

User Defined Configuration Modification

Introduction

The system configuration files included asterisk configurations are stored into database first and then logistically generated by the background applications. Now during the usage, some of the customers (asterisk specialist) would like to manually modify the configurations sometimes. If they operate directly to the configurations of /etc/asterisk/, the configuration file will be updated when customers apply the changes on WEB GUI next time. Reboot the device will result in the update of configuration file as well. Namely, the customized modifications by clients can't be saved successfully.

Solution

If we want to save the user-defined data, we must save it to the database.

Now in the directory of '/persistent/var/lib/asterisk/db/', we create a directory '**custom-cfg**' to store user-defined data. If users want, they can supplement or modify an appropriate configuration file under this directory.

Modified data must saved in this format '*_custom.conf', such as the changes of sip.conf must be saved in '/persistent/custom-cfg/sip_custom.conf'.

Every time, when background application generates a configuration according to database, there will be a **specific thread** to check the contents of custom-cfg directory and modify the configuration file again.

Users-defined Format Data

Configuration files supported to be modified manually:

[modifyfile]

```
/etc/asterisk/extensions.conf  
/etc/asterisk/sip.conf  
/etc/asterisk/users.conf  
/etc/asterisk/iax.conf  
/etc/asterisk/queues.conf  
/etc/asterisk/features.conf  
/etc/asterisk/voicemail.conf  
/etc/asterisk/meetme.conf
```

```
/etc/asterisk/rc_org.conf
/etc/asterisk/rtp.conf
/etc/asterisk/http.conf
/etc/asterisk/musiconhold.conf
/etc/asterisk/followme.conf
/etc/asterisk/logger.conf
/etc/asterisk/manager.conf
/etc/asterisk/chan_dahdi.conf
/etc/asterisk/digital-channel.conf
```

```
[replacefile]
/etc/asterisk/dahdi-channels.conf
```

1. To modify or delete a specified field in the context.

The format of configuration files as below:

```
/persistent/custom-cfg/voicemail_custom.conf
```

```
[general]
```

```
maxmsg = 100 // It means the default value of masmsg is 100
```

```
;maxmin = 200 // It means the deletion of this configuration
```

```
...
```

```
features.conf -> followme.conf
```

As these configuration changes are generally for a configuration item's value, this approach should work well.

Every time, system read the configuration item that user requires to operation, and then operate this item from the corresponding configuration file.

Search the configuration item by the format 'configuration item=', such as search maxmsg=

2. extensions.conf special processing: replace specified context

Format as:

```
[macro-stdPrivacyexten]
```

```
exten => s,1,Dial(${ARG2},20,p)
```

```
exten => s,2,Goto(s-${DIALSTATUS},1)
```

```
exten => s-NOANSWER,1,VoiceMail(u${ARG1})
```

```
exten => s-NOANSWER,2,Goto(default,s,1)
```

```
...
```

It mainly used for extensions.conf modification.

Replace the context in extensions.conf with user-defined context.

3. sip.conf special processing: replace the fields of 'register ='

Configuration of sip.conf, except config options of register, all other modifications can use the first format.

For fields of register =, we need to replace the data of 'register=' in sip.conf with

user-defined data of 'register='.

4. users.conf/iax.conf/queue.conf special processing: modify multiple context at the same time

Those configurations allow using the first format of modification.

Other context modifications of support blank name, which means the modification to all contexts except general.

Format as:

[]

hassip = yes

It means the all context except [general] in current configuration need to do the modification of hassip = yes

5. reserve

<The end>