NAME

default.conf − IPA configuration file

SYNOPSIS

/etc/ipa/default.conf, ˜/.ipa/default.conf, /etc/ipa/server.conf, /etc/ipa/cli.conf

DESCRIPTION

The *default.conf* configuration file is used to set system−wide defaults to be applied when running IPA clients and servers.

Users may create an optional configuration file in *˜/.ipa/default.conf* which will be merged into the system−wide defaults file.

The following files are read, in order:

```
˜/.ipa/default.conf
/etc/ipa/<context>.conf
/etc/ipa/default.conf
built−in constants
```

The IPA server does not read ˜/.ipa/default.conf.

The first setting wins.

SYNTAX

The configuration options are not case sensitive. The values may be case sensitive, depending on the option.

Blank lines are ignored. Lines beginning with # are comments and are ignored.

Valid lines consist of an option name, an equals sign and a value. Spaces surrounding equals sign are ignored. An option terminates at the end of a line.

Values should not be quoted, the quotes will not be stripped.

```
# Wrong − don’t include quotes
verbose = "True"

# Right − Properly formatted options
verbose = True
verbose=True
```

Options must appear in the section named [global]. There are no other sections defined or used currently.

Options may be defined that are not used by IPA. Be careful of misspellings, they will not be rejected.

OPTIONS

The following options are relevant for the server:

basedn <base>

Specifies the base DN to use when performing LDAP operations. The base must be in DN format (dc=example,dc=com).

ca_agent_port <port>

Specifies the secure CA agent port. The default is 9443 for Dogtag 9, and 8443 for Dogtag 10.

ca_ee_port <port>

Specifies the secure CA end user port. The default is 9444 for Dogtag 9, and 8443 for Dogtag 10.

ca_host <hostname>
Specifies the hostname of the dogtag CA server. The default is the hostname of the IPA server.

ca_port <port>
Specifies the insecure CA end user port. The default is 9180 for Dogtag 9, and 8080 for Dogtag 10.

context <context>
Specifies the context that IPA is being executed in. IPA may operate differently depending on the context. The current defined contexts are cli and server. Additionally this value is used to load /etc/ipa/**context**.conf to provide context−specific configuration. For example, if you want to always perform client requests in verbose mode but do not want to have verbose enabled on the server, add the verbose option to */etc/ipa/cli.conf*.

debug <boolean>
When True provides detailed information. Specifically this set the global log level to "debug". Default is False.

dogtag_version <version>
Stores the version of Dogtag. Value 9 is assumed if not specified otherwise.

domain <domain>
The domain of the IPA server e.g. example.com.

enable_ra <boolean>
Specifies whether the CA is acting as an RA agent, such as when dogtag is being used as the Certificate Authority. This setting only applies to the IPA server configuration.

fallback <boolean>
Specifies whether an IPA client should attempt to fall back and try other services if the first connection fails.

host <hostname>
Specifies the hostname of the IPA server. This value is used to construct URL values on the client and server.

in_server <boolean>
Specifies whether requests should be forwarded to an IPA server or handled locally. This is used internally by IPA in a similar way as context. The same IPA framework is used by the ipa command−line tool and the server. This setting tells the framework whether it should execute the command as if on the server or forward it via XML−RPC to a remote server.

in_tree <boolean>
This is used in development and is generally a detected value. It means that the code is being executed within a source tree.

interactive <boolean>
Specifies whether values should be prompted for or not. The default is True.

ldap_uri <URI>
Specifies the URI of the IPA LDAP server to connect to. The URI scheme may be one of **ldap** or **ldapi**. The default is to use ldapi, e.g. ldapi://%2fvar%2frun%2fslapd−EXAMPLE−COM.socket

log_logger_XXX <comma separated list of regexps>
loggers matching regexp will be assigned XXX level.

Logger levels can be explicitly specified for specific loggers as opposed to a global logging level. Specific loggers are indicated by a list of regular expressions bound to a level. If a logger’s name matches the regexp then it is assigned that level. This config item must begin with "log_logger_level_" and then be followed by a symbolic or numeric log level, for example:

```
  log_logger_level_debug = ipalib\.dn\..*
```

```
log_logger_level_35 = ipalib\.plugins\.dogtag
```

The first line says any logger belonging to the ipalib.dn module will have it's level configured to debug.

The second line say the ipa.plugins.dogtag logger will be configured to level 35.

This config item is useful when you only want to see the log output from one or more selected loggers. Turning on the global debug flag will produce an enormous amount of output. This allows you to leave the global debug flag off and selectively enable output from a specific logger. Typically loggers are bound to classes and plugins.

Note: logger names are a dot ('.') separated list forming a path in the logger tree. The dot character is also a regular expression metacharacter (matches any character) therefore you will usually need to escape the dot in the logger names by preceeding it with a backslash.

mode <mode>

Specifies the mode the server is running in. The currently support values are **production** and **development**. When running in production mode some self−tests are skipped to improve performance.

mount_ipa <URI>

Specifies the mount point that the development server will register. The default is /ipa/

prompt_all <boolean>

Specifies that all options should be prompted for in the IPA client, even optional values. Default is False.

ra_plugin <name>

Specifies the name of the CA back end to use. The current options are **selfsign** and **dogtag**. This is a server−side setting. Changing this value is not recommended as the CA back end is only set up during initial installation.

realm <realm>

Specifies the Kerberos realm.

session_auth_duration <time duration spec>

Specifies the length of time authentication credentials cached in the session are valid. After the duration expires credentials will be automatically reacquired. Examples are "2 hours", "1h:30m", "10 minutes", "5min, 30sec".

session_duration_type <inactivity_timeout|from_start>

Specifies how the expiration of a session is computed. With **inactivity_timeout** the expiration time is advanced by the value of session_auth_duration everytime the user accesses the service. With **from_start** the session expiration is the start of the user's session plus the value of session_auth_duration.

server <hostname>

Specifies the IPA Server hostname. This option is deprecated.

startup_timeout <time in seconds>

Controls the amount of time waited when starting a service. The default value is 120 seconds.

startup_traceback <boolean>

If the IPA server fails to start and this value is True the server will attempt to generate a python traceback to make identifying the underlying problem easier.

validate_api <boolean>

Used internally in the IPA source package to verify that the API has not changed. This is used to prevent regressions. If it is true then some errors are ignored so enough of the IPA framework can be loaded to verify all of the API, even if optional components are not installed. The default is False.

verbose <boolean>

When True provides more information. Specifically this sets the global log level to "info".

wait_for_attr <boolean>

Debug option. Waits for asynchronous execution of 389-ds postoperation plugins before returning data to the client, therefore data added by postoperation plugins is included in the result. Increases execution time.

xmlrpc_uri <URI>

Specifies the URI of the XML−RPC server for a client. This is used by IPA and some external tools as well, such as ipa−getcert. e.g. https://ipa.example.com/ipa/xml

The following define the containers for the IPA server. Containers define where in the DIT that objects can be found. The full location is the value of container + basedn.

container_accounts: cn=accounts
container_applications: cn=applications,cn=configs,cn=policies
container_automount: cn=automount
container_configs: cn=configs,cn=policies
container_dns: cn=dns
container_entitlements: cn=entitlements,cn=etc
container_group: cn=groups,cn=accounts
container_hbac: cn=hbac
container_hbacservice: cn=hbacservices,cn=hbac
container_hbacservicegroup: cn=hbacservicegroups,cn=hbac
container_host: cn=computers,cn=accounts
container_hostgroup: cn=hostgroups,cn=accounts
container_netgroup: cn=ng,cn=alt
container_permission: cn=permissions,cn=pbac
container_policies: cn=policies
container_policygroups: cn=policygroups,cn=configs,cn=policies
container_policylinks: cn=policylinks,cn=configs,cn=policies
container_privilege: cn=privileges,cn=pbac
container_rolegroup: cn=roles,cn=accounts
container_roles: cn=roles,cn=policies
container_service: cn=services,cn=accounts
container_sudocmd: cn=sudocmds,cn=sudo
container_sudocmdgroup: cn=sudocmdgroups,cn=sudo
container_sudorule: cn=sudorules,cn=sudo
container_user: cn=users,cn=accounts
container_virtual: cn=virtual operations,cn=etc

FILES

/etc/ipa/default.conf

system−wide IPA configuration file

$HOME/.ipa/default.conf

user IPA configuration file

It is also possible to define context−specific configuration files. The **context** is set when the IPA api is initialized. The two currently defined contexts in IPA are **cli** and **server**. This is helpful, for example, if you only want **debug** enabled on the server and not in the client. If this is set to True in *default.conf* it will affect both the ipa client tool and the IPA server. If it is only set in *server.conf* then only the server will have **debug** set. These files will be loaded if they exist:

/etc/ipa/cli.conf

system−wide IPA client configuration file

/etc/ipa/server.conf
system−wide IPA server configuration file

SEE ALSO

ipa(1)