

UAPI Functions - EmailAuth::validate_current_spfs

Guide to UAPI

[Return Data](#)

[Filter Output](#)

[Sort Output](#)

[Paginate Output](#)

[Output Columns](#)

[Calls from the WHM API](#)

[Custom UAPI Modules](#)

Backup:: [fullbackup_to_ftp](#)

Backup:: [fullbackup_to_homedir](#)

Backup:: [fullbackup_to_scp_with_key](#)

Backup:: [fullbackup_to_scp_with_password](#)

Backup:: [list_backups](#)

Bandwidth:: [get_enabled_protocols](#)

Bandwidth:: [get_retention_periods](#)

Bandwidth:: [query](#)

Batch:: [strict](#)

BoxTrapper:: [blacklist_messages](#)

BoxTrapper:: [delete_messages](#)

BoxTrapper:: [deliver_messages](#)

BoxTrapper:: [get_configuration](#)

BoxTrapper:: [get_log](#)

BoxTrapper:: [get_message](#)

BoxTrapper:: [get_status](#)

BoxTrapper:: [ignore_messages](#)

BoxTrapper:: [list_queued_messages](#)

BoxTrapper:: [process_messages](#)

Description

This function retrieves the the Sender Policy Framework (SPF) records for one or more domains.



Important:

When you disable the MailSend role, the system **disables** this function. For more information, read our [How to Use Server Profiles](#) documentation.

Examples

```
https://hostname.example.com:2083/cpsess#####/execute/EmailAuth/validate_current_spfs?domain=example.com
```



Note:

This example calls the UAPI function via a cPanel session. For more information, read our [Guide to UAPI](#) documentation.

```
$cpanel = new CPANEL(); // Connect to cPanel - only do this once.
```

```
// Validate the domain's SPF records.  
$poll = $cpanel->uapi(  
    'EmailAuth', 'validate_current_spfs',  
    array(  
        'domain' => 'example.com'  
    )  
);
```



Note:

For more information, read our [Guide to the LiveAPI System](#).

```
my $cpliveapi = Cpanel::LiveAPI->new(); # Connect to cPanel - only do this once.
```

```
# Validate the domain's SPF records.  
my $poll = $cpliveapi->uapi(  
    'EmailAuth', 'validate_current_spfs',  
    {  
        'domain' => 'example.com'  
    }  
);
```



Note:

For more information, read our [Guide to the LiveAPI System](#).

Function information

Module:
EmailAuth

API Version:
[UAPI Available in:](#)
cPanel 78+
Methods:
GET, POST
Required Parameters:
domain
Return Formats:
JSON
Legacy Equivalents:

(none)

About this module

The EmailAuth module manages email authentication to ensure email deliverability.

About UAPI

UAPI accesses and modifies data for cPanel accounts.



Notes:

BoxTrapper:: [save_configuration](#)

BoxTrapper:: [set_status](#)

BoxTrapper:: [whitelist_messages](#)

Brand:: [read](#)

Branding:: [file](#)

Branding:: [get_applications](#)

Branding:: [get_application_information](#)

Branding:: [get_available_applications](#)

Branding:: [get_information_for_applications](#)

Branding:: [image](#)

Branding:: [include](#)

Branding:: [spritelist](#)

CacheBuster:: [read](#)

CacheBuster:: [update](#)

Chkserverd:: [get_existing_ports](#)

Chkserverd:: [get_existing_ports_ssl](#)

Chrome:: [get_domain](#)

Contactus:: [is_enabled](#)

cPAddons:: [get_available_addons](#)

cPAddons:: [get_instance_settings](#)

cPAddons:: [list_addon_instances](#)

cPGreyList:: [disable_all_domains](#)

cPGreyList:: [disable_domains](#)

cPGreyList:: [enable_all_domains](#)

cPGreyList:: [enable_domains](#)

cPGreyList:: [has_greylisting_enabled](#)

cPGreyList:: [list_domains](#)

DAV:: [disable_shared_global_addressbook](#)

```
<!-- Validate the domain's SPF records. -->
[% data = execute('EmailAuth', 'validate_current_spfs', {
'domain' => 'example.com' }) %]
[% FOREACH q = data.validate_current_spfs %]
  <p>
    [% q %]
  </p>
[% END %]
```



Note:

For more information, read our [Guide to Template Toolkit](#) documentation.

```
uapi --user=username EmailAuth validate_current_spfs
domain=example.com
```



Notes:

- You **must** [URI-encode](#) values.
- `username` represents your account-level username.
- For more information and additional output options, read our [Guide to UAPI](#) documentation or run the `uapi --help` command.
- If you run CloudLinux™, you **must** use the full path of the `uapi` command:

```
/usr/local/cpanel/bin/uapi
```

- These functions may require specific features. To enable or disable WHM's *Feature Manager* interface, use WHM's *Feature Manager* interface.
- Unless you call this API via the WHM

DAV:: enable_share
d_global_addressbo
ok

DAV:: get_calendar
_contacts_config

DAV:: has_shared_
global_addressbook

DAV:: is_dav_servic
e_enabled

DAV:: is_horde_ena
bled

DCV:: check_domai
ns_via_dns

DCV:: check_domai
ns_via_http

DCV:: ensure_doma
ins_can_pass_dcv

DomainInfo:: domai
ns_data

DomainInfo:: list_do
mains

DomainInfo:: main_
domain_builtin_sub
domain_aliases

DomainInfo:: single
_domain_data

DNS:: ensure_doma
ins_reside_only_loc
ally

DNS:: has_local_au
thority

DNSSEC:: disable_
dnssec

DNSSEC:: enable_
dnssec

DNSSEC:: fetch_ds
_records

DNSSEC:: set_nsec3

DNSSEC:: unset_ns
ec3

Email:: account_na
me

Email:: add_auto_re
sponder

Email:: add_domain
_forwarder

Email:: add_forward
er

Email:: add_list

Email:: add_mailma
n_delegates

Email:: add_mx

Email:: add_pop

```
{
  "apiversion": 3,
  "module": "EmailAuth",
  "func": "validate_current_spfs",
  "result": {
    "data": [
      {
        "domain": "example.com",
        "state": "VALID",
        "expected": "ip6:0:0:0:0:ffff:c0a8:101",
        "ip_address": "0:0:0:0:0:ffff:
c0a8:101",
        "ip_version": 6,
        "records": [
          {
            "state": "PASS",
            "current": "v=spf1 +a +mx ip6:0:0:0:0:
0:ffff:c0a8:101 -all"
          }
        ]
      }
    ],
    "errors": null,
    "messages": null,
    "metadata": {
      "transformed": 1
    },
    "status": 1,
    "warnings": null
  }
}
```



Note:

Use cPanel's *API Shell* interface (*cPanel >> Home >> Advanced >> API Shell*) to directly test cPanel API calls.

Parameters

Parameter	Type	Description	Possible values	Example
domain	<i>string</i>	Required The domain for which to check the SPF records.	A valid domain.	example.com

Note:
To query multiple domains, duplicate the parameter. For example, use the domain=example.com, domain=example2.com, and domain=example3.com parameters.

Returns

Return	Type	Description	Possible values	Example
--------	------	-------------	-----------------	---------

M
A
P
I
,
y
o
u
m
u
s
t
u
s
e
t
h
e
a
p
p
r
o
p
r
i
a
t
e
c
p
a
n
e
l
p
o
r
t
s
(
2
0
8
2
o
r
2
0
8
3
).
Y
o
u
c
a
n
a
l
s
o
c
a
l
l
s
o
m
e
f
u
n
c
t
i
o
n
s
v
i
a
c
p
a
n
e
l
'
s
W
e
b
m
a
i
l
p
o
r
t
s
(
2
0
9
5
o
r
2
0
9
6
).

Email:: [add_spam_filter](#)

Email:: [browse_mailbox](#)

Email:: [change_mx](#)

Email:: [check_fastmail](#)

Email:: [check_pipe_forwarder](#)

Email:: [delete_auto_responder](#)

Email:: [delete_domain_forwarder](#)

Email:: [delete_filter](#)

Email:: [delete_forwarder](#)

Email:: [delete_held_messages](#)

Email:: [delete_list](#)

Email:: [delete_mx](#)

Email:: [delete_pop](#)

Email:: [disable_filter](#)

Email:: [disable_mailbox_autocreate](#)

Email:: [disable_spam_assassin](#)

Email:: [disable_spam_autodelete](#)

Email:: [disable_spam_box](#)

Email:: [dispatch_client_settings](#)

Email:: [edit_pop_quota](#)

Email:: [enable_filter](#)

Email:: [enable_mailbox_autocreate](#)

Email:: [enable_spam_assassin](#)

Email:: [enable_spam_box](#)

Email:: [fetch_charmaps](#)

Email:: [fts_rescan_mailbox](#)

Email:: [generate_mailman_opt](#)

Email:: [get_auto_responder](#)

Email:: [get_charsets](#)

data	<i>array of hashes</i>	An array that contains information about a domain's SPF records.	Each hash contains the domain, state, expected, and records returns.	
domain	<i>string</i>	The queried domain. This function returns this value in the data array.	A valid domain.	example.com
state	<i>string</i>	The SPF record's status. This function returns this value in the data array.	<ul style="list-style-type: none"> VALID — A single SPF TXT record exists in the domain's DNS with the correct ip_address value or redirect mechanism. MISMATCHED — An SPF TXT record exists for the domain that does not match the ip_address value. MULTIPLE — Multiple SPF TXT records exist in the domain's DNS. MISSING — No SPF TXT record exists for the domain's DNS. 	VALID
expected	<i>string</i>	The SPF record for the domain in the DNS. This function returns this value in the data array.	A valid string.	ip6:0:0:0:0:ffff:c0a8:101
ip_address	<i>string</i>	The domain's IP address. This function returns this value in the data array.	A valid string.	0:0:0:0:0:ffff:c0a8:101
ip_version	<i>integer</i>	The IP address version. This function returns this value in the data array.	<ul style="list-style-type: none"> 4 6 	6

Find a function

Related functions

- [UAPI Modules - DCV](#) — The DCV module checks Domain Control Validation (DCV).
- [UAPI Functions - DCV::check_domains_via_http](#) — This function checks whether the account's domains can pass Domain Control Validation (DCV) via an HTTP request.
- [UAPI Functions - DCV::ensure_domains_can_pass_dcv](#) — This function indicates whether the account's domains can pass a Domain Control Validation (DCV) check.
- [UAPI Modules - DNS](#) — The DNS module checks DNS resolution issues.
- [UAPI Functions - DNS::ensure_domains_reside_only_locally](#) — This function indicates whether the account's domains resolve exclusively to this server.

Email: [get_client_settings](#)

Email: [get_default_email_quota](#)

Email: [get_default_email_quota_mib](#)

Email: [get_disk_usage](#)

Email: [get_filter](#)

Email: [get_held_message_count](#)

Email: [get_mailbox_autocreate](#)

Email: [get_mailman_delegates](#)

Email: [get_main_account_disk_usage](#)

Email: [get_main_account_disk_usage_bytes](#)

Email: [get_max_email_quota](#)

Email: [get_max_email_quota_mib](#)

Email: [get_pop_quota](#)

Email: [get_spam_settings](#)

Email: [get_webmail_settings](#)

Email: [has_delegated_mailman_lists](#)

Email: [has_plaintext_authentication](#)

Email: [hold_outgoing](#)

Email: [is_integer](#)

Email: [list_auto_responders](#)

Email: [list_default_address](#)

Email: [list_domain_forwarders](#)

Email: [list_filters](#)


Email: [list_filters_backups](#)

Email: [list_forwards](#)

Email: [list_forwards_backups](#)

Email: [list_lists](#)

Email: [list_mail_domains](#)

records	<i>array of hashes</i>	The SPF records of the domain's DNS. This function returns this value in the data array.	Each hash contains the current, reason and state returns.	
current	<i>string</i>	The SPF record's contents. This function returns this value in the records array.	A valid string.	v=spf1 +a +mx ip6:0:0:0:0:0:ffff:c0a8:101 ~all
reason	<i>string</i>	The reason why the SPF record is not correct, if one exists. This function returns this value in the records array.	An error message.  Note: If no errors exist, the function does not return this value.	example.com: Sender is not authorized by default to use 'example.com' in 'helo' identity, however domain is not currently prepared for false failures (mechanism '~all' matched)

Email: [list_mx](#)

Email: [list_pops](#)

Email: [list_pops_wit
h_disk](#)

Email: [list_system_
filter_info](#)

Email: [passwd_list](#)

Email: [passwd_pop](#)

Email: [release_out
going](#)

Email: [remove_mail
man_delegates](#)

Email: [reorder_filters](#)

Email: [set_always_
accept](#)

Email: [set_default_
address](#)

Email: [set_list_priv
acy_options](#)

Email: [stats_db_sta
tus](#)

Email: [store_filter](#)

Email: [suspend_inc
oming](#)

Email: [suspend_log
in](#)

Email: [suspend_out
going](#)

Email: [trace_filter](#)

Email: [unsuspend_i
ncoming](#)

Email: [unsuspend_l
ogin](#)

Email: [unsuspend_
outgoing](#)

Email: [verify_pass
word](#)

EmailAuth: [disable_
dkim](#)

EmailAuth: [enable_
dkim](#)

EmailAuth: [ensure_
dkim_keys_exist](#)

EmailAuth: [fetch_d
kim_private_keys](#)

EmailAuth: [install_d
kim_private_keys](#)

EmailAuth: [install_s
pf_records](#)

EmailAuth: [validate_
current_dkims](#)

s
t
a
t
e

string

The SPF record's status.



Note:

Th
es
e
va
lu
e
s
cor
res
po
nd
wit
h
RF
C7
20
8
se
cti
on
2.6.

This function returns this value in the records array.

- **PASS** — The SPF record confirms that the ip_address value is a valid sender.
- **NEUTRAL** — The current SPF record configuration does not determine the ip_address value's validity.
- **FAIL** — The SPF record states that the ip_address value is **not** a valid sender.
- **SOFTFAIL** — The SPF record states that the ip_address value is **not** a valid sender, but does not FAIL state it.
- **TEMPERROR** — The SPF record check resulted in a failure. For example, a network failure.
- **PERMERROR** — The domain's SPF records are **incorrect** and require manual correction.

PASS

EmailAuth:: validate
_current_ptrs

EmailAuth:: validate
_current_spfs

ExternalAuthenticati
on:: add_authn_link

ExternalAuthenticati
on:: configured_mo
dules

ExternalAuthenticati
on:: get_authn_links

ExternalAuthenticati
on:: has_external_a
uth_modules_config
ured

ExternalAuthenticati
on:: remove_authn_
link

Features:: get_featu
re_metadata

Features:: has_feat
ure

Features:: list_featu
res

Fileman:: autocompl
etedir

Fileman:: empty_tra
sh

Fileman:: get_file_c
ontent

Fileman:: get_file_in
formation

Fileman:: list_files

Fileman:: save_file_
content

Fileman:: transcode

Fileman:: upload_fil
es

Ftp:: add_ftp

Ftp:: allows_anony
mous_ftp

Ftp:: allows_anony
mous_ftp_incoming

Ftp:: delete_ftp

Ftp:: ftp_exists

Ftp:: get_ftp_daemo
n_info

Ftp:: get_port

Ftp:: get_quota

Ftp:: get_welcome_
message

Ftp:: kill_session

Ftp:: [list_ftp](#)

Ftp:: [list_ftp_with_dir](#)

Ftp:: [list_sessions](#)

Ftp:: [passwd](#)

Ftp:: [server_name](#)

Ftp:: [set_anonymous_ftp](#)

Ftp:: [set_anonymous_ftp_incoming](#)

Ftp:: [set_homedir](#)

Ftp:: [set_quota](#)

Ftp:: [set_welcome_message](#)

Gpg:: [delete_keypair](#)

Gpg:: [export_public_key](#)

Gpg:: [export_secret_key](#)

Gpg:: [generate_key](#)

Gpg:: [import_key](#)

Gpg:: [list_public_keys](#)

Gpg:: [list_secret_keys](#)

ImageManager:: [convert_file](#)

ImageManager:: [create_thumbnails](#)

ImageManager:: [get_dimensions](#)

ImageManager:: [resize_image](#)

Integration:: [fetch_url](#)

KnownHosts:: [create](#)

KnownHosts:: [delete](#)

KnownHosts:: [update](#)

KnownHosts:: [verify](#)

LangPHP:: [php_get_domain_handler](#)

LangPHP:: [php_get_impacted_domains](#)

LangPHP:: [php_get_installed_versions](#)

LangPHP:: [php_get_system_default_version](#)

LangPHP:: [php_get_vhost_versions](#)

LangPHP:: [php_ini_get_user_basic_directives](#)

LangPHP:: [php_ini_get_user_content](#)

LangPHP:: [php_ini_get_user_paths](#)

LangPHP:: [php_ini_set_user_basic_directives](#)

LangPHP:: [php_ini_set_user_content](#)

LangPHP:: [php_set_vhost_versions](#)

LastLogin:: [get_last_or_current_logged_in_ip](#)

Locale:: [get_attributes](#)

Locale:: [list_locales](#)

Locale:: [set_locale](#)

Mailboxes:: [expunge_mailbox_messages](#)

Mailboxes:: [expunge_messages_for_mailbox_guid](#)

Mailboxes:: [get_mailbox_status_list](#)

Mailboxes:: [has_utf8_mailbox_names](#)

Mailboxes:: [set_utf8_mailbox_names](#)

Market:: [cancel_pending_ssl_certificate](#)

Market:: [create_shopping_cart](#)

Market:: [get_all_products](#)

Market:: [get_certificate_status_details](#)

Market:: [get_login_url](#)

Market:: [get_pending_ssl_certificates](#)

Market:: [get_provider_specific_dcv_constraints](#)

Market:: [get_providers_list](#)

Market:: [get_ssl_certificate_if_available](#)

Market:: [process_ssl_pending_queue](#)

Market:: request_ssl
_certificates

Market:: set_status_
of_pending_queue_i
tems

Market:: set_url_aft
er_checkout

Market:: validate_lo
gin_token

Mime:: add_handler

Mime:: add_hotlink

Mime:: add_mime

Mime:: add_redirect

Mime:: delete_handl
er

Mime:: delete_hotlink

Mime:: delete_mime

Mime:: delete_redir
ect

Mime:: list_handlers

Mime:: list_hotlinks

Mime:: list_mime

Mime:: list_redirects

Mime:: redirect_info

ModSecurity:: disabl
e_all_domains

ModSecurity:: disabl
e_domains

ModSecurity:: enabl
e_all_domains

ModSecurity:: enabl
e_domains

ModSecurity:: has_
modsecurity_installed

ModSecurity:: list_d
omains

Mysql:: add_host

Mysql:: add_host_n
ote

Mysql:: check_data
base

Mysql:: create_data
base

Mysql:: create_user

Mysql:: delete_data
base

Mysql:: delete_host

Mysql:: delete_user

Mysql::dump_database_schema

Mysql::get_host_notes

Mysql::get_privileges_on_database

Mysql::get_restrictions

Mysql::get_server_information

Mysql::list_databases

Mysql::list_routines

Mysql::list_users

Mysql::locate_server

Mysql::rename_database

Mysql::rename_user

Mysql::repair_database

Mysql::revoke_access_to_database

Mysql::set_password

Mysql::set_privileges_on_database

Mysql::update_privileges

Notifications::get_notifications_count

NVData::get

NVData::set

Parser::firstfile_relative_uri

PassengerApps::disable_application

PassengerApps::edit_application

PassengerApps::enable_application

PassengerApps::ensure_deps

PassengerApps::list_applications

PassengerApps::register_application

PassengerApps::unregister_application

PasswdStrength::get_required_strength

Personalization::set

Personalization ::get

Postgresql:: create_
database

Postgresql:: create_
user

Postgresql:: delete_
database

Postgresql:: get_res
trictions

Postgresql:: grant_a
ll_privileges

Postgresql:: rename_
database

Postgresql:: rename_
user

Postgresql:: rename_
user_no_password

Postgresql:: revoke_
all_privileges

Postgresql:: set_pas
sword

Pushbullet:: send_te
st_message

Quota:: get_quota_i
nfo

Resellers:: list_acc
ounts

ResourceUsage:: g
et_usages

Restore:: directory_l
isting

Restore:: get_users

Restore:: query_file
_info

Restore:: restore_file

ServerInformation::
get_information

Session:: create_te
mp_user

SiteTemplates:: list_
site_templates

SiteTemplates:: list_
user_settings

SiteTemplates:: pub
lish

SpamAssassin :: cle
ar_spam_box

SpamAssassin:: get_
symbolic_test_na
mes

SpamAssassin:: get_
user_preferences

SpamAssassin:: update_user_preference

SSH:: get_port

SSL:: add_autossl_excluded_domains

SSL:: can_ssl_redirect

SSL:: check_shared_cert

SSL:: delete_cert

SSL:: delete_csr

SSL:: delete_key

SSL:: delete_ssl

SSL:: disable_mail_sni

SSL:: enable_mail_sni

SSL:: fetch_best_for_domain

SSL:: fetch_cert_info

SSL:: fetch_certificates_for_fqdns

SSL:: fetch_key_and_cabundle_for_certificate

SSL:: find_certificates_for_key

SSL:: find_csrs_for_key

SSL:: generate_cert

SSL:: generate_csr

SSL:: generate_key

SSL:: get_autossl_excluded_domains

SSL:: get_autossl_pending_queue

SSL:: get_autossl_problems

SSL:: get_cabundle

SSL:: get_cn_name

SSL:: install_ssl

SSL:: installed_host

SSL:: installed_hosts

SSL:: is_autossl_check_in_progress

SSL:: is_mail_sni_supported

SSL:: is_sni_supported

SSL:: list_certs
SSL:: list_csrs
SSL:: list_keys
SSL:: list_ssl_items
SSL:: mail_sni_status
SSL:: rebuild_mail_sni_config
SSL:: rebuildssldb
SSL:: remove_autossl_excluded_domains
SSL:: set_autossl_excluded_domains
SSL:: set_cert_friendly_name
SSL:: set_csr_friendly_name
SSL:: set_key_friendly_name
SSL:: set_primary_ssl
SSL:: set_ssl_share
SSL:: show_cert
SSL:: show_csr
SSL:: show_key
SSL:: start_autossl_check
SSL:: toggle_ssl_rejects_for_domains
SSL:: upload_cert
SSL:: upload_key
StatsBar:: get_stats
Styles:: current
Styles:: list
Styles:: set_default
Styles:: update
SubDomain:: addsubdomain
Themes:: get_theme_base
Themes:: list
Themes:: update
Tokens:: create_full_access
Tokens:: list
Tokens:: rename

Tokens:: [revoke](#)

TwoFactorAuth:: [generate_user_configuration](#)

TwoFactorAuth:: [get_user_configuration](#)

TwoFactorAuth:: [remove_user_configuration](#)

TwoFactorAuth:: [set_user_configuration](#)

UserManager:: [check_account_conflicts](#)

UserManager:: [create_user](#)

UserManager:: [delete_user](#)

UserManager:: [dismiss_user](#)

UserManager:: [edit_user](#)

UserManager:: [list_users](#)

UserManager:: [lookup_service_account](#)

UserManager:: [lookup_user](#)

UserManager:: [merge_service_account](#)

UserManager:: [unlink_service_account](#)

UserTasks:: [delete](#)

UserTasks:: [retrieve](#)

VersionControl:: [create](#)

VersionControl:: [delete](#)

VersionControl:: [retrieve](#)

VersionControl:: [update](#)

VersionControlDeployment:: [create](#)

VersionControlDeployment:: [delete](#)

VersionControlDeployment:: [retrieve](#)

WebDisk:: [delete_user](#)

WebDisk:: [set_home_dir](#)

WebDisk:: [set_password](#)

WebDisk:: [set_permissions](#)

WebmailApps:: [listwebmailapps](#)

WebVhost:: [listdomains](#)

WebVhost:: [list_ssl_capable_domains](#)

WordPressBackup:: [any_running](#)

WordPressBackup:: [cancel](#)

WordPressBackup:: [cleanup](#)

WordPressBackup:: [get_available_backups](#)

WordPressBackup:: [is_running](#)

WordPressBackup:: [start](#)

WordPressInstance
Manager:: [cancel_scan](#)

WordPressInstance
Manager:: [cleanup_scan](#)

WordPressInstance
Manager:: [change_admin_password](#)

WordPressInstance
Manager:: [configure_autoupdate](#)

WordPressInstance
Manager:: [get_api_version](#)

WordPressInstance
Manager:: [get_instance_by_id](#)

WordPressInstance
Manager:: [get_instances](#)

WordPressInstance
Manager:: [get_latest_wordpress_version_from_wordpress_org](#)

WordPressInstance
Manager:: [get_scan_results](#)

WordPressInstance
Manager:: [is_installable_addon_available](#)

WordPressInstance
Manager:: [start_scan](#)

WordPressRestore:: [any_running](#)

WordPressRestore::
cleanup

WordPressRestore::
start