Migration from Heritrix (under construction)

If you are migrating from Heritrix into Aspider, there are some general steps you should follow to cover all mayor configuration differences between the two crawlers.

In each case the Aspider configuration options are followed in order and details on how to determine what to configure there are provided based on the type of Heritrix configuration you have.

When using standard Aspire configuration for Heritrix

This section is under construction

When using a custom crawler-beans file for Heritrix

1. Seed URLs
    - Copy seed URLs into Aspider
2. Crawl Scope
    - If the crawler beans contains a **org.archive.modules.deciderules.surt.NotOnDomainsDecideRule** bean
        - Set this option to ***Domain only***
    - If contains a **org.archive.modules.deciderules.surt.NotOnHostsDecideRule** bean instead
        - Set this option to ***Host Only***
    - If there is not any of the previous beans
        - Set this option to **Everything**
3. User Agent
    - If any **userAgentTemplate** property is set in the crawler beans
        - Copy the user-agent into this option
4. Crawl Depth
    - This is as a bean in the crawler beans:

```
<bean class="org.archive.modules.deciderules.TooManyHopsDecideRule">
 <property name="maxHops" value="HOPS-NUMBER" />
</bean>
```

        - Copy the "HOPS-NUMBER" into this option
5. Max Links per page
    - Find the **maxOutLinks** property in the crawler beans and copy its value into this option
    - If there is no **maxOutLinks** leave the default as 6000
6. Max content size (in bytes)
    - Find the **maxLengthBytes** property in the crawler beans
        - If the value is greater than 0, copy that value to this option
        - If the value is 0, it means is unlimited so you could use ***9223372036854775807*** (max long size) as the maximum value
        - If you don't see this property in the crawler beans, configure as you think it is appropriate
7. Case Sensitivity URLs
    - By default Heritrix checks uniqueness of URLs by lowercasing them before comparing.
        - If your site URLs are NOT case sensitive then you should uncheck this option
        - If it is case sensitive, leave this option checked
8. Deletes Policy
    - If the **Configure Incremental Indexing** option is checked in the Heritrix content source
        - You need to decide which delete policy to use, as Aspider only allows one at the time:
            - Days before delete (sends the deletes after an specific number of days has passed since the last time the URLs were accessed)
                - In Aspider select ***Time Based*** and use the same value as in Heritrix
            - Iterations before delete (sends the deletes after an specific number of incremental crawls since the last time the URLs were accessed)
                - In Aspider select ***After X Incrementals*** and use the same value as in Heritrix
    - If the option is not checked, Leave the Aspider option as ***Immediate*** (which means sending deletes as soon as a URL is not accessed during an incremental crawl)
9. Customize Connection Timeouts
    - If you see a **soTimeoutMs** property in the crawler beans, copy the value into the ***Socket Timeout*** option in Aspider
    - If your network is having troubles and you see a lot of connection timeout errors, then you should also increase the ***Connection Timeout*** and ***Connection Request Timeout***.
10. Connection Throttling
    - This is how fast Aspider will be able to fetch pages
    - There is no direct mapping to this from Heritrix.

- The ***Max Urls per Hostname per minute*** option must be configured according to your network architecture and web server load.
    - You should ask the web server owners what is a good number to configure here.

11. Obey Robots.txt & Obey Robots Meta Tags
    - Find the **robotsPolicyName** property in the crawler beans
        - If it is set to **"obey"** or **"classic"** leave "***Obey Robots.txt***" checked in Aspider
        - If it is set to **"ignore"** uncheck the "***Obey Robots.txt***" option only
    - Find a bean for **"RobotsHonoringPolicy"** in the crawler beans
        - If the type is set to **"IGNORE",** uncheck both options
        - If the type is set to **"CLASSIC",** leave both options checked
    - If no configuration for robots is found in the crawler beans, leave both options checked
12. Trust All Certificates
    - Find **sslTrustLevel** in crawler beans
        - If **OPEN,** check the aspider option
        - Otherwise leave unchecked
13. Use Proxy
    - Find the **"httpProxyHost"** option in the crawler beans
        a. Copy the **"httpProxyHost"** into the "***Proxy Host***" option in Aspider
        b. Copy the **"httpProxyPort"** into the "***Proxy Port***" option in Aspider
14. Use Authentication

    If your site requires authentication the very first step you need to do is to identify the type of authentication to use.This can be found in the crawler-beans file by looking for the bean with id of "**credential**".There can be two different beans:

    a. **HtmlFormCredential**

    Used for cookie based authentication, generally using a login page and POST requests to authenticate.

    If this is the authentication used in your crawler-beans file, you should add a "**Cookie Based (HTML Forms)**" mechanism in Aspider.

        i. In the **Login URL** field you should copy the address of the Login Page of your site
        ii. In the **Form Element Path** field you should inspect into your login page structure to determine where the form can be found, for example, if your login page HTML looks like this:

```
<html>
<head>..</head>
<body>
  <div id="content">
    <form id="login" method="POST" action="login.php">
      <label><b>Username</b></label>
      <input type="text" placeholder="Enter Username" name="uname" required>
      <label><b>Password</b></label>
      <input type="password" placeholder="Enter Password" name="psw" required>
      <input type="hidden" name="clientToken" value="wAAAMLCwkJCQgAAAGJiYoKCgpKSkiH">
      <button type="submit">Login</button>
    </form>
  </div>
</body>
</html>
```

    your Form Element Path should look like:

```
/html/body/div[@id="content"]/form[@id="login"]
```

    if you are using 3.1.0.6 version or later, a CSS Selector should be use and it should look like:

```
#login
```

    b. **HttpAuthenticationCredential**

    There can be different types of authentication such as BASIC, DIGEST or NTLM. These kinds of authentication mechanisms work on the connection level, so the server returns a 401 status code challenging the client for credentials. The crawler beans mapping would go like:

Crawler beans property	Aspider field
domain	host
realm	realm
login	domain + user
password	password

15. Include patterns

    - Find the **"MatchesListRegexDecideRule"** which have the **"decision"** property of **"ACCEPT"**
        - Copy the regex patterns into the "***Include Patterns***" section in Aspider

- Any URL that does not match with any pattern in this list will be EXCLUDED
- Any pattern set here will overwrite the "***Crawl Scope***"
    - For example adding ***.**** is like having Crawl Scope ***"Everything"***
- If you add a pattern here that doesn't match the seed URLs the crawler won't be able to get anything, so make sure your seed URLs are covered

16. Exclude patterns

- Find the**"MatchesListRegexDecideRule"** which have the **"decision"** property of **"REJECT"**
    - Copy the regex patterns into the "***Exclude patterns***" section in Aspider
- Any URL matching a pattern in this list will NOT be processed in the workflow so it will NOT be indexed.
- These rules will also prevent the crawler from discovering links from any URL that matches a pattern in this list (unless "***scan excluded items***" is checked).

17. Scan Excluded Items

- This option will allow to extract (scan) links also from excluded items.
- If there is a pattern in the Heritrix "***Index Exclude Patterns***" but not in the crawler beans **REJECT MatchesListRegexDecideRule,** you may want to check this option
- If there is a subset of pattern of rejects that you don't want to extract links (scan) you should add those patterns into the "***Do not follow patterns***" option

18. Reject Images / Videos / Javascript /CSS

- If you have the following bean in your crawler beans

```
<bean class="org.archive.modules.deciderules.MatchesListRegexDecideRule">
   <property name="decision" value="REJECT"/>
   <property name="listLogicalOr" value="true"/>
   <property name="regexList">
      <list>
         <value>.*\.js.*</value>
         <value>.*\.css.*</value>
         <value>.*\.swf.*</value>
         <value>.*\.gif.*</value>
         <value>.*\.png.*</value>
         <value>.*\.jpg.*</value>
         <value>.*\.jpeg.*</value>
         <value>.*\.bmp.*</value>
         <value>.*\.mp3.*</value>
         <value>.*\.mp4.*</value>
         <value>.*\.avi.*</value>
         <value>.*\.mpg.*</value>
         <value>.*\.mpeg.*</value>
     </list>
   </property>
</bean>
```

- Then you should check this option in Aspider
- This will exclude any multimedia files from getting processed by the crawler.