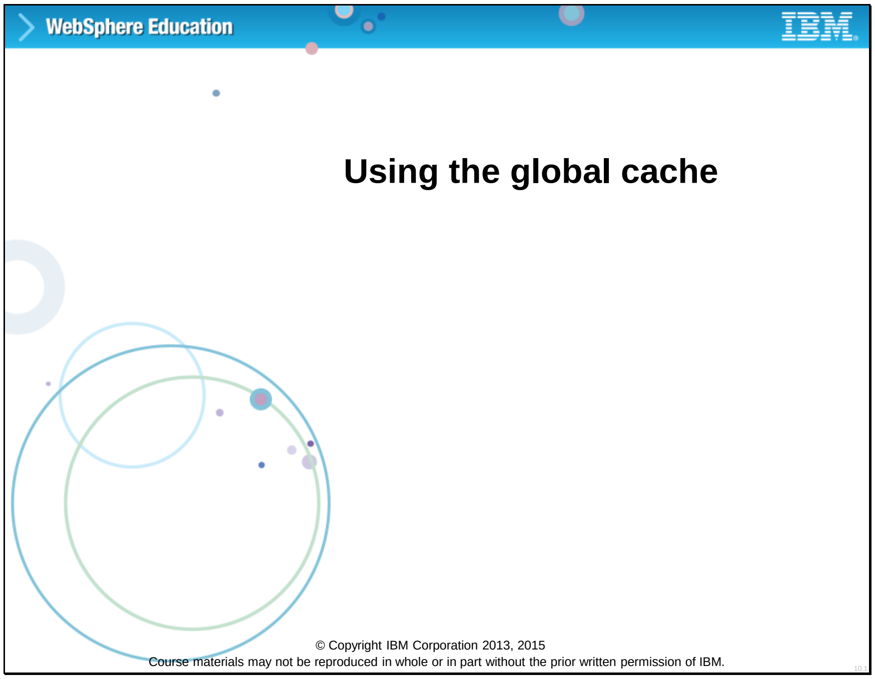
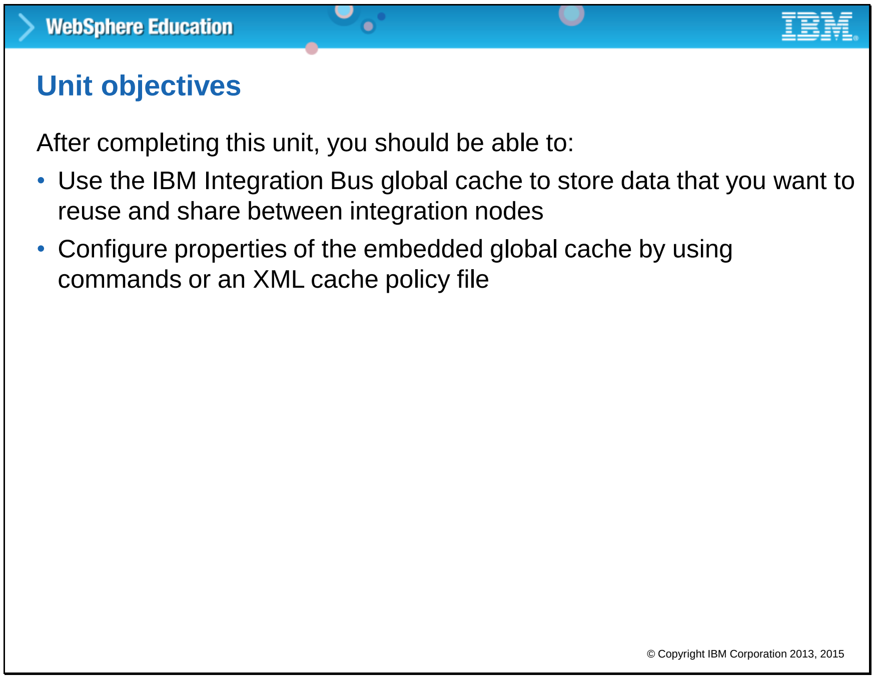
Slide 1



Using the global cache

Slide 2



WebSphere Education

Unit objectives

After completing this unit, you should be able to:

- Use the IBM Integration Bus global cache to store data that you want to reuse and share between integration nodes
- Configure properties of the embedded global cache by using commands or an XML cache policy file

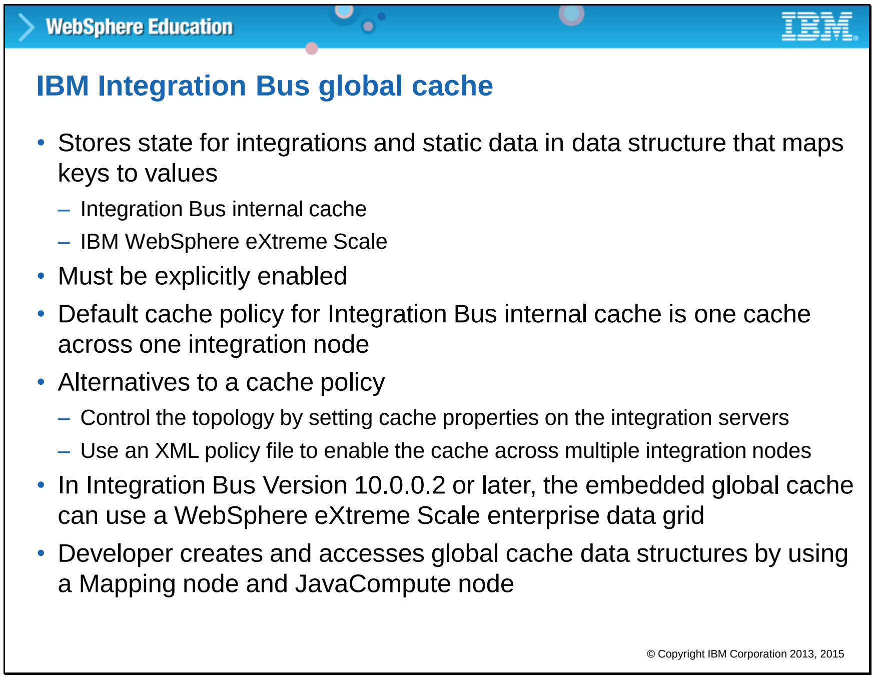
Unit objectives

A global cache is a repository for data that you want to reuse across processes (both in the same integration node, and across integration nodes). IBM Integration Bus supports an embedded global cache or and you can configure IBM Integration Bus to connect to an external WebSphere eXtreme Scale grid. In this unit, you learn how to configure and use the IBM Integration Bus embedded global cache to store the data that you want to reuse.

After completing this unit, you should be able to:

- Use the IBM Integration Bus global cache to store data that you want to reuse and share between integration nodes
- Configure properties of the embedded global cache by using commands or an XML cache policy file

Slide 3



WebSphere Education

IBM Integration Bus global cache

- Stores state for integrations and static data in data structure that maps keys to values
  - Integration Bus internal cache
  - IBM WebSphere eXtreme Scale
- Must be explicitly enabled
- Default cache policy for Integration Bus internal cache is one cache across one integration node
- Alternatives to a cache policy
  - Control the topology by setting cache properties on the integration servers
  - Use an XML policy file to enable the cache across multiple integration nodes
- In Integration Bus Version 10.0.0.2 or later, the embedded global cache can use a WebSphere eXtreme Scale enterprise data grid
- Developer creates and accesses global cache data structures by using a Mapping node and JavaCompute node

IBM Integration Bus global cache

You can use the global cache that is embedded in an integration node to store data that you want to reuse and share between message flows and integration nodes.

As an option, you can use one or more external WebSphere eXtreme Scale grids to store data that you want to reuse. WebSphere eXtreme Scale provides a scalable, in-memory data grid. The data grid dynamically caches, partitions, replicates, and manages data across multiple servers. In IBM Integration Bus Version 10.0.0.2 or later, the embedded global cache can use a WebSphere eXtreme Scale enterprise data grid.

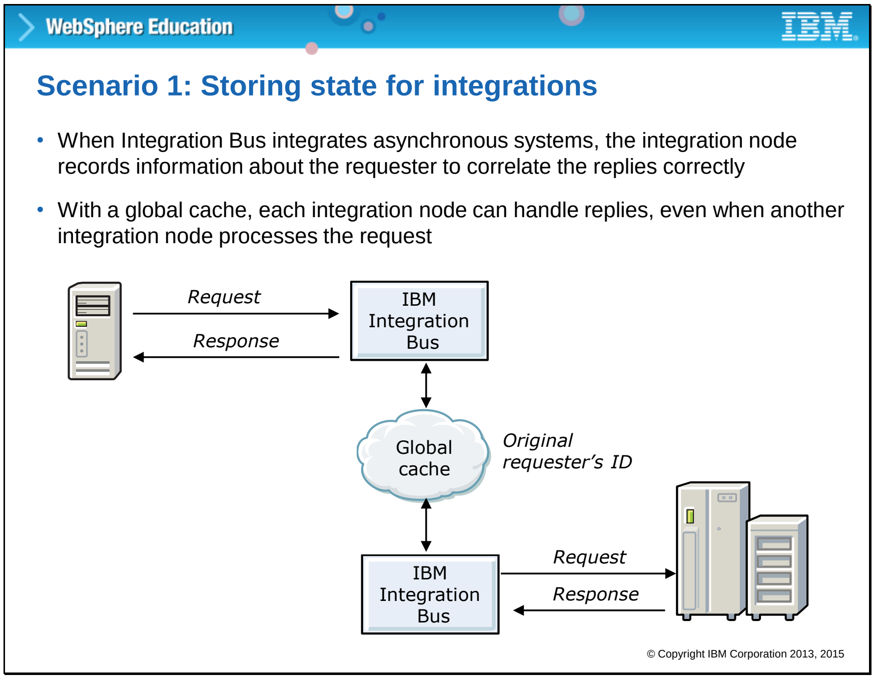
By default, the integration node’s embedded cache is not enabled. To enable the embedded global cache, you must select an appropriate cache policy.

The default cache policy creates a default topology of cache components in a single integration node. The alternatives to the default topology are to have no policy. If you do not have a policy, you can control your own topology by setting cache properties on the integration servers. Or use an XML policy file to enable the cache across multiple integration nodes.

Data in the global cache is stored in *maps*. A map is a data structure that maps keys to values. In a message flow, you can access the Java methods for writing to or reading from the global cache in a JavaCompute node or in Compute node by using the ESQL CREATE PROCEDURE statement. You can also access the global cache by using a Mapping node with a graphical data map.

The next slides describe some scenarios where a global cache is useful.

Slide 4



WebSphere Education

Scenario 1: Storing state for integrations

- When Integration Bus integrates asynchronous systems, the integration node records information about the requester to correlate the replies correctly
- With a global cache, each integration node can handle replies, even when another integration node processes the request

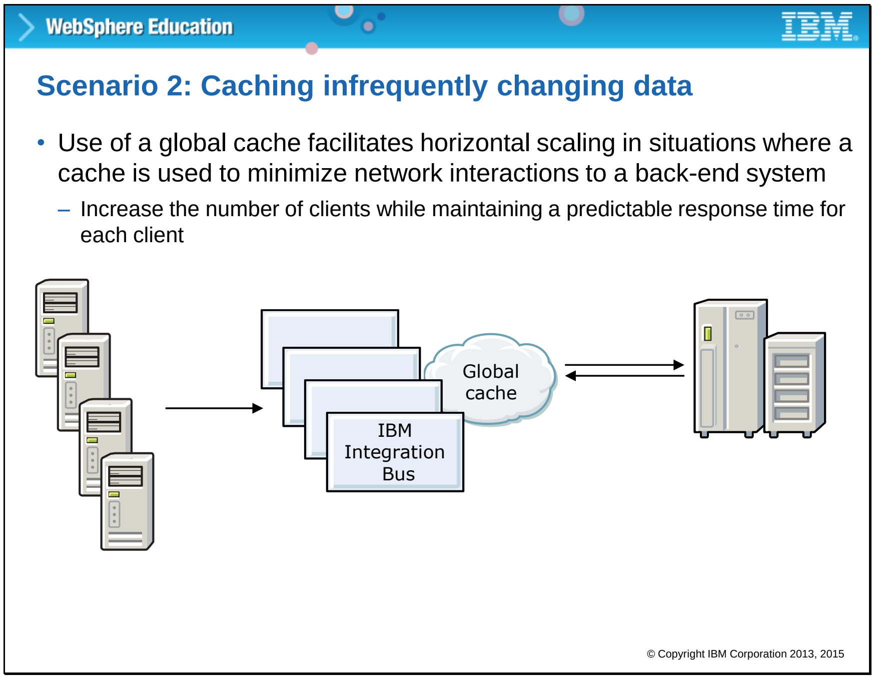
Request
Response
IBM Integration Bus
Global cache
Original requester's ID
IBM Integration Bus
Request
Response

Scenario 1: Storing data for integrations

In a request and reply scenario, the message might pass through other applications that require a different reply-to name. In this type of scenario, the developer typically needs to save the original reply-to information so that it can be restored to ensure that the message is returned to the original requester.

A global cache allows integration nodes to save and share this message information in memory. Before the implementation of a global cache, the problem that is described in this scenario was solved by storing the requester information in an external file or database, for example.

Slide 5



WebSphere Education

Scenario 2: Caching infrequently changing data

- Use of a global cache facilitates horizontal scaling in situations where a cache is used to minimize network interactions to a back-end system
  - Increase the number of clients while maintaining a predictable response time for each client

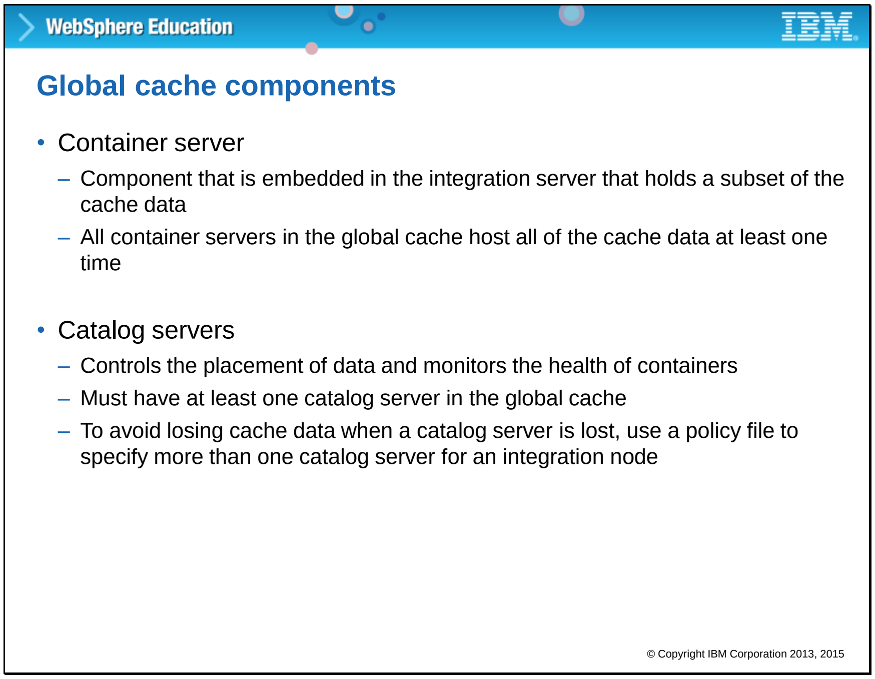
Global cache

IBM Integration Bus

Scenario 2: Caching infrequently changing data

You might have applications that use static information, such as cross-reference or lookup tables. The I/O time to access this information in a database table or a file might be unacceptable. In this scenario, a global cache is used to help to maintain a predictable response time for each client by caching infrequently changing data.

Slide 6



WebSphere Education

IBM

Global cache components

- Container server
  - Component that is embedded in the integration server that holds a subset of the cache data
  - All container servers in the global cache host all of the cache data at least one time
- Catalog servers
  - Controls the placement of data and monitors the health of containers
  - Must have at least one catalog server in the global cache
  - To avoid losing cache data when a catalog server is lost, use a policy file to specify more than one catalog server for an integration node

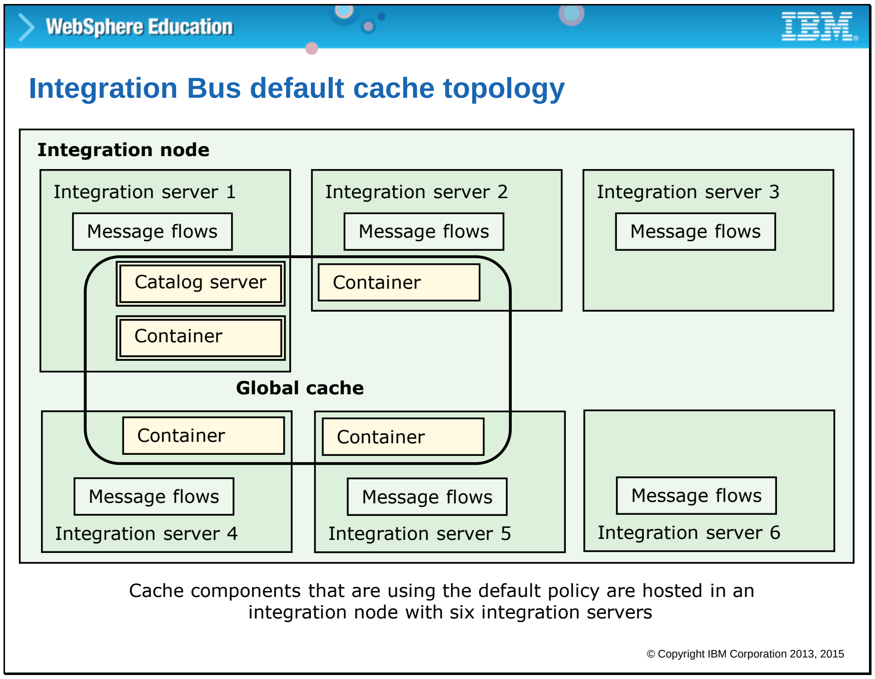
Global cache components

Integration Bus uses two types of components to control the global cache: a container server and a catalog server.

A container server is a component that is embedded in the integration server that holds a subset of the cache data. Between them, all container servers in the global cache host all of the cache data at least one time. If more than one container exists, the default cache policy ensures that all data is replicated at least one time. In this way, the global cache can cope with the loss of container servers without losing data.

The catalog server controls placement of data and monitors the health of containers. You must have at least one catalog server in your global cache. To avoid losing cache data when a catalog server is lost, use a policy file to specify more than one catalog server for an integration node. For example, if you specify two catalog servers for a single integration node, if one catalog server fails, the integration node switches to the other catalog server.

Slide 7



WebSphere Education

Integration Bus default cache topology

Integration node

Integration server 1

Message flows

Catalog server

Container

Integration server 2

Message flows

Container

Integration server 3

Message flows

Global cache

Container

Message flows

Integration server 4

Container

Message flows

Integration server 5

Message flows

Integration server 6

Cache components that are using the default policy are hosted in an integration node with six integration servers

Integration Bus default cache topology

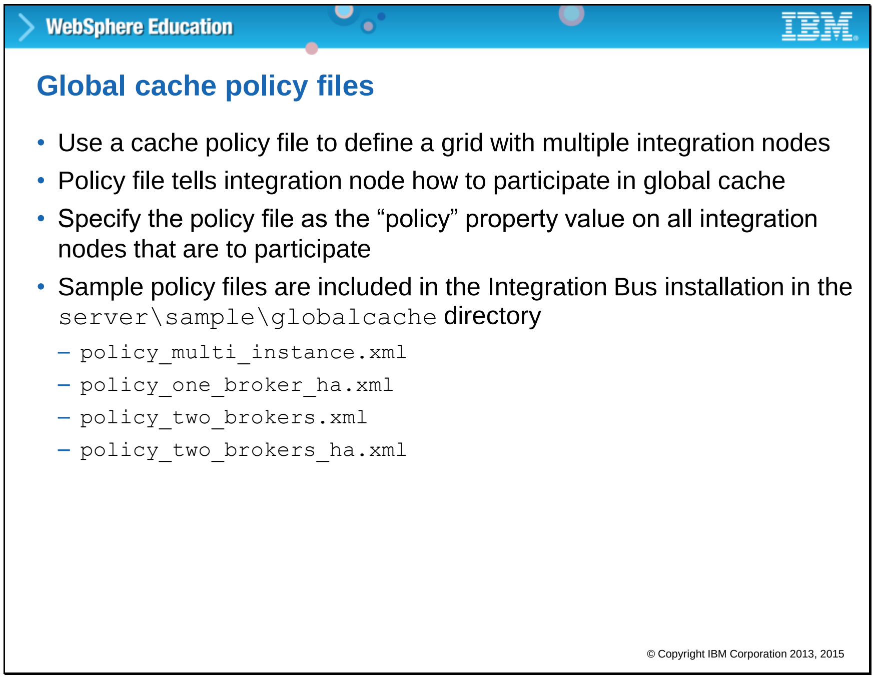
This slide shows the Integration Bus default global cache topology. By default, one integration server in the integration node hosts a catalog server. The catalog server controls placement of data and monitors the health of container servers.

Up to three other integration servers in that integration node can host container servers. A container server is a component that is embedded in the integration server that holds a subset of the cache data.

The catalog server and container servers are placed in integration servers dynamically when the integration node starts. All integration servers can communicate with the global cache, regardless of whether they are hosting catalog servers, container servers, or neither.

Each integration server contains a cache manager, which manages the cache components that are embedded in that integration server.

Slide 8



WebSphere Education

Global cache policy files

- Use a cache policy file to define a grid with multiple integration nodes
- Policy file tells integration node how to participate in global cache
- Specify the policy file as the “policy” property value on all integration nodes that are to participate
- Sample policy files are included in the Integration Bus installation in the `server\sample\globalcache` directory
  - `policy_multi_instance.xml`
  - `policy_one_broker_ha.xml`
  - `policy_two_brokers.xml`
  - `policy_two_brokers_ha.xml`

Global cache policy files

You can configure properties of the embedded global cache by using the Integration web user interface, commands, an XML cache policy file, or the IBM Integration API.

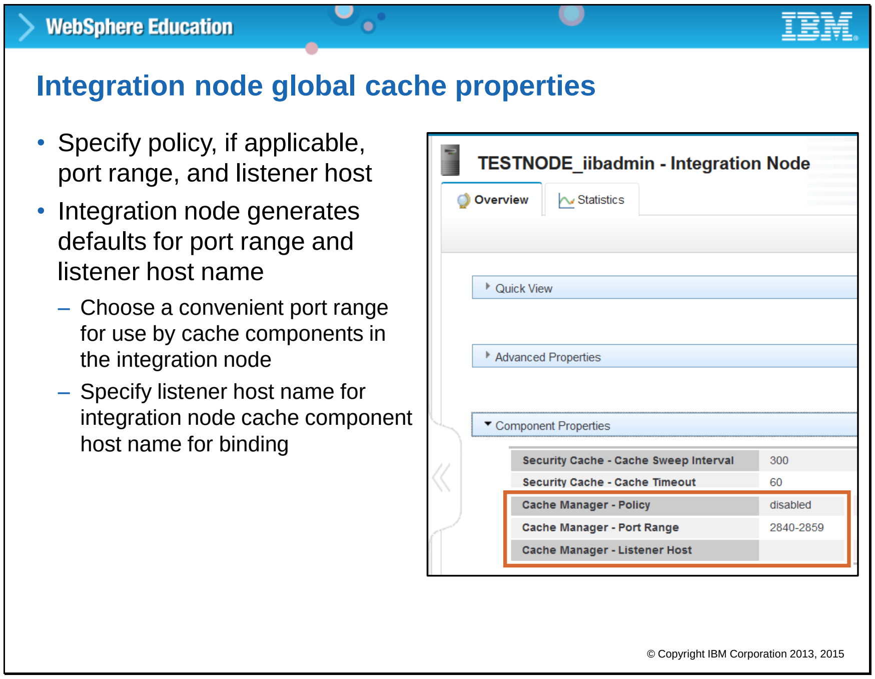
You can customize the default global cache topology to use a specific port range and listener host. You can turn off the default topology and specify your own integration server properties.

You can also use a policy file to configure the global cache to span multiple integration nodes.

The policy file lists the integration nodes that share the cache. For each integration node, the policy file specifies the listener host, port range, and the number of hosted catalog servers.

Specify a policy file for your integration nodes to use by setting the cache policy to the name and file path of the policy file. Sample policy files are provided in the Integration Bus sample\globalcache directory.

Slide 9



Integration node global cache properties

- Specify policy, if applicable, port range, and listener host
- Integration node generates defaults for port range and listener host name
  - Choose a convenient port range for use by cache components in the integration node
  - Specify listener host name for integration node cache component host name for binding

TESTNODE_iibadmin - Integration Node

Security Cache - Cache Sweep Interval	300
Security Cache - Cache Timeout	60
Cache Manager - Policy	disabled
Cache Manager - Port Range	2840-2859
Cache Manager - Listener Host	

Integration node global cache properties

The global cache has a default single-integration node topology that can be used immediately without any configuration. To use the default topology, change the cache policy property to default.

The integration node sets a range of ports to use, but you can specify a particular range of ports. You can also specify the listener host that is used by the integration node cache components. If your computer has more than one host name, setting the listener host ensures that the cache components use the correct host name.

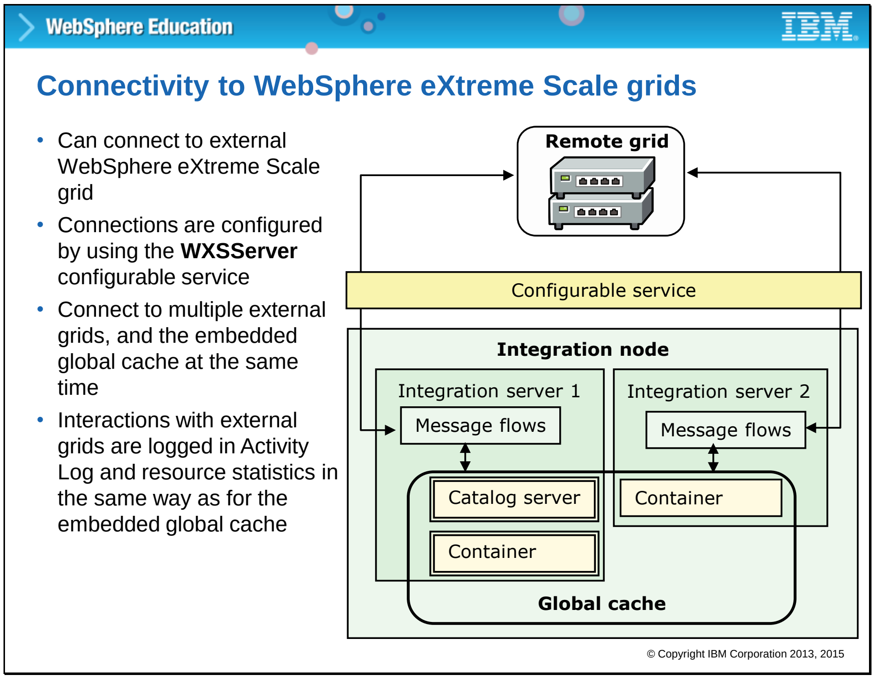
You can set cache manager properties at integration node level and integration server level.

You can set the cache policy to none, default, disabled, or the fully qualified name of an XML policy file.

- If you set the cache policy to default, the default global cache topology is used.
- If you set the cache policy to none, you must set the integration server properties explicitly.
- If you set the cache policy to disabled, all cache components in the integration node are not enabled. The cache is not enabled by default.

- If you specify the fully qualified name of a policy file, the integration nodes that are listed in the policy file are configured to share the data in the global cache. The path must be absolute, not relative.

Slide 10



WebSphere Education

Connectivity to WebSphere eXtreme Scale grids

- Can connect to external WebSphere eXtreme Scale grid
- Connections are configured by using the **WXSServer** configurable service
- Connect to multiple external grids, and the embedded global cache at the same time
- Interactions with external grids are logged in Activity Log and resource statistics in the same way as for the embedded global cache

Remote grid

Configurable service

Integration node

Integration server 1

Message flows

Integration server 2

Message flows

Catalog server

Container

Container

Global cache

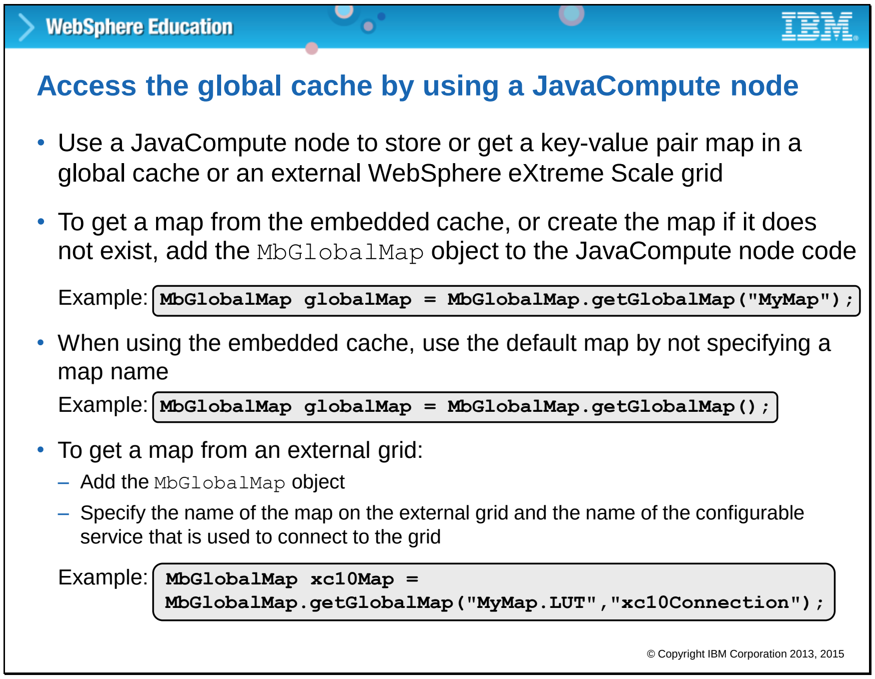
Connectivity to WebSphere eXtreme Scale grids

As an option, you can use one or more external WebSphere eXtreme Scale grids to store data that you want to reuse. An integration node can connect to multiple external grids and the embedded grid, at the same time.

To connect to an external grid, you need the name of the grid, the host name, and port of each catalog server for the grid. By using a configurable service to specify the parameters, you can connect to an external WebSphere eXtreme Scale grid. Create a WXSServer configurable service by using the mqsicreateconfigurableservice command or in the Integration web interface.

Statistics are available for all activity in the embedded global cache and external WebSphere eXtreme Scale grids. One line of statistics is shown for each configurable service that is used to connect to an external grid.

Slide 11



WebSphere Education

Access the global cache by using a JavaCompute node

- Use a JavaCompute node to store or get a key-value pair map in a global cache or an external WebSphere eXtreme Scale grid
- To get a map from the embedded cache, or create the map if it does not exist, add the `MbGlobalMap` object to the JavaCompute node code

  Example:
  ```
  MbGlobalMap globalMap = MbGlobalMap.getGlobalMap("MyMap");
  ```
- When using the embedded cache, use the default map by not specifying a map name

  Example:
  ```
  MbGlobalMap globalMap = MbGlobalMap.getGlobalMap();
  ```
- To get a map from an external grid:
  - Add the `MbGlobalMap` object
  - Specify the name of the map on the external grid and the name of the configurable service that is used to connect to the grid

  Example:
  ```
  MbGlobalMap xc10Map =
  MbGlobalMap.getGlobalMap("MyMap.LUT","xc10Connection");
  ```

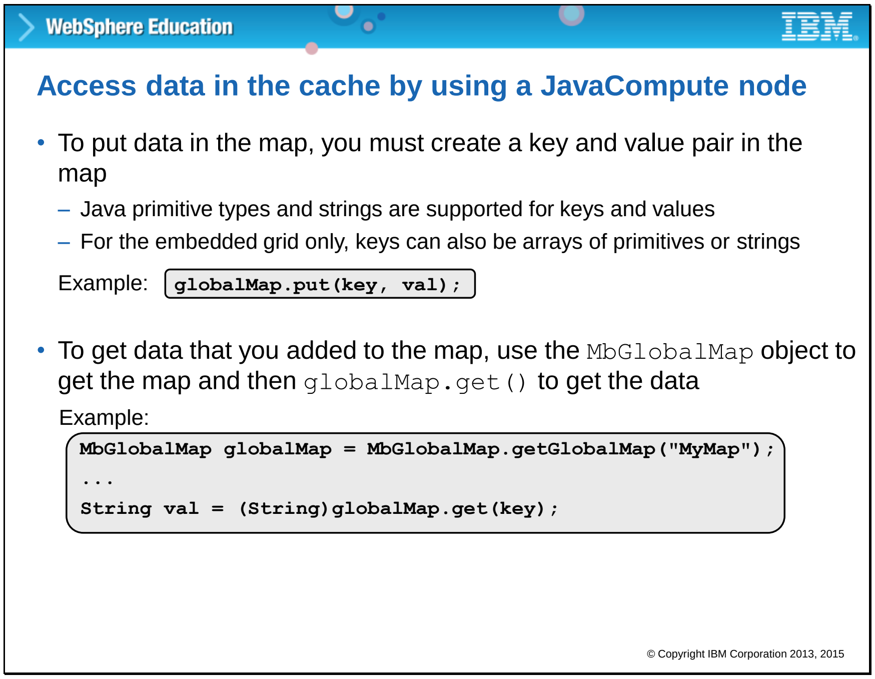
Access the global cache by using a JavaCompute node

In a message flow, you can use a JavaCompute node to interact with a map in a global. You can use a JavaCompute node to store data in a global cache map. You can then create another JavaCompute node to get data from the global cache.

The JavaCompute node uses the MbGlobalMap object to access the global cache or external grid. This class can be used to get a key-value pair map, and to put or get data in a global cache map. You cannot create a global cache map explicitly, but if you get a map that does not exist, the map is created automatically. When you get a global map from an external grid, the getGlobalMap method makes a connection to the grid if one does not exist. The MbGlobalMap object is described in the Java plug-in API documentation.

Interactions with the cache happen outside the message flow transaction, and are committed immediately. If an exception is generated downstream of the node that interacts with the cache, the cache interactions are not rolled back.

Slide 12



WebSphere Education

Access data in the cache by using a JavaCompute node

- To put data in the map, you must create a key and value pair in the map
  - Java primitive types and strings are supported for keys and values
  - For the embedded grid only, keys can also be arrays of primitives or strings

  Example: `globalMap.put(key, val);`

- To get data that you added to the map, use the `MbGlobalMap` object to get the map and then `globalMap.get()` to get the data

  Example:

```
MbGlobalMap globalMap = MbGlobalMap.getGlobalMap("MyMap");
...
String val = (String)globalMap.get(key);
```

Access data in the cache by using a JavaCompute node

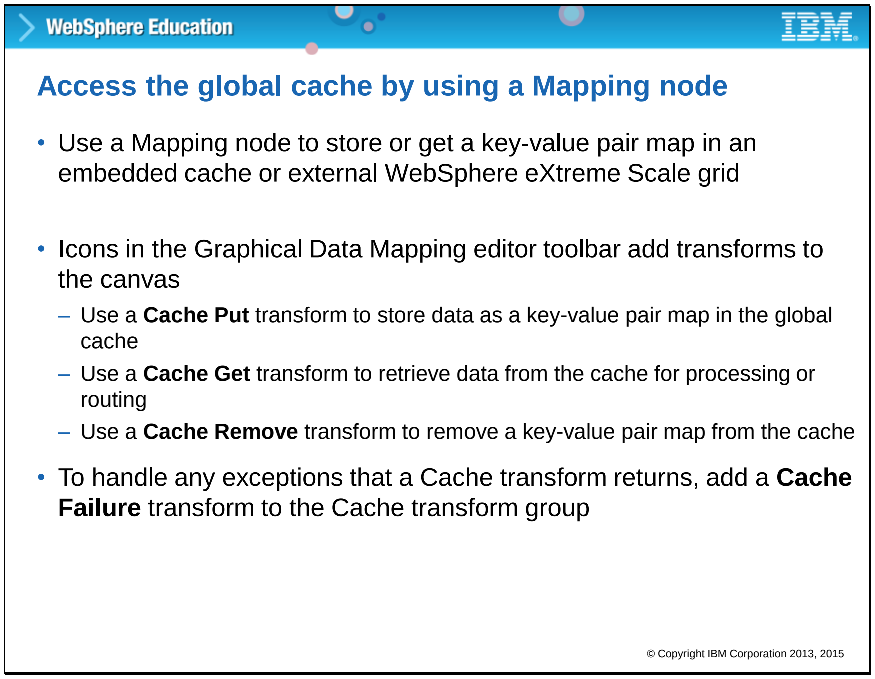
A map is a data structure in the global cache that maps keys to values.

To put data in the map, you must create a key and value pair, as shown in the example. For the Integration Bus embedded grid only, keys can also be arrays of primitives or strings.

If you get a map that does not exist, the map is created automatically. The example gets the map called "MyMap". If the map does not exist, it is created.

To use your own Java classes with the global cache, put the JAR files that contain the Java classes in one of the shared-classes directories. These Java classes must be available to all integration nodes and integration servers that participate in the global cache.

Slide 13



WebSphere Education
IBM

Access the global cache by using a Mapping node

- Use a Mapping node to store or get a key-value pair map in an embedded cache or external WebSphere eXtreme Scale grid
- Icons in the Graphical Data Mapping editor toolbar add transforms to the canvas
  - Use a **Cache Put** transform to store data as a key-value pair map in the global cache
  - Use a **Cache Get** transform to retrieve data from the cache for processing or routing
  - Use a **Cache Remove** transform to remove a key-value pair map from the cache
- To handle any exceptions that a Cache transform returns, add a **Cache Failure** transform to the Cache transform group

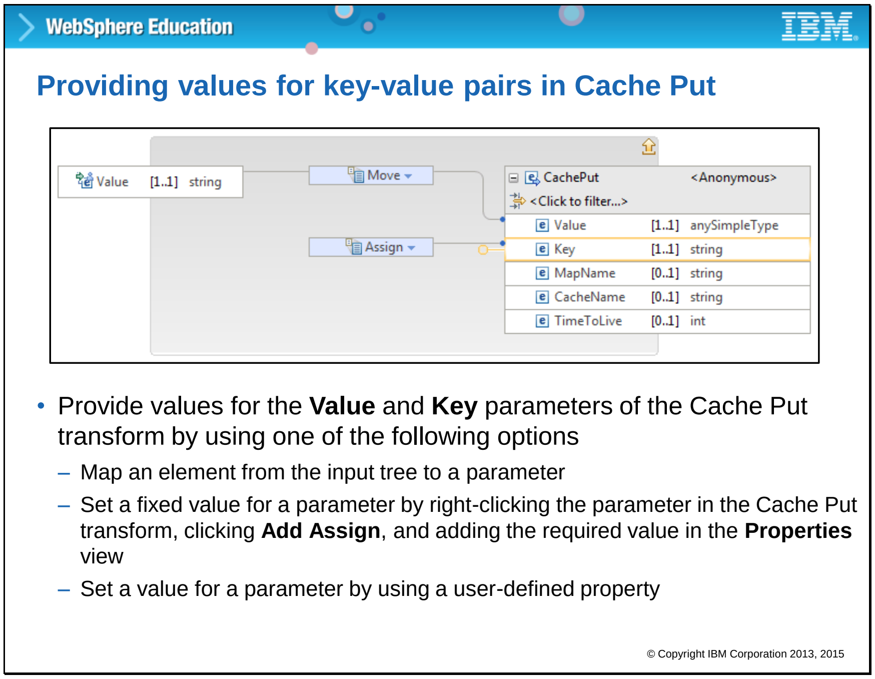
Access the global cache by using a Mapping node

You can use a Mapping node in a message flow to store or get a key-value pair global cache map that is stored in an embedded cache or external grid.

In Integration Bus V10.0.0.2 and later, the Mapping node includes specific transforms to access the global cache.

- You can use a Cache Put transform in a Mapping node to store data in the global cache.
- You can use a Cache Get transform in a Mapping node to retrieve data from the cache for processing or routing.
- You can use a Cache Remove transform in a Mapping node to remove a key-value pair from the cache.
- If you want to handle any exceptions that the Cache transform returns, you can add a Cache Failure transform to the Cache transform group.

Slide 14



Providing values for key-value pairs in Cache Put

- Provide values for the **Value** and **Key** parameters of the Cache Put transform by using one of the following options
  - Map an element from the input tree to a parameter
  - Set a fixed value for a parameter by right-clicking the parameter in the Cache Put transform, clicking **Add Assign**, and adding the required value in the **Properties** view
  - Set a value for a parameter by using a user-defined property

Providing values for key-value pairs in Cache Put

In a Mapping node, you can use the Cache Put transform to provide key-value pairs by using one of the following options:

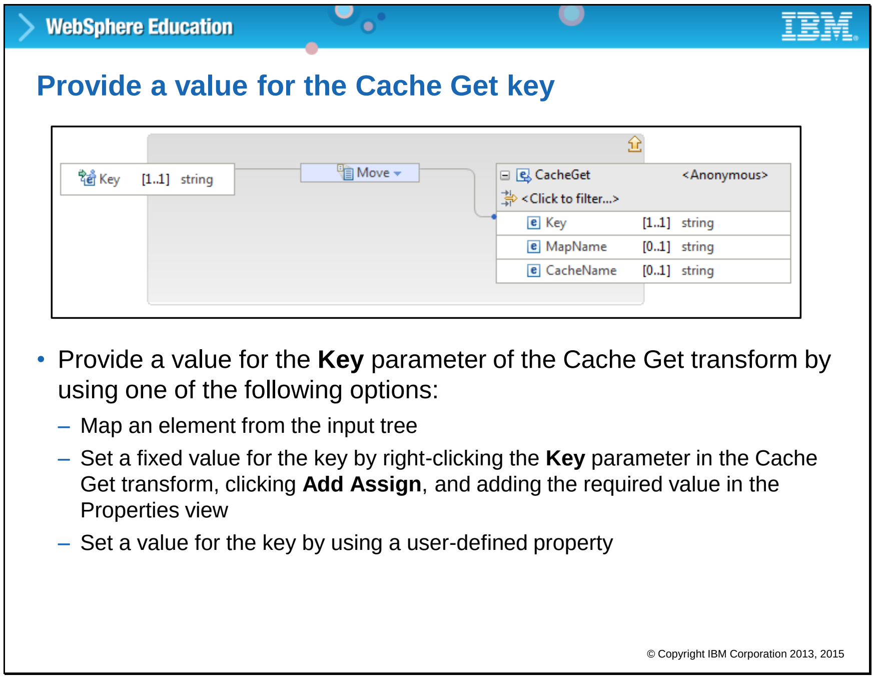
- Map an element from the input tree to a parameter.
- Set a fixed value for a parameter by right-clicking the parameter in the Cache Put transform, clicking **Add Assign**, and adding the required value in the **Properties** view.
- Set a value for a parameter by using a user-defined property.

In the example, the content for the **Value** parameter is provided by mapping an element that is named Value, and the content for the **Key** parameter is provided by assigning a fixed value.

By default, the Cache Put transform adds a key-value pair to the global cache. You can change the action by selecting one of the following options from the **General** tab of the **Properties** for the Cache Put transform:

- Select **Update entry in cache** to update the value that is associated with the key that is already in the cache.
- Select **Insert or update entry in cache** to add the key if it does not exist or update the value that is associated with the key if the key exists.

Slide 15



Provide a value for the Cache Get key

- Provide a value for the **Key** parameter of the Cache Get transform by using one of the following options:
  - Map an element from the input tree
  - Set a fixed value for the key by right-clicking the **Key** parameter in the Cache Get transform, clicking **Add Assign**, and adding the required value in the Properties view
  - Set a value for the key by using a user-defined property

Provide a value for the Cache Get key

You can use a Mapping node to retrieve a value from a map in an embedded cache or external grid.

You can provide a value for the **Key** parameter of the Cache Get transform by using one of the following options:

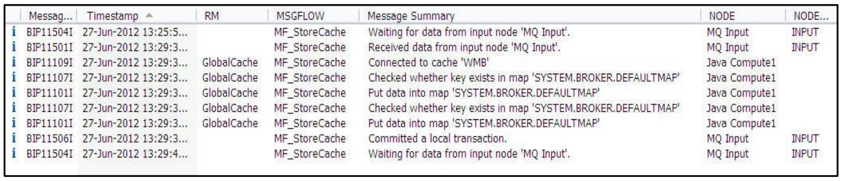
- Map an element from the input tree.
- Set a fixed value for the key by right-clicking the Key parameter in the Cache Get transform, clicking **Add Assign**, and adding the required value in the **Properties** view.
- Set a value for the key by using a user-defined property.

In the example, the content for the Key parameter is provided by mapping an element that is named Key from the input tree.

Slide 16

WebSphere Education

Global cache administrative tools



	Messag...	Timestamp	RM	MSGFLOW	Message Summary	NODE	NODE...
i	BIP11504I	27-Jun-2012 13:25:5...		MF_StoreCache	Waiting for data from input node 'MQ Input'.	MQ Input	INPUT
i	BIP11501I	27-Jun-2012 13:29:3...		MF_StoreCache	Received data from input node 'MQ Input'.	MQ Input	INPUT
i	BIP11109I	27-Jun-2012 13:29:3...	GlobalCache	MF_StoreCache	Connected to cache 'WMB'	Java Compute1	
i	BIP11107I	27-Jun-2012 13:29:3...	GlobalCache	MF_StoreCache	Checked whether key exists in map 'SYSTEM.BROKER.DEFAULTMAP'	Java Compute1	
i	BIP11101I	27-Jun-2012 13:29:3...	GlobalCache	MF_StoreCache	Put data into map 'SYSTEM.BROKER.DEFAULTMAP'	Java Compute1	
i	BIP11107I	27-Jun-2012 13:29:3...	GlobalCache	MF_StoreCache	Checked whether key exists in map 'SYSTEM.BROKER.DEFAULTMAP'	Java Compute1	
i	BIP11101I	27-Jun-2012 13:29:3...	GlobalCache	MF_StoreCache	Put data into map 'SYSTEM.BROKER.DEFAULTMAP'	Java Compute1	
i	BIP11506I	27-Jun-2012 13:29:3...		MF_StoreCache	Committed a local transaction.	MQ Input	INPUT
i	BIP11504I	27-Jun-2012 13:29:4...		MF_StoreCache	Waiting for data from input node 'MQ Input'.	MQ Input	INPUT

- Full resource statistics and activity log in the Integration web interface **Administration Log** provide information about the state of the cache and cache interactions

Global cache administrative tools

You can use the mqsicacheadmin command, resource statistics, and the activity log to monitor the global cache.

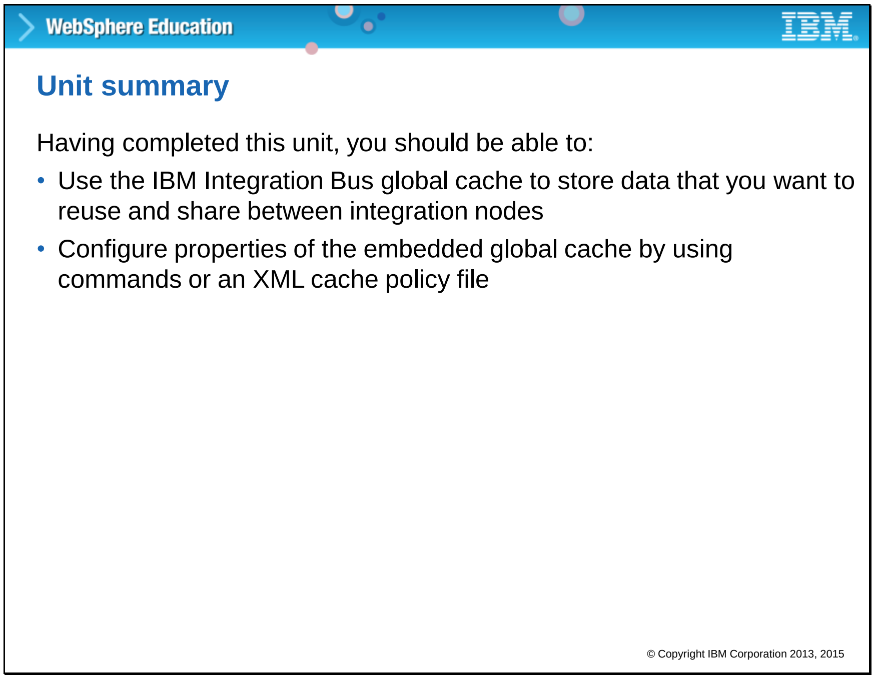
Activity logs provide a high-level overview of how Integration Bus interacts with external resources. The logs help you to understand what your message flows are doing.

An integration node collects resource statistics to record performance and operating details of resources that integration servers use.

You learned about how to access activity logs and resource statistics in the prerequisite course *IBM Integration Bus V10 Application Development I*.

The mqsicacheadmin command provides detailed information about the global cache that is embedded in an integration node. For example, you can find out the size of a global cache map, list the hosts that are participating in the cache, and clear data from a global cache map.

Slide 17



WebSphere Education

Unit summary

Having completed this unit, you should be able to:

- Use the IBM Integration Bus global cache to store data that you want to reuse and share between integration nodes
- Configure properties of the embedded global cache by using commands or an XML cache policy file

Unit summary

A global cache is a repository for data that you want to reuse across processes. IBM Integration Bus supports an embedded global cache and you can configure IBM Integration Bus to connect to an external WebSphere eXtreme Scale grid. In this unit, you learned how to configure and use the IBM Integration Bus embedded global cache to store the data that you want to reuse.

Having completed this unit, you should be able to:

- Use the IBM Integration Bus global cache to store data that you want to reuse and share between integration nodes
- Configure properties of the embedded global cache by using commands or an XML cache policy file