

# **PlanetLab Central API Documentation**

**Aaron Klingaman**

**Mark Huang**

**PlanetLab Central API Documentation**  
by Aaron Klingaman and Mark Huang

# Table of Contents

<b>1. PlanetLab API Methods.....</b>	<b>1</b>
AddAddressType .....	1
AddAddressTypeToAddress .....	1
AddBootState .....	2
AddConfFile.....	2
AddConfFileToNodeGroup .....	4
AddConfFileToNode .....	4
AddKeyType .....	5
AddMessage .....	5
AddNetworkMethod .....	6
AddNetworkType .....	7
AddNodeGroup .....	7
AddNodeNetwork .....	8
AddNode.....	9
AddNodeToNodeGroup .....	10
AddNodeToPCU .....	11
AddPCU .....	12
AddPeer.....	12
AddPersonKey.....	13
AddPerson.....	14
AddPersonToSite.....	15
AddPersonToSlice .....	15
AddRole.....	16
AddRoleToPerson .....	17
AddSiteAddress .....	18
AddSite .....	18
AddSliceAttribute .....	19
AddSliceAttributeType.....	20
AddSliceInstantiation .....	21
AddSlice.....	22
AddSliceToNodes.....	22
AdmAddAddressType .....	23
AdmAddNodeGroup .....	24
AdmAddNodeNetwork.....	24
AdmAddNode.....	25
AdmAddNodeToNodeGroup .....	26
AdmAddPersonKey .....	27
AdmAddPerson .....	28
AdmAddPersonToSite.....	28
AdmAddSitePowerControlUnit .....	29
AdmAddSite .....	30
AdmAssociateNodeToPowerControlUnitPort .....	31
AdmAuthCheck .....	31
AdmDeleteAddressType.....	32
AdmDeleteAllPersonKeys .....	33
AdmDeleteNodeGroup .....	33
AdmDeleteNodeNetwork.....	34
AdmDeleteNode .....	35
AdmDeletePersonKeys .....	35
AdmDeletePerson .....	36
AdmDeleteSitePowerControlUnit.....	37
AdmDeleteSite .....	37
AdmDisassociatePowerControlUnitPort.....	38
AdmGenerateNodeConfFile.....	38
AdmGetAllAddressTypes.....	39
AdmGetAllKeyTypes.....	40
AdmGetAllNodeNetworks .....	40
AdmGetAllRoles .....	42

AdmGetNodeGroupNodes .....	42
AdmGetNodeGroups .....	43
AdmGetNodes .....	44
AdmGetPersonKeys .....	45
AdmGetPersonRoles.....	46
AdmGetPersonSites .....	47
AdmGetPersons.....	48
AdmGetPowerControlUnitNodes.....	49
AdmGetPowerControlUnits.....	49
AdmGetSiteNodes .....	50
AdmGetSitePersons .....	51
AdmGetSitePIs .....	52
AdmGetSitePowerControlUnits .....	52
AdmGetSites .....	53
AdmGetSiteTechContacts .....	55
AdmGrantRoleToPerson .....	56
AdmIsPersonInRole .....	56
AdmQueryConfFile .....	57
AdmQueryNode .....	58
AdmQueryPerson .....	58
AdmQueryPowerControlUnit .....	59
AdmQuerySite .....	60
AdmRebootNode .....	60
AdmRemoveNodeFromNodeGroup .....	61
AdmRemovePersonFromSite .....	62
AdmRevokeRoleFromPerson.....	63
AdmSetPersonEnabled.....	63
AdmSetPersonPrimarySite .....	64
AdmUpdateNodeGroup .....	65
AdmUpdateNodeNetwork.....	65
AdmUpdateNode.....	66
AdmUpdatePerson .....	67
AdmUpdateSitePowerControlUnit .....	68
AdmUpdateSite.....	69
AuthCheck .....	70
BlacklistKey.....	71
BootCheckAuthentication.....	71
BootGetNodeDetails .....	72
BootNotifyOwners .....	73
BootUpdateNode.....	74
DeleteAddress.....	75
DeleteAddressTypeFromAddress.....	75
DeleteAddressType.....	76
DeleteBootState.....	77
DeleteConfFileFromNodeGroup .....	77
DeleteConfFileFromNode .....	78
DeleteConfFile .....	79
DeleteKey .....	79
DeleteKeyType.....	80
DeleteMessage .....	80
DeleteNetworkMethod.....	81
DeleteNetworkType.....	81
DeleteNodeFromNodeGroup.....	82
DeleteNodeFromPCU.....	83
DeleteNodeGroup .....	84
DeleteNodeNetwork.....	84
DeleteNode.....	85
DeletePCU .....	85
DeletePeer.....	86
DeletePersonFromSite .....	87

DeletePersonFromSlice.....	87
DeletePerson.....	88
DeleteRoleFromPerson.....	89
DeleteRole.....	90
DeleteSession.....	90
DeleteSite.....	91
DeleteSliceAttribute.....	92
DeleteSliceAttributeType.....	92
DeleteSliceFromNodes.....	93
DeleteSliceInstantiation.....	94
DeleteSlice.....	94
GetAddresses.....	95
GetAddressTypes.....	96
GetBootStates.....	97
GetConfFiles.....	97
GetEvents.....	98
GetKeys.....	100
GetKeyTypes.....	100
GetMessages.....	101
GetNetworkMethods.....	102
GetNetworkTypes.....	102
GetNodeGroups.....	103
GetNodeNetworks.....	104
GetNodes.....	105
GetPCUs.....	106
GetPeerData.....	107
GetPeerName.....	109
GetPeers.....	109
GetPersons.....	110
GetRoles.....	112
GetSession.....	112
GetSites.....	113
GetSliceAttributes.....	114
GetSliceAttributeTypes.....	115
GetSliceInstantiations.....	116
GetSlices.....	117
GetSlivers.....	118
RebootNode.....	121
RefreshPeer.....	122
SetPersonPrimarySite.....	123
SliceCreate.....	124
SliceDelete.....	124
UpdateAddress.....	125
UpdateAddressType.....	126
UpdateConfFile.....	126
UpdateKey.....	127
UpdateMessage.....	128
UpdateNodeGroup.....	129
UpdateNodeNetwork.....	129
UpdateNode.....	131
UpdatePCU.....	132
UpdatePeer.....	132
UpdatePerson.....	133
UpdateSite.....	134
UpdateSliceAttribute.....	135
UpdateSliceAttributeType.....	136
UpdateSlice.....	137
system.listMethods.....	138
system.methodHelp.....	138
system.methodSignature.....	139

system.multicall.....139

# Chapter 1. PlanetLab API Methods

## AddAddressType

Status:

current

Prototype:

AddAddressType (auth, address\_type\_fields)

Description:

Adds a new address type. Fields specified in address\_type\_fields are used.

Returns the new address\_type\_id (> 0) if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
address_type_fields	struct	False	
name	string		Address type
description	string		Address type description

Returns:

Name	Type	Optional	Description
	int		New address_type_id (> 0) if successful

## AddAddressTypeToAddress

Status:

current

Prototype:

AddAddressTypeToAddress (auth, address\_type\_id\_or\_name, address\_id)

Description:

Adds an address type to the specified address.

PIs may only update addresses of their own sites.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
address_type_id_or_name	int	False	Address type identifier
ad- dress_type_id_or_name	string		Address type
address_id	int	False	Address identifier

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AddBootState

Status:

current

Prototype:

AddBootState (auth, name)

Description:

Adds a new node boot state.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
name	string	False	Boot state

Returns:

Name	Type	Optional	Description
	int		1 if successful



## AddConfFile

Status:

current

Prototype:

AddConfFile (auth, conf\_file\_fields)

Description:

Adds a new node configuration file. Any fields specified in conf\_file\_fields are used, otherwise defaults are used.

Returns the new conf\_file\_id (> 0) if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
conf_file_fields	struct	False	
file_owner	string		chown(1) owner
postinstall_cmd	string		Shell command to execute after installing
error_cmd	string		Shell command to execute if any error occurs
preinstall_cmd	string		Shell command to execute prior to installing
dest	string		Absolute path where file should be installed
ignore_cmd_errors	boolean		Install file anyway even if an error occurs
enabled	boolean		Configuration file is active
file_permissions	string		chmod(1) permissions
source	string		Relative path on the boot server where file can be downloaded
always_update	boolean		Always attempt to install file even if unchanged
file_group	string		chgrp(1) owner

Returns:

Name	Type	Optional	Description
	int		New conf_file_id (> 0) if successful

## AddConfFileToNodeGroup

Status:

current

Prototype:

AddConfFileToNodeGroup (auth, conf\_file\_id, nodegroup\_id\_or\_name)

Description:

Adds a configuration file to the specified node group. If the node group is already linked to the configuration file, no errors are returned.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
conf_file_id	int	False	Configuration file identifier
nodegroup_id_or_name	mixed	False	
node-group_id_or_name	int		Node group identifier
node-group_id_or_name	string		Node group name

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AddConfFileToNode

Status:

current

Prototype:

AddConfFileToNode (auth, conf\_file\_id, node\_id\_or\_hostname)

Description:

Adds a configuration file to the specified node. If the node is already linked to the configuration file, no errors are returned.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
conf_file_id	int	False	Configuration file identifier
node_id_or_hostname	mixed	False	
node_id_or_hostname	int		Node identifier
node_id_or_hostname	string		Fully qualified hostname

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AddKeyType

Status:

current

Prototype:

AddKeyType (auth, name)

Description:

Adds a new key type.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
name	string	False	Key type

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AddMessage

Status:

current

Prototype:

AddMessage (auth, message\_fields)

Description:

Adds a new message template. Any values specified in message\_fields are used, otherwise defaults are used.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
message_fields	struct	False	
enabled	boolean		Message is enabled
message_id	string		Message identifier
template	string		Message template
subject	string		Message summary

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AddNetworkMethod

Status:

current

Prototype:

AddNetworkMethod (auth, name)

Description:

Adds a new network method.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
name	string	False	Network method

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AddNetworkType

Status:

current

Prototype:

AddNetworkType (auth, name)

Description:

Adds a new network type.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
name	string	False	Network type

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AddNodeGroup

Status:

current

Prototype:

AddNodeGroup (auth, nodegroup\_fields)

Description:

Adds a new node group. Any values specified in nodegroup\_fields are used, otherwise defaults are used.

Returns the new nodegroup\_id (> 0) if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
nodegroup_fields	struct	False	
name	string		Node group name
description	string		Node group description

Returns:

Name	Type	Optional	Description
	int		New nodegroup_id (> 0) if successful

## AddNodeNetwork

Status:

current

Prototype:

AddNodeNetwork (auth, node\_id\_or\_hostname, nodenetwork\_fields)

Description:

Adds a new network for a node. Any values specified in nodenetwork\_fields are used, otherwise defaults are used. Acceptable values for method may be retrieved via GetNetworkMethods. Acceptable values for type may be retrieved via GetNetworkTypes.

If type is static, ip, gateway, network, broadcast, netmask, and dns1 must all be specified in nodenetwork\_fields. If type is dhcp, these parameters, even if specified, are ignored.

PIs and techs may only add networks to their own nodes. Admins may add networks to any node.

Returns the new nodenetwork\_id (> 0) if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_id_or_hostname	mixed	False	
node_id_or_hostname	int		Node identifier
node_id_or_hostname	string		Fully qualified hostname
nodenetwork_fields	struct	False	
network	string		Subnet address
is_primary	boolean		Is the primary interface for this node

Name	Type	Optional	Description
dns1	string		IP address of primary DNS server
hostname	string		(Optional) Hostname
mac	string		MAC address
bwlimit	int		Bandwidth limit
broadcast	string		Network broadcast address
method	string		Addressing method (e.g., 'static' or 'dhcp')
netmask	string		Subnet mask
dns2	string		IP address of secondary DNS server
ip	string		IP address
type	string		Address type (e.g., 'ipv4')
gateway	string		IP address of primary gateway

Returns:

Name	Type	Optional	Description
	int		New nodenetwork_id (> 0) if successful

## AddNode

Status:

current

Prototype:

AddNode (auth, site\_id\_or\_login\_base, node\_fields)

Description:

Adds a new node. Any values specified in node\_fields are used, otherwise defaults are used.

PIs and techs may only add nodes to their own sites. Admins may add nodes to any site.

Returns the new node\_id (> 0) if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure

Name	Type	Optional	Description
site_id_or_login_base	mixed	False	
site_id_or_login_base	int		Site identifier
site_id_or_login_base	string		Site slice prefix
node_fields	struct	False	
boot_state	string		Boot state
model	string		Make and model of the actual machine
version	string		Apparent Boot CD version
hostname	string		Fully qualified hostname

Returns:

Name	Type	Optional	Description
	int		New node_id (> 0) if successful

## AddNodeToNodeGroup

Status:

current

Prototype:

AddNodeToNodeGroup (auth, node\_id\_or\_hostname, nodegroup\_id\_or\_name)

Description:

Add a node to the specified node group. If the node is already a member of the node-group, no errors are returned.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_id_or_hostname	mixed	False	
node_id_or_hostname	int		Node identifier
node_id_or_hostname	string		Fully qualified hostname
nodegroup_id_or_name	mixed	False	



Name	Type	Optional	Description
node-group_id_or_name	int		Node group identifier
node-group_id_or_name	string		Node group name

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AddNodeToPCU

Status:

current

Prototype:

AddNodeToPCU (auth, node\_id\_or\_hostname, pcu\_id, port)

Description:

Adds a node to a port on a PCU. Faults if the node has already been added to the PCU or if the port is already in use.

Non-admins may only update PCUs at their sites.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_id_or_hostname	mixed	False	
node_id_or_hostname	int		Node identifier
node_id_or_hostname	string		Fully qualified hostname
pcu_id	int	False	PCU identifier
port	int	False	PCU port number

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AddPCU

Status:

current

Prototype:

AddPCU (auth, site\_id\_or\_login\_base, pcu\_fields)

Description:

Adds a new power control unit (PCU) to the specified site. Any fields specified in pcu\_fields are used, otherwise defaults are used.

PIs and technical contacts may only add PCUs to their own sites.

Returns the new pcu\_id (> 0) if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
site_id_or_login_base	mixed	False	
site_id_or_login_base	int		Site identifier
site_id_or_login_base	string		Site slice prefix
pcu_fields	struct	False	
username	string		PCU username
protocol	string		PCU protocol, e.g. ssh, https, telnet
ip	string		PCU IP address
notes	string		Miscellaneous notes
hostname	string		PCU hostname
model	string		PCU model string
password	string		PCU username

Returns:

Name	Type	Optional	Description
	int		New pcu_id (> 0) if successful

## AddPeer

Status:

current

Prototype:

AddPeer (auth, peer\_fields)

Description:

Adds a new peer.

Returns the new peer\_id (> 0) if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
peer_fields	struct	False	
key	string		Peer GPG public key
cacert	string		Peer SSL public certificate
peername	string		Peer name
peer_url	string		Peer API URL

Returns:

Name	Type	Optional	Description
	int		New peer_id (> 0) if successful

## AddPersonKey

Status:

current

Prototype:

AddPersonKey (auth, person\_id\_or\_email, key\_fields)

Description:

Adds a new key to the specified account.

Non-admins can only modify their own keys.

Returns the new key\_id (> 0) if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	
person_id_or_email	int		Account identifier

Name	Type	Optional	Description
person_id_or_email	string		Primary e-mail address
key_fields	struct	False	
key_type	string		Key type
peer_id	int		Peer at which this node is managed
key	string		Key value

Returns:

Name	Type	Optional	Description
	int		New key_id (> 0) if successful

## AddPerson

Status:

current

Prototype:

AddPerson (auth, person\_fields)

Description:

Adds a new account. Any fields specified in person\_fields are used, otherwise defaults are used.

Accounts are disabled by default. To enable an account, use SetPersonEnabled() or UpdatePerson().

Returns the new person\_id (> 0) if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_fields	struct	False	
bio	string		Biography
first_name	string		Given name
last_name	string		Surname
title	string		Title
url	string		Home page
phone	string		Telephone number
password	string		Account password in crypt() form
email	string		Primary e-mail address

Returns:

Name	Type	Optional	Description
	int		New person_id (> 0) if successful

## AddPersonToSite

Status:

current

Prototype:

AddPersonToSite (auth, person\_id\_or\_email, site\_id\_or\_login\_base)

Description:

Adds the specified person to the specified site. If the person is already a member of the site, no errors are returned. Does not change the person's primary site.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	
person_id_or_email	int		Account identifier
person_id_or_email	string		Primary e-mail address
site_id_or_login_base	mixed	False	
site_id_or_login_base	int		Site identifier
site_id_or_login_base	string		Site slice prefix

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AddPersonToSlice

Status:

current

Prototype:

AddPersonToSlice (auth, person\_id\_or\_email, slice\_id\_or\_name)

Description:

Adds the specified person to the specified slice. If the person is already a member of the slice, no errors are returned.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	
person_id_or_email	int		Account identifier
person_id_or_email	string		Primary e-mail address
slice_id_or_name	mixed	False	
slice_id_or_name	int		Slice identifier
slice_id_or_name	string		Slice name

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AddRole

Status:

current

Prototype:

AddRole (auth, role\_id, name)

Description:

Adds a new role.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
role_id	int	False	Role identifier
name	string	False	Role

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AddRoleToPerson

Status:

current

Prototype:

AddRoleToPerson (auth, role\_id\_or\_name, person\_id\_or\_email)

Description:

Grants the specified role to the person.

PIs can only grant the tech and user roles to users and techs at their sites. Admins can grant any role to any user.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
role_id_or_name	mixed	False	
role_id_or_name	int		Role identifier
role_id_or_name	string		Role
person_id_or_email	mixed	False	
per-son_id_or_email	int		Account identifier
per-son_id_or_email	string		Primary e-mail address

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AddSiteAddress

Status:

current

Prototype:

AddSiteAddress (auth, site\_id\_or\_login\_base, address\_fields)

Description:

Adds a new address to a site. Fields specified in address\_fields are used; some are not optional.

PIs may only add addresses to their own sites.

Returns the new address\_id (> 0) if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
site_id_or_login_base	mixed	False	
site_id_or_login_base	int		Site identifier
site_id_or_login_base	string		Site slice prefix
address_fields	struct	False	
city	string		City
country	string		Country
line3	string		Address line 3
line2	string		Address line 2
line1	string		Address line 1
state	string		State or province
postalcode	string		Postal code

Returns:

Name	Type	Optional	Description
	int		New address_id (> 0) if successful

## AddSite

Status:

current



Prototype:

AddSite (auth, site\_fields)

Description:

Adds a new site, and creates a node group for that site. Any fields specified in site\_fields are used, otherwise defaults are used.

Returns the new site\_id (> 0) if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
site_fields	struct	False	
name	string		Full site name
url	string		URL of a page that describes the site
longitude	double		Decimal longitude of the site
latitude	double		Decimal latitude of the site
max_slices	int		Maximum number of slices that the site is able to create
login_base	string		Site slice prefix
max_slivers	int		Maximum number of slivers that the site is able to create
is_public	boolean		Publicly viewable site
abbreviated_name	string		Abbreviated site name

Returns:

Name	Type	Optional	Description
	int		New site_id (> 0) if successful

## AddSliceAttribute

Status:

current

Prototype:

AddSliceAttribute (auth, slice\_id\_or\_name, attribute\_type\_id\_or\_name, value, node\_id\_or\_hostname)

Description:

Sets the specified attribute of the slice (or sliver, if `node_id_or_hostname` is specified) to the specified value.

Attributes may require the caller to have a particular role in order to be set or changed. Users may only set attributes of slices or slivers of which they are members. PIs may only set attributes of slices or slivers at their sites, or of which they are members. Admins may set attributes of any slice or sliver.

Returns the new `slice_attribute_id` (> 0) if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
<code>auth</code>	struct	False	API authentication structure
<code>slice_id_or_name</code>	mixed	False	
<code>slice_id_or_name</code>	int		Slice identifier
<code>slice_id_or_name</code>	string		Slice attribute type name
<code>attribute_type_id_or_name</code>	mixed	False	
<code>attribute_type_id_or_name</code>	int		Slice attribute type identifier
<code>attribute_type_id_or_name</code>	string		Slice attribute type name
<code>value</code>	string	False	Slice attribute value
<code>node_id_or_hostname</code>	mixed	True	
<code>node_id_or_hostname</code>	int		Node identifier
<code>node_id_or_hostname</code>	string		Fully qualified hostname

Returns:

Name	Type	Optional	Description
	int		New <code>slice_attribute_id</code> (> 0) if successful

## AddSliceAttributeType

Status:

current

Prototype:

AddSliceAttributeType (auth, attribute\_type\_fields)

Description:

Adds a new type of slice attribute. Any fields specified in `attribute_type_fields` are used, otherwise defaults are used.

Returns the new `attribute_type_id` (> 0) if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
<code>auth</code>	struct	False	API authentication structure
<code>attribute_type_fields</code>	struct	False	
<code>min_role_id</code>	int		Minimum (least powerful) role that can set or change this attribute
<code>name</code>	string		Slice attribute type name
<code>description</code>	string		Slice attribute type description

Returns:

Name	Type	Optional	Description
	int		New <code>attribute_id</code> (> 0) if successful

## AddSliceInstantiation

Status:

current

Prototype:

`AddSliceInstantiation (auth, name)`

Description:

Adds a new slice instantiation state.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
<code>auth</code>	struct	False	API authentication structure
<code>name</code>	string	False	Slice instantiation state

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AddSlice

Status:

current

Prototype:

AddSlice (auth, slice\_fields)

Description:

Adds a new slice. Any fields specified in slice\_fields are used, otherwise defaults are used.

Valid slice names are lowercase and begin with the login\_base (slice prefix) of a valid site, followed by a single underscore. Thereafter, only letters, numbers, or additional underscores may be used.

PIs may only add slices associated with their own sites (i.e., slice prefixes must always be the login\_base of one of their sites).

Returns the new slice\_id (> 0) if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
slice_fields	struct	False	
url	string		URL further describing this slice
max_nodes	int		Maximum number of nodes that can be assigned to this slice
instantiation	string		Slice instantiation state
name	string		Slice name
description	string		Slice description

Returns:

Name	Type	Optional	Description
	int		New slice_id (> 0) if successful

## AddSliceToNodes

Status:

current

Prototype:

AddSliceToNodes (auth, slice\_id\_or\_name, node\_id\_or\_hostname\_list)

Description:

Adds the specified slice to the specified nodes. Nodes can be either local or foreign nodes, as returned by GetNodes

If the slice is already associated with a node, no errors are returned.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
slice_id_or_name	mixed	False	
slice_id_or_name	int		Slice identifier
slice_id_or_name	string		Slice name
node_id_or_hostname_list	list	False	
	mixed		
	int		Node identifier
	string		Fully qualified hostname

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmAddAddressType

Status:

deprecated

Prototype:

AdmAddAddressType (auth, name)

Description:

Deprecated. See AddAddressType.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
name	string	False	Address type

Returns:

Name	Type	Optional	Description
	int		New address_type_id (> 0) if successful

## AdmAddNodeGroup

Status:

deprecated

Prototype:

AdmAddNodeGroup (auth, name, description)

Description:

Deprecated. See AddNodeGroup.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
name	string	False	Node group name
description	string	False	Node group description

Returns:

Name	Type	Optional	Description
	int		New nodegroup_id (> 0) if successful

## AdmAddNodeNetwork

Status:

deprecated

Prototype:

AdmAddNodeNetwork (auth, node\_id, method, type, nodenetwork\_fields)

Description:

Deprecated. See AddNodeNetwork.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_id	int	False	Node associated with this interface
method	string	False	Addressing method (e.g., 'static' or 'dhcp')
type	string	False	Address type (e.g., 'ipv4')
nodenetwork_fields	struct	True	
network	string		Subnet address
is_primary	boolean		Is the primary interface for this node
dns1	string		IP address of primary DNS server
hostname	string		(Optional) Hostname
mac	string		MAC address
bwlimit	int		Bandwidth limit
broadcast	string		Network broadcast address
netmask	string		Subnet mask
dns2	string		IP address of secondary DNS server
ip	string		IP address
gateway	string		IP address of primary gateway

Returns:

Name	Type	Optional	Description
	int		New nodenetwork_id (> 0) if successful

## AdmAddNode

Status:

deprecated

Prototype:

AdmAddNode (auth, site\_id, hostname, boot\_state, node\_fields)

Description:

Deprecated. See AddNode.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
site_id	int	False	Site identifier
hostname	string	False	Fully qualified hostname
boot_state	string	False	Boot state
node_fields	struct	True	
model	string		Make and model of the actual machine
version	string		Apparent Boot CD version

Returns:

Name	Type	Optional	Description
	int		New node_id (> 0) if successful

## AdmAddNodeToNodeGroup

Status:

deprecated

Prototype:

AdmAddNodeToNodeGroup (auth, node\_id\_or\_hostname, nodegroup\_id\_or\_name)

Description:

Deprecated. See AddNodeToNodeGroup.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure



Name	Type	Optional	Description
node_id_or_hostname	mixed	False	
node_id_or_hostname	int		Node identifier
node_id_or_hostname	string		Fully qualified hostname
nodegroup_id_or_name	mixed	False	
node-group_id_or_name	int		Node group identifier
node-group_id_or_name	string		Node group name

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmAddPersonKey

Status:

deprecated

Prototype:

AdmAddPersonKey (auth, person\_id\_or\_email, key\_type, key\_value, is\_primary)

Description:

Deprecated. See AddPersonKey. Keys can no longer be marked as primary, i.e. the is\_primary argument does nothing.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	
person_id_or_email	int		Account identifier
person_id_or_email	string		Primary e-mail address
key_type	string	False	Key type
key_value	string	False	Key value

Name	Type	Optional	Description
is_primary	int	False	Make this key the primary key

Returns:

Name	Type	Optional	Description
	int		New key_id (> 0) if successful

## AdmAddPerson

Status:

deprecated

Prototype:

AdmAddPerson (auth, first\_name, last\_name, person\_fields)

Description:

Deprecated. See AddPerson.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
first_name	string	False	Given name
last_name	string	False	Surname
person_fields	struct	True	
bio	string		Biography
title	string		Title
url	string		Home page
phone	string		Telephone number
password	string		Account password in crypt() form
email	string		Primary e-mail address

Returns:

Name	Type	Optional	Description
	int		New person_id (> 0) if successful

## AdmAddPersonToSite

Status:

deprecated

Prototype:

AdmAddPersonToSite (auth, person\_id\_or\_email, site\_id\_or\_login\_base)

Description:

Deprecated. See AddPersonToSite.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	
person_id_or_email	int		Account identifier
person_id_or_email	string		Primary e-mail address
site_id_or_login_base	mixed	False	
site_id_or_login_base	int		Site identifier
site_id_or_login_base	string		Site slice prefix

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmAddSitePowerControlUnit

Status:

deprecated

Prototype:

AdmAddSitePowerControlUnit (auth, site\_id\_or\_login\_base, pcu\_fields)

Description:

Deprecated. See AddPCU.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
site_id_or_login_base	mixed	False	
site_id_or_login_base	int		Site identifier
site_id_or_login_base	string		Site slice prefix
pcu_fields	struct	False	
username	string		PCU username
protocol	string		PCU protocol, e.g. ssh, https, telnet
ip	string		PCU IP address
notes	string		Miscellaneous notes
hostname	string		PCU hostname
model	string		PCU model string
password	string		PCU username

Returns:

Name	Type	Optional	Description
	int		New pcu_id (> 0) if successful

## AdmAddSite

Status:

deprecated

Prototype:

AdmAddSite (auth, name, abbreviated\_name, login\_base, site\_fields)

Description:

Deprecated. See AddSite.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
name	string	False	Full site name
abbreviated_name	string	False	Abbreviated site name
login_base	string	False	Site slice prefix
site_fields	struct	True	

Name	Type	Optional	Description
url	string		URL of a page that describes the site
latitude	double		Decimal latitude of the site
longitude	double		Decimal longitude of the site
is_public	boolean		Publicly viewable site

Returns:

Name	Type	Optional	Description
	int		New site_id (> 0) if successful

## AdmAssociateNodeToPowerControlUnitPort

Status:

deprecated

Prototype:

AdmAssociateNodeToPowerControlUnitPort (auth, pcu\_id, port, node\_id\_or\_hostname)

Description:

Deprecated. See AddNodeToPCU.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
pcu_id	int	False	PCU identifier
port	int	False	PCU port number
node_id_or_hostname	mixed	False	
node_id_or_hostname	int		Node identifier
node_id_or_hostname	string		Fully qualified hostname

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmAuthCheck

Status:

deprecated

Prototype:

AdmAuthCheck (auth)

Description:

Deprecated. See AuthCheck.

Parameters:

Name	Type	Optional	Description
auth	mixed	False	
auth	struct		API authentication structure
auth	struct		API authentication structure

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmDeleteAddressType

Status:

deprecated

Prototype:

AdmDeleteAddressType (auth, address\_type\_id\_or\_name)

Description:

Deprecated. See DeleteAddressType.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
address_type_id_or_name	mixed	False	
address_type_id_or_name	int		Address type identifier
address_type_id_or_name	string		Address type

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmDeleteAllPersonKeys

Status:

deprecated

Prototype:

AdmDeleteAllPersonKeys (auth, person\_id\_or\_email)

Description:

Deprecated. Functionality can be implemented with GetPersons and DeleteKey.

Deletes all of the keys associated with an account. Non-admins may only delete their own keys.

Non-admins may only delete their own keys.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	
person_id_or_email	int		Account identifier
person_id_or_email	string		Primary e-mail address

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmDeleteNodeGroup

Status:

deprecated

Prototype:

AdmDeleteNodeGroup (auth, node\_group\_id\_or\_name)

Description:

Deprecated. See DeleteNodeGroup.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_group_id_or_name	mixed	False	
node_group_id_or_name	int		Node group identifier
node_group_id_or_name	string		Node group name

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmDeleteNodeNetwork

Status:

deprecated

Prototype:

AdmDeleteNodeNetwork (auth, node\_id\_or\_hostname, nodenetwork\_id)

Description:

Deprecated. See DeleteNodeNetwork.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_id_or_hostname	mixed	False	
node_id_or_hostname	int		Node identifier
node_id_or_hostname	string		Fully qualified hostname
nodenetwork_id	int	False	Node interface identifier



Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmDeleteNode

Status:

deprecated

Prototype:

AdmDeleteNode (auth, node\_id\_or\_hostname)

Description:

Deprecated. See DeleteNode.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_id_or_hostname	mixed	False	
node_id_or_hostname	int		Node identifier
node_id_or_hostname	string		Fully qualified hostname

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmDeletePersonKeys

Status:

deprecated

Prototype:

AdmDeletePersonKeys (auth, person\_id\_or\_email, key\_ids)

Description:

Deprecated. Functionality can be implemented with GetPersons and DeleteKey.

Deletes the specified keys. Non-admins may only delete their own keys.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	
person_id_or_email	int		Account identifier
person_id_or_email	string		Primary e-mail address
key_ids	array int	False	Key identifier

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmDeletePerson

Status:

deprecated

Prototype:

AdmDeletePerson (auth, person\_id\_or\_email)

Description:

Deprecated. See DeletePerson.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	
person_id_or_email	int		Account identifier
person_id_or_email	string		Primary e-mail address

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmDeleteSitePowerControlUnit

Status:

deprecated

Prototype:

AdmDeleteSitePowerControlUnit (auth, pcu\_id)

Description:

Deprecated. See DeletePCU.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
pcu_id	int	False	PCU identifier

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmDeleteSite

Status:

deprecated

Prototype:

AdmDeleteSite (auth, site\_id\_or\_login\_base)

Description:

Deprecated. See DeleteSite.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure

Name	Type	Optional	Description
site_id_or_login_base	mixed	False	
site_id_or_login_base	int		Site identifier
site_id_or_login_base	string		Site slice prefix

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmDisassociatePowerControlUnitPort

Status:

deprecated

Prototype:

AdmDisassociatePowerControlUnitPort (auth, pcu\_id, port)

Description:

Deprecated. See DeleteNodeFromPCU.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
pcu_id	int	False	PCU identifier
port	int	False	PCU port number

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmGenerateNodeConfFile

Status:

deprecated

Prototype:

AdmGenerateNodeConfFile (auth, node\_id\_or\_hostname)

Description:

Deprecated. Functionality can be implemented with GetNodes, GetNodeNetworks, and UpdateNode.

Creates a new node configuration file if all network settings are present. This function will generate a new node key for the specified node, effectively invalidating any old configuration files.

Non-admins can only generate files for nodes at their sites.

Returns the contents of the file if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_id_or_hostname	mixed	False	
node_id_or_hostname	int		Node identifier
node_id_or_hostname	string		Fully qualified hostname

Returns:

Name	Type	Optional	Description
	string		Node configuration file

## AdmGetAllAddressTypes

Status:

deprecated

Prototype:

AdmGetAllAddressTypes (auth, address\_type\_filter, return\_fields)

Description:

Deprecated. See GetAddressTypes.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
address_type_filter	mixed	True	
address_type_filter	array		
	mixed		

Name	Type	Optional	Description
	int		Address type identifier
	string		Address type
address_type_filter	struct		Attribute filter
return_fields	array	True	List of fields to return

Returns:

Name	Type	Optional	Description
	array		
	struct		
name	string		Address type
address_type_id	int		Address type identifier
description	string		Address type description

## AdmGetAllKeyTypes

Status:

deprecated

Prototype:

AdmGetAllKeyTypes (auth)

Description:

Deprecated. See GetKeyTypes.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure

Returns:

Name	Type	Optional	Description
	array		
	string		Key type

## AdmGetAllNodeNetworks

Status:

deprecated

Prototype:

AdmGetAllNodeNetworks (auth, node\_id\_or\_hostname)

Description:

Deprecated. Functionality can be implemented with GetNodes and GetNodeNetworks.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_id_or_hostname	mixed	False	
node_id_or_hostname	int		Node identifier
node_id_or_hostname	string		Fully qualified hostname

Returns:

Name	Type	Optional	Description
	array		
	struct		
broadcast	string		Network broadcast address
is_primary	boolean		Is the primary interface for this node
network	string		Subnet address
ip	string		IP address
dns1	string		IP address of primary DNS server
hostname	string		(Optional) Hostname
netmask	string		Subnet mask
gateway	string		IP address of primary gateway
nodenetwork_id	int		Node interface identifier
mac	string		MAC address
node_id	int		Node associated with this interface
dns2	string		IP address of secondary DNS server

Name	Type	Optional	Description
bwlimit	int		Bandwidth limit
type	string		Address type (e.g., 'ipv4')
method	string		Addressing method (e.g., 'static' or 'dhcp')

## AdmGetAllRoles

Status:

deprecated

Prototype:

AdmGetAllRoles (auth)

Description:

Deprecated. See GetRoles.

Return all possible roles as a struct:

```
{'10': 'admin', '20': 'pi', '30': 'user', '40': 'tech'}
```

Note that because of XML-RPC marshalling limitations, the keys to this struct are string representations of the integer role identifiers.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure

Returns:

Name	Type	Optional	Description
	struct		

## AdmGetNodeGroupNodes

Status:

deprecated

Prototype:

AdmGetNodeGroupNodes (auth, nodegroup\_id\_or\_name)

Description:

Deprecated. See GetNodeGroups.

Returns a list of node\_ids for the node group specified.



Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
nodegroup_id_or_name	mixed	False	
node-group_id_or_name	int		Node group identifier
node-group_id_or_name	string		Node group name

Returns:

Name	Type	Optional	Description
	array		List of nodes in this node group

## AdmGetNodeGroups

Status:

deprecated

Prototype:

AdmGetNodeGroups (auth, nodegroup\_filter, return\_fields)

Description:

Deprecated. See GetNodeGroups.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
nodegroup_filter	mixed	True	
nodegroup_filter	array		
	mixed		
	int		Node group identifier
	string		Node group name
nodegroup_filter	struct		Attribute filter
return_fields	array	True	List of fields to return

Returns:

Name	Type	Optional	Description
	array		
	struct		
node_ids	array		List of nodes in this node group
nodegroup_id	int		Node group identifier
name	string		Node group name
conf_file_ids	array		List of configuration files specific to this node group
description	string		Node group description

## AdmGetNodes

Status:

deprecated

Prototype:

AdmGetNodes (auth, node\_id\_or\_hostname\_list, return\_fields)

Description:

Deprecated. See GetNodes. All fields are now always returned.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_id_or_hostname_list	list	True	
node_id_or_hostname_list	list		
	mixed		
	int		Node identifier
	string		Fully qualified hostname
node_id_or_hostname_list	list		Attribute filter
return_fields	array	True	List of fields to return

Returns:

Name	Type	Optional	Description
	array		
	struct		

Name	Type	Optional	Description
session	string		(Admin only) Node session value
pcu_ids	array		List of PCUs that control this node
nodegroup_ids	array		List of node groups that this node is in
last_updated	int		Date and time when node entry was created
version	string		Apparent Boot CD version
nodenetwork_ids	array		List of network interfaces that this node has
boot_state	string		Boot state
hostname	string		Fully qualified hostname
site_id	int		Site at which this node is located
ports	array		List of PCU ports that this node is connected to
slice_ids	array		List of slices on this node
boot_nonce	string		(Admin only) Random value generated by the node at last boot
node_id	int		Node identifier
key	string		(Admin only) Node key
date_created	int		Date and time when node entry was created
model	string		Make and model of the actual machine
peer_id	int		Peer at which this node is managed
conf_file_ids	array		List of configuration files specific to this node
ssh_rsa_key	string		Last known SSH host key

## AdmGetPersonKeys

Status:

deprecated

Prototype:

AdmGetPersonKeys (person\_id\_or\_email)

Description:

Deprecated. Functionality can be implemented with GetPersons and GetKeys.

Parameters:

Name	Type	Optional	Description
person_id_or_email	struct	False	API authentication structure

Returns:

Name	Type	Optional	Description
	array		
	struct		
key_type	string		Key type
key_id	int		Key identifier
peer_id	int		Peer at which this node is managed
key	string		Key value

## AdmGetPersonRoles

Status:

deprecated

Prototype:

AdmGetPersonRoles (auth, person\_id\_or\_email)

Description:

Deprecated. See GetPersons.

Return the roles that the specified person has as a struct:

```
{'10': 'admin', '30': 'user', '20': 'pi', '40': 'tech'}
```

Admins can get the roles for any user. PIs can only get the roles for members of their sites. All others may only get their own roles.

Note that because of XML-RPC marshalling limitations, the keys to this struct are string representations of the integer role identifiers.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	

Name	Type	Optional	Description
person_id_or_email	int		Account identifier
person_id_or_email	string		Primary e-mail address

Returns:

Name	Type	Optional	Description
	struct		

## AdmGetPersonSites

Status:

deprecated

Prototype:

AdmGetPersonSites (auth, person\_id\_or\_email)

Description:

Deprecated. See GetPersons.

Returns the sites that the specified person is associated with as an array of site identifiers.

Admins may retrieve details about anyone. Users and techs may only retrieve details about themselves. PIs may retrieve details about themselves and others at their sites.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	
person_id_or_email	int		Account identifier
person_id_or_email	string		Primary e-mail address

Returns:

Name	Type	Optional	Description
	array		List of site identifiers

## AdmGetPersons

Status:

deprecated

Prototype:

AdmGetPersons (auth, person\_id\_or\_email\_list, return\_fields)

Description:

Deprecated. See GetPersons.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email_list	mixed	True	
per-son_id_or_email_list	array		
	mixed		
	int		Account identifier
	string		Primary e-mail address
per-son_id_or_email_list	struct		Attribute filter
return_fields	array	True	List of fields to return

Returns:

Name	Type	Optional	Description
	array		
	struct		
bio	string		Biography
first_name	string		Given name
last_name	string		Surname
last_updated	int		Date and time of last update
roles	array		List of roles
title	string		Title
url	string		Home page
key_ids	array		List of key identifiers
enabled	boolean		Has been enabled
slice_ids	array		List of slice identifiers
phone	string		Telephone number

Name	Type	Optional	Description
role_ids	array		List of role identifiers
person_id	int		Account identifier
date_created	int		Date and time when account was created
site_ids	array		List of site identifiers
peer_id	int		Peer at which this slice was created
email	string		Primary e-mail address

## AdmGetPowerControlUnitNodes

Status:

deprecated

Prototype:

AdmGetPowerControlUnitNodes (auth, pcu\_id)

Description:

Deprecated. See GetPCUs.

Returns a list of the nodes, and the ports they are assigned to, on the specified PCU.

Admin may query all PCUs. Non-admins may only query the PCUs at their sites.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
pcu_id	int	False	PCU identifier

Returns:

Name	Type	Optional	Description
	array		
	struct		
node_id	int		Node identifier
port_number	int		Port number

## AdmGetPowerControlUnits

Status:

deprecated

Prototype:

AdmGetPowerControlUnits (auth, pcu\_filter, return\_fields)

Description:

Deprecated. See GetPCUs.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
pcu_filter	mixed	True	
pcu_filter	array		
	int		PCU identifier
pcu_filter	struct		Attribute filter
return_fields	array	True	List of fields to return

Returns:

Name	Type	Optional	Description
	array		
	struct		
username	string		PCU username
protocol	string		PCU protocol, e.g. ssh, https, telnet
node_ids	array		List of nodes that this PCU controls
ip	string		PCU IP address
pcu_id	int		PCU identifier
hostname	string		PCU hostname
site_id	int		Identifier of site where PCU is located
ports	array		List of the port numbers that each node is connected to
model	string		PCU model string
password	string		PCU username
notes	string		Miscellaneous notes



## AdmGetSiteNodes

Status:

deprecated

Prototype:

AdmGetSiteNodes (auth, site\_id\_or\_name\_list)

Description:

Deprecated. See GetSites.

Return a struct containing an array of node\_ids for each of the sites specified. Note that the keys of the struct are strings, not integers, because of XML-RPC marshalling limitations.

Admins may retrieve details about all nodes on a site by not specifying site\_id\_or\_name or by specifying an empty list. Users and techs may only retrieve details about themselves. PIs may retrieve details about themselves and others at their sites.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
site_id_or_name_list	array	True	
	mixed		
	int		Site identifier
	string		Full site name

Returns:

Name	Type	Optional	Description
	struct		

## AdmGetSitePersons

Status:

deprecated

Prototype:

AdmGetSitePersons (auth, site\_id\_or\_login\_base)

Description:

Deprecated. See GetSites.

Return a list of person\_ids for the site specified.

PIs may only retrieve the person\_ids of accounts at their site. Admins may retrieve the person\_ids of accounts at any site.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
site_id_or_login_base	mixed	False	
site_id_or_login_base	int		Site identifier
site_id_or_login_base	string		Site slice prefix

Returns:

Name	Type	Optional	Description
	array		List of account identifiers

## AdmGetSitePIs

Status:

deprecated

Prototype:

AdmGetSitePIs (auth, site\_id\_or\_login\_base)

Description:

Deprecated. Functionality can be implemented with GetSites and GetPersons.

Return a list of person\_ids of the PIs for the site specified.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
site_id_or_login_base	mixed	False	
site_id_or_login_base	int		Site identifier
site_id_or_login_base	string		Site slice prefix

Returns:

Name	Type	Optional	Description
	array		List of account identifiers

## AdmGetSitePowerControlUnits

Status:

deprecated

Prototype:

AdmGetSitePowerControlUnits (auth, site\_id\_or\_login\_base)

Description:

Deprecated. Functionality can be implemented with GetSites and GetPCUs.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
site_id_or_login_base	mixed	False	
site_id_or_login_base	int		Site identifier
site_id_or_login_base	string		Site slice prefix

Returns:

Name	Type	Optional	Description
	array		
	struct		
username	string		PCU username
protocol	string		PCU protocol, e.g. ssh, https, telnet
node_ids	array		List of nodes that this PCU controls
ip	string		PCU IP address
pcu_id	int		PCU identifier
hostname	string		PCU hostname
site_id	int		Identifier of site where PCU is located
ports	array		List of the port numbers that each node is connected to
model	string		PCU model string
password	string		PCU password
notes	string		Miscellaneous notes

## AdmGetSites

Status:

deprecated

Prototype:

AdmGetSites (auth, site\_id\_or\_login\_base\_list, return\_fields)

Description:

Deprecated. See GetSites.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
site_id_or_login_base_list	list	True	
site_id_or_login_base_attr	array		
	mixed		
	int		Site identifier
	string		Site slice prefix
site_id_or_login_base_attr	list		Attribute filter
return_fields	array	True	List of fields to return

Returns:

Name	Type	Optional	Description
	array		
	struct		
address_ids	array		List of address identifiers
pcu_ids	array		List of PCU identifiers
last_updated	int		Date and time when site entry was last updated, in seconds since UNIX epoch
name	string		Full site name
node_ids	array		List of site node identifiers
url	string		URL of a page that describes the site
person_ids	array		List of account identifiers
site_id	int		Site identifier
longitude	double		Decimal longitude of the site

Name	Type	Optional	Description
slice_ids	array		List of slice identifiers
max_slivers	int		Maximum number of slivers that the site is able to create
max_slices	int		Maximum number of slices that the site is able to create
login_base	string		Site slice prefix
date_created	int		Date and time when site entry was created, in seconds since UNIX epoch
latitude	double		Decimal latitude of the site
is_public	boolean		Publicly viewable site
peer_id	int		Peer at which this slice was created
abbreviated_name	string		Abbreviated site name

## AdmGetSiteTechContacts

Status:

deprecated

Prototype:

AdmGetSiteTechContacts (auth, site\_id\_or\_login\_base)

Description:

Deprecated. Functionality can be implemented with GetSites and GetPersons.

Return a list of person\_ids of the technical contacts for the site specified.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
site_id_or_login_base	mixed	False	
site_id_or_login_base	int		Site identifier
site_id_or_login_base	string		Site slice prefix

Returns:

Name	Type	Optional	Description
	array		List of account identifiers

## AdmGrantRoleToPerson

Status:

deprecated

Prototype:

AdmGrantRoleToPerson (auth, person\_id\_or\_email, role\_id\_or\_name)

Description:

Deprecated. See AddRoleToPerson.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	
person_id_or_email	int		Role identifier
person_id_or_email	string		Role
role_id_or_name	mixed	False	
role_id_or_name	int		Account identifier
role_id_or_name	string		Primary e-mail address

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmIsPersonInRole

Status:

deprecated

Prototype:

AdmIsPersonInRole (auth, person\_id\_or\_email, role\_id\_or\_name)

**Description:**

Deprecated. Functionality can be implemented with GetPersons.

Returns 1 if the specified account has the specified role, 0 otherwise. This function differs from AdmGetPersonRoles() in that any authorized user can call it. It is currently restricted to verifying PI roles.

**Parameters:**

<b>Name</b>	<b>Type</b>	<b>Optional</b>	<b>Description</b>
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	
person_id_or_email	int		Account identifier
person_id_or_email	string		Primary e-mail address
role_id_or_name	mixed	False	
role_id_or_name	int		Role identifier
role_id_or_name	string		Role name

**Returns:**

<b>Name</b>	<b>Type</b>	<b>Optional</b>	<b>Description</b>
	int		1 if account has role, 0 otherwise

## AdmQueryConfFile

**Status:**

deprecated

**Prototype:**

AdmQueryConfFile (auth, search\_vals)

**Description:**

Deprecated. See GetConfFiles.

**Parameters:**

<b>Name</b>	<b>Type</b>	<b>Optional</b>	<b>Description</b>
auth	struct	False	API authentication structure
search_vals	struct	False	
node_id	int		Node identifier

Returns:

<b>Name</b>	<b>Type</b>	<b>Optional</b>	<b>Description</b>
	array		
	int		Configuration file identifier

## AdmQueryNode

Status:

deprecated

Prototype:

AdmQueryNode (auth, search\_vals)

Description:

Deprecated. Functionality can be implemented with GetNodes and GetNodeNetworks.

Parameters:

<b>Name</b>	<b>Type</b>	<b>Optional</b>	<b>Description</b>
auth	struct	False	API authentication structure
search_vals	struct	False	
nodenet-work_method	string		Addressing method (e.g., 'static' or 'dhcp')
nodenetwork_mac	string		MAC address
nodenetwork_ip	string		IP address
node_hostname	string		Fully qualified hostname

Returns:

<b>Name</b>	<b>Type</b>	<b>Optional</b>	<b>Description</b>
	array		
	int		Node identifier

## AdmQueryPerson

Status:

deprecated



Prototype:

`AdmQueryPerson (auth, search_vals)`

Description:

Deprecated. See `GetPersons`.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
search_vals	struct	False	
email	string		Primary e-mail address

Returns:

Name	Type	Optional	Description
	array		
	int		Account identifier

## AdmQueryPowerControlUnit

Status:

deprecated

Prototype:

`AdmQueryPowerControlUnit (auth, search_vals)`

Description:

Deprecated. Functionality can be implemented with `GetPCUs` or `GetNodes`.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
search_vals	struct	False	
node_id	int		Node identifier
pcu_hostname	string		PCU hostname
node_hostname	string		Fully qualified hostname
pcu_ip	string		PCU IP address

Returns:

Name	Type	Optional	Description
	array		
	int		PCU identifier

## AdmQuerySite

Status:

deprecated

Prototype:

AdmQuerySite (auth, search\_vals)

Description:

Deprecated. Functionality can be implemented with GetSites and GetNodes.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
search_vals	struct	False	
nodenetwork_mac	string		MAC address
nodenetwork_ip	string		IP address
site_abbreviatedname	string		Abbreviated site name
site_loginbase	string		Site slice prefix
node_id	int		Node identifier
node_hostname	string		Fully qualified hostname
site_name	string		Full site name

Returns:

Name	Type	Optional	Description
	array		
	int		Site identifier

## AdmRebootNode

Status:

deprecated

Prototype:

AdmRebootNode (auth, node\_id\_or\_hostname)

Description:

Deprecated. See RebootNode.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_id_or_hostname	mixed	False	
node_id_or_hostname	int		Node identifier
node_id_or_hostname	string		Fully qualified hostname

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmRemoveNodeFromNodeGroup

Status:

deprecated

Prototype:

AdmRemoveNodeFromNodeGroup (auth, node\_id\_or\_hostname, nodegroup\_id\_or\_name)

Description:

Deprecated. See DeleteNodeFromNodeGroup.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_id_or_hostname	mixed	False	
node_id_or_hostname	int		Node identifier
node_id_or_hostname	string		Fully qualified hostname
nodegroup_id_or_name	mixed	False	

Name	Type	Optional	Description
node-group_id_or_name	int		Node group identifier
node-group_id_or_name	string		Node group name

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmRemovePersonFromSite

Status:

deprecated

Prototype:

AdmRemovePersonFromSite (auth, person\_id\_or\_email, site\_id\_or\_login\_base)

Description:

Deprecated. See DeletePersonFromSite.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	
person_id_or_email	int		Account identifier
person_id_or_email	string		Primary e-mail address
site_id_or_login_base	mixed	False	
site_id_or_login_base	int		Site identifier
site_id_or_login_base	string		Site slice prefix

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmRevokeRoleFromPerson

Status:

deprecated

Prototype:

AdmRevokeRoleFromPerson (auth, person\_id\_or\_email, role\_id\_or\_name)

Description:

Deprecated. See DeleteRoleFromPerson.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	
person_id_or_email	int		Role identifier
person_id_or_email	string		Role
role_id_or_name	mixed	False	
role_id_or_name	int		Account identifier
role_id_or_name	string		Primary e-mail address

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmSetPersonEnabled

Status:

deprecated

Prototype:

AdmSetPersonEnabled (auth, person\_id\_or\_email, enabled)

Description:

Deprecated. See UpdatePerson.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	
person_id_or_email	int		Account identifier
person_id_or_email	string		Primary e-mail address
enabled	boolean	False	Has been enabled

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmSetPersonPrimarySite

Status:

deprecated

Prototype:

AdmSetPersonPrimarySite (auth, person\_id\_or\_email, site\_id\_or\_login\_base)

Description:

Deprecated. See SetPersonPrimarySite.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	
person_id_or_email	int		Account identifier
person_id_or_email	string		Primary e-mail address
site_id_or_login_base	mixed	False	
site_id_or_login_base	int		Site identifier
site_id_or_login_base	string		Site slice prefix

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmUpdateNodeGroup

Status:

deprecated

Prototype:

AdmUpdateNodeGroup (auth, nodegroup\_id\_or\_name, name, description)

Description:

Deprecated. See UpdateNodeGroup.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
nodegroup_id_or_name	mixed	False	
node-group_id_or_name	int		Node group identifier
node-group_id_or_name	string		Node group name
name	string	False	Node group name
description	string	False	Node group description

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmUpdateNodeNetwork

Status:

deprecated

Prototype:

AdmUpdateNodeNetwork (auth, nodenetwork\_id, nodenetwork\_fields)

Description:

Deprecated. See UpdateNodeNetwork.

Parameters:

<b>Name</b>	<b>Type</b>	<b>Optional</b>	<b>Description</b>
auth	struct	False	API authentication structure
nodenetwork_id	int	False	Node interface identifier
nodenetwork_fields	struct	False	
network	string		Subnet address
is_primary	boolean		Is the primary interface for this node
dns1	string		IP address of primary DNS server
hostname	string		(Optional) Hostname
mac	string		MAC address
bwlimit	int		Bandwidth limit
broadcast	string		Network broadcast address
method	string		Addressing method (e.g., 'static' or 'dhcp')
netmask	string		Subnet mask
node_id	int		Node associated with this interface
dns2	string		IP address of secondary DNS server
ip	string		IP address
type	string		Address type (e.g., 'ipv4')
gateway	string		IP address of primary gateway

Returns:

<b>Name</b>	<b>Type</b>	<b>Optional</b>	<b>Description</b>
	int		1 if successful



## AdmUpdateNode

Status:

deprecated

Prototype:

AdmUpdateNode (auth, node\_id\_or\_hostname, node\_fields)

Description:

Deprecated. See UpdateNode.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_id_or_hostname	mixed	False	
node_id_or_hostname	int		Node identifier
node_id_or_hostname	string		Fully qualified hostname
node_fields	struct	False	
version	string		Apparent Boot CD version
boot_state	string		Boot state
hostname	string		Fully qualified hostname
boot_nonce	string		(Admin only) Random value generated by the node at last boot
session	string		(Admin only) Node session value
key	string		(Admin only) Node key
model	string		Make and model of the actual machine

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmUpdatePerson

Status:

deprecated

Prototype:

AdmUpdatePerson (auth, person\_id\_or\_email, person\_fields)

Description:

Deprecated. See UpdatePerson.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	
person_id_or_email	int		Account identifier
person_id_or_email	string		Primary e-mail address
person_fields	struct	False	
bio	string		Biography
first_name	string		Given name
last_name	string		Surname
title	string		Title
url	string		Home page
enabled	boolean		Has been enabled
phone	string		Telephone number
password	string		Account password in crypt() form
email	string		Primary e-mail address

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmUpdateSitePowerControlUnit

Status:

deprecated

Prototype:

AdmUpdateSitePowerControlUnit (auth, pcu\_id, pcu\_fields)

Description:

Deprecated. See UpdatePCU.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
pcu_id	int	False	PCU identifier
pcu_fields	struct	False	
username	string		PCU username
protocol	string		PCU protocol, e.g. ssh, https, telnet
node_ids	array		List of nodes that this PCU controls
ip	string		PCU IP address
notes	string		Miscellaneous notes
hostname	string		PCU hostname
model	string		PCU model string
password	string		PCU password
ports	array		List of the port numbers that each node is connected to

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AdmUpdateSite

Status:

deprecated

Prototype:

AdmUpdateSite (auth, site\_id\_or\_login\_base, site\_fields)

Description:

Deprecated. See UpdateSite.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
site_id_or_login_base	mixed	False	

Name	Type	Optional	Description
site_id_or_login_base	int		Site identifier
site_id_or_login_base	string		Site slice prefix
site_fields	struct	False	
name	string		Full site name
url	string		URL of a page that describes the site
longitude	double		Decimal longitude of the site
latitude	double		Decimal latitude of the site
max_slices	int		Maximum number of slices that the site is able to create
login_base	string		Site slice prefix
max_slivers	int		Maximum number of slivers that the site is able to create
is_public	boolean		Publicly viewable site
abbreviated_name	string		Abbreviated site name

Returns:

Name	Type	Optional	Description
	int		1 if successful

## AuthCheck

Status:

current

Prototype:

AuthCheck (auth)

Description:

Returns 1 if the user or node authenticated successfully, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	mixed	False	
auth	struct		API authentication structure

Name	Type	Optional	Description
auth	struct		API authentication structure

Returns:

Name	Type	Optional	Description
	int		1 if successful

## BlacklistKey

Status:

current

Prototype:

BlacklistKey (auth, key\_id)

Description:

Blacklists a key, disassociating it and all others identical to it from all accounts and preventing it from ever being added again.

WARNING: Identical keys associated with other accounts with also be blacklisted.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
key_id	int	False	Key identifier

Returns:

Name	Type	Optional	Description
	int		1 if successful

## BootCheckAuthentication

Status:

deprecated

Prototype:

BootCheckAuthentication (auth)

Description:

Deprecated. See AuthCheck.

Parameters:

Name	Type	Optional	Description
auth	mixed	False	
auth	struct		API authentication structure
auth	struct		API authentication structure

Returns:

Name	Type	Optional	Description
	int		1 if successful

## BootGetNodeDetails

Status:

current

Prototype:

BootGetNodeDetails (auth)

Description:

Returns a set of details about the calling node, including a new node session value.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure

Returns:

Name	Type	Optional	Description
	struct		
boot_state	string		Boot state
model	string		Make and model of the actual machine
hostname	string		Fully qualified hostname
networks	array		
	struct		
broadcast	string		Network broadcast address

Name	Type	Optional	Description
is_primary	boolean		Is the primary interface for this node
network	string		Subnet address
ip	string		IP address
dns1	string		IP address of primary DNS server
hostname	string		(Optional) Hostname
netmask	string		Subnet mask
gateway	string		IP address of primary gateway
nodenetwork_id	int		Node interface identifier
mac	string		MAC address
node_id	int		Node associated with this interface
dns2	string		IP address of secondary DNS server
bwlimit	int		Bandwidth limit
type	string		Address type (e.g., 'ipv4')
method	string		Addressing method (e.g., 'static' or 'dhcp')
session	string		Session key

## BootNotifyOwners

Status:

current

Prototype:

BootNotifyOwners (auth, message\_id, include\_pis, include\_techs, include\_support)

Description:

Notify the owners of the node, and/or support about an event that happened on the machine.

Returns 1 if successful.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
message_id	string	False	Message identifier
include_pis	int	False	Notify PIs

Name	Type	Optional	Description
include_techs	int	False	Notify technical contacts
include_support	int	False	Notify support

Returns:

Name	Type	Optional	Description
	int		1 if successful

## BootUpdateNode

Status:

current

Prototype:

BootUpdateNode (auth, node\_fields)

Description:

Allows the calling node to update its own record. Only the primary network can be updated, and the node IP cannot be changed.

Returns 1 if updated successfully.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_fields	struct	False	
boot_state	string		Boot state
primary_network	struct		
network	string		Subnet address
dns2	string		IP address of secondary DNS server
dns1	string		IP address of primary DNS server
netmask	string		Subnet mask
method	string		Addressing method (e.g., 'static' or 'dhcp')
broadcast	string		Network broadcast address
mac	string		MAC address
gateway	string		IP address of primary gateway



Name	Type	Optional	Description
ssh_host_key	string		Last known SSH host key

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteAddress

Status:

current

Prototype:

DeleteAddress (auth, address\_id)

Description:

Deletes an address.

PIs may only delete addresses from their own sites.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
address_id	int	False	Address identifier

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteAddressTypeFromAddress

Status:

current

Prototype:

DeleteAddressTypeFromAddress (auth, address\_type\_id\_or\_name, address\_id)

Description:

Deletes an address type from the specified address.  
PIs may only update addresses of their own sites.  
Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
address_type_id_or_name		False	
address_type_id_or_name	int		Address type identifier
address_type_id_or_name	string		Address type
address_id	int	False	Address identifier

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteAddressType

Status:

current

Prototype:

DeleteAddressType (auth, address\_type\_id\_or\_name)

Description:

Deletes an address type.  
Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
address_type_id_or_name		False	
address_type_id_or_name	int		Address type identifier
address_type_id_or_name	string		Address type

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteBootState

Status:

current

Prototype:

DeleteBootState (auth, name)

Description:

Deletes a node boot state.

WARNING: This will cause the deletion of all nodes in this boot state.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
name	string	False	Boot state

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteConfFileFromNodeGroup

Status:

current

Prototype:

DeleteConfFileFromNodeGroup (auth, conf\_file\_id, nodegroup\_id\_or\_name)

Description:

Deletes a configuration file from the specified nodegroup. If the nodegroup is not linked to the configuration file, no errors are returned.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
conf_file_id	int	False	Configuration file identifier
nodegroup_id_or_name	mixed	False	
node-group_id_or_name	int		Node group identifier
node-group_id_or_name	string		Node group name

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteConfFileFromNode

Status:

current

Prototype:

DeleteConfFileFromNode (auth, conf\_file\_id, node\_id\_or\_hostname)

Description:

Deletes a configuration file from the specified node. If the node is not linked to the configuration file, no errors are returned.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
conf_file_id	int	False	Configuration file identifier
node_id_or_hostname	mixed	False	
node_id_or_hostname	int		Node identifier
node_id_or_hostname	string		Fully qualified hostname

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteConfFile

Status:

current

Prototype:

DeleteConfFile (auth, conf\_file\_id)

Description:

Returns an array of structs containing details about node configuration files. If conf\_file\_ids is specified, only the specified configuration files will be queried.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
conf_file_id	int	False	Configuration file identifier

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteKey

Status:

current

Prototype:

DeleteKey (auth, key\_id)

Description:

Deletes a key.

Non-admins may only delete their own keys.

Returns 1 if successful, faults otherwise.

Parameters:

<b>Name</b>	<b>Type</b>	<b>Optional</b>	<b>Description</b>
auth	struct	False	API authentication structure
key_id	int	False	Key identifier

Returns:

<b>Name</b>	<b>Type</b>	<b>Optional</b>	<b>Description</b>
	int		1 if successful

## DeleteKeyType

Status:

current

Prototype:

DeleteKeyType (auth, name)

Description:

Deletes a key type.

WARNING: This will cause the deletion of all keys of this type.

Returns 1 if successful, faults otherwise.

Parameters:

<b>Name</b>	<b>Type</b>	<b>Optional</b>	<b>Description</b>
auth	struct	False	API authentication structure
name	string	False	Key type

Returns:

<b>Name</b>	<b>Type</b>	<b>Optional</b>	<b>Description</b>
	int		1 if successful

## DeleteMessage

Status:

current

Prototype:

DeleteMessage (auth, message\_id)

## Description:

Deletes a message template.

Returns 1 if successful, faults otherwise.

## Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
message_id	string	False	Message identifier

## Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteNetworkMethod

## Status:

current

## Prototype:

DeleteNetworkMethod (auth, name)

## Description:

Deletes a network method.

WARNING: This will cause the deletion of all network interfaces that use this method.

Returns 1 if successful, faults otherwise.

## Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
name	string	False	Network method

## Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteNetworkType

Status:

current

Prototype:

DeleteNetworkType (auth, name)

Description:

Deletes a network type.

WARNING: This will cause the deletion of all network interfaces that use this type.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
name	string	False	Network type

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteNodeFromNodeGroup

Status:

current

Prototype:

DeleteNodeFromNodeGroup (auth, node\_id\_or\_hostname, nodegroup\_id\_or\_name)

Description:

Removes a node from the specified node group.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_id_or_hostname	mixed	False	
node_id_or_hostname	int		Node identifier
node_id_or_hostname	string		Fully qualified hostname



Name	Type	Optional	Description
nodegroup_id_or_name	mixed	False	
node-group_id_or_name	int		Node group identifier
node-group_id_or_name	string		Node group name

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteNodeFromPCU

Status:

current

Prototype:

DeleteNodeFromPCU (auth, node\_id\_or\_hostname, pcu\_id)

Description:

Deletes a node from a PCU.

Non-admins may only update PCUs at their sites.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_id_or_hostname	mixed	False	
node_id_or_hostname	int		Node identifier
node_id_or_hostname	string		Fully qualified hostname
pcu_id	int	False	PCU identifier

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteNodeGroup

Status:

current

Prototype:

DeleteNodeGroup (auth, node\_group\_id\_or\_name)

Description:

Delete an existing Node Group.

ins may delete any node group

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_group_id_or_name	mixed	False	
node_group_id_or_name	int		Node group identifier
node_group_id_or_name	string		Node group name

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteNodeNetwork

Status:

current

Prototype:

DeleteNodeNetwork (auth, nodenetwork\_id)

Description:

Deletes an existing node network interface.

Admins may delete any node network. PIs and techs may only delete node network interfaces associated with nodes at their sites.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
------	------	----------	-------------

Name	Type	Optional	Description
auth	struct	False	API authentication structure
nodenetwork_id	int	False	Node interface identifier

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteNode

Status:

current

Prototype:

DeleteNode (auth, node\_id\_or\_hostname)

Description:

Mark an existing node as deleted.

PIs and techs may only delete nodes at their own sites. ins may delete nodes at any site.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_id_or_hostname	mixed	False	
node_id_or_hostname	int		Node identifier
node_id_or_hostname	string		Fully qualified hostname

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeletePCU

Status:

current

Prototype:

DeletePCU (auth, pcu\_id)

Description:

Deletes a PCU.

Non-admins may only delete PCUs at their sites.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
pcu_id	int	False	PCU identifier

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeletePeer

Status:

current

Prototype:

DeletePeer (auth, peer\_id\_or\_name)

Description:

Mark an existing peer as deleted. All entities (e.g., slices, keys, nodes, etc.) for which this peer is authoritative will also be deleted or marked as deleted.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
peer_id_or_name	mixed	False	
peer_id_or_name	int		Peer identifier
peer_id_or_name	string		Peer name

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeletePersonFromSite

Status:

current

Prototype:

DeletePersonFromSite (auth, person\_id\_or\_email, site\_id\_or\_login\_base)

Description:

Removes the specified person from the specified site. If the person is not a member of the specified site, no error is returned.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	
person_id_or_email	int		Account identifier
person_email	string		Primary e-mail address
site_id_or_login_base	mixed	False	
site_id_or_login_base	int		Site identifier
site_id_or_login_base	string		Site slice prefix

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeletePersonFromSlice

Status:

current

Prototype:

DeletePersonFromSlice (auth, person\_id\_or\_email, slice\_id\_or\_name)

Description:

Deletes the specified person from the specified slice. If the person is not a member of the slice, no errors are returned.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	
person_id_or_email	int		Account identifier
person_id_or_email	string		Primary e-mail address
slice_id_or_name	mixed	False	
slice_id_or_name	int		Slice identifier
slice_id_or_name	string		Slice name

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeletePerson

Status:

current

Prototype:

DeletePerson (auth, person\_id\_or\_email)

Description:

Mark an existing account as deleted.

Users and techs can only delete themselves. PIs can only delete themselves and other non-PIs at their sites. ins can delete anyone.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	
person_id_or_email	int		Account identifier
person_id_or_email	string		Primary e-mail address

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteRoleFromPerson

Status:

current

Prototype:

DeleteRoleFromPerson (auth, role\_id\_or\_name, person\_id\_or\_email)

Description:

Deletes the specified role from the person.

PIs can only revoke the tech and user roles from users and techs at their sites. ins can revoke any role from any user.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
role_id_or_name	mixed	False	
role_id_or_name	int		Role identifier
role_id_or_name	string		Role
person_id_or_email	mixed	False	
person_id_or_email	int		Account identifier

Name	Type	Optional	Description
person_id_or_email	string		Primary e-mail address

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteRole

Status:

current

Prototype:

DeleteRole (auth, role\_id\_or\_name)

Description:

Deletes a role.

WARNING: This will remove the specified role from all accounts that possess it, and from all node and slice attributes that refer to it.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
role_id_or_name	mixed	False	
role_id_or_name	int		Role identifier
role_id_or_name	string		Role

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteSession

Status:

current

Prototype:



DeleteSession (auth)

Description:

Invalidates the current session.

Returns 1 if successful.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteSite

Status:

current

Prototype:

DeleteSite (auth, site\_id\_or\_login\_base)

Description:

Mark an existing site as deleted. The accounts of people who are not members of at least one other non-deleted site will also be marked as deleted. Nodes, PCUs, and slices associated with the site will be deleted.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
site_id_or_login_base	mixed	False	
site_id_or_login_base	int		Site identifier
site_id_or_login_base	string		Site slice prefix

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteSliceAttribute

Status:

current

Prototype:

DeleteSliceAttribute (auth, slice\_attribute\_id)

Description:

Deletes the specified slice or sliver attribute.

Attributes may require the caller to have a particular role in order to be deleted. Users may only delete attributes of slices or slivers of which they are members. PIs may only delete attributes of slices or slivers at their sites, or of which they are members. Admins may delete attributes of any slice or sliver.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
slice_attribute_id	int	False	Slice attribute identifier

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteSliceAttributeType

Status:

current

Prototype:

DeleteSliceAttributeType (auth, attribute\_type\_id\_or\_name)

Description:

Deletes the specified slice attribute.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
------	------	----------	-------------

Name	Type	Optional	Description
auth	struct	False	API authentication structure
attribute_type_id_or_name	mixed	False	
at-tribute_type_id_or_name	int		Slice attribute type identifier
at-tribute_type_id_or_name	string		Slice attribute type name

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteSliceFromNodes

Status:

current

Prototype:

DeleteSliceFromNodes (auth, slice\_id\_or\_name, node\_id\_or\_hostname\_list)

Description:

Deletes the specified slice from the specified nodes. If the slice is not associated with a node, no errors are returned.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
slice_id_or_name	mixed	False	
slice_id_or_name	int		Slice identifier
slice_id_or_name	string		Slice name
node_id_or_hostname_list	list	False	
	mixed		
	int		Node identifier
	string		Fully qualified hostname

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteSliceInstantiation

Status:

current

Prototype:

DeleteSliceInstantiation (auth, instantiation)

Description:

Deletes a slice instantiation state.

WARNING: This will cause the deletion of all slices of this instantiation.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
instantiation	string	False	Slice instantiation state

Returns:

Name	Type	Optional	Description
	int		1 if successful

## DeleteSlice

Status:

current

Prototype:

DeleteSlice (auth, slice\_id\_or\_name)

Description:

Deletes the specified slice.

Users may only delete slices of which they are members. PIs may delete any of the slices at their sites, or any slices of which they are members. Admins may delete any slice.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
slice_id_or_name	mixed	False	
slice_id_or_name	int		Slice identifier
slice_id_or_name	string		Slice name

Returns:

Name	Type	Optional	Description
	int		1 if successful

## GetAddresses

Status:

current

Prototype:

GetAddresses (auth, address\_filter, return\_fields)

Description:

Returns an array of structs containing details about addresses. If address\_filter is specified and is an array of address identifiers, or a struct of address attributes, only addresses matching the filter will be returned. If return\_fields is specified, only the specified details will be returned.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
address_filter	mixed	True	
address_filter	array		
address_filter	int		Address identifier
address_filter	struct		Attribute filter
return_fields	array	True	List of fields to return

Returns:

Name	Type	Optional	Description
	array		
	struct		
city	string		City
address_id	int		Address identifier

Name	Type	Optional	Description
country	string		Country
line3	string		Address line 3
line2	string		Address line 2
line1	string		Address line 1
address_type_ids	array		Address type identifiers
state	string		State or province
postalcode	string		Postal code
address_types	array		Address types

## GetAddressTypes

Status:

current

Prototype:

GetAddressTypes (auth, address\_type\_filter, return\_fields)

Description:

Returns an array of structs containing details about address types. If `address_type_filter` is specified and is an array of address type identifiers, or a struct of address type attributes, only address types matching the filter will be returned. If `return_fields` is specified, only the specified details will be returned.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
address_type_filter	mixed	True	
address_type_filter	array		
	mixed		
	int		Address type identifier
	string		Address type
address_type_filter	struct		Attribute filter
return_fields	array	True	List of fields to return

Returns:

Name	Type	Optional	Description
	array		
	struct		

Name	Type	Optional	Description
name	string		Address type
address_type_id	int		Address type identifier
description	string		Address type description

## GetBootStates

Status:

current

Prototype:

GetBootStates (auth)

Description:

Returns an array of all valid node boot states.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure

Returns:

Name	Type	Optional	Description
	array		
	string		Boot state

## GetConfFiles

Status:

current

Prototype:

GetConfFiles (auth, conf\_file\_filter, return\_fields)

Description:

Returns an array of structs containing details about configuration files. If `conf_file_filter` is specified and is an array of configuration file identifiers, or a struct of configuration file attributes, only configuration files matching the filter will be returned. If `return_fields` is specified, only the specified details will be returned.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
conf_file_filter	mixed	True	
conf_file_filter	array		
	int		Configuration file identifier
conf_file_filter	struct		Attribute filter
return_fields	array	True	List of fields to return

Returns:

Name	Type	Optional	Description
	array		
	struct		
file_owner	string		chown(1) owner
postinstall_cmd	string		Shell command to execute after installing
error_cmd	string		Shell command to execute if any error occurs
preinstall_cmd	string		Shell command to execute prior to installing
node_ids	int		List of nodes linked to this file
dest	string		Absolute path where file should be installed
ig-nore_cmd_errors	boolean		Install file anyway even if an error occurs
enabled	boolean		Configuration file is active
conf_file_id	int		Configuration file identifier
file_permissions	string		chmod(1) permissions
source	string		Relative path on the boot server where file can be downloaded
nodegroup_ids	int		List of node groups linked to this file
always_update	boolean		Always attempt to install file even if unchanged
file_group	string		chgrp(1) owner



## GetEvents

Status:

current

Prototype:

GetEvents (auth, event\_filter, return\_fields)

Description:

Returns an array of structs containing details about events and faults. If event\_filter is specified and is an array of event identifiers, or a struct of event attributes, only events matching the filter will be returned. If return\_fields is specified, only the specified details will be returned.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
event_filter	mixed	True	
event_filter	array		
event_filter	int		Event identifier
event_filter	struct		Attribute filter
return_fields	array	True	List of fields to return

Returns:

Name	Type	Optional	Description
	array		
	struct		
fault_code	int		Event fault code
event_id	int		Event identifier
object_ids	array		IDs of objects affected by this event
node_id	int		Identifier of node responsible for event, if any
call	string		Call responsible for this event, including parameters
time	int		Date and time that the event took place, in seconds since UNIX epoch
person_id	int		Identifier of person responsible for event, if any
runtime	double		Runtime of event
call_name	string		Call responsible for this event

## GetKeys

Status:

current

Prototype:

GetKeys (auth, key\_filter, return\_fields)

Description:

Returns an array of structs containing details about keys. If key\_filter is specified and is an array of key identifiers, or a struct of key attributes, only keys matching the filter will be returned. If return\_fields is specified, only the specified details will be returned.

Admin may query all keys. Non-admins may only query their own keys.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
key_filter	mixed	True	
key_filter	array		
key_filter	mixed		
key_filter	int		Key identifier
key_filter	struct		Attribute filter
return_fields	array	True	List of fields to return

Returns:

Name	Type	Optional	Description
	array		
	struct		
key_type	string		Key type
key_id	int		Key identifier
peer_id	int		Peer at which this node is managed
key	string		Key value

## GetKeyTypes

Status:

current

Prototype:

GetKeyTypes (auth)

Description:

Returns an array of all valid key types.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure

Returns:

Name	Type	Optional	Description
	array		
	string		Key type

## GetMessages

Status:

current

Prototype:

GetMessages (auth, message\_ids)

Description:

Return an array of structs containing details about message templates. If message\_ids is specified, only the specified messages will be queried.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
message_ids	array string	True	Message identifier

Returns:

Name	Type	Optional	Description
	array		
	struct		
enabled	boolean		Message is enabled
message_id	string		Message identifier
template	string		Message template
subject	string		Message summary

## GetNetworkMethods

Status:

current

Prototype:

GetNetworkMethods (auth)

Description:

Returns a list of all valid network methods.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure

Returns:

Name	Type	Optional	Description
	array		
	string		Network method

## GetNetworkTypes

Status:

current

Prototype:

GetNetworkTypes (auth)

Description:

Returns a list of all valid network types.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure

Returns:

Name	Type	Optional	Description
------	------	----------	-------------

Name	Type	Optional	Description
	array		
	string		Network type

## GetNodeGroups

Status:

current

Prototype:

GetNodeGroups (auth, nodegroup\_filter, return\_fields)

Description:

Returns an array of structs containing details about node groups. If `nodegroup_filter` is specified and is an array of node group identifiers or names, or a struct of node group attributes, only node groups matching the filter will be returned. If `return_fields` is specified, only the specified details will be returned.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
nodegroup_filter	mixed	True	
nodegroup_filter	array		
	mixed		
	int		Node group identifier
	string		Node group name
nodegroup_filter	struct		Attribute filter
return_fields	array	True	List of fields to return

Returns:

Name	Type	Optional	Description
	array		
	struct		
node_ids	array		List of nodes in this node group
nodegroup_id	int		Node group identifier
name	string		Node group name
conf_file_ids	array		List of configuration files specific to this node group
description	string		Node group description

## GetNodeNetworks

Status:

current

Prototype:

GetNodeNetworks (auth, nodenetwork\_filter, return\_fields)

Description:

Returns an array of structs containing details about node network interfaces. If `nodenetworks_filter` is specified and is an array of node network identifiers, or a struct of node network attributes, only node network interfaces matching the filter will be returned. If `return_fields` is specified, only the specified details will be returned.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
nodenetwork_filter	mixed	True	
nodenetwork_filter	array		
	int		Node interface identifier
nodenetwork_filter	struct		Attribute filter
return_fields	array	True	List of fields to return

Returns:

Name	Type	Optional	Description
	array		
	struct		
broadcast	string		Network broadcast address
is_primary	boolean		Is the primary interface for this node
network	string		Subnet address
ip	string		IP address
dns1	string		IP address of primary DNS server
hostname	string		(Optional) Hostname
netmask	string		Subnet mask
gateway	string		IP address of primary gateway

Name	Type	Optional	Description
nodenetwork_id	int		Node interface identifier
mac	string		MAC address
node_id	int		Node associated with this interface
dns2	string		IP address of secondary DNS server
bwlimit	int		Bandwidth limit
type	string		Address type (e.g., 'ipv4')
method	string		Addressing method (e.g., 'static' or 'dhcp')

## GetNodes

Status:

current

Prototype:

GetNodes (auth, node\_filter, return\_fields)

Description:

Returns an array of structs containing details about nodes. If node\_filter is specified and is an array of node identifiers or hostnames, or a struct of node attributes, only nodes matching the filter will be returned. If return\_fields is specified, only the specified details will be returned.

Some fields may only be viewed by admins.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_filter	mixed	True	
node_filter	array		
	mixed		
	int		Node identifier
	string		Fully qualified hostname
node_filter	struct		Attribute filter
return_fields	array	True	List of fields to return

Returns:

Name	Type	Optional	Description
	array		
	struct		
session	string		(Admin only) Node session value
pcu_ids	array		List of PCUs that control this node
nodegroup_ids	array		List of node groups that this node is in
last_updated	int		Date and time when node entry was created
version	string		Apparent Boot CD version
nodenetwork_ids	array		List of network interfaces that this node has
boot_state	string		Boot state
hostname	string		Fully qualified hostname
site_id	int		Site at which this node is located
ports	array		List of PCU ports that this node is connected to
slice_ids	array		List of slices on this node
boot_nonce	string		(Admin only) Random value generated by the node at last boot
node_id	int		Node identifier
key	string		(Admin only) Node key
date_created	int		Date and time when node entry was created
model	string		Make and model of the actual machine
peer_id	int		Peer at which this node is managed
conf_file_ids	array		List of configuration files specific to this node
ssh_rsa_key	string		Last known SSH host key

## GetPCUs

Status:

current



Prototype:

GetPCUs (auth, pcu\_filter, return\_fields)

Description:

Returns an array of structs containing details about power control units (PCUs). If `pcu_filter` is specified and is an array of PCU identifiers, or a struct of PCU attributes, only PCUs matching the filter will be returned. If `return_fields` is specified, only the specified details will be returned.

Admin may query all PCUs. Non-admins may only query the PCUs at their sites.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
pcu_filter	mixed	True	
pcu_filter	array		
	int		PCU identifier
pcu_filter	struct		Attribute filter
return_fields	array	True	List of fields to return

Returns:

Name	Type	Optional	Description
	array		
	struct		
username	string		PCU username
protocol	string		PCU protocol, e.g. ssh, https, telnet
node_ids	array		List of nodes that this PCU controls
ip	string		PCU IP address
pcu_id	int		PCU identifier
hostname	string		PCU hostname
site_id	int		Identifier of site where PCU is located
ports	array		List of the port numbers that each node is connected to
model	string		PCU model string
password	string		PCU username
notes	string		Miscellaneous notes

## GetPeerData

Status:

current

Prototype:

GetPeerData (auth, peer\_id\_or\_peername)

Description:

Gather all data needed by RefreshPeer in a single xmlrpc request

Expects a peer id or peer name, that identifies the requesting peer

Returns a dict containing, for the various types of cached entities, the local objects as well as the ones attached to that peer

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
peer_id_or_peername	mixed	False	
peer_id_or_peernam	int		Peer identifier
peer_id_or_peernam	string		Peer name

Returns:

Name	Type	Optional	Description
	struct		
Keys-peer	array		List of foreign keys
Sites-local	array		List of local sites
Nodes-local	array		List of local nodes
Keys-local	array		List of local keys
SliceAttributeTypes-peer	array		List of foreign slice attribute types
Nodes-peer	array		List of foreign nodes
Persons-peer	array		List of foreign users
Slices-local	array		List of local slices
Sites-peer	array		List of foreign sites
SliceAttributes-local	array		List of local slice attributes
Slices-peer	array		List of foreign slices
SliceAttributeTypes-local	array		List of local slice attribute types
Persons-local	array		List of local users

Name	Type	Optional	Description
SliceAttributes-peer	array		List of foreign slice attributes

## GetPeerName

Status:

current

Prototype:

GetPeerName (auth)

Description:

Returns this peer's name, as defined in the config as PLC\_NAME

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure

Returns:

Name	Type	Optional	Description
	string		Peer name

## GetPeers

Status:

current

Prototype:

GetPeers (auth, peer\_filter, return\_fields)

Description:

Returns an array of structs containing details about peers. If person\_filter is specified and is an array of peer identifiers or peer names, or a struct of peer attributes, only peers matching the filter will be returned. If return\_fields is specified, only the specified details will be returned.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure

Name	Type	Optional	Description
peer_filter	mixed	True	
peer_filter	array		
	mixed		
	int		Peer identifier
	string		Peer name
peer_filter	struct		Attribute filter
return_fields	array	True	List of fields to return

Returns:

Name	Type	Optional	Description
	array		
	struct		
attribute_type_ids	array		List of slice attribute types for which this peer is authoritative
node_ids	array		List of nodes for which this peer is authoritative
key_ids	array		List of keys for which this peer is authoritative
person_ids	array		List of users for this peer is authoritative
slice_attribute_ids	array		List of slice attributes for which this peer is authoritative
peername	string		Peer name
peer_url	string		Peer API URL
slice_ids	array		List of slices for which this peer is authoritative
key	string		Peer GPG public key
cacert	string		Peer SSL public certificate
site_ids	array		List of sites for this peer is authoritative
peer_id	int		Peer identifier

## GetPersons

Status:

current

Prototype:

GetPersons (auth, person\_filter, return\_fields)

Description:

Returns an array of structs containing details about users. If `person_filter` is specified and is an array of user identifiers or usernames, or a struct of user attributes, only users matching the filter will be returned. If `return_fields` is specified, only the specified details will be returned.

Users and techs may only retrieve details about themselves. PIs may retrieve details about themselves and others at their sites. Admins may retrieve details about all accounts.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_filter	mixed	True	
person_filter	array		
	mixed		
	int		Account identifier
	string		Primary e-mail address
person_filter	struct		Attribute filter
return_fields	array	True	List of fields to return

Returns:

Name	Type	Optional	Description
	array		
	struct		
bio	string		Biography
first_name	string		Given name
last_name	string		Surname
last_updated	int		Date and time of last update
roles	array		List of roles
title	string		Title
url	string		Home page
key_ids	array		List of key identifiers
enabled	boolean		Has been enabled
slice_ids	array		List of slice identifiers
phone	string		Telephone number
role_ids	array		List of role identifiers
person_id	int		Account identifier
date_created	int		Date and time when account was created

Name	Type	Optional	Description
site_ids	array		List of site identifiers
peer_id	int		Peer at which this slice was created
email	string		Primary e-mail address

## GetRoles

Status:

current

Prototype:

GetRoles (auth)

Description:

Get an array of structs containing details about all roles.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure

Returns:

Name	Type	Optional	Description
	array		
	struct		
name	string		Role
role_id	int		Role identifier

## GetSession

Status:

current

Prototype:

GetSession (auth)

Description:

Returns a new session key if a user or node authenticated successfully, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure

Returns:

Name	Type	Optional	Description
	string		Session key

## GetSites

Status:

current

Prototype:

GetSites (auth, site\_filter, return\_fields)

Description:

Returns an array of structs containing details about sites. If site\_filter is specified and is an array of site identifiers or hostnames, or a struct of site attributes, only sites matching the filter will be returned. If return\_fields is specified, only the specified details will be returned.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
site_filter	mixed	True	
site_filter	array		
	mixed		
	int		Site identifier
	string		Site slice prefix
site_filter	struct		Attribute filter
return_fields	array	True	List of fields to return

Returns:

Name	Type	Optional	Description
	array		
	struct		
address_ids	array		List of address identifiers
pcu_ids	array		List of PCU identifiers

Name	Type	Optional	Description
last_updated	int		Date and time when site entry was last updated, in seconds since UNIX epoch
name	string		Full site name
node_ids	array		List of site node identifiers
url	string		URL of a page that describes the site
person_ids	array		List of account identifiers
site_id	int		Site identifier
longitude	double		Decimal longitude of the site
slice_ids	array		List of slice identifiers
max_slivers	int		Maximum number of slivers that the site is able to create
max_slices	int		Maximum number of slices that the site is able to create
login_base	string		Site slice prefix
date_created	int		Date and time when site entry was created, in seconds since UNIX epoch
latitude	double		Decimal latitude of the site
is_public	boolean		Publicly viewable site
peer_id	int		Peer at which this slice was created
abbreviated_name	string		Abbreviated site name

## GetSliceAttributes

Status:

current

Prototype:

GetSliceAttributes (auth, slice\_attribute\_filter, return\_fields)

Description:

Returns an array of structs containing details about slice and sliver attributes. An attribute is a sliver attribute if the node\_id field is set. If slice\_attribute\_filter is specified and is an array of slice attribute identifiers, or a struct of slice attribute attributes, only slice attributes matching the filter will be returned. If return\_fields is specified, only the specified details will be returned.



Users may only query attributes of slices or slivers of which they are members. PIs may only query attributes of slices or slivers at their sites, or of which they are members. Admins may query attributes of any slice or sliver.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
slice_attribute_filter	mixed	True	
slice_attribute_filter	array		
	int		Slice attribute identifier
slice_attribute_filter	struct		Attribute filter
return_fields	array	True	List of fields to return

Returns:

Name	Type	Optional	Description
	array		
	struct		
name	string		Slice attribute type name
slice_id	int		Slice identifier
slice_attribute_id	int		Slice attribute identifier
value	string		Slice attribute value
attribute_type_id	int		Slice attribute type identifier
node_id	int		Node identifier, if a sliver attribute
min_role_id	int		Minimum (least powerful) role that can set or change this attribute
peer_id	int		Peer at which this node is managed
description	string		Slice attribute type description

## GetSliceAttributeTypes

Status:

current

Prototype:

GetSliceAttributeTypes (auth, attribute\_type\_filter, return\_fields)

Description:

Returns an array of structs containing details about slice attribute types. If attribute\_type\_filter is specified and is an array of slice attribute type identifiers, or a struct of slice attribute type attributes, only slice attribute types matching the filter will be returned. If return\_fields is specified, only the specified details will be returned.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
attribute_type_filter	mixed	True	
at-tribute_type_filter	array		
	mixed		
	int		Slice attribute type identifier
	string		Slice attribute type name
at-tribute_type_filter	struct		Attribute filter
return_fields	array	True	List of fields to return

Returns:

Name	Type	Optional	Description
	array		
	struct		
attribute_type_id	int		Slice attribute type identifier
min_role_id	int		Minimum (least powerful) role that can set or change this attribute
peer_id	int		Peer at which this node is managed
name	string		Slice attribute type name
description	string		Slice attribute type description

## GetSliceInstantiations

Status:

current

Prototype:

GetSliceInstantiations (auth)

Description:

Returns an array of all valid slice instantiation states.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure

Returns:

Name	Type	Optional	Description
	array		
	string		Slice instantiation state

## GetSlices

Status:

current

Prototype:

GetSlices (auth, slice\_filter, return\_fields)

Description:

Returns an array of structs containing details about slices. If slice\_filter is specified and is an array of slice identifiers or slice names, or a struct of slice attributes, only slices matching the filter will be returned. If return\_fields is specified, only the specified details will be returned.

Users may only query slices of which they are members. PIs may query any of the slices at their sites. Admins may query any slice. If a slice that cannot be queried is specified in slice\_filter, details about that slice will not be returned.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
slice_filter	mixed	True	
slice_filter	array		
	mixed		

Name	Type	Optional	Description
	int		Slice identifier
	string		Slice name
slice_filter	struct		Attribute filter
return_fields	array	True	List of fields to return

Returns:

Name	Type	Optional	Description
	array		
	struct		
creator_person_id	int		Identifier of the account that created this slice
instantiation	string		Slice instantiation state
slice_attribute_ids	array		List of slice attributes
name	string		Slice name
slice_id	int		Slice identifier
created	int		Date and time when slice was created, in seconds since UNIX epoch
url	string		URL further describing this slice
max_nodes	int		Maximum number of nodes that can be assigned to this slice
person_ids	array		List of accounts that can use this slice
expires	int		Date and time when slice expires, in seconds since UNIX epoch
site_id	int		Identifier of the site to which this slice belongs
node_ids	array		List of nodes in this slice
peer_id	int		Peer at which this slice was created
description	string		Slice description

## GetSlivers

Status:

current

Prototype:

GetSlivers (auth, node\_filter)

Description:

Returns an array of structs representing nodes and their slivers (slices bound to nodes). If node\_filter is specified, only information about the specified nodes will be returned. If node\_filter is not specified and called by a node, only information about the caller will be returned.

All of the information returned by this call can be gathered from other calls, e.g. GetNodes, GetNodeNetworks, GetSlices, etc. This function exists primarily for the benefit of Node Manager and Federation Manager.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_filter	mixed	True	
node_filter	array		
	mixed		
	int		Node identifier
	string		Fully qualified hostname
node_filter	struct		Attribute filter

Returns:

Name	Type	Optional	Description
	array		
	struct		
timestamp	int		Timestamp of this call, in seconds since UNIX epoch
hostname	string		Fully qualified hostname
conf_files	array		
	struct		
file_owner	string		chown(1) owner
postinstall_cmd	string		Shell command to execute after installing
error_cmd	string		Shell command to execute if any error occurs
preinstall_cmd	string		Shell command to execute prior to installing

Name	Type	Optional	Description
node_ids	int		List of nodes linked to this file
dest	string		Absolute path where file should be installed
ignore_cmd_errors	boolean		Install file anyway even if an error occurs
enabled	boolean		Configuration file is active
conf_file_id	int		Configuration file identifier
file_permissions	string		chmod(1) permissions
source	string		Relative path on the boot server where file can be downloaded
nodegroup_ids	int		List of node groups linked to this file
always_update	boolean		Always attempt to install file even if unchanged
file_group	string		chgrp(1) owner
node_id	int		Node identifier
groups	array		
	string		Node group name
networks	array		
	struct		
broadcast	string		Network broadcast address
is_primary	boolean		Is the primary interface for this node
network	string		Subnet address
ip	string		IP address
dns1	string		IP address of primary DNS server
hostname	string		(Optional) Hostname
netmask	string		Subnet mask
gateway	string		IP address of primary gateway
nodenetwork_id	int		Node interface identifier
mac	string		MAC address
node_id	int		Node associated with this interface
dns2	string		IP address of secondary DNS server

Name	Type	Optional	Description
bwlimit	int		Bandwidth limit
type	string		Address type (e.g., 'ipv4')
method	string		Addressing method (e.g., 'static' or 'dhcp')
slivers	array struct		
instantiation	string		Slice instantiation state
name	string		Slice name
slice_id	int		Slice identifier
keys	array struct		
key_type	string		Key type
key	string		Key value
expires	int		Date and time when slice expires, in seconds since UNIX epoch
attributes	array struct		
name	string		Slice attribute type name
value	string		Slice attribute value

## RebootNode

Status:

current

Prototype:

RebootNode (auth, node\_id\_or\_hostname)

Description:

Sends the specified node a specially formatted UDP packet which should cause it to reboot immediately.

Admins can reboot any node. Techs and PIs can only reboot nodes at their site.

Returns 1 if the packet was successfully sent (which only whether the packet was sent, not whether the reboot was successful).

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_id_or_hostname	mixed	False	

Name	Type	Optional	Description
node_id_or_hostname	int		Node identifier
node_id_or_hostname	string		Fully qualified hostname

Returns:

Name	Type	Optional	Description
	int		1 if successful

## RefreshPeer

Status:

current

Prototype:

RefreshPeer (auth, peer\_id\_or\_peername)

Description:

First queries a remote PLC for its name and updates the local peers table

Then proceeds with fetching objects on the remote PLC, and updates the local database accordingly

It requires the remote peer to be aware of our own name (as configured in PLC\_NAME)

Returns a dict containing (\*) 'peername': the peer's name (\*) 'new\_xxx': the number of new objects from that peer - may be negative (\*) 'timers': various stats on performance for optimization

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
peer_id_or_peername	mixed	False	
peer_id_or_peername	int		Peer identifier
peer_id_or_peername	string		Peer name

Returns:

Name	Type	Optional	Description
	struct		
new_keys	array		List of new keys
new_sites	array		List of new sites



Name	Type	Optional	Description
new_slice_attributes	array		List of new slice attributes
new_slices	array		List of new slices
timers	struct		(Debug) Timing information
new_persons	array		List of new users
new_nodes	array		List of new nodes
new_slice_attribute_types	array		List of new slice attribute types

## SetPersonPrimarySite

Status:

current

Prototype:

SetPersonPrimarySite (auth, person\_id\_or\_email, site\_id\_or\_login\_base)

Description:

Makes the specified site the person's primary site. The person must already be a member of the site.

ins may update anyone. All others may only update themselves.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	
person_id_or_email	int		Account identifier
person_id_or_email	string		Primary e-mail address
site_id_or_login_base	mixed	False	
site_id_or_login_base	int		Site identifier
site_id_or_login_base	string		Site slice prefix

Returns:

Name	Type	Optional	Description
	int		1 if successful

## SliceCreate

Status:

deprecated

Prototype:

SliceCreate (auth, name, slice\_fields)

Description:

Deprecated. See AddSlice.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
name	string	False	Slice name
slice_fields	struct	True	
url	string		URL further describing this slice
max_nodes	int		Maximum number of nodes that can be assigned to this slice
instantiation	string		Slice instantiation state
name	string		Slice name
description	string		Slice description

Returns:

Name	Type	Optional	Description
	int		New slice_id (> 0) if successful

## SliceDelete

Status:

deprecated

Prototype:

SliceDelete (auth, slice\_id\_or\_name)

Description:

Deprecated. See DeleteSlice.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
slice_id_or_name	mixed	False	
slice_id_or_name	int		Slice identifier
slice_id_or_name	string		Slice name

Returns:

Name	Type	Optional	Description
	int		1 if successful

## UpdateAddress

Status:

current

Prototype:

UpdateAddress (auth, address\_id, address\_fields)

Description:

Updates the parameters of an existing address with the values in address\_fields.

PIs may only update addresses of their own sites.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
address_id	int	False	Address identifier
address_fields	struct	False	
city	string		City
country	string		Country
line3	string		Address line 3
line2	string		Address line 2
line1	string		Address line 1
state	string		State or province
postalcode	string		Postal code

Returns:

Name	Type	Optional	Description
	int		1 if successful

## UpdateAddressType

Status:

current

Prototype:

UpdateAddressType (auth, address\_type\_id\_or\_name, address\_type\_fields)

Description:

Updates the parameters of an existing address type with the values in address\_type\_fields.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
address_type_id_or_name	int	False	Address type identifier
address_type_id_or_name	string		Address type
address_type_fields	struct	False	
name	string		Address type
description	string		Address type description

Returns:

Name	Type	Optional	Description
	int		1 if successful

## UpdateConfFile

Status:

current

Prototype:

UpdateConfFile (auth, conf\_file\_id, conf\_file\_fields)

Description:

Updates a node configuration file. Only the fields specified in conf\_file\_fields are updated, all other fields are left untouched.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
conf_file_id	int	False	Configuration file identifier
conf_file_fields	struct	False	
file_owner	string		chown(1) owner
postinstall_cmd	string		Shell command to execute after installing
error_cmd	string		Shell command to execute if any error occurs
preinstall_cmd	string		Shell command to execute prior to installing
dest	string		Absolute path where file should be installed
ignore_cmd_errors	boolean		Install file anyway even if an error occurs
enabled	boolean		Configuration file is active
file_permissions	string		chmod(1) permissions
source	string		Relative path on the boot server where file can be downloaded
always_update	boolean		Always attempt to install file even if unchanged
file_group	string		chgrp(1) owner

Returns:

Name	Type	Optional	Description
	int		1 if successful

## UpdateKey

Status:

current

Prototype:

UpdateKey (auth, key\_id, key\_fields)

Description:

Updates the parameters of an existing key with the values in key\_fields.

Non-admins may only update their own keys.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
key_id	int	False	Key identifier
key_fields	struct	False	
key_type	string		Key type
key	string		Key value

Returns:

Name	Type	Optional	Description
	int		1 if successful

## UpdateMessage

Status:

current

Prototype:

UpdateMessage (auth, message\_id, message\_fields)

Description:

Updates the parameters of an existing message template with the values in message\_fields.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
message_id	string	False	Message identifier

Name	Type	Optional	Description
message_fields	struct	False	
enabled	boolean		Message is enabled
template	string		Message template

Returns:

Name	Type	Optional	Description
	int		1 if successful

## UpdateNodeGroup

Status:

current

Prototype:

UpdateNodeGroup (auth, nodegroup\_id\_or\_name, nodegroup\_fields)

Description:

Updates a custom node group.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
nodegroup_id_or_name	mixed	False	
node-group_id_or_name	int		Node group identifier
node-group_id_or_name	string		Node group name
nodegroup_fields	struct	False	
name	string		Node group name
description	string		Node group description

Returns:

Name	Type	Optional	Description
	int		1 if successful

## UpdateNodeNetwork

Status:

current

Prototype:

UpdateNodeNetwork (auth, nodenetwork\_id, nodenetwork\_fields)

Description:

Updates an existing node network. Any values specified in `nodenetwork_fields` are used, otherwise defaults are used. Acceptable values for method are `dhcp` and `static`. If type is `static`, then `ip`, `gateway`, `network`, `broadcast`, `netmask`, and `dns1` must all be specified in `nodenetwork_fields`. If type is `dhcp`, these parameters, even if specified, are ignored.

PIs and techs may only update networks associated with their own nodes. Admins may update any node network.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
nodenetwork_id	int	False	Node interface identifier
nodenetwork_fields	struct	False	
network	string		Subnet address
is_primary	boolean		Is the primary interface for this node
dns1	string		IP address of primary DNS server
hostname	string		(Optional) Hostname
mac	string		MAC address
bwlimit	int		Bandwidth limit
broadcast	string		Network broadcast address
method	string		Addressing method (e.g., 'static' or 'dhcp')
netmask	string		Subnet mask
node_id	int		Node associated with this interface
dns2	string		IP address of secondary DNS server
ip	string		IP address
type	string		Address type (e.g., 'ipv4')
gateway	string		IP address of primary gateway



Returns:

Name	Type	Optional	Description
	int		1 if successful

## UpdateNode

Status:

current

Prototype:

UpdateNode (auth, node\_id\_or\_hostname, node\_fields)

Description:

Updates a node. Only the fields specified in node\_fields are updated, all other fields are left untouched.

PIs and techs can update only the nodes at their sites. Only admins can update the key, session, and boot\_nonce fields.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
node_id_or_hostname	mixed	False	
node_id_or_hostname	int		Node identifier
node_id_or_hostname	string		Fully qualified hostname
node_fields	struct	False	
version	string		Apparent Boot CD version
boot_state	string		Boot state
hostname	string		Fully qualified hostname
boot_nonce	string		(Admin only) Random value generated by the node at last boot
session	string		(Admin only) Node session value
key	string		(Admin only) Node key
model	string		Make and model of the actual machine

Returns:

Name	Type	Optional	Description
	int		1 if successful

## UpdatePCU

Status:

current

Prototype:

UpdatePCU (auth, pcu\_id, pcu\_fields)

Description:

Updates the parameters of an existing PCU with the values in pcu\_fields.

Non-admins may only update PCUs at their sites.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
pcu_id	int	False	PCU identifier
pcu_fields	struct	False	
username	string		PCU username
protocol	string		PCU protocol, e.g. ssh, https, telnet
node_ids	array		List of nodes that this PCU controls
ip	string		PCU IP address
notes	string		Miscellaneous notes
hostname	string		PCU hostname
model	string		PCU model string
password	string		PCU username
ports	array		List of the port numbers that each node is connected to

Returns:

Name	Type	Optional	Description
	int		1 if successful

## UpdatePeer

Status:

current

Prototype:

UpdatePeer (auth, peer\_id\_or\_name, peer\_fields)

Description:

Updates a peer. Only the fields specified in peer\_fields are updated, all other fields are left untouched.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
peer_id_or_name	mixed	False	
peer_id_or_name	int		Peer identifier
peer_id_or_name	string		Peer name
peer_fields	struct	False	
key	string		Peer GPG public key
cacert	string		Peer SSL public certificate
peername	string		Peer name
peer_url	string		Peer API URL

Returns:

Name	Type	Optional	Description
	int		1 if successful

## UpdatePerson

Status:

current

Prototype:

UpdatePerson (auth, person\_id\_or\_email, person\_fields)

Description:

Updates a person. Only the fields specified in person\_fields are updated, all other fields are left untouched.

Users and techs can only update themselves. PIs can only update themselves and other non-PIs at their sites.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
person_id_or_email	mixed	False	
person_id_or_email	int		Account identifier
person_id_or_email	string		Primary e-mail address
person_fields	struct	False	
bio	string		Biography
first_name	string		Given name
last_name	string		Surname
title	string		Title
url	string		Home page
enabled	boolean		Has been enabled
phone	string		Telephone number
password	string		Account password in crypt() form
email	string		Primary e-mail address

Returns:

Name	Type	Optional	Description
	int		1 if successful

## UpdateSite

Status:

current

Prototype:

UpdateSite (auth, site\_id\_or\_login\_base, site\_fields)

Description:

Updates a site. Only the fields specified in update\_fields are updated, all other fields are left untouched.

PIs can only update sites they are a member of. Only admins can update max\_slices, max\_slivers, and login\_base.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
site_id_or_login_base	mixed	False	
site_id_or_login_base	int		Site identifier
site_id_or_login_base	string		Site slice prefix
site_fields	struct	False	
name	string		Full site name
url	string		URL of a page that describes the site
longitude	double		Decimal longitude of the site
latitude	double		Decimal latitude of the site
max_slices	int		Maximum number of slices that the site is able to create
login_base	string		Site slice prefix
max_slivers	int		Maximum number of slivers that the site is able to create
is_public	boolean		Publicly viewable site
abbreviated_name	string		Abbreviated site name

Returns:

Name	Type	Optional	Description
	int		1 if successful

## UpdateSliceAttribute

Status:

current

Prototype:

UpdateSliceAttribute (auth, slice\_attribute\_id, value)

Description:

Updates the value of an existing slice or sliver attribute.

Users may only update attributes of slices or slivers of which they are members. PIs may only update attributes of slices or slivers at their sites, or of which they are members.

Admins may update attributes of any slice or sliver.  
Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
slice_attribute_id	int	False	Slice attribute identifier
value	string	False	Slice attribute value

Returns:

Name	Type	Optional	Description
	int		1 if successful

## UpdateSliceAttributeType

Status:

current

Prototype:

UpdateSliceAttributeType (auth, attribute\_type\_id\_or\_name, attribute\_type\_fields)

Description:

Updates the parameters of an existing attribute with the values in attribute\_type\_fields.  
Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
attribute_type_id_or_name	<del>int</del>	False	
at-tribute_type_id_or_name	int		Slice attribute type identifier
at-tribute_type_id_or_name	string		Slice attribute type name
attribute_type_fields	struct	False	
min_role_id	int		Minimum (least powerful) role that can set or change this attribute

Name	Type	Optional	Description
name	string		Slice attribute type name
description	string		Slice attribute type description

Returns:

Name	Type	Optional	Description
	int		1 if successful

## UpdateSlice

Status:

current

Prototype:

UpdateSlice (auth, slice\_id\_or\_name, slice\_fields)

Description:

Updates the parameters of an existing slice with the values in slice\_fields.

Users may only update slices of which they are members. PIs may update any of the slices at their sites, or any slices of which they are members. Admins may update any slice.

Only PIs and admins may update max\_nodes. Slices cannot be renewed (by updating the expires parameter) more than 8 weeks into the future.

Returns 1 if successful, faults otherwise.

Parameters:

Name	Type	Optional	Description
auth	struct	False	API authentication structure
slice_id_or_name	mixed	False	
slice_id_or_name	int		Slice identifier
slice_id_or_name	string		Slice name
slice_fields	struct	False	
url	string		URL further describing this slice
max_nodes	int		Maximum number of nodes that can be assigned to this slice
instantiation	string		Slice instantiation state
expires	int		Date and time when slice expires, in seconds since UNIX epoch

Name	Type	Optional	Description
description	string		Slice description

Returns:

Name	Type	Optional	Description
	int		1 if successful

## system.listMethods

Status:

current

Prototype:

system.listMethods ()

Description:

This method lists all the methods that the XML-RPC server knows how to dispatch.

Parameters:

None

Returns:

Name	Type	Optional	Description
	array		List of methods

## system.methodHelp

Status:

current

Prototype:

system.methodHelp (method)

Description:

Returns help text if defined for the method passed, otherwise returns an empty string.

Parameters:

Name	Type	Optional	Description
method	string	False	Method name

Returns:



Name	Type	Optional	Description
	string		Method help

## system.methodSignature

Status:

current

Prototype:

system.methodSignature (method)

Description:

Returns an array of known signatures (an array of arrays) for the method name passed. If no signatures are known, returns a none-array (test for type != array to detect missing signature).

Parameters:

Name	Type	Optional	Description
method	string	False	Method name

Returns:

Name	Type	Optional	Description
	array		
	array		Method signature

## system.multicall

Status:

current

Prototype:

system.multicall (calls)

Description:

Process an array of calls, and return an array of results. Calls should be structs of the form {methodName: string, params: array}

Each result will either be a single-item array containing the result value, or a struct of the form

{faultCode: int, faultString: string}

This is useful when you need to make lots of small calls without lots of round trips.

Parameters:

Chapter 1. PlanetLab API Methods

<b>Name</b>	<b>Type</b>	<b>Optional</b>	<b>Description</b>
calls	array struct	False	
params	array		Method arguments
methodName	string		Method name

Returns:

<b>Name</b>	<b>Type</b>	<b>Optional</b>	<b>Description</b>
	mixed array mixed struct		
faultCode	int		XML-RPC fault code
faultString	int		XML-RPC fault detail