

# Como eu escrevo filtros da mensagem dos mais eficiente?

## Índice

### [Pergunta](#)

## Pergunta

Como eu escrevo filtros da mensagem dos mais eficiente?

Enquanto os filtros da mensagem obtêm mais por muito tempo, podem afetar as características de desempenho de seu ESA. Para um pequeno número de filtros ou filtros curtos, a eficiência não é um interesse significativo. Contudo, ao construir uns filtros mais longos ou se sua aplicação tem muitos filtros, você deve ser consciente da eficiência relativa de determinadas operações.

Ao passar mensagens através do encaimento da mensagem, todos os filtros da mensagem são combinados em uma única expressão que seja avaliada em uma maneira atômica contra cada mensagem. Isto significa que pedir dos filtros é muito importante, e pode procurar um caminho mais curto uma avaliação mais adicional da expressão combinada. Por exemplo, se você tem um número de filtros que se aplicarão às mensagens, mas um filtro aplicar-se-á muito frequentemente e ter-se-á um deliver() da ação final, um bounce(), ou um drop() associado com ele, que o filtro deve ser movido tão cedo na lista como possível.

Embora o ESA seja muito eficiente no seu processamento das expressões regulares, você pode abusar o motor da expressão regular de modo que causar o processamento adicional ou desnecessário. Cada avaliação de uma expressão regular toma aproximadamente o mesmo valor dos recursos, assim que significa que aquela reduzir o número de expressões que você avalia renderá a maior eficiência. Por exemplo, no seguinte filtro, todas as expressões regulares em cada "gota-acessório-por-nome" são avaliadas individualmente, significando que a avaliação da expressão regular ocorre as épocas 7 ao comparar o nome do acessório contra o teste padrão no gota-acessório-por-nome:

```
strip_all_dangerous: se (verdadeiro) {
gota-acessório-por-nome "(? i) \\ .pif$";
gota-acessório-por-nome "(? i) \\ .exe$";
gota-acessório-por-nome "(? i) \\ .scr$";
gota-acessório-por-nome "(? i) \\ .msi$";
gota-acessório-por-nome "(? i) \\ .java$";
gota-acessório-por-nome "(? i) \\ .dll$";
gota-acessório-por-nome "(? i) \\ .com$";
}
```

No exemplo seguinte, os resultados são equivalentes, mas o exemplo é muitos mais eficiente, causando somente uma única avaliação da expressão regular:

```
strip_all_dangerous: se (verdadeiro) {
gota-acessório-por-nome "(? i) \\ . (pif|exe|SCR|msi|Javas|DLL|COM) $";
```

}

Embora a segunda expressão regular seja mais complexa do que sete no primeiro filtro, é muito mais eficiente para avaliar uma expressão regular complexa do que sete simples.

Contudo, esta técnica deve ser equilibrada contra o custo de manter tal filtro.