [{ "value": "ANY", "name": "Any Region" }, { "value": "ASH1", "name": "Ashburn, VA" }, { "value": "BOS", "name": "Boston, MA" }, { "value": "CHI", "name": "Chicago, IL" }, { "value": "DAL", "name": "Dallas, TX" }, { "value": "DEN", "name": "Denver, CO" }, { "value": "FRKT", "name": "Frankfurt, DE" }, { "value": "LON1", "name": "London, UK" }, { "value": "MIA", "name": "Miami, FL" }, { "value": "PAR", "name": "Paris, FR" }, { "value": "SFO", "name": "San Francisco, CA" }, { "value": "SEA", "name": "Seattle, WA" }, { "value": "Tokyo", "name": "Tokyo" }, { "value": "Singapore", "name": "Singapore" }, { "value": "Jakarta", "name": "Jakarta" }] }</pre> </td> </tr> <tr> <td data-bbox="824 1115 1149 1199">ERROR</td> <td data-bbox="1149 1115 1479 1199"> <pre>{ "success": 0, "message": "error message" }</pre> </td> </tr> </table>	SUCCESSFUL	<pre>{ "success": 1, "message": "Regions Retrieved.", "data": [{ "value": "ANY", "name": "Any Region" }, { "value": "ASH1", "name": "Ashburn, VA" }, { "value": "BOS", "name": "Boston, MA" }, { "value": "CHI", "name": "Chicago, IL" }, { "value": "DAL", "name": "Dallas, TX" }, { "value": "DEN", "name": "Denver, CO" }, { "value": "FRKT", "name": "Frankfurt, DE" }, { "value": "LON1", "name": "London, UK" }, { "value": "MIA", "name": "Miami, FL" }, { "value": "PAR", "name": "Paris, FR" }, { "value": "SFO", "name": "San Francisco, CA" }, { "value": "SEA", "name": "Seattle, WA" }, { "value": "Tokyo", "name": "Tokyo" }, { "value": "Singapore", "name": "Singapore" }, { "value": "Jakarta", "name": "Jakarta" }] }</pre>	ERROR	<pre>{ "success": 0, "message": "error message" }</pre>
SUCCESSFUL	<pre>{ "success": 1, "message": "Regions Retrieved.", "data": [{ "value": "ANY", "name": "Any Region" }, { "value": "ASH1", "name": "Ashburn, VA" }, { "value": "BOS", "name": "Boston, MA" }, { "value": "CHI", "name": "Chicago, IL" }, { "value": "DAL", "name": "Dallas, TX" }, { "value": "DEN", "name": "Denver, CO" }, { "value": "FRKT", "name": "Frankfurt, DE" }, { "value": "LON1", "name": "London, UK" }, { "value": "MIA", "name": "Miami, FL" }, { "value": "PAR", "name": "Paris, FR" }, { "value": "SFO", "name": "San Francisco, CA" }, { "value": "SEA", "name": "Seattle, WA" }, { "value": "Tokyo", "name": "Tokyo" }, { "value": "Singapore", "name": "Singapore" }, { "value": "Jakarta", "name": "Jakarta" }] }</pre>				
ERROR	<pre>{ "success": 0, "message": "error message" }</pre>				

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.
Example URLs	https://ops.6connect.com/qa-4.1/api/v1/api.php?target=ipam&action=utilization&id=23131
Returns	<p>Examples:</p>

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	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>block*</td> <td>STRING</td> <td>213.37.29.0/24</td> <td>CIDR block description</td> </tr> <tr> <td>id*</td> <td>INTEGER</td> <td>125</td> <td>ID of the IP block</td> </tr> </tbody> </table> <p>*Either block or id can be used, but only one must be provided</p>	Name	Type	Example	Description	block*	STRING	213.37.29.0/24	CIDR block description	id*	INTEGER	125	ID of the IP block
Name	Type	Example	Description										
block*	STRING	213.37.29.0/24	CIDR block description										
id*	INTEGER	125	ID of the IP block										
Optional Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>mask</td> <td>int</td> <td>24</td> <td>The specific mask size to retrieve utilization for. If using this parameter, the id parameter should be the id of the aggregate.</td> </tr> </tbody> </table>	Name	Type	Example	Description	mask	int	24	The specific mask size to retrieve utilization for. If using this parameter, the id parameter should be the id of the aggregate.				
Name	Type	Example	Description										
mask	int	24	The specific mask size to retrieve utilization for. If using this parameter, the id parameter should be the id of the aggregate.										

Aggregate									
URL	/api/v1/api.php?target=ipam&action=aggregate								
Description	Aggregates a selected block to the mask specified. If no mask specified, re-aggregates blocks to next parent. IE. calling aggregate on a /25 will aggregate both children back to the parent /24. All child blocks must be Available for aggregation to succeed.								
Returns	<p>Examples:</p> <table border="1"> <tbody> <tr> <td data-bbox="824 1268 1149 1486">SUCCESSFUL</td> <td data-bbox="1149 1268 1479 1486">{"success":1, "message": "10.2.0.128\25 aggregated into 10.2.0.0\24", "id":16326}</td> </tr> <tr> <td data-bbox="824 1486 1149 1570">ERROR</td> <td data-bbox="1149 1486 1479 1570">{'success':0, 'message':'error message'}</td> </tr> </tbody> </table>	SUCCESSFUL	{"success":1, "message": "10.2.0.128\25 aggregated into 10.2.0.0\24", "id":16326}	ERROR	{'success':0, 'message':'error message'}				
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	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>int</td> <td>125</td> <td>ID of the IP block.</td> </tr> </tbody> </table>	Name	Type	Example	Description	id	int	125	ID of the IP block.
Name	Type	Example	Description						
id	int	125	ID of the IP block.						

Optional Parameters

Name	Type	Example	Description
mask	int	24	Auto aggregate the block back to this mask size. Note all blocks up this mask size must be Available or call will fail.
block*	STRING	213.37.29.0/24	CIDR block. Only the block id OR the block CIDR can be specified. Not both.

Split

URL

/api/v1/api.php?target=ipam&action=split

Description

Splits a selected block to the mask specified. If no mask specified, it split blocks to next child. IE. calling aggregate on a /24 will split both parent to the child /25s. All parent blocks must be Available, or have Allow Sub Assignments on for a split to succeed.

Returns

Examples:

SUCCESSFUL	<pre>{ "success":1, "message": "10.1.0.0\24 split into 10.1.0.0\25 and 10.1.0.128\25", "data": {"child1":23441 , "child2":23451} }</pre>
ERROR	<pre>{'success':0, 'message':'error message'}</pre>

Required Parameters

Name	Type	Example	Description
id	int	125	ID of the IP block.

Optional Parameters

Name	Type	Example	Description
block*	STRING	213.37.29.0/24	CIDR block. Only the block id OR the block CIDR can be specified. Not both.
autoSplitToMask	int	24	Auto aggregate the block back to this mask size. Note all blocks up this mask size must be Available or call will fail.
autoSplitLimit	int	4	A number the power of 2 (^2).

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	int	125	ID of the IP block.

Optional Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>block*</td> <td>STRING</td> <td>213.37.29.0/24</td> <td>CIDR block. Only the block id OR the block CIDR can be specified. Not both.</td> </tr> </tbody> </table>	Name	Type	Example	Description	block*	STRING	213.37.29.0/24	CIDR block. Only the block id OR the block CIDR can be specified. Not both.
Name	Type	Example	Description						
block*	STRING	213.37.29.0/24	CIDR block. Only the block id OR the block CIDR can be specified. Not both.						
Example	/api/v1/api.php?target=ipam&action=scanBlock&block=8.8.8.0/27								

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	<p>Examples:</p> <table border="1"> <tbody> <tr> <td>SUCCESSFUL</td> <td> <pre>{ "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" }], "status": "completed", "hostsScanned": 4, "activeHosts": 4 } }</pre> </td> </tr> <tr> <td>ERROR</td> <td><pre>{'success':0, 'message':'error message'}</pre></td> </tr> </tbody> </table>	SUCCESSFUL	<pre>{ "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" }], "status": "completed", "hostsScanned": 4, "activeHosts": 4 } }</pre>	ERROR	<pre>{'success':0, 'message':'error message'}</pre>				
SUCCESSFUL	<pre>{ "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" }], "status": "completed", "hostsScanned": 4, "activeHosts": 4 } }</pre>								
ERROR	<pre>{'success':0, 'message':'error message'}</pre>								
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>block*</td> <td>STRING</td> <td>213.37.29.0/24</td> <td>CIDR block. Only the block id OR the block CIDR can be specified. Not both.</td> </tr> </tbody> </table>	Name	Type	Example	Description	block*	STRING	213.37.29.0/24	CIDR block. Only the block id OR the block CIDR can be specified. Not both.
Name	Type	Example	Description						
block*	STRING	213.37.29.0/24	CIDR block. Only the block id OR the block CIDR can be specified. Not both.						
Optional Parameters	None								
Example	/api/v1/api.php?target=ipam&action=getScanResults&block=8.8.8.0/27								

API Module - LIR

LIR Management

- Get
- Delete

Get	
URL	/api/v1/api.php?target=lir&action=get
Description	Returns a list of LIRs
Returns	<p>Examples:</p> <pre> SUCCESSFUL { "success": 1, "message": "2 objects found", "data": [{ "id": "100", "name": "RIPE Test LIR", "slug": "ripe-test-lir", "entities": [{ "mnt_by": "mntner@email.com" "mnt_by_password": "password", "admin_c": "test-admin-c", "tech_c": "test-tech-c", "api_key": null }], "rir": "RIPE" }, { "id": "101", "name": "ARIN Test LIR", "slug": "arin-test-lir", "entities": [{ "org_handle": "TEST-10", "admin_poc": </pre>

```
"TEST-ARIN",  
  
"net_poc": "TEST-ARIN",  
  
"abuse_poc": "",  
  
"net_name_prefix":  
"PRFX",  
  
"api_key":  
"API-XXXX-YYYY-ZZZZ-123  
4"  
    }  
  ],  
  "rir":  
"ARIN",  
    "asn":  
"1000"  
  }  
}
```

	<pre>] } </pre>
ERROR	<pre> { "success": 0, "message": "error message" } </pre>
Example URL	/api/v1/api.php?target=lir&action=get

Delete					
URL	/api/v1/api.php?target=lir&action=delete&id=<ID>				
Description	Deletes and LIR				
Returns	<p>Examples:</p> <table border="1"> <tr> <td>SUCCESSFUL</td> <td> <pre> { "success": 1, "message": "LIR deleted." } </pre> </td> </tr> <tr> <td>ERROR</td> <td> <pre> { "success": 0, "message": "error message" } </pre> </td> </tr> </table>	SUCCESSFUL	<pre> { "success": 1, "message": "LIR deleted." } </pre>	ERROR	<pre> { "success": 0, "message": "error message" } </pre>
SUCCESSFUL	<pre> { "success": 1, "message": "LIR deleted." } </pre>				
ERROR	<pre> { "success": 0, "message": "error message" } </pre>				
Example URL	/api/v1/api.php?target=lir&action=delete&id=100				

API Module - 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,"qualified":null,"is_peer":false,"id":"262","log_data":[]}, {"name":"Akamai Technologies","asn":"20940","request_status":null,"qualified":null,"is_peer":false,"id":"2","log_data":[]}, {"name":"Amazon.com","asn":"16509","request_status":null,"qualified":null,"is_peer":false,"id":"1418","log_data":[]}, {"name":"Atrato IP Networks","asn":"5580","request_status":null,"qualified":null,"is_peer":false,"id":"2283","log_data":[]}, {"name":"OVH","asn":"16276","request_status":null,"qualified":null,"is_peer":false,"id":"1264","log_data":[]}, {"name":"Verisign","asn":"7342","request_status":null,"qualified":null,"is_peer":false,"id":"873","log_data":[]}, {"name":"Walmart.com","asn":"17374","request_status":null,"qualified":null,"is_peer":false,"id":"627","log_data":[]}, {"name":"Windstream Communications","asn":"7029","request_status":null,"qualified":null,"is_peer":false,"id":"1820","log_data":[]}, {"name":"Yahoo!","asn":"10310","request_status":null,"qualified":null,"is_peer":false,"id":"27","log_data":[]}, {"name":"Zayo","asn":"19092","request_status":null,"qualified":null,"is_peer":false,"id":"3611","log_data":[]}, {"name":"Zayo (Abovenet Communications Inc.)","asn":"6461","request_status":null,"qualified":null,"is_peer":false,"id":"541","log_data":[]}, {"name":"tw telecom","asn":"4323","request_status":null,"qualified":null,"is_peer":false,"id":"540","log_data":[]}]}
```

ERROR

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

Required Parameters	<table border="1"> <thead> <tr> <th data-bbox="818 155 987 205">Name</th> <th data-bbox="987 155 1149 205">Type</th> <th data-bbox="1149 155 1312 205">Example</th> <th data-bbox="1312 155 1487 205">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="818 205 987 470">public_id</td> <td data-bbox="987 205 1149 470">INTEGER</td> <td data-bbox="1149 205 1312 470">1</td> <td data-bbox="1312 205 1487 470">The unique numerical identifier of the exchange to retrieve peering communication records for.</td> </tr> </tbody> </table>	Name	Type	Example	Description	public_id	INTEGER	1	The unique numerical identifier of the exchange to retrieve peering communication records for.
Name	Type	Example	Description						
public_id	INTEGER	1	The unique numerical identifier of the exchange to retrieve peering communication records for.						
Optional Parameters	None								
Example URL	/api/v1/api.php?target=peering&action=getCommunications&public_id=1								

getPeers	
URL	/api/v1/api.php?target=peering&action=getPeers
Description	Returns a list of all peers available at an exchange

Returns

Examples:

```
SUCCESSFUL: {"success":1,"message":"184 peers found.,"data":[{"id":"262","public_id":"1","asn":"8560","name":"1&1 Internet","qualified":true,"is_peer":0,"request_status":"sent","info_prefixes":"150","public_ips":[],"contacts":[],"log_data":[]},"id":"286","public_id":null,"asn":"3856","name":"Packet Clearing House","qualified":true,"is_peer":0,"request_status":null,"info_prefixes":"100","public_ips":[],"contacts":[],"log_data":[]},"id":"890","public_id":null,"asn":"13768","name":"PEER 1 Hosting","qualified":true,"is_peer":0,"request_status":null,"info_prefixes":"3000","public_ips":[],"contacts":[],"log_data":[]},"id":"1676","public_id":null,"asn":"3737","name":"PenTeleData","qualified":true,"is_peer":0,"request_status":null,"info_prefixes":"80","public_ips":[],"contacts":[],"log_data":[]},"id":"1560","public_id":null,"asn":"23265","name":"Pocketinet Communications, Inc.,"qualified":true,"is_peer":0,"request_status":null,"info_prefixes":"200","public_ips":[],"contacts":[],"log_data":[]},"id":"576","public_id":null,"asn":"32787","name":"Prolexic","qualified":true,"is_peer":0,"request_status":null,"info_prefixes":"500","public_ips":[],"contacts":[],"log_data":[]},"id":"353","public_id":null,"asn":"27524","name":"Xeex Communications","qualified":true,"is_peer":0,"request_status":null,"info_prefixes":"3000","public_ips":[],"contacts":[],"log_data":[]},"id":"27","public_id":null,"asn":"10310","name":"Yahoo!","qualified":true,"is_peer":0,"request_status":null,"info_prefixes":"500","public_ips":[],"contacts":[],"log_data":[]},"id":"1751","public_id":null,"asn":"13238","name":"Yandex LLC","qualified":true,"is_peer":0,"request_status":null,"info_prefixes":"60","public_ips":[],"contacts":[],"log_data":[]},"id":"4078","public_id":null,"asn":"19468","name":"YieldBuild Inc","qualified":true,"is_peer":0,"request_status":null,"info_prefixes":"5","public_ips":[],"contacts":[],"log_data":[]},"id":"541","public_id":null,"asn":"6461","name":"Zayo (Abovenet Communications Inc.)","qualified":true,"is_peer":0,"request_status":null,"info_prefixes":"20000","public_ips":[],"contacts":[],"log_data":[]},"id":"2351","public_id":null,"asn":"14824","name":"Zynga Game Network, Inc.,"qualified":true,"is_peer":0,"request_status":null,"info_prefixes":null,"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 retrieve peering communication records for.
id	INT	1	The unique numerical identifier of the peer in peeringDB.
asn	INT	4436	

name	STRING	GTT	
aka	STRING	nLayer	
website	STRING	http://www.gtt.net	
notes_public	STRING		
notes_private	STRING		
irr_as_set	STRING	AS-NLAYER	
info_traffic	ENUM	1 Tbps+	enum('Not Disclosed','0-20 Mbps','20-100 Mbps','100-1000Mbps','1-5 Gbps','5-10Gbps','10-20Gbps','20-50 Gbps','50-100 Gbps','100+ Gbps','100-200 Gbps','200-300 Gbps','300-500 Gbps','500-1000 Gbps','1 Tbps+') DEFAULT 'Not Disclosed'
info_ratio	ENUM	Mostly Outbound	enum('Not Disclosed','Heavy Outbound','Mostly Outbound','Balanced','Mostly Inbound','Heavy Inbound') DEFAULT 'Not Disclosed'
info_scope	ENUM	Global	enum('Not Disclosed','Regional','North America','Asia Pacific','Europe','Africa','South America','Global') DEFAULT NULL

info_type	ENUM	NSP	enum('Not Disclosed','NSP','Content','Cable/DSL/ISP','Enterprise','Educational/Research','Non-Profit') DEFAULT 'Not Disclosed'
info_prefixes	INT	10000	
info_lookingglass	STRING	http://lg.nlayer.net/	
info_router_server	STRING	telnet://router-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 NULL
policy_locations	ENUM	Required - International	enum('Not Required','Preferred','Required - US','Required - International') DEFAULT NULL
policy_ratio	ENUM	No	enum('Yes','No') DEFAULT NULL
policy_contracts	ENUM	Not Required	enum('Not Required','Private Only','Required') DEFAULT NULL
policy_nopublic	ENUM	N	enum('Y','N') NOT NULL DEFAULT 'N'

policy_noprivate	ENUM	N	enum('Y','N') NOT NULL DEFAULT 'N'
date_created	DATETIME	2013-03-21 15:36:42	Date the peeringdb entry was created
date_lastupdated	DATETIME	2013-03-21 15:36:42	Date the peeringdb entry was last updated
include_public_ips	BOOL	TRUE	Returns a list of all public facing IPs
include_contacts	BOOL	TRUE	Returns a list of all contacts associated with peer(s)
include_log_data	BOOL	TRUE	Returns a list of all log data associated with the peer(s) (use with care)

Example URL

/api/v1/api.php?target=peering&action=getPeers&public_id=1

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_asn":"16509","peer_name":"Amazon.com","email_type":"text","email_from":"ops@6connect.com","email_to":"nalinmk@gmail.com","subject":"Peering request from 6connect, Inc.,"body":"Peering,\n\n\n6connect, Inc., 8038, would like to peer with Amazon.com at our common locations.\n\n\nFacility, IP Address\nEquinix Ashburn - 206.126.236.68\nEquinix Palo Alto - 198.32.176.36\nEquinix Ashburn - 206.126.236.35\nEquinix San Jose - 206.223.116.177\nLINX Juniper LAN - 195.66.225.175\n\n\nSincerely,\nOperations\nops@6connect.com\n\n\n\n6connect, Inc. information:\nEquinix Palo Alto, 2001:504:d::33\nEquinix Palo Alto, 198.32.176.51\n\n\nPeeringDB: http://as8038.peeringdb.com\n\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	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>peer_participant_id</td> <td>INTEGER</td> <td>1</td> <td>The numerical id of the peer</td> </tr> </tbody> </table>	Name	Type	Example	Description	peer_participant_id	INTEGER	1	The numerical id of the peer
Name	Type	Example	Description						
peer_participant_id	INTEGER	1	The numerical id of the peer						
Example URL	/api/v1/api.php?target=peering&action=getRequests&peer_participant_id=1								

getSessions									
URL	/api/v1/api.php?target=peering&action=getSessions								
Description	Returns a list of all bgp peering sessions								
Returns	<p>Examples:</p> <p>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_asn":"20940", "peer_name":"Akamai Technologies", "peer_participant_id":"2", "peer_ipaddr":"206.126.236.102", "peer_hostname":null, "peer_group":"group b", "public_id":"1", "public_name":"Equinix Ashburn", "ip_type":"ipv4", "type":"Peer", "state":"not configured", "prfx_max":"20", "prfx_received":null, "password":"0", "note":null},]}</p> <p>ERROR: {"success":1,"message":"No peers found."}</p>								
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>public_id</td> <td>INTEGER</td> <td>1</td> <td>The unique numerical identifier of the exchange to retrieve peering communication records for.</td> </tr> </tbody> </table>	Name	Type	Example	Description	public_id	INTEGER	1	The unique numerical identifier of the exchange to retrieve peering communication records for.
Name	Type	Example	Description						
public_id	INTEGER	1	The unique numerical identifier of the exchange to retrieve peering communication records for.						

Optional Parameters

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	<p>Examples:</p> <p>SUCCESSFUL: {"success":1,"message":"Session added: Amazon.com (AS8038V1.2.3.5) - (AS16509V206.126.236.68)","data":{"id":111,"source_asn":"8038","source_ipaddr":"1.2.3.5","resource_id":"422","resource_name":null,"peer_asn":"16509","peer_name":"Amazon.com","peer_participant_id":"1418","peer_ipaddr":"206.126.236.68","peer_hostname":null,"peer_group":"LAME-PEERS","public_id":"1","public_name":null,"ip_type":"ipv4","type":"Peer","state":"not configured","prfx_max":"200","prfx_received":null,"password":"ace12345","note":"I'm a fancy note."}}</p> <p>ERROR: {"success":1,"message":"No request found.,"data":[]}</p>																																																																
Required Parameters	None																																																																
Optional Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>source_asn</td> <td>INTEGER</td> <td>1</td> <td>The numerical id of the peer</td> </tr> <tr> <td>source_ipaddr</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>resource_id</td> <td>INTEGER</td> <td></td> <td></td> </tr> <tr> <td>peer_asn</td> <td>INTEGER</td> <td></td> <td></td> </tr> <tr> <td>peer_name</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>peer_participant_id</td> <td>INTEGER</td> <td></td> <td></td> </tr> <tr> <td>peer_ipaddr</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>peer_hostname</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>peer_group</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>public_id</td> <td></td> <td></td> <td></td> </tr> <tr> <td>type</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>ip_type</td> <td>ENUM</td> <td></td> <td>enum('ipv4','ipv6')</td> </tr> <tr> <td>state</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>prfx_max</td> <td>INTEGER</td> <td></td> <td></td> </tr> <tr> <td>note</td> <td>STRING</td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Example	Description	source_asn	INTEGER	1	The numerical id of the peer	source_ipaddr	STRING			resource_id	INTEGER			peer_asn	INTEGER			peer_name	STRING			peer_participant_id	INTEGER			peer_ipaddr	STRING			peer_hostname	STRING			peer_group	STRING			public_id				type	STRING			ip_type	ENUM		enum('ipv4','ipv6')	state	STRING			prfx_max	INTEGER			note	STRING		
Name	Type	Example	Description																																																														
source_asn	INTEGER	1	The numerical id of the peer																																																														
source_ipaddr	STRING																																																																
resource_id	INTEGER																																																																
peer_asn	INTEGER																																																																
peer_name	STRING																																																																
peer_participant_id	INTEGER																																																																
peer_ipaddr	STRING																																																																
peer_hostname	STRING																																																																
peer_group	STRING																																																																
public_id																																																																	
type	STRING																																																																
ip_type	ENUM		enum('ipv4','ipv6')																																																														
state	STRING																																																																
prfx_max	INTEGER																																																																
note	STRING																																																																
Example URL	/api/v1/api.php?target=peering&action=getRequests&peer_participant_id=1																																																																

configureSession	
URL	/api/v1/api.php?target=peering&action=configureSession
Description	Configure a BGP session on the router

Returns	<p>Examples:</p> <p>SUCCESSFUL:</p> <p>ERROR: {"success":0,"message":"Unable to authenticate "}</p>																																																																
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>session_id</td> <td>INTEGER</td> <td>1</td> <td></td> </tr> </tbody> </table>	Name	Type	Example	Description	session_id	INTEGER	1																																																									
Name	Type	Example	Description																																																														
session_id	INTEGER	1																																																															
Optional Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>session_id</td> <td>INTEGER</td> <td>1</td> <td>The numerical id of the peer</td> </tr> <tr> <td>source_ipaddr</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>resource_id</td> <td>INTEGER</td> <td></td> <td></td> </tr> <tr> <td>peer_asn</td> <td>INTEGER</td> <td></td> <td></td> </tr> <tr> <td>peer_name</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>peer_participant_id</td> <td>INTEGER</td> <td></td> <td></td> </tr> <tr> <td>peer_ipaddr</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>peer_hostname</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>peer_group</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>public_id</td> <td></td> <td></td> <td></td> </tr> <tr> <td>type</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>ip_type</td> <td>ENUM</td> <td></td> <td>enum('ipv4','ipv6')</td> </tr> <tr> <td>state</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>prfx_max</td> <td>INTEGER</td> <td></td> <td></td> </tr> <tr> <td>note</td> <td>STRING</td> <td></td> <td></td> </tr> </tbody> </table>	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		
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	<pre>/api/v1/api.php?target=peering&action=configureSession&session_id=51&username=&config=conf+t%0A%0Arouter+bgp+32787%0A%0A%0Aneighbor+206.126.236.102%0A%0Aneighbor+206.126.236.102+remote-as+20940%0Aneighbor+206.126.236.102+peer-group+group+b%0Aneighbor+206.126.236.102+description+Akamai+Technologies%0Aneighbor+206.126.236.102+maximum-prefix+20+%0Aneighbor+206.126.236.102+password+0+0%0A%0Aend%0Awrite</pre>																																																																

deleteSession	
URL	/api/v1/api.php?target=peering&action=deleteSession
Description	Delete sessions matching criteria

Returns	<p>Examples:</p> <p>SUCCESSFUL: {"success":1,"message":"1 sessions deleted."}</p> <p>ERROR: {"success":0,"message":"No sessions found to delete."}</p>																																																																																												
Required Parameters	None																																																																																												
Optional Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>INTEGER</td> <td>41</td> <td></td> </tr> <tr> <td>public_id</td> <td>INTEGER</td> <td></td> <td></td> </tr> <tr> <td>source_asn</td> <td>INTEGER</td> <td></td> <td></td> </tr> <tr> <td>source_ipaddr</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>resource_id</td> <td>INTEGER</td> <td></td> <td></td> </tr> <tr> <td>peer_asn</td> <td>INTEGER</td> <td></td> <td></td> </tr> <tr> <td>peer_name</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>peer_participant_id</td> <td>INTEGER</td> <td></td> <td></td> </tr> <tr> <td>peer_ipaddr</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>peer_hostname</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>peer_group</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>password</td> <td>INTEGER</td> <td></td> <td></td> </tr> <tr> <td>type</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>state</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>prfx_max</td> <td>INTEGER</td> <td></td> <td></td> </tr> <tr> <td>prfx_received</td> <td>INTEGER</td> <td></td> <td></td> </tr> <tr> <td>ip_type</td> <td>ENUM</td> <td></td> <td>enum('ipv4','ipv6') NOT NULL DEFAULT 'ipv4'</td> </tr> <tr> <td>note</td> <td>STRING</td> <td></td> <td></td> </tr> <tr> <td>created</td> <td>TIMESTAMP</td> <td></td> <td></td> </tr> <tr> <td>modified</td> <td>TIMESTAMP</td> <td></td> <td></td> </tr> <tr> <td>deleted</td> <td>INTEGER</td> <td></td> <td></td> </tr> <tr> <td>public_id</td> <td>INTEGER</td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Example	Description	id	INTEGER	41		public_id	INTEGER			source_asn	INTEGER			source_ipaddr	STRING			resource_id	INTEGER			peer_asn	INTEGER			peer_name	STRING			peer_participant_id	INTEGER			peer_ipaddr	STRING			peer_hostname	STRING			peer_group	STRING			password	INTEGER			type	STRING			state	STRING			prfx_max	INTEGER			prfx_received	INTEGER			ip_type	ENUM		enum('ipv4','ipv6') NOT NULL DEFAULT 'ipv4'	note	STRING			created	TIMESTAMP			modified	TIMESTAMP			deleted	INTEGER			public_id	INTEGER		
Name	Type	Example	Description																																																																																										
id	INTEGER	41																																																																																											
public_id	INTEGER																																																																																												
source_asn	INTEGER																																																																																												
source_ipaddr	STRING																																																																																												
resource_id	INTEGER																																																																																												
peer_asn	INTEGER																																																																																												
peer_name	STRING																																																																																												
peer_participant_id	INTEGER																																																																																												
peer_ipaddr	STRING																																																																																												
peer_hostname	STRING																																																																																												
peer_group	STRING																																																																																												
password	INTEGER																																																																																												
type	STRING																																																																																												
state	STRING																																																																																												
prfx_max	INTEGER																																																																																												
prfx_received	INTEGER																																																																																												
ip_type	ENUM		enum('ipv4','ipv6') NOT NULL DEFAULT 'ipv4'																																																																																										
note	STRING																																																																																												
created	TIMESTAMP																																																																																												
modified	TIMESTAMP																																																																																												
deleted	INTEGER																																																																																												
public_id	INTEGER																																																																																												
Example URL	/api/v1/api.php?target=peering&action=deleteSession&id=171																																																																																												

updateSession

URL	/api/v1/api.php?target=peering&action=updateSession
Description	Updates session values with any new values specified
Returns	<p>Examples:</p> <p>SUCCESSFUL:{"success":1,"message":"Session updated: 123.net (AS32787V1.2.3.4) - (AS12129V206.126.236.70)","data":{"id":"41","source_asn":"32787","source_ipaddr":"1.2.3.4","resource_id":"422","resource_name":"ar1.in.oc.com","peer_asn":"12129","peer_name":"123.net","peer_participant_id":"3899","peer_ipaddr":"206.126.236.70","peer_hostname":null,"peer_group":"group a","public_id":"1","public_name":"Equinix Ashburn","ip_type":"ipv4","type":"Peer","state":"not configured","prfx_max":"10","prfx_received":null,"password":"0","note":"Adding an awesome note."}}</p> <p>ERROR:</p>
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
```

resetPeerStatus

URL

```
/api/v1/api.php?target=peering&action=resetPeerStatus
```

Description

Returns	<p>Examples:</p> <p>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_pre fixes":null,"public_ips":[],"contacts":[],"log_data":{"message":"Peer 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 (AS32787V1.2.3.4) - (AS8560V206.126.236.200)","time":"2014-05-22 22:39:43","request_id":null,"session_id":"71","public_id":"1"},"message":"Request sent: ","time":"2014-04-12 13:24:43","request_id":"121","session_id":null,"public_id":"1"},"message":"Session added: 1&1 Internet (AS32787V1.2.3.4) - (AS8560V206.126.236.200)","time":"2014-04-07 11:32:37","request_id":null,"session_id":"71","public_id":"1"}}}</p> <p>ERROR: {"success":0,"message":"Could not find peer matching parameters"}</p>												
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>participant_id</td> <td>INTEGER</td> <td>262</td> <td>The id of the peer in from the peeringDB peerParticipants table.</td> </tr> <tr> <td>public_id</td> <td>INTEGER</td> <td>1</td> <td>The id of the exchange point from the peeringDB mgmtPublics table.</td> </tr> </tbody> </table>	Name	Type	Example	Description	participant_id	INTEGER	262	The id of the peer in from the peeringDB peerParticipants table.	public_id	INTEGER	1	The id of the exchange point from the peeringDB mgmtPublics table.
Name	Type	Example	Description										
participant_id	INTEGER	262	The id of the peer in from the peeringDB peerParticipants table.										
public_id	INTEGER	1	The id of the exchange point from the peeringDB mgmtPublics table.										
Optional Parameters	None												
Example URL	/api/v1/api.php?target=peering&action=resetPeerStatus&participant_id=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 against using.

Returns

Examples:
 SUCCESSFUL: {"success":1,"message":"Request sent","data":{"id":"922","public_id":"1","asn":"10933","name":"ATX Communications, Inc.,"qualified":true,"is_peer":0,"request_status":"sent","info_prefixes":null,"public_ips":[],"contacts":[],"log_data":{"message":"Request sent to ","time":"2014-05-27 16:59:01","request_id":"181","session_id":null,"public_id":"1"},"message":"Request sent to ","time":"2014-05-27 16:49:30","request_id":"171","session_id":null,"public_id":"1"}}}
 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=28929&peer_name=ASDASD+srl&source_participant_id=2335&source_asn=8038 &public_id=1&type=text&email_from=ops%406connect.com &email_to=operations%40as28929.net&cc=&bcc=&subject=Peering+request+from+6connect%2C+Inc. &body=%0D%0APeering%2C%0D%0A%0D%0A6connect%2C+Inc.%2C+8038%2C+would+like+to+peer+with+ASDASD+srl+at+our+common+locations.%3A%0D%0APeeringDB%3A+http%3A%2F%2Fas8038.peeringdb.com%2F%0D%0A

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	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>participant_id</td> <td>INTEGER</td> <td>262</td> <td></td> </tr> <tr> <td>public_id</td> <td>INTEGER</td> <td>1</td> <td></td> </tr> </tbody> </table>	Name	Type	Example	Description	participant_id	INTEGER	262		public_id	INTEGER	1	
Name	Type	Example	Description										
participant_id	INTEGER	262											
public_id	INTEGER	1											
Optional Parameters	None												
Example URL													

updatePeer													
URL	/api/v1/api.php?target=peering&action=updatePeer												
Description													
Returns	Examples: SUCCESSFUL: ERROR:												
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>participant_id</td> <td>INTEGER</td> <td>262</td> <td></td> </tr> <tr> <td>public_id</td> <td>INTEGER</td> <td>1</td> <td></td> </tr> </tbody> </table>	Name	Type	Example	Description	participant_id	INTEGER	262		public_id	INTEGER	1	
Name	Type	Example	Description										
participant_id	INTEGER	262											
public_id	INTEGER	1											
Optional Parameters	None												
Example URL													

API Module - Resource

Resources

- [get](#)
- [add](#)
- [update](#)
- [delete](#)

get	
URL	/api/v1/api.php?target=resource&action=get
Description	Get a resource or resources
Returns	Examples: SUCCESSFUL: <code>{"success":1,"message":"Search successful","data":[{"id":"57","name":"2nd Email","slug":"6c-contact-email2","type":"field","parent_id":"1","category_id":null,"attr":[]}]}</code> ERROR: <code>{"success":0,"message":"Search failed"}</code>

Optional Parameters

Name	Type	Notes/Example
name	STRING	Name of the resource. Example: 6Connect, Inc.
slug	STRING	The unique URL friendly name of the resource. Example: 6connect-inc
type	STRING	Type of resource (eg. <i>entry</i> , <i>field</i> , <i>category</i>)

At most, one of the following:

Name	Type	Notes/Example
id	INTEGER	Get the resource which has this ID
resource_in	ARRAY	Get any resource which has any of these IDs
resource_not_in	ARRAY	Get all the resources which do not have any of these IDs

At most, one of the following:

Name	Type	Notes/Example
parent_id	INTEGER	Get the resources whose parent has this ID
parent__in	ARRAY	Get any resource whose parents have any of these IDs
parent__not_in	ARRAY	Get all resources whose parents do not have any of these IDs

At most, one of the following:

Name	Type	Notes/Example
category_id	INTEGER	Get the resources of the category that has this ID
category__in	ARRAY	Get the resources of the categories that have any of these IDs
category__not_in	ARRAY	Get the resources of all the categories that do not have any of these IDs

You can set the order of the results by setting the STRING value of the parameter `orderby` to one of the following :

- none
- id
- name (*default*)
- slug
- type
- parent_id
- date
- resource__in (*preserve order given in the resource__in array*)

You can set the direction of the ordering of the results by setting the STRING value of the parameter **order** to one of the following :

- ASC (*default*)
- DESC

You can further limit the results based on attributes the resources may have:

Name	Type	Notes/Example
attr_key	STRING	The name of the attribute. Example: network-fqdn
attr_value	STRING	The value of any attribute, or if attr_key is specified, the value of the attribute defined in attr_key.

attr_compare	STRING	<p>If both attr_key and attr_value are given, the results are by default compared based on the value given as attr_value being equal to the value stored in the database. You can optionally change this by setting the STRING value of attr_compare to one of the following:</p> <ul style="list-style-type: none">• = (<i>default</i>)• !=• >• >=• <• <=• 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 restrict the range of the resources returned.

Name	Type	Notes/Example
resources_per_page	INTEGER	How many resources to return.
offset	INTEGER	How many resources to offset (the initial resource is 0, not 1).
paged	INTEGER	The page to return (starts at 1, not 0). This parameter is provided for convenience and is used to calculate the offset where: offset=(paged-1)*resources_per_page

Example URL

/api/v1/api.php?target=resource&action=get&id=7

add

URL

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

Description

Add a resource.

Returns

Examples:

/api/v1/api.php?target=resource&action=add&meta[name]=apitest&meta[type]=entry&meta[section]=firewall&fields[network-fqdn]=www.example.com

SUCCESSFUL: {"success":1,"message":"Resource added","data":{"id":1077,"name":"apitest","slug":"apitest","type":"entry","parent_id":1,"category_id":"NULL","attr":{"_section":"70","network-fqdn":"www.example.com"},"section":{"id":"70","name":"Firewall","slug":"firewall","type":"section","parent_id":"1","category_id":null,"attr":{}}}}

/api/v1/api.php?target=resource&action=add&meta[name]=apitest&meta[type]=entry&fields[network-fqdn]=www.example.com

ERROR:{"success":0,"message":"Entries must be assigned to a section"}

Required Parameters

Name	Type	Notes/Example
meta[name]	STRING	Name of the resource
meta[type]	STRING	Type of resource (entry, section, field, ect)

Optional Parameters

Name	Type	Notes/Example
meta[parent_id]	INTEGER	ID of the parent resource
meta[category_id]	INTEGER	ID of the category

Required Parameters

(meta[type] = entry)

One of the following:

Name	Type	Notes/Example
meta[section_id]	INTEGER	ID of the section that the entry will be assigned to
meta[section]	STRING	Slug of the section that the entry will be assigned to

Optional Parameters

(meta[type] = entry)

Name	Type	Notes/Example
------	------	---------------

fields[]	ARRAY	<p>Entry field values (for fields that have already been assigned to the section) can be populated when the entry is created.</p> <p>The format is field[field-slug][field-instance]. If the field instance is left blank, it will simply be the next value in the instance array. For example:</p> <p><i>fields[network-fqdn][]=example.com&fields[network-fqdn][]=test.com</i></p> <p>would be written in JSON as</p> <pre>var fields = { "network-fqdn" : ["example.com", "test.com"] }</pre> <p>A field can be added to a section multiple times. The field instance is used to keep track of which field occurrence we are referring. In this example, the network-fqdn field had been added twice to the section so we were able to store two values for it.</p>
----------	-------	--

meta[custom_id]	STRING	A custom ID for the entry. In the past this has been called the Resource Holder ID or Customer ID. Most recently it was implemented as a text field with the slug "6c-resourceholder-id." Now it is a fundamental part the entry type resources.
-----------------	--------	--

Required Parameters

(meta[type] = field)

Name	Type	Notes/Example
meta[field_type]	STRING	Type of field <ul style="list-style-type: none"> • 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 radio buttons or a choicebox (dropdown) with several options. For example: meta[type]=field&meta[name]=Colors&meta[field_type]=choicebox&meta[options][]=Blue&meta[options][]=Green Will create a choicebox with dropdown options of Blue and Green.

update										
URL	/api/v1/api.php?target=resource&action=update									
Description	Update a resource.									
Returns	<p>Examples:</p> <p>SUCCESSFUL: <code>{"success":1,"message":"Resource Updated","data":{"id":"1055","name":"87-child-1","slug":"87-child-1","type":"entry","parent_id":"87","category_id":"65","attr":{"_section":"70"},"section":{"id":"70","name":"Firewall","slug":"firewall","type":"section","parent_id":"1","category_id":null,"attr":{}}}}</code></p> <p>ERROR: <code>{"success":0,"message":"No resource found with ID: 1079"}</code></p>									
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Notes/Example</th> </tr> </thead> <tbody> <tr> <td>meta[id]</td> <td>INTEGER</td> <td>ID of resource</td> </tr> <tr> <td>meta[type]</td> <td>STRING</td> <td>Type of resource (entry, section, field, ect)</td> </tr> </tbody> </table>	Name	Type	Notes/Example	meta[id]	INTEGER	ID of resource	meta[type]	STRING	Type of resource (entry, section, field, ect)
Name	Type	Notes/Example								
meta[id]	INTEGER	ID of resource								
meta[type]	STRING	Type of resource (entry, section, field, ect)								
Optional Parameters (meta[type] = entry)	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Notes/Example</th> </tr> </thead> <tbody> <tr> <td>fields[]</td> <td>ARRAY</td> <td>See "add" documentation</td> </tr> </tbody> </table>	Name	Type	Notes/Example	fields[]	ARRAY	See "add" documentation			
Name	Type	Notes/Example								
fields[]	ARRAY	See "add" documentation								

Optional Parameters

(meta[type] = section)

Name	Type	Notes/Example
fields[]	ARRAY	<p>The fields value should be all the fields that are assigned to the section. Giving an empty array as the fields value will remove all fields from the section.</p> <p>The format is:</p> <pre>fields[position][key]</pre> <p>The position value is the position that the field will appear in (0 is first). The position value must always be included. An example field format for an existing field could be:</p> <pre>fields[0][id]=2 fields[0][slug]=asset-serial-number fields[0][help_block]=something fields[0][new]=false</pre> <ul style="list-style-type: none">▪ Either the id or the slug is required, not both.▪ When the "new" parameter is not included, FALSE is assumed <p>If you want to create a new field and assign it to the section, use a format like this:</p> <pre>fields[10][name]=TextArea fields[10][field_type]=textarea fields[10][new]=true</pre>

delete

URL	/api/v1/api.php?target=resource&action=delete								
Description	Delete a resource.								
Returns	Examples: SUCCESSFUL: {"success":1,"message":"Resource deleted."} ERROR: {"success":0,"message":"No resource found with ID: 57"}								
Required Parameters	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Notes/Example</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>INTEGER</td> <td>ID of the resource</td> </tr> </tbody> </table>			Name	Type	Notes/Example	id	INTEGER	ID of the resource
Name	Type	Notes/Example							
id	INTEGER	ID of the resource							
Example URL	/api/v1/api.php?target=resource&action=delete&id=57								

How Do I...

If you want to get a jumpstart on common API use cases, you came to the right place! Expand the text areas below for walkthroughs and code samples of API calls...

Context: I unassigned an IP address and now it's in the Holding Tank. Now I want to assign an IP from the Holding Tank. I don't want to unassign an IP randomly, in case it is allocated to a Resource. What are my options?

▼ [Click here to expand...](#)

There are 3 options:

1) If you know the specific IP, you can use the ipam-get api call to determine if it is in Holding:

```

/api/v1/api.php?target=ipam&action=get&cidr=1.2.3.4/32

{
  id:1234,
  cidr:"1.2.3.4",
  ...
  resource_name:"Holding"
}

```

2) If you want to show all blocks/IPs in Holding, you can use the following ipam-get API call:

```

/api/v1/api.php?target=ipam&action=get&resourceQuery={"name":"Holding"}

```

3) If you know the block is in Holding, you can issue another ipam-unassign API call to move it from Holding to Available:

```

/api/v1/api.php?target=ipam&action=unassign&block=1.2.3.4/32

```

Context: I need to create a Resource Holder, assign them an IP block, then subassign some IPs out of that block to two new Resource Holders. What does this look like?

▼ [Click here to expand...](#)

We broke this up in a few steps so it's easier to link together.

1) Let's create a Resource Holder called "Ned"

```
query_string =
'target=resource&action=add&meta[type]=entry&meta[section]=resource-holder&meta[name
]=Ned'
query_string += '&apiKey=' + api_key
hash = base64.b64encode( hmac.new(api_secret_key, query_string,
hashlib.sha256).digest() )
url = base_url + '?' + query_string + '&hash=' + hash
print 'Create Ned resource holder'
print url, "\n"
data = json.load(urllib2.urlopen(url))
ned_resource_id = data['data']['id']
```

2) Now let's add the 213.29.27.0/24 IP block

```
query_string = 'target=ipam&action=add&rir=RIPE&block=213.29.27.0/24'
query_string += '&apiKey=' + api_key
hash = base64.b64encode( hmac.new(api_secret_key, query_string,
hashlib.sha256).digest() )
url = base_url + '?' + query_string + '&hash=' + hash
print 'Create 213.29.27.0/24 block'
print url, "\n"
data = json.load(urllib2.urlopen(url))
```

3) With the block in the system, we can assign 213.29.27.0/24 to "Ned" the Resource Holder

```
query_string =
'target=ipam&action=directAssign&block=213.29.27.0/24&resourceId=%d' %
(ned_resource_id)
query_string += '&apiKey=' + api_key
hash = base64.b64encode( hmac.new(api_secret_key, query_string,
hashlib.sha256).digest() )
url = base_url + '?' + query_string + '&hash=' + hash
print 'Assign 213.29.27.0/24 block to Ned'
print url, "\n"
data = json.load(urllib2.urlopen(url))
```

4) Since we plan on assigning IPs out of this block, we should enable subassignments for 213.29.27.0/24

```
query_string =
'target=ipam&action=update&block=213.29.27.0/24&allowSubAssignments=true'
query_string += '&apiKey=' + api_key
hash = base64.b64encode( hmac.new(api_secret_key, query_string,
hashlib.sha256).digest() )
url = base_url + '?' + query_string + '&hash=' + hash
print 'Update 213.29.27.0/24 to allow sub assignments'
print url, "\n"
data = json.load(urllib2.urlopen(url))
```

5) Now let's create a Resource Holder "Tara"

```

query_string =
"target=resource&action=add&meta[type]=entry&meta[section]=resource-holder&meta[name
]=Tara&meta[parent_id]=%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 '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
]=Una&meta[parent_id]=%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 '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&assignedRe
sourceId=%d" % (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&assignedRe
sourceId=%d" % (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))

```

Release Notes

Release Notes

- ProVision 5.0.0
- ProVision 4.2.0
- ProVision 4.1.0
- ProVision 4.0.0
- ProVision 3.9.3
- ProVision 3.9.2
- ProVision 3.9.1
- ProVision 3.9.0
- ProVision 3.8.0
- ProVision 3.7.0
- ProVision 3.6.0
- ProVision 3.5.0
- ProVision 3.4.0
- ProVision 3.3.0
- ProVision 3.2.0
- ProVision 3.0.0
- ProVision 2.5

ProVision 4.2.0

The 4.2.0 release is a minor release that packs in some big updates and customer feature requests.

Contact 6connect at info@6connect.com to schedule a demo or get more information.

▼ [Click here to see 4.2.x Release Notes...](#)

- [ProVision 4.2.1](#)

New Features

Peering Revamp

[6connect Peering v2](#)

DHCP Integration into Resources

[DHCP Administration](#)

Bug Fixes

IM-1505 - IPAM Manage - Resource Filter improvements

IM-606 - Zone import optimizations for larger data sets

ProVision 4.2.1

ProVision 4.2.1 is a minor release with some customer feature requests.

Contact 6connect at info@6connect.com to schedule a demo or get more information.

New Features

(CFR denotes customer requested)

CFR-86: Add "enable sub assignments" to IPAM gadget "edit" functions

1234 main road
asdkjh dnm
Khandahar, 24876 G2873

ARIN Info

10.48.0.0 - 10.63.255.255

Allow sub assignments for this block

Region **DataCenter1** **VLAN**

Region

Notes

Notes

Select Tags

<input type="checkbox"/> Anycast	<input type="checkbox"/> BB	<input type="checkbox"/> BGP
<input type="checkbox"/> Cable	<input type="checkbox"/> Customer	<input type="checkbox"/> Dev
<input type="checkbox"/> DHCP	<input type="checkbox"/> DNS	<input type="checkbox"/> DSL
<input type="checkbox"/> Infrastructure	<input type="checkbox"/> Internal	<input type="checkbox"/> Loopback
<input type="checkbox"/> Management	<input type="checkbox"/> MPLS	<input type="checkbox"/> PTP
<input type="checkbox"/> Security	<input type="checkbox"/> Static	<input type="checkbox"/> Tower
<input type="checkbox"/> VMware	<input type="checkbox"/> VOIP	<input type="checkbox"/> VPN
<input type="checkbox"/> VRF	<input type="checkbox"/> Loopback A	<input type="checkbox"/> Loopback B

Save

Los Angeles, CA Anycast, PTP 2014-07-21

CFR-71: Require CRUD permissions for a user to delete DNS server -> zone links

IM-1261: Pagination for IPAM aggregates

Aggregate Blocks

All - IPv4 - IPv6 - DHCP

67.

67.21.0.0/20
67.221.241.0/24
67.221.255.0/24

Page: 1 2 3 4 5

65.39.196.0/23 – ARIN Manage

Hosts			Blocks			Resources		
Available	496	96.87%	Available	31	93.93%	6connect Available		96.87%
Assigned	16	3.12%	Assigned	2	6.06%	6connect Labz		1.56%
Holding	0	0.00%	Holding		0.00%	Apple		1.56%
Allocated	16	3.12%	Allocated	2	6.06%			

3.12% assigned

3.12% allocated

Filter By Mask: All Masks Go To Reporting

IM-1496: DNS - Add zone UI revamp

Create a DNS Zone ✕

Enter Zone Name Select Customer... +

IM-1184/IM-554: Option to allow imported zones to automatically add target server

Review Data

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

View: All Valid Warnings Invalid Ignored Hide

Zone Name: Resource Holder: View Save

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

IM-1413: Reporting for IPAM, improve aggregate selection dropdown so it doesn't conflict with the IP Type checkboxes

Bug Fixes

IM-1500: Remove old remaining dns files from code base

IM-1154: Relabel "Transfer Type: SCP" to "ISC BIND"

IM-1126: Improved UTF-8 support for data imports

IM-944: Restructure Import Classes for more flexible data import formats

IM-323: logAction updates for global logging

IM-320: Cleanup of .txt file references/dependencies

ProVision 4.1.0

The 4.1.0 release is a minor release that packs in some big updates and customer feature requests.

Contact 6connect at info@6connect.com to schedule a demo or get more information.

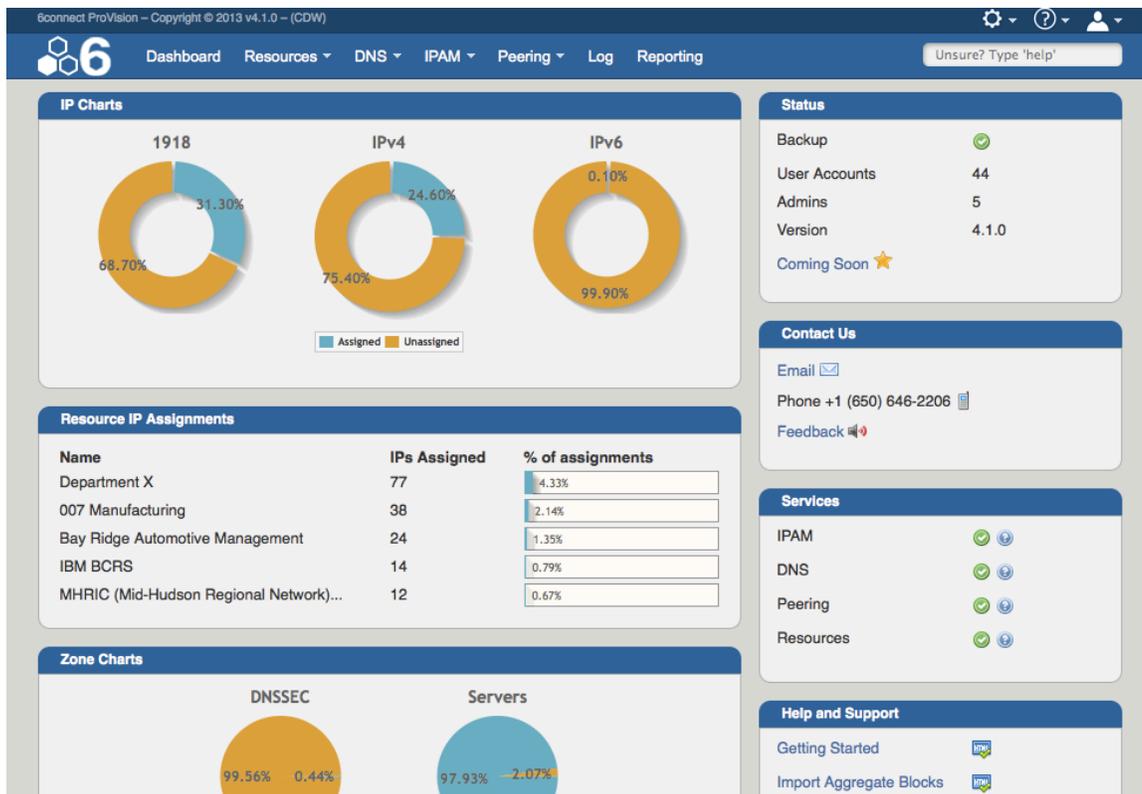
Click here to see 4.1.x Release Notes...

- ProVision 4.1.1
- ProVision 4.1.2
- ProVision 4.1.3
- ProVision 4.1.4
- ProVision 4.1.5
- ProVision 4.1.6
- ProVision 4.1.7
- ProVision 4.1.8
- ProVision 4.1.9
- ProVision 4.1.10
- ProVision 4.1.11
- ProVision 4.1.12
- ProVision 4.1.13
- ProVision 4.1.14
- ProVision 4.1.15
- ProVision 4.1.16
- ProVision 4.1.17
- ProVision 4.1.18
- ProVision 4.1.19
- ProVision 4.1.20
- ProVision 4.1.21
- ProVision 4.1.22
- ProVision 4.1.23

New Features

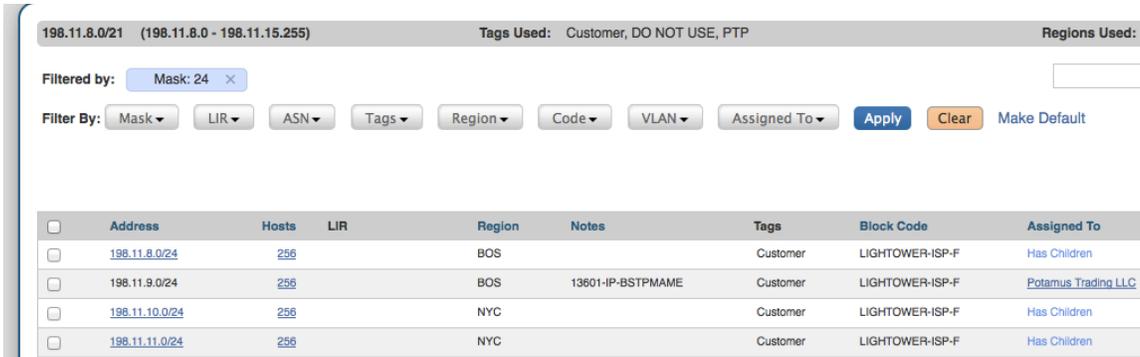
Dashboard Redesign

The main dashboard is completely redesigned with pie and bar graphs for a variety of IPAM, DNS, and resource data.



IP Detail View Options

View your IP aggregates in the way you want to view them. Set filters for each aggregate based on Mask, LIR, ASN, Tags, Region, Code, VLAN, or Resource Assignment.



198.11.8.0/21 (198.11.8.0 - 198.11.15.255) Tags Used: Customer, DO NOT USE, PTP Regions Used: 1

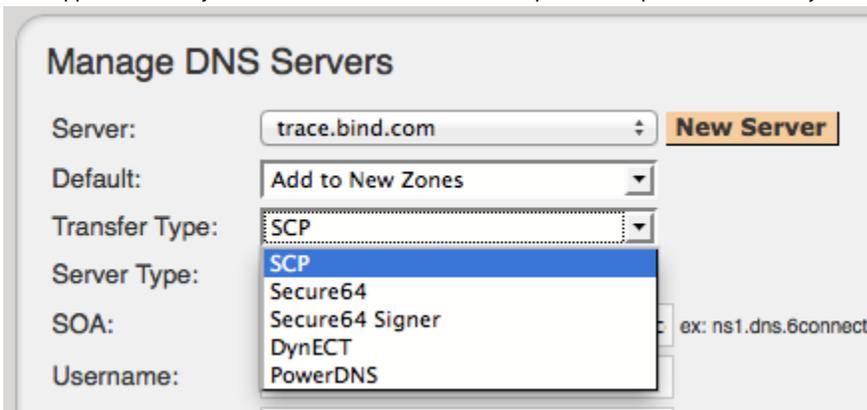
Filtered by: Mask: 24

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

<input type="checkbox"/>	Address	Hosts	LIR	Region	Notes	Tags	Block Code	Assigned To
<input type="checkbox"/>	198.11.8.0/24	256		BOS		Customer	LIGHTOWER-ISP-F	Has Children
<input type="checkbox"/>	198.11.9.0/24	256		BOS	13601-IP-BSTPMAME	Customer	LIGHTOWER-ISP-F	Potamus Trading LLC
<input type="checkbox"/>	198.11.10.0/24	256		NYC		Customer	LIGHTOWER-ISP-F	Has Children
<input type="checkbox"/>	198.11.11.0/24	256		NYC		Customer	LIGHTOWER-ISP-F	Has Children

PowerDNS - MySQL Support

Full support for the MySQL PowerDNS backend. Zone updates are pushed out in easy SQL updates to the target server.



Manage DNS Servers

Server: trace.bind.com **New Server**

Default: Add to New Zones

Transfer Type: SCP

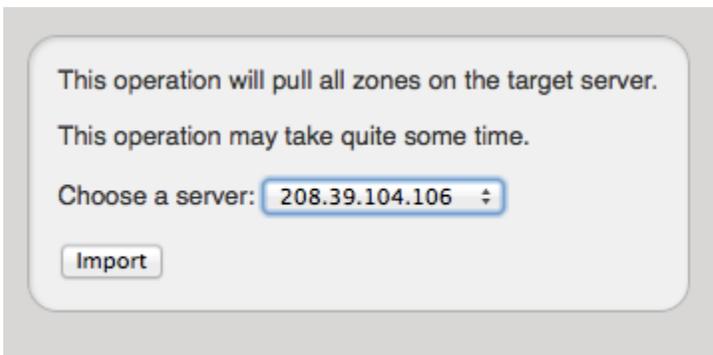
Server Type: SCP
Secure64
Secure64 Signer
DynECT
PowerDNS

SOA: ex: ns1.dns.6connect.r

Username:

PowerDNS MySQL Importer

Just enter the target server information, click import, and watch all your zone data populate the 6connect database.



This operation will pull all zones on the target server.

This operation may take quite some time.

Choose a server: 208.39.104.106

Import

Global Tag Update/Delete Functionality

Update and delete tags globally from the primary tags list, just as nature intended.

DNS Audit Tools (Beta)

View both forward and reverse DNS for entire /24 to get a clear view for how the block is used. Audit against public DNS to see if records are mis-matched or out of sync.

CLI (Alpha)

Hate clicking and have a general disdain for UI? This feature is for you! Just press ctrl+shift+s to try 6connect on the CLI with simple commands for quick no-frills interaction.

```

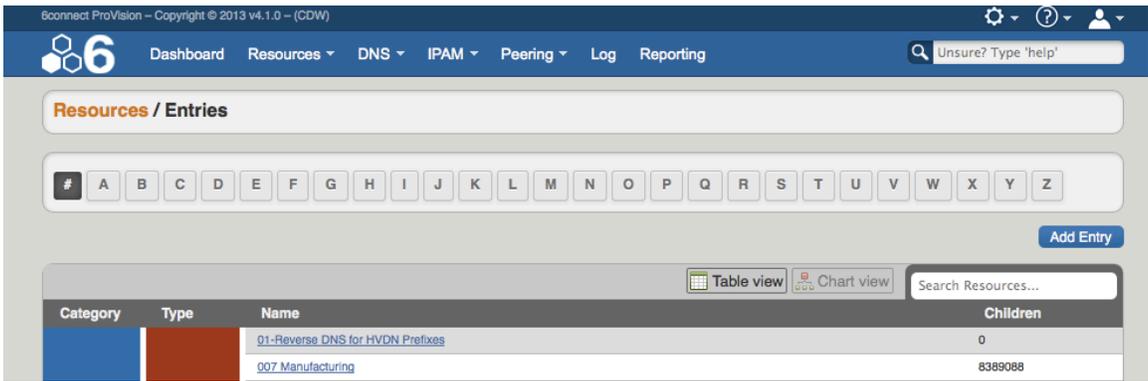
6connect> ipam man
ipam <command> [-t] [<cidr>] [<resource name>] [<args>]
show: show details for a block. Examples:
      - "ipam show 10.0.0.0/8" will show details for the block 10.0.0.0/8
      - "ipam show holding" will show details for all blocks in the Holding Tank
      - "ipam show "<resource name>" will show details for all blocks assigned to <resource name>
add: add a block. ex: "ipam add 192.168.0.0/24"
update: update attributes for a block. ex: ipam update 192.168.0.0/24 --vlan=100 tags=VM,Dev
assign: assign a block to a resource. ex: ipam assign 192.168.0.0/24 "<resource name>"
assign: smart assign a block to a resource. ex: ipam assign --mask=24 --rir=ARIN --type=ipv4 "<resource name>"
unassign: reclaims a block from a resource and places it in the Holding Tank. If the block is already in the holding tank, reclaims it and makes it available.
6connect> |

```

Feature Improvements

Resource UI Functionality

Easier navigation and button placement, alphabetical browse, data usage charts and more!

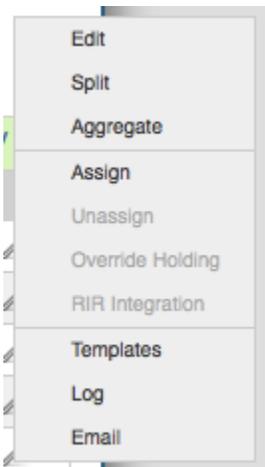


Header Redesign

The header and menu bars have been streamlined to take up less space, and present menu/options in a clean format.

Option Menu Redesign

Say goodbye to icons. All option menus have been switched to simple text drop down for easier and more intuitive interaction.



IP Import Option Additions

You can now import all data relevant to an IP including Resource Holder ID, Resource ID, Org ID, Net Handle, Customer Handle, Generic Code, ASN, Allow Sub-Assignments, and of course all the basic information.

Step 1: New Import
 The IP Import accepts CSV files in a variety of configurations and formats. For an example file, [click here](#).

Job Name: Description: No file selected.

File must be in CSV Format.

Step 2: Define Columns
 The Import process requires you to enumerate the function of the columns in the provided CSV. There must be either a CIDR column or both an IP and a Mask column. There can be multiple Notes, Tags, and Regions fields. Either a defined RIR column or a Default RIR is required.

Resource Holder ID	IP	Mask	Notes	RIR
Allow Subassignments	IP Block	subnet	Notes	RIR
ASN	10.2.3.0	/24	Internal space	1918
CIDR String	10.5.3.2	/32	Customer assign	ARIN
Generic Code	10.10.10.0	/30	PTP	1918
IP	10.1.0.0	/24		ARIN
Mask	10.2.0.0	/24	Development	1918
Net Handle	10.4.0.0	/24	Ticket 1515	1918
Notes				
Org ID				
Region				

Default RIR:

Step 3: Summary
 The importer has detected references which do not currently exist in the system. Rows referencing non-existent data cannot be imported. Please select the

Bug Fixes

- IM-919 - IPAM menu UI issues in IE9
- IM-1054 - DNS Add Zone allowing spaces in domain name
- IM-1057 - DNS edit - hitting enter key to save record creates strange pop up in Safari
- IM-1119 - Odd session behavior on API calls
- IM-1162 - Read Only user can edit IPAM blocks from IPAM gadget
- IM-1163 - Read Only user can't see time machine data from IPAM gadget

ProVision 4.1.1

The 4.1.1 is bug fix release only.

Bug Fixes

- IM-1193 - Path to zones incorrect in 6connect conf file

ProVision 4.1.2

The 4.1.2 release is a bug fix release only.

Bug Fixes

- IM-1196 - Error on IP Imports (php 5.4 and up)

ProVision 4.1.3

The 4.1.3 is bug fix release only.

Bug Fixes

- IM-1159 - Can't generate Resource report .csv download
- IM-1170 - Update copyright text

IM-1183 - Memory allocation exhaustion on large PowerDNS import
IM-1197 - IPAM view aggregate tree not restricting to top block
IM-1201 - IPAM gadget not displaying addresses
IM-1204 - Javascript error when splitting blocks
IM-1205 - IPAM Gadget browse assign does not have correct IP type set
IM-1207 - IPAM Gadget RIR integration broken

ProVision 4.1.4

The 4.1.4 is bug fix release only.

Bug Fixes

IM-1270 - Fix table name in 4.1.0 upgrade script

ProVision 4.1.5

The 4.1.5 is bug fix release only.

Bug Fixes

IM-1187 - Reporting Resource count incorrect
IM-1194 - Javascript errors on dashboard (text not properly escaped)
IM-1211 - Update EULA
IM-1212 - Update text on Welcome
IM-1215 - Assigned IP count on dashboard inaccurate
IM-1216 - IPAM - can aggregate assigned block
IM-1217 - IPAM - Sub-assigning block to same resource results in being unable to reclaim block and other weirdness
IM-1222 - Move HOSTED_COMPANY define from constants to globals

ProVision 4.1.6

The 4.1.6 is bug fix release only.

Bug Fixes

IM-990 - ResourceQuery returning incorrect results from empty arrays
IM-1060 - Edit DNS zone missing mouse over text
IM-1155 - IPAM Manage Email function does nothing
IM-1199 - DNS Import Error
IM-1202 - IPAM utilization incorrect or unsplit aggregates
IM-1213 - FireFox on Windows - Admin text areas too wide
IM-1220 - IPAM edit multiple not working as expected
IM-1221 - IPAM manage loading gif not going away
IM-1226 - DNS import does not correctly handle multi-line zone declarations
IM-1228 - Warning on dashboard when no data present

IM-1232 - IP assigned totals incorrect on dashboard

IM-1233 - File swipNetBlock spelled incorrectly - problems on case sensitive systems

IM-1235 - SQL templates contain log data

IM-1238 - SQL template contains 6connect default SOA, should be generic

IM-1241 - SQL template should have more generic default IP list

IM-1242 - SQL template contains /67 in default v4 masks

IM-1243 - SQL template contains bad assortment of default v6 masks

IM-1244 - IPAM admin contains lists for v4/v6 exact matches - no longer used

IM-1245 - SQL template default LIR should contain more generic data

IM-1246 - Process holding tank completed text should contain resource name, not number

IM-1250 - IPAM delete throwing error, instead of returning error on bad delete action

ProVision 4.1.7

The 4.1.7 release is a bug fix release that contains a few feature improvements.

Feature Improvements

IM-1300 - Add all RIRs to LIR manager - All RIR's are now standardized as 1918, ARIN, RIPE, LacNIC, APNIC, and AfriNIC). All customer RIR's should be moved to LIRs and the next update will provide automatic migration.

IM-1218 - Move Resource Holder ID to fundamental attributes for type Resource Holder - The Resource Holder ID is now the Customer ID under the fundamental resource properties.

IM-1268 - Handle multiple ORGs per LIR for ARIN - Added the ability to handle multiple ORGs per LIR for the ARIN RIR.

IM-1156 - Link Resources on Dashboard to Resource records.

IM-1269 - Move DNS and IPAM admin sections to their correct locations - IPAM and DNS admin options are now under the IPAM Admin and DNS Admin sections.

Bug Fixes

IM-1166 - Sorting functions inconsistent on zone list

IM-1195 - Remove old help URLs from admin

IM-1225 - Zone import giving incorrect error when conf file omitted

IM-1249 - Available/Reverse/Holding resources should be protected

IM-1251 - DNS server add by default behavior is not correct

IM-1271 - Field mis-alignment on IPAM gadget

IM-1275 - EULA has escaped quotes

IM-1298 - Default SOA for zones being used when over ridden

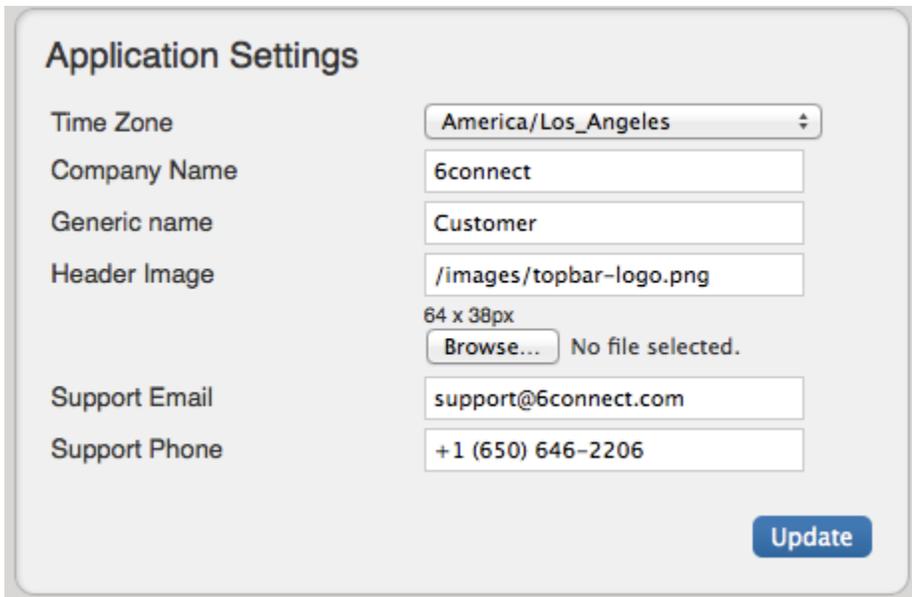
IM-1299 - IP Import not working for Radius/LDAP users

ProVision 4.1.8

The 4.1.8 release is a bug fix release that contains a new feature improvement.

Feature Improvements

IM-1255 - User customizable header and support links. Under Admin->Application Settings, you can now set your own custom header image (note the image size), as well as email and phone number to contact for support.



Application Settings

Time Zone	America/Los_Angeles
Company Name	6connect
Generic name	Customer
Header Image	/images/topbar-logo.png 64 x 38px Browse... No file selected.
Support Email	support@6connect.com
Support Phone	+1 (650) 646-2206

Update

Bug Fixes

- IM-1317 - IPAM Gadget - Region dropdown is blank when editing
- IM-1313 - \$GENERATE on DNS not producing appropriate PTRs
- IM-1310 - Admin-Users - Check User Permissions field no longer working
- IM-1306 - Unable to add maintainers to new LIRs
- IM-1302 - Can't Smart Assign when using the LIR parameter
- IM-1237 - Admin->API page loading very slow even with no data
- IM-1229 - RIR Integration missing link to LIR Manager if no LIRs exist
- IM-1182 - API - fix request generator to updated resource parameters
- IM-1121 - Logging message missing data for outbound contact email

ProVision 4.1.9

The 4.1.9 release is a bug fix release that contains a few new feature improvements.

Feature Improvements

IM-1258 - Add separate name field to DNS servers, so there is both an FQDN and name. There are now a separate fields for a server Nick Name and it's FQDN/IP. Now DNS servers can be easily recognized by their common/nick name or their FQDN/IP address, as both are displayed representing the server.

Manage DNS Servers

Server: New Server

Nick Name:

FQDN or IP: ex: ns1.dns.6connect.net or 216.239.32.10

Default:

Transfer Type:

Server Type:

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

IM-1329 - Add field override for Net Name on swip.

IM-1312 - Add override field for Customer Name on swip

Both the ARIN Customer Name and Net Name can be over ridden from the automatically generated defaults at swip time using the fields provided on the RIR Integration Panel.

ARIN Integration: 67.221.244.0/28 (67.221.244.0 - 67.221.244.15)

Org Handle	Admin POC	Net POC	Abuse POC	Net Name Prefix	API Key
<input checked="" type="radio"/> CONNE-81	admin-c	tech-c	abuse-c	NET	API-B7BF-F4AD-4695-8508

Net Name:

Registrar Public Name (Simple Reassign only):

By default, when ARIN blocks are SWIPed the customer name in the WHOIS database will be set to the assigned resource name. To override this, enter a public name to use in this field.

Bug Fixes

IM-1330 - API - IPAM get parameter block not working as expected

IM-1324 - Missing images on admin-dhcp.php

IM-1323 - API - Inconsistent results between Resource Get and Resource Add

IM-1322 - IPAM Gadget - Action menu display issue sub assignments allowed

IM-1321 - IPAM Gadget - All tags should display during edit, not just in use

IM-1319 - IPAM Admin - Bad URL "Back to IPAM Admin" in tag editor

IM-1259 - configTest - Add test items for expect, namp, and php hash extension

IM-1142 - Center command results given by assistant

ProVision 4.1.10

Bug Fixes

IM-1290 - Resource: list view does not have default when Resource link clicked - defaults to empty list.

IM-1320 - Resource: allowed to become it's own parent.

IM-1327 - Dashboard: displaying information for resources above current sign in permission level.

IM-1331 - Import : progress bar flashes/drops to 0 at times while upload in progress.

IM-1334 - Import: new resources not adding correctly from IP import.

IM-1337 - API: Valid requests are being rejected.

IM-1338 - License check interval too long

ProVision 4.1.11

New Features

DNS Deletion Queue

Addition of a deletion queue for zones for flexible removal of zones from DNS Servers.

6connect ProVision - © 2014 v4.1.1 - (Awesome Co.)

Dashboard Resources DNS IPAM Peering Log Reporting Search or type help

Create a DNS Zone

Enter Zone Name Select Awesomers...

Page: 1 (14 total)

Zone	Customer	Tags	DNSSEC	DS	Records	Actions
1800druidia.com	Apple Computer				2	
blimey.com	Acer Worldwide				2	
cnn.com	Acer Worldwide				0	
expect.com	Apple Computer				2	
fabulous.com	APC Schneider				2	
fancy.com	Acer Worldwide				2	
happybirthday.com	Joe's Bait and Tackle and ISP	super	DNSSEC	X	2	
happyfestivus.com	Joe's Bait and Tackle and ISP	less super			4	
jd-sweatshop.com	Jack Daniel's Sweat Shop				2	
memychristmas.com	Joe's Bait and Tackle and ISP	most super			0	
tuoo.com	Acer Worldwide				2	
zone1	Anna's Test Site				3	
1.1.62.in-addr.arpa	Jack Daniel's Sweat Shop				2	
0.0.0.5.b.d.0.1.0.0.2.ip6.arpa	APC Schneider				1	

- Edit Tags
- View
- Delete
- Delete & Push
- Reassign
- Log

Bug Fixes

IM-1344 to IM-1351 - All bug fixes related to a bad version of bootstrap.

ProVision 4.1.12

Bug Fixes

The 4.1.12 release is bug fix only release.

IM-1372 - Search - Search box breaks on Admin navigation when large custom header image set

IM-1371 - Dashboard - Reverse DNS resource appearing on dashboard stats

IM-1369 - DNS - Large DNS pushes appear to be hanging

IM-1368, IM-1343 - Reporting - CSV and HTML column headers do not match values

- IM-1366 - DNS - Menus not appearing on pages after 1
- IM-1361 - DNS - Admin DNS "Edit DNS Records" links to nonexistent page (404 error)
- IM-1360 - IPAM - Admin IPAM "Process Holding Tank" Back to IPAM links to nonexistent page (404 error)
- IM-1358 - DNS Import - Can't edit resource assignment during import
- IM-1357 - DNS Import - Hanging on single zone import
- IM-1355 - IPAM - Aggregate details log menu not pulling log entries
- IM-1354 - Reporting - Error on user reporting
- IM-1352 - Reporting - All users showing in user list when permissions should restrict
- IM-1335 - IPAM Import - Data not revalidated after editing import row
- IM-1305 - DNS - Download of all zip files permissions issue
- IM-1301 - Resource - Customer field type box disappearing on add
- IM-1296 - Admin - Create backup now option updates general settings
- IM-1224 - Admin - API request generator slow to load
- IM-1223 - IPAM - LIR manager not saving Org info
- IM-1039 - Resource - ZIP/Postal code field missing in edit dialog

ProVision 4.1.13

New Features

Add User CLI (IM-1387)

Local installs can now add a user from the CLI with the add-user tool found in the ./tools directory.

```

/tools> php add-user.php
Add New User - 6connect ProVision
-----
62 - Remove edit/delete options from ipam tags list, options no longer needed
Enter email (username):
gary@6connect.com
First Name:
Gary
Last Name:
Canty
Password:
changemefast!
Add to Global Admins? (y/n):
y
5 - DNS - Delete and Push options shows in action menu when user does not have Delete per
User `gary@6connect.com` was added.

```

Updates

- IM-1385 - IPAM - RIPE API updates/invalid RIPE API URIs
- IM-1382 - Remove edit/delete options from ipam tags list, options no longer needed

Bug Fixes

- IM-1389 - Reporting - Utilization numbers incorrect for IPAM reporting
- IM-1386 - IPAM API - Direct Assign - resourceHolderId parameter not working as expected
- IM-1383 - Session timeout logs do not set username

IM-1375 - DNS - Delete and Push options shows in action menu when user does not have Delete permissions

IM-1374 - Resource - Unable to save resources on edit (javascript error)

IM-1359 - Dashboard - IPAM/DNS graphs inaccurate for some users, counting data out of permissions scope

IM-1041 - IPAM Admin - Edit Contact Roles link depreciated, should be removed

ProVision 4.1.14

New Features

Rebuild Permissions CLI Tool (IM-1405)

Command line tool to rebuild the user group permissions for a specific group or resource, or group/resource combination.

Bug Fixes

IM-1404 - User Admin - Group Management page error 'Circular Reference

IM-1403 - Dashboard - Warnings on dashboard when user has no IPAM permissions

IM-1402 - Reporting - IPAM reporting assigned numbers still incorrect

IM-1401 - DNS - Text update for Delete and Push confirmation window

IM-1400 - DNS - Default servers not added to zones as expected

IM-1399 - IPAM API - IPAM get call ignoring valid tags

IM-1392 - DNS - Action menu disappears when using filter function

IM-1390 - Admin DNS - Push to Secure64 errors (short timeout)

ProVision 4.1.15

The 4.1.15 release is a bug fix release that contains a few feature improvements.

Feature Improvements

CFR-76 - Add Configuration setting for submitting changes to RIPE test v. live database

IM-140 (part) - Password management improvements - stored passwords are now stored outside of primary database structure.

Bug Fixes

IM-1395 - DNS - Assign DNS zone drop down displaying incorrect information

IM-1396 - IPAM - SWIP check mark displayed when IP is only assigned

IM-1421 - IPAM - Smart assign producing two log messages (should be one)

IM-1423 - API - Hashed API requests fail if first character of key is 0

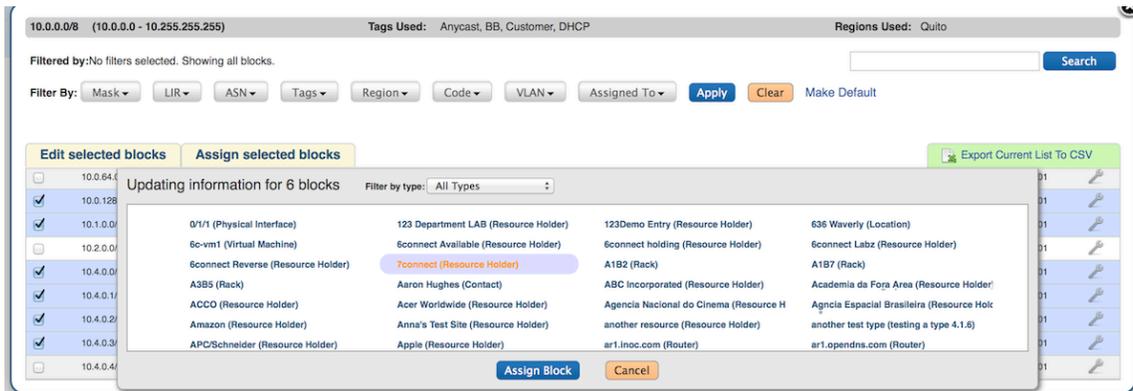
IM-1424 - IPAM - Scan loading indicator never goes away

ProVision 4.1.16

New Features

IPAM - Multi-assign

Assign multiple IP addresses to a resource via a single API call, or just a few clicks in the user interface! When multiple IP addresses are selected in the UI, a new tab to assign all at once will appear.



Bug Fixes

IM-1418 - IPAM - De-Swip estimated time too high

IM-1426 -IPAM - Filters resetting to default settings after multiple edits

IM-1427 - Resource - Chart view fatal error

IM-1428 - IPAM - Swip timer counting negative

ProVision 4.1.17

Beta Features

Resource Importer

A new stand alone application to import complex resource sets. Screen shots TBA.

Improvements

PowerDNS Server Connection Test

Test and diagnose connectivity issues to a PowerDNS server.

Manage DNS Servers

Server: [New Server](#)

Display Name:

FQDN or IP: ex: ns1.dns.6connect.net or 216.239.32.10

Default:

Transfer Type:

Server Type:

Backend Type:

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

Username:

Password:

DB Username:

DB Password:

DB Port:

DB Name:

[Test Config](#) [Update Server](#) [Delete Server](#)

Connected socket to 208.39.104.106:3306 but MySQL connection failed: could not find driver

Bug Fixes

- IM-1411 - DNS - Add "Test Config" option for PowerDNS servers in DNS Admin
- IM-1430 - IPAM - Direct assign using block parameter throws error
- IM-1431 - Setup - configBootstrap allows an empty registration email

ProVision 4.1.18

Beta Features

Resource Importer

A new stand alone application to import complex resource sets. Give it a test run!

Improvements

- Resource Importer - Add default name server columns to import
- Resource - Add count of zone to Customer section display

Bug Fixes

- IM-1277 - Dashboard - Resource zone assignments display even with no data
- IM-1291 - Resource - Ability to move system resources via Resource Chart View

IM-1292 - Resource - Ability to move TLR anywhere in tree - subsequent crash

IM-1328 - Admin - Radius dictionary and LDAP schema links broken

IM-1442 - Resource - Redirect non-existent categories to uncategorized category

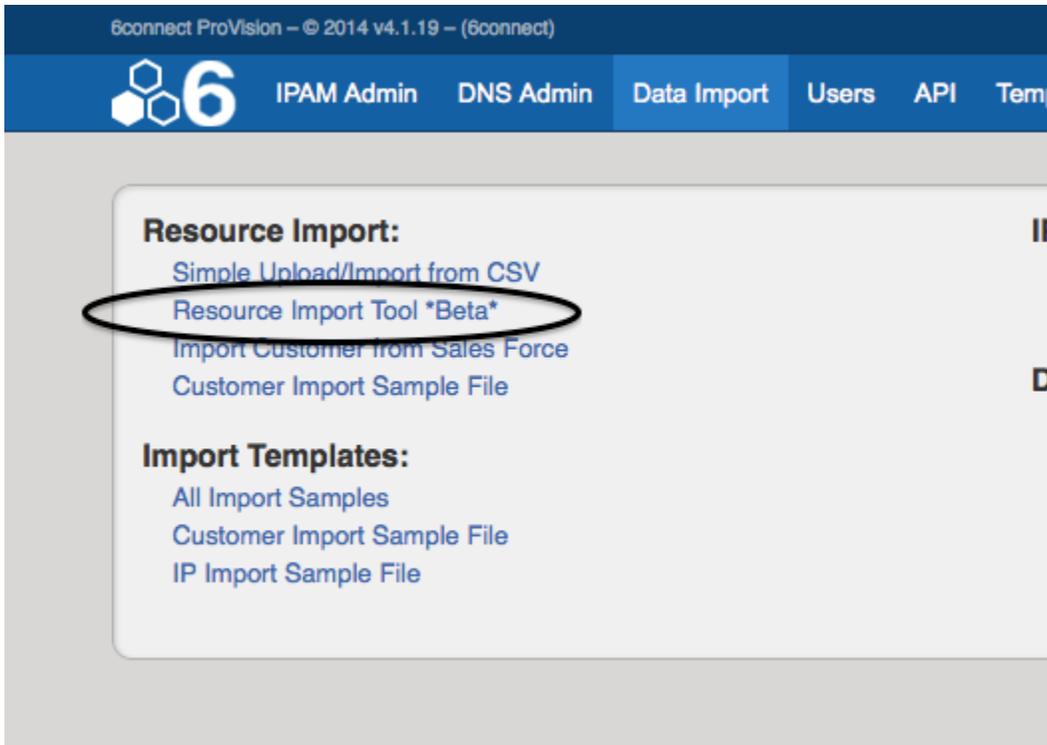
IM-1446 - IPAM - Add Swip error message to error displayed in UI on failure

ProVision 4.1.19

Beta Features

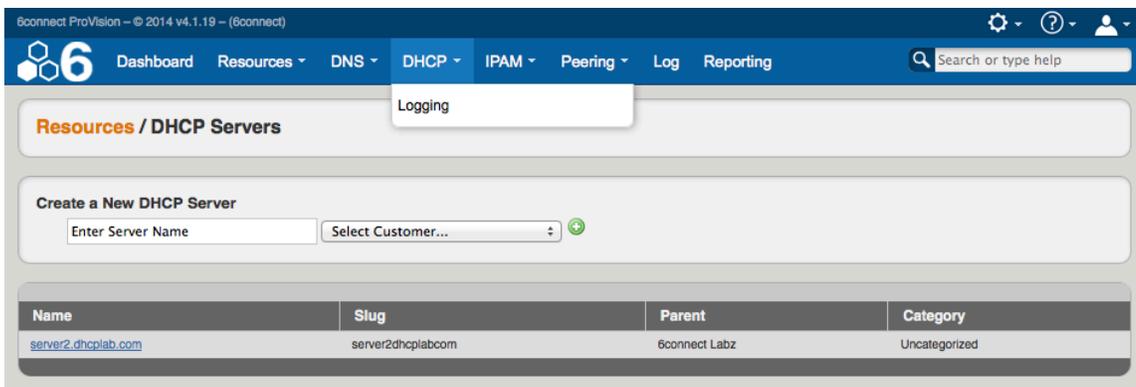
Resource Importer

A new stand alone application to import complex resource sets. Give it a test run! Go to Admin -> Data Import and click on the Link for "Resource Import Tool Beta". As always, please send feedback to support@6connect.com.



DHCP Management

New DHCP Tab - See all your DHCP server entries in a single place. The usual search functions work as well.



DHCP Management (Now with more Resources!) - We have integrated DHCP functions into a "Gadget". This means you can assign DHCP functionality to a Server of your choice and manage it just like any other resource in ProVision.

+ Connection Configuration

Manual IP:

Notes:

SSH

Username:

Password:

Port:

+ Server Details

+ Server Details

DHCP Vendor:

DHCP Config File Path:

Server Options

Routers:

Domain Name Servers:

Domain Name:

Free Lines (appended to DHCP Server Config):

1)

2)

3)

Add a new Line:

Server Commands

Config Test:

Server Stop:

Server Start:

Advanced Options

Authoritative:

Default Lease Time:

Max Lease Time:

Local Port:

Log Facility:

+ DHCP Pools

Linked Pools

TestHost1 [Host]	
TestHost2 [Host]	
TestHost3 [Host]	
TestHost4 [Host]	
TestSub1 [Subnet]	
TestSub2 [Subnet]	
TestSub3 [Subnet]	
TestSub4 [Subnet]	

+ Create a New DHCP Pool

Create a new

Subnet Name: (ex: Lab #1)

New IP Assignment:

Free Lines: No lines saved.

Add a New Line:

Improvements

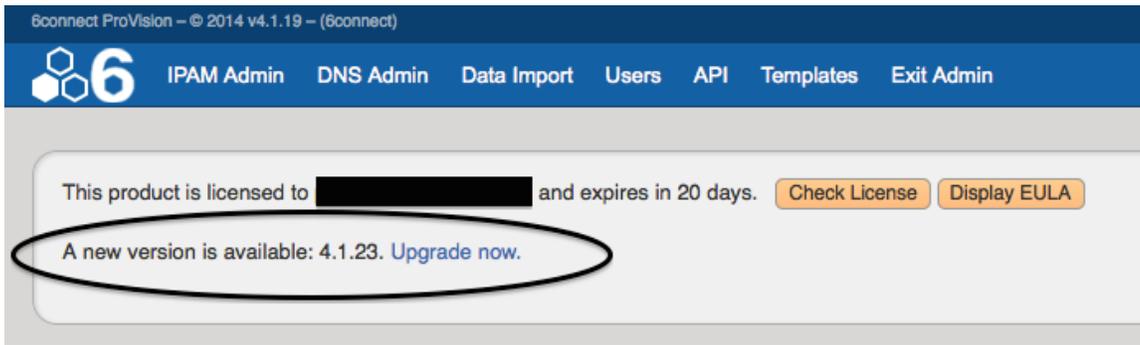
Automatic Upgrades

You can now upgrade locally installed ProVision instances via the GUI, API, or command line. The product will automatically check if any new updates are available, fetch and install them. Note that you can only use this feature from 4.1.18 forward. If you are on an older version, please contact support@6connect.com to assist you.

GUI - Just click on Admin. Information will be at the top of general admin page.

API documentation: <http://docs.6connect.com/display/DOC/API+Module++Upgrade>

Command line: `<6connect web root>/upgrade/upgrade.php (-h for help)`



ARIN Swip Error Messages

ARIN swip error messages now show the full detail of the error in the GUI.

Bug Fixes

IM-1453 - Resources - Unable to remove existing fields from field collection

IM-1454 - IPAM - IPv6 radio button typo

IM-1459 - IPAM - Manage screen does not auto scroll/highlight block after edit

ProVision 4.1.20

Bug Fixes

IM-1463 - DNS - Create zones drop down pulling incorrect resource list (DHCP)

IM-1449 - IPAM - Add loading/waiting indicator when waiting for swip/allocate transaction

IM-1332 - Import - File type issues, can't import files that are not Windows csv

ProVision 4.1.21

Improvements

Call Subnet Scans via the API

Subnet scan can now be issued and retrieved via the API. Examples of the two commands and links to documentation are listed below:

```
/api/v1/api.php?target=ipam&action=scanBlock&block=<cidr>
```

```
/api/v1/api.php?target=ipam&action=getScanResults&block=<cidr>
```

[API Module - IPAM](#)

Call Swip Functions via the API

[API Module - IPAM](#)

Delete Functionality for All Blocks

Aggregates can now be deleted regardless of state of sub-blocks. An aggregate can be deleted even if sub-blocks or hosts are assigned to resources.

11.1.2.0/24 – ARIN [Manage](#)

Hosts			Blocks			Resources	
Available	191	74.60%	Available	7	77.77%	LargerTest	74.60%
Assigned	65	25.39%	Assigned	2	22.22%	ABCAM	25.00%
Holding	0	0.00%	Holding	0	0.00%	1stdibs.com	0.39%
Allocated	256	100.00%	Allocated	9	100.00%		

IPAM Aggregate Statistics

All IPAM aggregates now display statistics on how they are assigned to resource on the main IPAM screen.

11.1.2.0/24 – ARIN [Manage](#)

Hosts			Blocks			Resources	
Available	191	74.60%	Available	7	77.77%	LargerTest	74.60%
Assigned	65	25.39%	Assigned	2	22.22%	ABCAM	25.00%
Holding	0	0.00%	Holding	0	0.00%	1stdibs.com	0.39%
Allocated	256	100.00%	Allocated	9	100.00%		

Resource Field Management

A new easy to read display of field usage statistics for usage of specific resource field names, as well as usage of field types, and a list of all available fields for improved resource field management.

Resources / Field

Most Used Fields

Field Name	Usage Count (Approximate)
First Name	1830
ARIN POC Id	1525
Phone Main	1220
Street 2	915
Company Name	610
State	305
City	305
Zip	305
Street 1	305
Country	305

Field Types

Field Type	Count
Text	69
Number	5
Choicebox	3
Textarea	1
Checkboxes	0
Radios	0

Name

- [2nd Email](#)
- [2nd Phone](#)
- [Abuse POC](#)
- [Abuse POC](#)
- [Admin Contact](#)
- [Admin POC](#)
- [API Key](#)
- [ARIN POC Id](#)
- [ASN](#)
- [Asset Tag](#)

Bug Fixes

CFR-87 - IPAM - Consistently display Region name across IPAM management and gadgets
IM-1467 - Resource - Unable to enter Canadian Provinces
IM-1476 - IPAM - Smart Assign generic code parameter being ignored
IM-1477 - IPAM - Remove ability to dis-allow sub-assignments on blocks with children
IM-1482 - Resource - Can't edit fields in Section in Chrome/Safari
IM-1484 - Resource - Fix verbiage around Types and Sections, change all to Sections
IM-1486 - Resource - Remove system TLR field from breadcrumb display

ProVision 4.1.22

Bug Fixes/Feature Updates

IM-1493 - IPAM - Invalid LIR ID or entity handle error message when performing SWIP action
IM-1124 - DNS - PTR autogenerate - UPDATE

Reverse DNS

ProVision can be configured auto-generate missing IPv4 PTR records in reverse zones based on templates. The default configuration values here can be overridden by local configuration values on individual zones. At this time this feature is limited to zones which cover /24 sized blocks. The variables '\$oct1', '\$oct2', '\$oct3', '\$oct4' may be used. They correspond to the first through fourth octets of the PTR's IP.

Generate missing IPv4 PTR records by default	<input checked="" type="checkbox"/>
PTR Host Template	<input type="text" value="\$oct4"/>
PTR Value Template	<input type="text" value="\$oct4.available.lighttower.net"/>

ProVision 4.1.23

Bug Fixes

IM-1494 - IPAM - Incorrect SWIP parameters sent from IPAM gadget

ProVision 4.0.0

The 4.0.0 release is a major release that contains many new features and improvements. The improvements include: a complete revamp of the Resource (Asset) system, an LIR/RIR management system to manage multiple ASNs and RIR credential sets, a new permissions structure with user groups and Resource based access, global search auto-complete, and an improved GUI setup (available since 3.9.3).

New Features

RIR/LIR Manager

If you have multiple ASNs, LIRs, or RPSL Maintainer Objects to manage - this is the feature for you. You can now enter an unlimited amount of ARIN or RIPE credentials and select which set of credentials to use when updating an IP block. Additional detailed error reporting has been added to provide specific information in the event of an update failure.

The credentials for all LIR's under a particular RIR are displayed when the RIR Integration icon is clicked, allowing quick and easy switches of blocks between different organizations or maintainers.

Permissions

Control and set access to your IP data, DNS, and Resources exactly how you want it. Full CRUD permissions for each section of ProVision can now be controlled on per Resource basis through user groups. Previously, permissions were based on sections of the tool: IPAM, DNS, Peering, Resources, etc. and applied globally. Permissions for each section are now attached to individual Resources. Additionally, user groups have been added to simplify permission assignment. Full CRUD permissions for each section of ProVision can now be controlled on per Resource basis through user groups.

The new permission structure has completely replaced the Customer Portal functionality in versions previous to 4.0.0. Customer Portal users will be migrated to standard directory of users that have permissions only on their specific customer resource.

Note: The Radius and LDAP integrations have changed significantly with the addition of user groups. The configurations for 3.9.3 and prior will NOT work with 4.0.

For more information, see the documentation: <http://docs.6connect.com/display/DOC/Manage+Users>

Global Search Auto-complete

Find what you are looking for quicker and easier. Global search will now provide a drop down of auto-complete answers, and the section that contains the relevant results.

Resource (Asset) System

ProVision has a new ultra flexible asset management system that looks good while doing it. The Resource system has a smoother look and feel with pagination, search, and quicker, more detailed methods for creating custom fields and data types. Both Categories and types have been added to help better organize Resources.

And if you are a fan of pretty things, there is a new graphical Resource view that displays the hierarchical layout of Resources, and allows for drag/drop to reorganize Resources.

Documentation

We have moved to a new and improved documentation system at: <http://docs.6connect.com/>. Help documents are now fully indexed and searchable. Video tutorials are being added soon!

Improvements

No more changing text configuration files! 4.0.0 now has a GUI setup wizard to check the system configuration for errors, and load the configuration files and database from a few simple fields.

Minor Improvements

IPAM - generic code field added to API Smart Assign call. (CFR-44)
Authentication - set default login type via constant. (CFR-43)
Reporting - Add reporting functionality on Top Level IP Aggregates. (CFR-42)
DNS - List zones alphabetically, and then numerically by IP octet/nibble.

Bug Fixes

IM-812 - Confirming the Org ID of IANA on welcome page produces blank screen.
IM-922 - Successful Radius login rejected.
IM-455 - Do not allow duplicate zone creation.
IM-833 - Fix to save record on Zone View when hitting enter key.
IM-937 - Fix to close record using view icon on Zone View.
IM-941 - Fix for global search timeouts.
IM-940 - Updated import CSV header type checking to accommodate .csv files generated from a variety of different sources.

Minor Release 4.0.1

IM-1079 - IPs in Holding can be assigned via Browse Assign.

IM-1088 - Daemon user visible in accounts section.

Minor Release 4.0.2

IM-1099 - IPs in Holding can't be manually over ride reclaimed.

IM-1100 - Latest import code not checked into 4.0.0 branch.

Minor Release 4.0.3

IM-1102 - Contact info gadget information does not save on first click.

IM-1101 - IP block scan Found hosts number does not refresh on completion.

Minor Release 4.0.4

IM-1119 - Odd session behavior/results with api calls.

Minor Release 4.0.5

IM-1120 - Generate forward/reverse DNS for resources not working.

IM-1125 - Session Bug - User timeout/logout allows API request as user with no permissions

IM-1131 - Warning in search.php line 92

IM-1132 - Customer import from .csv field count issue

Minor Release 4.0.6

Maintenance release - Peering code integration

Minor Release 4.0.7

IM-1136 - ContactInfo gadget reading and saving incorrect field name for phone and fax

ProVision 3.9.3

Overview

The 3.9.3 minor release provides a few new DNS features.

New Features

Per Server DNS View Support

Enable, disable, and manage views on a server level. This enables every server to have it's own unique combination of views implemented, or none at all.

Secure64 DNS View Support

Support for views with the Secure64 DNS system.

Bug Fixes

None.

ProVision 3.9.2

Overview

The 3.9.2 minor release provides a few new DNS features, additions to the API request generator, minor IPAM changes, and minor bug fixes.

New Features

Secure64 DNS Integration

Secure64 Master/Slave integration and DNSSEC integration.

Manage DNS Servers

Server:

Default:

Transfer Type:

Server Type:

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

Username:

Password:

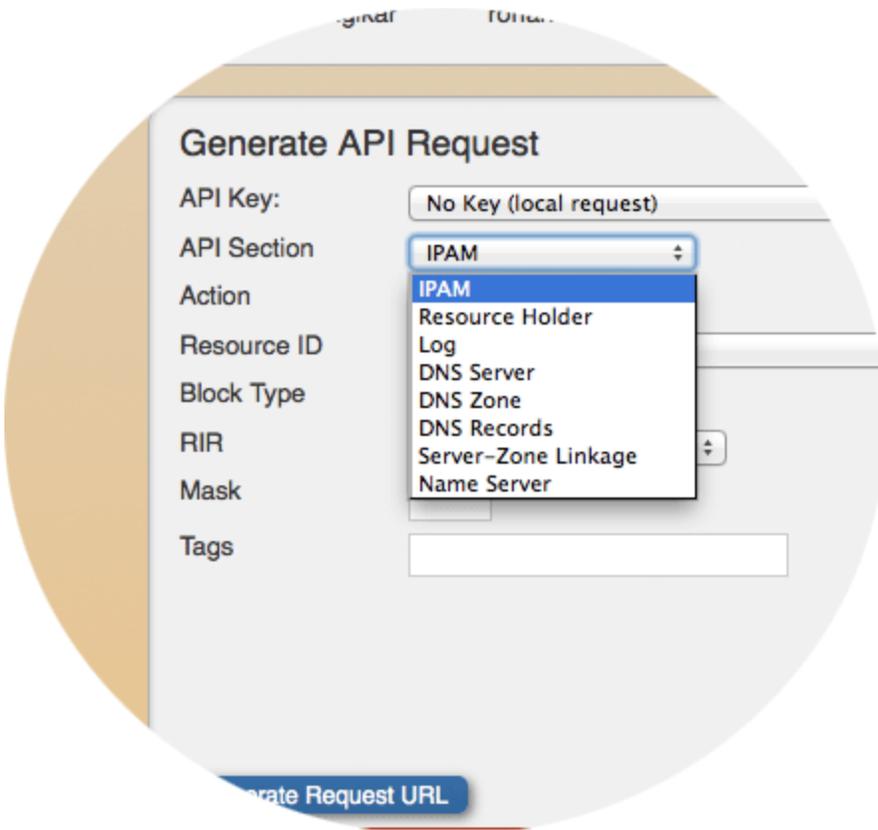
Port:

Remote Directory:

Named Conf Path:

API Request Generator - DNS Utilities

DNS Zones, DNS Records, DNS Servers, Server-Zone Linkage, and Name Server API sections have been added to the API request generator.



Changes and Updates

SWIP Information Display

When a block is swipped, it is now indicated by a green check mark next to the RIR name on the IP manage screen. Hovering over this check reveals the swip details.

67.221.240.0/20 (67.221.240.0 - 67.221.255.255)		Tags Used: No tags used		Regions Used: No regions used			
Address	Mask	RIR	Notes	Region	Assigned To	Updated	Actions
67.221.240.0/22	1000	ARIN			Available	2013-04-23 15:46:27	Σ 🛠
67.221.244.0/24	256	ARIN	67.221.244.0/24 (NET-67-221-244-0-1) SWIPed to CO3381230		BitandB, Inc.	2013-04-23 15:46:27	Σ 🛠
67.221.245.0/24	256	ARIN			Available	2013-04-23 15:46:27	Σ 🛠
67.221.246.0/23	512	ARIN			Available	2013-04-23 15:46:27	Σ 🛠
67.221.248.0/21	2,048	ARIN			Available	2013-04-23 15:46:27	Σ 🛠

Bug Fixes

Fix for IM-807 - Missing email address and exchange information on some peering request forms.

Fix for IM-808 - Scan results not displaying on IPAM manage screen.

Fix for IM-809 - Setting incorrect path for imports temp directory on local installs.

ProVision 3.9.1

Overview

The 3.9.1 minor release provides a few new IPAM features and improved data visualization. It also contains some changes and improvements DNS Master/Slave relationships, and the file structure created on BIND servers during file push.

New Features

IPAM Bulk Edit

There are now checkboxes by each IP block on the IP block management screen. When you select more than one check box, the option to edit multiple blocks appears at the top left of the IP list. Clicking "Edit selected blocks" will bring up the standard edit menu and all selected blocks will be updated to the properties chosen.

IPAM Export to .csv

The "Export Current List to .csv" link is located at the upper right of the Manage list. When clicked it exports all blocks displayed on the screen to a .csv file. You can export select sub-sets of blocks simply by filtering or searching the list.

DNS Master/Slave Relationships

On the zone edit page, there is now the ability to push a zone out to a server as a master or a slave to accommodate complex and mixed master/slave environments.

Bulk Zone Server Assignments

On the DNS Admin tab, the second second provides the ability to search for zones, then perform a bulk assignment of those zones to a specific server.

IP Assignments - Hierarchical View

On the IPAM manage screen, blocks that allow sub-assignments, and have at least one sub-assigned block become a link that expands to show a hierarchical listing of space within the block and what resources the space is assigned to.

Changes and Updates

BIND Directory Structure

The directory structure for BIND has been changed so that all zone files now go into a sub folder under ./6connectGeneric that is the first character of the zone files name, either 0-9 or a-z.

RIPE IP Look-up

IP based look-up on the welcome screen for RIPE blocks is now using an improved reverse lookup on mnt-ref and displays all discovered organizations associated with an IP, allowing the user to identify the correct organization.

Bug Fixes

Fix for missing time zone, state, and country information in the admin section and customer address section. This would have only impacted new turn ups between 3/14/2013 and present.

ProVision 3.9.0

Released March 2013

Network Discovery Scanlet (v1)

- IPv4 scanning agent for network discovery
- Discovers and imports IP, Ports and MAC Address as devices

ProVision - IPAM

- IP Sub-assignments
- IP Templates

ProVision - DNS

- DNS Views (Split Horizon Support): Includes ACL manager!
- Improved DNS Master/Slave support
- Secure64 DNS Integration (update)

ProVision - Assets

- Browse, Direct, and Smart Assign added to Assets
- Asset UI cleanup (multiple)

ProVision - Platform

- Logging fixed when entering duplicate blocks
- Propel ORM removal

ProVision 3.8.0

Launched in December 2012

ProVision - IPAM

- IP Import Wizard
- IP Templates/Splitting

ProVision - DNS

- Updated Zone Editor
- Zone History and Versioning (with rollback)

ProVision - Assets

- Migration to more flexible container structure

ProVision 3.7.0

Launched in September 2012

ProVision - IPAM

- Multiple UI updates, including new "Aggregate" view

ProVision - DNS

- PowerDNS support (v1)
- Zone editor refresh
- Zonelist page UI update
- Selective zone push (single zone, bulk push, etc.)
- Multiple DNS Server Support
- RFC based zone validation

ProVision - Assets

- Complete refresh, unlimited Asset Types and Custom Fields

Network Automation - Peering

- Complete UI refresh of Communication Manager and Session Manager

ProVision - Platform

- LDAP Support
- Revamped Tree View with real-time search

ProVision 3.6.0

Launched in August 2012

DNS

- Asynchronous zone push

DHCP

- ISC dhcpd support
- Support for common DHCP features
- DHCP configuration pushes and execution of custom commands
- Support for custom DHCP configurations

Portal

- Integrated UI updates from Resource Holder page

Platform

- Revamped Resource Holder page
- Updated Smart Assign function
- Multiple data visualization updates

ProVision 3.5.0

Launched in June 2012

API

- IP block history lookup
- Support for lookup by Custom field

IPAM

- tag editor refresh
- ARIN Reallocation support (via REST)
- RIPE block lookup for aggregate importing (welcome screen only)

DNS

- Zone template editor/builder

- Zone editor refresh
- Zone importer refresh
- DNS Admin UI updates including pre/post commands per server
- DynECT integration (v1) for Dyn.com DNS customers

Platform

- Customer/Delegate Portal - version 1 (includes permissions for IPAM/DNS functions)
- Real-time streaming syslog function (incl JSON support)
- Admin Tab UI updates (multiple)
- Reporting option (filter by Tag, Custom Field, Region, etc.)
- Timeserver refresh

ProVision 3.4.0

Launched in April 2012

API

- DNS functionality now accessible
- Log functionality
- php SDK v1

IPAM

- IPv6 Direct Assign Refresh
- Prep for UI refresh

DNS

- Multi-server support for zone export/push
- DNS Admin UI refresh

Platform

- Authentication revamp (login, logging, etc.)
- Sitelist UI update
- Paging UI update
- Salesforce integration (version 1)
- Prep for RIR/LIR support

ProVision 3.3.0

Launched in March 2012

API

- IPAM functionality vis RESTful web service (including SWIP)
- Add/Remove Assignee (all fields)
- Add/Remove Assignee contacts (all fields)

IPAM

- View/Edit Devices from detail screen
- SWIP data lookup (ARIN only)

DNS

- SCP support for zone exports
- Secure64 Integration (version 1)

Peering

- Support for multiple locations within an Exchange

Platform

- Dashboard refresh
- Reporting Tab refresh
- Automated IP Import wizard
- Enhanced RADIUS VSA support
- RESTful API integration (application side)
- Multiple navigation updates (tab/subtab)
- FreeBSD Virtual Appliance released

ProVision 3.2.0

Launched in February 2012

IPAM

- RPI RPSL API integration
- SWIP as "private" for ARIN region
- Parent/Child relationships for Sites (multiple levels permitted)
- Revamped Site Editing screen
- Multiple Tag support for assignments
- Filter by NetMask option

DNS

- Zone editor enhancements (multiple)
- Bulk DNS zone editor
- DNS Zone download (ZIP format)
- DNS zone file incremter
- DNS Zone file pre-population

Peering

- Integrated Communication Manager unto UI

Asset and Device Tracking

- Rack level diagramming from Asset data
- Device tracking (IP, hostname, MAC address, VLAN, etc.)
- Generate DNS records from Device data

Platform

- Revamped global search interface
- Assistant (integrated network tools and search)
- Additional Filtering enhancements for views
- Additional Header sorting enhancements

- Double byte support
- User Management enhancements
- Integrated documentation (HTML/PDF)
- Revamped "tree view" functionality
- Granular logging with filtering and highlighting
- New "Reporting" Tab for statistics, data export and RIR reporting

ProVision 3.0.0

Launched in Fall 2011

IPAM

- ARIN API integration
- Enhanced block search functionality
- v4 network host scanning

DNS

- Zone searching and filtering enhancements
- Note storage and search per zone record entry

Asset and Device Tracking

- Create/Delete and Modify Assets (beta)

Platform

- Time Machine functionality for Zones, IP blocks and User Actions
- Local authentication engine
- Enhanced Global search functionality
- Enhanced Tree View functionality

ProVision 2.5

Launched in Summer 2011

IPAM

- Added support from /16 to /128 for IPv6
- Custom field per Block (accessed via Admin Tab)
- Custom field per Site (accessed via Admin Tab)
- Detail page retains IP settings for multiple allocations
- Added VLAN tracking per block
- Added customizable email templates to Admin Tab
- Direct Assign auto subnets for v4 and v6
- Track blocks by Region

DNS

- DNSSEC support including DS key generation
- Added paste in forward IPv6 address to resolve IPv6 PTR while editing reverse zones

Peering

- Production release (version 1)
- Support for Cisco, Juniper, Foundry, Brocade

Platform

- Data importer (beta) - including DNS zones
- Updated Admin Tab functionality and layout
- Added permissions checks for working files/directories
- php 5.3.6 compatibility
- browser compatibility updates (various)