Atrax Keeper

Web Crawler Administration Tool

# Introduction

Atrax is an advanced web crawling system with many capabilities. Initiating, monitoring, and managing crawl jobs is a complex task. At the lowest level and without additional tools only those that are intimately knowledgeable about the system internals have the ability to use the system. Atrax Keeper is an administration tool designed to abstract this complexity away so that the system can be controlled by one or more lay users.

Using Atrax Keeper users must, with minimal training, be able to:

* Create crawl jobs
* Configure crawl jobs
* Secure crawl jobs
* Start and pause crawl jobs
* Monitor crawl job progress
* Manage crawl job costs

# Definitions

**Atrax:** A spider found in Australia.

**Atrax Keeper:** Someone that tends to Atrax spiders.

**Client:** The software requesting something from a server. Web browsers and crawlers are types of clients.

**Server:** The software that responds to client requests and sends back what is requested.

**Resource:** Any discrete file or set of information that can be referenced using a URL.

**Document:** The embodiment of a resource, fully rendered, and downloaded to a client. These are often dynamically created on the server. They can be scripts, pictures, videos, web pages, files, etc.

**Web Crawler:** A type of client that systematically requests resources from one or more servers. It uses the URLs found in the documents in order to request more resources. Often called a spider. It is a type of web bot.

**Crawl Job:** A single instance of a crawl defined by when it was first started and its scope. The scope may change slightly overtime but it is still considered the same job.

**Frontier:** The list of URLs that have been identified but have not yet been crawled.

**Wild Card:** Appears in pattern expressions to signify that the expression can match anything in place of the wild card. Atrax uses \* as the wild card character.

**URL:** Uniform Resource Locator

**Anatomy of a URL**

It is important to understand the parts of a URL in order to understand the rest of this document. A URL can be broken up into parts as follows. Only the parts that have \* beside the name must be found in a URL to make it valid.

|  |  |
| --- | --- |
| **http://my.example.com.de/software/download/apps.html?en=eng&color=red#top** | |
|  | |
| **Protocol/Scheme\*** | Other common examples include: https:// ftp:// |
| **Domain** | This is the address of the server that hosts the site. The following four parts are part of the domain portion of the URL. |
| **Subdomain** | There can be one or more of these. For example: my.cool.site. A very common subdomain is www. |
| **Second Level Domain\*** | Other examples include: selinc amazon ebay nytimes |
| **Top Level Domain\*** | Other common examples include: .net .gov .edu .info |
| **Country Code TLD** | Not needed for domains in the US. Other examples: .cn .uk .eu .ru |
| **Path** | The path and query/parameters portion of the URL tell the Web server which resource is being requested. |
| **Directory** | Usually ends with a / but not always. |
| **Filename** | Often not seen in URL paths. It provides file extension information. |
| **Query/Parameters** | The format is name=value. This is used for many things but it often changes the content of the resource returned. |
| **Fragment/Anchor** | Refers to a part of the page returned. This is usually ignored by crawlers. |

# Creating a Crawl Job

There are two types of crawl jobs: new crawls and re-crawls.

## New Crawls

New crawls create a completely new repository for storing crawled content and metadata. New crawls can be created either with a blank initial configuration or they can be based off of another crawl job’s configuration. Specifying another crawl job as its base copies the configuration information and seeds only. No crawled data or meta-data is shared.

## Re-Crawls

Re-crawls are always based off of a previous crawl job. The configuration of the previous crawl is copied to the re-crawl job so that it can be altered before the re-crawl starts. Re-crawls only download new resources if the web servers identify them as newer than the previously downloaded resources. All status metrics are reset for a re-crawl but the crawl may be faster because it is only downloading new resources or refreshing documents that have changed. There are two types of re-crawl jobs: overwrite and versioned.

* **Overwrite:** Overwrite re-crawl jobs reduce storage costs but only the latest version of the crawled content is preserved. The crawled content of the crawl job that the re-crawl is based off of is overwritten by the latest downloaded version.
* **Versioned:** Versioned re-crawl jobs preserve the content found by previous crawls. The new or updated content found by the re-crawl is stored as a new version of the document.

# Crawl Job Configuration

## Crawl Scope

The settings in this section are used to determine which URLs should be crawled.

### Include/Exclude Rules

There are four lists of rules that are used to determine which URLs should be included or excluded from the crawl. Atrax only crawls URLs using the HTTP and HTTPS protocols so specifying the protocol in URL pattern expressions is not necessary.

The following function determines if a URL will be crawled. Operator precedence is from right to left. For instance if a URL is matched by a rule in the Excluded URLs list and by a rule in the Included URLs list then it will be excluded from the crawl.

[Included Domains] – [Excluded Domains] + [Included URLs] – [Excluded URLs]

#### Included Domains

The user must specify at least one or more domains to crawl. If any portion of a URL’s domain contains any of the items in this list then it is considered an in-scope URL unless it is excluded by one of the rules in Excluded Domains or Excluded URLs. Rules in this list cannot contain wild cards.

#### Excluded Domains

There are some cases where URLs may be included in the crawl based on one of the rules in the Included Domains list but the user wants to exclude a set of more specific domains from being crawled. URLs that match rules in the Excluded Domain list are excluded from the crawl even if they are matched by one of the rules in the Included Domain list. Rules in this list cannot contain wild cards.

#### Included URLs

There are still other cases where URLs may match rules in both the Included Domains and Excluded Domains list and the user still wants them to be crawled. These are typically URLs that have an excluded domain but all URLs that are within a given path should still be considered in-scope. Wild card characters can be used to create a pattern expression for matching such URLs. For example:

www.example.com/products/\*

#### Excluded URLs

The Excluded URLs list can be used to exclude URLs that match a pattern even if they are considered in-scope by all other rules. For example:

[www.example.com/products/\*?color=Black](http://www.example.com/products/*?color=Black)

### Finding URLs to exclude

Atrax Keeper provides some analysis tools that can be very helpful when looking for ways to reduce the crawl job size.

#### URL Analysis

Atrax keeps a database of every URL detected in the crawl. The user may view which paths contain the most URLs and then check to see if those pages are of enough value to continue crawling URLs in the path. The most common paths across the entire crawl or in each domain can be shown and the user may create exclude URL patterns from this information. This is very helpful for finding infinite webs that that may cause the crawler to crawl one portion of a site indefinitely.

#### robots.txt

The main purpose of the robots.txt file is to help crawlers avoid parts of the site that may not of be of substantial value to search engines. Often times, crawling these parts of a site can cause a crawler to enter an infinite web and crawl forever.

Atrax Keeper lists all domains that are in the crawl so far and allows the user to view the robots.txt files and chose whether or not to crawl the parts of the site listed there in. Users may also do this for domains that have not yet been discovered by the crawler by typing in the domain name.

This method of adjusting crawl scope is only available when robot exclusion compliance is disabled.

### Robot Exclusion Compliance

There are various ways that a web site can tell Web crawlers that it prefers to not have certain parts of the web site crawled. If the robot exclusion compliant setting is enabled the crawler will comply with the web site’s wishes otherwise it will ignore these mechanisms and crawl the URLs anyway.

### URL Transformations

Often times it is helpful to apply transformations to a URL in order to put it in a more normal form or in cases where the URL hints at the existence of another resource and applying the transformation will allow the crawler to access the resource. For instance a URL may contain “print=True” in its query/parameters section. This is a common way to tell the web server that the browser is requesting the printer friendly version of a web page. Atrax is more interested in the original web page so a transformation can be used to remove this parameter from the URL in order to get the web page’s URL. Many transformation rules can be listed.

To date Atrax only supports the remove parameter transformation.

#### Remove Parameter

The remove parameter transformation works like the example above. Each transformation rule requires two settings.

* **URL Pattern:** A pattern expression used to specify which URLs should have the transformation applied. A single wild card can be used to specify that the transformation should be attempted on all URLs.
* **Parameter Pattern:** The parameter name pattern. For example if a user wanted remove parameters whose names were “prnt” or “print” then “pr\*nt” can be specified for this pattern.

## Additional Settings

### Name

The user must provide a name for the crawl job. Crawl job names are limited to 256 characters. Only alphanumeric characters plus hyphens (-) and underscores (\_) are allowed. The date the job is first started is appended to the job name when it starts and that becomes the actual name of the crawl job.

### Alias

Any crawl job may have an alias which can also be used to refer to a crawl job. This is helpful if there are many crawl jobs whose name differs only by the date and the user wants only the latest job to have the alias. Other tools that refer to the crawl job can use its alias and therefore always refer to the latest job.

### Web Proxy

All web requests are routed through a proxy in order to handle crawler throttling and anonymity. If no proxy is specified here then Atrax will use its own proxy. Atrax’s default proxy provides the highest reasonable level of anonymity possible. If the user chooses to use a different proxy they can specify one by providing an address and port number.

### Throttling

Atrax by default attempts to crawl a site as fast as possible. Its sophisticated proxy handles this in such a way that minimizes the chances of being blocked by a web server. Alternatively, a user can specify a maximum number of requests that are allowed per minute per server. Setting an upper limit can potentially slow the crawl down but it may also reduce the burden that Atrax puts on web servers and reduce the chances of the site owner paying extra attention to it.

### Seed URLs

All crawl jobs need an initial list of URLs as a starting point for beginning the crawl. A list of seed URLs may be uploaded from a file. Re-crawls do not need a seed list because they are seeded from the previous crawl but additional seed URLs may be provided.

### Automatic Re-crawl

When a crawl job completes it can optionally be automatically restarted in order to ensure that the content is always up to date. In this case a new crawl job is created with the same name as the original crawl job but with a different date. It is automatically started upon creation. If the original crawl job has an alias specified the new re-crawl job will take this alias for itself. The user may choose to have the system do either an overwrite re-crawl or a versioned re-crawl.

The re-crawl job will not start until 24 hours after the original crawl job started. This is to ensure that a site is not crawled more than once a day.

### User Access

There are three levels of access for each crawl job. One or more users in the corporate Active Directory domain can be granted these levels of access. Only crawl job administrators and crawl job viewers may log into Atrax Keeper. Once logged in they may only see information regarding the crawl jobs for which they are either an administrator or a viewer.

* **Crawl Job Administrator:** Crawl job administrators may perform any action affecting a crawl job for which they are a crawl job administrator. The creator of a crawl job is implicitly given this level of access.
* **Crawl Job Viewer:** Users in this category may view but not alter the configuration and status of the crawl job in Atrax Keeper.
* **Crawl Job Consumer:** Users with only consumer level access may not log into Atrax Keeper. The purpose of this access level is to allow users to view or search the crawled data using other applications such as Nereus and Mimeo.

#### Crawl Job Creator

The Crawl Job Creator access level is more global than the crawl job specific access levels. Crawl job creators may create crawl jobs. They may only see the existence of crawl jobs they have created or for which they are administrators or viewers. Only an Atrax system administrator may grant a user this level of access. This is not part of the crawl job configuration.

### Crawl Job Resources

Each crawl job can be provisioned with varying amounts of resources. This affects how quickly the crawl completes. Typically the more resources are available for the crawl the sooner the crawl will complete and the more it will cost per time period. Theoretically the total cost of the crawl will be the same no matter how long it takes. It is very difficult to determine how long it will take or how much it will cost for a crawl job to complete but users can get a sense of its progress by looking at the crawl job status provided by Atrax Keeper.

Atrax Keeper makes managing costs and resources easy by providing a single cost per month setting. The initial value of this setting is $100 per month. This value does not include the cost of storing the crawled content. The cost of storing the crawled content is usually much less than the cost of running the crawler.

Users can use the information they get from the crawl job status provided by Atrax Keeper to help determine if they want to increase/decrease the resources or change the job’s scope with consideration to their timeline and budget.

### Index Content

If index content is enabled a search index will be created for the crawled content. The index is updated as new content is downloaded by the crawler.

### Store Content

Storing crawled content can be enabled or disabled. In most cases users will want to store the crawled content so that it can searched be viewed. If index content is enabled but store crawled content is disabled then the content will be search able but won’t be viewable other than on the live site.

### Offline Browsing

There are two ways that a crawled page may be viewed: online and offline. Online viewing means that the users is viewing the page live directly from the Web. Offline viewing means that the users is viewing what the page looked like when it was crawled and stored in the content repository.

Atrax provides two types of offline browsing: Fully-anonymous and non-anonymous. Fully anonymous offline browsing means that the sites will have no record of the user viewing a page. This is because all of the resources that the page relies on are downloaded along with the page. The live web server never needs to be contacted. In order to improve the quality of the offline browsing experience, it is recommended that the Robot Exclusion Compliance be disabled. Fully-anonymous offline browsing is provided by the Mimeo service.

Non-anonymous offline browsing displays the page approximately how it appeared with it was crawled and indexed. Non-indexable resources that the page relies on are still fetched from the live site. This leaves a “footprint” on the site. Non-anonymous offline browsing reduces the time required to crawl the site and the amount of space required to store the downloaded content because only content that indexable (searchable) is downloaded and stored.

# Crawl Job Status

The following status information is displayed for a given crawl job:

## State

* + Current State: Running / Paused / Faulted / Complete
  + Date and time first started
  + Hours since first start
  + Total hours running
  + Percent up-time
  + Start/pause history
  + Fetcher count (The number of machines that are working on the crawl job)

## Progress

* + Total number of resources crawled
  + Repository size in GB
  + Histogram of the number of URLs crawled minute/hour/day
  + Histogram of the number of URLs skipped per minute/hour/day
  + Histogram of frontier size per minute/hour/day
  + Histogram of number of URLs added vs removed to/from the frontier per minute/hour/day

## Costs

* + Current monthly billing period start date
  + Actual cost of crawling since start of current monthly billing period
  + Actual cost of storage since start of current monthly billing period
  + Total actual costs since start of current monthly billing period
  + Histogram of actual cost of crawling per day/week/month
  + Histogram of actual cost of storage per day/week/month
  + Histogram of total actual costs per time day/week/month

## Logs

* + The last 10 log entries emitted by the crawler for the crawl job

# Starting and Pausing Crawl Jobs

Any crawl job can be started or paused with the click of a button. Additional confirmation is required for both starting and pausing crawl jobs.

# Clean-up Actions

## Delete

This action deletes the entire crawl job including all crawled content and meta-data. The crawl job may not be running when performing this action. The user must enter their password in order to confirm this action.

## Purge

This action deletes everything except the crawl job configuration. The crawl job may not be running when performing this action. The user must enter their password in order to confirm this action.

## Purge Out of Scope

This action deletes all out of scope content. This can be done after changing a crawl job’s scope rules. The crawl job may be running when performing this action. The user must enter their password in order to confirm this action.

## Add Skipped to Frontier

This should be done after changing a crawl job’s scope rules. It is possible that the crawler had skipped URLs that are not considered in-scope. Adding the skipped URLs to the frontier will cause Atrax to reconcile the scope changes. The crawl job may be running when performing this action.