

BGP: Regular Expressions for AS-PATH Filtering

Symbol	Description
	logical OR
.	match any
[x..y]	match one in range
^	match beginning of string
\$	match end of string
_	match any delimiter/white space
()	group as a single atom
*	match 0 or more instances of previous atom
?	match 0 or 1 instance of previous atom
+	match 1 or more instances of previous atom
\	escape character; if followed by a number <i>n</i> , points to the <i>nth</i> atom

Example 1: Advertise routes with empty AS-PATH (internal routes)

```
router bgp 123
neighbor 5.6.7.8 remote-as 387
neighbor 5.6.7.8 filter-list 1 out
!
ip as-path access-list 1 permit ^$
!!match "blank" atom at the "start" and "end" of string
!
```

Example 2: Accept only default routes, preferring primary route based on AS PATH:

```
AS387 (primary ISP)
/
AS123
\  
AS462 (backup ISP)
```

```
router bgp 123
neighbor 1.2.3.4 remote-as 462
neighbor 1.2.3.4 route-map FILTER in
neighbor 5.6.7.8 remote-as 387
neighbor 5.6.7.8 route-map FILTER in
!
route-map FILTER permit 10
!! default routes from primary ISP (AS 387) accepted are preferred (larger weight)
```

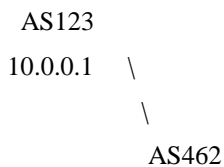
```

match ip prefix-list DEFAULT_ONLY
match as-path 10
set weight 150
!
route-map FILTER permit 20
! default routes from backup ISP are accepted, with lower preference than primary ISP routes
match ip prefix-list DEFAULT_ONLY
set weight 100
!
ip as-path access-list 10 permit _387$
ip prefix-list DEFAULT_ONLY seq 10 permit 0.0.0.0/0
!

```

Example 3: AS PATH Filtering with AS Path Prepending:

- customer in AS123 is performing AS-PATH pre-pending



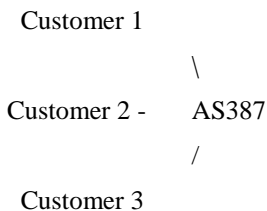
```

router bgp 387
neighbor 10.0.0.1 remote-as 213
neighbor 10.0.0.1 filter-list 10 in
!
ip as-path access-list 10 permit ^123(_123)*$
! accepts "123", "123 123", or "123 123 123"

```

Example 4: AS PATH Filtering with AS Path Prepending, multiple customers:

- multiple customers performing AS PATH Prepending



```

!
router bgp 387
neighbor 10.0.0.1 remote-as 123
neighbor 10.0.0.1 filter-list 10 in
neighbor 20.0.0.1 remote-as 456

```

```
neighbor 20.0.0.1 filter-list 10 in
neighbor 30.0.0.1 remote-as 789
neighbor 30.0.0.1 filter-list 10 in
!
ip as-path access-list 10 permit ^([0-9]+)(_1)*$
! accepts repeating instances of "123", "456", and "789"
! does not accept strings non-repeating strings (e.g. "123 123 100")
!
```

Notes:

atom 1 = at least one instance of a number at the beginning of the string;

atom 2 = 0 or more instances of a whitespace *and* atom 1 until the end of the string.