

Publish to GSA Application Bundle

- [Configuration](#)
- [Example Configuration](#)
 - [Simple](#)
- [Edit Xsl](#)
 - [Add metadata field](#)
 - [Change the id URL](#)
 - [Change the display URL](#)
 - [Advanced Edit](#)
- [Output](#)

THIS ITEM IS BEING DEPRECATED.

The Publish to GSA application performs *content feeds* to a GSA of metadata and content of files extracted by Aspire connectors. The feed to the GSA can be customized by editing the XSL transformation file provided by the user.

Publish to GSA Application Bundle	
Factory Name	com.searchtechnologies.aspire:app-publish-to-gsa
subType	default
Inputs	AspireObject from a connector's subjob with metadata and content extracted from a specific file/folder.
Outputs	An XML transformation of the AspireObject sent to the GSA's xmlfeed URL.

Configuration

This section lists all configuration parameters available to configure the Publish to GSA Application Bundle component.

Property	Type	Default	Description
GSANoUrl	boolean	true	Indicates if the publisher must use a Url or build one from the host and port entered.
GSAPort	interger	19900	GSA port where to send the feeds
GSAHost	string	none	GSA hostname or IP adress. e.g. gsa.domain.com
GSAUrl	string	none	Complete Url where the feeds are going to be send. e.g. http://localhost:19900/xmlfeed
makePublic	boolean	false	Makes public all content published to the GSA.
aspireToGSAXsl	string	\${appbundle.home}/config/xsl/aspireToGSA.xsl	Location of the XSL to transform the job data to a GSA feed. See Edit Xsl .
maxResults	int	1000000	(Index dump) How many documents can be fetched by the search engine for the same query
dumpSlices	int	10000	(Index dump) How many documents to fetch per page
urlField	string	displayUrl	(Index dump) Field used to store the url in the search engine
idField	string	id	(Index dump) Field used to store the id in the search engine.

Example Configuration

Simple

```
<application config="com.searchtechnologies.aspire:app-publish-to-gsa">
  <properties>
    <property name="GSANoUrl">true</property>
    <property name="GSAPort">19900</property>
    <property name="GSAHost">localhost</property>
    <property name="makePublic">true</property>
    <property name="debug">true</property>
    <property name="aspireToGSAXsl">${appbundle.home}/config/xsl/aspireToGSA.xsl</property>
  </properties>
</application>
```



Any optional properties can be removed from the configuration to use the default value described on the table above.

Edit Xsl

The default XSL transformation file can be found in [AspireToGSA.xsl](#).

The default transformation XSL file provided by the publisher expects metadata as described in [Connector AspireObject Metadata](#).

Add metadata field

To add a new metadata field extracted by an Aspire Connector add an xsl element under the **<metadata>** tag.

```
<meta name="metafieldNameInGSA">
  <xsl:attribute name="content">
    <xsl:value-of select="metafieldNameFromAspireObject" />
  </xsl:attribute>
</meta>
```

Change the id URL

The **id URL** is the URL the GSA uses to uniquely identify a file in the index. To change the Aspire metadata field being used as the id, change the value of the record's attribute: **url** to the desired field.

```
<xsl:attribute name="url">
```

Change the display URL

The **display URL** is the URL the GSA uses as the result's file location display in the results page. Change the *google:displayurl* meta field content attribute to change this value.

```
<meta name="google:displayurl">
  <xsl:attribute name="content">
    <xsl:value-of select="displayUrlFromAspireObject" />
  </xsl:attribute>
</meta>
```

Advanced Edit

More advanced changes can be accomplished following the [Feeds Protocol Developer's Guide](#) instructions.

Output

The output of the component will be a GSA Feed formatted XML that will be sent through the /xmlfeed URL of the GSA. The output XML of the XSL transformation needs to follow the definition established by the [gsafeed_dtd](#).