The Trust Router configuration format (v1.0)

This page is for the Trust Router v1.x trusts.cfg format. For the Trust Router v2.x/3.x format, please see a page still under construction.

The Trust Router trusts.cfg file is in JSON format, and for processing and automation reasons, the format generated by the Moonshot portal lists each delimiter (square and curly brackets) on a separate lines.

This page is designed to make the file easier to read and understand.

The top level

The top level of the trusts.cfg file defines three lists of entities, which are the communities in this trust infrastructure, the Identity Provider (IdP) realms and the Relying Party (RP) clients. The latter list defines groups of RPs that share sets of APC credentials.

The top level

```
{
  "communities": [
    {community1},
    {community2}, ...
  ],
  "idp_realms": [
    {idp_realm1},
    {idp_realm2}, ...
  ],
  "rp_clients": [
    {rp_client_group1},
    {rp_client_group2}, ...
  ],
  "default_servers": [
    "host1", ...
  ]
}
```

Communities

The **`communities`** list contains the communities in this trust infrastructure in alphabetical order by `community_id`. There is always a **minimum of one** community in a trust infrastructure, the Authentication Policy Community (APC). It is the over-arching community that includes all RPs and IdPs.

community

```
{
  "apcs": [ "community_id of APC" | empty ],
  "community_id": "name of the community",
  "idp_realms": [
    "idp_realm1",
    "idp_realm2", ...
  ],
  "rp_realms": [
    "rp_realm1",
    "rp_realm2", ...
  ],
  "type": "apc" | "coi",
  "expiration_interval": number
}
```

- The community ID, `community_id`, must be in FQDN format, i.e. ov-apc.moonshot.ja.net, or csc.communities.moonshot.ja.net
- The APC community has an **empty** `apcs` field and its `type` field is **"apc"**. The APC community also requires an `expiration_interval` to be set.
- Communities of interest (COI) will set the `apcs` field to **"apc"** and their `type` field to **"coi"**, and will have no `expiration_interval` setting.
- The community's `idp_realms` list contains the identity provider realm names that are part of the community.

    - In the case of a COI, each entry **must also** be part of the APC community's `idp_realms` list.
    - Each entry in this list must have an entry in the top-level `idp_realms` list for an `aaa_server` with a matching `realm_id` value.
- The community's `rp_realms` list contains the relying party (RP) realms that are part of the community.
    - In the case of a COI, each entry **must also** be part of the APC community's `rp_realms` list.
    - Each entry must have a corresponding `filter_lines` entry in one of the `rp_clients` groups in the top-level `rp_clients` list
- The expiration_interval value is a figure set in minutes, ranging from 10 to 129600. The default is 30 days (43200 minutes).

A sample APC containing just camford.ac.uk

```
{
  "apcs": [  ],
  "community_id": "ov-apc.moonshot.ja.net",
  "idp_realms": [
    "ov-apc.moonshot.ja.net",
    "camford.ac.uk"
  ],
  "rp_realms": [
    "tr1.moonshot.ja.net",
    "moonshot.camford.ac.uk"
  ],
  "type": "apc",
  "expiration_interval": 10
}
```

A sample COI containing just camford.ac.uk

```
{
  "apcs": [ "ov-apc.moonshot.ja.net" ],
  "community_id": "camford.communities.moonshot.ja.net",
  "idp_realms": [
    "camford.ac.uk"
  ],
  "rp_realms": [
    "moonshot.camford.ac.uk"
  ],
  "type": "coi"
}
```

Identity Provider realms

The identity provider realms list **`idp_realms`** contains a list of entries in alphabetical order by `realm_id` that define the identity realms available in this trust infrastructure. This realm list will include the APC as well, as the APC is not just a community, but also the identity provider for all the relying parties in the trust infrastructure.

Each identity provider realm uses the below format:

idp_realm

```
{
  "aaa_servers": [ "hostname" ],
  "apcs": [ "community_id of APC" ],
  "realm_id": "idp_realm1",
  "shared_config": "yes" | "no"
}
```

- The `aaa_servers` entry must contain a `hostname` that belongs to the organisation that owns (or manages) the realm in `realm_id`.
  This `hostname` must be able to match a corresponding `filter_lines` entry in one of the `rp_clients` groups in the top-level `rp_clients` list.

> ⚠ The aaa_servers entries on the upstream trust router for any idp_realms entries connected to downstream trust routers must point to the trust router they are connected to.
>
> Example: IDP1 is connected to Trust Router B, which is downstream from Trust Router A. On Trust Router A, IDP1's aaa_servers entry must be set to Trust Router B's hostname, while on Trust Router B, the aaa_servers entry for IDP1 points to its real hostname.

- The realm_id **must** be listed in the idp_realms list of at least the APC. You may add it to other communities as well to make that realm available as an ID Provider in those communities.
- The shared_config option is currently not used and should be said to **"no"**.

> ⓘ **A sample aaa_servers entry for camford.ac.uk**
>
> **idp_realm**
>
> ```
> {
>   "aaa_servers": [ "moonshot.camford.ac.uk" ],
>   "apcs": [ "ov-apc.moonshot.ja.net" ],
>   "realm_id": "camford.ac.uk",
>   "shared_config": "no"
> }
> ```

Relying Party client groups

The relying party clients list **rp_clients** contains a list of groups that define the relying party clients available in this trust infrastructure. Relying party (RP) clients authenticated with the same credential in gss_name are grouped together, with each RP client identified by a filter_lines entry. Each RP client group uses the below format:

rp_client

```
{
  "filter": {
      "filter_lines": [
        {filter_line1},
        {filter_line2}, ...
      ],
      "type": "rp_permitted"
  },
  "gss_names": [
    "gss_name1",
    "gss_name2", ...
  ]
}
```

- The gss_names entries are the accepted APC credentials for this group of rp_clients.
- The filter_lines entries in the filter entity are rp_realm entries that will be authenticated with the same gss_name entries.

> ⚠ The rp_realm entries on the upstream trust router for any rp_realm entries connected to downstream trust routers must be grouped together in an rp_clients group that authenticates with the gss_names entry of the downstream trust router.
>
> Example: IDP1 and RP2 are connected to Trust Router B, which is downstream from Trust Router A. On Trust Router A, IDP1 and RP2's filter_lines entries are authenticated using Trust Router B's gss_names entry, while on Trust Router B, each host has its own rp_clients group with its own gss_names entry (where appropriate).

Filter lines (RP client filter definitions)

The `filter_lines` list in an `rp_clients` group contains RP client filters for RP realms that are identified by this `rp_clients` group. Each `filter_line` entry follows the below format:

filter_lines

```
{
  "action": "accept",
  "domain_constraints": [
    "domain_constraint_1", ...
  ],
  "filter_specs": [
      { "field": "rp_realm", "match": "value1_of_rp_realm" },
      { "field": "rp_realm", "match": "value2_of_rp_realm" }, ...
  ],
  "realm_constraints": [
    "value1_of_rp_realm",
    "value2_of_rp_realm", ...
  ]
}
```

- The `domain_constraints` list should at least contain one of the `realm_constraints` entries, but an empty list is acceptable.
- Each entry in the `realm_constraints` list must have a corresponding entry in the `filter_specs` list.
- The bare minimum of such an entry should contain the FQDN name of the RP in the `domain_constraints` and `realm_constraints`, and a corresponding `filter_specs` entry.

Default servers

The default servers list **`default_servers`** contains a list of one or more AAA servers that should be contacted if a TID request is received that this trust router cannot resolve.

This list is used for static peering between trust routers, and it is **optional**. It is sensible to store this list in a separate file.

If it does not exist, the trust router assumes that it is the only or top-level trust router.

default_servers

```
  "default_servers": [
    "host1", ...
  ]
```

An example file:

Here is an example trusts.cfg file. A full description of the various sections follows

Example trusts.cfg

```
{
  "communities": [
    {
      "apcs": [
      ],
      "community_id": "ov-apc.moonshot.ja.net",
      "idp_realms": [
        "ov-apc.moonshot.ja.net",
        "dev.ja.net",
        "ja.net"
      ],
      "rp_realms": [
        "ms-idp.dev.ja.net",
        "ms-idp.ja.net",
        "ms-ssh-sp.dev.ja.net"
      ],
      "type": "apc",
      "expiration_interval": 30
    },
    {
      "apcs": [
        "ov-apc.moonshot.ja.net"
      ],
      "community_id": "pilot.communities.moonshot.ja.net",
      "idp_realms": [
        "dev.ja.net"
      ],
      "rp_realms": [
        "ms-ssh-sp.dev.ja.net"
      ],
      "type": "coi"
    }
  ],
  "idp_realms": [
    {
      "aaa_servers": [
        "ov-apc.moonshot.ja.net"
      ],
      "apcs": [
        "ov-apc.moonshot.ja.net"
      ],
      "realm_id": "ov-apc.moonshot.ja.net",
      "shared_config": "no"
    },
    {
      "aaa_servers": [
        "ms-idp.dev.ja.net"
      ],
      "apcs": [
        "ov-apc.moonshot.ja.net"
      ],
      "realm_id": "dev.ja.net",
      "shared_config": "no"
    },
    {
      "aaa_servers": [
        "ms-idp.ja.net"
      ],
      "apcs": [
        "ov-apc.moonshot.ja.net"
      ],
      "realm_id": "ja.net",
      "shared_config": "no"
    }
  ],
  "rp_clients": [
    {
      "filter": {
        "filter_lines": [
          {
            "action": "accept",
            "domain_constraints": [
            ],
            "filter_specs": [
              {
                "field": "rp_realm",
                "match": "ms-ssh-sp.dev.ja.net"
              },
              {
                "field": "rp_realm",
                "match": "*.ms-ssh-sp.dev.ja.net"
              }
            ],
            "realm_constraints": [
              "ms-ssh-sp.dev.ja.net",
              "*.ms-ssh-sp.dev.ja.net"
            ]
          },
          {
            "action": "accept",
            "domain_constraints": [
              "ms-idp.dev.ja.net"
            ],
            "filter_specs": [
              {
                "field": "rp_realm",
                "match": "ms-idp.dev.ja.net"
              },
              {
                "field": "rp_realm",
                "match": "*.ms-idp.dev.ja.net"
              }
            ],
            "realm_constraints": [
              "ms-idp.dev.ja.net",
              "*.ms-idp.dev.ja.net"
            ]
          }
        ],
        "type": "rp_permitted"
      },
      "gss_names": [
        "e018e5bd-c37b-45d1-b48c-93c92a15aa31@ov-apc.moonshot.ja.net"
      ]
    },
    {
      "filter": {
        "filter_lines": [
          {
            "action": "accept",
            "domain_constraints": [
              "ms-idp.ja.net"
            ],
            "filter_specs": [
              {
                "field": "rp_realm",
                "match": "ms-idp.ja.net"
              },
              {
                "field": "rp_realm",
                "match": "*.ms-idp.ja.net"
              }
            ],
            "realm_constraints": [
              "ms-idp.ja.net",
              "*.ms-idp.ja.net"
            ]
          }
        ],
        "type": "rp_permitted"
      },
      "gss_names": [
        "edc3fa84-4bb7-4df4-b90a-11f807000511@ov-apc.moonshot.ja.net"
      ]
    }
  ]
}
```

The `communities` list

This list starts at line 1 and ends at line 33.

It contains two communities, `ov-apc.moonshot.ja.net` and `pilot.communities.moonshot.ja.net`.

The APC community starts at line 3 and ends at line 19. It contains three `idp_realms` (line 7-11) and three `rp_realms` (line 12-16) entries. The trust keys between ID and RP realms in this APC expire 30 minutes later (line 18)

The pilot COI contains one idp_realms (line 25-27) and one rp_realms (line 28-30) entry. Each of these entries is present in the APC's corresponding lists.

The idp_realms list

This list starts at line 34 and ends at line 65.

It contains three idp_realms entries, ov-apc.moonshot.ja.net (line 35-44), ms-idp.dev.ja.net (line 45-54) and ms-idp.ja.net (line 55-64).

Each entry (excepting ov-apc.moonshot.ja.net) contains a aaa_servers entry that matches one of the rp_realms in lines 12-19, and each entry contains a realm_id entry that matches one of the idp_realms entries in lines 7-11.

The rp_clients list

This list starts at line 66 and ends at line 177.

It contains two RP client groups, one for e018e5bd-c37b-45d1-b48c-93c92a15aa31@ov-apc.moonshot.ja.net (line 70-118) and one for edc3fa84-4bb7-4df4-b90a-11f807000511@ov-apc.moonshot.ja.net (line 119-148).

The first group contains two rp_clients, ms-ssh-sp.dev.ja.net (line 73-91) and ms-idp.dev.ja.net (line 92-111), the other group contains one rp_client (line 122-141).

Each of these rp_clients entries corresponds to one of the rp_realms defined in the APC community.

Each of these rp_clients entries has a filter_specs list entry for each realm_constraints list entry, and the domain_constraints list contains at least one of the realm_constraints.