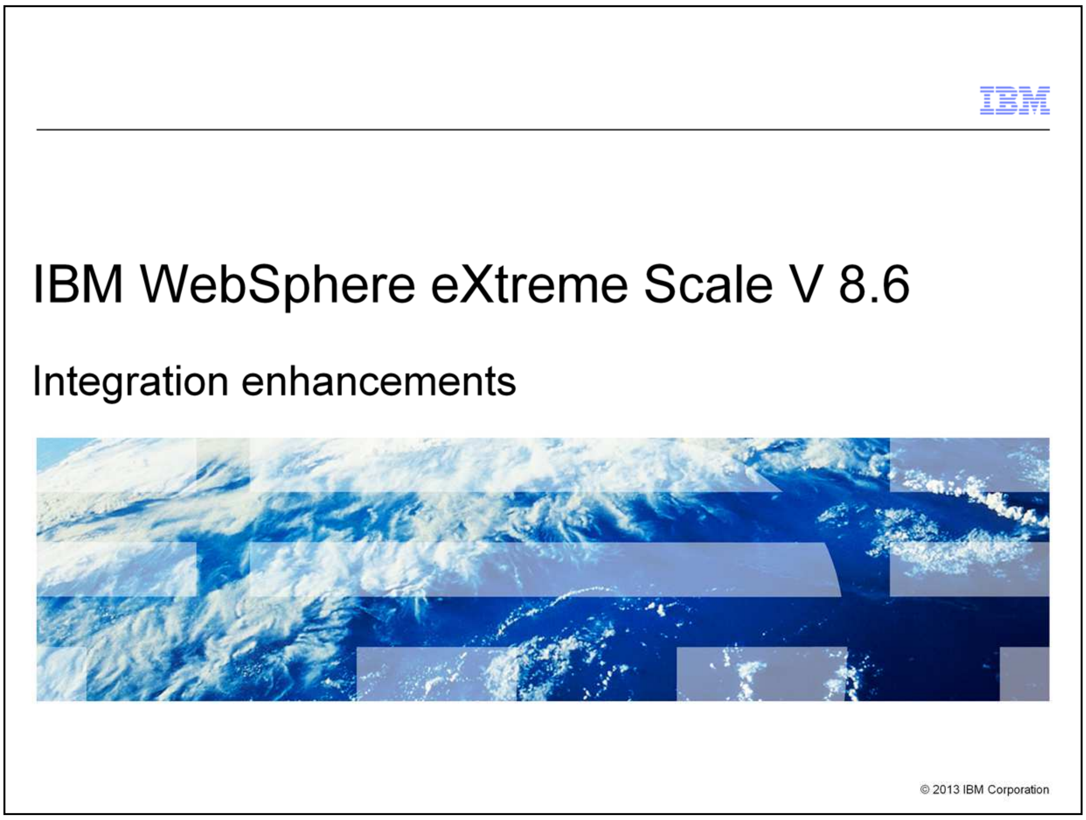


IBM WebSphere eXtreme Scale V 8.6

Integration enhancements

WebSphere® eXtreme Scale version 8.6 includes several enhancements related to integration.

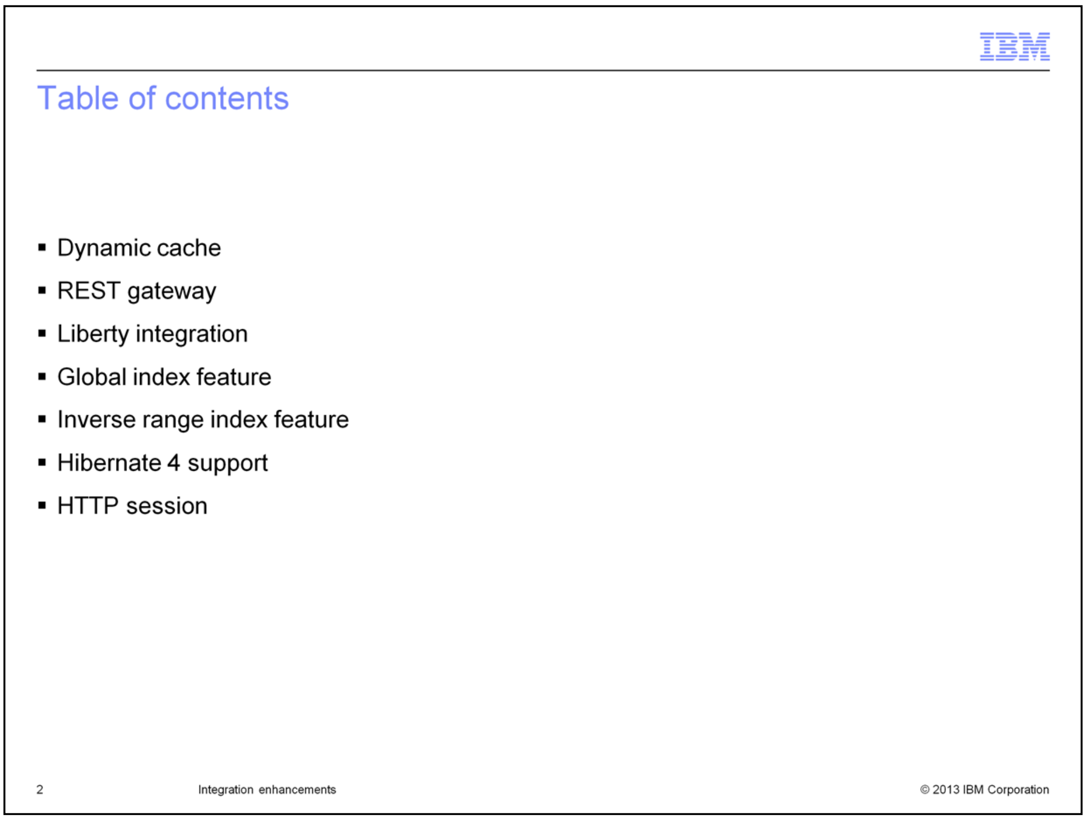
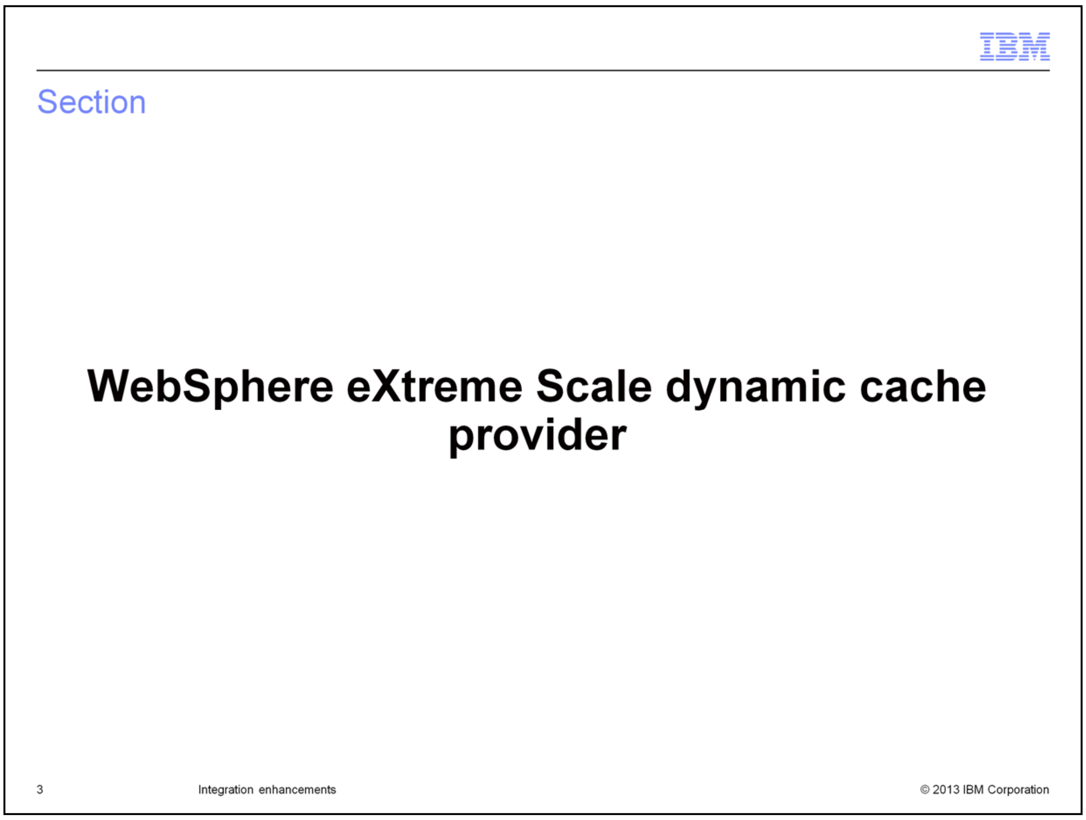


Table of contents

This presentation covers the enhancements related to the dynamic cache plug-in, REST gateway function, integration with the Liberty profile, the new global index and inverse range index features, hibernate version 4 support, and the HTTP session plug-in.

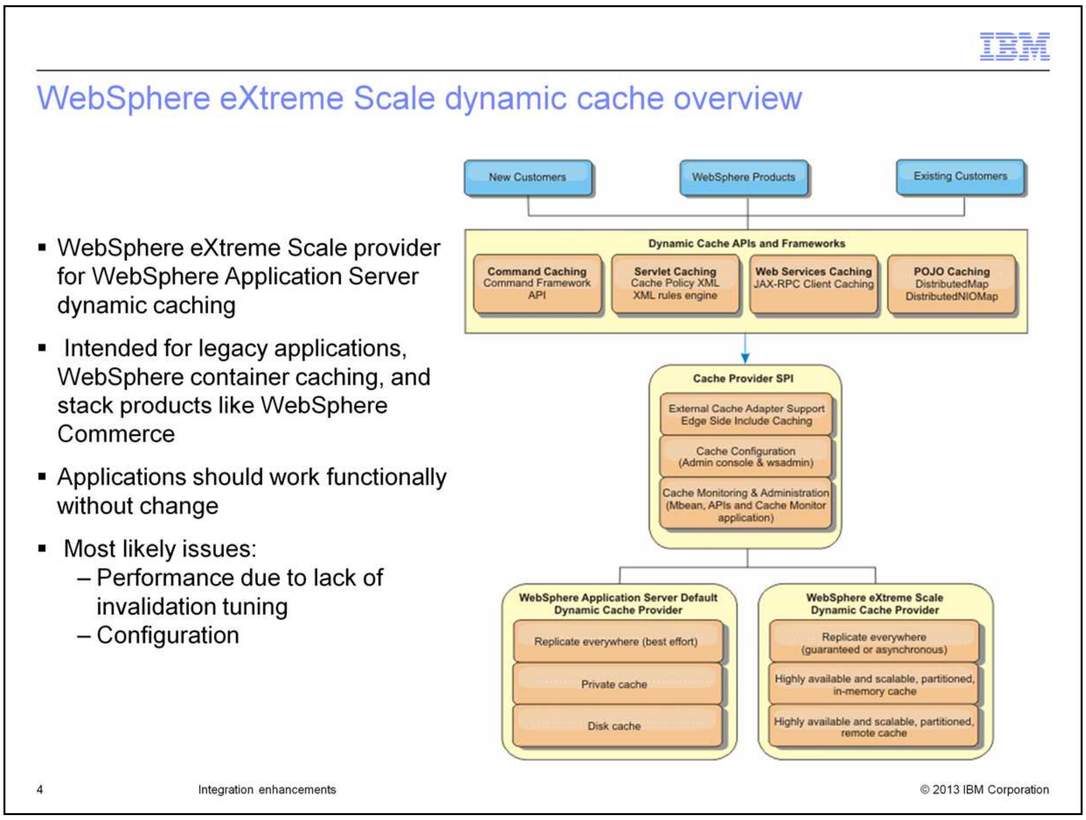


Section

WebSphere eXtreme Scale dynamic cache provider

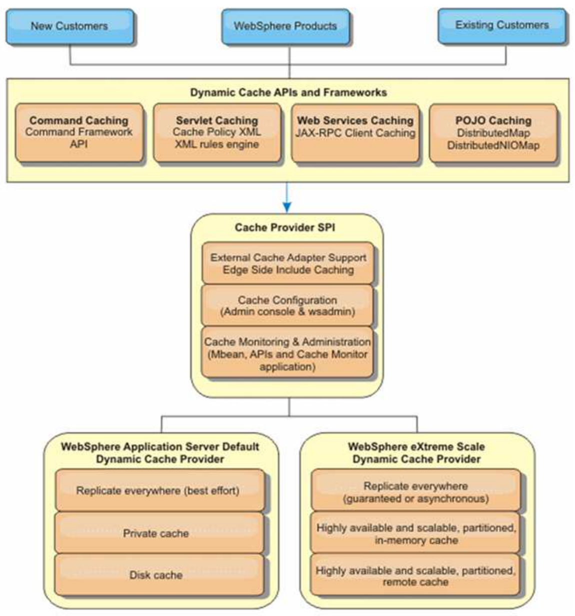
3 Integration enhancements © 2013 IBM Corporation

This section covers the dynamic cache plug-in support.



WebSphere eXtreme Scale dynamic cache overview

- WebSphere eXtreme Scale provider for WebSphere Application Server dynamic caching
- Intended for legacy applications, WebSphere container caching, and stack products like WebSphere Commerce
- Applications should work functionally without change
- Most likely issues:
  - Performance due to lack of invalidation tuning
  - Configuration

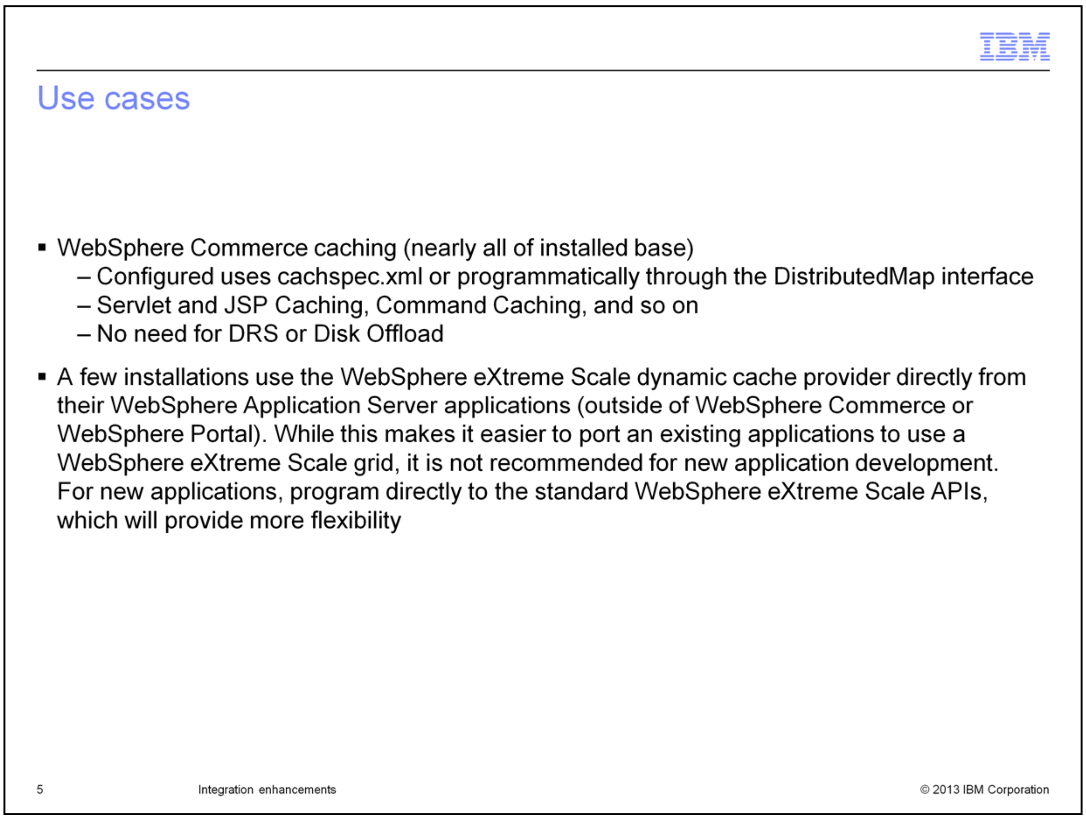


New Customers
WebSphere Products
Existing Customers
Dynamic Cache APIs and Frameworks
Command Caching Command Framework API
Servlet Caching Cache Policy XML XML rules engine
Web Services Caching JAX-RPC Client Caching
POJO Caching DistributedMap DistributedNIOMap
Cache Provider SPI
External Cache Adapter Support Edge Side Include Caching
Cache Configuration (Admin console & wsadmin)
Cache Monitoring & Administration (Mbean, APIs and Cache Monitor application)
WebSphere Application Server Default Dynamic Cache Provider
Replicate everywhere (best effort)
Private cache
Disk cache
WebSphere eXtreme Scale Dynamic Cache Provider
Replicate everywhere (guaranteed or asynchronous)
Highly available and scalable, partitioned, in-memory cache
Highly available and scalable, partitioned, remote cache

4 Integration enhancements © 2013 IBM Corporation

The WebSphere Application Server provides a dynamic cache service that is available to deployed Java EE applications. This service is used to cache data such as output from servlets, JSPs or commands, and object data programmatically specified within an enterprise application using the DistributedMap application programming interface.

Initially, the only service provider for the dynamic cache service was the default dynamic cache engine that is built into the WebSphere Application Server. You can also specify WebSphere eXtreme Scale to be the cache provider for any given cache instance. By setting up this capability, you can enable applications that use the dynamic cache service, to use the features and performance capabilities of WebSphere eXtreme Scale.

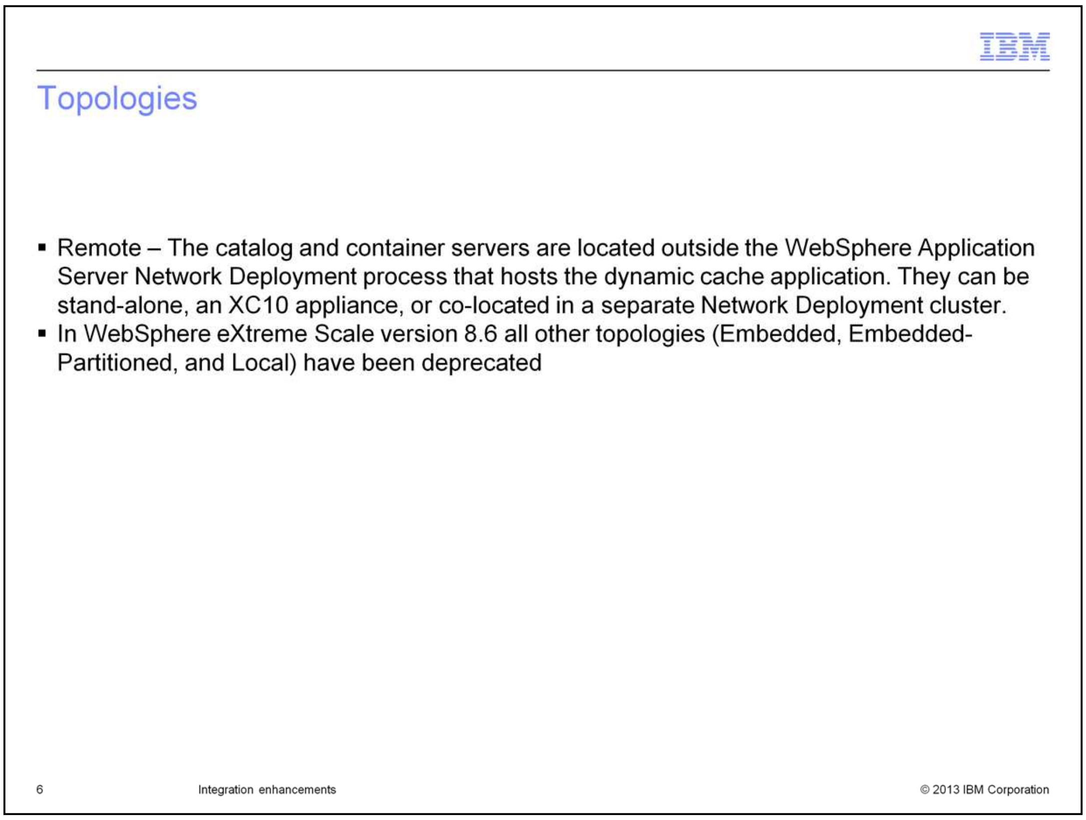


Use cases

- WebSphere Commerce caching (nearly all of installed base)
  - Configured uses cachspec.xml or programmatically through the DistributedMap interface
  - Servlet and JSP Caching, Command Caching, and so on
  - No need for DRS or Disk Offload
- A few installations use the WebSphere eXtreme Scale dynamic cache provider directly from their WebSphere Application Server applications (outside of WebSphere Commerce or WebSphere Portal). While this makes it easier to port an existing applications to use a WebSphere eXtreme Scale grid, it is not recommended for new application development. For new applications, program directly to the standard WebSphere eXtreme Scale APIs, which will provide more flexibility

5 Integration enhancements © 2013 IBM Corporation

Most users of the WebSphere eXtreme Scale dynamic cache provider use it in combination with WebSphere Commerce or the WebSphere Portal Server. While it is not practical or recommended for all WebSphere Commerce or WebSphere Portal Server dynamic cache instances to use the WebSphere eXtreme Scale dynamic cache provider, its use on selected cache instances can greatly improve the efficiency, performance, and availability. For new applications that do not use WebSphere Commerce or WebSphere Portal Server it is preferable to use the WebSphere eXtreme Scale APIs directly rather than using the WebSphere Application Server dynamic cache APIs.



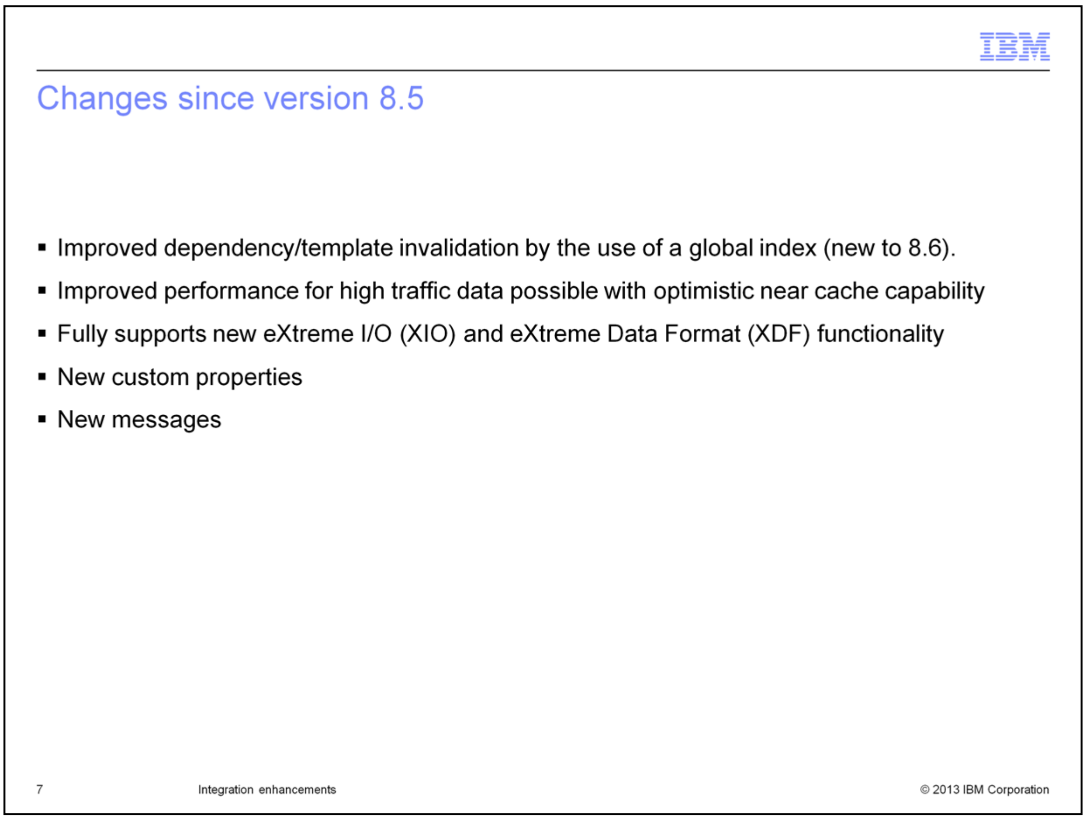
Topologies

- Remote – The catalog and container servers are located outside the WebSphere Application Server Network Deployment process that hosts the dynamic cache application. They can be stand-alone, an XC10 appliance, or co-located in a separate Network Deployment cluster.
- In WebSphere eXtreme Scale version 8.6 all other topologies (Embedded, Embedded-Partitioned, and Local) have been deprecated

6 Integration enhancements © 2013 IBM Corporation

For the WebSphere eXtreme Scale dynamic cache provider the only recommended topology is “remote”, where the extreme scale server exists in a separate tier that will connect remotely with the application server tier.

All other topologies have been deprecated in version 8.6.



Changes since version 8.5

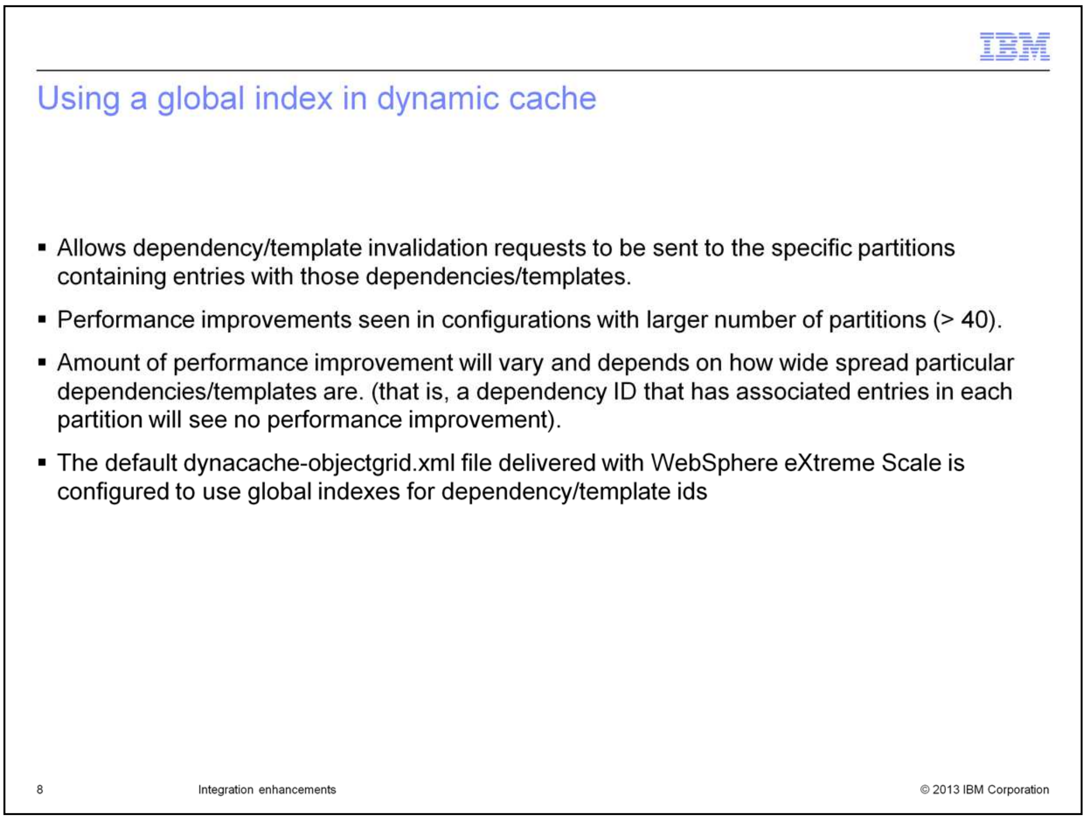
- Improved dependency/template invalidation by the use of a global index (new to 8.6).
- Improved performance for high traffic data possible with optimistic near cache capability
- Fully supports new eXtreme I/O (XIO) and eXtreme Data Format (XDF) functionality
- New custom properties
- New messages

7 Integration enhancements © 2013 IBM Corporation

Several new features have been added to the eXtreme Scale dynamic cache provider in version 8.6.

A new Global Index feature has been added that will improve the performance of dependency and template ID invalidations. A local near-caching capability is also now available.

In addition, the dynamic cache provider supports the new extreme I/O and extreme data format capabilities along with some new custom properties and messages.



Using a global index in dynamic cache

- Allows dependency/template invalidation requests to be sent to the specific partitions containing entries with those dependencies/templates.
- Performance improvements seen in configurations with larger number of partitions (> 40).
- Amount of performance improvement will vary and depends on how wide spread particular dependencies/templates are. (that is, a dependency ID that has associated entries in each partition will see no performance improvement).
- The default dynacache-objectgrid.xml file delivered with WebSphere eXtreme Scale is configured to use global indexes for dependency/template ids

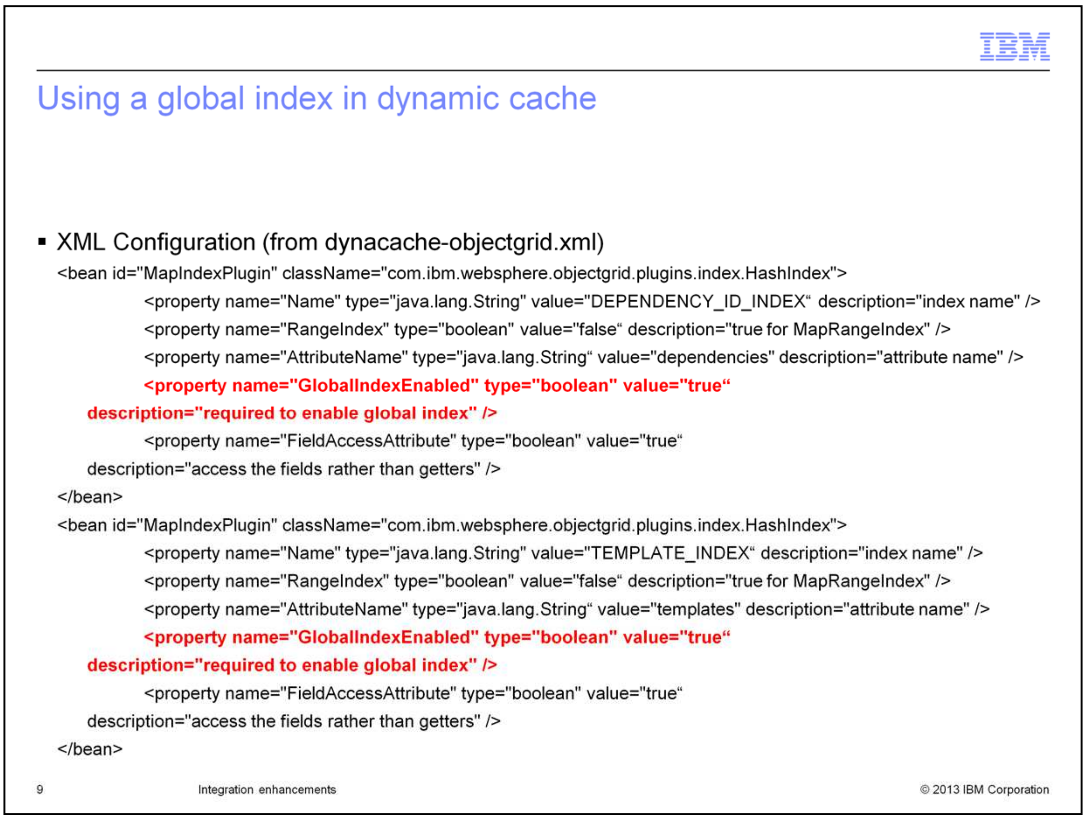
8 Integration enhancements © 2013 IBM Corporation

One of the historical “pain points” of the WebSphere eXtreme Scale dynamic cache provider has been seen in the area of cache invalidations. By design, data within an eXtreme Scale grid is distributed among the partitions that make up the grid. Invalidation events historically had to flow to every partition within the grid to ensure that any entry with a particular dependency or template ID would be invalidated when that dependency or template ID was invalidated. As a result, large amounts of invalidation traffic can put significant strain on network and system resources.

The global index function added in version 8.6 allows eXtreme Scale to identify exactly which partitions contain entries that have a particular dependency or template ID associated with them, and invalidation events will only flow to that sub-set of partitions, minimizing the load on network and system resources.

The performance improvement seen for any given scenario will depend on how wide spread across the grid entries with specific dependency or template IDs are. For example, invalidation of a dependency ID that has entries in every partition across the grid will see no performance improvement, while invalidation of a dependency ID that has entries in only a small number of partitions will experience much better performance.

The global index function is configured and automatically enabled in the dynacache-objectgrid.xml file installed with version 8.6.



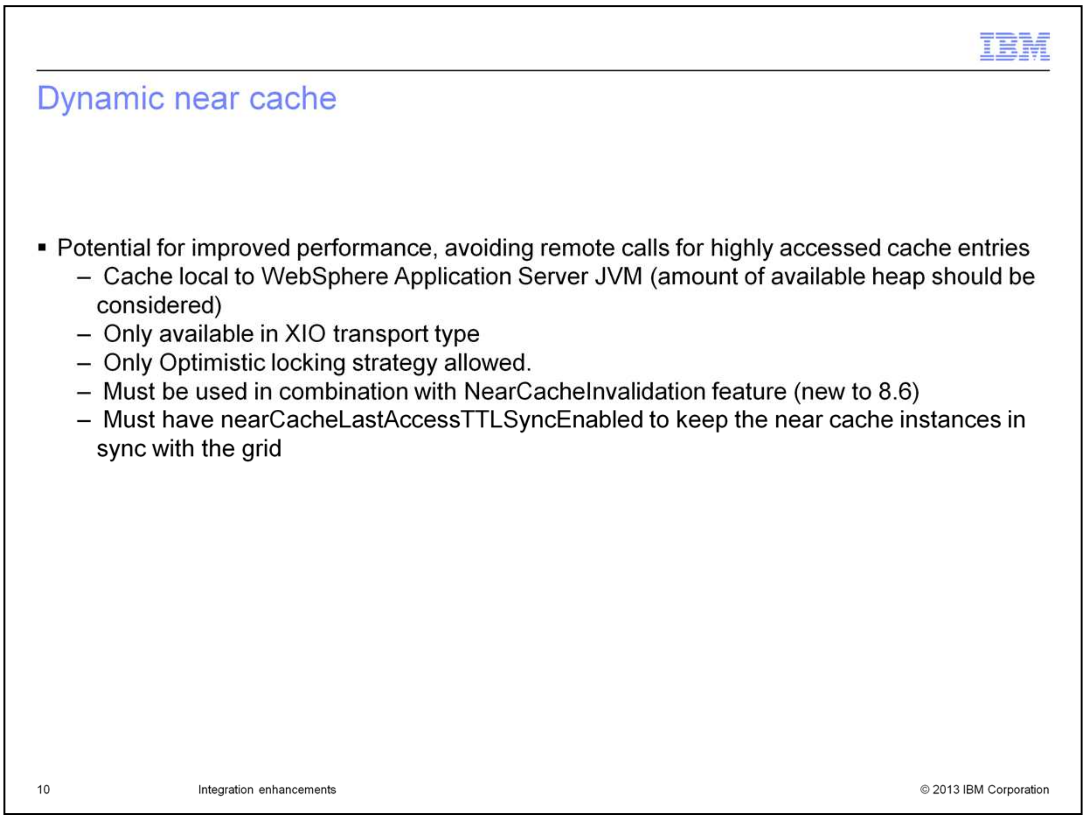
Using a global index in dynamic cache

- XML Configuration (from dynacache-objectgrid.xml)

```
<bean id="MapIndexPlugin" className="com.ibm.websphere.objectgrid.plugins.index.HashIndex">
            <property name="Name" type="java.lang.String" value="DEPENDENCY_ID_INDEX“ description="index name" />
            <property name="RangeIndex" type="boolean" value="false“ description="true for MapRangeIndex" />
            <property name="AttributeName" type="java.lang.String“ value="dependencies" description="attribute name" />
            <property name="GlobalIndexEnabled" type="boolean" value="true“
    description="required to enable global index" />
            <property name="FieldAccessAttribute" type="boolean" value="true“
    description="access the fields rather than getters" />
</bean>
<bean id="MapIndexPlugin" className="com.ibm.websphere.objectgrid.plugins.index.HashIndex">
            <property name="Name" type="java.lang.String" value="TEMPLATE_INDEX“ description="index name" />
            <property name="RangeIndex" type="boolean" value="false“ description="true for MapRangeIndex" />
            <property name="AttributeName" type="java.lang.String“ value="templates" description="attribute name" />
            <property name="GlobalIndexEnabled" type="boolean" value="true“
    description="required to enable global index" />
            <property name="FieldAccessAttribute" type="boolean" value="true“
    description="access the fields rather than getters" />
</bean>
```

9 Integration enhancements © 2013 IBM Corporation

This chart shows how the required dependency and template indexes are configured within the dynacache-objectgrid.xml file that is delivered with version 8.6. No changes to this configuration are required.



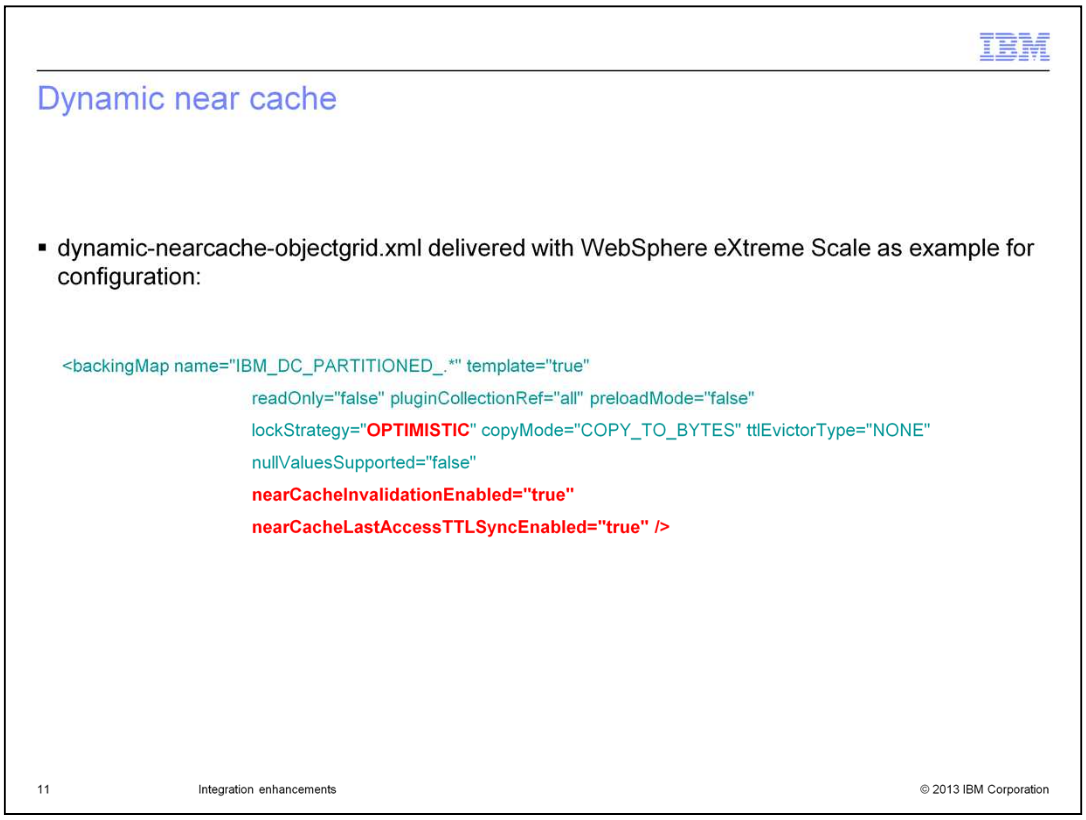
Dynamic near cache

- Potential for improved performance, avoiding remote calls for highly accessed cache entries
  - Cache local to WebSphere Application Server JVM (amount of available heap should be considered)
  - Only available in XIO transport type
  - Only Optimistic locking strategy allowed.
  - Must be used in combination with NearCacheInvalidation feature (new to 8.6)
  - Must have nearCacheLastAccessTTLSyncEnabled to keep the near cache instances in sync with the grid

10 Integration enhancements © 2013 IBM Corporation

A near cache function has also been added to the WebSphere eXtreme Scale dynamic cache provider in version 8.6. A near cache is a data cache that is local to the application's JVM and is an option that allows a sub-set of "highly used" entries within the remote eXtreme Scale data grid to be accessed locally, saving the cost of remote access across the network to the remote data grid.

As with all eXtreme Scale near cache instances, the optimistic locking strategy must be used. In addition, the new NearCacheInvalidation and nearCacheLastAccessTTLSyncEnabled functions are required so that the dynamic near cache instances in each local JVM are kept in sync with the remote data grid.



Dynamic near cache

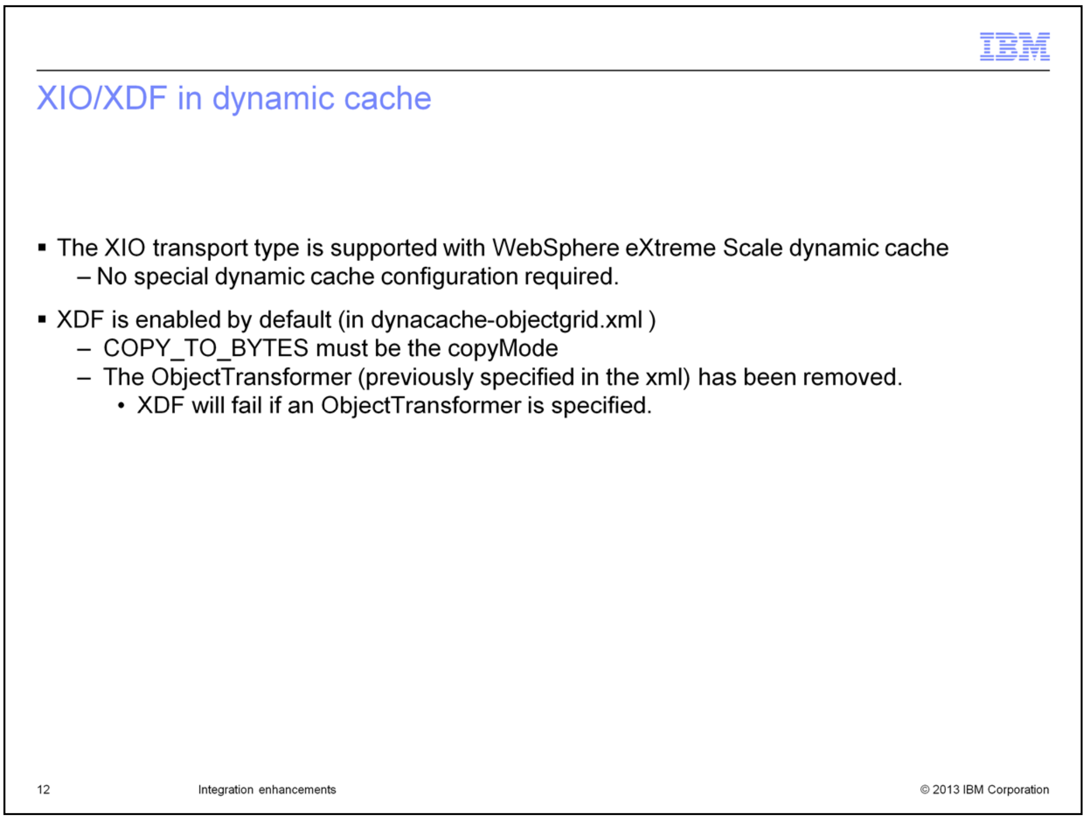
- dynamic-nearcache-objectgrid.xml delivered with WebSphere eXtreme Scale as example for configuration:

```
<backingMap name="IBM_DC_PARTITIONED_.*" template="true"
                       readOnly="false" pluginCollectionRef="all" preloadMode="false"
                       lockStrategy="OPTIMISTIC" copyMode="COPY_TO_BYTES" ttlEvictorType="NONE"
                       nullValuesSupported="false"
                       nearCacheInvalidationEnabled="true"
                       nearCacheLastAccessTTLSyncEnabled="true" />
```

11 Integration enhancements © 2013 IBM Corporation

The dynamic-nearcache-objectgrid.xml file that is delivered with eXtreme Scale 8.6 is configured to automatically enable a near cache.

This file can be copied and used when starting the eXtreme Scale catalog and container servers.

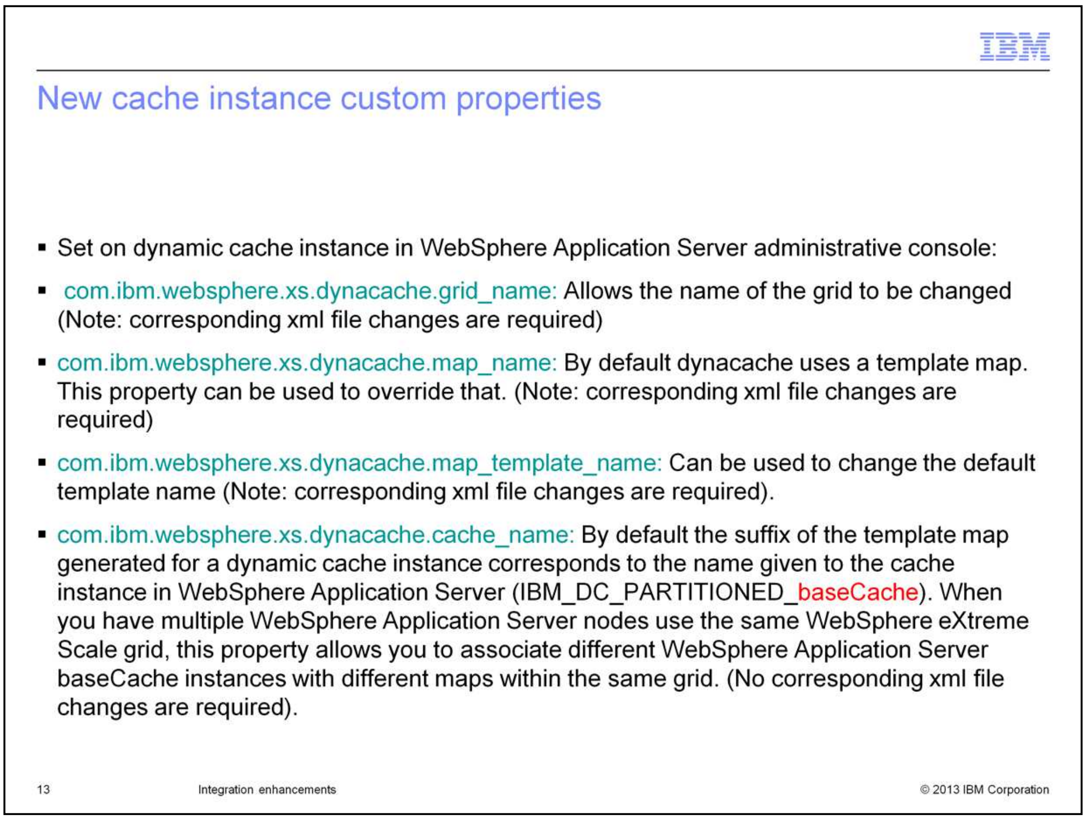


XIO/XDF in dynamic cache

- The XIO transport type is supported with WebSphere eXtreme Scale dynamic cache
  - No special dynamic cache configuration required.
- XDF is enabled by default (in dynacache-objectgrid.xml)
  - COPY_TO_BYTES must be the copyMode
  - The ObjectTransformer (previously specified in the xml) has been removed.
    - XDF will fail if an ObjectTransformer is specified.

12 Integration enhancements © 2013 IBM Corporation

The eXtreme I/O and Extreme Data Format functions are recommended default functions in eXtreme Scale version 8.6, and are configured as the default behavior in the dynacache-objectgrid.xml and dynacache-nearcache-objectgrid.xml files that are delivered with the product.



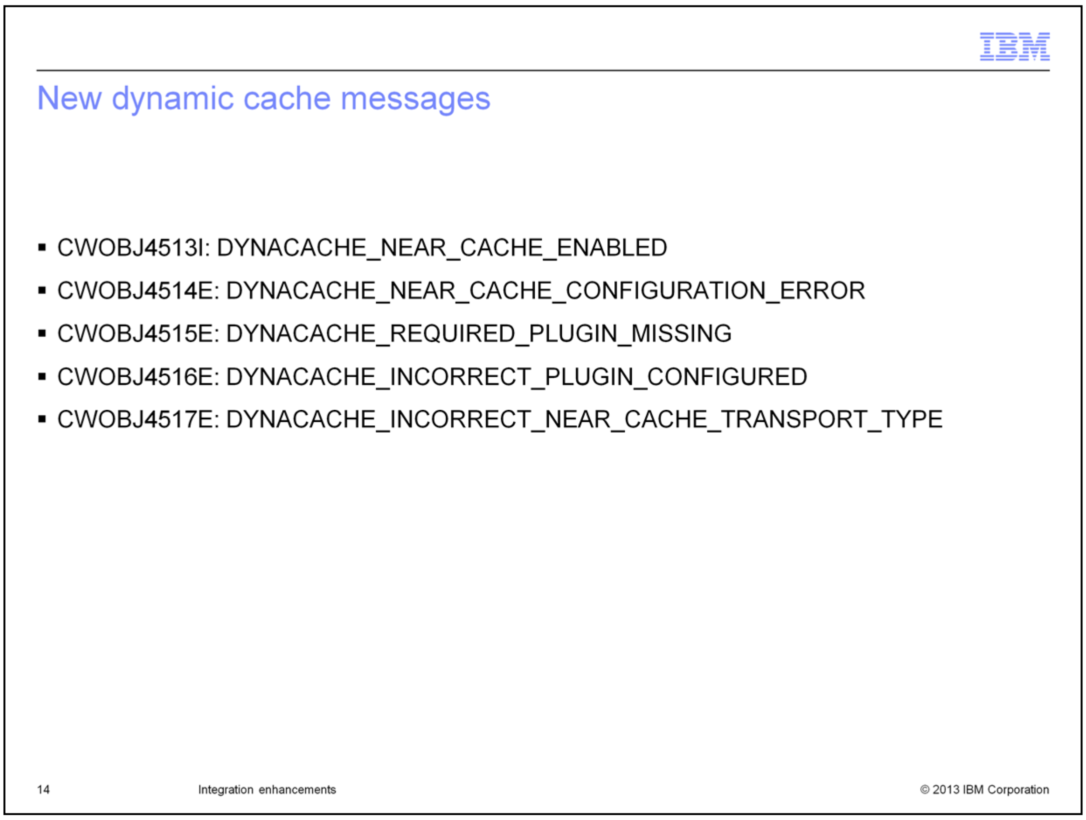
New cache instance custom properties

- Set on dynamic cache instance in WebSphere Application Server administrative console:
- com.ibm.websphere.xs.dynacache.grid_name: Allows the name of the grid to be changed (Note: corresponding xml file changes are required)
- com.ibm.websphere.xs.dynacache.map_name: By default dynacache uses a template map. This property can be used to override that. (Note: corresponding xml file changes are required)
- com.ibm.websphere.xs.dynacache.map_template_name: Can be used to change the default template name (Note: corresponding xml file changes are required).
- com.ibm.websphere.xs.dynacache.cache_name: By default the suffix of the template map generated for a dynamic cache instance corresponds to the name given to the cache instance in WebSphere Application Server (IBM_DC_PARTITIONED_baseCache). When you have multiple WebSphere Application Server nodes use the same WebSphere eXtreme Scale grid, this property allows you to associate different WebSphere Application Server baseCache instances with different maps within the same grid. (No corresponding xml file changes are required).

13 Integration enhancements © 2013 IBM Corporation

In version 8.6, several custom properties have been added that can be set on the dynamic cache instances in the WebSphere Application Server administrative console to allow for additional flexibility and control of individual cache instances.

The function and use of each of these custom properties are described within the eXtreme Scale version 8.6 information center.



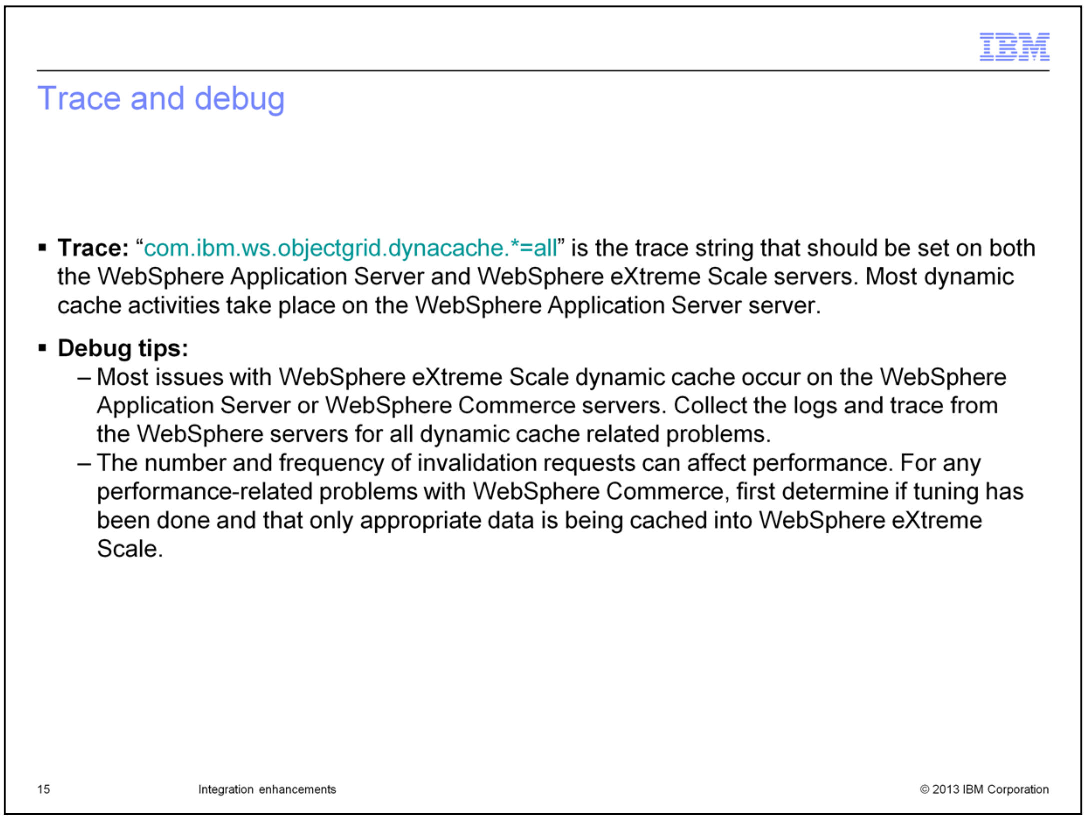
New dynamic cache messages

- CWOBJ4513I: DYNACACHE_NEAR_CACHE_ENABLED
- CWOBJ4514E: DYNACACHE_NEAR_CACHE_CONFIGURATION_ERROR
- CWOBJ4515E: DYNACACHE_REQUIRED_PLUGIN_MISSING
- CWOBJ4516E: DYNACACHE_INCORRECT_PLUGIN_CONFIGURED
- CWOBJ4517E: DYNACACHE_INCORRECT_NEAR_CACHE_TRANSPORT_TYPE

14 Integration enhancements © 2013 IBM Corporation

These informational and error messages have also been added for dynamic cache instances in eXtreme Scale version 8.6.

They are documented in the WebSphere eXtreme Scale version 8.6 information center.

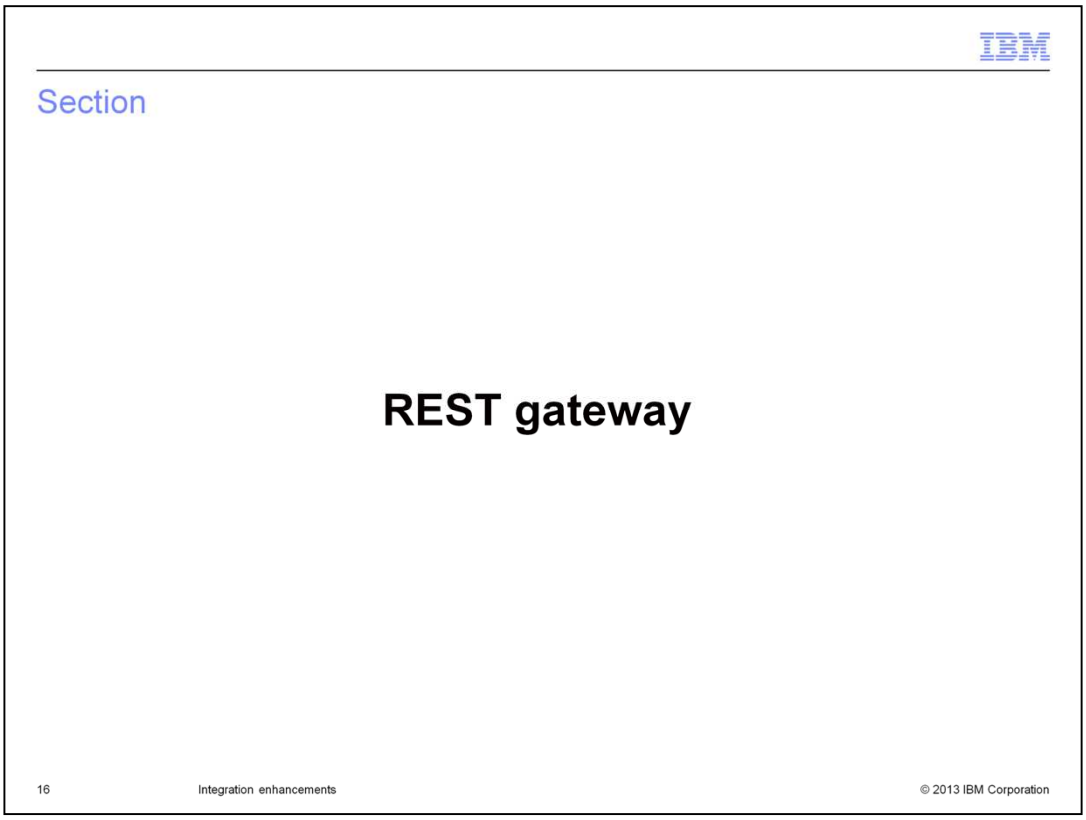


Trace and debug

- **Trace:** “com.ibm.ws.objectgrid.dynacache.*=all” is the trace string that should be set on both the WebSphere Application Server and WebSphere eXtreme Scale servers. Most dynamic cache activities take place on the WebSphere Application Server server.
- **Debug tips:**
  - Most issues with WebSphere eXtreme Scale dynamic cache occur on the WebSphere Application Server or WebSphere Commerce servers. Collect the logs and trace from the WebSphere servers for all dynamic cache related problems.
  - The number and frequency of invalidation requests can affect performance. For any performance-related problems with WebSphere Commerce, first determine if tuning has been done and that only appropriate data is being cached into WebSphere eXtreme Scale.

15 Integration enhancements © 2013 IBM Corporation

This slide shows the appropriate trace string that can be added to help with debugging issues related to eXtreme Scale dynamic caching. This trace can be added to both the WebSphere Application Server tier and the WebSphere eXtreme Scale tier.



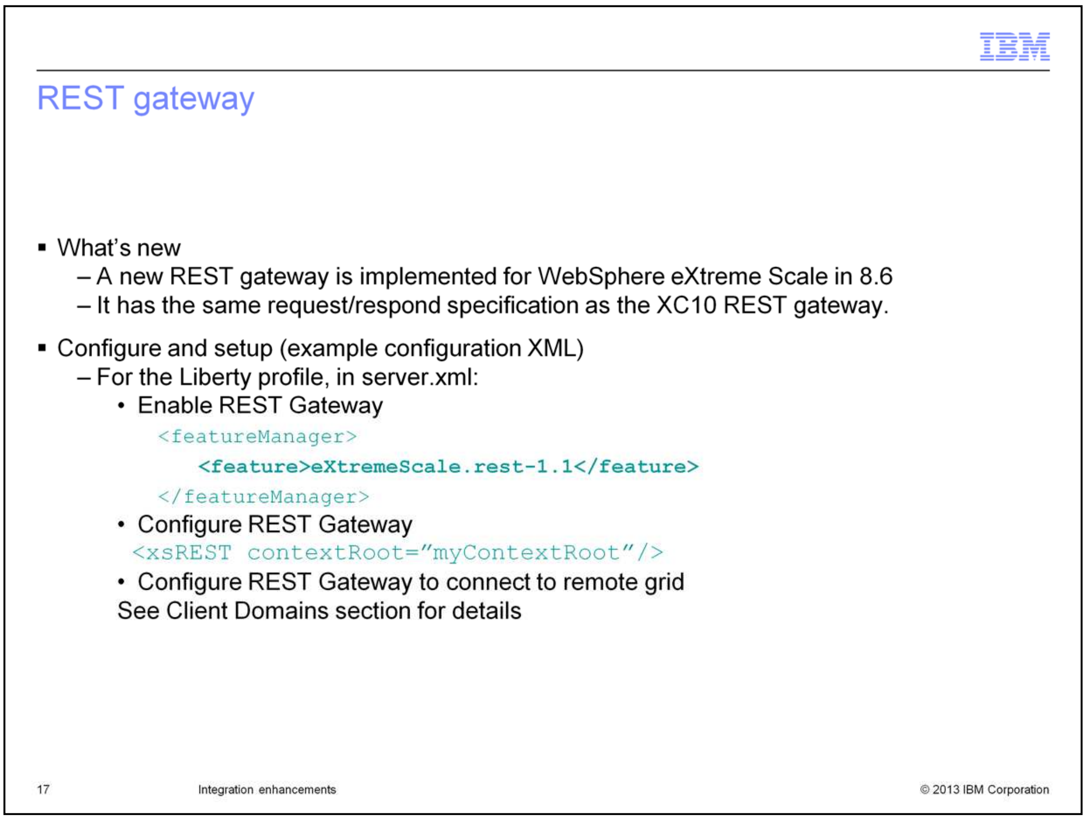
Section

REST gateway

16 Integration enhancements © 2013 IBM Corporation

With WebSphere eXtreme Scale version 8.6, the Representational State Transfer (REST) gateway can be used to access simple data grids that are hosted in eXtreme Scale.

The REST gateway is used to access eXtreme Scale data grid data from non-Java environments such as the DataPower® XI50 Appliance or a .NET application. You can also use the REST gateway to access map data from a Java virtual machine that cannot host the IBM Object Request Broker that is used by the eXtreme Scale Java-based client ObjectMap APIs.



REST gateway

- What's new
  - A new REST gateway is implemented for WebSphere eXtreme Scale in 8.6
  - It has the same request/respond specification as the XC10 REST gateway.
- Configure and setup (example configuration XML)
  - For the Liberty profile, in server.xml:
    - Enable REST Gateway

```
<featureManager>
    <feature>eXtremeScale.rest-1.1</feature>
</featureManager>
```

    - Configure REST Gateway

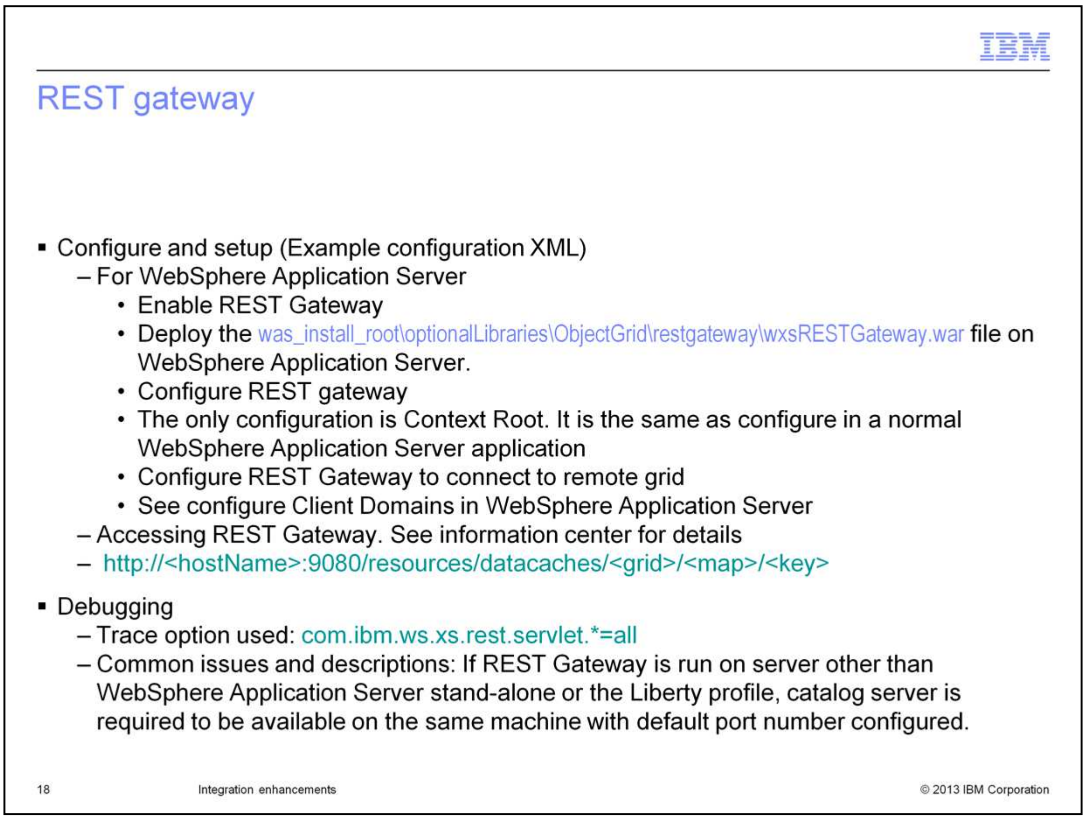
```
<xsREST contextRoot=”myContextRoot”/>
```

    - Configure REST Gateway to connect to remote grid

    See Client Domains section for details

17 Integration enhancements © 2013 IBM Corporation

This slide shows how to enable and configure the REST gateway in the server.xml file for a Liberty profile configuration.



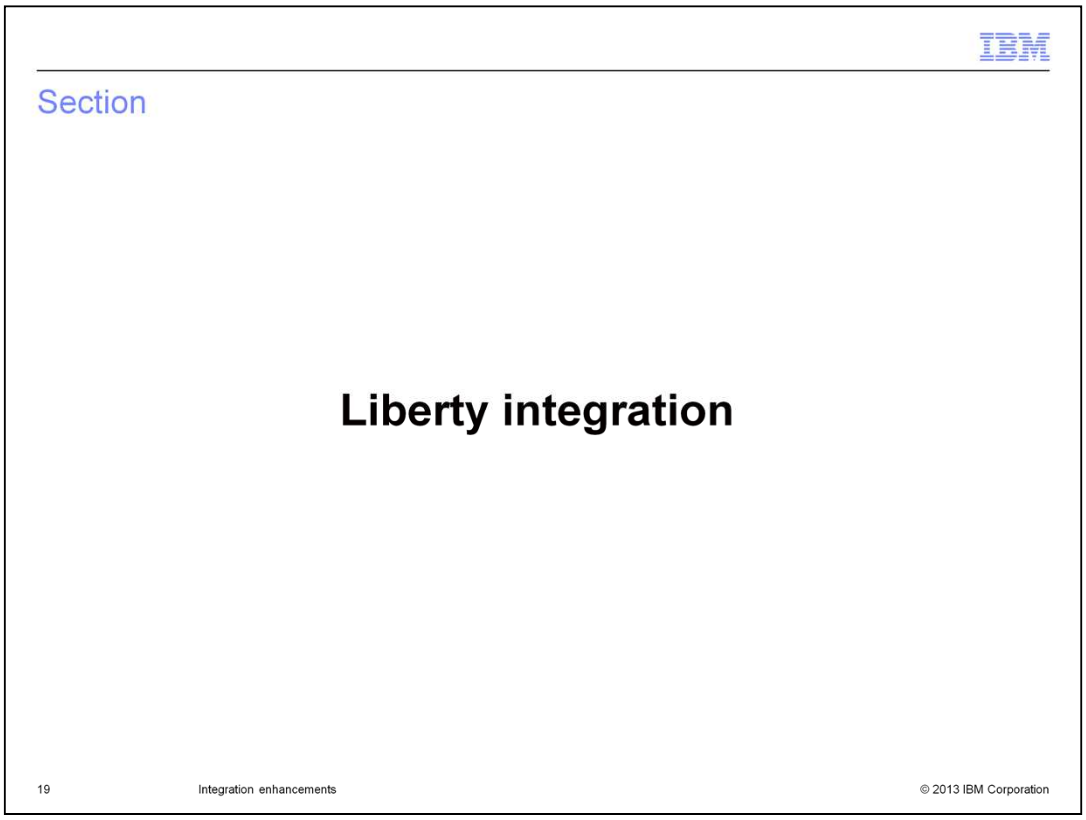
REST gateway

- Configure and setup (Example configuration XML)
  - For WebSphere Application Server
    - Enable REST Gateway
    - Deploy the was_install_root\optionalLibraries\ObjectGrid\restgateway\wxsRESTGateway.war file on WebSphere Application Server.
    - Configure REST gateway
    - The only configuration is Context Root. It is the same as configure in a normal WebSphere Application Server application
    - Configure REST Gateway to connect to remote grid
    - See configure Client Domains in WebSphere Application Server
  - Accessing REST Gateway. See information center for details
  - http://<hostName>:9080/resources/datacaches/<grid>/<map>/<key>
- Debugging
  - Trace option used: com.ibm.ws.xs.rest.servlet.*=all
  - Common issues and descriptions: If REST Gateway is run on server other than WebSphere Application Server stand-alone or the Liberty profile, catalog server is required to be available on the same machine with default port number configured.

18 Integration enhancements © 2013 IBM Corporation

This slide shows how to configure the REST Gateway for a traditional WebSphere Application Server environment.

It also shows the trace string that can be used to debug gateway activities.



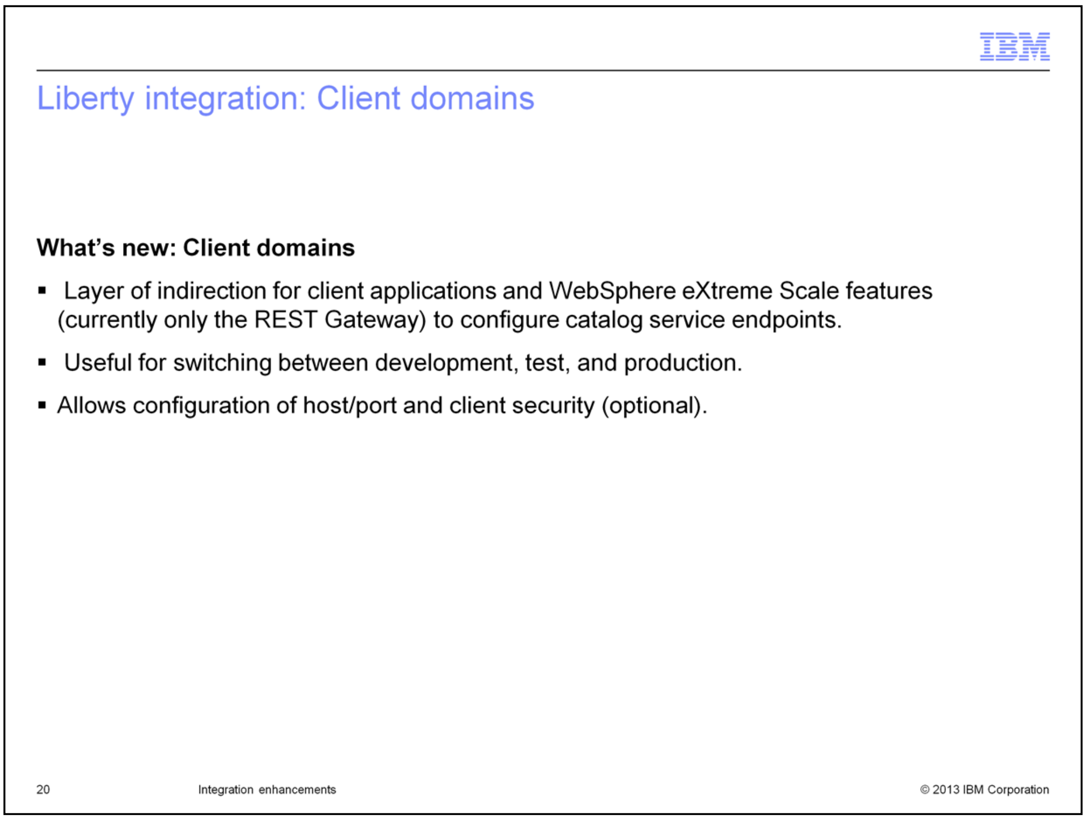
Section

Liberty integration

The Liberty profile is a highly composable, fast-to-start, dynamic application server runtime environment.

You install the Liberty profile when you install WebSphere eXtreme Scale with WebSphere Application Server Version 8.5 along with a JRE provided by either Oracle or IBM.

While not introduced in eXtreme Scale version 8.6, there have been several new capabilities added.

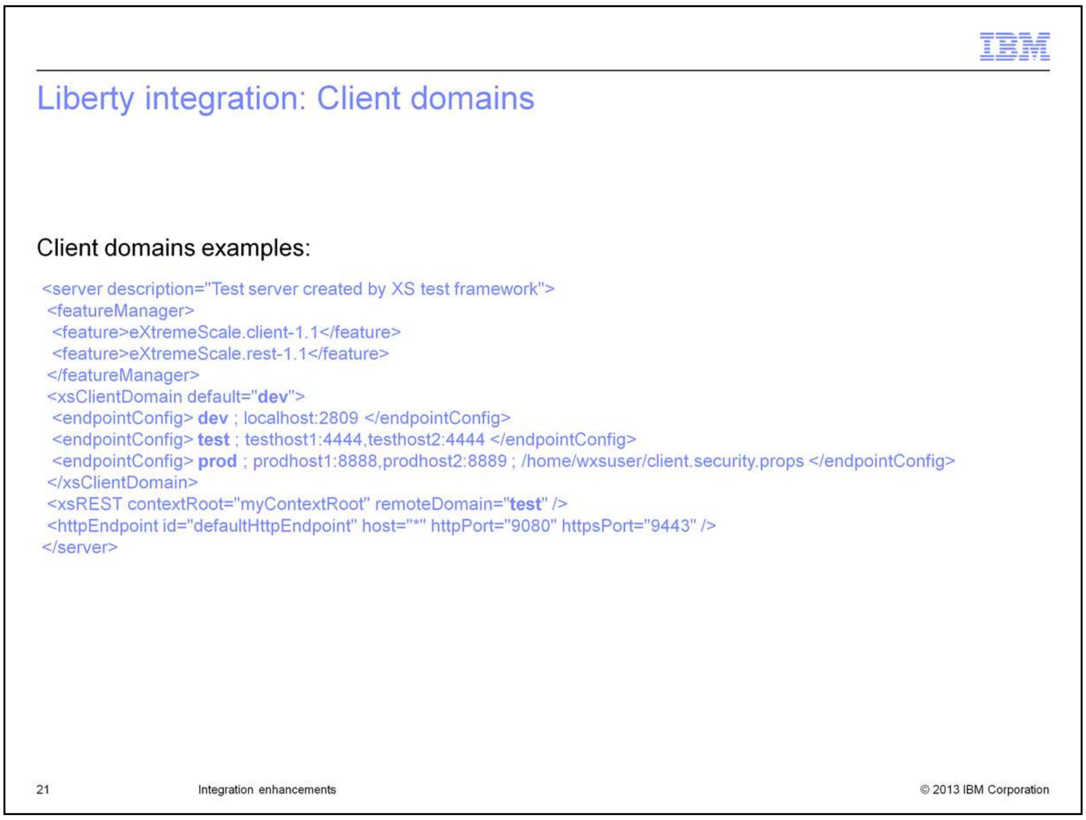


Liberty integration: Client domains

What's new: Client domains

- Layer of indirection for client applications and WebSphere eXtreme Scale features (currently only the REST Gateway) to configure catalog service endpoints.
- Useful for switching between development, test, and production.
- Allows configuration of host/port and client security (optional).

Liberty Client domains are new to eXtreme Scale version 8.6. They provide a layer of indirection for client applications and eXtreme Scale features using the REST Gateway to configure catalog service endpoints. Client domains allow you to switch between development, test, and production environments, and let you configure host, port, and client security settings.



Liberty integration: Client domains

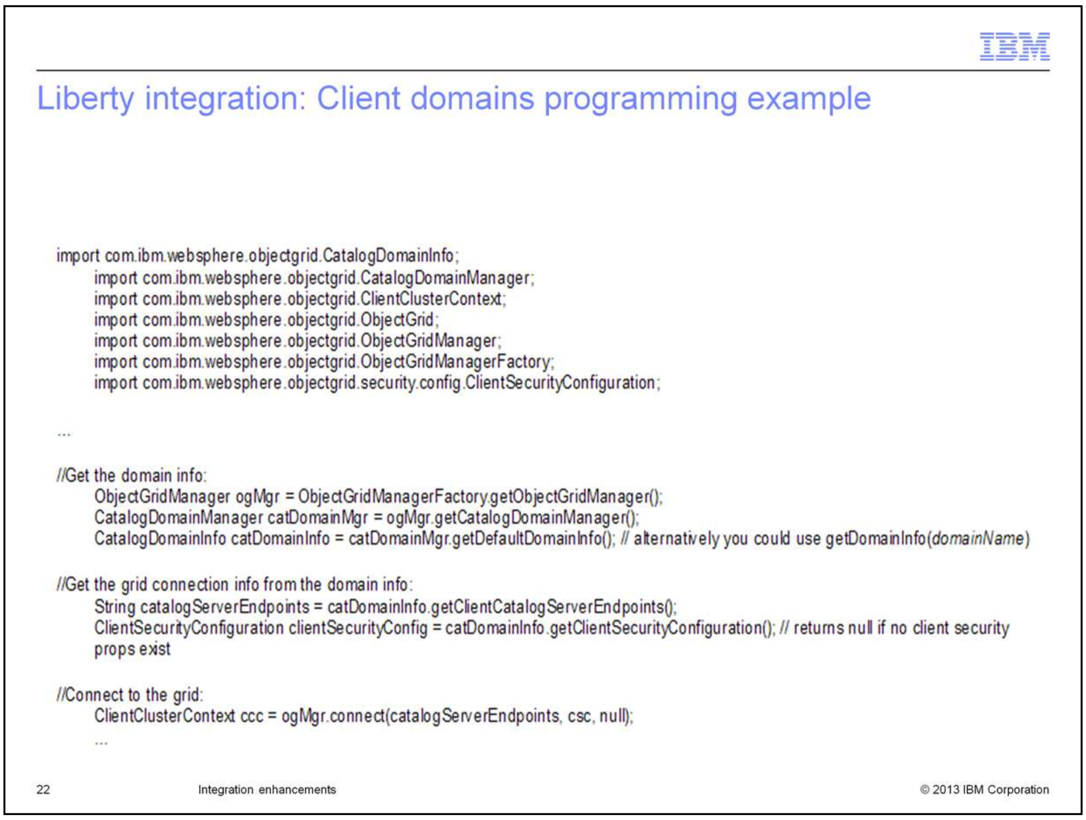
Client domains examples:

```
<server description="Test server created by XS test framework">
 <featureManager>
  <feature>eXtremeScale.client-1.1</feature>
  <feature>eXtremeScale.rest-1.1</feature>
 </featureManager>
 <xsClientDomain default="dev">
  <endpointConfig> dev ; localhost:2809 </endpointConfig>
  <endpointConfig> test ; testhost1:4444,testhost2:4444 </endpointConfig>
  <endpointConfig> prod ; prodhost1:8888,prodhost2:8889 ; /home/wxsuser/client.security.props </endpointConfig>
 </xsClientDomain>
 <xsREST contextRoot="myContextRoot" remoteDomain="test" />
 <httpEndpoint id="defaultHttpEndpoint" host="*" httpPort="9080" httpsPort="9443" />
</server>
```

21 Integration enhancements © 2013 IBM Corporation

This example shows a configuration with three different client domains. The “dev” domain is the default, and points to a catalog server listening on port 2809 on the local host. The “test” domain points to a catalog cluster that is listening on port 4444 on hosts testhost1 and testhost2. The “prod” domain points to a catalog cluster hosted on prodhost1 and prodhost2, listening on ports 8888 and 8889 – note that the production domain also contains a client security properties file indicating that grid security is enabled for the production grid, but not for the development and test grids.

The other thing to notice with this example is that xsREST element, used for configuring the REST Gateway, specifies a remoteDomain of “test”. This means that the REST gateway web application will point to the test domain grid. You can change this setting on the fly to use a different domain, and then the REST gateway web application points at the new domain.

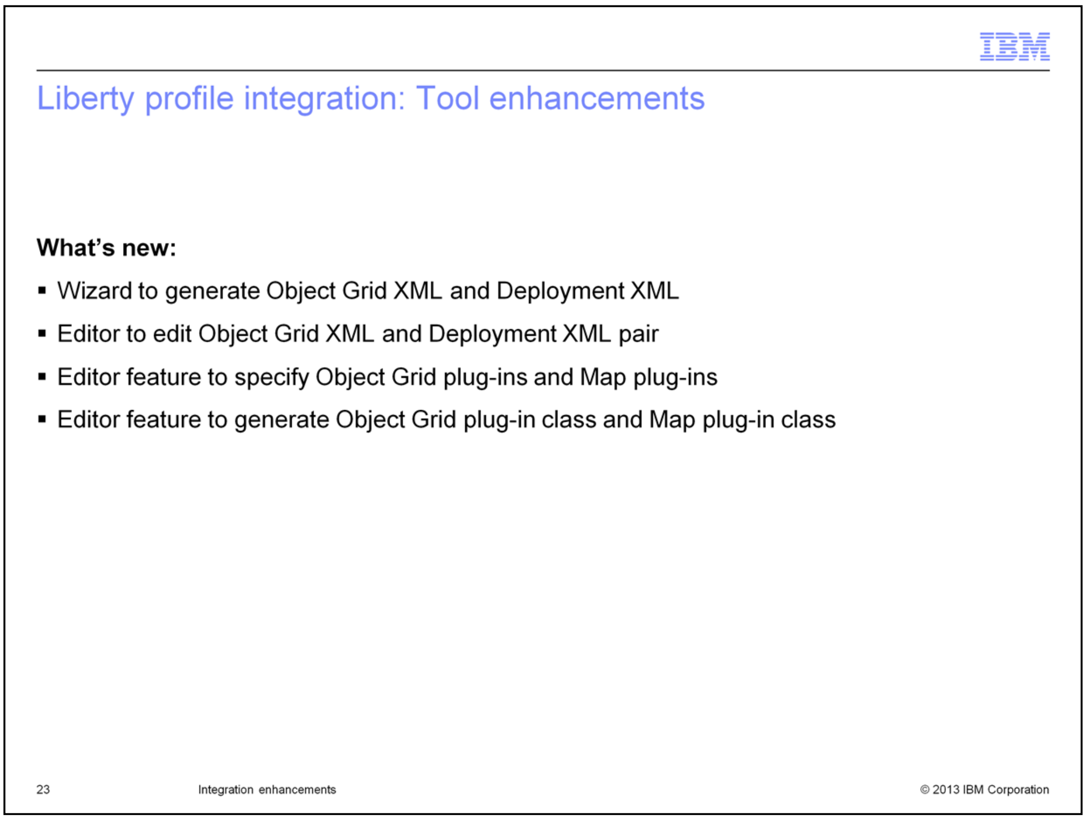


Liberty integration: Client domains programming example

```
import com.ibm.websphere.objectgrid.CatalogDomainInfo;
        import com.ibm.websphere.objectgrid.CatalogDomainManager;
        import com.ibm.websphere.objectgrid.ClientClusterContext;
        import com.ibm.websphere.objectgrid.ObjectGrid;
        import com.ibm.websphere.objectgrid.ObjectGridManager;
        import com.ibm.websphere.objectgrid.ObjectGridManagerFactory;
        import com.ibm.websphere.objectgrid.security.config.ClientSecurityConfiguration;

...

//Get the domain info:
        ObjectGridManager ogMgr = ObjectGridManagerFactory.getObjectGridManager();
        CatalogDomainManager catDomainMgr = ogMgr.getCatalogDomainManager();
        CatalogDomainInfo catDomainInfo = catDomainMgr.getDefaultDomainInfo(); // alternatively you could use getDomainInfo(domainName)

//Get the grid connection info from the domain info:
        String catalogServerEndpoints = catDomainInfo.getClientCatalogServerEndpoints();
        ClientSecurityConfiguration clientSecurityConfig = catDomainInfo.getClientSecurityConfiguration(); // returns null if no client security
        props exist

//Connect to the grid:
        ClientClusterContext ccc = ogMgr.connect(catalogServerEndpoints, csc, null);
        ...
```

22 Integration enhancements © 2013 IBM Corporation

This example shows how to programmatically get the domain information, get the grid connection information from the domain information, and how to connect to the grid.



Liberty profile integration: Tool enhancements

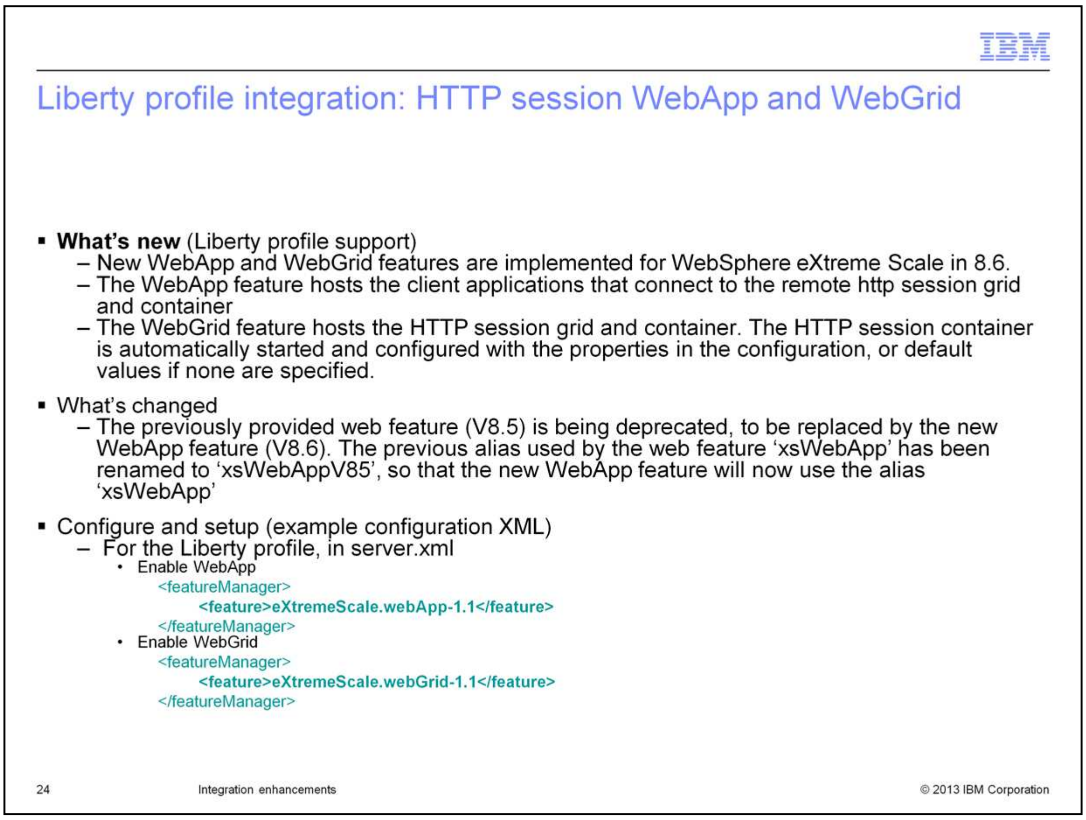
What's new:

- Wizard to generate Object Grid XML and Deployment XML
- Editor to edit Object Grid XML and Deployment XML pair
- Editor feature to specify Object Grid plug-ins and Map plug-ins
- Editor feature to generate Object Grid plug-in class and Map plug-in class

23 Integration enhancements © 2013 IBM Corporation

In Version 8.6 there have been several tool enhancements for the Liberty profile.

A new wizard has been added to generate ObjectGrid XML and Deployment XML files, an editor to edit these files, an editor to specify ObjectGrid and Map plug-ins, and an editor to generate ObjectGrid plug-in and Map plug-in classes.



Liberty profile integration: HTTP session WebApp and WebGrid

- **What's new** (Liberty profile support)
  - New WebApp and WebGrid features are implemented for WebSphere eXtreme Scale in 8.6.
  - The WebApp feature hosts the client applications that connect to the remote http session grid and container
  - The WebGrid feature hosts the HTTP session grid and container. The HTTP session container is automatically started and configured with the properties in the configuration, or default values if none are specified.
- What's changed
  - The previously provided web feature (V8.5) is being deprecated, to be replaced by the new WebApp feature (V8.6). The previous alias used by the web feature 'xsWebApp' has been renamed to 'xsWebAppV85', so that the new WebApp feature will now use the alias 'xsWebApp'
- Configure and setup (example configuration XML)
  - For the Liberty profile, in server.xml
    - Enable WebApp

```
<featureManager>
        <feature>eXtremeScale.webApp-1.1</feature>
</featureManager>
```

    - Enable WebGrid

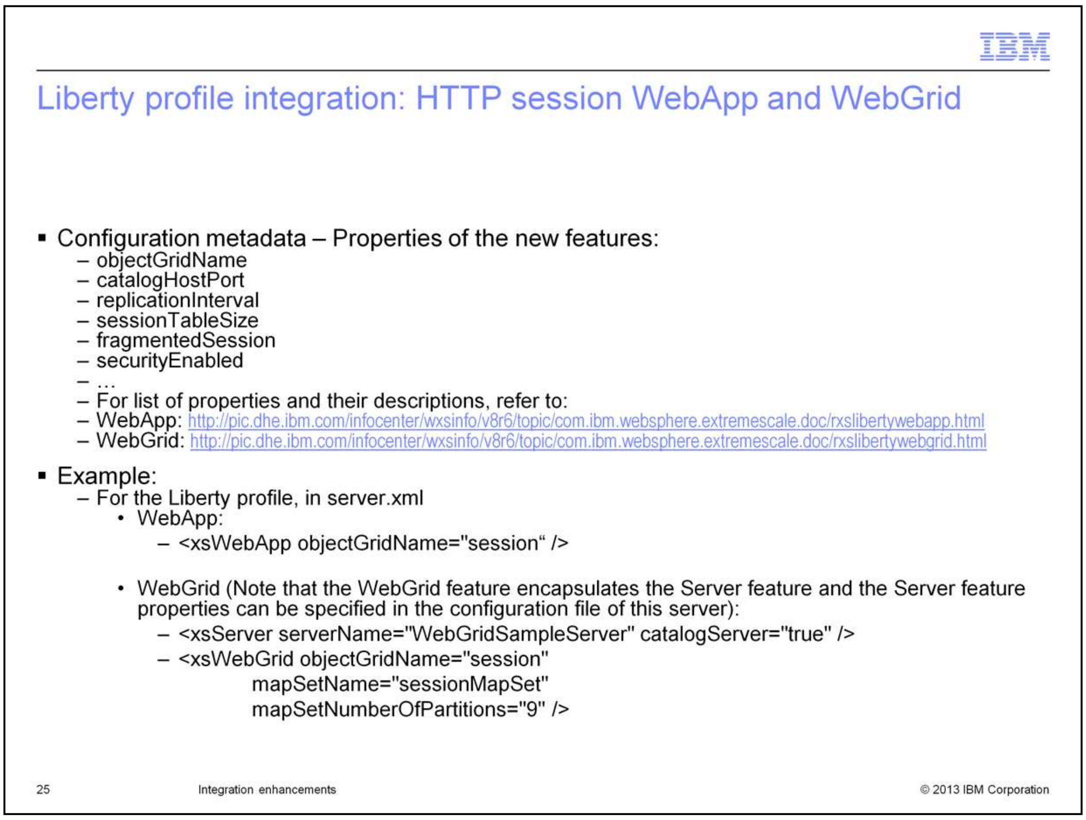
```
<featureManager>
        <feature>eXtremeScale.webGrid-1.1</feature>
</featureManager>
```

24 Integration enhancements © 2013 IBM Corporation

New WebApp and WebGrid features have also been added in eXtreme Scale version 8.6. The WebApp feature hosts client applications that connect to the remote grid and the WebGrid feature hosts the http session grid and containers.

The previous web feature supported in eXtreme Scale version 8.5 has been deprecated and renamed "xsWebAppV85" so that users using the "xsWebApp" alias will automatically use the new feature.

Finally, you can see how to configure the WebApp and WebGrid features with the example xml at the bottom of the slide.



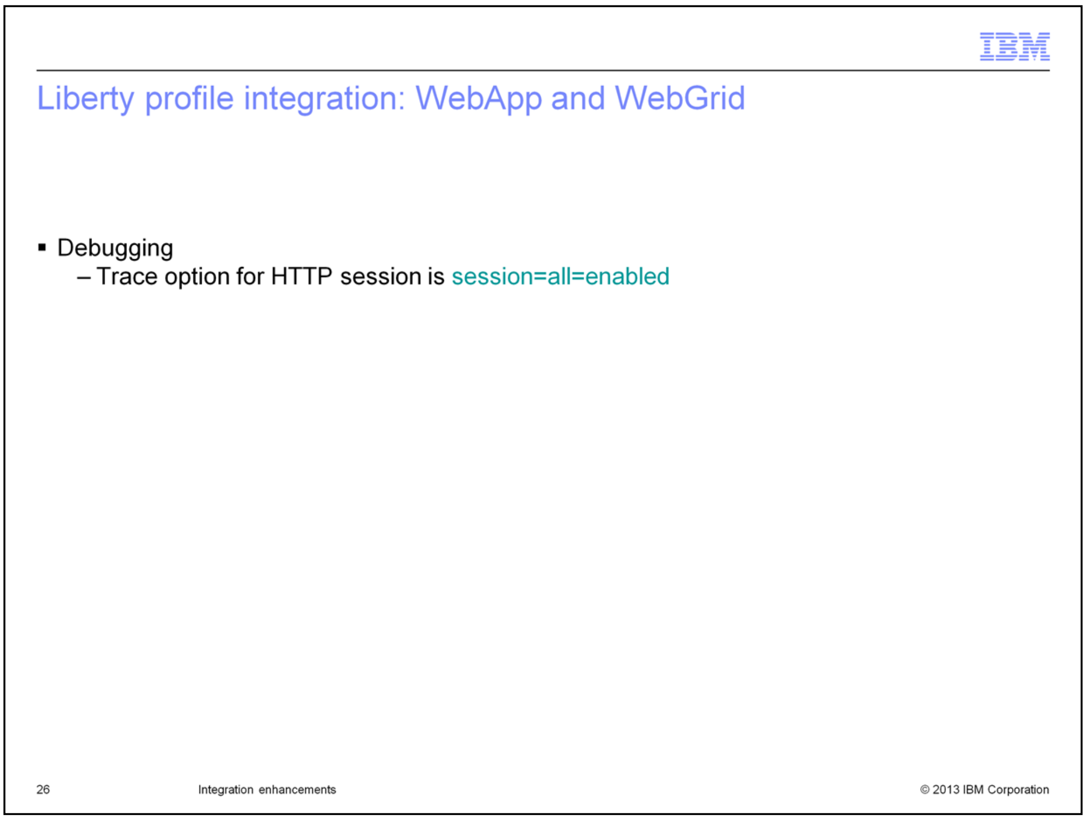
IBM

Liberty profile integration: HTTP session WebApp and WebGrid

- Configuration metadata – Properties of the new features:
  - objectGridName
  - catalogHostPort
  - replicationInterval
  - sessionTableSize
  - fragmentedSession
  - securityEnabled
  - …
  - For list of properties and their descriptions, refer to:
  - WebApp: http://pic.dhe.ibm.com/infocenter/wxsinfo/v8r6/topic/com.ibm.websphere.extremescale.doc/rxslibertywebapp.html
  - WebGrid: http://pic.dhe.ibm.com/infocenter/wxsinfo/v8r6/topic/com.ibm.websphere.extremescale.doc/rxslibertywebgrid.html
- Example:
  - For the Liberty profile, in server.xml
    - WebApp:
      - <xsWebApp objectGridName="session“ />
    - WebGrid (Note that the WebGrid feature encapsulates the Server feature and the Server feature properties can be specified in the configuration file of this server):
      - <xsServer serverName="WebGridSampleServer" catalogServer="true" />
      - <xsWebGrid objectGridName="session"
        mapSetName="sessionMapSet"
        mapSetNumberOfPartitions="9" />

25 Integration enhancements © 2013 IBM Corporation

This slide shows the configuration options for the WebApp and WebGrid features and how to define them within the server.xml file.

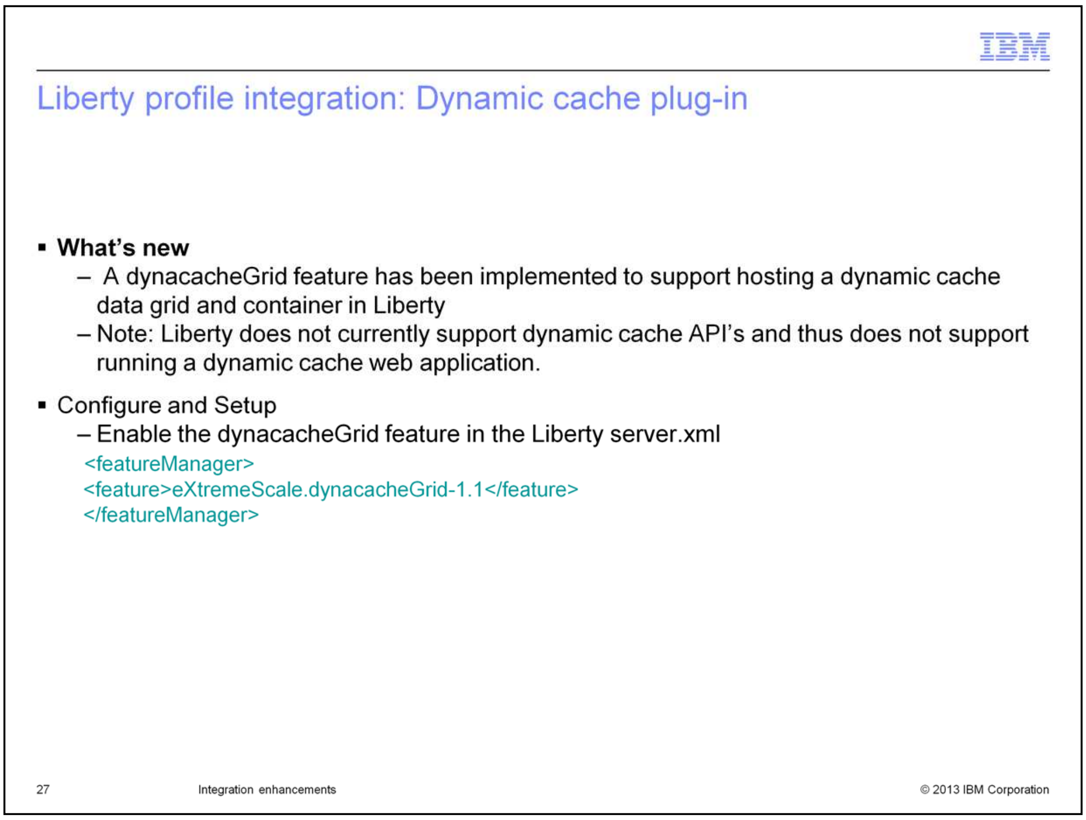


Liberty profile integration: WebApp and WebGrid

- Debugging
  - Trace option for HTTP session is session=all=enabled

26 Integration enhancements © 2013 IBM Corporation

Listed on this slide is the trace string that can be enabled for debugging the WebApp or WebGrid features.



Liberty profile integration: Dynamic cache plug-in

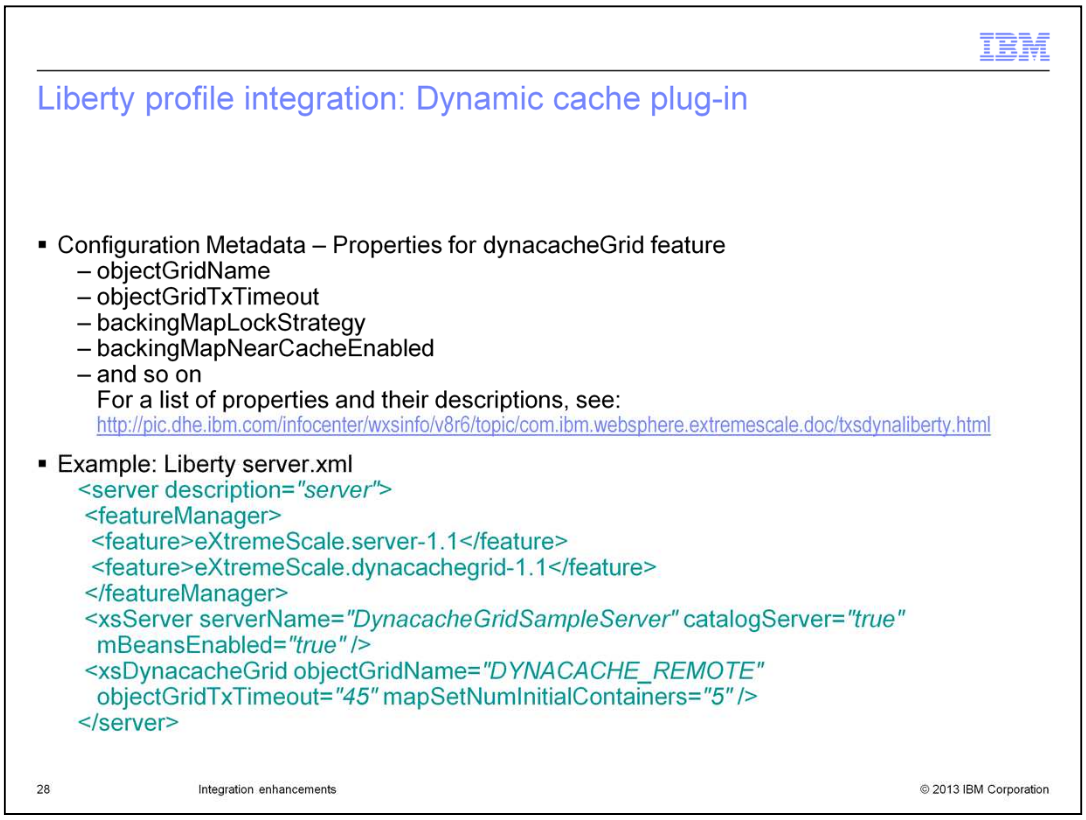
- **What's new**
  - A dynacacheGrid feature has been implemented to support hosting a dynamic cache data grid and container in Liberty
  - Note: Liberty does not currently support dynamic cache API's and thus does not support running a dynamic cache web application.
- Configure and Setup
  - Enable the dynacacheGrid feature in the Liberty server.xml

```
<featureManager>
<feature>eXtremeScale.dynacacheGrid-1.1</feature>
</featureManager>
```

27 Integration enhancements © 2013 IBM Corporation

In Version 8.6, a dynamic cache grid feature has been added to support hosting a dynamic cache data grid in the Liberty profile. This support is only for stand-alone data grids and does not include support of dynamic cache clients using the dynamic cache APIs within a Liberty profile.

The bottom of the slide demonstrates how to change the Liberty profile's server.xml file to define a dynamic cache grid.



Liberty profile integration: Dynamic cache plug-in

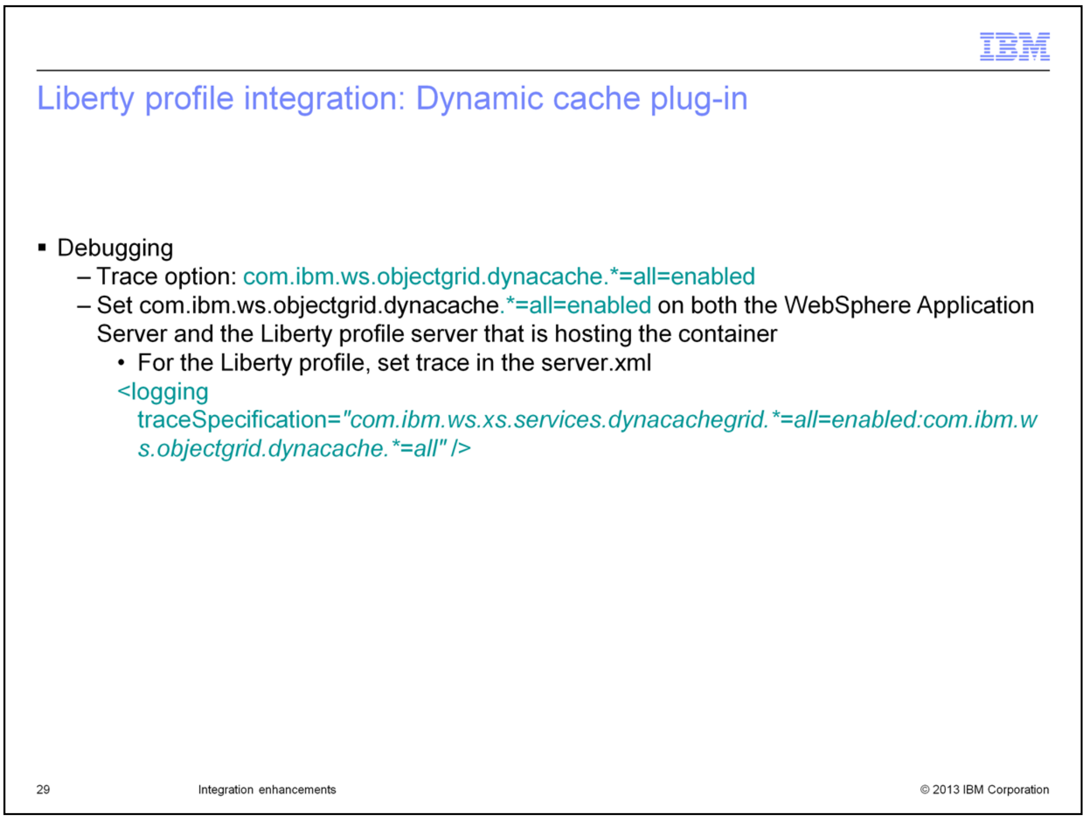
- Configuration Metadata – Properties for dynacacheGrid feature
  - objectGridName
  - objectGridTxTimeout
  - backingMapLockStrategy
  - backingMapNearCacheEnabled
  - and so on

  For a list of properties and their descriptions, see:
  http://pic.dhe.ibm.com/infocenter/wxsinfo/v8r6/topic/com.ibm.websphere.extremescale.doc/txsdynaliberty.html
- Example: Liberty server.xml

```
<server description="server">
 <featureManager>
  <feature>eXtremeScale.server-1.1</feature>
  <feature>eXtremeScale.dynacachegrid-1.1</feature>
 </featureManager>
 <xsServer serverName="DynacacheGridSampleServer" catalogServer="true"
   mBeansEnabled="true" />
 <xsDynacacheGrid objectGridName="DYNACACHE_REMOTE"
   objectGridTxTimeout="45" mapSetNumInitialContainers="5" />
</server>
```

28 Integration enhancements © 2013 IBM Corporation

This slide shows the configuration options for the dynamic cache plug-in feature and how to define it within the server.xml file.



Liberty profile integration: Dynamic cache plug-in

- Debugging
  - Trace option: com.ibm.ws.objectgrid.dynacache.*=all=enabled
  - Set com.ibm.ws.objectgrid.dynacache.*=all=enabled on both the WebSphere Application Server and the Liberty profile server that is hosting the container
    - For the Liberty profile, set trace in the server.xml

```
<logging
  traceSpecification="com.ibm.ws.xs.services.dynacachegrid.*=all=enabled:com.ibm.w
  s.objectgrid.dynacache.*=all" />
```

29 Integration enhancements © 2013 IBM Corporation

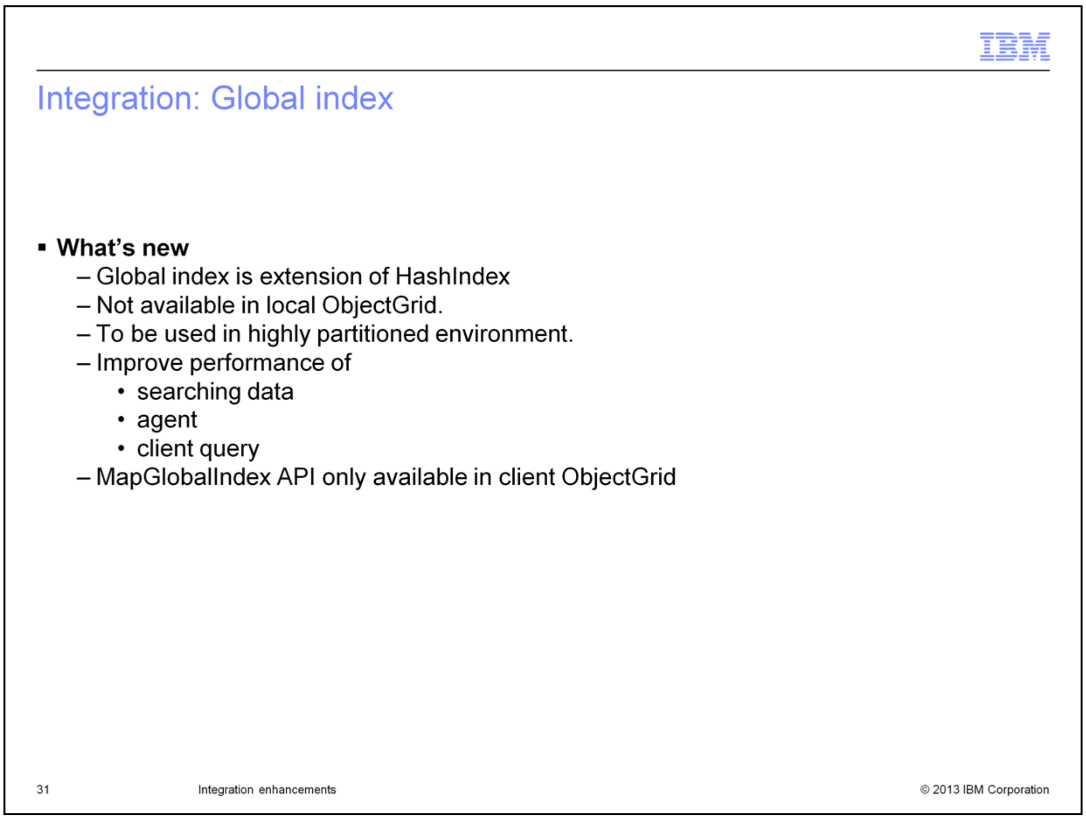
Here you see how to turn on trace for a dynamic cache plug-in in the Liberty profile.

Section

Global index

30 Integration enhancements © 2013 IBM Corporation

Global Index functionality is new in version 8.6.



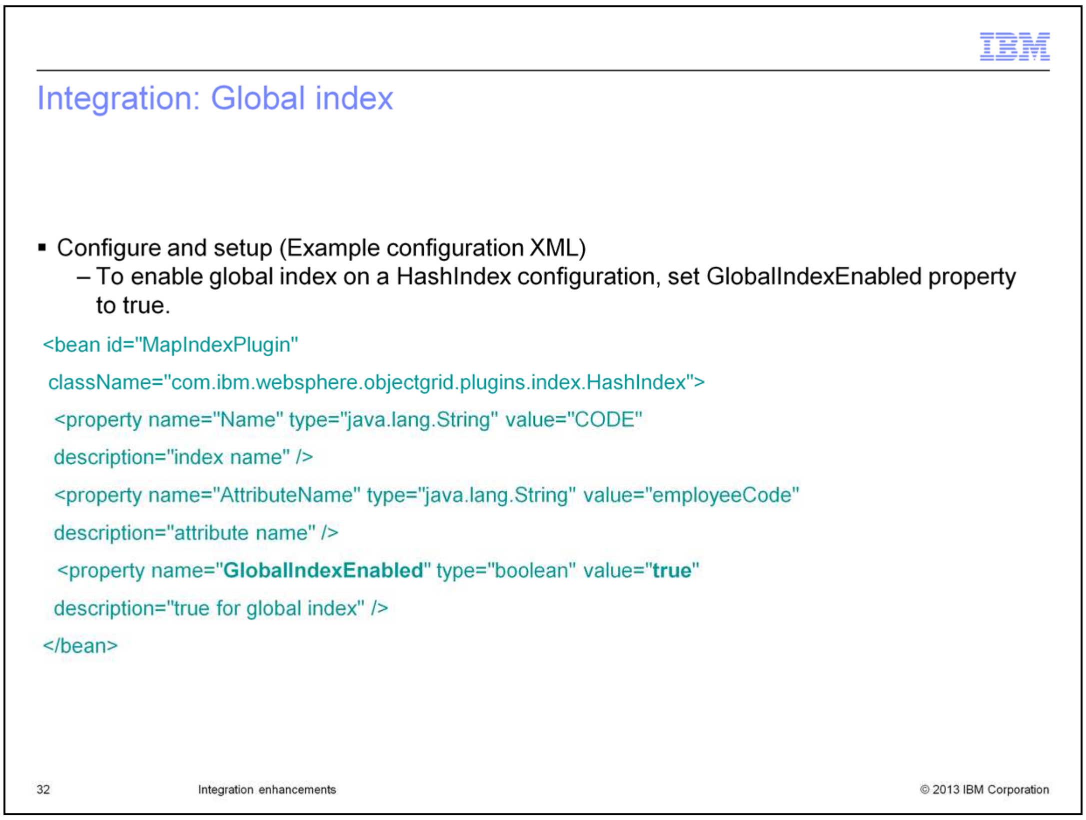
Integration: Global index

- **What's new**
  - Global index is extension of HashIndex
  - Not available in local ObjectGrid.
  - To be used in highly partitioned environment.
  - Improve performance of
    - searching data
    - agent
    - client query
  - MapGlobalIndex API only available in client ObjectGrid

31 Integration enhancements © 2013 IBM Corporation

In an eXtreme Scale grid configuration with multiple partitions, cached objects are spread into all of the partitions. To have complete results, regular indexes, queries, and agents must run on all partitions. This can be expensive. Ideally, these operations should only run on applicable partitions, and therefore, eliminate unnecessary overhead. The global index feature can track the location of indexed attributes and can determine applicable partitions for those attributes. Typically, the applicable partitions found are only a subset of all partitions.

This will improve performance running indexes, queries, and agents since the requests will only be sent to a subset of partitions within the data grid.

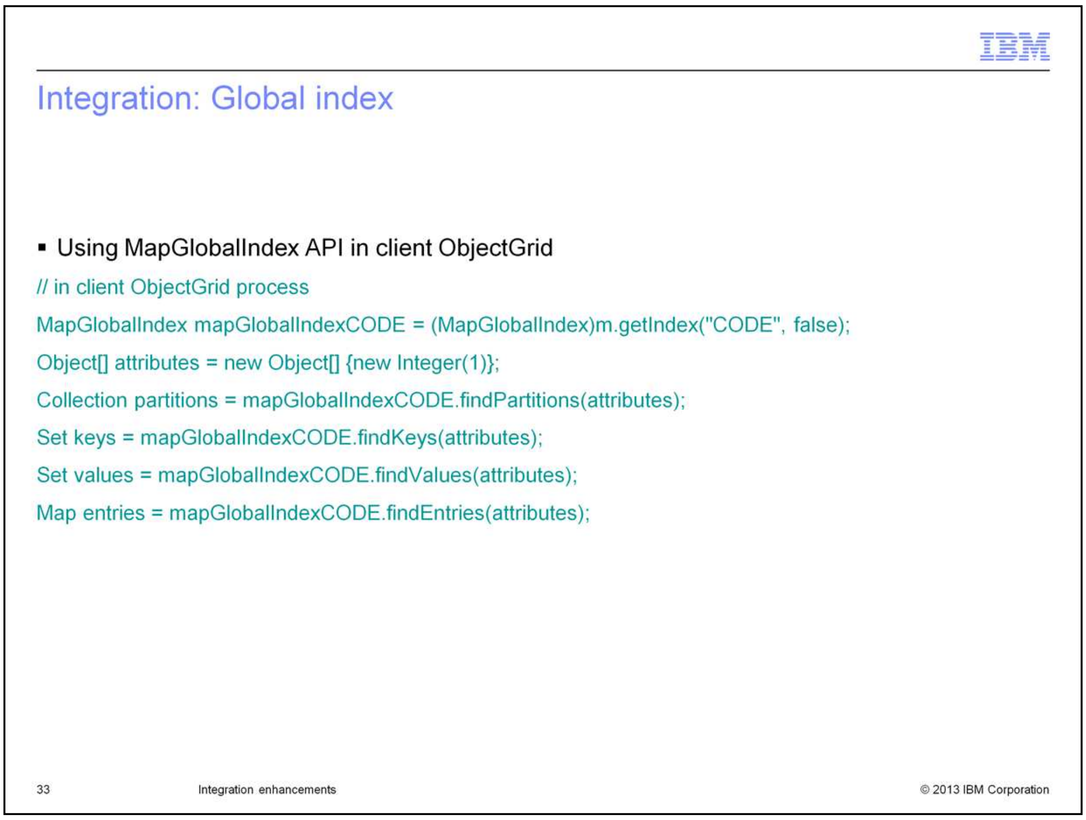

Integration: Global index

- Configure and setup (Example configuration XML)
  - To enable global index on a HashIndex configuration, set GlobalIndexEnabled property to true.

```
<bean id="MapIndexPlugin"
 className="com.ibm.websphere.objectgrid.plugins.index.HashIndex">
  <property name="Name" type="java.lang.String" value="CODE"
  description="index name" />
  <property name="AttributeName" type="java.lang.String" value="employeeCode"
  description="attribute name" />
   <property name="GlobalIndexEnabled" type="boolean" value="true"
  description="true for global index" />
</bean>
```

32 Integration enhancements © 2013 IBM Corporation

Here is how a Global Index is defined within a HashIndex definition within the objectgrid.xml. In this example, a Global Index is generated for the employeeCode attribute in the HashIndex. The global index can then be queried to determine which partitions have entries for a particular employeeCode and only send requests to those partitions.



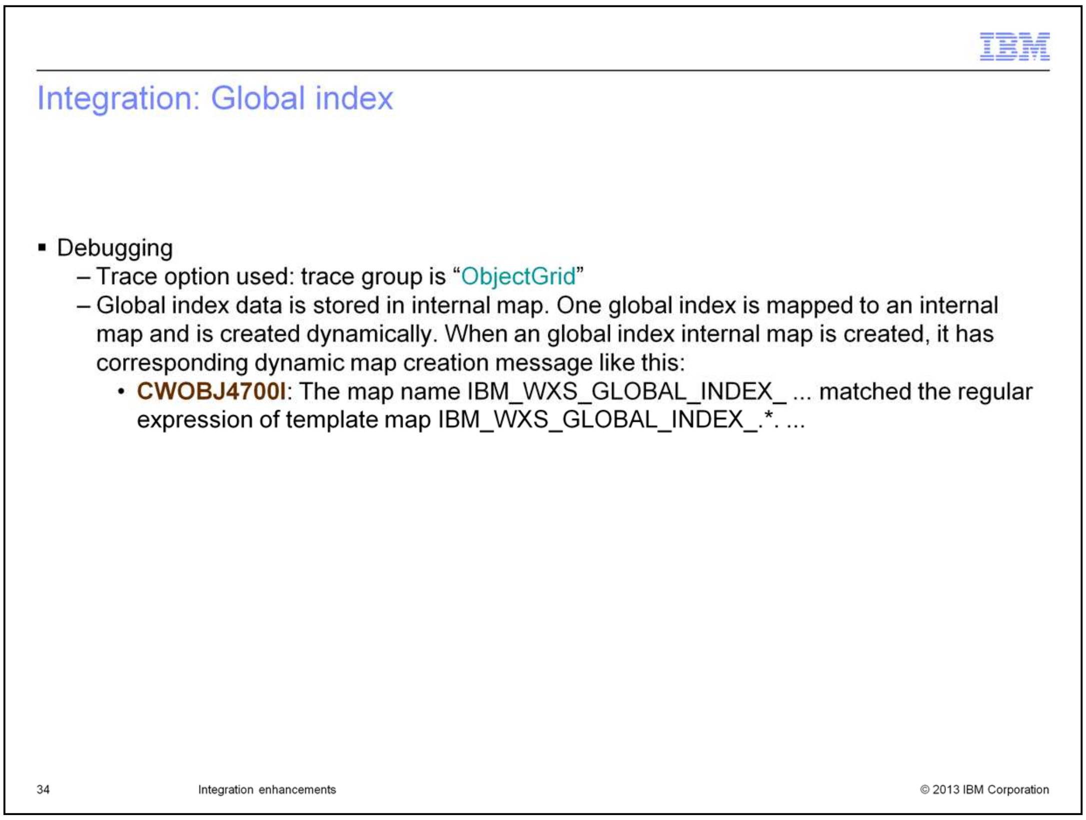
Integration: Global index

- Using MapGlobalIndex API in client ObjectGrid

```
// in client ObjectGrid process
MapGlobalIndex mapGlobalIndexCODE = (MapGlobalIndex)m.getIndex("CODE", false);
Object[] attributes = new Object[] {new Integer(1)};
Collection partitions = mapGlobalIndexCODE.findPartitions(attributes);
Set keys = mapGlobalIndexCODE.findKeys(attributes);
Set values = mapGlobalIndexCODE.findValues(attributes);
Map entries = mapGlobalIndexCODE.findEntries(attributes);
```

33 Integration enhancements © 2013 IBM Corporation

This slide shows how an application will programmatically use a Global Index to find the partitions, keys, values, or entries. See the WebSphere eXtreme Scale information center for more details.

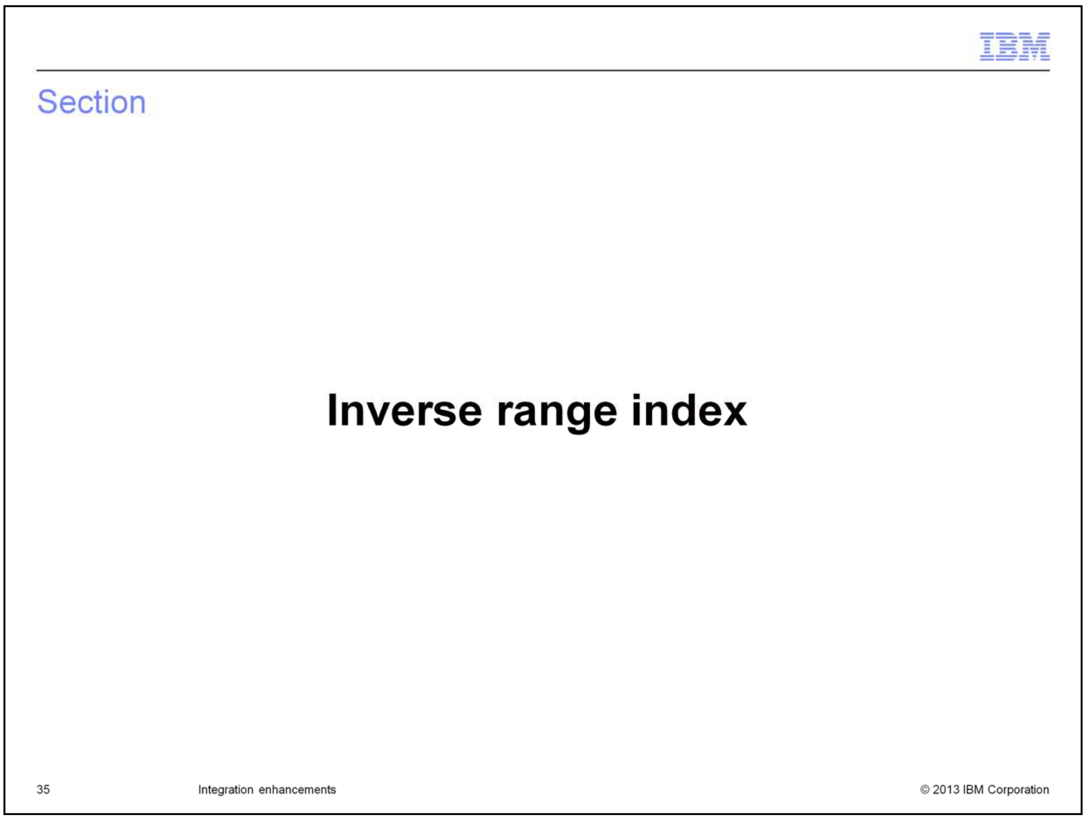


Integration: Global index

- Debugging
  - Trace option used: trace group is "ObjectGrid"
  - Global index data is stored in internal map. One global index is mapped to an internal map and is created dynamically. When an global index internal map is created, it has corresponding dynamic map creation message like this:
    - **CWOBJ4700I**: The map name IBM_WXS_GLOBAL_INDEX_ ... matched the regular expression of template map IBM_WXS_GLOBAL_INDEX_.*. ...

34 Integration enhancements © 2013 IBM Corporation

This slide shows the trace that can be used to debug global index issues, along with the informational message to look for in the SystemOut.log file to ensure the global index is configured properly.

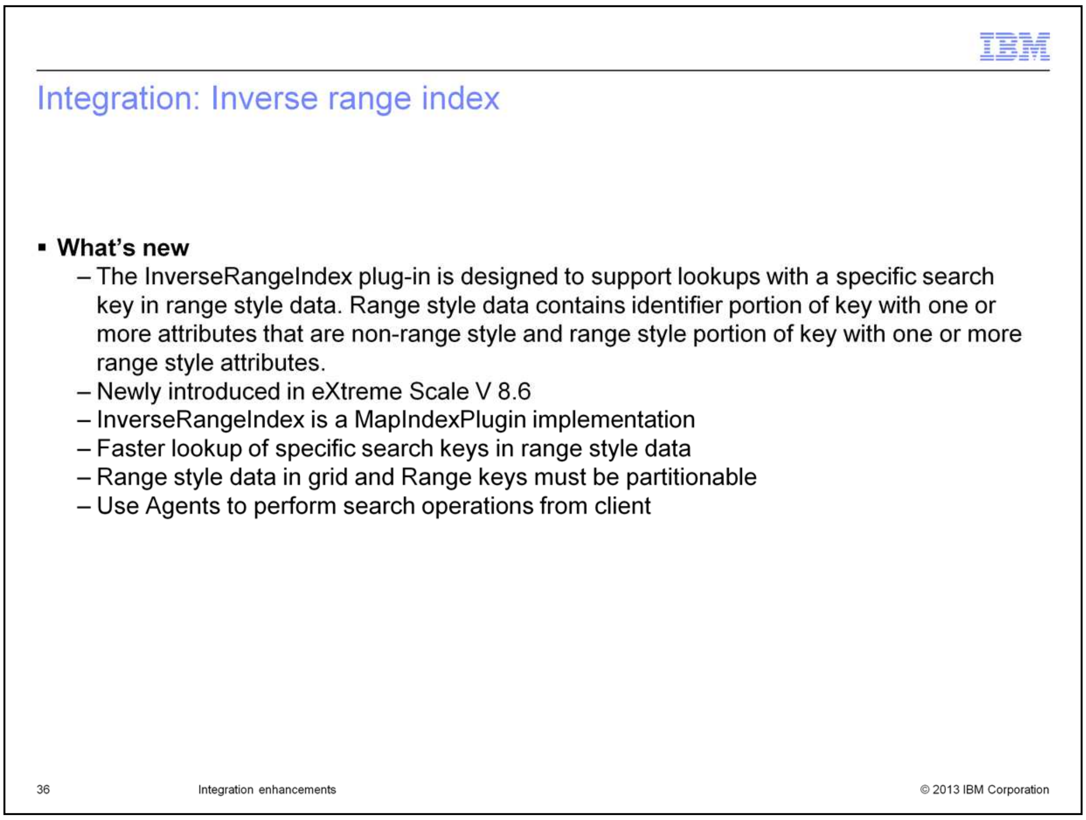


Section

Inverse range index

35 Integration enhancements © 2013 IBM Corporation

The Inverse Range Index is another index-related feature added in version 8.6.



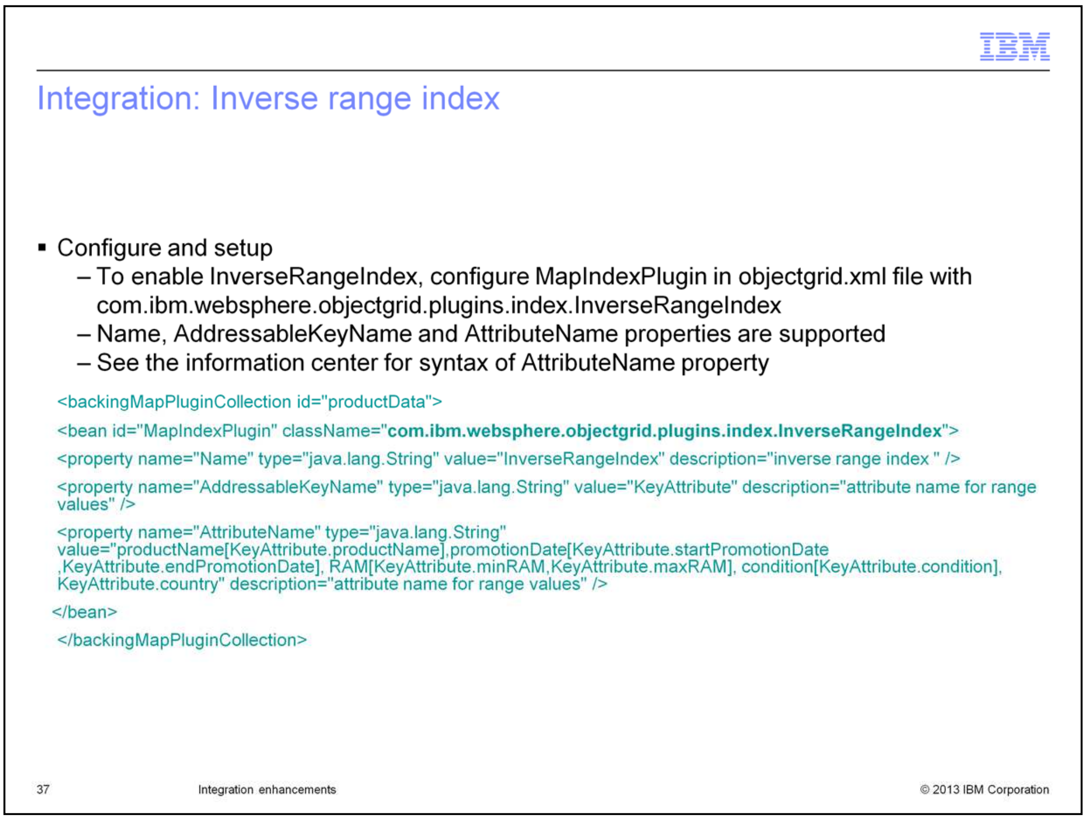
Integration: Inverse range index

- **What's new**
  - The InverseRangeIndex plug-in is designed to support lookups with a specific search key in range style data. Range style data contains identifier portion of key with one or more attributes that are non-range style and range style portion of key with one or more range style attributes.
  - Newly introduced in eXtreme Scale V 8.6
  - InverseRangeIndex is a MapIndexPlugin implementation
  - Faster lookup of specific search keys in range style data
  - Range style data in grid and Range keys must be partitionable
  - Use Agents to perform search operations from client

36 Integration enhancements © 2013 IBM Corporation

The InverseRangeIndex plug-in is designed to let you look up entries in an eXtreme Scale data grid using a search key that will fall into a specified range of data. For example, retrieving all entries that have a month field with values that fall between January and March.

All Range style data and keys stored in the eXtreme Scale data grid must be partitionable, and this data must be accessed through agents so that it runs on the WebSphere eXtreme Scale servers.



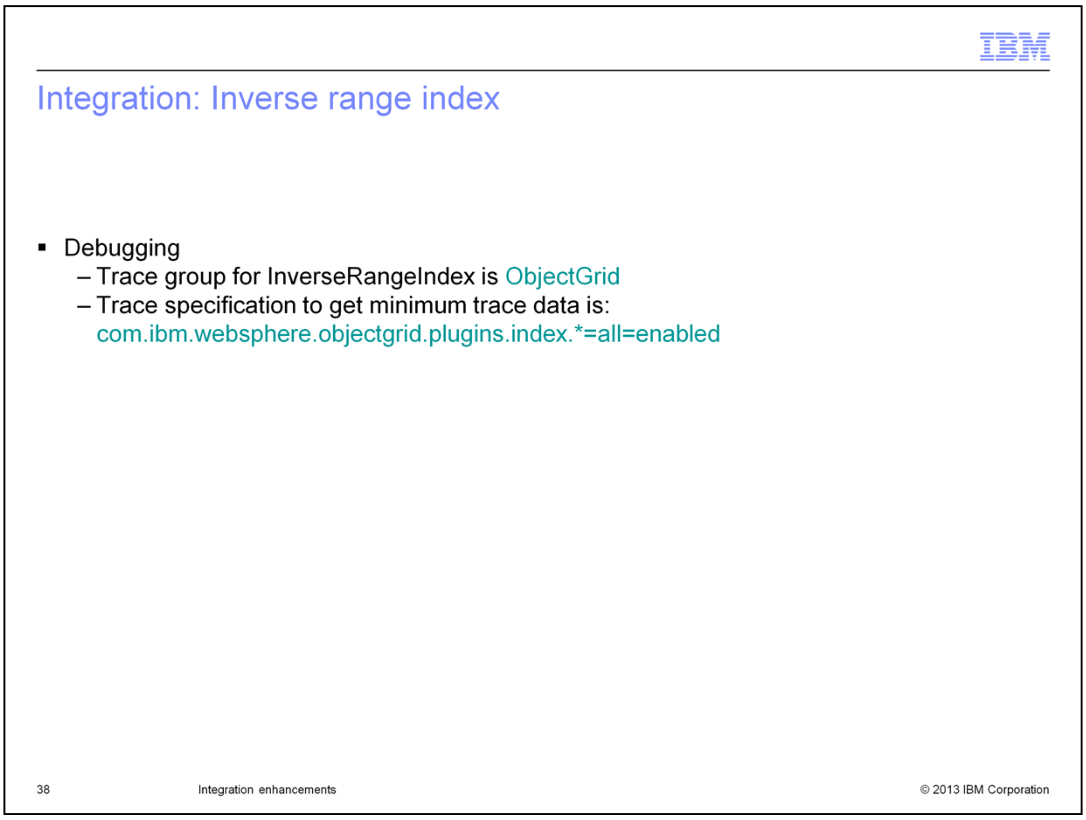
Integration: Inverse range index

- Configure and setup
  - To enable InverseRangeIndex, configure MapIndexPlugin in objectgrid.xml file with com.ibm.websphere.objectgrid.plugins.index.InverseRangeIndex
  - Name, AddressableKeyName and AttributeName properties are supported
  - See the information center for syntax of AttributeName property

```
<backingMapPluginCollection id="productData">
<bean id="MapIndexPlugin" className="com.ibm.websphere.objectgrid.plugins.index.InverseRangeIndex">
<property name="Name" type="java.lang.String" value="InverseRangeIndex" description="inverse range index " />
<property name="AddressableKeyName" type="java.lang.String" value="KeyAttribute" description="attribute name for range
values" />
<property name="AttributeName" type="java.lang.String"
value="productName[KeyAttribute.productName],promotionDate[KeyAttribute.startPromotionDate
,KeyAttribute.endPromotionDate], RAM[KeyAttribute.minRAM,KeyAttribute.maxRAM], condition[KeyAttribute.condition],
KeyAttribute.country" description="attribute name for range values" />
</bean>
</backingMapPluginCollection>
```

37 Integration enhancements © 2013 IBM Corporation

This slide shows how to configure an Inverse Range Index within the objectgrid xml file associated with an eXtreme scale data grid. For more information see the eXtreme Scale information center.



Integration: Inverse range index

- Debugging
  - Trace group for InverseRangeIndex is ObjectGrid
  - Trace specification to get minimum trace data is: com.ibm.websphere.objectgrid.plugins.index.*=all=enabled

38 Integration enhancements © 2013 IBM Corporation

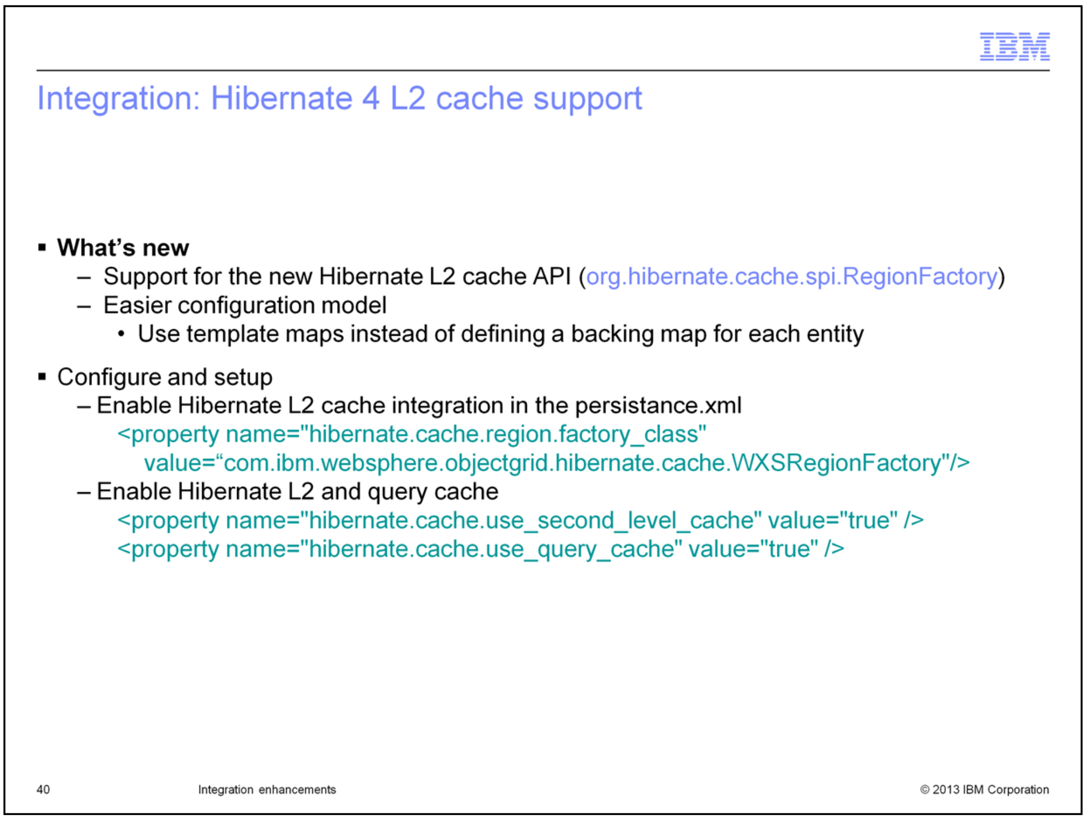
ObjectGrid is the trace group used for the inverse range index. This slide also shows the minimum trace to use, since an ObjectGrid trace is quite large.

Section

Hibernate enhancements

39 Integration enhancements © 2013 IBM Corporation

Hibernate also has enhancements in version 8.6



Integration: Hibernate 4 L2 cache support

- **What’s new**
  - Support for the new Hibernate L2 cache API (org.hibernate.cache.spi.RegionFactory)
  - Easier configuration model
    - Use template maps instead of defining a backing map for each entity
- Configure and setup
  - Enable Hibernate L2 cache integration in the persistance.xml

```
<property name="hibernate.cache.region.factory_class"
   value="com.ibm.websphere.objectgrid.hibernate.cache.WXSRegionFactory"/>
```

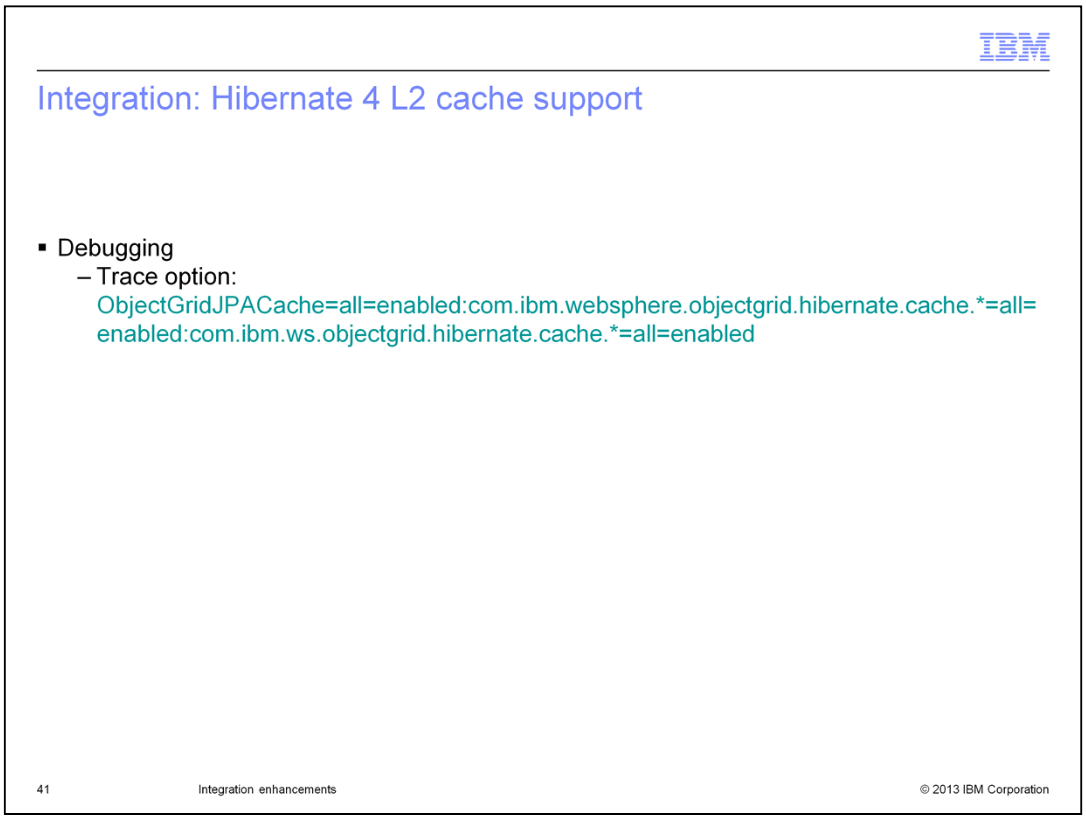
  - Enable Hibernate L2 and query cache

```
<property name="hibernate.cache.use_second_level_cache" value="true" />
<property name="hibernate.cache.use_query_cache" value="true" />
```

40 Integration enhancements © 2013 IBM Corporation

Hibernate support is not new to eXtreme Scale version 8.6, but it has been upgraded to support Hibernate version 4, which has an easier configuration model that allows for using template maps instead of defining a backing map for each entity.

This slide also shows how to configure the hibernate L2 cache in the persistence.xml file.

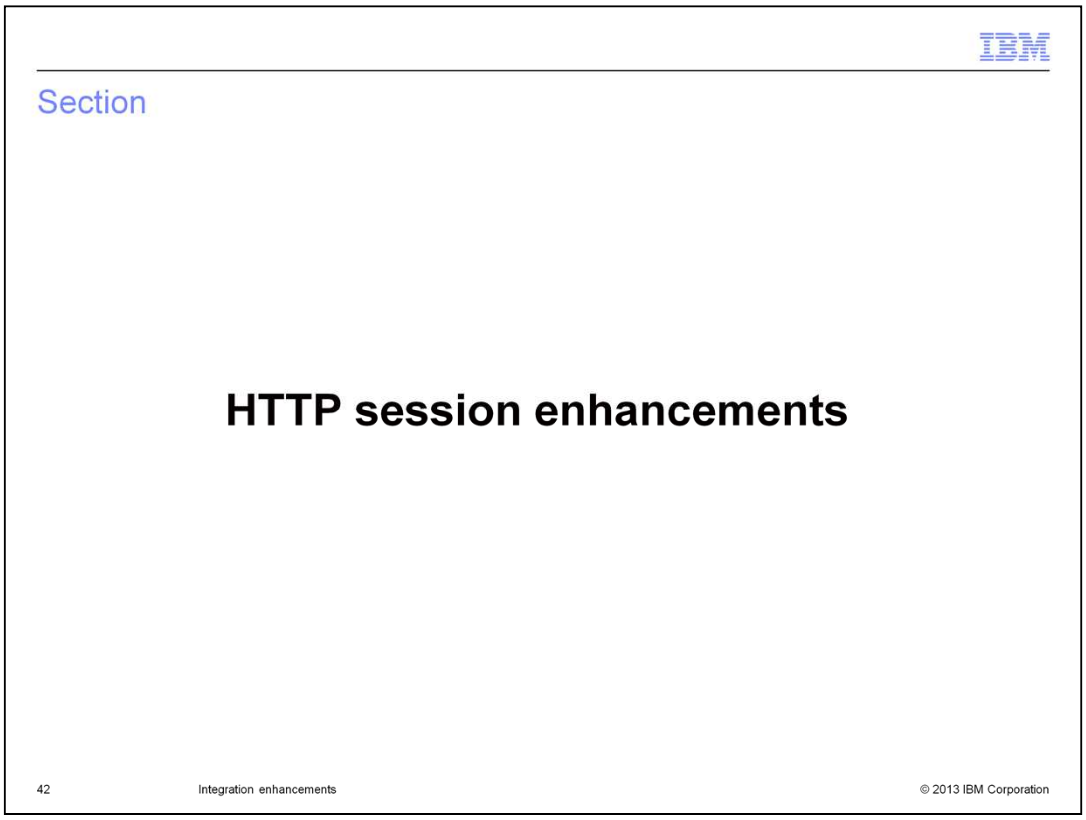


Integration: Hibernate 4 L2 cache support

- Debugging
  - Trace option: ObjectGridJPACache=all=enabled:com.ibm.websphere.objectgrid.hibernate.cache.*=all=enabled:com.ibm.ws.objectgrid.hibernate.cache.*=all=enabled

41 Integration enhancements © 2013 IBM Corporation

Here is how to configure trace for debugging hibernate related issues in WebSphere eXtreme Scale.

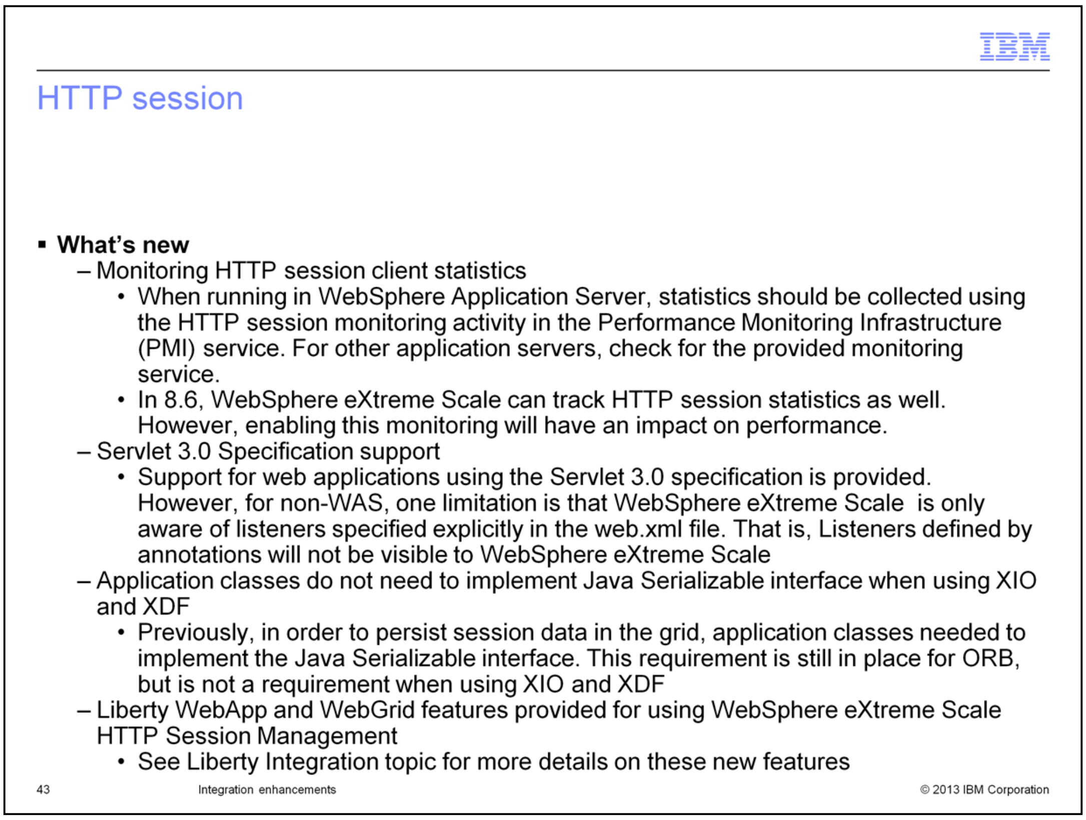


Section

HTTP session enhancements

42 Integration enhancements © 2013 IBM Corporation

The next topic is the version 8.6 enhancements for the eXtreme Scale HTTP Session plug-in.


HTTP session

- **What's new**
  - Monitoring HTTP session client statistics
    - When running in WebSphere Application Server, statistics should be collected using the HTTP session monitoring activity in the Performance Monitoring Infrastructure (PMI) service. For other application servers, check for the provided monitoring service.
    - In 8.6, WebSphere eXtreme Scale can track HTTP session statistics as well. However, enabling this monitoring will have an impact on performance.
  - Servlet 3.0 Specification support
    - Support for web applications using the Servlet 3.0 specification is provided. However, for non-WAS, one limitation is that WebSphere eXtreme Scale is only aware of listeners specified explicitly in the web.xml file. That is, Listeners defined by annotations will not be visible to WebSphere eXtreme Scale
  - Application classes do not need to implement Java Serializable interface when using XIO and XDF
    - Previously, in order to persist session data in the grid, application classes needed to implement the Java Serializable interface. This requirement is still in place for ORB, but is not a requirement when using XIO and XDF
  - Liberty WebApp and WebGrid features provided for using WebSphere eXtreme Scale HTTP Session Management
    - See Liberty Integration topic for more details on these new features

43 Integration enhancements © 2013 IBM Corporation

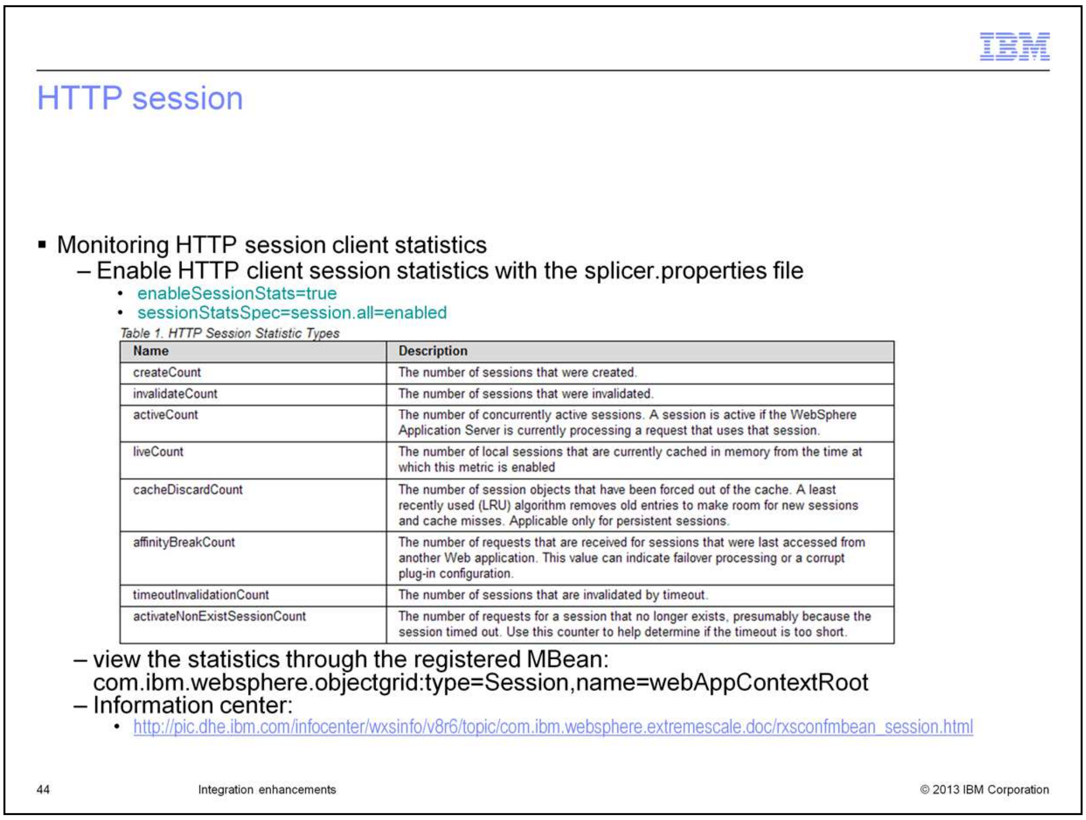
Listed here are the new feature and functions available for eXtreme Scale HTTP Session users.

Historically, HTTP session statistics have been available in the application server hosting this work. For the WebSphere Application Server, this has been the Performance Monitoring Infrastructure service.

With version 8.6, WebSphere eXtreme Scale also supports tracking of session statistics using a registered MBean.

Also, in version 8.6, HTTP Session users can now use Web applications coded to the Servlet 3.0 Specification. Non-WebSphere Application Server users should be aware that eXtreme Scale will only be aware of listeners specified in the web.xml file and will ignore listeners defined in annotations.

In addition, with the advent of the eXtreme data format, data persisted to a grid from the HTTP Session no longer needs to implement the Java Serializable interface, and Liberty profile WebApp and WebGrid features are provided as described earlier.



HTTP session

- Monitoring HTTP session client statistics
  - Enable HTTP client session statistics with the splicer.properties file
    - enableSessionStats=true
    - sessionStatsSpec=session.all=enabled

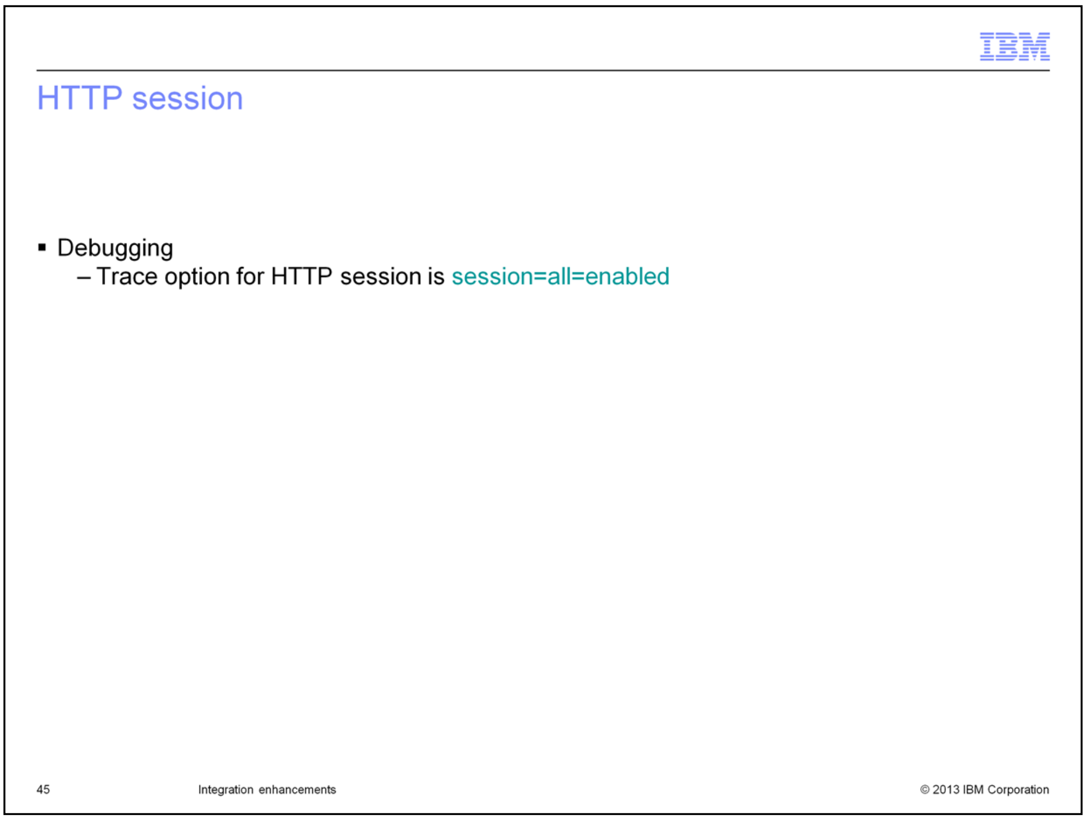
Table 1. HTTP Session Statistic Types

Name	Description
createCount	The number of sessions that were created.
invalidateCount	The number of sessions that were invalidated.
activeCount	The number of concurrently active sessions. A session is active if the WebSphere Application Server is currently processing a request that uses that session.
liveCount	The number of local sessions that are currently cached in memory from the time at which this metric is enabled
cacheDiscardCount	The number of session objects that have been forced out of the cache. A least recently used (LRU) algorithm removes old entries to make room for new sessions and cache misses. Applicable only for persistent sessions.
affinityBreakCount	The number of requests that are received for sessions that were last accessed from another Web application. This value can indicate failover processing or a corrupt plug-in configuration.
timeoutInvalidationCount	The number of sessions that are invalidated by timeout.
activateNonExistSessionCount	The number of requests for a session that no longer exists, presumably because the session timed out. Use this counter to help determine if the timeout is too short.

  - view the statistics through the registered MBean: com.ibm.websphere.objectgrid:type=Session,name=webAppContextRoot
  - Information center:
    - http://pic.dhe.ibm.com/infocenter/wxsinfo/v8r6/topic/com.ibm.websphere.extremescale.doc/rxsconfmbean_session.html

44 Integration enhancements © 2013 IBM Corporation

This slide shows how to enable HTTP client statistics within the splicer.properties file and how to view these statistics through the registered MBean.

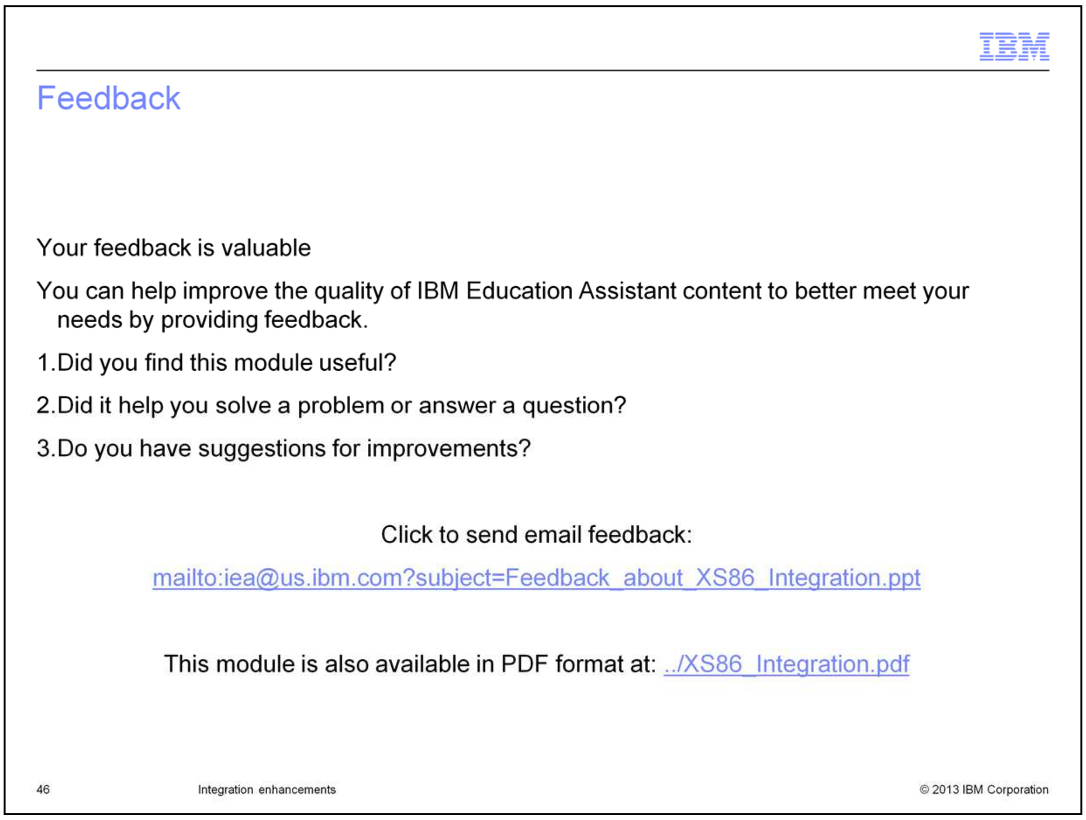


HTTP session

- Debugging
  - Trace option for HTTP session is session=all=enabled

45 Integration enhancements © 2013 IBM Corporation

This slide shows the trace string that can be set to allow for tracing of the eXtreme Scale HTTP Session function. This trace can be set on both the eXtreme Scale client and server, but it is primarily only set on the client.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

1. Did you find this module useful?
2. Did it help you solve a problem or answer a question?
3. Do you have suggestions for improvements?

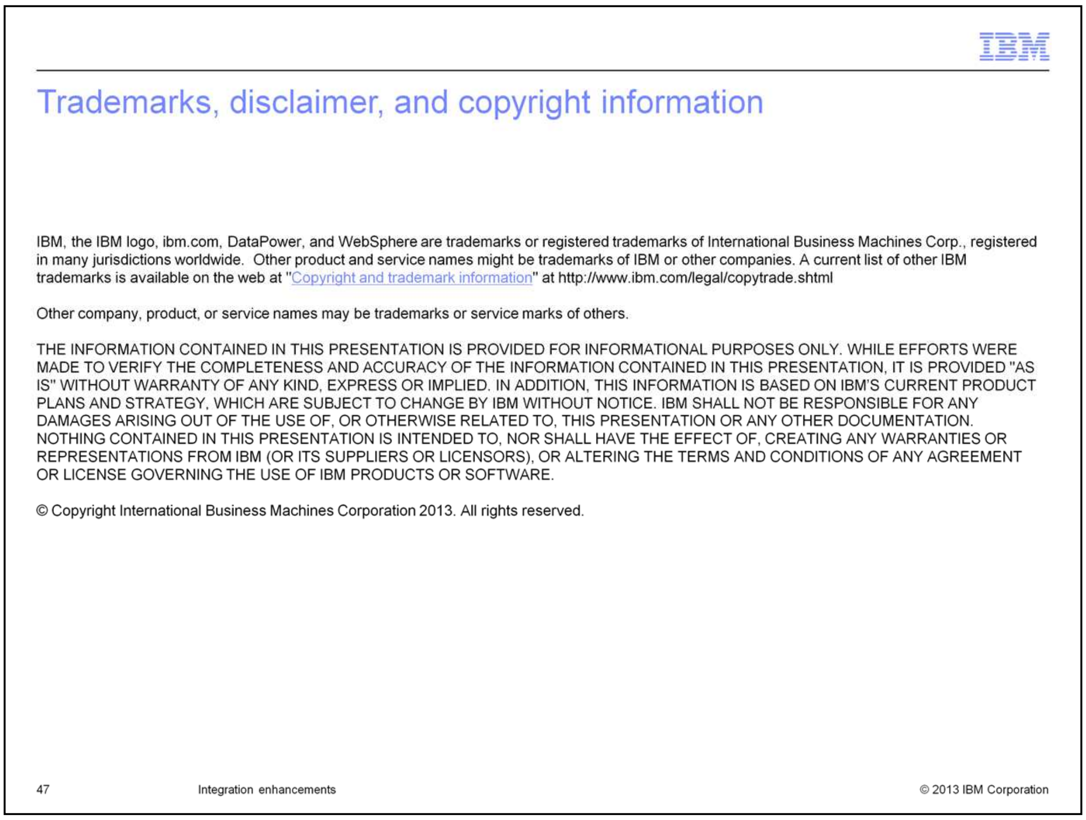
Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_XS86_Integration.ppt

This module is also available in PDF format at: ../XS86_Integration.pdf

46 Integration enhancements © 2013 IBM Corporation

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information