

Quick Start

Client-side Cache in DXA

Feb 2017 – SDL Web

The logo features a large, white, stylized asterisk shape in the background. The text 'SDL*' is positioned in the lower right quadrant, with 'SDL' in a dark blue font and the asterisk in a green font.

SDL*

Contents

Client-side caching	3
Information	3
Pre-requisites	3
Steps	Error! Bookmark not defined.
About SDL	8

Client-side caching

Information

SDL Web's Digital Experience Accelerator (DXA) provides a pre-configured solution to incorporate client-side (Content Interaction Library (CIL)) caching to improve site performance. The following steps detail how this can be implemented.

Redis is an in-memory data structure store that is used as a database, cache and message broker. SDL Web 8.5 uses the Redis store for caching (Client-side and Server-side).

Pre-requisites

The following pre-requisites must be met before you can setup DXA client-side caching:

- SDL Web 8.5 (this guide allows you to use <http://sdl-training.com>)
- Digital Experience Accelerator 1.7 (Check out the Quick Start DXA Developer Guide)
- Redis database installed

Installing Redis and Configuring DXA Cache

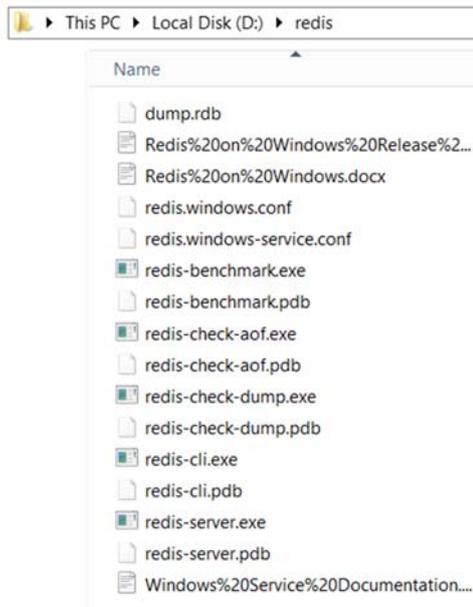
Redis Version 3.0

Redis Desktop Manager Version 0.8.8

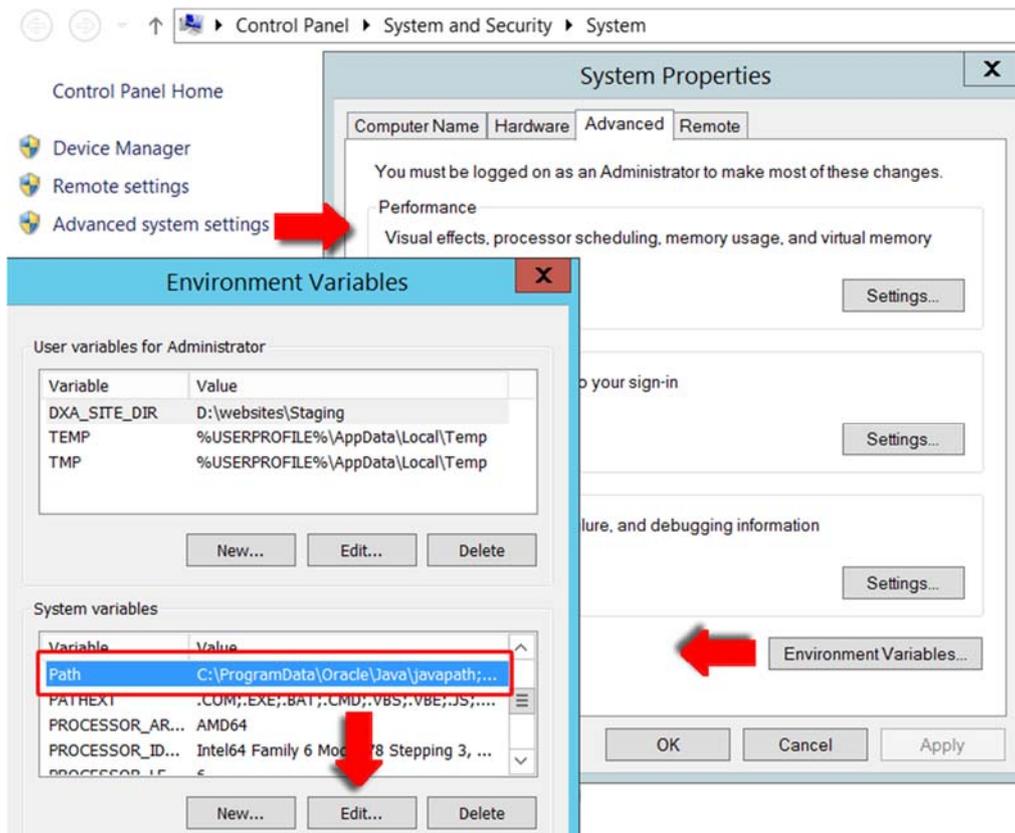
Redis Download: <https://github.com/ServiceStack/redis-windows/blob/master/downloads/redis64-3.0.501.zip>

Redis Desktop Manager Download: <https://redisdesktop.com/>

1. Download Redis using the link above.
2. Install the database by extracting the zip and placing the files in a directory of your choice. In this guide the location is **d:\redis**.



3. Open Control Panel -> System -> Advanced system settings -> Environment Variable.
 - a. Scroll to the System variable 'Path', select it and click on 'Edit'.



4. At the end of the Path variable enter the root directory of your Redis installation i.e.

Variable name:

Variable value:

OK Cancel

- a. Click 'OK'. You can now access the Redis Commands from any location within the Command Line Interface.
5. Open the Command Line Interface and enter the following commands:
 - a. **redis-server** (to start the Redis server)
 - b. **redis-cli** (to open the Redis Command Line Interface)
 - c. **ping** (to check the status of the Redis server) – it should return **PONG**

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>redis-server
[8628] 03 Feb 15:24:38.188 # Warning: no config file specified, using the default
t config. In order to specify a config file use redis-server /path/to/redis.conf

[8628] 03 Feb 15:24:38.195 # Creating Server TCP listening socket *:6379: bind:
No such file or directory

C:\Users\Administrator>redis-cli
127.0.0.1:6379> ping
PONG
127.0.0.1:6379>
```

6. We can now configure the DXA Application to use **Redis** for client-side caching.
7. From your DXA Web Application, open the **Web.config**.

```
<sdl.web.delivery>
<キャッシング defaultHandler="regularCache">
  <handlers>
    <add name="noCache" type="NullCacheHandler" />
    <add name="regularCache" type="DefaultMemCacheHandler">
      <!-- Use a low expiration time on a Staging site to prevent caching issues with XPM Session Preview. The set
      <policy absoluteExpiration="5" />
    </add>
    <add name="longLivedCache" type="DefaultMemCacheHandler">
      <policy absoluteExpiration="30" />
    </add>
    <!-- Set hashkey="false" if you want to see the full key instead of a hash code in Redis -->
    <add name="regularDistributedCache" type="RedisCacheHandler" instanceName="DXA-regular" hashkey="true">
      <endpoint host="localhost" port="6379" />
      <policy absoluteExpiration="5" />
    </add>
    <add name="longLivedDistributedCache" type="RedisCacheHandler" instanceName="DXA-longLived" hashkey="true">
      <endpoint host="localhost" port="6379" />
      <policy absoluteExpiration="30" />
    </add>
  </handlers>
  <regions>
    <!-- DXA Framework uses below cache regions. The cache mappings are just examples; adjust to your needs. -->
    <!-- DD4T objects: -->
    <add name="Page" cacheName="regularDistributedCache" />
    <add name="ComponentPresentation" cacheName="regularDistributedCache" />
    <!-- DXA objects: -->
    <add name="PageModel" cacheName="regularDistributedCache" />
    <add name="IncludePageModel" cacheName="longLivedDistributedCache" />
    <add name="EntityModel" cacheName="regularDistributedCache" />
    <add name="Navigation_Static" cacheName="regularDistributedCache" />
    <add name="Navigation_Dynamic" cacheName="regularDistributedCache" />
    <add name="NavTaxonomy" cacheName="longLivedDistributedCache" />
    <add name="BinaryPublishDate" cacheName="regularDistributedCache" />
  </regions>
</キャッシング>
</sdl.web.delivery>
```

8. Navigate to **<sdl.web.delivery>** node. You will notice that the Redis caching is already setup to use.
9. In the **<regions>** node update the caching options to use Redis as required (or as shown).
10. Save and close the **Web.config**.
11. Recycle or restart your Web Application in IIS.
12. Open your Home Page of your site (to trigger Page caching) and the items will be added to the Redis database as defined above.

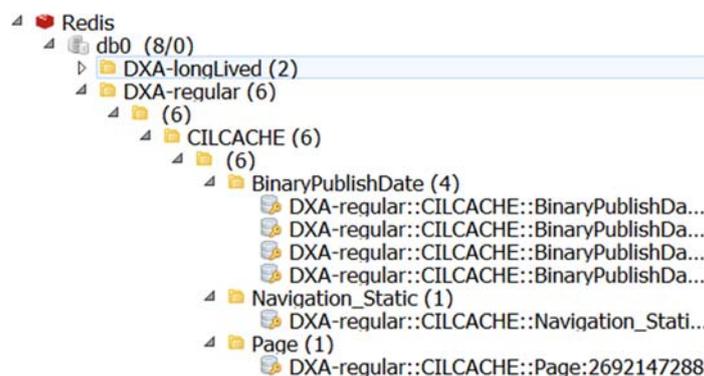
Testing methods

Note: You may want to raise the expiry time of the cached items to run this exercise:

```
<add name="regularDistributedCache" type="RedisCacheHandler" instanceName="DXA-regular" hashkey="true">
  <endpoint host="localhost" port="6379" />
  <policy absoluteExpiration="120" />
</add>
<add name="longLivedDistributedCache" type="RedisCacheHandler" instanceName="DXA-longLived" hashkey="true">
  <endpoint host="localhost" port="6379" />
  <policy absoluteExpiration="120" />
</add>
```

Set in your site's web.config. The above example sets cache expiration to 120 seconds. This will afford you time to see the cache in the following steps. In reality, you will need to gauge your respective cache expirations so they meet your optimization requirements within the site.

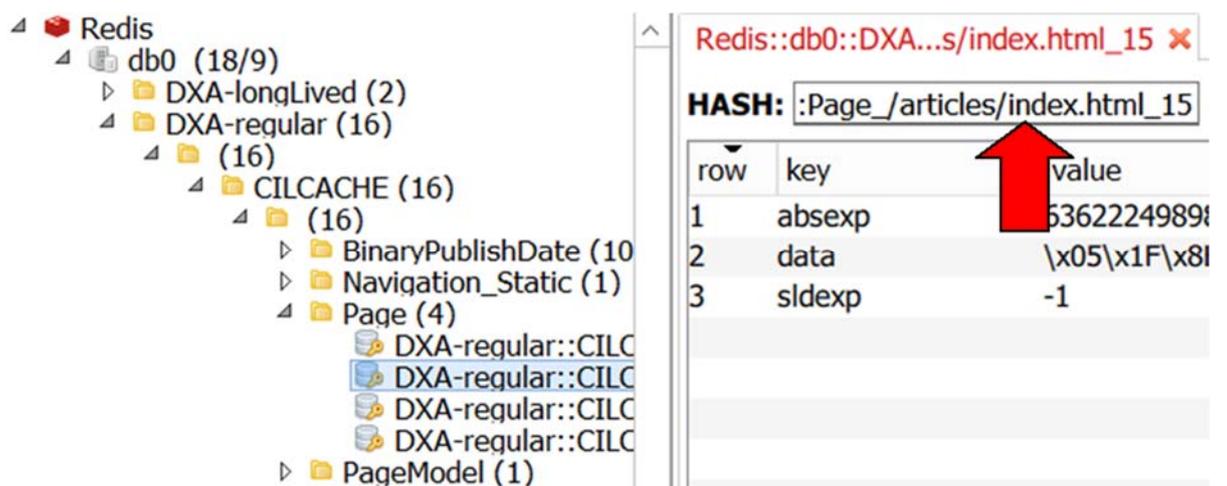
1. Install Redis Desktop Manager using the link at the beginning of this guide.
2. Refresh your web page to reload the cache in Redis.
3. Open Redis Manager and you will see your cached items as shown.



- When you check the items in the Redis database, by default they are not too descriptive as the key is hashed.
- Open the site's Web.config and update the distributed caching entry to set the hashkey value to false, as shown:

```
<add name="regularDistributedCache" type="RedisCacheHandler" instanceName="DXA-regular" hashkey="false">
  <endpoint host="localhost" port="6379" />
  <policy absoluteExpiration="120" />
</add>
<add name="longLivedDistributedCache" type="RedisCacheHandler" instanceName="DXA-longLived" hashkey="false">
  <endpoint host="localhost" port="6379" />
  <policy absoluteExpiration="120" />
</add>
```

- Refresh your site's web page and then reload the database in Redis.



The screenshot shows the Redis Explorer interface. On the left, a tree view displays the database structure: Redis > db0 (18/9) > DXA-longLived (2) > DXA-regular (16) > (16) > CILCACHE (16) > (16) > BinaryPublishDate (10) > Navigation_Static (1) > Page (4) > DXA-regular::CILC > DXA-regular::CILC > DXA-regular::CILC > DXA-regular::CILC > PageModel (1). On the right, a detailed view of a key is shown. The title bar reads 'Redis::db0::DXA...s/index.html_15'. Below it, the 'HASH:' field contains ':Page_/articles/index.html_15'. A table below shows the hash structure:

row	key	value
1	absexp	5362224989t
2	data	\x05\x1F\x8l
3	sldexp	-1

In the Hash value you can now see that this cached item is the Page **index.html**.

About SDL



SDL (LSE: SDL) is the global innovator in language translation technology, services and content management. For more than 20 years, SDL has transformed business results by enabling nuanced digital experiences with customers across the globe so they can create personalized connections anywhere and on any device. Are you in the know? Find out more at [SDL.com](https://www.sdl.com).

Copyright © 2017 SDL plc. All Rights Reserved. All company product or service names referenced herein are properties of their respective owners.