

Redis Connection

The **Aspire Redis Connection** opens a connection to a Redis database. When requesting a connection, one is returned from the connection pool if it exists. If there are no available connections in the pool, then a new one will be allocated.

When used as a pipeline stage, the stage will fetch a connection from the connection pool and will store it as a job variable with a custom name defined at initialization time so it can be used by pipeline modules down-stream. It also puts a connection handler so that when the job is closed, the connection will be automatically returned back to the connection pool.

Redis Connection	
Factory Name	com.searchtechnologies.aspire:aspire-redis-connection
subType	default
Inputs	None
Outputs	Job variable (when used as a pipeline stage), holds a Redis connection.

[blocked URL](#)

? Unknown Attachment

Configuration

Element	Type	Default	Description
redisServer	string	localhost	Hostname of the Redis server
redisPort	Integer	6379	Port where the Redis server is hosted.
redisTimeout	Integer	10000	Redis Connection timeout.
database	string	aspire_databases1	Name of the database to use.
variable	string	redisConn	Name of the job variable that will hold the redis connection.
maxRetries	Integer	3	If there is a connection failure, the number of times to retry before failing the action.
metaDB	Integer	-1	Database number in Redis to store a mapping of database names and the corresponding Redis database number. If -1, uses the HASH storing approach in Redis.

Example Configuration

```
<component factoryName="aspire-redis-connection" name="RedisAllDocuments" subType="default">
  <redisServer>myRedisServer</redisServer>
  <metaDB>0</metaDB>
  <database>all_documents</database>
  <variable>redisAllDocuments</variable>
</component>
```

Example Use Within A Pipeline

```
<pipeline name="process-patent" default="true">
  <stages>
    <stage component="FetchUrl" />
    <stage component="LoadXML" />
    <stage component="RedisAllDocuments"/>

    <!-- Now that the connection has been opened and attached to the job, you can
         use the connection inside of Groovy scripts with the "redisAllDocuments" variable.

         In the future, there may be other components which will use it as well.
    -->
  </stages>
</pipeline>
```

Example Use from Within a Groovy scripting component

```
<component name="TestRedis" subType="default" factoryName="aspire-groovy">
  <script>
    def documentTotal = redisAllDocuments.dbSize();

    redisAllDocuments.set("myKey", "myValue");
  </script>
</component>
```

Available methods

- **set()**: Single key/value set.
- **get()**: Single key get.

- **mset()**: Multivalue set
- **mget()**: Multivalue get
- **dbSize()**: Database size
- **close()**: Close connection