

Aspire Shell

Aspire Shell is a Groovy driven console for configuring, testing, and executing any document processing tasks in Aspire.

On this page:

- [Start up Aspire shell](#)
- [Use Aspire Shell](#)
 - [Groovy in Aspire shell](#)
 - [Load components](#)
 - [Create jobs in Aspire shell](#)
 - [Use Groovy pipelines](#)
 - [Aspire Shell commands](#)
 - [Run applications](#)
- [Start content sources](#)
- [Enable Aspire HTTP web server](#)

Related pages:

- [Launch control](#)
- [Directory structure](#)
- [In-Depth administration](#)

Start Up Aspire Shell

To start Aspire, execute the "aspiresh.sh" or "aspiresh.bat" script in the "bin" directory.

Linux

Set your working directory to your Aspire distribution and then execute "bin/aspiresh.sh" (make sure it has execute permission for your user):

```
$ cd my-aspire/target/my-aspire-1.0-SNAPSHOT-distribution.dir
$ bin/aspiresh.sh

*****
*
* ASPIRE BOOTLOADER
*
* Bundle id : 10
*
* Location  : file:bundles/boot/aspire-bootloader-2.2.2-SNAPSHOT.jar
*

Maven update policy: always

Settings (org.apache.felix.webconsole.internal.servlet.OsgiManager): {password=admin, manager.root=/osgi,
username=admin}
Maven update policy: always
Aspire Console Mode
Aspire>
```

Microsoft Windows from the Command Prompt

Set your working directory to your Aspire distribution and then execute "bin\aspiresh.bat":

```

$ cd my-aspire\target\my-aspire-1.0-SNAPSHOT-distribution.dir
$ bin\aspire.bat
Removing Felix-Cache and AppBundle-Cache directories

*****
*
* ASPIRE BOOTLOADER
*
* Bundle id : 10
*
* Location  : file:bundles/boot/aspire-bootloader-2.2.2-SNAPSHOT.jar
*

Maven update policy: always

Settings (org.apache.felix.webconsole.internal.servlet.OsgiManager): {password=admin, manager.root=/osgi,
username=admin}
Maven update policy: always
Aspire Console Mode
Aspire>

```

Use Aspire Shell

Groovy in Aspire Shell

One of the most powerful features in Aspire Shell is the ability to parse and evaluate Groovy Expressions on the fly, just as [GroovyShell](#) does. But not only normal Groovy Expressions but also [Groovy Pipelines](#) expressions.

```

Aspire> 5 + 6
11
Aspire> println "hello world, this is Aspire Shell"
hello world, this is Aspire Shell

```

Load components

You can load any component by just typing its configuration:

Example:

```

Aspire> <component name="FetchUrl" subType="default" factoryName="aspire-fetch-url" />
Aspire> <component name="ExtractText" subType="default" factoryName="aspire-extract-text" />
Aspire>

```

You can also load components with multiple line configurations:

```

Aspire> <component name="PostHTTP" subType="default" factoryName="aspire-post-http">
Aspire:   <postUrl>http://localhost:8983/solr/update</postUrl>
Aspire:   <postXsl>config/xsl/aspireToSolr.xsl</postXsl>
Aspire:   <okayResponse><![CDATA[<int name="status">0</int>]]></okayResponse>
Aspire: </component>
Aspire>

```

Each component loaded can be accessed by its name:

```

Aspire> FetchUrl
<component name="FetchUrl" subType="default" factoryName="aspire-fetch-url"/>

Aspire> ExtractText
<component name="ExtractText" subType="default" factoryName="aspire-extract-text"/>

Aspire> PostHTTP
<component name="PostHTTP" subType="default" factoryName="aspire-post-http">
  <postUrl>http://localhost:8983/solr/update</postUrl>
  <postXsl>config/xsl/aspireToSolr.xsl</postXsl>
  <okayResponse><int name="status">0</int></okayResponse>
</component>

Aspire>

```

Create jobs in Aspire Shell

From Aspire Shell you can create your own jobs in order to process them through your own Groovy Pipelines.

Example:

```

Aspire> myJob = <doc>
Aspire:   <fetchUrl>http://www.searchtechnologies.com</fetchUrl>
Aspire:   </doc>
** localhost:50505/2013-06-04T15:42:28Z/0 **
<doc>
  <fetchUrl>http://www.searchtechnologies.com</fetchUrl>
</doc>

Aspire>

```

Use Groovy Pipelines

If you have already loaded some components and created a job to process, you can use groovy pipelines to process it. (For more details on groovy pipelines go to [Groovy Pipelines](#))

Example:

```

Aspire> myJob | FetchUrl | ExtractText
** localhost:50505/2013-06-04T19:03:29Z/0 **
<doc>
  <content>
    ...
    //CONTENT EXTRACTED
    ...
  </content>
  <extension source="FetchURLStage">
    <field name="status">HTTP/1.1 200 OK</field>
    <field name="Cache-Control">private</field>
    <field name="Content-Length">24233</field>
    <field name="Content-Type">text/html; charset=utf-8</field>
    <field name="Server">Microsoft-IIS/7.5</field>
    <field name="X-AspNet-Version">2.0.50727</field>
    <field name="Set-Cookie">ASP.NET_SessionId=rs31fw3r23u3j42unooa5tjl; path=/; HttpOnly</field>
    <field name="X-Powered-By">ASP.NET</field>
    <field name="Date">Tue, 04 Jun 2013 19:21:19 GMT</field>
  </extension>
  <extension source="ExtractTextStage">
    <field name="Content-Location">http://www.searchtechnologies.com</field>
    <field name="Content-Encoding">ISO-8859-1</field>
    <field name="resourceName">http://www.searchtechnologies.com</field>
    <field name="google-site-verification">jP1bIfjuuyZUYfTkYc_O6ZlTHxCm07voTDcMk72Z8oQ</field>
  </extension>
  <title source="ExtractTextStage/title">The Enterprise Search Implementation Experts</title>
  <protocol source="FetchURLStage/protocol">http</protocol>
  <keywords source="ExtractTextStage/keywords">Enterprise Search, Search Engine Experts, Consulting</keywords>
  <host source="FetchURLStage/host">www.searchtechnologies.com</host>
  <fetchUrl>http://www.searchtechnologies.com</fetchUrl>
  <description source="ExtractTextStage/description">Search Technologies is the largest IT services company dedicated to enterprise search implementation, consulting, and managed services. Our expertise covers all leading search products, and all aspects of search applications.</description>
  <encoding source="FetchURLStage/encoding">utf-8</encoding>
  <contentType source="ExtractTextStage/Content-Type">application/xhtml+xml</contentType>
  <httpResponse code="200" source="FetchURLStage">OK</httpResponse>
  <mimeType source="FetchURLStage/mimeType">text/html</mimeType>
</doc>

Aspire>

```

Aspire Shell commands

Besides groovy expressions you can also use some special commands in Aspire Shell. Those commands are listed in the following table:

Command	Parameters	Description
cd	<directory>	Change the working directory for the 'dir' groovy pipeline function. (For more details click here)
clear	none	Clears the command buffer and discard it. Any multiple line command input will be escaped. It can be invoked by the shortcut CTRL-D
cls	none	Clears the console output
components	none	Shows the components loaded inside the current Aspire shell application
createJob	XML string of an AspireObject	Creates a job based on the XML string of an AspireObject.
createJobFromFile	<file path>	Creates a job based on the content of file. The file passed as a parameter can be either XML or Json
createJobFromJsonFile	<file path>	Creates a job based on the content of file. The file passed as a parameter should be Json
createJobFromXMLFile	<file path>	Creates a job based on the content of file. The file passed as a parameter should be either XML
exit	none	If performed in felix mode, it returns to Aspire Shell mode, else exit AspireShell and then shutdown felix.

felix	none	Enter felix mode to execute felix commands. (No AspireShell commands will be available here)
help	none or <command-name>	Shows the help for Aspire shell. If no arguments then it shows all the available commands, if specified a command-name as argument, it shows the description for that command.
load	<application-xml-file>	Loads the specified application into Aspire. Any previous Aspire shell application will be unloaded. Cleans up every non saved groovy expression.
pwd	none	Shows the working directory for the 'dir' groovy function. (For more details click here)
reset	none	Removes the current Aspire shell application and load it again. Also and cleans every non saved groovy expression.
run	<application-xml-file>	Run the groovy expressions saved in the specified application file. Every loaded components and groovy expressions will be replaced by those in the specified application file.
save	<application-xml-file>	Saves the current Aspire shell application as XML into the specified file. Every groovy expression will be saved as a groovy pipeline, ready to be executed.
shutdown	none	Exit AspireShell and shutdown felix.
subJobs	List list, String elementName, Closure closure	Iterates through list, creating a job for each item using the element name specified (or fetchUrl if the element name is not given) and passes the job to the closure (a Groovy pipeline). Sub jobs are processed in parallel, with the command waiting for the for the sub jobs to finish processing
subJobsSerial	List list, String elementName, Closure closure	Iterates through list, creating a job for each item using the element name specified (or fetchUrl if the element name is not given) and passes the job to the closure (a Groovy pipeline). Sub jobs are processed in series, with the command waiting for the for the sub jobs to finish processing

Command usage

components

```
Aspire> components
FetchUrl      com.searchtechnologies:aspire-fetch-url:2.2.2 / default
ExtractText   com.searchtechnologies:aspire-extract-text:2.2.2 / default
PostHTTP      com.searchtechnologies:aspire-post-http:2.2.2 / default
Aspire>
```

createJob

```
Aspire> myJob = createJob('<doc><url>http://www.searchtechnologies.com</url></doc>')
** 10.10.21.34:50505/2014-06-23T20:53:32Z/0 **
<doc>
  <url>http://www.searchtechnologies.com</url>
</doc>
Aspire>
```

You can also skip the createJob command and create a job directly (which will call the createJob method in the background):

```
Aspire> myJob = <doc><url>http://www.searchtechnologies.com</url></doc>
** 10.10.21.34:50505/2014-06-23T20:53:32Z/0 **
<doc>
  <url>http://www.searchtechnologies.com</url>
</doc>
Aspire>
```

createJobFromFile

```

Aspire> myJob = createJobFromFile "/temp/doc.xml"
** 192.168.2.106:50505/2016-06-09T10:37:07Z/0 **
<doc>
  <url>http://www.searchtechnologies.com/xml</url>
</doc>
Aspire>
Aspire> myJob = createJobFromFile "/temp/doc.json"
** 192.168.2.106:50505/2016-06-09T10:37:07Z/1 **
<doc>
  <url>http://www.searchtechnologies.com/json</url>
</doc>
Aspire>

```

createJobFromJsonFile

```

Aspire> myJob = createJobFromJsonFile "/temp/doc.json"
** 192.168.2.106:50505/2016-06-09T10:37:07Z/2 **
<doc>
  <url>http://www.searchtechnologies.com/json</url>
</doc>
Aspire>

```

createJobFromXMLFile

```

Aspire> myJob = createJobFromXMLFile "/temp/doc.xml"
** 192.168.2.106:50505/2016-06-09T10:37:07Z/3 **
<doc>
  <url>http://www.searchtechnologies.com/xml</url>
</doc>
Aspire>

```

subJobs

```

Aspire> <component name="FetchUrl" subType="default" factoryName="aspire-fetch-url" />
INFO [/AspireShellApplication/Main]: Starting component: FetchUrl (aspire-fetch-url)
INFO [aspire]: Loaded factory com.searchtechnologies.aspire:aspire-fetch-url:3.0-SNAPSHOT (jar file file:/C:/Users/administrator/.m2/repository/com/searchtechnologies/aspire/aspire-fetch-url/3.0-SNAPSHOT/aspire-fetch-url-3.0-SNAPSHOT.jar) using repository MavenRepository[C:\Users\administrator\.m2/repository]
INFO [/AspireShellApplication/Main]: Installed component: /AspireShellApplication/Main/FetchUrl
Aspire>
Aspire> subJobs(["http://www.seznam.cz", "http://www.chessbase.com", "http://www.searchtechnologies.com"],
"fetchUrl", {it | FetchUrl >> "c:\\temp\\someFile.txt"})
(void)
Aspire>

```

The above results in the following output to *someFile.txt*

```
<job id="192.168.2.106:50505/2016-06-10T09:52:25Z/0/0" time="2016-06-10T09:57:23Z">
<doc>
<fetchUrl>http://www.seznam.cz</fetchUrl>
<httpResponse code="302" source="FetchURLStage">Moved Temporarily</httpResponse>
<protocol source="FetchURLStage/protocol">http</protocol>
<host source="FetchURLStage/host">www.seznam.cz</host>
<mimeType source="FetchURLStage/mimeType">text/html</mimeType>
<extension source="FetchURLStage">
<field name="status">HTTP/1.1 302 Moved Temporarily</field>
<field name="Server">nginx</field>
<field name="Date">Fri, 10 Jun 2016 09:57:23 GMT</field>
<field name="Content-Type">text/html</field>
<field name="Content-Length">154</field>
<field name="Connection">close</field>
<field name="Location">https://www.seznam.cz/</field>
</extension>
</doc>
</job>

<job id="192.168.2.106:50505/2016-06-10T09:52:25Z/0/1" time="2016-06-10T09:57:23Z">
<doc>
<fetchUrl>http://www.chessbase.com</fetchUrl>
<httpResponse code="200" source="FetchURLStage">OK</httpResponse>
<protocol source="FetchURLStage/protocol">http</protocol>
<host source="FetchURLStage/host">www.chessbase.com</host>
<mimeType source="FetchURLStage/mimeType">text/html</mimeType>
<encoding source="FetchURLStage/encoding">utf-8</encoding>
<redirectUrl source="FetchURLStage/redirectUrl">http://en.chessbase.com/</redirectUrl>
<extension source="FetchURLStage">
<field name="status">HTTP/1.1 200 OK</field>
<field name="Cache-Control">no-cache</field>
<field name="Pragma">no-cache</field>
<field name="Content-Type">text/html; charset=utf-8</field>
<field name="Expires">-1</field>
<field name="X-UA-Compatible">IE=edge</field>
<field name="Date">Fri, 10 Jun 2016 09:57:26 GMT</field>
<field name="Content-Length">41985</field>
</extension>
</doc>
</job>

<job id="192.168.2.106:50505/2016-06-10T09:52:25Z/0/2" time="2016-06-10T09:57:23Z">
<doc>
<fetchUrl>http://www.searchtechnologies.com</fetchUrl>
<httpResponse code="200" source="FetchURLStage">OK</httpResponse>
<protocol source="FetchURLStage/protocol">http</protocol>
<host source="FetchURLStage/host">www.searchtechnologies.com</host>
<mimeType source="FetchURLStage/mimeType">text/html</mimeType>
<encoding source="FetchURLStage/encoding">utf-8</encoding>
<extension source="FetchURLStage">
<field name="status">HTTP/1.1 200 OK</field>
<field name="Cache-Control">private</field>
<field name="Content-Type">text/html; charset=utf-8</field>
<field name="Server">Microsoft-IIS/8.5</field>
<field name="X-AspNetMvc-Version">4.0</field>
<field name="X-AspNet-Version">4.0.30319</field>
<field name="X-Powered-By">ASP.NET</field>
<field name="Date">Fri, 10 Jun 2016 09:57:30 GMT</field>
<field name="Content-Length">28313</field>
</extension>
</doc>
</job>
```

```
Aspire> <component name="FetchUrl" subType="default" factoryName="aspire-fetch-url" />
INFO [/AspireShellApplication/Main]: Starting component: FetchUrl (aspire-fetch-url)
INFO [aspire]: Loaded factory com.searchtechnologies.aspire:aspire-fetch-url:3.0-SNAPSHOT (jar file file:/C:/Users/administrator/.m2/repository/com/searchtechnologies/aspire/aspire-fetch-url/3.0-SNAPSHOT/aspire-fetch-url-3.0-SNAPSHOT.jar) using repository MavenRepository[C:\Users\administrator\.m2/repository]
INFO [/AspireShellApplication/Main]: Installed component: /AspireShellApplication/Main/FetchUrl
Aspire>
Aspire> subJobsSerial(["http://www.seznam.cz", "http://www.chessbase.com", "http://www.searchtechnologies.com"], "fetchUrl", {it | FetchUrl >> "c:\\temp\\someFile.txt"})
[http://www.seznam.cz, http://www.chessbase.com, http://www.searchtechnologies.com]
Aspire>
```

Output to *someFile.txt* is identical to that for the **subJobs** command

reload

```
Aspire> reload
Aspire>
```

reset

```
Aspire> reset
Aspire>
```

save

```
Aspire> println "hello world, this is AspireShell"
hello world, this is AspireShell
Aspire> save config/savedApp.xml
Aspire>
```

Load

```
Aspire> load config/savedApp.xml
Aspire>
```

run

```
Aspire> run config/savedApp.xml
hello world, this is AspireShell
Aspire>
```

cd & pwd

```
Aspire> cd config
Aspire> pwd
C:\\distributions\\my-aspire\\target\\my-aspire-2.2.2-distribution.dir\\config
Aspire> cd ..
Aspire> pwd
C:\\distributions\\my-aspire\\target\\my-aspire-2.2.2-distribution.dir
```

clear

```
Aspire> <component name="PostHTTP" subType="default" factoryName="aspire-post-http">
Aspire: clear
Aspire>
```


You can also use the *clear* shortcut CTRL-D

```
Aspire> <component name="PostHTTP" subType="default" factoryName="aspire-post-http">
Aspire: <CTRL-D>
Aspire>
```

cls

```
Aspire> cls
Aspire>
```

felix

```
Aspire> felix
Felix> find aspire
START LEVEL 1
  ID   State      Level  Name
[ 11] [Active]    ] [    1] Aspire Application (2.2.2)
Felix> exit
Aspire>
```

shutdown

```
Aspire> shutdown
Aspire> Stopping bundle: 11 - file:/C:/Users/aaguiar/.m2/repository/com/searchtechnologies/aspire-
application/2.2.2/aspire-application-2.2.2.jar
Stopped component factory: aspire-application (bundle 11)
AspireShell: Shutting down...
$
```

Run Applications

You can use Aspire Shell to run at startup applications you have previously saved in your filesystem:

```
$ bin\aspiresh.bat config/savedApp.xml
Removing Felix-Cache and AppBundle-Cache directories

*****
*
* ASPIRE BOOTLOADER
*
* Bundle id : 10
*
* Location  : file:bundles/boot/aspire-bootloader-2.2.2.jar
*

Maven update policy: always

Settings (org.apache.felix.webconsole.internal.servlet.OsgiManager): {password=admin, manager.root=/osgi,
username=admin}
Maven update policy: always
Aspire Console Mode
hello world, this is AspireShell
Aspire> Stopping bundle: 11 - file:/C:/Users/aaguiar/.m2/repository/com/searchtechnologies/aspire-
application/2.2.2/aspire-application-2.2.2.jar
Stopped component factory: aspire-application (bundle 11)
AspireShell: Shutting down...

$
```

You can add the --no-exit option to avoid AspireShell from shutting down after running the application.

```
$ bin\aspiresh.bat config/savedApp.xml --no-exit
Removing Felix-Cache and AppBundle-Cache directories

*****
*
* ASPIRE BOOTLOADER
*
* Bundle id : 10
*
* Location : file:bundles/boot/aspire-bootloader-2.2.2-SNAPSHOT.jar
*

Maven update policy: always

Settings (org.apache.felix.webconsole.internal.servlet.OsgiManager): {password=admin, manager.root=/osgi,
username=admin}
Maven update policy: always
Aspire Console Mode
hello world, this is AspireShell
Aspire>
```

Passing parameters to applications

If required, you can pass parameters to your applications. To variables are available that facilitate this:

Variable	Description
argn	The number of parameters passed to on the command line
argv	A map of the parameters. The first will always be the application file name

Consider the following application xml file:

```
C:\aspire>cat savedApp.xml
<?xml version="1.0" encoding="UTF-8"?>
<application name="AspireShellApplication">
  <components>
    <component name="Main" subType="pipeline" factoryName="aspire-application">
      <components>
      </components>
      <pipelines>
        <pipeline name="defaultPipeline" default = "true">
          <script><![CDATA[
            println "-----";
            println "-----argv   =${argv} -----";
            println "-----file    =${argv[0]} -----";
            println "-----first   =${argv[1]} -----";
            println "-----second  =${argv[2]} -----";
            println "-----third   =${argv[3]} -----";
            println "-----argn    =${argn} -----";
            println "-----";
          ]]></script>
        </pipeline>
      </pipelines>
    </component>
  </components>
</application>
C:\aspire>
```

When run, this application prints out the parameters passed

```
C:\aspire>bin\aspiresh.bat savedApp.xml p1 p2 p3
```

```
*****
*
* ASPIRE BOOTLOADER
*
* Bundle id : 13
*
* Location  : file:bundles/boot/aspire-bootloader-3.0-SNAPSHOT.jar
*

Successfully started application: /AspireShellApplication (location: savedApp.xml)
-----
-----argv  =[savedApp.xml, p1, p2, p3] -----
-----file   =savedApp.xml -----
-----first  =p1 -----
-----second =p2 -----
-----third  =p3 -----
-----argn   =4 -----
-----
```

Start Content Sources

Aspire Shell does not load the autostart section of the settings.xml file, neither load any content sources configured inside **config/content-sources**. To enable the autostart and content sources loading you will need to add the **--load-content-sources** option when starting aspiresh.bat or aspiresh.sh.

```
$ bin\aspiresh.bat --load-content-sources
```

Enable Aspire HTTP web server

Aspire Shell doesn't start the HTTP web server by default. To enable it you need to edit the **config/felix.properties.aspiresh** file, find **org.osgi.service.http.port=-1** and add the port that you want (for example **org.osgi.service.http.port=50505**)