

Sender Policy Framework SPF Record Syntax

Note: This page serves as an introduction and quick overview of SPF mechanism syntax. For the complete and definitive picture, please see the specification.

Domains define zero or more **mechanisms**. Mechanisms can be used to describe the set of hosts which are designated outbound mailers for the domain.

```
all | ip4 | ip6 | a | mx | ptr | exists | include
```

Domains may also define **modifiers**. Each modifier can appear only once.

```
redirect | exp
```

Mechanisms

Mechanisms can be prefixed with one of four qualifiers:

"+" Pass

"-" Fail

"~" SoftFail

"?" Neutral

If a mechanism results in a hit, its qualifier value is used. The default qualifier is "+", i.e. "Pass". For example:

```
"v=spf1 -all"  
"v=spf1 a -all"  
"v=spf1 a mx -all"  
"v=spf1 +a +mx -all"
```

Mechanisms are evaluated in order. If no mechanism or modifier matches, the default result is "Neutral".

If a domain has no SPF record at all, the result is "None". If a domain has a temporary error during DNS processing, you get the result "TempError" (called "error" in earlier drafts). If some kind of syntax or evaluation error occurs (eg. the domain specifies an unrecognized mechanism) the result is "PermError" (formerly "unknown").

Evaluation of an SPF record can return any of these results:

Result	Explanation	Intended action
Pass	The SPF record designates the host to be allowed to send	accept
Fail	The SPF record has designated the host as NOT being allowed to send	reject

Sender Policy Framework

SPF Record Syntax

SoftFail	The SPF record has designated the host as NOT being allowed to send but is in transition	accept but mark
Neutral	The SPF record specifies explicitly that nothing can be said about validity	accept
None	The domain does not have an SPF record or the SPF record does not evaluate to a result	accept
PermError	A permanent error has occurred (eg. badly formatted SPF record)	unspecified
TempError	A transient error has occurred	accept or reject

The "all" mechanism

all

This mechanism always matches. It usually goes at the end of the SPF record.

Examples:

```
"v=spf1 mx -all"
```

Allow domain's MXes to send mail for the domain, prohibit all others.

```
"v=spf1 -all"
```

The domain sends no mail at all.

```
"v=spf1 +all"
```

The domain owner thinks that SPF is useless and/or doesn't care.

The "ip4" mechanism

```
ip4:<ip4-address>
```

```
ip4:<ip4-network>/<prefix-length>
```

The argument to the "ip4:" mechanism is an IPv4 network range. If no *prefix-length* is given, /32 is assumed (singling out an individual host address).

Examples:

```
"v=spf1 ip4:192.168.0.1/16 -all"
```

Allow any IP address between 192.168.0.1 and 192.168.255.255.

The "ip6" mechanism

Sender Policy Framework

SPF Record Syntax

```
ip6:<ip6-address>  
ip6:<ip6-network>/<prefix-length>
```

The argument to the "ip6:" mechanism is an IPv6 network range. If no *prefix-length* is given, /128 is assumed (singling out an individual host address).

Examples:

```
"v=spf1 ip6:1080::8:800:200C:417A/96 -all"
```

Allow any IPv6 address between 1080::8:800:0000:0000 and 1080::8:800:FFFF:FFFF.

```
"v=spf1 ip6:1080::8:800:68.0.3.1/96 -all"
```

Allow any IPv6 address between 1080::8:800:0000:0000 and 1080::8:800:FFFF:FFFF.

The "a" mechanism

```
a  
a/<prefix-length>  
a:<domain>  
a:<domain>/<prefix-length>
```

All the A records for *domain* are tested. If the client IP is found among them, this mechanism matches.

If *domain* is not specified, the *current-domain* is used.

The A records have to match the client IP exactly, unless a *prefix-length* is provided, in which case each IP address returned by the A lookup will be expanded to its corresponding CIDR prefix, and the client IP will be sought within that subnet.

```
"v=spf1 a -all"
```

The *current-domain* is used.

```
"v=spf1 a:example.com -all"
```

Equivalent if the *current-domain* is example.com.

```
"v=spf1 a:mailers.example.com -all"
```

Perhaps example.com has chosen to explicitly list all the outbound mailers in a special A record under mailers.example.com.

```
"v=spf1 a/24 a:offsite.example.com/24 -all"
```

If example.com resolves to 192.0.2.1, the entire class C of 192.0.2.0/24 would be searched for the client IP. Similarly for offsite.example.com. If more than one A record were returned, each one would be expanded to a CIDR subnet.

The "mx" mechanism

Sender Policy Framework SPF Record Syntax

```
mx
mx/<prefix-length>
mx:<domain>
mx:<domain>/<prefix-length>
```

All the A records for all the MX records for *domain* are tested in order of MX priority. If the client IP is found among them, this mechanism matches.

If *domain* is not specified, the *current-domain* is used.

The A records have to match the client IP exactly, unless a prefix-length is provided, in which case each IP address returned by the A lookup will be expanded to its corresponding CIDR prefix, and the client IP will be sought within that subnet.

Examples:

```
"v=spf1 mx mx:deferrals.domain.com -all"
```

Perhaps a domain sends mail through its MX servers plus another set of servers whose job is to retry mail for deferring domains.

```
"v=spf1 mx/24 mx:offsite.domain.com/24 -all"
```

Perhaps a domain's MX servers receive mail on one IP address, but send mail on a different but nearby IP address.

The "ptr" mechanism

```
ptr
ptr:<domain>
```

The hostname or hostnames for the client IP are looked up using PTR queries. The hostnames are then validated: at least one of the A records for a PTR hostname must match the original client IP. Invalid hostnames are discarded. If a valid hostname ends in domain, this mechanism matches.

If *domain* is not specified, the *current-domain* is used.

If at all possible, you should avoid using this mechanism in your SPF record, because it will result in a larger number of expensive DNS lookups.

Examples:

```
"v=spf1 ptr -all"
```

A domain which directly controls all its machines (unlike a dialup or broadband ISP) allows all its servers to send mail. For example, hotmail.com or paypal.com might do this.

```
"v=spf1 ptr:otherdomain.com -all"
```

Any server whose hostname ends in otherdomain.com is designated.

Sender Policy Framework

SPF Record Syntax

The "exists" mechanism

```
exists:<domain>
```

Perform an A query on the provided domain. If a result is found, this constitutes a match. It doesn't matter what the lookup result is – it could be 127.0.0.2.

When you use macros with this mechanism, you can perform RBL-style reversed-IP lookups, or set up per-user exceptions.

Examples:

In the following example, the client IP is 1.2.3.4 and the *current-domain* is example.com.

```
"v=spf1 exists:example.com -all"
```

If example.com does not resolve, the result is fail. If it does resolve, this mechanism results in a match.

The "include" mechanism

```
include:<domain>
```

The specified *domain* is searched for a match. If the lookup does not return a match or an error, processing proceeds to the next directive. **Warning:** If the *domain* does not have a valid SPF record, the result is a permanent error. Some mail receivers will reject based on a *PermError*.

Examples:

In the following example, the client IP is 1.2.3.4 and the *current-domain* is example.com.

```
"v=spf1 include:example.com -all"
```

If example.com has no SPF record, the result is *PermError*.

Suppose example.com's SPF record were "v=spf1 a -all".

Look up the A record for example.com. If it matches 1.2.3.4, return *Pass*.

If there is no match, other than the included domain's "-all", the include as a whole fails to match; the eventual result is still *Fail* from the outer directive set in this example.

Trust relationships — The "include:" mechanism is meant to cross administrative boundaries. Great care is needed to ensure that "include:" mechanisms do not place domains at risk for giving SPF *Pass* results to messages that result from cross user forgery. Unless technical mechanisms are in place at the specified otherdomain to prevent cross user forgery, "include:" mechanisms should give a *Neutral* rather than *Pass* result. This is done by adding "?" in front of "include:". The example above would be:

```
"v=spf1 ?include:example.com -all"
```

In hindsight, the name "include" was poorly chosen. Only the evaluated result of the referenced SPF record is used, rather than acting as if the referenced SPF record was literally included in the first. For example, evaluating a "-all" directive in the referenced record does not terminate the overall processing and does not necessarily result in an overall *Fail*. (Better names for this mechanism would have been "if-pass", "on-pass", etc.)

Sender Policy Framework SPF Record Syntax

Modifiers

Modifiers are optional. A modifier may appear only once per record. Unknown modifiers are ignored.

The "redirect" modifier

```
redirect=<domain>
```

The SPF record for *domain* replace the current record. The macro-expanded *domain* is also substituted for the *current-domain* in those look-ups.

Examples:

In the following example, the client IP is 1.2.3.4 and the *current-domain* is example.com.

```
"v=spf1 redirect=example.com"
```

If example.com has no SPF record, that is an error; the result is unknown.
Suppose example.com's SPF record was "v=spf1 a -all".
Look up the A record for example.com. If it matches 1.2.3.4, return *Pass*.
If there is no match, the exec fails to match, and the -all value is used.

The "exp" modifier

```
exp=<domain>
```

If an SMTP receiver rejects a message, it can include an explanation. An SPF publisher can specify the explanation string that senders see. This way, an ISP can direct nonconforming users to a web page that provides further instructions about how to configure SASL.

The *domain* is expanded; a TXT lookup is performed. The result of the TXT query is then macro-expanded and shown to the sender. Other macros can be used to provide an customized explanation.