

Regular Expression for TelePresence Video Communication Server

Presenter:

Date:

Why Regex?

- Regex stand for “Regular Expressions” which is standard.
- Provide a concise and flexible means for matching and transforming strings

Alias	
Alias	★ 9000@tandberg.com

Pattern	
Pattern	★ (.*@tandberg.com
Type	Regex <input type="button" value="i"/>
Behavior	Replace <input type="button" value="i"/>
Replace	\1

Result	
Result	Succeeded
Details	Alias matched pattern and was successfully transformed
Transformed alias	9000

Alias	
Alias	★ tomo.0081.movi@tandberg.net

Pattern	
Pattern	★ (.*)(\.[0-9]{3})\.((?=movi)[a-z]{4})@tandberg.(com net)
Type	Regex <input type="button" value="i"/>
Behavior	Replace <input type="button" value="i"/>
Replace	(\1)(\2)9(\3)(\4)(\5)@tandberg.int

Result	
Result	Succeeded
Details	Alias matched pattern and was successfully transformed
Transformed alias	tomo.9081.movi@tandberg.int

- Used simply it is simple – but complex extensions also available
- Regex is a simple programming language, there can be more than one right answer!

Regex Expressions

- Basic expressions

Parameter	Description
.	Any single character
\d	A digit (numeric character), may works same way by writing [0-9]
*	0 or more repetitions of previous character or expression
+	1 or more repetitions of previous character or expression
?	0 or 1 repetitions of previous character or expression
{n}	n repetitions of previous character or expression
\	Literalise, e.g. \ really is *
	'or' – match, e.g. (wxy wyx) mean wxy or wyx
^	Start of line
\$	End of line
[abc]	A character from this set of characters
[1-9]	A character from this range of characters
[^def]	A character NOT including these characters
()	Group digits and store in store id \n

Regex Expressions

- Extend expressions

Parameter	Description
(?!xxx)	Non-absorbing RegEx, string is NOT xxx, e.g (?!abc) mean 3 characters are not “abc”
(?=xxx)	Non-absorbing RegEx, string is xxx, e.g (?=abc) mean 3 characters are “abc”
%localdomains%	Matches all SIP domains currently configured on VCS
%localdomain#%	Matches the specified SIP domain currently configured on VCS. Up to 20 SIP domains can be configured on the VCS and they are identified by an index number between 1 and 20,
%localdomain1%	Matches 1st SIP domain currently configured on VCS.
%localdomain20%	Matches 20th SIP domain currently configured on VCS.
(?!.*@%localdomains%).*	Matches all URL with domains that currently NOT configured on VCS

Regex Expressions

- Extend expressions for future enhancement (not yet support)

Parameter	Description
%ip%	Matches all IPv4 and IPv6 addresses currently configured on VCS
%ipv4%	Matches all IPv4 addresses currently configured for LAN 1 and LAN 2 on VCS
%ipv4_1%	Matches all IPv4 addresses currently configured for LAN 1 on VCS
%ipv4_2%	Matches all IPv4 addresses currently configured for LAN 2 on VCS
%ipv6%	Matches all IPv6 addresses currently configured for LAN 1 and LAN 2 on VCS
%ipv6_1%	Matches all IPv6 addresses currently configured for LAN 1 on VCS
%ipv6_2%	Matches all IPv6 addresses currently configured for LAN 2 on VCS
%systemname%	Matches the VCS's System name

Regex – some to try 😊

- Add domain to E164 number

`(\d*)` → `\1@tandberg.com`

- Remove a domain

`(.*)@.+` → `\1`

- Add a prefix '01189' to a 6 digit number

`(\d{6})` → `01189\1`

- Reverse the order of 3 digits and put a dot between each

`(\d)(\d)(\d)` → `\3.\2.\1`

- Match either

`123@tandberg.com` or `123@tandberg.net` → `123@tandberg\.(net|com)`

Basic Sample

- Add domain to E.164 Alias number

[Example of configuration](#)
(\d*) → \1@tandberg.com

Alias	
Alias	* 1000
Pattern	
Pattern	* (\d*)
Type	Regex <input type="button" value="i"/>
Behavior	Replace <input type="button" value="i"/>
Replace	\1@tandberg.com
Result	
Result	Succeeded
Details	Alias matched pattern and was successfully transformed
Transformed alias	1000@tandberg.com

Alias	
Alias	* demo
Pattern	
Pattern	* (\d*)
Type	Regex <input type="button" value="i"/>
Behavior	Replace <input type="button" value="i"/>
Replace	\1@tandberg.com
Result	
Result	Failed
Details	Alias did not match pattern

Does not match pattern since called ID is not numeric characters

Basic Sample

- Remove domain

Example of configuration

`(.*)@.+ → \1`

Alias	
Alias	* demo@tandberg.com
Pattern	
Pattern	* (.*)@.+
Type	Regex <input type="button" value="i"/>
Behavior	Replace <input type="button" value="i"/>
Replace	\1
Result	
Result	Succeeded
Details	Alias matched pattern and was successfully transformed
Transformed alias	demo

Alias	
Alias	* demo123
Pattern	
Pattern	* (.*)@.+
Type	Regex <input type="button" value="i"/>
Behavior	Replace <input type="button" value="i"/>
Replace	\1
Result	
Result	Failed
Details	Alias did not match pattern

Does not match pattern since called ID does not contain domain strings

Basic Sample

- Add prefix “0065” to 6 digit number for GW call

Example of configuration
(\d{6}) → 0065\1

Alias	
Alias	* 123456
Pattern	
Pattern	* (\d{6})
Type	Regex <input type="button" value="i"/>
Behavior	Replace <input type="button" value="i"/>
Replace	0065\1
Result	
Result	Succeeded
Details	Alias matched pattern and was successfully transformed
Transformed alias	0065123456

Alias	
Alias	* 1234
Pattern	
Pattern	* (\d{6})
Type	Regex <input type="button" value="i"/>
Behavior	Replace <input type="button" value="i"/>
Replace	0065\1
Result	
Result	Failed
Details	Alias did not match pattern

Does not match pattern since called ID is not 6 digit number

Basic Sample

- Dial IP address system without type “.”

Example of configuration

$(\d{3})(\d{3})(\d{3})(\d{3}) \rightarrow \backslash1\.\backslash2\.\backslash3\.\backslash4$

Alias	
Alias	* 010081150120
Pattern	
Pattern	* $(\d{3})(\d{3})(\d{3})(\d{3})$
Type	Regex <input type="button" value="i"/>
Behavior	Replace <input type="button" value="i"/>
Replace	\1\.\2\.\3\.\4
Result	
Result	Succeeded
Details	Alias matched pattern and was successfully transformed
Transformed alias	010.081.150.120

Alias	
Alias	* 1081150120
Pattern	
Pattern	* $(\d{3})(\d{3})(\d{3})(\d{3})$
Type	Regex <input type="button" value="i"/>
Behavior	Replace <input type="button" value="i"/>
Replace	\1\.\2\.\3\.\4
Result	
Result	Failed
Details	Alias did not match pattern

Does not match pattern since called ID is not 12 digit number strings

Basic Sample

- Replace domain to “@tandberg.int”, if domain in called ID is @tandberg.com or @tandberg.net

[Example of configuration](#)

`(.*)@tandberg.(com|net) → \1@tandberg.int`

Alias	
Alias	<input type="text" value="demo@tandberg.com"/>
Pattern	
Pattern	<input type="text" value="(.*)@tandberg.(com net)"/>
Type	Regex <input type="button" value="i"/>
Behavior	Replace <input type="button" value="i"/>
Replace	<input type="text" value="\1@tandberg.int"/>
Result	
Result	Succeeded
Details	Alias matched pattern and was successfully transformed
Transformed alias	demo@tandberg.int

Alias	
Alias	<input type="text" value="demo@tandberg.jp"/>
Pattern	
Pattern	<input type="text" value="(.*)@tandberg.(com net)"/>
Type	Regex <input type="button" value="i"/>
Behavior	Replace <input type="button" value="i"/>
Replace	<input type="text" value="\1@tandberg.int"/>
Result	
Result	Failed
Details	Alias did not match pattern

Does not match pattern since domain in called ID is not tandberg.com nor tandberg.net

Basic Sample

- Add domain, if called ID is 4 digits numeric number and NOT start with 0, 1, or 2

Example of configuration

`([^012])(\d{3})` → `\1\2@tandberg.com` or
`([^0-2])(\d{3})` → `\1\2@tandberg.com`

Alias	
Alias	<input type="text" value="3000"/>
Pattern	
Pattern	<input type="text" value="([^012])(\d{3})"/>
Type	Regex <input type="button" value="i"/>
Behavior	Replace <input type="button" value="i"/>
Replace	<input type="text" value="\1\2@tandberg.com"/>
<input type="button" value="Check pattern"/>	
Result	
Result	Succeeded
Details	Alias matched pattern and was successfully transformed
Transformed alias	3000@tandberg.com

Alias	
Alias	<input type="text" value="1000"/>
Pattern	
Pattern	<input type="text" value="([^012])(\d{3})"/>
Type	Regex <input type="button" value="i"/>
Behavior	Replace <input type="button" value="i"/>
Replace	<input type="text" value="\1\2@tandberg.com"/>
Result	
Result	Failed
Details	Alias did not match pattern

Does not match pattern since called ID starting with "1".

Basic Sample

- Process if Called ID is NOT contain “@tandberg.net” strings

Example of configuration

.*(?<@tandberg.net\$) → Leave
(?!.*@tandberg.net\$).* → Leave

or

Alias	
Alias	* demo@tandberg.com
Pattern	
Pattern	* .*(?<@tandberg.net\$)
Type	Regex <input type="button" value="i"/>
Behavior	Leave <input type="button" value="i"/>
Result	
Result	Succeeded
Details	Alias matched pattern

Alias	
Alias	* demo@tandberg.net
Pattern	
Pattern	* .*(?<@tandberg.net\$)
Type	Regex <input type="button" value="i"/>
Behavior	Leave <input type="button" value="i"/>
Result	
Result	Failed
Details	Alias did not match pattern

Does not match pattern since called ID starting with “@tandberg.net”.

Basic Sample

- Use last 2 digit number and add prefix 908100, if call strings contain “jpn”

Example of configuration

`([^\+]{2})+\.[^\.]+\.(?=jpn)[a-z]{3} → 908100\1`

Alias	
Alias	* 8162303525.office.jpn
Pattern	
Pattern	* <code>([^\+]{2})+\.[^\.]+\.(?=jpn)[a-z]{3}</code>
Type	Regex <input type="button" value="i"/>
Behavior	Replace <input type="button" value="i"/>
Replace	908100\1
Result	
Result	Succeeded
Details	Alias matched pattern and was successfully transformed
Transformed alias	90810025

Alias	
Alias	* 8162303525.office.anz
Pattern	
Pattern	* <code>([^\+]{2})+\.[^\.]+\.(?=jpn)[a-z]{3}</code>
Type	Regex <input type="button" value="i"/>
Behavior	Replace <input type="button" value="i"/>
Replace	908100\1
Result	
Result	Failed
Details	Alias did not match pattern

Does not match pattern since called ID does not have “jpn” starting

Non-absorbing expressions

- Non absorbing expressions – Lookbehind
 - ✓ Checks characters but does not remove them from match string
 - ✓ The rest of the Regex must work, this expression will just ensure that certain characters are or are not as specified.
- (?<!jpn) - three characters are not jpn(VCS is not case sensitive)
 - ✓ Format name.surname.3 characters country@cisco.com
`[^\.]+\.[^\.]+\.[a-z]{3}(?<!jpn)@cisco\.com`
- (?<=jpn) - three characters are jpn
 - ✓ Format name.surname.3 char country@cisco.com
`[^\.]+\.[^\.]+\.[a-z]{3}(?<=jpn)@cisco\.com`
 - ✓ Can be write in this way as well
`[^\.]+\.[^\.]+\.[a-z]{3}jpn@cisco\.com`

Non-absorbing expressions

- `(?!chn)` - 3 characters not chn
- `(?=chn)` - 3 characters are chn
- ******* same function different way of representing it *******
`[^\.]+\.[^\.]+\.(?!chn)[a-z]{3}@tandberg\.com`
`[^\.]+\.[^\.]+\.(?=chn)[a-z]{3}@tandberg\.com`
- ... from the Admin Guide ... `(?!.*@tandberg.net$).*`
Matches anything which does not end with `@tandberg.net`
Same as `.*(?!.*@tandberg.net$)`

Exercise

- Handling SIP and H.323 endpoints dialling one another
 - ✓ For routing use URI format, so need to convert E164 to URI
All destinations dialled from SIP, H323 URI or H323 E164 result in a URI
 - ✓ For local zone route URI, then e164
Routes to SIP or URI registered endpoint first, then to E164 registered endpoint
- What transform(s) & local zone match(es) / transform(s) are required?

Exercise

- Transform

(\d+) → \1@cisco.com

or ...

([^\@]+) → \1@cisco.com

- Local zone URI

(.+@cisco\.com) → Leave

- Local zone E164

(.+@cisco\.com) → \1

Further reading

- www.cisco.com VCS Support
 - Read the release notes and administrator guide for any special instructions.



Question?



