



Chao Beck

Integrating WebSphere Commerce with IBM WebSphere DataPower XC10

Like IBM® WebSphere® eXtreme scale, WebSphere DataPower® XC10 Version 2 provides business and technical benefits to new and existing WebSphere Commerce deployments.

This IBM Redpapers™ publication provides an overview of these benefits and explains how to configure WebSphere Commerce to use IBM WebSphere DataPower XC10. It also covers the technical aspects of installation, configuration, and monitoring.

The paper is designed for current and potential users of WebSphere Commerce who want to add the DataPower XC10 appliance to their configuration.

This paper includes the following sections:

- ▶ “Why use DataPower XC10 with WebSphere Commerce” on page 2
- ▶ “Installing WebSphere eXtreme Scale client over WebSphere Commerce” on page 4
- ▶ “Configuring dynamic caching” on page 6
- ▶ “Testing WebSphere Commerce caching” on page 27
- ▶ “Benefits of integrating WebSphere Commerce with WebSphere DataPower XC10” on page 36
- ▶ “Summary” on page 38

Why use DataPower XC10 with WebSphere Commerce

WebSphere Commerce is provided as a Java EE application running on the powerful and robust WebSphere Application Server platform. As such, Commerce already fully utilizes the WebSphere Application Server scalability and performance features such as clustering and dynamic caching. The dynamic cache service is a feature of WebSphere Application Server that provides an in-memory cache to improve performance by eliminating the need to regenerate dynamic web content that has not recently changed. Content is stored in the cache and shared until it is invalidated.

The IBM WebSphere DataPower XC10 Appliance provides a solution to the challenges of data scalability. The WebSphere DataPower XC10 Appliance caches data from the data servers for high-speed access by applications. The cache acts as a cushion for the data servers by storing frequently used data, thus reducing the number of required read and write accesses to the data servers. This caching tier is situated between the application server tier (which hosts the business applications) and the data tier. The cache is easily expandable and can store and manage large amounts of data securely with integrity and consistent performance.

WebSphere Commerce can take advantage of WebSphere DataPower XC10 caching (similar to eXtreme Scale) to provide performance and scale improvements by eliminating duplicate dynamic cache entries and the frequent invalidation processing necessary to keep the cache synchronized during high-stress situations. The combination of WebSphere Commerce and WebSphere DataPower XC10 can potentially reduce response time and facilitate faster start time for new Commerce Servers added to a cluster.

As such, you can use the IBM WebSphere DataPower XC10 appliance to store data from your WebSphere Commerce Server dynamic cache instead of caching the data in local memory. DataPower XC10 provides a plug-in for WebSphere Application Server to support DynaCache API requests to the appliance. By setting up this capability, you can enable applications that are written with the Dynamic Cache API or applications using container-level caching, such as servlets, to take advantage of the features and performance capabilities of the appliance. IBM provides the necessary client code to allow applications to access dynamic cache data on the DataPower XC10 appliance.

The DataPower XC10 client adds additional dynamic cache configuration options to the WebSphere administrative console. Using the DataPower XC10 dynamic cache requires a few simple configuration changes; your deployed code does not need to change.

Although both DataPower XC10 and eXtreme scale both provide a beneficial caching mechanism for WebSphere commerce, DataPower XC10 provides easier configuration steps and has a larger capacity. Additionally, with eXtreme scale you are limited by the size of the heap that you can set for the eXtreme scale container Java virtual machines (JVMs), unlike DataPower XC10.

WebSphere dynamic cache provider

Figure 1 shows the default WebSphere dynamic cache provider. The top row represents a WebSphere Commerce application server cluster consisting of WebSphere Commerce JVMs. Notice that each JVM contains a separate copy of cached data. When the in-memory cache in a JVM reaches its capacity, the dynamic cache provider can optionally use its disk offload feature to write cached entries to local disk storage, as illustrated by the bottom row.

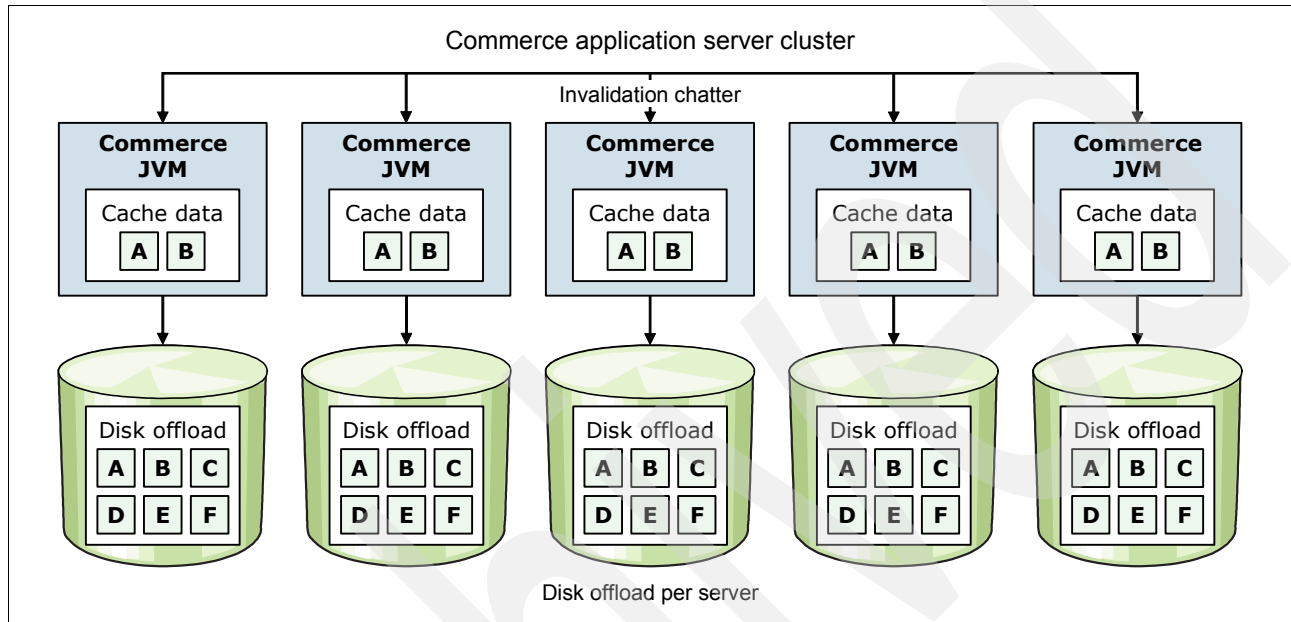


Figure 1 WebSphere dynamic cache provider overview

The default WebSphere dynamic cache provider uses the Data Replication Service (DRS) to replicate cache data and provide a distributed cache. Due to limitations of dynamic cache and DRS architecture, this distributed cache does not scale and is less flexible in terms of configuration which leads to excess invalidation and replication chatter.

DataPower XC10 uses the WebSphere eXtreme Scale dynamic cache provider. This provider uses a slightly different architecture. As shown in Figure 2, the top row represents a WebSphere Commerce application server cluster, but this time connected to a DataPower XC10 *collective* (consisting of one or more XC10 appliances). Note that all of the cached data is stored on the DataPower XC10 collective.

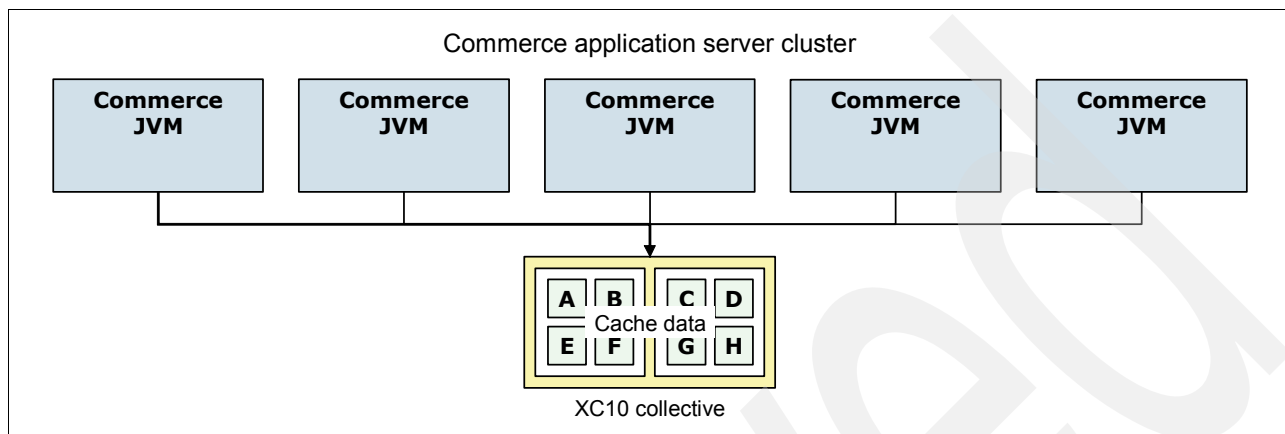


Figure 2 Commerce application server cluster with DataPower XC10

This setup allows for lower local memory requirements, enabling the WebSphere Commerce JVMs to run more efficiently. Because all data is cached in the collective, individual WebSphere Commerce JVMs are less likely to have “stale” data and do not have to call back-end systems that generate the data as often, reducing load. This setup also allows for high availability of cache data and improved performance.

Installing WebSphere eXtreme Scale client over WebSphere Commerce

Installing the WebSphere eXtreme Scale Client is required for your client applications to communicate with the IBM WebSphere DataPower XC10 appliance. Note that the IBM WebSphere DataPower XC10 uses the same client code as WebSphere eXtreme Scale V7.1. The client is available as a download from the IBM Support home page:

<http://www-947.ibm.com/support/entry/portal/Overview?brandind=Software~WebSphere>

. Note that IBM DataPower XC10 appliance firmware level V2.0 is the version that supports integration with WebSphere Commerce.

Installation options

There are two installation options for WebSphere eXtreme Scale: client and server.

- If you are installing a version of WebSphere eXtreme Scale earlier than version 7.1, you need to select the server option to install both client and server on the Commerce nodes. Although WebSphere Commerce is technically an eXtreme Scale client, WebSphere Commerce Servers require the server installation because it contains the dynamic cache provider files.
- With WebSphere eXtreme Scale version 7.1, you only need install the client on the Commerce nodes.

WebSphere eXtreme Scale V7.1 can be installed in either a new directory, for a stand-alone deployment, or it can be installed on top of a WebSphere Application Server cell for an integrated deployment. When using DataPower XC10, integrated deployment is recommended.

Requirements

The following assumptions pertain to the target environment for WebSphere eXtreme Scale installation:

- ▶ WebSphere Commerce is already installed on WebSphere Application Server Network Deployment v7.0.0.11 (or any supported release).
- ▶ A WebSphere Commerce application server or cluster already exists that can be configured for use with eXtreme Scale.

We must install WebSphere eXtreme Scale into the same Network Deployment cell as WebSphere Commerce. This is the only deployment available for DataPower XC10, but it introduces the operational consideration that the firmware level on the appliance should be at the same level or later than the eXtreme Scale client running on the WebSphere Application Server.

For more details of WebSphere eXtreme Scale client, refer to WebSphere DataPower XC10 Version 2 information center:

<http://publib.boulder.ibm.com/infocenter/wdpxc/v2r0/index.jsp>

Installation steps

The following high-level steps are required to prepare the Commerce environment:

1. Install WebSphere Application Server Network Deployment at the correct maintenance level to support the DataPower XC10 appliances you intend to use for your data caching environment.
2. Install WebSphere eXtreme Scale on top of the WebSphere Commerce and the WebSphere Application Server installations.
3. Augment all WebSphere Commerce nodes with eXtreme Scale using the profile management tool of WebSphere Application Server.

The following profiles will typically need to be augmented:

- The deployment manager profile. Augmenting this profile enables portions of the administrative console specific to the WebSphere eXtreme Scale.
- Custom profiles that contain the Commerce application servers.

Note: During the eXtreme Scale client installation, the installer will ask if you want to augment the nodes, which you can do during the install, or after the install if you also run the profile management tool. If you augment the nodes during the install, it is important to run the `./versionInfo.sh` command (`versionInfo.bat` for Windows) on each of the nodes to verify that augmentation was successful.

You are now ready to configure the WebSphere eXtreme Scale dynamic caching.

Configuring dynamic caching

After installing the client, the next step is configuring dynamic caching. There are several steps for performing this configuration, which we cover in both summarized and detailed forms.

Summary of configuration process

1. Configure a catalog service domain through the Commerce WebSphere Application Server administrative console.
2. Create dynamic cache data grids on the DataPower XC10 appliance and corresponding dynamic cache instances on WebSphere commerce. Cache instances can be servlet cache, object cache, or base cache, depending on what needs to be cached. After the cache instances are created, caching is enabled and topology settings created.
3. Configure the WebSphere eXtreme Scale dynamic cache provider on each managed server instance. You must enable replication settings and set certain custom properties.

After completing these steps, the WebSphere Commerce application is configured for caching. The *dynacache-ready* application can now be installed for caching. Dynacache-ready as applied to servlet caches means that the application includes XML configuration files specifying how it uses dynamic caching.

You can monitor the cache on the DataPower XC10 appliance console Monitoring menus.

Figure 3 shows end-to-end summary steps for configuring dynamic caching.

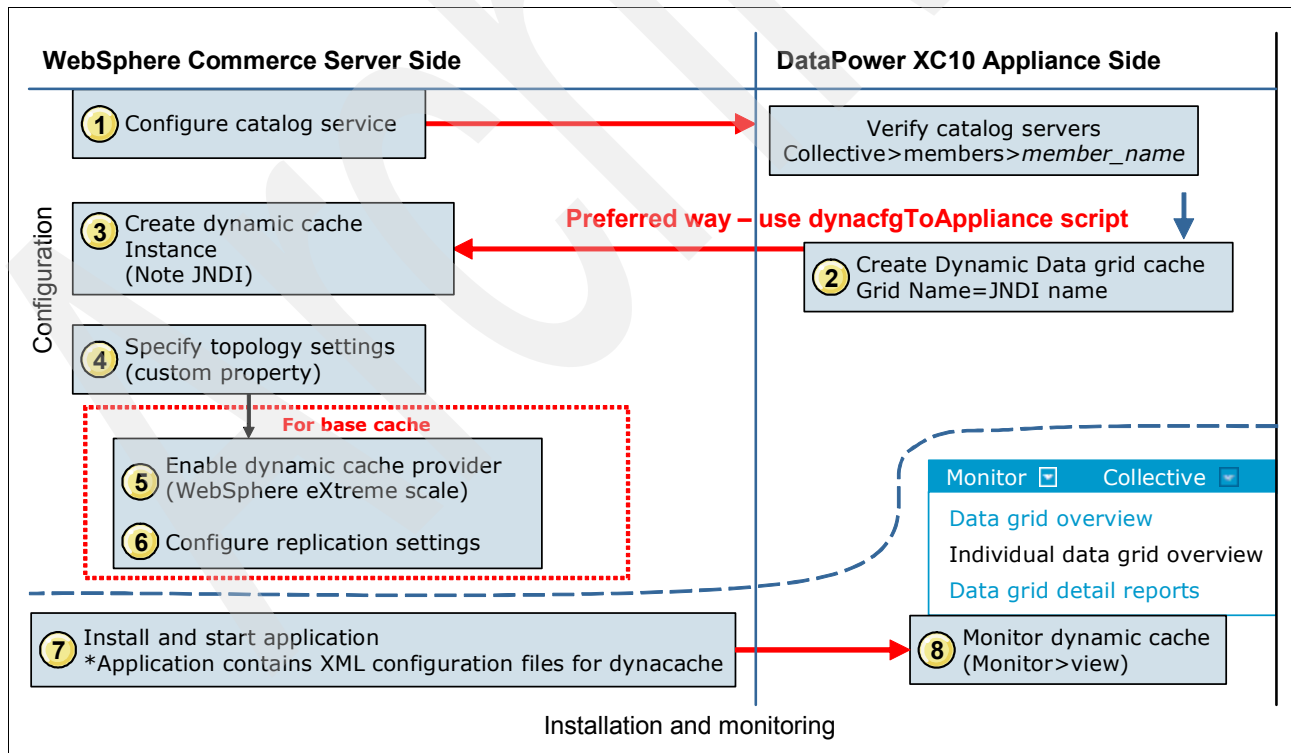


Figure 3 End-to-end scenario for configuring dynamic caching for WebSphere Commerce with DataPower XC10

Configuration steps

This section provides detailed information about configuring WebSphere Commerce and the DataPower XC10 appliance.

Step 1. Creating the catalog service domain

After the installation of the WebSphere eXtreme Scale client is complete, the first step in configuring WebSphere Commerce with the XC10 is to create the catalog server domain.

WebSphere has to know how to communicate with the catalog service that is running on the DataPower XC10 appliance. The appliance uses the catalog service to track how data is partitioned within the appliance and across a collective.

Configure the catalog service communication through the WebSphere Application Server administrative console by creating a catalog service domain, as follows:

1. In the WebSphere Application Server administration console (Figure 4), click **System administration** → **WebSphere eXtreme Scale**, and then click **Catalog service domains** (item 1).
2. On the catalog service domains panel (item 2) click **New**.

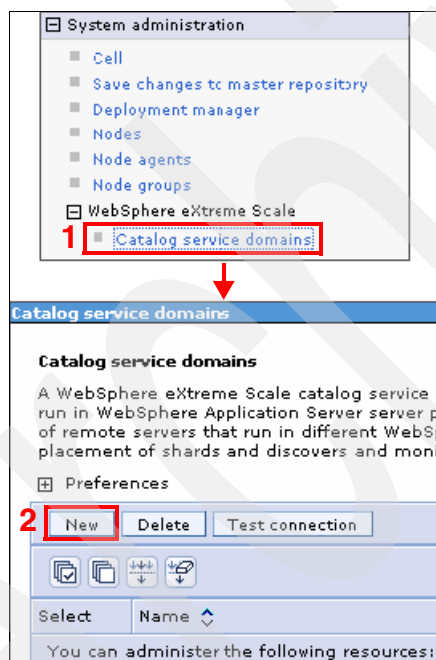


Figure 4 Selecting Catalog service domains menu

- On the **Catalog service domain** panel (item 3 in Figure 5), enter an administrative name for your catalog service domain. Note that by default the check box for **Enable this catalog service domain as the default unless another catalog service domain is explicitly specified** is checked. If there is another catalog service enabled as the default, uncheck this box.
- Add catalog server endpoints under **Catalog servers** (item 4). Click **Remote server** and type the host name or IP address of the DataPower XC10 appliance. The Listener Port is 2809.

Catalog service domains

[Catalog service domains](#) > New

A WebSphere eXtreme Scale catalog service domain is a highly available collection of catalog servers. This collection of catalog servers can run in WebSphere Application Server server processes within a single cell and core group. The catalog service domain can also define a group of remote servers that run in different WebSphere Application Server cells or as Java SE processes. The catalog service controls the placement of shards and discovers and monitors the health of the container servers in the data grid.

General Properties

* Name **3**

☒ Enable this catalog service domain as the default unless another catalog service domain is explicitly specified.

Catalog Servers

Select	Catalog Server Endpoint	Client Port	Listener Port
<input type="radio"/>	Existing application server <input type="text" value="WSBETA177Cell01\WSBETA177CellManager01\dmgr"/>	<input type="text"/>	<input type="text" value="2809"/>
<input checked="" type="radio"/> 4	Remote server <input type="text" value="xc29.rtp.raleigh.ibm.com"/>	<input type="text"/>	<input type="text"/>

The additional properties will not be available until the general properties for this item are applied or saved.

Additional Properties

☐ Custom

Figure 5 Creating catalog service domain

You can also define more than one catalog server endpoint on one catalog service domain as shown in Figure 6. Simply click **New** under the Catalog Servers panel and a new line will be added. Enter the new endpoint. Each endpoint corresponds to an appliance. There is no need to create several domains for several endpoints.

Catalog service domains

[Catalog service domains](#) > **WC_catalog service**

A WebSphere eXtreme Scale catalog service domain is a highly available collection of catalog servers. This collection of catalog servers can run in WebSphere Application Server server processes within a single cell and core group. The catalog service domain can also define a group of remote servers that run in different WebSphere Application Server cells or as Java SE processes. The catalog service controls the placement of shards and discovers and monitors the health of the container servers in the data grid.

General Properties

* Name

☒ Enable this catalog service domain as the default unless another catalog service domain is explicitly specified.

Catalog Servers

Select	Catalog Server Endpoint	Client Port	Listener Port	Status
<input type="checkbox"/>	xc29.rtp.raleigh.ibm.com		2809	
<input type="checkbox"/>	xc30.rtp.raleigh.ibm.com		2809	
<input type="checkbox"/>	xc9.rtp.raleigh.ibm.com		2809	

Several endpoints under one domain

Additional Properties

- ☐ [Client security properties](#)
- ☐ [Custom properties](#)

Figure 6 Catalog service domain endpoints

Note: If you are configuring a collective and there is more than one catalog server listed for the collective, you have to create a separate entry for each one by clicking **New** under the Catalog Servers panel. To view the catalog servers that are running in the collective, log into the DataPower XC10 appliance and click the menu **Collective** → **Members** → *member_name*. The Catalog servers field lists the catalog servers that are running in the collective (as shown in Figure 7).

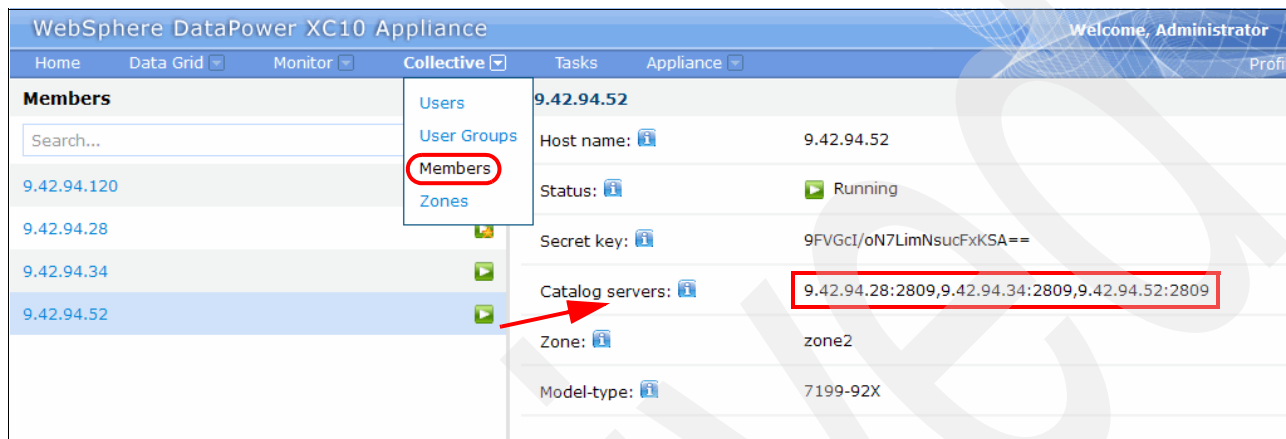


Figure 7 Sample catalog servers in a collective

5. Test the connection to the catalog service domain.

After the catalog service domain has been defined on WebSphere, it is important to test the connection before proceeding. To test the connection to your catalog service domain from the WebSphere Application Server administrative console, click **System administration** → **WebSphere eXtreme Scale** → **Catalog service domains**.

Select the catalog service domain that you want to test and click **Test connection** as shown in Figure 8. When you click this button, all of the defined catalog service domain end points are queried one by one. If any one endpoint is available, WebSphere returns a message that indicates that the connection to the catalog service domain was successful.

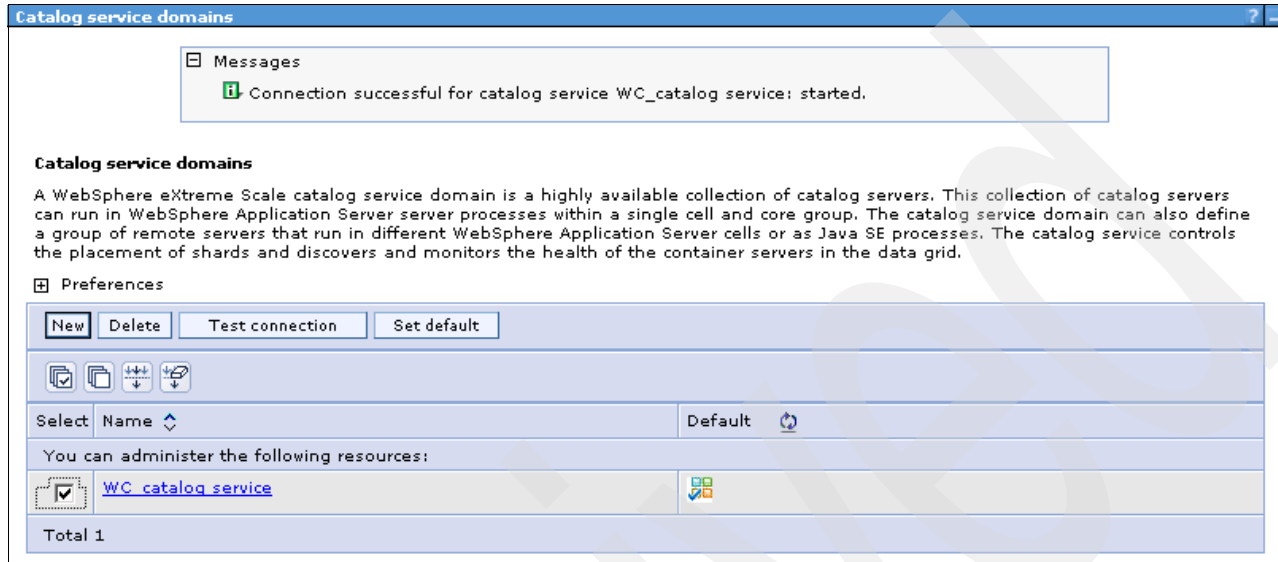


Figure 8 Testing connection to the catalog service domain

Step 2: Creating the dynamic cache data grid on the DataPower XC10 appliance

There are *two* ways to create dynamic cache data grids on the DataPower XC10 appliance: Run the **dynaCfgToAppliance** script or use the DataPower XC10 appliance console.

dynaCfgToAppliance script

One way to create dynamic cache data grid is by using the **dynaCfgToAppliance** script provided with the DataPower XC10 client or by using the DataPower XC10 appliance console, as follows:

```
dynaCfgToAppliance <IP_address> <cache_name> <appliance_admin> <admin_pw>
[<SOAP_port> <sas.client.props>]
```

This script is installed in the **bin** directory of the deployment manager when you install the client. This is the preferred way to create a cache data grid because it not only creates the grid on the appliance, but also performs additional WebSphere configuration.

Before you run this script, verify that both the appliance and the deployment manager are running.

The script requires several parameters:

- ▶ The IP address or host name of your DataPower XC10 appliance
- ▶ The name of the dynamic data cache you are creating,
- ▶ A user name and password. This user name and password pertain to a user defined on the appliance with “data cache creation” permission.

Optional parameters include the deployment manager SOAP port and a WebSphere Application Server security properties file.

You will use the name of the data grid that is created as the JNDI name when creating the dynamic cache instance from the WebSphere admin console.

DataPower XC10 appliance console

The other way to create dynamic cache data grid is to use the **DataPower XC10 appliance console** as shown in Figure 9. The numbers in the figure correspond to the following steps:

1. Select the data grid type you want to create from the Data Grid menu
2. Click the + sign to open a popup for creating a new grid
3. In the popup, enter a grid name
4. Click OK

Note that the grid name that you specify must match the JNDI name that you specified in the WebSphere Commerce Server configuration.

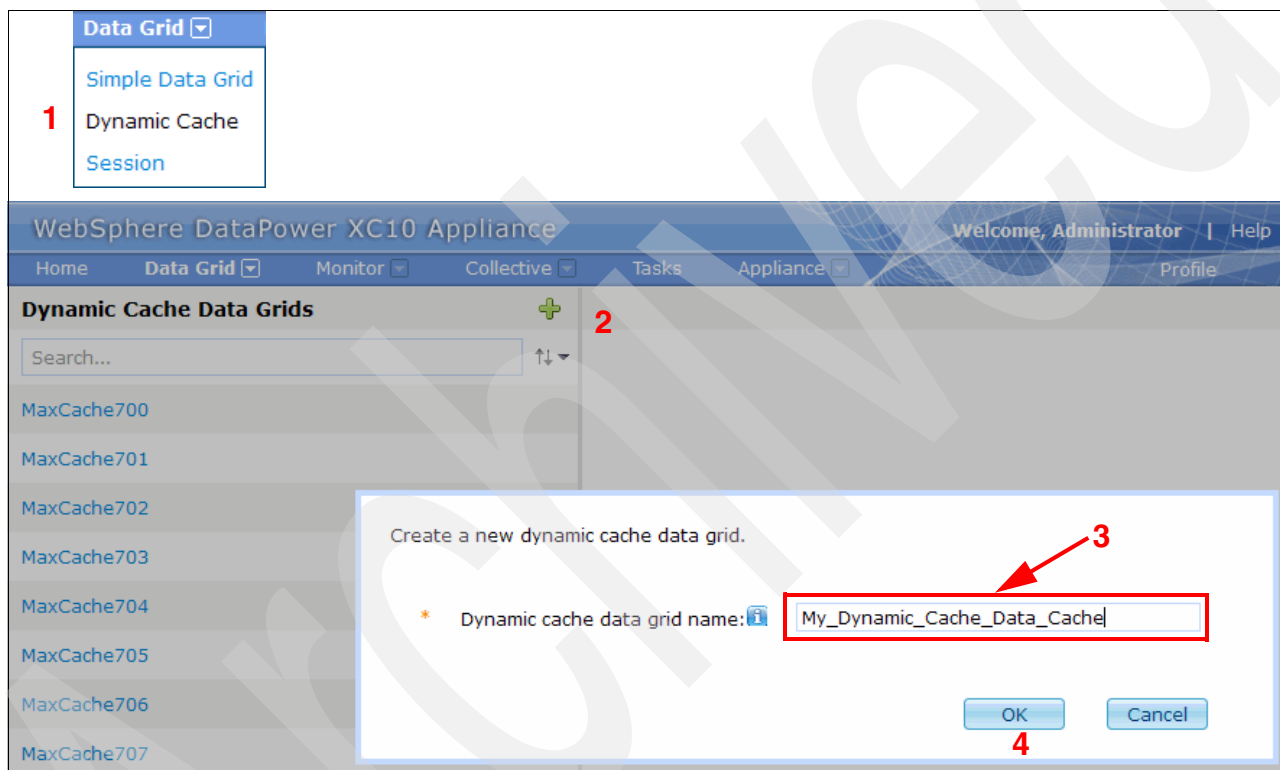


Figure 9 Creating data grid with DataPower XC10 appliance console

Figure 10 shows the details of the data grid.

This console page contains the following information and settings:

- Catalog servers (item 1): Displays the catalog server host address and port.
- Security settings (item 2): Facilitates authentication and authorization preferences.

If **Enable security** is selected, then you have to provide an authenticated user name and password to access the cache. If **Enable authorization** is checked, then the user name must also be granted specific permissions to the data cache. These permissions are read, write, create, and all, and they are checked each time you perform an action against the data cache.
- Access granted to (item 3): Allows you to select the user name that can access and modify this cache and the permissions of this user
- Show advanced attributes (item 4): If you expand this option you can see the replication settings. The settings can be from zero to five synchronous replicas, and from zero to five asynchronous replicas. These settings default to zero synchronous and one asynchronous replicas.
- “Eraser” icon at the top right (item 5): Clears grid.

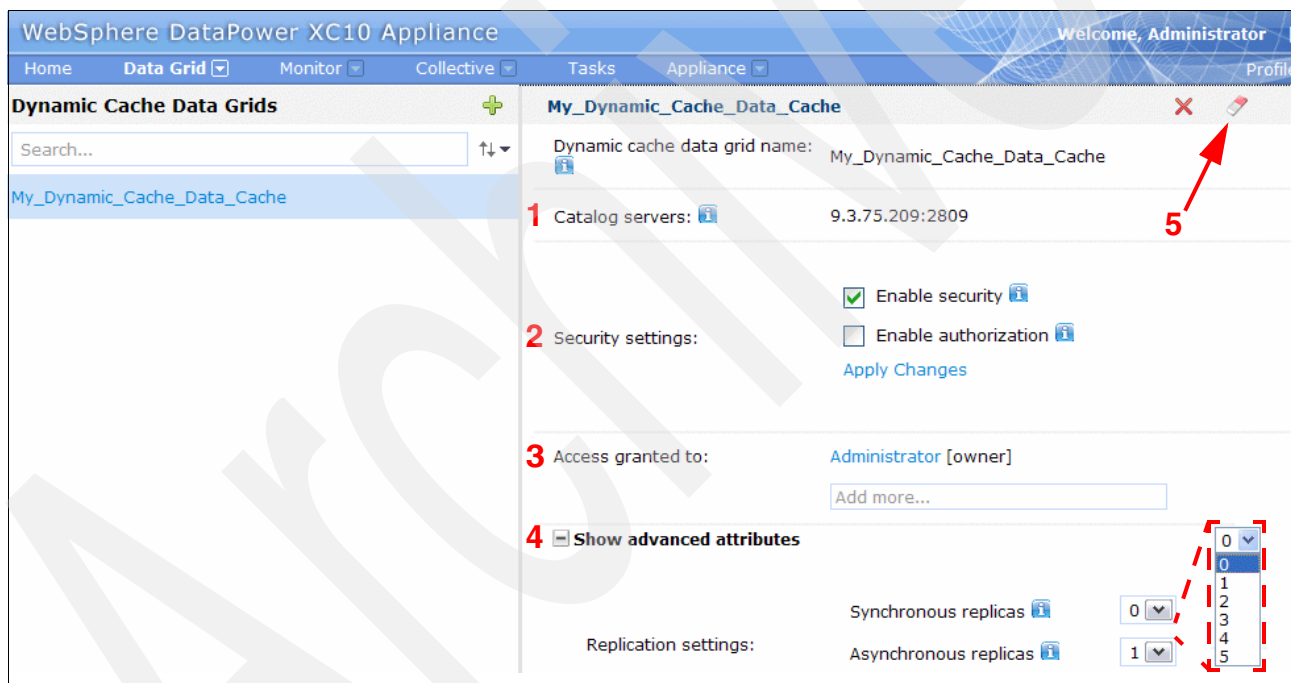


Figure 10 Data grid specifics on DataPower XC10 appliance console

Cache ID and password custom properties

If you enabled security on the dynamic cache data grid, you will need to add a cache user name and password to the WebSphere configuration as shown in Figure 11.

Note: The `dynaCfgToAppliance` script automatically creates these custom properties, so if you use this preferred method, you do not have to perform this additional step.

1. Create the user ID and password custom properties on the WebSphere Commerce Server. In the WebSphere Application Server administrative console, select **System administration** → **Cell**.
2. On the cell details panel, click **Custom Properties**.

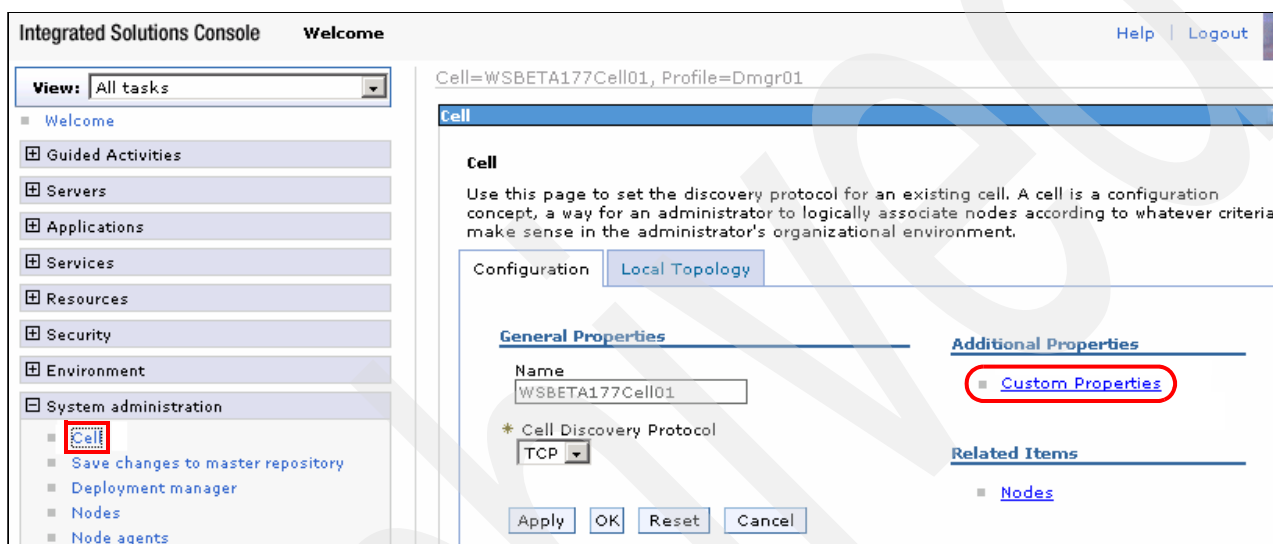


Figure 11 Custom properties

3. Click **New** to create a property for the user name.
4. Type the property name `xc10.data_grid_name.userid`
Here `data_grid_name` is the name of the cache you created on the appliance and `userid` is the user name created with admin permissions to access the cache on the XC10 console.
Once that user is created, when you create the custom properties on the WAS side, you only need to specify the cache `data_grid_name`, which will retrieve the information for the user associated with that cache.
5. Click **OK** to complete the creation.

6. Click **New** to create a property for the user's password, as shown in Figure 12.
7. Type the property name `xc10.data_grid_name.password`. Like with the userid, the password is automatically pulled in when `data_grid_name` is specified.

The password value should be encrypted, or encoded, by using the `encodePassword` script, located in the `bin` directory of the deployment manager. Encoding can be done prior to creating this custom property.

Cell > Custom properties

Use this page to specify an arbitrary name and value pair, internal system configuration properties.

Preferences

New Delete

Select Name

You can administer the following resources:

Cell > Custom properties > New

Use this page to specify an arbitrary name and value pair, internal system configuration properties.

Configuration

General Properties

Name: xc10.My_Dynamic_Cache_Data

Value: testUser1

Description: user ID

Apply OK Reset Cancel

Cell > Custom properties

Use this page to specify an arbitrary name and value pair. The value that is specified for the name and value pair is a string that can set internal system configuration properties.

Preferences

New Delete

Select Name Value Description

You can administer the following resources:

<input type="checkbox"/>	xc10.My_Dynamic_Cache_Data.Cache.password	test1337fghdhjss	password
<input type="checkbox"/>	xc10.My_Dynamic_Cache_Data.Cache.userid	testUser1	user ID

Figure 12 Cache ID/password custom properties

If you make any security setting changes, you can rerun the script without causing any harm to your configuration.

Step 3. Creating a dynamic cache instance

After you have configured the catalog service domain and created the data grid, your next step is to create a dynamic cache instance on WebSphere Commerce configuration.

As shown in Figure 13, there are *two* types of dynamic cache instances that can be created on WebSphere Commerce Server: servlet cache instances and object cache instances.

- ▶ An **object cache** instance is a location, in addition to the default shared dynamic cache, where any Java EE application can store, distribute, and share data. This gives applications greater flexibility and better tuning of the cache resources.
- ▶ **Servlet cache instances** are locations in addition to the default dynamic cache where the dynamic cache service can store, distribute, and share the output and the side effects of an invoked servlet. By configuring a servlet cache instance, your applications have greater flexibility and better tuning of cache resources. The Java Naming and Directory Interface (JNDI) name specified for the cache instance is mapped to the name attribute in the <cache-instance> tag in the cachespec.xml configuration file included with your application.

To create a cache instance from the WebSphere administrative console, expand **Resources**, click **Cache instances**, and then click the type of instance you want to create.



Figure 13 Dynamic cache instances

Object cache instance

Object cache instances can improve the performance of your application because you can programmatically store and share frequently used objects. By using object cache instances, you also have the necessary control over the dynamic cache when you are running multiple applications on the WebSphere Commerce Server.

Create and configure an object cache instance as follows. This process is illustrated in Figure 14.

1. On the WebSphere administrative console, expand **Resources** → **Cache instances** and click **Object cache instance** (item 1). Specify the scope for the cache instance:
 - **Cell** scope makes the cache instance available to all servers within the cell.
 - **Node** scope makes the cache instance available to all servers on the particular node.
 - **Cluster** scope makes the cache instance available to all members in a specified cluster.
 - **Server** scope makes the cache instance available to only the selected server.

You can create object cache instances within any of these scopes. If you have more than one object cache instances, you can create them at separate scopes.
2. Click **New** to create a new cache instance (item 2). Type the Name and the Java Naming and Directory Interface (JNDI) name for this cache instance (item 3). The JNDI name must match the name of the dynamic cache data grid you created on the appliance.
3. Select **WebSphere eXtreme Scale** as the cache provider to utilize DataPower XC10 caching from the drop down list.
4. Check **Enable cache replication** for cached data to be sent to the DataPower XC10 appliance. If you do not check this entry, the XC10 client will provide a local cache only.

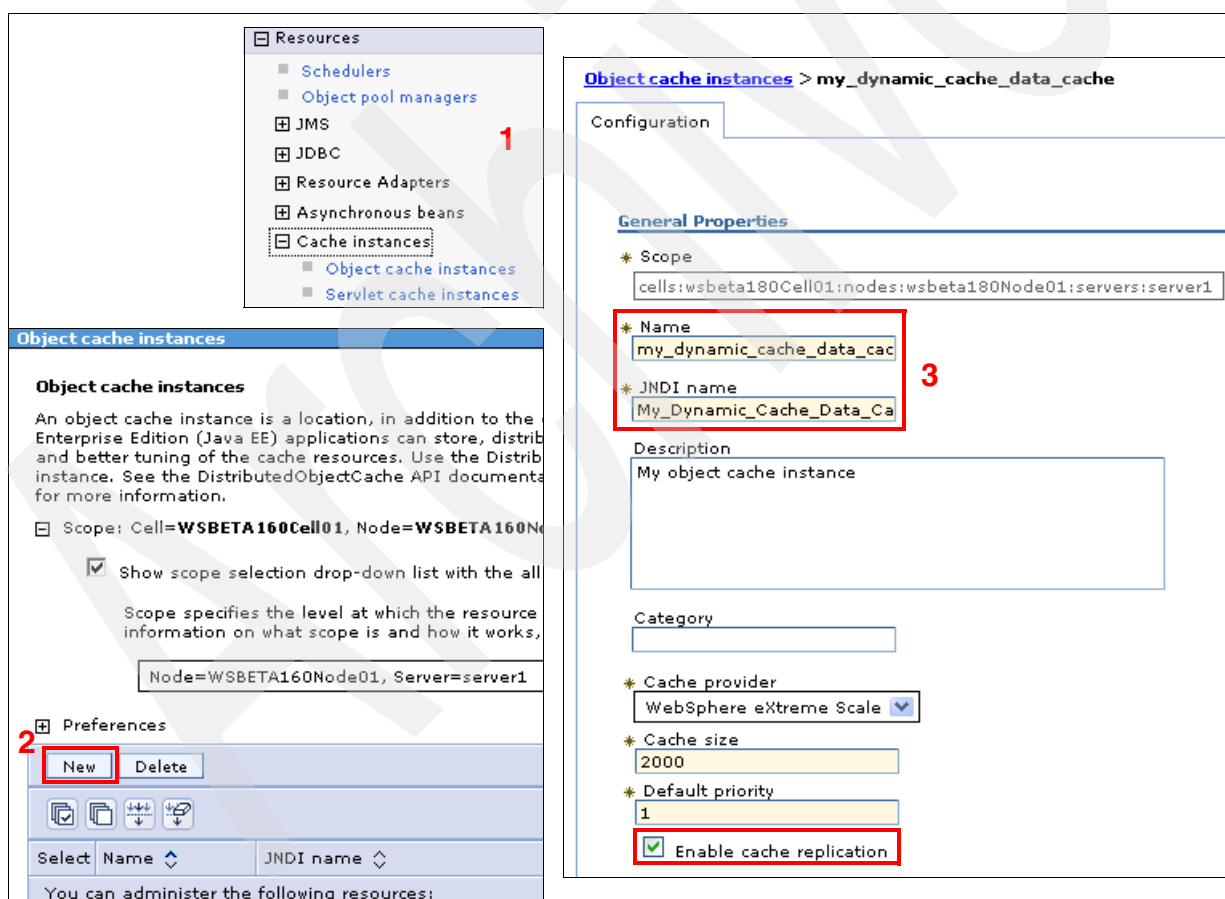


Figure 14 Creating an object cache instance

Note: The **Cache size** entry sets the maximum number of entries that the cache will hold in the appliance memory. The WebSphere default cache size is 2000. If you leave this size as the default, this translates to the following on DataPower XC10:

When the first request to place an entry on the grid comes through, the grid on DataPower XC10 is initialized. At this time, DataPower XC10 creates all of the partitions and sets the size of each partition to 2000. DataPower XC10 grids have 83 partitions, which means that the default cache size on DataPower XC10 is 166000 (2000*83).

Servlet cache instance

Using servlet cache instances can improve the performance of your application because you can store the output and the side effects of an invoked servlet. Servlet cache instances also give you the necessary control over the cache for multiple applications that are running in an application server.

1. On the WebSphere administrative console, click **Resources** → **Cache instances** → **Servlet cache instances**, as shown in Figure 15.
2. Specify the scope of the cache instance from the drop-down list.
 - **Cell** scope makes the cache instance available to all servers within the cell.
 - **Node** scope makes the cache instance available to all servers on the particular node.
 - **Cluster** scope makes the cache instance available to all members in a specified cluster.
 - **Server** scope makes the cache instance available to only the selected server.
3. Click **New** to create a new servlet cache instance.

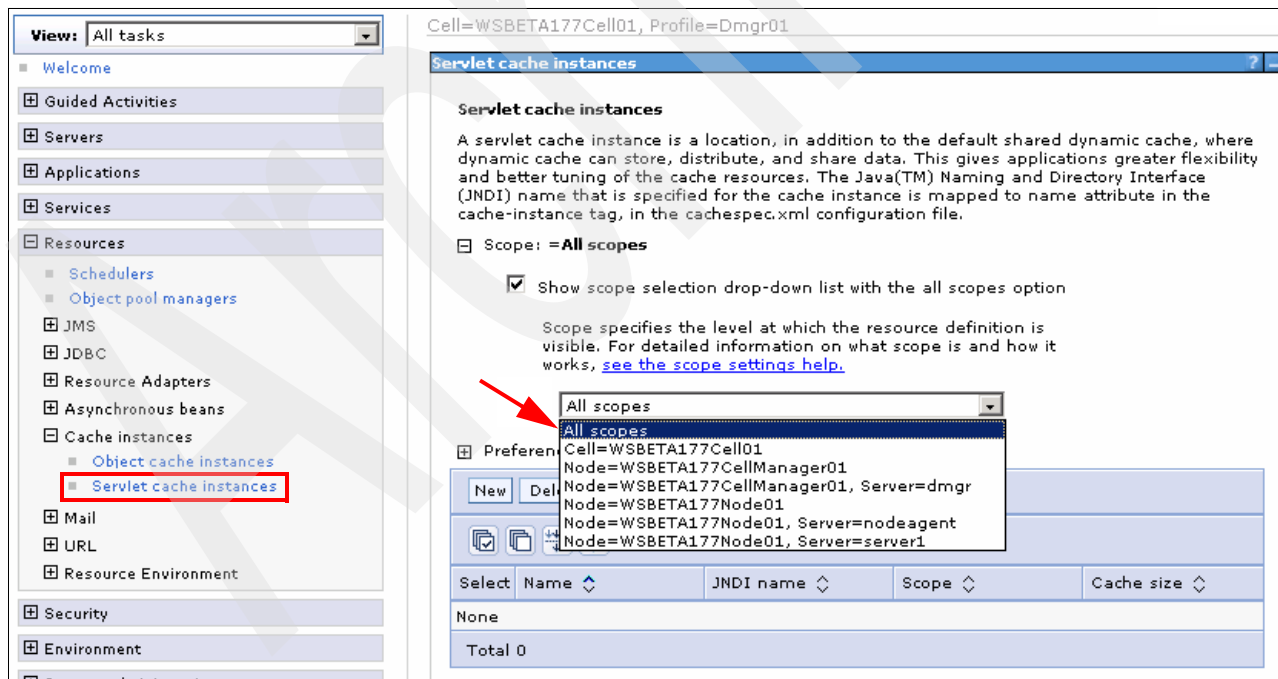


Figure 15 Servlet cache instance

4. As shown in Figure 16, configure the servlet cache instance on the WebSphere administrative console as follows:
 - a. Enter the name of the servlet cache instance.
 - b. Enter the Java Naming and Directory interface (JNDI) in the JNDI name fields. The JNDI name is the name attribute that is specified in the <cache-instance> element in the cachespec.xml file.

Servlet cache instances > New

A servlet cache instance is a location, in addition to the default shared dynamic cache, where dynamic cache can store, distribute, and share data. This gives applications greater flexibility and better tuning of the cache resources. The Java(TM) Naming and Directory Interface (JNDI) name that is specified for the cache instance is mapped to name attribute in the cache-instance tag, in the cachespec.xml configuration file.

Configuration

General Properties

- * Scope: cells:WSBETA177Cell01:nodes:WSBETA177CellManager01:servers:dmgr
- * Name: Commerce_servlet_cache_ins
- * JNDI name: services/cache/instance_one
- Description: [Text Area]
- Category: [Text Field]
- * Cache provider: WebSphere eXtreme Scale
- * Cache size: 2000
- ☒ Enable cache replication

Additional Properties

- Custom properties

Buttons: Apply, OK, Reset, Cancel

Figure 16 Creating a servlet cache instance

Typically, the cache parses the cachespec.xml file when the server starts and extracts a set of configuration parameters from each cache-entry element. Every time a new servlet initializes, the cache attempts to match each of the cache-entry elements to find the configuration information for that object.

Here is an example of a JNDI name that is specified in the cachespec.xml file:

```
<cache-instance name= "services/cache/instance_one">
```

In this example, specify **services/cache/instance_one** as the JNDI name.

To use a servlet cache instance, you must specify a <cache-instance> element whose name is the same as the JNDI Name for this cache instance in your application. If a name is not specified for <cache-instance>, baseCache will be used as a default (See "baseCache" on page 23). Also, the JNDI name must match the data grid name created in the DataPower XC10 appliance.

Note: Use of “/” in a JNDI name

In certain situations, it might be necessary to create a WebSphere cache instance with a JNDI name containing “/”, such as **services/cache/cache1**. If such a JNDI is created, a corresponding grid for this cache instance will be created by running the **dynaCfgToAppliance** script specifying **services/cache/cache1** as the grid.

However, if an attempt is made to create the grid manually on the DataPower XC10 Appliance console, the “/” will need to be replaced with “_”. Thus, in this example, the grid name **services_cache_cache1** would be specified on the appliance console.

- c. Select **WebSphere eXtreme Scale** from the drop-down list as the cache provider.
 - d. Check **Enable cache replication** for cached data to be sent to the XC10 appliance. If you do not check this entry, the XC10 client will provide a local cache only.
5. Click **OK** and save the changes.

Step 4. Configuring topology settings

When you use the DataPower XC10 Appliance to store cache data, you are using a *remote* topology. You therefore have to set a custom property on the dynamic cache instance in the WebSphere Application Server administrative console to notify the dynamic cache provider of this fact.

The following steps outline how to configure topology settings:

1. Select **Resources** → **Cache instances** and click **cache_instance_type**.
2. Click the cache instance name.
3. On the cache instance's General properties page, click **Custom properties**, and then click **New**.
4. Type the name of the custom property as **com.ibm.websphere.xs.dynacache.topology** and type the value of the custom property as **remote**.

Topology settings for servlet cache

Figure 17 shows the settings for the servlet cache instance.

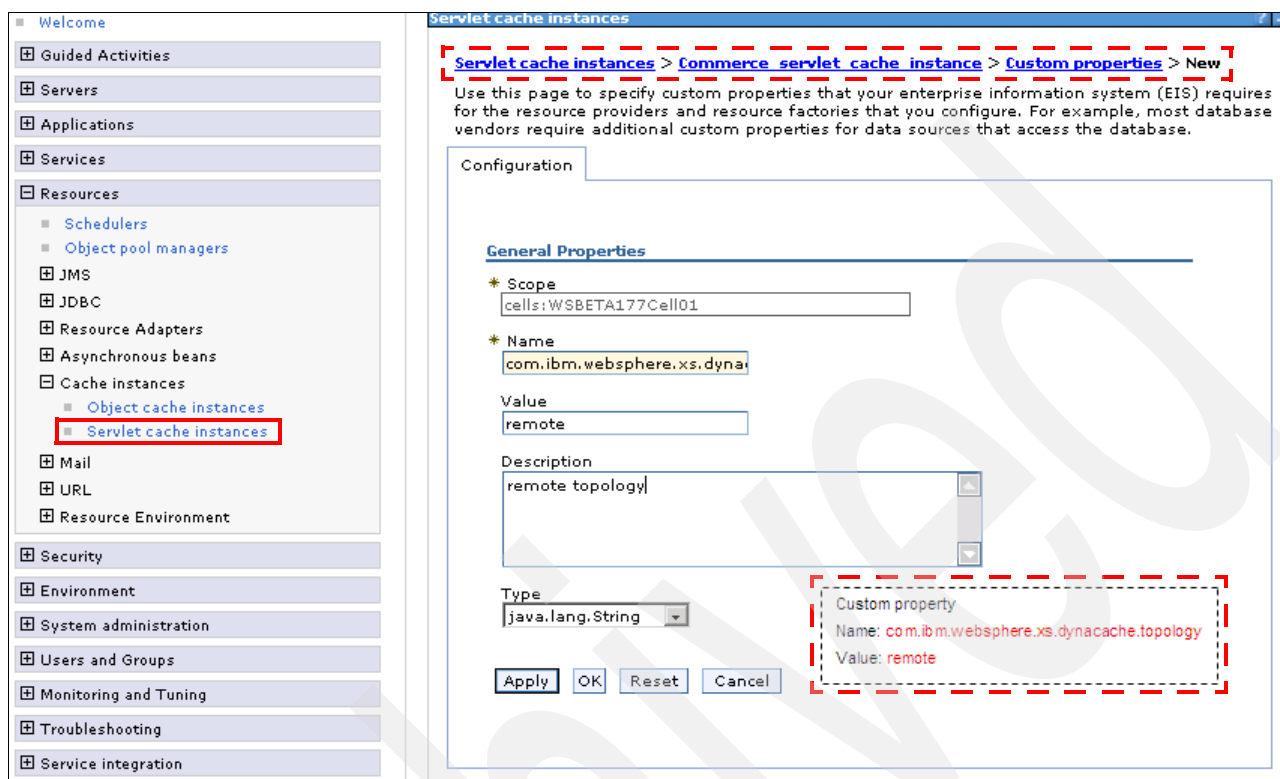


Figure 17 Topology settings for the servlet cache instance

Topology settings for object cache

Figure 18 shows the settings for the object cache instance.

The screenshot shows the 'Object cache instances' configuration page. On the left is a navigation tree with 'Object cache instances' highlighted. The main area is titled 'Object cache instances > Commerce object cache instance > Custom properties > New'. It contains a 'Configuration' section with 'General Properties' fields: Scope (cells:WSBETA177Cell01), Name (com.ibm.websphere.xs.dyna), Value (remote), Description (remote topology), and Type (java.lang.String). Buttons for Apply, OK, Reset, and Cancel are at the bottom.

Figure 18 Topology settings for the Object cache instance

Figure 19 shows the results of configuring topology for the object cache as shown in the custom properties list.

The screenshot shows the 'Object cache instances' configuration page with the 'Custom properties' tab selected. It displays a table of custom properties. The table has columns: Select, Name, Value, Description, and Required. One property is listed: 'com.ibm.websphere.xs.dynacache.topology' with value 'remote', description 'remote topology', and required 'false'. The table is titled 'You can administer the following resources:' and has a 'Total 1' summary at the bottom.

Select	Name	Value	Description	Required
<input type="checkbox"/>	com.ibm.websphere.xs.dynacache.topology	remote	remote topology	false

Total 1

Figure 19 Results of setting object cache topology

baseCache

If an explicit dynamic cache instance for an application is not set, the application will use the default baseCache (the default dynamic cache instance). baseCache is only a servlet cache instance.

The baseCache grid is not configured on the appliance by default. You can create the baseCache using the **dynaCfgToAppliance** script or from the DataPower XC10 appliance console. The cache name must be **baseCache**.

The custom property specifying the baseCache uses a remote topology which must be set on the Java virtual machine (JVM) of the WebSphere Commerce Server, as follows:

1. From the WebSphere Application Server administrative console, select **Servers** → **Server types** → **WebSphere Application Servers** → **WebSphere ommerce_Server_name** → **Java and process management** → **process definition** → **Java virtual machine**, and then click **Custom properties**.
2. Click **New**.
3. Specify the name of the custom property as `com.ibm.websphere.xs.dynacache.topology` and the value of the custom property as `remote` (see Figure 20).

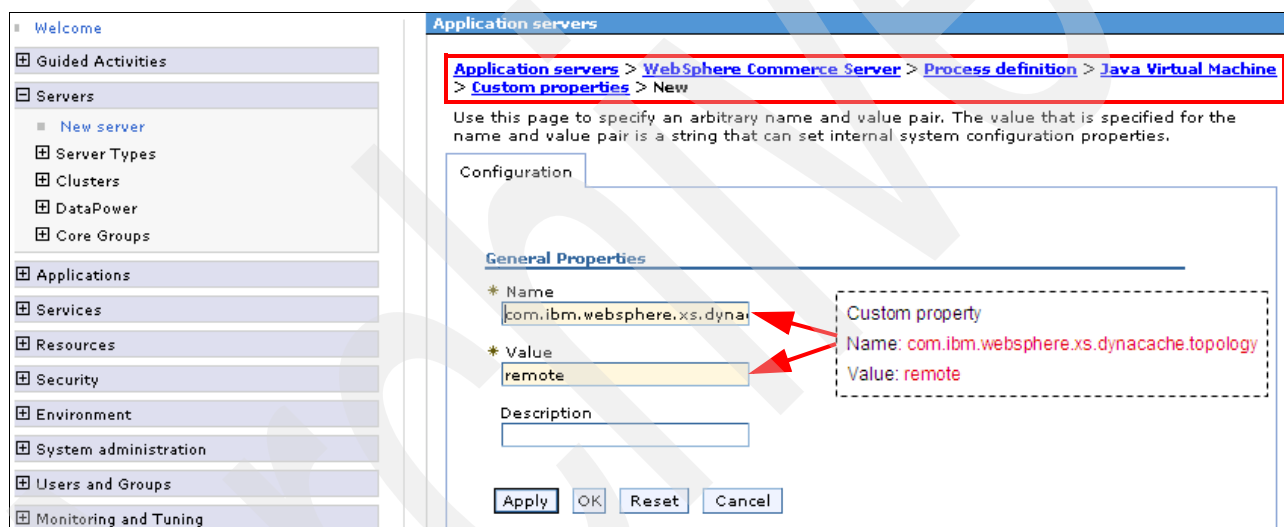


Figure 20 Topology settings for a baseCache instance

Configuring the baseCache instance sets all properties of the cache instances in the JVM. You can override the values of the baseCache custom properties by creating custom properties on the individual cache instances.

Authentication credentials for the baseCache are configured exactly like they are for any other cache instance: the user ID and password properties are set at the WebSphere cell level.

Step 5. Enabling the dynamic cache service and servlet caching

The last step in configuring dynamic caching involves configuring the baseCache and any custom settings.

Configuring the baseCache

Configure the baseCache to use the WebSphere eXtreme Scale service provider as follows:

1. Log into WebSphere Application Server administrative console.
2. Select **Servers** → **Server Type** → **WebSphere application servers** → **your_Commerce_server** → **Container services** → **Dynamic cache service**, as shown in Figure 21.

The screenshot displays the 'Application servers' configuration page for 'server1'. The 'Configuration' tab is active. The 'General Properties' section on the left includes fields for 'Name' (server1) and 'Node name' (WSBETA160Node01), and checkboxes for 'Run in development mode', 'Parallel start' (checked), and 'Start components as needed'. Below these are 'Access to internal server classes' (set to 'Allow') and 'Server-specific Application Settings' (Classloader policy: Multiple, Class loading mode: Classes loaded with parent class loader first). The 'Container Settings' section on the right lists various services, with 'Dynamic cache service' highlighted by a red rectangle. Other services include Session management, SIP Container Settings, Web Container Settings, Portlet Container Settings, EJB Container Settings, Container Services (expanded), Application profiling service, Transaction Service, Compensation service, Internationalization Service, Default Java Persistence API settings, Object pool service, ORB service, Startup beans service, and Business Process Services.

Figure 21 Dynamic cache service configuration

3. From the Dynamic cache service page (as shown in Figure 22), configure cache provider settings.
 - a. Select **WebSphere eXtreme Scale** from the cache provider list.
 - b. Specify a cache size in the Cache size field. Start with the default setting of 2000.
 - c. Ensure **Enable cache replication** is checked. If this setting is not checked and only WebSphere eXtreme scale client is installed, initialization of the client might fail and the cache instance will default to using the default cache implementation.
 - d. Click **OK** to save the changes.
4. Enable servlet caching on the Commerce Server.

From the Dynamic cache service page, you can click **Web container** under General properties to see container settings.

Application servers > **server1** > **Dynamic cache service**

The dynamic cache service consolidates caching activities to improve application performance. By caching the response from servlets, Web services, Java(TM) Server Pages (JSP) files, and WebSphere(R) Application Server commands, the application server does not have to perform the same computations and back-end queries multiple times.

Configuration

General Properties

- To enable servlet caching, go to [Web container](#) settings.
- To enable portlet caching, go to [Portlet container](#) settings.
- * Cache provider: WebSphere eXtreme Scale
- * Default dynamic cache: 2000
- ☒ Enable cache replication

Additional Properties

- External cache groups
- Custom properties

Apply OK Cancel

Figure 22 Cache provider settings

Alternately, you can select **Servers** → **Server Type** → **WebSphere application servers** → **your_commerce_server** → **Web Container settings** → **Web container** as shown in Figure 23.

- Check the box **Enable servlet caching**.
- Click **OK** to save the changes.

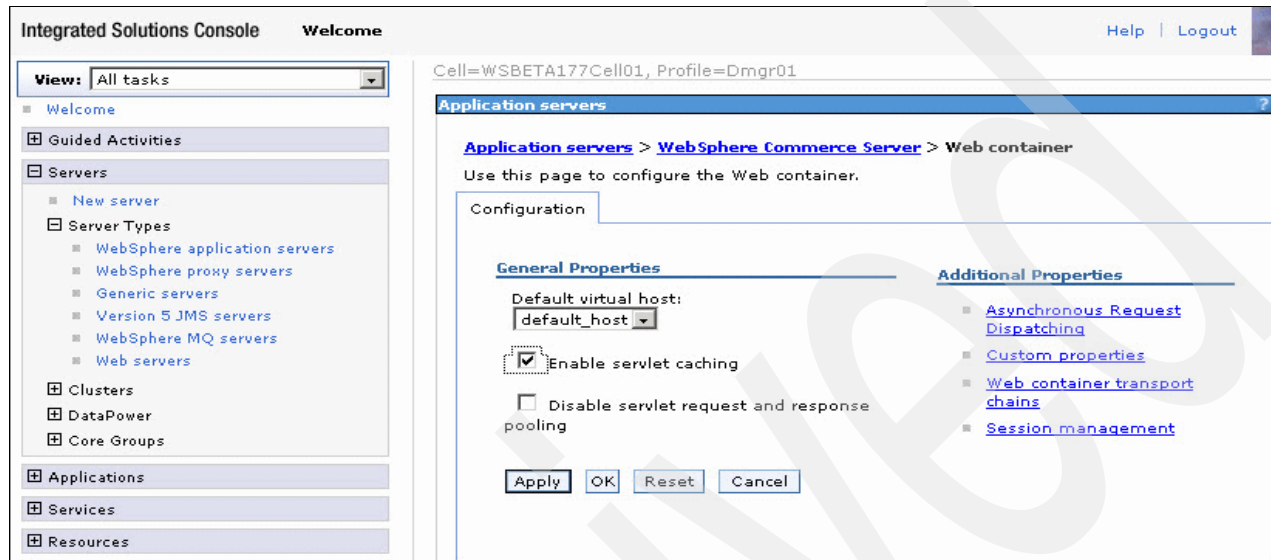


Figure 23 Web container settings

Additional Custom properties for baseCache

The additional custom properties shown in Table 1 must be defined for the baseCache to use the WebSphere eXtreme Scale service provider:

Table 1 JVM custom properties

Custom Property Name	Value
com.ibm.ws.cache.CacheConfig.filterLRUInvalidation	true
com.ibm.ws.cache.CacheConfig.filterTimeOutInvalidation	true
com.ibm.ws.cache.Cache.Config.filterInactivityInvalidation	true
com.ibm.ws.cache.CacheConfig.disableTemplateInvalidation	true
com.ibm.ws.cache.CacheConfig.ignoreValueInInvalidationEvent	true
com.ibm.ws.cache.CacheConfig.useServerClassLoader	true
com.ibm.websphere.xs.dynacache.disable_recursive_invalidate	true
com.ibm.websphere.xs.dynacache.ignore_value_in_change_event	true
com.ibm.websphere.xs.dynacache.enable_compression	true
com.ibm.websphere.xs.dynacache.topology	remote

More detailed information about configuring baseCache to use WebSphere eXtreme Scale service provider can be found in the IBM developerWorks® article “Configure WebSphere commerce to use WebSphere eXtreme Scale for dynamic cache to improve performance and scale.”

Where to place caches

Table 2 shows suggested environments for cache types used by WebSphere Commerce. The `baseCache`, which is the default servlet cache, should be configured on the WebSphere DataPower XC10. Similarly, servlet caching is also available on WebSphere DataPower XC10. WebSphere Commerce assumes all data is local, and therefore makes frequent access to certain types of data, such as user data. Caching these small but frequently accessed pieces of data on the DataPower XC10 can impact performance, therefore object caches for distributed map data should be configured to use WebSphere's default dynamic cache provider.

Table 2 Data caches and their locations

Data cache type	Best cache environment
baseCache Default servlet cache. Assume all web content fits into this category	WebSphere DataPower XC10
Servlet caching Used as specific cache name is used for web content	WebSphere DataPower XC10
Object caches (Used for DistributedMap data)	Default dynamic cache provider

Testing WebSphere Commerce caching

After your WebSphere Commerce system has been configured to use the DataPower XC10 for dynamic caching, you can ensure the setup is correct by looking at your `SystemOut.log`. After your server has completed startup, look for entries in the log that indicate the WebSphere eXtreme Scale Dynamic provider created the `baseCache` with remote topology as shown in Example 1.

Example 1 Caching output from using WebSphere eXtreme Scale Dynamic provider

`SystemOut.log` - using WebSphere eXtreme Scale Dynamic provider

[10/25/10 8:46:08:538 CDT] 00000000 CacheProvider 1 CWOBJ45001: WebSphere eXtreme Scale Dynamic Cache provider is successfully initialized.

[10/25/10 8:46:09:913 CDT] 00000000 Object GridImp 1 CWOBJ47001: The map name `IBM_DC_PARTITIONED_baseCache` matched the regular expression of template map `IBM_DC_PARTITIONED_.*`. **The `IBM_DC_PARTITIONED_baseCache` map has been created for ObjectGrid DYNACACHE_REMOTE.**

[10/25/10 8:46:09:991 CDT] 00000000 CacheProvider 1 CWOBJ45081: **The WebSphere eXtreme Scale provider has created a Dynamic Cache instance with name `baseCache` using topology `remote`.**

Example 2 shows SystemOut.log messages where something went wrong with the configuration and instead of WebSphere eXtreme Scale dynamic Cache provider being used, the WebSphere default dynamic cache provider was used instead. These messages might indicate your DataPower XC10 or your WebSphere Commerce cell is not configured correctly.

Example 2 Caching output from possible failure in configuration resulting in use of default provider

Error - default provider used

[11/26/10 14:32:37:324 CST] 00000000 CacheProvider E CWOBJ4501E: **The WebSphere eXtreme Scale Dynamic Cache provider encountered an error while creating the cache instance: baseCache.**

[11/26/10 14:32:37:449 CST] 00000000 CacheProvider W CWOBJ0006W: An exception occurred: com.ibm.websphere.objectgrid.ObjectGridRuntimeException: The catalog server did not contain an ObjectGridDeployment matching the name DYNACACHE_REMOTE At com.ibm.ws.objectgrid.ObjectGridManagerImpl.getRemoteObjectGridDeployment(ObjectGridManagerImpl.java:440)

[11/26/10 14:32:37:449 CST] 00000000 ServerCache E DYNA1066E: Unable to initialize the cache provider "com.ibm.ws.objectgrid.dynacache.CacheProviderImpl". The Dynamic cache is used to create the cache instance "baseCache" instead of the configured cache provider.

[11/26/10 14:32:37:449 CST] 00000000 ServerCache E ENGLISH ONLY MESSAGE: coreCache == null||featureSupport == null. ...Check FFDC logs for exceptions.

[11/26/10 14:32:37:480 CST] 00000000 ServerCache I DYNA1001I: WebSphere Dynamic Cache instance named baseCache initialized successfully.

[11/26/10 14:32:37:480 CST] 00000000 ServerCache I DYNA1071I: **The cache provider "default" is being used.**

Testing involves monitoring the cached data in DataPower XC10 appliance. You can use the charting capabilities of WebSphere DataPower XC10 Appliance to view the overall performance of the data grid.

To test the WebSphere Commerce application, invoke the URL you want to use for testing dynamic cache functionality. As an example, you can administer WebSphere Commerce sites using the administration console from an Internet Explorer browser:

http://<hostname>:port/adminconsole

Activity on the commerce page, such as that shown in Figure 24, generates web content that is saved in the dynamic cache. After several tests, login to the DataPower XC10 appliance and navigate to the monitoring panel for your dynamic data cache.

Allow some time to pass for the statistics to become available on the DataPower XC10 appliance. With a dynamic cache data grid, statistics are not available until a WebSphere Commerce Server that is running a dynamic cache connects to the dynamic cache data grid on the appliance. If you are using a collective, the collective initialization must be complete before statistics are available. In general, wait up to one minute after a major configuration change to see the changes in your statistics.

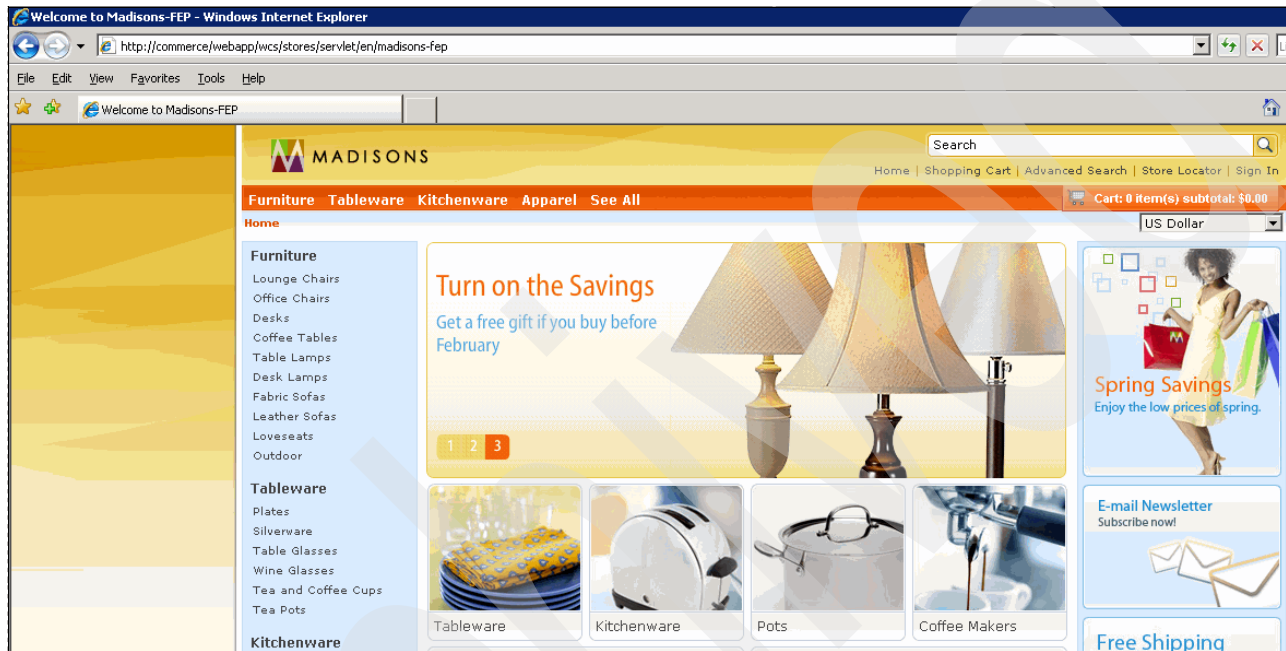


Figure 24 Sample commerce page

After logging into the DataPower XC10 appliance, navigate to the **Monitoring** menu as shown in Figure 25.

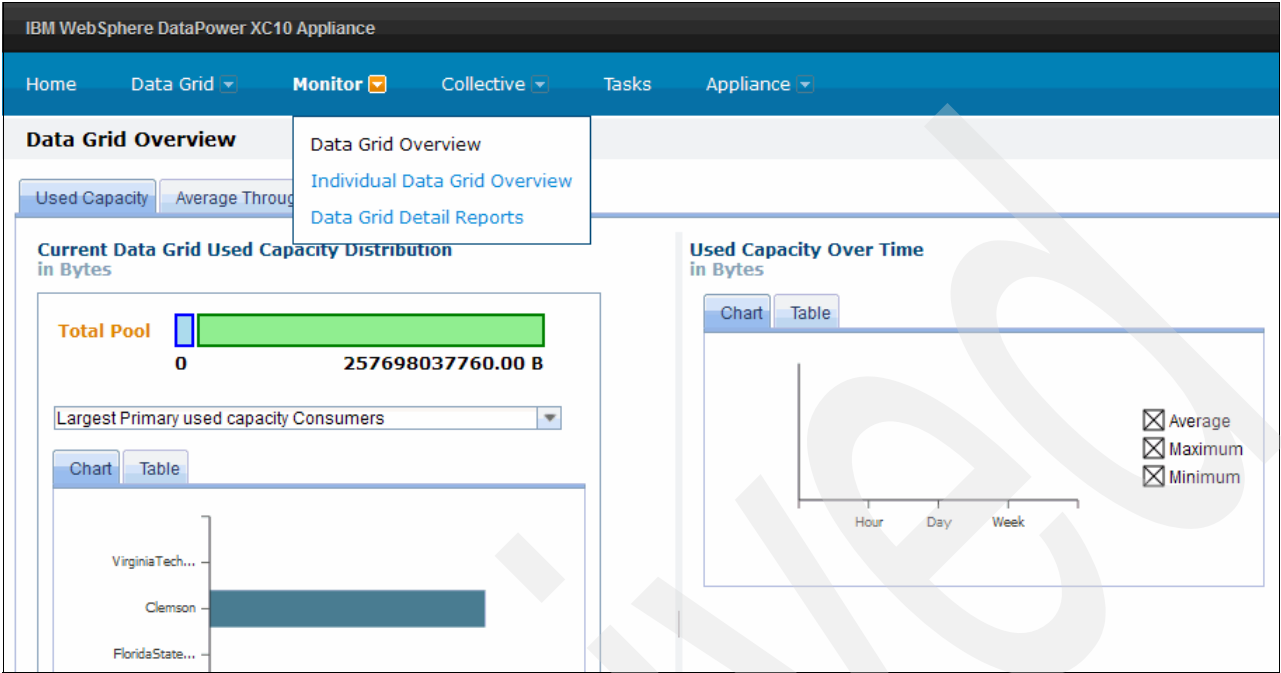


Figure 25 Monitoring menu options

To view your specific dynamic cache grid, expand **Monitor** → **Individual Data Grid Overview** → **dynamic_data_grid_name**. The resulting summary page (as shown in Figure 26) shows the number of cache entries, the average transaction time, average throughput, cache hit rate, and the percentage of limited capacity over the last 30 seconds.

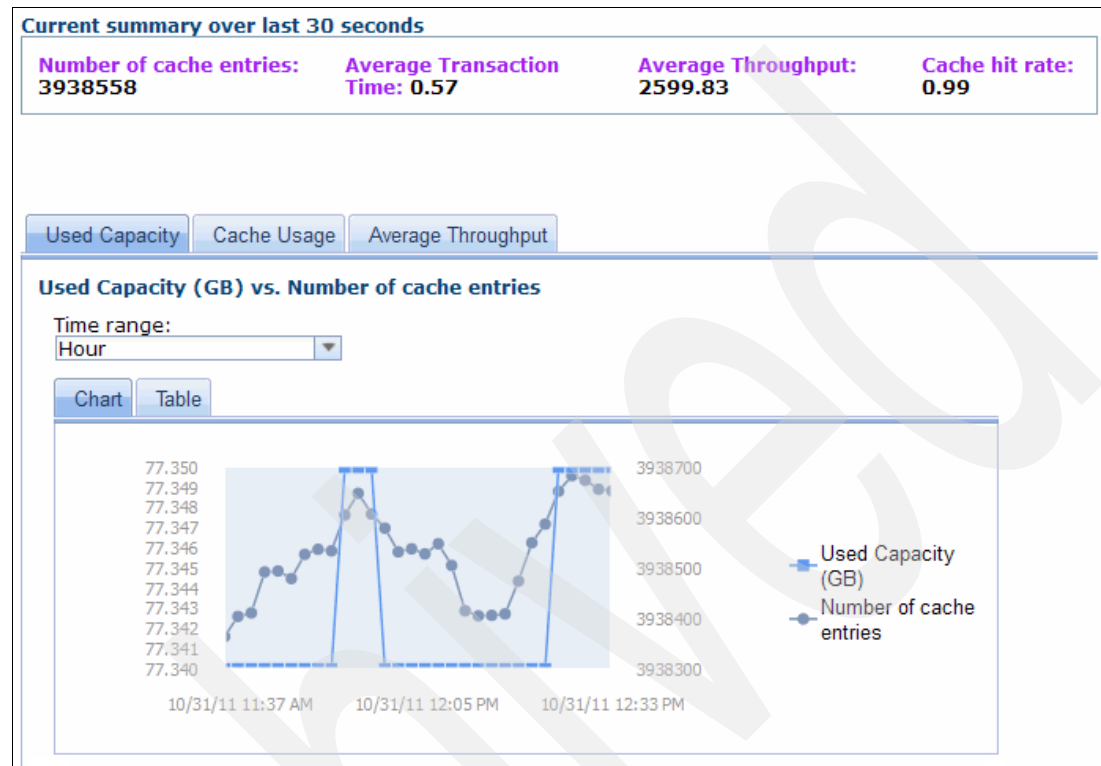


Figure 26 Sample data grid summary results

Interpreting the number of cache entries

For dynamic cache data grids, 166 cache entries are created by default for every dynamic cache data grid (83 partitions x 2):

- ▶ Each partition exists as a primary copy, or shard, and also as a replica shard for backing up the data.
- ▶ Each shard has two entries, one to keep track of statistics and the other to store the configuration.

However, because the grid does not get initialized until at least one entry is placed on the grid, the minimum number of entries you are likely to see is 167. This number will be displayed before you add any data to the grid. These cache entries contain dynamic cache provider statistics and the dynamic cache configuration for your WebSphere Commerce Server.

Used capacity

The Used Capacity chart (Figure 27) shows the used capacity of the cache versus the actual number of entries and the configured capacity limit of the cache. You can edit the time range that is displayed, over the last hour, last day, last week, and last month. The level of detail that is shown on the chart varies depending on the time range that you select.

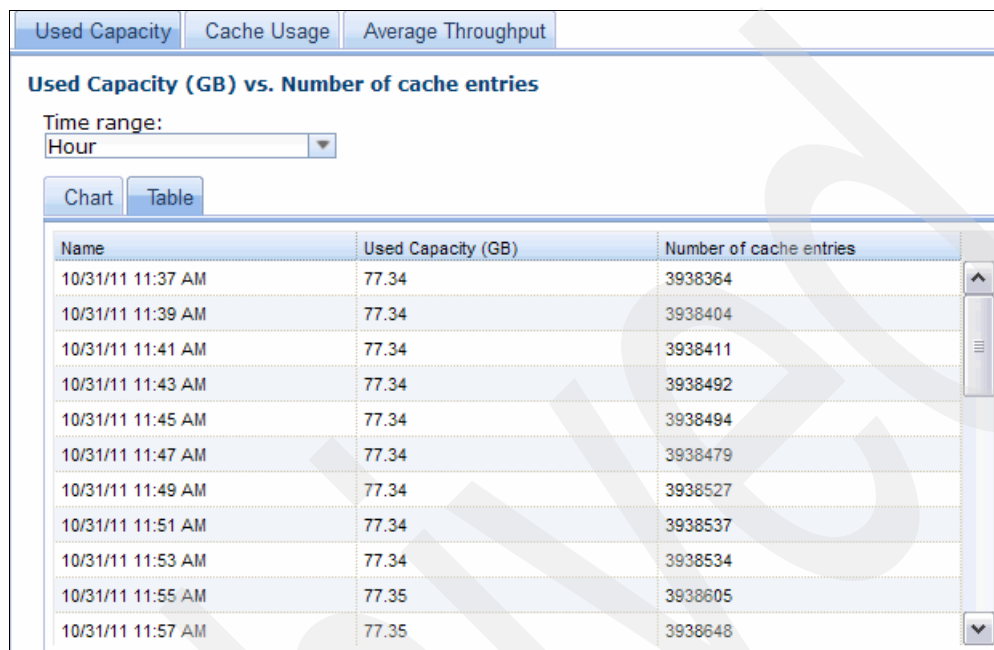


Figure 27 Used capacity

Cache usage

The Cache Usage chart (Figure 28) helps you visualize the number of successful queries to the cache. You can display cache attempts, cache hits, and the cache hit rate in the chart.

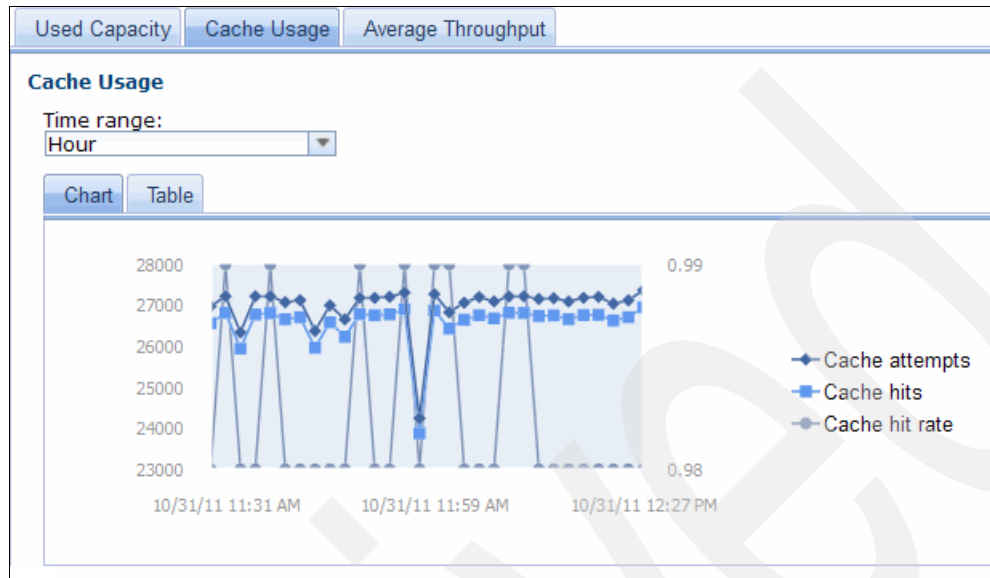


Figure 28 Sample results of Cache Usage

Average Throughput

The Average Throughput chart (Figure 29) displays the average number of transactions per second that are being processed over a given time range and the average length of time for each transaction.

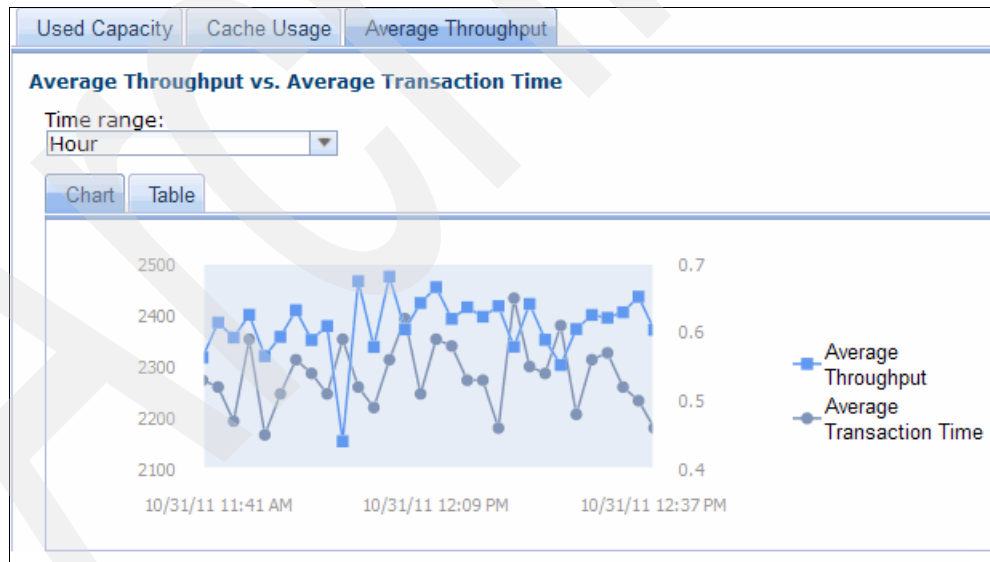


Figure 29 Sample results of Average throughput

Data grid/map details

To further view specific details of the data grid, select **Monitor** → **Data grid detail reports**. A tree displays the data grids. Select your data grid and drill down to see the specifics of that grid.

You can view the used capacity and a list of zones to which the data grid belongs from the detail reports (Figure 30).

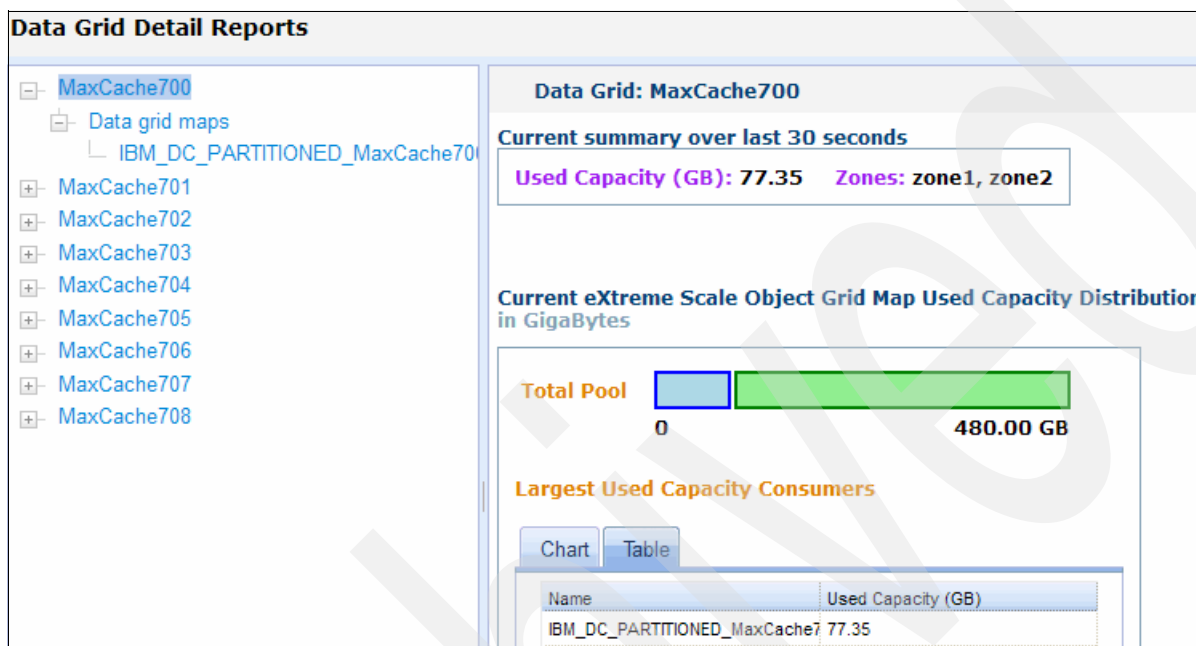


Figure 30 Sample data grid detail report

You can view further information about the maps in each grid. Select a map to view its total number of cache entries, average throughput, average transaction time, and the total capacity (Figure 31).

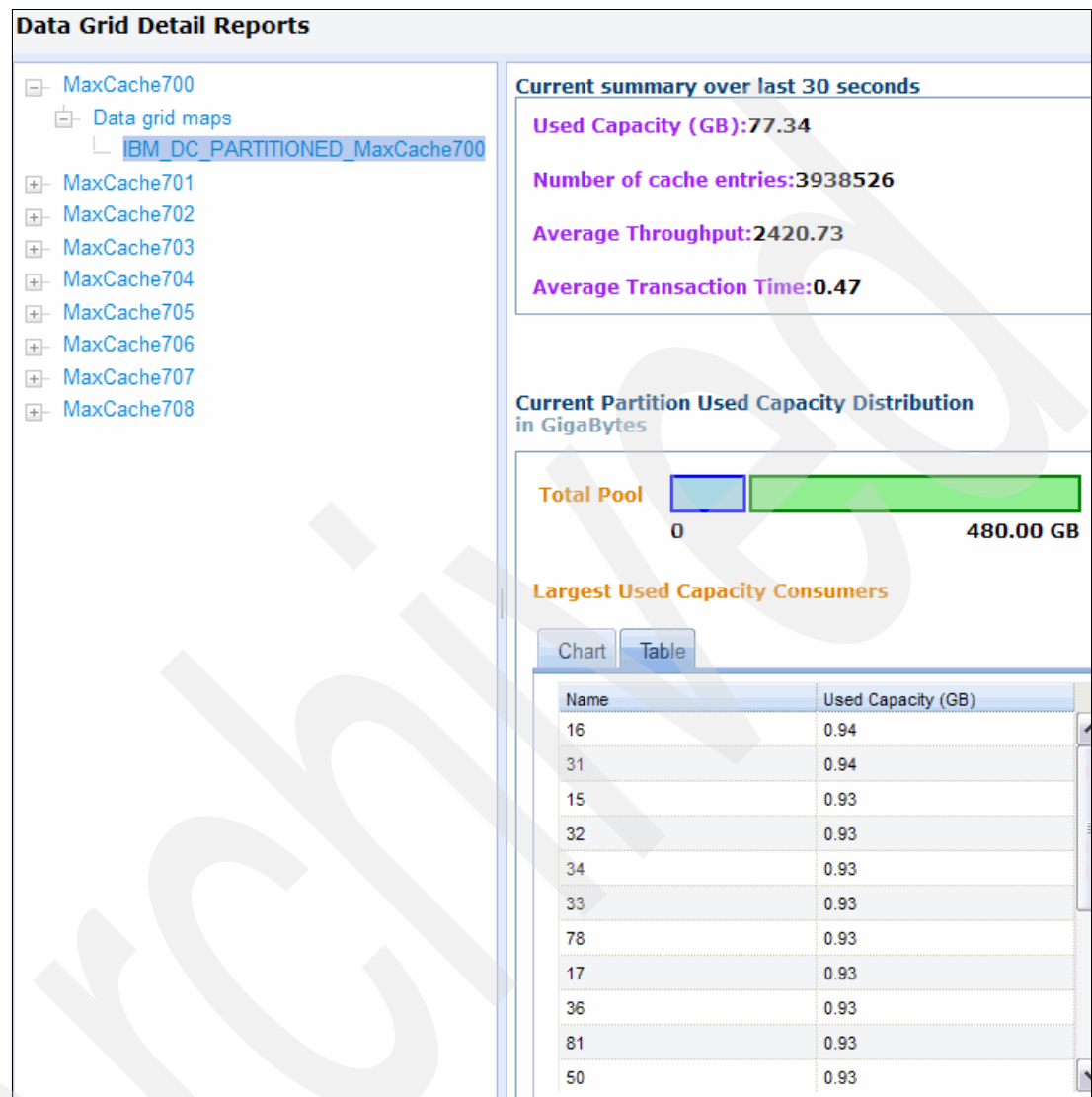


Figure 31 Sample data grid map results

Benefits of integrating WebSphere Commerce with WebSphere DataPower XC10

The IBM DynaCache API, as shown in Figure 32, is available to Java EE applications that are deployed in WebSphere Application Server. In the past, the only service provider for the dynamic cache API was the default dynamic cache engine built into WebSphere Application Server.

With the introduction of WebSphere eXtreme Scale, customers can use the dynamic cache service provider interface in WebSphere Application Server to plug eXtreme Scale into dynamic cache. This enables applications written with the dynamic cache API or applications using container-level caching (such as servlets) to take advantage of the features and performance capabilities of WebSphere eXtreme Scale.

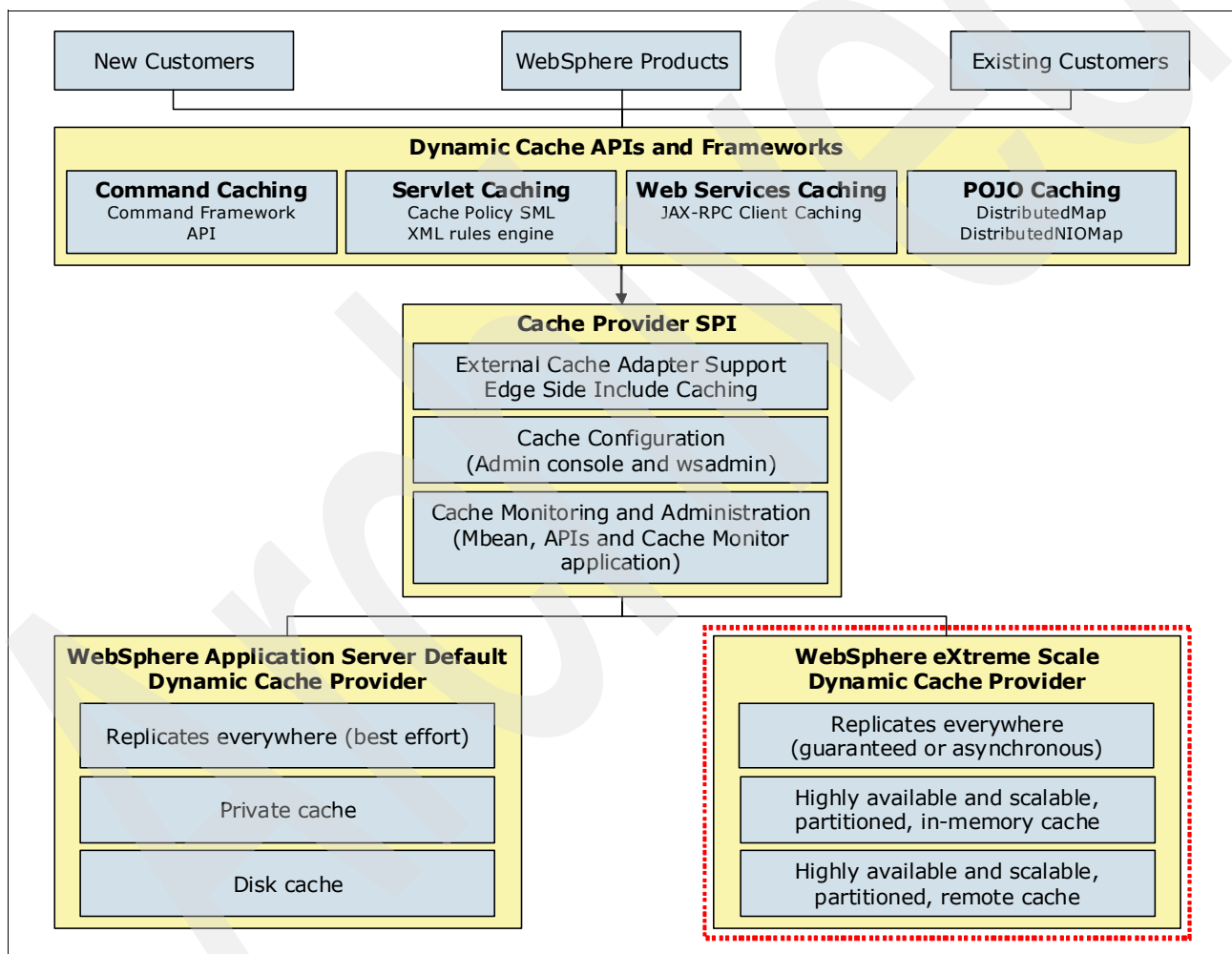


Figure 32 Benefits of WebSphere eXtreme scale dynamic cache provider

As you have seen, DataPower XC10 uses the WebSphere eXtreme Scale dynamic cache provider, which offers a range of benefits over the WebSphere default dynamic cache provider that comes with WebSphere. The features in WebSphere eXtreme Scale significantly increase the distributed capabilities of the dynamic cache API beyond what is offered by the default dynamic cache engine and data replication service (DRS). Here are several of these benefits:

- ▶ With both eXtreme Scale and DataPower XC10, you can create caches that are truly distributed across multiple servers, rather than just replicated and synchronized between the servers.
- ▶ Both eXtreme Scale and DataPower XC10 caches are transactional and highly available. All clients see the same contents from the dynamic cache service. (For example, a single invalidation invocation will invalidate that entry for all users of the cache.)
- ▶ Both WebSphere eXtreme Scale and DataPower XC10 offer a higher quality of service for cache replication than DRS.
- ▶ The default dynamic cache provider relies on disk cache to allow cache instances to grow beyond the size of a single process. The WebSphere eXtreme Scale dynamic cache provider eliminates the need for a disk cache and the expensive disk storage systems needed to make them perform.
- ▶ DataPower XC10 provides a caching layer that is independent of your client application. Therefore, client applications (including WebSphere Application Server) can be deployed on any platform and any configuration (32 bit or 64 bit) and still use the DataPower XC10 caching layer.
- ▶ The WebSphere eXtreme Scale dynamic cache provider has a performance feature called *compression* that can help users manage cache overhead. The default dynamic cache provider does not allow for compression of cached data in memory. With the eXtreme Scale provider, this becomes possible. Cache compression using the deflate algorithm can be enabled on any of the supported topologies. Enabling compression will increase the overhead for read and write operations but will drastically increase cache density for applications like servlet and JSP caching.
- ▶ The WebSphere eXtreme Scale dynamic cache provider can offer better performance than the default dynamic cache provider on large multi-processor enterprise servers because the eXtreme Scale code path is designed to maximize in-memory cache concurrency.
- ▶ The WebSphere eXtreme Scale dynamic cache provider eliminates the "warm-up time" after re-starting a WebSphere server that is required when using the default dynamic cache provider because a single cache is used for all WebSphere Commerce servers/cluster members.
 - Dynamic caching allows for easier maintenance and expansion of WebSphere Commerce servers
 - Dynamic caching also allows for easier/faster recovery from outages or unexpected failures of WebSphere Commerce servers.

Summary

Although WebSphere Application Server dynacache with disk offload can provide WebSphere Commerce sites with excellent caching and performance characteristics, WebSphere commerce integration with WebSphere DataPower XC10 adds the many benefits discussed in this paper. By configuring the dynamic cache service for WebSphere commerce to use the WebSphere DataPower XC10 appliance, less memory is used by the WebSphere Commerce Servers. The appliance allows for extremely large cache sizes and can be extended by adding additional appliances to a collective. Because data is stored in a central location, it can be easily shared among all servers within (or even across) a cell. Performance improvements are substantial, and configuration steps required for the integration are relatively simple.

Related materials

- ▶ Information on commerce integration with WebSphere eXtreme scale can be found in the IBM Redbook *Scalable, Integrated Solutions for Elastic Caching Using IBM WebSphere eXtreme Scale*, SG24-7926. (See chapter 6, Integrating WebSphere Commerce with WebSphere eXtreme Scale.)
- ▶ Configure WebSphere Commerce to use WebSphere eXtreme Scale for dynamic cache to improve performance and scale
<https://www.ibm.com/developerworks/wikis/display/extremescale/Configure+WebSphere+Commerce+to+use+WebSphere+eXtreme+Scale+for+dynamic+cache+to+improve+performance+and+scale>
- ▶ Enabling WebSphere Commerce data cache
<http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/topic/com.ibm.commerce.admin.doc/tasks/tdcenabcommdatcache.htm>
- ▶ Creating dynamic cache data grids
<http://publib.boulder.ibm.com/infocenter/wdpxc/v2r0/topic/com.ibm.websphere.datapower.xc.doc/tdyncache.html>
- ▶ Enhancing WebSphere commerce performance with WebSphere eXtreme scale
http://www.ibm.com/developerworks/websphere/techjournal/1008_genkin/1008_genkin.html?ca=drs-

About the author



Chao M Beck is a technical lead for IBM WebSphere DataPower XC10 early program. She has long been a member of IBM Application and Integration Middleware Software (AIM) early programs team and is responsible for the execution of early programs for IBM WebSphere Application Server products. She handles the development and delivery of education for new product functions and the provision of customer support during pre-GA (early) programs and post-GA (customer acceleration) programs. Prior to her current role, she was the technical lead for the feature pack for Service Component Architecture (SCA) early program, where she contributed to several white paper publications in the area of service component architecture. Previously, Chao was involved with AIM early programs in the area of tooling and trial programs and has also held positions in software testing and development, including Java and WAS for Telecom, Universal Discovery, Description and Integration, slingshot, and support for ISVs.

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:
<http://www.facebook.com/IBMRedbooks>
- ▶ Follow us on Twitter:
<http://twitter.com/ibmredbooks>
- ▶ Look for us on LinkedIn:
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new IBM Redbooks® publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:
<http://www.redbooks.ibm.com/rss.html>

Archived

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

This document REDP-4823-00 was created or updated on December 8, 2011.



Send us your comments in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:
ibm.com/redbooks
- ▶ Send your comments in an email to:
redbooks@us.ibm.com
- ▶ Mail your comments to:
IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400 U.S.A.




Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

DataPower®
developerWorks®
IBM®

Redbooks®
Redpapers™
Redbooks (logo) ®

WebSphere®

The following terms are trademarks of other companies:

Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.