Extensions Class (Manipulating Dialplan)

Introduction

The Extensions class is not included by any module. It is passed during Apply Config changes to the BMO function BMO Hooks#DialplanHooks or <moduleraw>_get_config if using functions.inc.php

- Introduction
- Core Methods
  - add
  - splice
  - replace
  - remove
  - addSectionComment
  - addSectionNoCustom
  - disableCustomContexts
  - addHint
  - addGlobal
  - addInclude
  - spliceInclude
  - addSwitch
  - addExec
- Dialplan Commands
  - gosub
  - messagesend
  - return
  - stackpop
  - gosubif
  - stasis
  - goto
  - gotoif
  - gotoiftime
  - while
  - endwhile
  - exitwhile

Core Methods

add

Add a new entry to the extensions_additional file
/** Add an entry to the extensions file
 * @param $section    The section to be added to
 * @param $extension  The extension used
 * @param $tag        A tag to use (to reference with basetag), use false
 *                    or "" if none
 * @param $command    The command to execute
 * @param $basetag    The tag to base this on. Only used in conjunction
 *                    with $addpriority
 * @param $addpriority  Finds the priority of the tag called $basetag, and
 *                        adds this
 * @return
 */
function add($section, $extension, $tag, $command, $basetag = false, $addpriority = false)

Example

PHP

```php
$ext->add('vmx, '555', '', new ext_noop('Timeout: going to timeout dest'));
$ext->add('vmx, '555', '', new ext_noop('Foo: Bar'));
```

extensions_additional

```plaintext
[my-context]
exten => 5555,1,Noop(Timeout: going to timeout dest)
exten => 5555,n,Noop(Foo: Bar)
```

splice
/**
 * This function allows new priorities to be injected into already
generated dialplan
 * usage: $ext->splice($context, $exten, $priority_number, new ext_goto
 * ('1','s','ext-did'));
 * if $priority is not numeric, it will interpret it as a tag and
try to inject
 * the command just prior to the first instruction it finds with
the specified tag
 * if it can't find the tag, it will inject it after the last
instruction
 * @method splice
 * @param  string  $section           The context to splice
 * @param  string  $extension         The extension to splice
 * @param  string  $priority          if $priority is not numeric, it will
interpolate it as a tag and try to
inject
 * the command just prior to the first
instruction
 * it if it can't find the tag, it will inject
it after the last instruction
 * @param  object  $command           Object of Extension
 * @param  string  $new_tag           New Priority tag to insert
 * @param  integer $offset            Offset of label
 * @param  boolean $fixmultiplelabels [description]
 */
function splice($section, $extension, $priority, $command, $new_tag="",
$offset=0, $fixmultiplelabels=false)  {

Example

PHP

$ext->splice('my-context', 's', 'dialapp', new \ext_noop('This is inserted
before priority dialapp'),'mypri');

extensions_additional

[my-context]
exten => s,n(mypri),Noop(This is inserted before priority dialapp)
exten => s,n(dialapp),Noop()
extensions_additional

[my-context]
  exten => s,n(nodial),Noop()
  exten => s,n(mypri),Noop(This is inserted after priority nodial)

replace

function replace($section, $extension, $priority, $command) {

remove

function remove($section, $extension, $priority) {

addSectionComment

function addSectionComment($section, $comment) {

addSectionNoCustom

function addSectionNoCustom($section, $setting) {

disableCustomContexts

function disableCustomContexts($setting) {

addHint

function addHint($section, $extension, $hintvalue) {

addGlobal
function addGlobal($globvar, $globval) {
}

addInclude

function addInclude($section, $incsection, $comment='') {
}

spliceInclude

function spliceInclude($section, $splicesection, $splicecomment, $incsection, $comment='') {
}

addSwitch

function addSwitch($section, $incsection) {
}

addExec

function addExec($section, $incsection) {

Dialplan Commands

gosub
Jump to label, saving return address.
Jumps to the label specified, saving the return address.
https://wiki.asterisk.org/wiki/display/AST/Asterisk+15+Application_Gosub

ext_gosub($pri, $ext = false, $context = false, $args='')

Example

PHP

$ext->add('context', '555', '', ext_gosub(1, '2', 'sub-record', 'arg'))

extensions_additional
messagesend

Send a text message.

Send a text message. The body of the message that will be sent is what is currently set to `MESSAGE(body)`. The technology chosen for sending the message is determined based on a prefix to the `to` parameter.

https://wiki.asterisk.org/wiki/display/AST/Asterisk+15+Application_MessageSend

```php
ext_messagesend($to, $from = null)
```

Example

PHP

```php
$ext->add('context', '555', '', ext_messagesend('to', 'from'))
```

extensions_additional

```lua
[context]
exten => 555,1,MessageSend(to,from)
```

return

Return from gosub routine.

Jumps to the last label on the stack, removing it. The return `value`, if any, is saved in the channel variable `GOSUB_RETVAL`.

https://wiki.asterisk.org/wiki/display/AST/Asterisk+15+Application_Return

```lua
ext_return($data = '')
```

stackpop

Remove one address from gosub stack.

Removes last label on the stack, discarding it.

https://wiki.asterisk.org/wiki/display/AST/Asterisk+15+Application_StackPop

```lua
ext_stackpop
```
Conditionally jump to label, saving return address.

If the condition is true, then jump to labeliftrue. If false, jumps to labeliffalse, if specified. In either case, a jump saves the return point in the dialplan, to be returned to with a Return.

https://wiki.asterisk.org/wiki/display/AST/Asterisk+15+Application_GosubIf

```plaintext
ext_gosubif($condition, $true_priority, $false_priority = false, $true_args = '', $false_args = '') {
```

**stasis**

Invoke an external Stasis application.

Invoke a Stasis application.

https://wiki.asterisk.org/wiki/display/AST/Asterisk+15+Application_Stasis

```plaintext
ext_stasis($app_name, $args='')
```

**goto**

Jump to a particular priority, extension, or context.

This application will set the current context, extension, and priority in the channel structure. After it completes, the pbx engine will continue dialplan execution at the specified location. If no specific extension, or extension and context, are specified, then this application will just set the specified priority of the current extension.

At least a priority is required as an argument, or the goto will return a -1, and the channel and call will be terminated.

If the location that is put into the channel information is bogus, and asterisk cannot find that location in the dialplan, then the execution engine will try to find and execute the code in the i (invalid) extension in the current context. If that does not exist, it will try to execute the h extension. If neither the h nor i extensions have been defined, the channel is hung up, and the execution of instructions on the channel is terminated. What this means is that, for example, you specify a context that does not exist, then it will not be possible to find the h or i extensions, and the call will terminate!

https://wiki.asterisk.org/wiki/display/AST/Asterisk+15+Application_Goto

```plaintext
ext_goto ($pri, $ext = false, $context = false)
```

**gotoif**

Conditional goto.

This application will set the current context, extension, and priority in the channel structure based on the evaluation of the given condition. After this application completes, the pbx engine will continue dialplan execution at the specified location in the dialplan. The labels are specified with the same syntax as used within the Goto application. If the label chosen by the condition is omitted, no jump is performed, and the execution passes to the next instruction. If the target location is bogus, and does not exist, the execution engine will try to find and execute the code in the i (invalid) extension in the current context. If that does not exist, it will try to execute the h extension. If neither the h nor i extensions have been defined, the channel is hung up, and the execution of instructions on the channel is terminated. Remember that this command can set the current context, and if the context specified does not exist, then it will not be able to find any 'h' or 'i' extensions there, and the channel and call will both be terminated!

https://wiki.asterisk.org/wiki/display/AST/Asterisk+15+Application_GotoIf

```plaintext
ext_gotoif($condition, $true_priority, $false_priority = false)
```
**gotoiftime**

Conditional Goto based on the current time.

This application will set the context, extension, and priority in the channel structure based on the evaluation of the given time specification. After this application completes, the pbx engine will continue dialplan execution at the specified location in the dialplan. If the current time is within the given time specification, the channel will continue at `labeltrue`. Otherwise the channel will continue at `labelfalse`. If the label chosen by the condition is omitted, no jump is performed, and execution passes to the next instruction. If the target jump location is bogus, the same actions would be taken as for `Goto`. Further information on the time specification can be found in examples illustrating how to do time-based context includes in the dialplan.

https://wiki.asterisk.org/wiki/display/AST/Asterisk+15+Application_GotoIfTime

```plaintext
ext_gotoiftime($condition, $true_priority)
```

**while**

```
ext_while($data = '')
```

**endwhile**

```
ext endwhile()
```

**exitwhile**

```
ext_exitwhile
```

```
ext_continuewhile
```

```
ext_dumpchan
```

```
ext_noop($data = '')
```

```
ext_noop_trace($string, $level=3)
```

```
ext_dial($number, $options = "tr")
```

```
ext_originate($tech_data, $type, $arg1, $arg2, $arg3 = '')
```
ext_setvar($var, $value = '')

documented

eext_setglobalvar($var, $value)

documented

eext_sipaddheader($header, $value)

documented

eext_sipgetheader($value, $header)

documented

eext_sipremoveheader($data = '')

documented

eext_alertinfo($value)

documented

eext_wait($data = '')

documented

eext_parkedcall($data = '')

documented

eext_parkandannounce($data = '')

documented

eext_park($data = '')

documented

eext_resetcdr($data = '')

documented

eext_nocdr()

documented

eext_forkcdr()

documented

eext_waitexten()
ext_answer

ext_macro($macro, $args="")

ext_execif($expr, $app_true, $data_true='', $app_false = '', $data_false = '')[/n]

ext_setcidname($data = '')

ext_setcallerpres($data = '')

ext_setcallernamelpres($data = '')

ext_setcallernumpres($data = '')

ext_record($data = '')

ext_playback($data = '')

ext_queue($queue, $options, $optionalurl, $announceoverride, $timeout, $agi='', $macro='', $gosub='', $rule='', $position='[/n]

ext_queuelog($queue, $uniqueid, $agent, $event, $additionalinfo = '')

ext_addqueuemember($queue, $channel)

ext_removequeuemember($queue, $channel)

ext_userevent($eventname, $body="")
ext_macroexit()

extHangup

ext_digittimeout($data = '')

ext_responsetimeout($data = '')

ext_background($data = '')

ext_read($astvar, $filename='', $maxdigits='', $option='', $attempts ='', $timeout = '')

ext_confbridge($confno, $options='', $pin=''

ext_meetmeadmin($confno, $command, $user=''

ext_meetme($confno, $options='', $pin=''

ext_authenticate($pass, $options=''

ext_vmauthenticate($mailbox='', $options=''

ext_page($data = '')

ext_disa($data = '')

ext_agi($data = '')
ext_deadagi($data = '')

ext_dbdel($data = '')

ext_dbdeltree($data = '')

ext_dbget($varname, $key)

ext_dbput($key, $data)

ext_vmmain($data = '')

ext_vm($data = '')

ext_vmexists($data = '')

ext_sayunixtime($data = '')

ext_echo($data = '')

ext_nvfaxdetect($data = '')

ext_receivefax($data = '')

ext_rxfax($data = '')

ext_sendfax($data = '')
ext_playtones($data = '')

ext_stopplaytones

ext_sayalpha($data = '')

ext_saynumber($data, $gender = 'f')

ext_sayphonetic($data = '')

ext_senddtmf($digits)

ext_system($data = '')

ext_festival($data = '')

ext_pickup($data = '')

ext_dpickup($data = '')

ext_lookupcidname

ext_txtcidname($cidnum)

ext_mysql_connect($connid, $dbhost, $dbuser, $dbpass, $dbname, $charset=''

ext_mysql_query($resultid, $connid, $query)
ext_mysql_fetch($fetchid, $resultid, $vars)

ext_mysql_clear($resultid)

ext_mysql_disconnect($connid)

ext_db_put($family, $key, $value)

ext_ringing

ext_zapateller($data = '')

ext_congestion($time = '20')

ext_busy($time = '20')

ext_flite($data = '')

ext_chanspy($prefix = '', $options = '')

ext_lookupblacklist($data = '')

ext_dictate($data = '')

ext_chanisavail($chan, $options = '')

ext_setlanguage($data = '')
ext_mixmonitor($file, $options = "", $postcommand = "")

ext_stopmonitor($data = '')

ext_callcompletionrequest($data = '')

ext_callcompletioncancel($data = '')

ext_tryexec($try_application = '')

ext_speechcreate($engine = null)

ext_speechloadgrammar($grammar_name, $path_to_grammar)

ext_speechunloadgrammar($grammar_name)

ext_speechactivategrammar($grammar_name)

ext_speechstart

ext_speechbackground($sound_file, $timeout=null)

ext_speechdeactivategrammar($grammar_name)

ext_backgrounddetect($filename, $silence=null, $min=null, $max=null)
ext_speechprocessingsound($sound_file)

ext_speechdestroy

ext_speechdtmfmaxdigits($digits)

ext_speechdtmfterminator($terminator)

ext_progress

ext_vqa($data = '')

ext_transfer($number)

ext_log($level,$msg)

ext_startmusiconhold($data = '')

ext_stopmusiconhold