

GSA connector wrapper POC

- The Repository
 - Simple repository description
 - An overview of the repository item/document types
 - An overview of the repository security model
 - Example of headers containing ACLs
 - Example of a decoded ACL from a document header
 - Example of cleaned Group feed
 - Available Documentation
 - How to setup the repository
- The Connector
 - Connection Layer
 - How it works
 - Running the tool
 - Connection Parameters
 - Item Types
 - Available Metadata
- The Crawler
 - Parameters
 - Feature Grid
 - Known Bugs / Limitations
 - Nice-to-Have Features (from a developer perspective)

CT-2422 - GSA - Support indexation of GSA content through the PUSH Api

Epic: **COMPLETED**

The Repository

Simple repository description

The GSA connectors are the equivalent of Coveo connectors but for GSA (Google Search Appliance). The goal of the POC was to implement a system that would replace the GSA and push the documents to our Cloud V2 using the Push API. This operation should almost be invisible to the GSA connector. For more information on how I had setup myself for testing with a FileShare connector and a Sharepoint connector: [GSA Connector setup](#)

MC+A GSA is available here: <https://devgsa2.mcplusa.com:8443/EnterpriseController#a=home> Login in the GSA folder of Lastpass of connector team.

The project is available in the DFS corp.coveo.com\dfs at \Softwares\Tools\GSAConectorWrapperPOC and the latest version is in the repository of the connector team under the GSA folder.

An overview of the repository item/document types

The items will be different depending on the GSA connector, but the main part of it stays identical. The first element is the web feed containing all the starting addresses. After this, we web crawl all the roots URLs to go through all the pages to get all the items. If some items ACLs are referencing URLs that do not exist, the connector is going to send a feed that only contains the missing ACLs. We also receive group feed for all the groups of the GSA Connector. Example below. [Example of cleaned Group feed](#)

An overview of the repository security model

The ACLs are either directly attach to the document or they have an URL that they cover. The URL can be a valid page, a valid page URL with URL parameters or even a none existing page.

The ACL can have four inheritance type:

- Parent-overrides: We put the parent permissions in a higher level.
- Child-overrides: We put the parent permissions in a lower level.
- And-both-permit: We put the parent and the child the same level but in different sets.
- Leaf-node: The Acl should not have children, the parser will stop the inheritance if it hit a "leaf-node" as a parent.

The documents in the feed are all linked to two virtual groups, one for the denied users and one for the allowed users.


```
X-gsa-doc-controls: acl=1b78b22entricsk223k3A5Sb87B22accce22k23A822permtk2228Ck22case_sensitivity_type22k23A822everything_case_insensitive22k23Ck22name22k3A822Excell220Services220V1ewers22k23A822namespace22k3A822defau
lt122k3A85Ck22F85Ck22Fvm-dev-sp201313k3A20817822n2k22c22scope22k23A82222access22k23A82222permt22k22Ck22case_sensitivity_type22k23A8222everything_case_insensitive22k23Ck22name22k3A8222collection220Member
sk22k23A8222namespace22k23A8222default1_http3A85Ck22F85Ck22Fvm-dev-sp201313k3A208178222k22C22scope22k23A8222group22k23A822227Dk22Ck2278k222access22k23A82222permt22k23A8222C22case_sensitivity_type22k23A8222everything_case_insensitive22k23Ck22name
22k3A8222collection220Members22k23A8222namespace22k23A82222default1_http3A85Ck22F85Ck22Fvm-dev-sp201313k3A208178222k23A8222group22k23A822227Dk22Ck2278k222access22k23A82222permt22k23A8222C22case_sensitivity_type22k23A82222everything
_case_insensitive22k23Ck22name22k3A8222collection220Visitors22k23A82222Ck22namespace22k23A82222default1_http3A85Ck22F85Ck22Fvm-dev-sp201313k3A208178222k22C22scope22k23A8222group22k23A822227Dk22Ck2278k222access22k23A82222permt22k23A8222C22case_sensitivity_type22k23A82222everything_case_insensitive22k23Ck22name22k3A8222Coveo85C5Cjcrotaau22k23Ck22case_sensitivity_type22k23A8222user22k23A822227Dk22Ck2278k222access22k23A82222permt22k23A8222C22case_sensitivity_type22k23A82222everything_case_inse
nitive22k23A8222Ck22name22k3A8222devco85C5Cdevuser2018k2222c22scope22k23A8222user22k23A822227Dk22Ck2278k222access22k23A82222permt22k23A8222C22case_sensitivity_type22k23A82222everything_case_inse
nitive22k23A8222Ck22name22k3A8222devco85C5Cdevuser2018k2222c22scope22k23A8222user22k23A822227Dk22Ck2278k222access22k23A82222permt22k23A8222C22case_sensitivity_type22k23A82222inheritance_type22k23A82222APAREnt122k23A82222N7D
X-gsa-doc-controls: display_url=http3A85Ck22F82Fvm-dev-sp201313k3A20817
X-gsa-doc-controls: crawl_once=false
X-gsa-doc-controls: lock=false
X-gsa-doc-controls: scoring=content
X-gsa-external-metadata: google85Ck22Aobjtype22type1,sharepoint85Ck22Aparentu22tle1mylection
X-gsa-serve-security: secure
```

Example of a decoded ACL from a document header

The value of the X-gsa-doc-controls once it is decoded:

The value is an ACL in JSON

Example of cleaned Group feed

Group feed

The definition of the groups are given in a XML groups feed. [Feeding Groups to the Search Appliance](#)

Official documentation of GSA on ACLs: [Per-URL ACLs and ACL Inheritance](#)

Available Documentation

Documentation on the pure feed(the connectors are converting the information in a similar model): [Feeds Protocol Developer's Guide](#) PDF: [Feeds Protocol Developer's Guide](#)

The feed for the groups: Feeding Groups to the Search Appliance

The GitHub containing all the open source connectors and the connector SDK: <https://github.com/google/gsa>

How to setup the repository

This is a page on how to setup a GSA connector: [GSA Connector setup](#)

The Connector

Connection Layer

The first information that we receive is the root URL in a web feed. The default address for the feed is: "[GSAHostname]:19900/xmlfeed". The default address for the groups is: "[GSAHostname]:19900/xmlgroups".

The connector is waiting to receive a "Success" with a 200 response code to continue sending the information

How it works

The connector only allows the request coming from the GSA hostname in the config of the connector. If you want to access the content of the connector with another computer, you need to add the computer name to the `server.fullAccessHost` parameter. In fact, you need to have the computer name that will crawl the content. Example: `server.fullAccessHosts=wks-000299, vm-dev-gsa-connector`

If you want to have a backup of all the feeds send by the connector, you only need to give a valid path to the `feed.archiveDirectory` parameter.
Example: `feed.archiveDirectory=C:\\temp\\`

All backslash need to be double in the properties file of the connector.

Important value for the properties

gsa.version=7.6.50-0

If we do not set the GSA version, the connector wrapper will not work because the ACLs will not be in the right format and the ACLs for the "virtual" items will not be sent.

We did not identify what exactly is triggering the sending of ACLs feed, but we know that when we crawl all the documents, we receive one with all the ACLs that we need.

Running the tool

The POC need to have a temp folder on the C: of the computer (c:\temp\), this is where the tool will save the request until it is process.

You need to set the configuration you want in a Json file(more info in the [Parameters](#) board below), then you give the path to the file in argument. The important part is that the listening URLs in the configuration matches the URLs in properties file of the GSA connector.

The computer that run the POC need to be accessible through the specify ports of the configuration.

When both the adaptor and the POC are running, if everything goes fine, you do not need to do something else.

Opening port on Windows 10

1. Navigate to Control Panel, System and Security and Windows Firewall.
2. Select Advanced settings and highlight Inbound Rules in the left pane.
3. Right click Inbound Rules and select New Rule.
4. Add the port you need to open and click Next.
5. Add the protocol (TCP or UDP) and the port number into the next window and click Next.
6. Select Allow the connection in the next window and hit Next.
7. Select the network type as you see fit and click Next.
8. Name the rule something meaningful and click Finish.

You can manually full refresh the connector from the dashboard of the adaptors.

You can also send the root feed with Postman as binary file in a POST request. The tool can handle multiple requests at the same time.

Example of request sent to the tool with Postman.

The screenshot shows the Postman interface for a POST request to `http://localhost:19900/xmlfeed`. The request body is set to a file named `feedDesAclManquants.txt` with the `binary` type selected. The response status is `200 OK` with a time of `30979 ms` and a size of `113 B`. The response body is displayed as `1 Success`.

The samples folder contain examples of feed sent by Adaptors.

Connection Parameters

Most of the parameters are in the GSA connectors, so each connector has different parameters.

Item Types

Name	Description	Sub Item Types
------	-------------	----------------

Record	The equivalent of the Coveo Document	ACL
ACL	They contain the permission for a record. Note: they can be standalone items only pointing to an URL.	

Available Metadata

The metadatas are changing a lot from a connector to another. All the metadata are in the headers of the GET calls to the items of GSA.

✓ [Metadata mapping for SharePoint2013](#)

```
[
  {
    "id": "q2tb4dc46oefuubhrpoknl2e2u",
    "kind": "COMMON",
    "fieldName": "istoplevelsite",
    "extractionMethod": "METADATA",
    "content": "%[spistoplevelsite]"
  },
  {
    "id": "q2vmnkm6bckjasc4bixowrkgsi",
    "kind": "COMMON",
    "fieldName": "version",
    "extractionMethod": "METADATA",
    "content": "%[coveo_sharepointversion]"
  },
  {
    "id": "q5xgrpivejvvyo7t4n2hgasknba",
    "kind": "MAPPING",
    "type": "userprofile",
    "fieldName": "searchablemeta",
    "extractionMethod": "METADATA",
    "content": "%[coveo_AllMetaData]"
  },
  {
    "id": "q6qsbzczqx47zemz2o7xzxu2gni",
    "kind": "MAPPING",
    "type": "listitem.contacts",
    "fieldName": "title",
    "extractionMethod": "METADATA",
    "content": "%[ows_FullName]"
  },
  {
    "id": "qbi4ffxnfx5xvqqyljoamachi4",
    "kind": "MAPPING",
    "type": "listitem.webpagelibrary",
    "fieldName": "title",
    "extractionMethod": "METADATA",
    "content": "%[ows_BaseName]"
  },
  {
    "id": "qcc5p6fsggeguqwwf7be5ykjmi",
    "kind": "MAPPING",
    "type": "listitem.gantttasks",
    "fieldName": "searchablemeta",
    "extractionMethod": "LITERAL",
    "content": "\n%[ows_Body]\r\nPriority: %[ows_Priority]\r\nStatus: %[ows_Status]\n "
  },
  {
    "id": "qevsvpjrfb6ycmzfel4wjkkonu",
    "kind": "COMMON",
    "fieldName": "loginname",
    "extractionMethod": "METADATA",
    "content": "%[accountname]"
  },
  {
    "id": "qhewp3ovh464hewdf77rzaskxi",
    "kind": "MAPPING",
    "type": "listitem.posts",
    "fieldName": "body",
    "extractionMethod": "METADATA",
    "content": "%[ows_Body]"
  },
]
```

```

{
  "id": "qhiwdibhkuechm2asby72ekhhm",
  "kind": "COMMON",
  "fieldName": "parentguids",
  "extractionMethod": "METADATA",
  "content": "%[coveo_parentguids]"
},
{
  "id": "qhkx77cqmcac2jrvsvgggkkgm",
  "kind": "COMMON",
  "fieldName": "docsetguid",
  "extractionMethod": "METADATA",
  "content": "%[spdocsetguid]"
},
{
  "id": "qjl5xat2cahw4yr42moudhcgssu",
  "kind": "COMMON",
  "fieldName": "siteuri",
  "extractionMethod": "METADATA",
  "content": "%[spsiteuri]"
},
{
  "id": "qld3qr4buiecuequ3flncspoq",
  "kind": "COMMON",
  "fieldName": "tagguids",
  "extractionMethod": "METADATA",
  "content": "%[sptagguids]"
},
{
  "id": "qpcuepaef77uwqfj7u3g2ucov4",
  "kind": "COMMON",
  "fieldName": "foldingparent",
  "extractionMethod": "METADATA",
  "content": "%[coveo_foldingparent]"
},
{
  "id": "qqd7xgww3okqgmztkaahs3cdi4",
  "kind": "COMMON",
  "fieldName": "relativelisturi",
  "extractionMethod": "METADATA",
  "content": "%[coveo_relativelisturi]"
},
{
  "id": "qsd27mqeno2aho23u55ajqkduq",
  "kind": "MAPPING",
  "type": "listitem.microfeed",
  "fieldName": "author",
  "extractionMethod": "METADATA",
  "content": "%[coveo_MicroFeedAuthor]"
},
{
  "id": "qsne44cbqycoabfg5s4ku3kiw4",
  "kind": "COMMON",
  "fieldName": "middlename",
  "extractionMethod": "METADATA",
  "content": "%[middlename]"
},
{
  "id": "qtq56no2a6dhgynmwx5frv2ov4",
  "kind": "COMMON",
  "fieldName": "mobileuri",
  "extractionMethod": "METADATA",
  "content": "%[mobileuri]"
},
{
  "id": "qw7stp4l4luyhkpxr444dr2jyq",
  "kind": "MAPPING",
  "type": "listitem.links",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[ows_Comments]"
},

```

```

{
  "id": "qx76kf6ns4dnp3n4b727mokfta",
  "kind": "COMMON",
  "fieldName": "iscontainer",
  "extractionMethod": "METADATA",
  "content": "%[coveo_iscontainer]"
},
{
  "id": "r2lsxewwme26h5bruy5fco2dhu",
  "kind": "COMMON",
  "fieldName": "allfieldvalues",
  "extractionMethod": "METADATA",
  "content": "%[allfieldvalues]"
},
{
  "id": "r4mip5m34lqufykat2qkv5sel4",
  "kind": "COMMON",
  "fieldName": "authorid",
  "extractionMethod": "METADATA",
  "content": "%[coveo_authorid]"
},
{
  "id": "r4pzgitrhff6nz576lx4excjge",
  "kind": "MAPPING",
  "type": "listitem.survey",
  "fieldName": "title",
  "extractionMethod": "LITERAL",
  "content": "Response # %[ows_DisplayResponse]"
},
{
  "id": "r7uohijbfnjomefsoqtqoccfdy",
  "kind": "COMMON",
  "fieldName": "worktitle",
  "extractionMethod": "METADATA",
  "content": "%[worktitle]"
},
{
  "id": "r7v2qyczqrsi7k24kbt7oucjb4",
  "kind": "MAPPING",
  "type": "listitem.whereabouts",
  "fieldName": "searchablemeta",
  "extractionMethod": "LITERAL",
  "content": "\n %[ows_Comment]\r\n %[ows_Whereabout]\r\n %[ows_ContactInfo]\n "
},
{
  "id": "rckjfwpjekxpc5unfq27bbcdoi",
  "kind": "COMMON",
  "fieldName": "listtype",
  "extractionMethod": "METADATA",
  "content": "%[splisttype]"
},
{
  "id": "rdqpdcvy2xxlietigln6fcke4",
  "kind": "MAPPING",
  "type": "listitem.customgrid",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[coveo_AllMetaData]"
},
{
  "id": "rgqvh7gitkcsniajnukao2gre",
  "kind": "COMMON",
  "fieldName": "clickableuri",
  "extractionMethod": "METADATA",
  "content": "%[clickableuri]"
},
{
  "id": "rh2nbwn4iyppag2tinh27kspzy",
  "kind": "MAPPING",
  "type": "listitem.calltrack",
  "fieldName": "searchablemeta",
  "extractionMethod": "LITERAL",

```

```

"content": "\n%[ows_Detail]\r\n%[ows_Comments]\r\n%[ows_CallbackNumber]\n "
},
{
  "id": "rhzdpiekxtr4yivgztbro5kn4u",
  "kind": "MAPPING",
  "type": "file",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[coveo_AllMetaData]"
},
{
  "id": "riik7k26vrkvm4uav6nmefso5i",
  "kind": "COMMON",
  "fieldName": "connectortype",
  "extractionMethod": "METADATA",
  "content": "%[connectortype]"
},
{
  "id": "rkcm3h64cecs2ifisuvfxdkjay",
  "kind": "MAPPING",
  "type": "listitem.tasks",
  "fieldName": "searchablemeta",
  "extractionMethod": "LITERAL",
  "content": "\n%[ows_Body]\r\nPriority: %[ows_Priority]\r\nStatus: %[ows_Status]\n "
},
{
  "id": "rkgtesx2bruv5dvlzfnsgcaya",
  "kind": "COMMON",
  "fieldName": "listtruncatedguid",
  "extractionMethod": "METADATA",
  "content": "%[splisttruncatedguid]"
},
{
  "id": "rkvqxlpbvbwazcbrkhcrkzcjhm",
  "kind": "MAPPING",
  "type": "listitem.performancepointcontent",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[ows_PPSMA_Description]"
},
{
  "id": "rm4nqqryk3xzz5v43nywpdsf2i",
  "kind": "MAPPING",
  "type": "listitem.microfeed",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[ows_SearchContent]"
},
{
  "id": "rq4azjscevub2m65qo6xdd2dfe",
  "kind": "MAPPING",
  "type": "listitem.sitedirectory",
  "fieldName": "searchablemeta",
  "extractionMethod": "LITERAL",
  "content": "\n%[Description]\r\n%[ows_Owner]\r\n%[ows_Division]\r\n%[ows_Region]\n "
},
{
  "id": "rqg3uzjmdnykiil5xijaoqcfzu",
  "kind": "MAPPING",
  "type": "documentset",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[coveo_AllMetaData]"
},
{
  "id": "rriexaqj5hlow7pw2jrwql2kfm",
  "kind": "MAPPING",
  "type": "listitem.keyperformanceindicator",
  "fieldName": "searchablemeta",
  "extractionMethod": "LITERAL",
  "content": "\n%[ows_KpiDescription]\r\n%[ows_KpiComments]\n "
},
{

```



```
{
  "id": "rrsnwbxj5iv2vz4puhb6nisjqm",
  "kind": "COMMON",
  "fieldName": "iscontainer",
  "extractionMethod": "METADATA",
  "content": "%[spiscontainer]"
},
{
  "id": "rsr34zq2hplwi3vjpic6ckehe",
  "kind": "COMMON",
  "fieldName": "mobilephone",
  "extractionMethod": "METADATA",
  "content": "%[mobile]"
},
{
  "id": "rtekm72hv26binzpfn3gcjklca",
  "kind": "COMMON",
  "fieldName": "listguid",
  "extractionMethod": "METADATA",
  "content": "%[splistguid]"
},
{
  "id": "rtljj7iwc7uffdbo2ndo5bsnwe",
  "kind": "COMMON",
  "fieldName": "mobilephone",
  "extractionMethod": "METADATA",
  "content": "%[cellphone]"
},
{
  "id": "rwdp5ylpt6tavnxnj7mnjwsic4",
  "kind": "COMMON",
  "fieldName": "sitetruncatedguid",
  "extractionMethod": "METADATA",
  "content": "%[spsitetruncatedguid]"
},
{
  "id": "rwijukanz4q6f6mbervptykgn4",
  "kind": "COMMON",
  "fieldName": "filetype",
  "extractionMethod": "METADATA",
  "content": "%[filetype]"
},
{
  "id": "rwuwa4n22vqlr6kv2xxy52smi",
  "kind": "MAPPING",
  "type": "listitem",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[coveo_AllMetaData]"
},
{
  "id": "ryxuajjutj4kebemagb5zz2ppu",
  "kind": "MAPPING",
  "type": "web",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[Description]"
},
{
  "id": "rzvufoeyg5gl7vju2esepkiuy",
  "kind": "COMMON",
  "fieldName": "folders",
  "extractionMethod": "METADATA",
  "content": "%[folders]"
},
{
  "id": "s3rzvsq4inzwbrgp2ijhfsclja",
  "kind": "MAPPING",
  "type": "listitem.webpagelibrary",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[ows_WikiField]"
}
```

```

},
{
  "id": "s5q2ogcafbv75hw7kurlfa2gme",
  "kind": "COMMON",
  "fieldName": "documenttype",
  "extractionMethod": "METADATA",
  "content": "%[documenttype]"
},
{
  "id": "s6xznfdti5skqgqncfzr2jkcwa",
  "kind": "COMMON",
  "fieldName": "ispersonalsite",
  "extractionMethod": "METADATA",
  "content": "%[spispersonalsite]"
},
{
  "id": "sbtquzgabu32d76knweeu62pze",
  "kind": "MAPPING",
  "type": "listitem.issuetracking",
  "fieldName": "searchablemeta",
  "extractionMethod": "LITERAL",
  "content": "\n%[ows_Comment]\r\n%[ows_V3Comments]\r\nStatus: %[ows_Status]\n "
},
{
  "id": "sda45e2ys6akeahhgwtheasgru",
  "kind": "COMMON",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[searchablemeta]"
},
{
  "id": "shd7lcjmh74l5667wxisw52hle",
  "kind": "MAPPING",
  "type": "userprofile",
  "fieldName": "title",
  "extractionMethod": "METADATA",
  "content": "%[coveo_UserProfileName]"
},
{
  "id": "sjnweex23apj2ezrw2uhq42oby",
  "kind": "COMMON",
  "fieldName": "listitemguid",
  "extractionMethod": "METADATA",
  "content": "%[splistitemguid]"
},
{
  "id": "slxk62wxyv6psn2ag7vz65cota",
  "kind": "MAPPING",
  "type": "list",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[Description]"
},
{
  "id": "snskg76tkcixnyarpf75iekekq",
  "kind": "COMMON",
  "fieldName": "cc",
  "extractionMethod": "METADATA",
  "content": "%[cc]"
},
{
  "id": "ssiwma6ymv3n5th5s477fpsgii",
  "kind": "COMMON",
  "fieldName": "parenttruncatedguid",
  "extractionMethod": "METADATA",
  "content": "%[spparenttruncatedguid]"
},
{
  "id": "sweftiwl4oybqxt4ge7ugfspiy",
  "kind": "COMMON",
  "fieldName": "istoplevelsite",
  "extractionMethod": "METADATA",

```

```
"content": "%[coveo_istoplevelsite]"
},
{
  "id": "swlbml7rbaxolor6dsdj74cja",
  "kind": "COMMON",
  "fieldName": "manager",
  "extractionMethod": "METADATA",
  "content": "%[manager]"
},
{
  "id": "sxfnrq77a3iv46pzbyfdgqcehq",
  "kind": "MAPPING",
  "type": "listitem.unknown",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[coveo_AllMetaData]"
},
{
  "id": "syg3u7lmsg6fpqjih2s3t7ci3m",
  "kind": "MAPPING",
  "type": "listitem.meetingobjective",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[ows_Objective]"
},
{
  "id": "sys72t472pvinfak3agz7bcjfu",
  "kind": "MAPPING",
  "type": "listitem.discussionboard",
  "fieldName": "title",
  "extractionMethod": "METADATA",
  "content": "%[ows_BaseName]"
},
{
  "id": "syz2xc6fzbdjpoitkaexnlsiiy",
  "kind": "COMMON",
  "fieldName": "conversationssubjectid",
  "extractionMethod": "METADATA",
  "content": "%[conversationssubjectid]"
},
{
  "id": "t4q26emfrpudnnwrpuab2bkpo4",
  "kind": "COMMON",
  "fieldName": "relatedlink",
  "extractionMethod": "METADATA",
  "content": "%[coveo_externalurl]"
},
{
  "id": "t72sbn4wbhfoduyqcjjbyu2ahy",
  "kind": "COMMON",
  "fieldName": "listitemguid",
  "extractionMethod": "METADATA",
  "content": "%[coveo_listitemguid]"
},
{
  "id": "tbtryngjzhuyxc3nhddfezkc4e",
  "kind": "COMMON",
  "fieldName": "firstname",
  "extractionMethod": "METADATA",
  "content": "%[firstname]"
},
{
  "id": "tcsfh5slfxragjl2cexcb2mw4",
  "kind": "COMMON",
  "fieldName": "sourcetype",
  "extractionMethod": "METADATA",
  "content": "%[sourcetype]"
},
{
  "id": "tecr7tfj765hrkuxqhukbp2gbm",
  "kind": "MAPPING",
  "type": "listitem.circulation",
```

```
"fieldName": "searchablemeta",
"extractionMethod": "LITERAL",
"content": "\n%[ows_Body]\r\n%[ows_Comments]\n "
},
{
  "id": "teddo3cen3i2ofqs2znmarsh6a",
  "kind": "COMMON",
  "fieldName": "tagnames",
  "extractionMethod": "METADATA",
  "content": "%[sptagnames]"
},
{
  "id": "teq6ecuheyive3alr6fnh62ine",
  "kind": "COMMON",
  "fieldName": "foldingchild",
  "extractionMethod": "METADATA",
  "content": "%[coveo_foldingchild]"
},
{
  "id": "tfbzyjdi2vjxfrkrhdhc5kkom",
  "kind": "MAPPING",
  "type": "listitem.meetings",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[ows_Location]"
},
{
  "id": "ti6aghn7veyc65bxkfgvjshe4",
  "kind": "COMMON",
  "fieldName": "conversationssubject",
  "extractionMethod": "METADATA",
  "content": "%[conversationssubject]"
},
{
  "id": "tjdyfiph7e4pieyx2ydiimspza",
  "kind": "MAPPING",
  "type": "listitem.admintasks",
  "fieldName": "searchablemeta",
  "extractionMethod": "LITERAL",
  "content": "\n%[Description]\r\n%[ows_Service]\r\nStatus: %[ows_Status]\n "
},
{
  "id": "tlzezeshwiyyfsbo2c54s2kt4",
  "kind": "COMMON",
  "fieldName": "office",
  "extractionMethod": "METADATA",
  "content": "%[office]"
},
{
  "id": "tmehljudenkz4gjo535ped2psy",
  "kind": "MAPPING",
  "type": "listitem.datasources",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[coveo_AllMetaData]"
},
{
  "id": "tmm5yd4zvnzgybt2ufpboskdea",
  "kind": "MAPPING",
  "type": "userprofile",
  "fieldName": "body",
  "extractionMethod": "LITERAL",
  "content": "First Name: %[FirstName]\r\nLast Name: %[LastName]\r\nAbout: %[AboutMe]\r\nTitle: %[JobTitle]\r\nWork Phone:
%[WorkPhone]\r\nWork Email: %[WorkEmail]\r\nWeb Site: %[WebSite]\r\n "
},
{
  "id": "tplztxhbm2vzgcfxp3aa36chmq",
  "kind": "COMMON",
  "fieldName": "printableuri",
  "extractionMethod": "METADATA",
  "content": "%[printableuri]"
},
}
```

```
{
  "id": "tpxgpnobx3bfxaojpmo3qsjbe",
  "kind": "COMMON",
  "fieldName": "company",
  "extractionMethod": "METADATA",
  "content": "%[company]"
},
{
  "id": "tu6ks4n7mn6f3nby4wajew2ecy",
  "kind": "MAPPING",
  "type": "listitem.announcements",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[ows_Body]"
},
{
  "id": "tvxgg36k7vlq727z7ykociso4m",
  "kind": "COMMON",
  "fieldName": "bcc",
  "extractionMethod": "METADATA",
  "content": "%[bcc]"
},
{
  "id": "tycjzzl3ihpe6m43q6qrmu2ncq",
  "kind": "COMMON",
  "fieldName": "lastname",
  "extractionMethod": "METADATA",
  "content": "%[lastname]"
},
{
  "id": "u2632da6tfpnequoyobojci7q",
  "kind": "COMMON",
  "fieldName": "workphone",
  "extractionMethod": "METADATA",
  "content": "%[workphone]"
},
{
  "id": "u37bhskgkx5sh3jubvno5knuq",
  "kind": "COMMON",
  "fieldName": "parentguids",
  "extractionMethod": "METADATA",
  "content": "%[spparentguids]"
},
{
  "id": "u3xbuoc7yyc7dd35wyrfa22lze",
  "kind": "COMMON",
  "fieldName": "siteguid",
  "extractionMethod": "METADATA",
  "content": "%[spsiteguid]"
},
{
  "id": "u4irfsp7msirouwyjjkrejsknu",
  "kind": "COMMON",
  "fieldName": "docsetname",
  "extractionMethod": "METADATA",
  "content": "%[spdocsetname]"
},
{
  "id": "u5pldw563247ben5clddwj2nji",
  "kind": "COMMON",
  "fieldName": "company",
  "extractionMethod": "METADATA",
  "content": "%[o]"
},
{
  "id": "uco332oalxltdeuiis7st7kgmy",
  "kind": "MAPPING",
  "type": "listitem.comments",
  "fieldName": "blogpostid",
  "extractionMethod": "METADATA",
  "content": "%[ows_PostID]"
},
}
```

```
{
  "id": "udgc3wlslrinhsijl566xzckkq",
  "kind": "MAPPING",
  "type": "listitem.workflowprocess",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[coveo_AllMetaData]"
},
{
  "id": "ugdezftvjjm5mrld6abatdkgym",
  "kind": "COMMON",
  "fieldName": "isdraft",
  "extractionMethod": "METADATA",
  "content": "%[coveo_isdraft]"
},
{
  "id": "uhms25tskrdyvk3t7qwtaskfu",
  "kind": "COMMON",
  "fieldName": "tagNames",
  "extractionMethod": "METADATA",
  "content": "%[coveo_tagNames]"
},
{
  "id": "ui2ny3zbusnlshkrpkegykkjfe",
  "kind": "COMMON",
  "fieldName": "listbasetype",
  "extractionMethod": "METADATA",
  "content": "%[splistbasetype]"
},
{
  "id": "ujppvqw7ryvrpza4m2pix2scza",
  "kind": "COMMON",
  "fieldName": "tagguids",
  "extractionMethod": "METADATA",
  "content": "%[coveo_tagguids]"
},
{
  "id": "um5ef376bmbh2yqbyupber2bli",
  "kind": "COMMON",
  "fieldName": "author",
  "extractionMethod": "METADATA",
  "content": "%[author]"
},
{
  "id": "umzwrozhekhovltzuykissezi",
  "kind": "MAPPING",
  "type": "listitem.imedic",
  "fieldName": "searchablemeta",
  "extractionMethod": "LITERAL",
  "content":
    "\n%[ows_IMEDisplay]\r\n%[ows_IMEComment1]\r\n%[ows_IMEComment2]\r\n%[ows_IMEComment3]\r\n%[ows_IMEUrl]\r\n%[ows_IMEPos]\n"
},
{
  "id": "un6aqmc264wzn273fixmrrkpom",
  "kind": "COMMON",
  "fieldName": "listtype",
  "extractionMethod": "METADATA",
  "content": "%[coveo_listtype]"
},
{
  "id": "unqd6k63qpmvdxesonl5772pge",
  "kind": "COMMON",
  "fieldName": "homephone",
  "extractionMethod": "METADATA",
  "content": "%[homephone]"
},
{
  "id": "uotpcsr4grpv7etcj5vtxsk5i",
  "kind": "COMMON",
  "fieldName": "date",
  "extractionMethod": "METADATA",

```

```

"content": "%[date]"
},
{
  "id": "uusg7svw7digne7pf6szl7kcy4",
  "kind": "MAPPING",
  "type": "listitem.links",
  "fieldName": "clickableuri",
  "extractionMethod": "METADATA",
  "content": "%[coveo_ExternalLinkUrl]"
},
{
  "id": "uvkieuwcgriaqkd2peumkwsm7u",
  "kind": "COMMON",
  "fieldName": "language",
  "extractionMethod": "METADATA",
  "content": "%[language]"
},
{
  "id": "ux3jqs54wioeno5ozffmkw2mdq",
  "kind": "COMMON",
  "fieldName": "siteuri",
  "extractionMethod": "METADATA",
  "content": "%[coveo_siteuri]"
},
{
  "id": "uybzpie4oqxao6pcab3nf2lqm",
  "kind": "COMMON",
  "fieldName": "sitename",
  "extractionMethod": "METADATA",
  "content": "%[coveo_sitename]"
},
{
  "id": "v43sdkspkvumg3wowp67jlsdbm",
  "kind": "MAPPING",
  "type": "listitem.healthreports",
  "fieldName": "searchablemeta",
  "extractionMethod": "LITERAL",
  "content":
    "\n%[ows_HealthReportExplanation]\r\n%[ows_HealthReportRemedy]\r\n%[ows_HealthReportServers]\r\n%[ows_HealthReportServices]\n "
},
{
  "id": "v5sb27wnc4wu35h3gnmjlv2kdu",
  "kind": "MAPPING",
  "type": "listitem.reportlibrary",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[ows_ReportDescription]"
},
{
  "id": "v6xmnwweg5jrhifctsgnvxkcp4",
  "kind": "COMMON",
  "fieldName": "contenttype",
  "extractionMethod": "METADATA",
  "content": "%[spcontenttype]"
},
{
  "id": "v7xbj7hyz7e6ew7obymh7ucove",
  "kind": "COMMON",
  "fieldName": "title",
  "extractionMethod": "METADATA",
  "content": "%[title]"
},
{
  "id": "vayly66z2oupj2kgrquxshck7a",
  "kind": "COMMON",
  "fieldName": "location",
  "extractionMethod": "METADATA",
  "content": "%[location]"
},
{
  "id": "vbzzk2tf2cpcvsioi6evud2bfy",
  "kind": "COMMON",

```

```

"fieldName": "listtruncatedguid",
"extractionMethod": "METADATA",
"content": "%[coveo_listtruncatedguid]"
},
{
  "id": "vc2axkekvjcj46bykeg7dvlcfku",
  "kind": "COMMON",
  "fieldName": "fileextension",
  "extractionMethod": "METADATA",
  "content": "%[fileextension]"
},
{
  "id": "vcd5jj2hkneuk43orcfcz53cj5e",
  "kind": "COMMON",
  "fieldName": "fax",
  "extractionMethod": "METADATA",
  "content": "%[fax]"
},
{
  "id": "vdjv4hxytq7mspcgnpjfnjsawi",
  "kind": "MAPPING",
  "type": "listitem.workflowhistory",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[coveo_AllMetaData]"
},
{
  "id": "ve3s2wj4pgvkv25qlfn722bbe",
  "kind": "MAPPING",
  "type": "listitem.publishingpageslibrary",
  "fieldName": "searchablemeta",
  "extractionMethod": "LITERAL",
  "content":
    "\n%[ows_PublishingPageContent]\r\n%[ows_PublishingImageCaption]\r\n%[ows_SummaryLinks]\r\n%[ows_SummaryLinks2]\r\n%[ows_ArticleByLine]\r\n%[ows_Comments]\n "
},
{
  "id": "vfgzbcyq37oe3oo4dx3ioocalq",
  "kind": "COMMON",
  "fieldName": "listitemid",
  "extractionMethod": "METADATA",
  "content": "%[coveo_listitemid]"
},
{
  "id": "vgnzltgmsweo77jt4bsfc5kmia",
  "kind": "MAPPING",
  "type": "listitem.events",
  "fieldName": "searchablemeta",
  "extractionMethod": "LITERAL",
  "content": "%[Description]\r\n%[ows_Location]\n "
},
{
  "id": "vgpb5f5ldukci6q42mxlwdcgaq",
  "kind": "COMMON",
  "fieldName": "contenttype",
  "extractionMethod": "METADATA",
  "content": "%[ows_contenttype]"
},
{
  "id": "vhgw7ctv57vmihfn65r6eecjgm",
  "kind": "COMMON",
  "fieldName": "enableofficeintegration",
  "extractionMethod": "METADATA",
  "content": "%[spenableofficeintegration]"
},
{
  "id": "vhrsgphspmklammap6qv5kcva",
  "kind": "COMMON",
  "fieldName": "size",
  "extractionMethod": "METADATA",
  "content": "%[size]"
},

```



```
{
  "id": "vunjwsjctiwt4npyi75jt72l2i",
  "kind": "MAPPING",
  "type": "listitem.comments",
  "fieldName": "body",
  "extractionMethod": "METADATA",
  "content": "%[ows_Body]"
},
{
  "id": "vn6mzzr4poxhksfegkt4652p44",
  "kind": "COMMON",
  "fieldName": "blogpostid",
  "extractionMethod": "METADATA",
  "content": "%[postid]"
},
{
  "id": "vpqnihfvuuvyrvmxmeq6oeccbi",
  "kind": "MAPPING",
  "type": "listitem.genericlist",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[coveo_AllMetaData]"
},
{
  "id": "vqylwqfpqcgmvl7zxru6we2dru",
  "kind": "COMMON",
  "fieldName": "relativelisturi",
  "extractionMethod": "METADATA",
  "content": "%[sprelativelisturi]"
},
{
  "id": "vtbkuyzivkf7plczbylpokkzu",
  "kind": "COMMON",
  "fieldName": "ispersonalsite",
  "extractionMethod": "METADATA",
  "content": "%[coveo_ispersonalsite]"
},
{
  "id": "vtir7wzkyrzcni3rmqlcm4m",
  "kind": "MAPPING",
  "type": "listitem.discussionboard",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[ows_Body]"
},
{
  "id": "w25wkyiglxtfj53wkt2dksaam",
  "kind": "COMMON",
  "fieldName": "siteguid",
  "extractionMethod": "METADATA",
  "content": "%[coveo_siteguid]"
},
{
  "id": "w3mt45ticynuletykqjlf2eb4",
  "kind": "COMMON",
  "fieldName": "listguid",
  "extractionMethod": "METADATA",
  "content": "%[coveo_listguid]"
},
{
  "id": "w3okgthmipccibuvbepyzusggy",
  "kind": "COMMON",
  "fieldName": "authorloginname",
  "extractionMethod": "METADATA",
  "content": "%[authorloginname]"
},
{
  "id": "w7rhka2ue67libbojukad52mzu",
  "kind": "MAPPING",
  "type": "listitem.agenda",
  "fieldName": "searchablemeta",
  "extractionMethod": "LITERAL",

```

```
"content": "\n%[ows_Owner]\r\n%[ows_Time]\r\n%[ows_Notes]\n "
},
{
  "id": "wb4fzymmhpu7h567j5yi2k2g6i",
  "kind": "COMMON",
  "fieldName": "itemtype",
  "extractionMethod": "METADATA",
  "content": "%[spitemtype]"
},
{
  "id": "wcbkk3vddrxdr15vllnrckmdq",
  "kind": "COMMON",
  "fieldName": "workemail",
  "extractionMethod": "METADATA",
  "content": "%[workemail]"
},
{
  "id": "wcdjob7zonhmt73csgaaxjkfxq",
  "kind": "COMMON",
  "fieldName": "authorloginname",
  "extractionMethod": "METADATA",
  "content": "%[coveo_authorloginname]"
},
{
  "id": "weieojg5j555dbjmw7snvkcpf4",
  "kind": "COMMON",
  "fieldName": "month",
  "extractionMethod": "METADATA",
  "content": "%[month]"
},
{
  "id": "wejkuwso5vjw2cje446f4ackja",
  "kind": "COMMON",
  "fieldName": "year",
  "extractionMethod": "METADATA",
  "content": "%[year]"
},
{
  "id": "wk74nxz3rapns05krtgnjvsdbu",
  "kind": "COMMON",
  "fieldName": "parents",
  "extractionMethod": "METADATA",
  "content": "%[parents]"
},
{
  "id": "wndoazzf2jz5ysyud6s3gtshqq",
  "kind": "COMMON",
  "fieldName": "from",
  "extractionMethod": "METADATA",
  "content": "%[from]"
},
{
  "id": "wqjkc72ndiz5dyf66zl5e2n5y",
  "kind": "COMMON",
  "fieldName": "listbasetype",
  "extractionMethod": "METADATA",
  "content": "%[coveo_listbasetype]"
},
{
  "id": "wvfgo7nboe43ywcadr2qkwspzm",
  "kind": "MAPPING",
  "type": "listitem.contacts",
  "fieldName": "searchablemeta",
  "extractionMethod": "LITERAL",
  "content":
    "\n%[ows_FullName]\r\n%[ows_Email]\r\n%[ows_Company]\r\n%[ows_JobTitle]\r\n%[ows_WorkPhone]\r\n%[ows_HomePhone]\r\n%[ows_CellPhone]\r\n%[ows_WorkFax]\r\n%[ows_WorkAddress]\r\n%[ows_WorkCity]\r\n%[ows_WorkState]\r\n%[ows_WorkZip]\r\n%[ows_WorkCountry]\r\n%[ows_WebPage]\r\n%[ows_Comments]\n "
},
{
  "id": "wvgisw5zredvrusqa47jjssgnm",
  "kind": "COMMON",
```

```
"fieldName": "listitemid",
"extractionMethod": "METADATA",
"content": "%[splistitemid]"
},
{
  "id": "wvzulxgyqmx47uwpdwk3e2pd4",
  "kind": "COMMON",
  "fieldName": "authorid",
  "extractionMethod": "METADATA",
  "content": "%[spauthorid]"
},
{
  "id": "wwbaacleffonhd3vcjsemg2eum",
  "kind": "COMMON",
  "fieldName": "version",
  "extractionMethod": "METADATA",
  "content": "%[spversion]"
},
{
  "id": "wwuipgscxucck6m7pw6v7ccpue",
  "kind": "MAPPING",
  "type": "listitem.decision",
  "fieldName": "searchablemeta",
  "extractionMethod": "LITERAL",
  "content": "\n%[ows_Contact]\r\nStatus: %[ows_Status]\n "
},
{
  "id": "wxrbepjckglmhyvjc356vcpdu",
  "kind": "COMMON",
  "fieldName": "listname",
  "extractionMethod": "METADATA",
  "content": "%[splistname]"
},
{
  "id": "wzn56kgdhrp6yxled36olz2efu",
  "kind": "COMMON",
  "fieldName": "fullName",
  "extractionMethod": "METADATA",
  "content": "%[fullName]"
},
{
  "id": "x5tlhil7usqp2npq7tqkg2o7m",
  "kind": "COMMON",
  "fieldName": "sitename",
  "extractionMethod": "METADATA",
  "content": "%[spsitename]"
},
{
  "id": "x5usgug3h6szie4ekuje6n2jbq",
  "kind": "COMMON",
  "fieldName": "itemtype",
  "extractionMethod": "METADATA",
  "content": "%[coveo_type]"
},
{
  "id": "x6fe4zwwkiyjmsklqjuidj7kd2e",
  "kind": "COMMON",
  "fieldName": "blogpostid",
  "extractionMethod": "METADATA",
  "content": "%[spblogpostid]"
},
{
  "id": "xairn2bpcjl7ambgldyx6bcjne",
  "kind": "COMMON",
  "fieldName": "listname",
  "extractionMethod": "METADATA",
  "content": "%[coveo_listname]"
},
{
  "id": "xbjctntekytqvrzb4s57jlk154",
  "kind": "MAPPING",
  "type": "listitem.whereabouts",
```

```
"fieldName": "title",
"extractionMethod": "METADATA",
"content": "%[ows_Name]"
},
{
  "id": "xevzgbwdcbvkmiu224x3qvsh64",
  "kind": "COMMON",
  "fieldName": "foldingcollection",
  "extractionMethod": "METADATA",
  "content": "%[coveo_foldingcollection]"
},
{
  "id": "xeyqsdxtteekrp7mopw7dvukady",
  "kind": "COMMON",
  "fieldName": "subject",
  "extractionMethod": "METADATA",
  "content": "%[subject]"
},
{
  "id": "xjkbmfma4r3iigtmfyzzkkg",
  "kind": "COMMON",
  "fieldName": "datebucket",
  "extractionMethod": "METADATA",
  "content": "%[datebucket]"
},
{
  "id": "xjqtpdowdg5bfhntyj2qg6khby",
  "kind": "COMMON",
  "fieldName": "filename",
  "extractionMethod": "METADATA",
  "content": "%[filename]"
},
{
  "id": "xm2kbww6gryx2ici4anyxd2czq",
  "kind": "COMMON",
  "fieldName": "to",
  "extractionMethod": "METADATA",
  "content": "%[to]"
},
{
  "id": "xm3z3yl57qdfc6ixpkg63lkkzq",
  "kind": "COMMON",
  "fieldName": "parentname",
  "extractionMethod": "METADATA",
  "content": "%[spparentname]"
},
{
  "id": "xnfy67ox4irbikiplm2t632biq",
  "kind": "MAPPING",
  "type": "listitem.textbox",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[ows_Text]"
},
{
  "id": "xnllt2kikjc3kwbs3nl5dxchnq",
  "kind": "COMMON",
  "fieldName": "sitetruncatedguid",
  "extractionMethod": "METADATA",
  "content": "%[coveo_sitetruncatedguid]"
},
{
  "id": "xnuectvf4xt7kuosboop66kc3a",
  "kind": "MAPPING",
  "type": "listitem.thingstobring",
  "fieldName": "searchablemeta",
  "extractionMethod": "LITERAL",
  "content": "\n%[ows_Comment]\r\n%[ows_Owner]\n "
},
{
  "id": "xpdidkf23zltg6v3f4afr72pfa",
  "kind": "COMMON",
```

```
"fieldName": "isdraft",
"extractionMethod": "METADATA",
"content": "%[spisdraft]"
},
{
  "id": "xrqjb6lh2jfcorgj6konjwsgdi",
  "kind": "MAPPING",
  "type": "listitem.userinformation",
  "fieldName": "searchablemeta",
  "extractionMethod": "METADATA",
  "content": "%[coveo_AllMetaData]"
},
{
  "id": "xuckn6fpvavm4bt3rmwm3c2pxa",
  "kind": "COMMON",
  "fieldName": "site",
  "extractionMethod": "METADATA",
  "content": "%[site]"
},
{
  "id": "xwrodl2prowxr4a6m5knhkfge",
  "kind": "COMMON",
  "fieldName": "mobilephone",
  "extractionMethod": "METADATA",
  "content": "%[mobilephone]"
},
{
  "id": "xz27v6tv2bxjtuizceeengkgli",
  "kind": "MAPPING",
  "type": "listitem.links",
  "fieldName": "title",
  "extractionMethod": "METADATA",
  "content": "%[coveo_ExternalLinkTitle]"
}
]
```

The Crawler

Parameters

Any missing required field will stop the program. The configuration file can be modify between each request.

Name	Description	Is Required	Default Value	In the config.json exept specify otherwise.
PLATFORM_API_ENDPOINT	The link to the platform API.	Yes		
PUSH_API_ENDPOINT	The link to the push API to push the documents.	Yes		
LISTENING_URL_FEED	The URL to listen for feeds.	No	http://localhost:19900/xmlfeed/	
LISTENING_URL_GROUPS	The URL to listen for groups.	No	http://localhost:19900/xmlgroups/	
GSA_AUTH_URL	The URL to listen for login requests.	No	http://localhost:8000/accounts/ClientLogin/	
ORGANIZATION_ID	The org. containing the source for the GSA content.	Yes		
PUSH_SOURCE_ID	The source to push the content of the adaptor.	Yes		
API_KEY	The API key to use to push the content.	Yes		
PROVIDER_ID	The ID of the expended provider used for the entities.	Yes		
DELETE_ON_INVALID_URL	If a document does not return a code 200/OK, we try to delete it.	No	True	

REQUIRE_DISPLAY_URL	If a document does not have a display URL, we do not index it.	No	False	
PUSH_RECORDS_WITHOUT_ACL	If we are only indexing documents with a display URL.	No	False	
Path to the configuration file	A path to the file containing the configuration. The tool also have a default configuration file that push to the dev environment of Coveo.	Yes	Samples\defaultconfig.json	Given in launch parameter to the tool.
PUSH_SOURCE_DICTONNARY	Json dictionary that allow to push to different source by giving a datacontent with a source id. "datasource": "sourceid"	No	Empty dictionary	

Feature Grid

Name	Is Supported	Additional Information
Pause/Resume	No	
Incremental Refresh	Yes	
Live Monitoring	No	
Refresh Folder	No	
Refresh Document	No	
Delete Folder	No	Delete in full refresh
Delete Document	No	Delete in full refresh
Security (Early or late binding?)	Yes	Might not be perfect.
Multi-Threading support	Yes	
Built-in Facets	No	

Known Bugs / Limitations

Si on l'utilise un seul instance pour plusieurs sources, on peut juste avoir un SP par instance du programme

With a single instance of the tool, we can have a source by adaptor by using the PUSH_SOURCE_DICTONNARY parameter to match each adaptor datasource with a source id. The problem is that we can only have one security provider by instance of the tool because the groups feeds do not contain the datasource. (We could parse the custom header of GSA to extract the datasource even in the groups feeds.)

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
CT-2755	Productize the GSA offer		May 15, 2017	Jul 13, 2017		Unassigned	Matthieu Theriault		ON HOLD	

1 issue

Nice-to-Have Features (from a developer perspective)

TODO > Add an option to allow the records to be part of any of the roots URL. (Now limited to the root URL that linked to that page.)

TODO> Parse the extra content given with the feed to get the datasource instead of using the header of the feed, because the group feed do not have that information in the feed part. This would allow use to link the provider to the datasource like the sourceids.

We could look to want we can do with this:

Status code

If you refresh on a page of content, the header contain less information and the status code change. :

Request URL: <http://vm-dev-gsa-connector.dev.coveo.com:5978/doc/%5C%5Ccorp.coveo.com/dfs/QBC/Temporary/jpcroteau/creatingFiles.py>

Request Method: GET

Status Code:  204 No Content

Remote Address: 192.168.52.183:5978

Response Headers [view source](#)

PUSH_SOURCE_DICTIONNARY