

Using the global cache in WebSphere Message Broker

Arup R. Pal

July 24, 2013

The new global cache feature in WebSphere Message Broker V8.0.0.1 enables you to store and reference data in an embedded memory cache or an external WebSphere eXtreme Scale grid. This article shows you how to implement a global cache to store and access reference data for use by message flows.

Introduction

Most enterprise IT solutions have a requirement for reference data that is accessed frequently during operations. Until recently, this reference data resided in either a file or a database and was accessed through queries. However, querying a file or database consumes time and must be done repeatedly, resulting in a substantial performance hit. To overcome this problem, applications started keeping reference data in a cache in memory, which greatly reduces the time needed to access the data.

IBM® WebSphere® Message Broker V8.0.0.1 (hereafter called Message Broker) provides various ways to manage the cache. You can use shared variables in ESQL to store the data in memory. But shared variables are private to a message flow and cannot be shared across message flows or execution groups, which can lead to duplication of stored data across different flows, and higher memory requirements to store the data.

Alternately, you can load static data into an execution group's JVM heap, so that the data can be shared by multiple flows deployed in an execution group. With this method however, the use of the data is confined to that execution group.

The new global cache feature in Message Broker enables a message broker application to store its reference data in an embedded or external WebSphere eXtreme Scale grid. In an embedded global cache, access to reference data is spread across multiple execution groups that host the eXtreme Scale container servers. The data can be accessed by all execution groups, even if they do not participate in hosting the container servers. This article shows you how to maintain and access static data in a global cache using Java routines from ESQL or Message Broker Mapping nodes.

Maintaining reference data

As mentioned, reference data can be organized and maintained in persistent storage, which can be a database or a file in the file system. For simplicity, this article uses an XML file to organize

and maintain the data. The data in this XML file is organized into different categories, with each category corresponding to a Map element. Each Map element has a specific name and contains a logical group of key- value pairs, as shown below:

Listing 1. Sample XML to maintain static data

```
<?xml version="1.0" encoding="UTF-8"?>
<Maps>
  <Map name="Map1">
    <Entry key="Key1">Value1</Entry>
  </Map>
  <Map name="Map2">
    <Entry key="Key2">Value2</Entry>
    <Entry key="Key3">Value3</Entry>
  </Map></Maps>
```

Loading reference data into the cache

You can read the XML file using an appropriate Input node in a Message Broker flow. The sample flow below uses an MQInput node to read the file from a WebSphere MQ queue. The flow parses the file, loads the data into the global cache, and writes a confirmation message to an output WebSphere MQ queue:

Figure 1. Load cache message flow



Use the Message Broker Java API to interact with the global cache. The Java class `CacheUtil.java` provides the static utility methods to add and retrieve reference data. The Java routine is called from the ESQLCompute node to insert or update the key-value pairs in the global cache. Here is the sample code in the ESQLCompute node to load the key-value pairs into the cache:

Listing 2. Sample ESQL code to load cache

```
DECLARE blnResult      BOOLEAN TRUE;
DECLARE blnMapResult   BOOLEAN;
DECLARE refMap          REFERENCE TO InputRoot.XMLNSC.*[1];

MOVE refMap FIRSTCHILD;
WHILE(LASTMOVE(refMap)) DO
  SET blnMapResult = addMapToCache(refMap);
  IF blnMapResult = FALSE THEN
    SET blnResult = FALSE;
  END IF;
  MOVE refMap NEXTSIBLING REPEAT TYPE NAME;
END WHILE;
```

The code uses the ESQL routine below:

Listing 3. ESQL routine to load map

```
CREATE PROCEDURE addMapToCache (IN refName REFERENCE)
RETURNS BOOLEAN
LANGUAGE JAVA
EXTERNAL NAME "Utilities.CacheUtil.addMap";
```

To access the cache, create the corresponding ESQL routines:

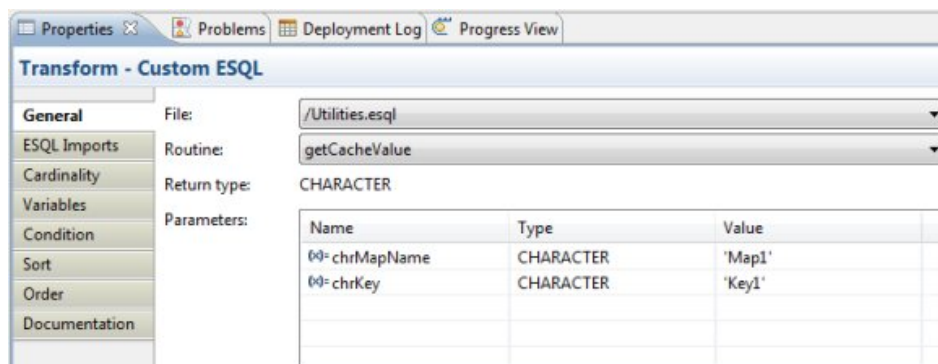
Listing 4. ESQL routines to retrieve data from cache

```
CREATE PROCEDURE getCacheValue(IN chrMapName CHARACTER, IN chrKey CHARACTER)
RETURNS CHARACTER
BEGIN
  DECLARE chrResult CHARACTER;
  SET chrResult = getCacheValueFromWXS(chrMapName,chrKey);
  IF (TRIM(chrResult) IS NULL OR TRIM(chrResult) ='' ) THEN
    THROW USER EXCEPTION MESSAGE 2999 VALUES('Failed to retrieve cache') ;
  END IF;
  RETURN chrResult;
END;

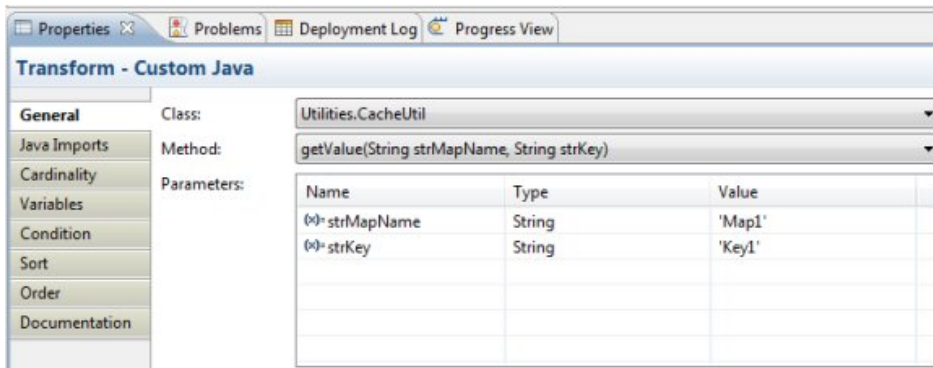
CREATE PROCEDURE getCacheValueFromWXS(IN chrMapName CHARACTER, IN chrKey CHARACTER)
RETURNS CHARACTER
LANGUAGE JAVA
EXTERNAL NAME "Utilities.CacheUtil.getValue";
```

The ESQL wrapper method `getCacheValue` calls the ESQL routine `getCacheValueFromWXS` to retrieve data from the cache, so that you can do your own processing depending on whether the value is obtained successfully. You can call the ESQL routine `getCacheValue` from any ESQLCompute node or by using the Custom ESQL transform in a Mapping node:

Figure 2. Custom ESQL in graphical data map



You can also call the corresponding Java routine directly using a custom Java transform from a Mapping node:

Figure 3. Custom Java in graphical data map

Java routines to access cache

The sample code below implements the static Java methods to access the global cache:

Listing 5. CacheUtil.Java

```
package Utilities;

import com.ibm.broker.plugin.MbElement;
import com.ibm.broker.plugin.MbException;
import com.ibm.broker.plugin.MbGlobalMap;

public class CacheUtil {
    /**
     * Method to get a value from Global Cache using map name and key
     */
    public static String getValue(String strMapName, String strKey) {

        String strValue = null;
        MbGlobalMap globalMap = null;

        try
        {
            globalMap = MbGlobalMap.getGlobalMap(strMapName);
            strValue = (String) globalMap.get(strKey);
        }
        catch(MbException mbe)
        {
            System.out.println(mbe.getMessage());
            mbe.printStackTrace();
        }

        return strValue;
    }

    /**
     * Method to add all the key-value pairs for a map in Global Cache
     */
    public static Boolean addMap(MbElement elmMap) {

        String strValue = null;
        String strKey = null;
        String strMapName = null;
        MbGlobalMap globalMap = null;

        try
        {
```

```

    elmMap = elmMap.getFirstChild();
    strMapName = elmMap.getValueAsString();

    globalMap = MbGlobalMap.getGlobalMap(strMapName);

    MbElement elmEntry = elmMap.getNextSibling();

    while (elmEntry != null) {

        strKey = elmEntry.getFirstChild().getValueAsString();
        strValue = elmEntry.getValueAsString();

        if(globalMap.containsKey(strKey)) {
            globalMap.update(strKey, strValue);
        } else {
            globalMap.put(strKey, strValue);
        }

        elmEntry = elmEntry.getNextSibling();
    }
}
catch(MbException mbe) {
    System.out.println(mbe.getMessage());
    mbe.printStackTrace();
    return Boolean.FALSE;
}
return Boolean.TRUE;
}

/**
 * Method to add a key-value pair to a map in Global Cache
 */
public static Boolean addUpdateKey(String strMapName, String strKey, String strValue) {

    MbGlobalMap globalMap = null;

    try
    {
        globalMap = MbGlobalMap.getGlobalMap(strMapName);

        if(globalMap.containsKey(strKey)) {
            globalMap.update(strKey, strValue);
        } else {
            globalMap.put(strKey, strValue);
        }
    }
    catch(MbException mbe) {
        System.out.println(mbe.getMessage());
        mbe.printStackTrace();
        return Boolean.FALSE;
    }

    return Boolean.TRUE;
}
}

```

Summary of Java methods in CacheUtil.java class

Table 1. Methods in CacheUtil.java

Modifier and type	Method	Description
static Boolean	addMap(MbElement elmMap)	This method gets the map name from the input map element and fetches the global map of the given name from the global cache. If an

		<p>existing map is found, the key value pairs from the input map element are added to the map (for those keys that are not present) or the values are updated in the map (for those keys that are already present).</p> <p>Parameters: <i>elmMap</i>: Map element</p> <p>Returns: Boolean True if all keys refreshed successfully, otherwise False.</p>
static String	getValue(String strMapName, String strKey)	<p>This method fetches the map with the specified map name from the global cache and returns the value associated with the specified key from the map.</p> <p>Parameters: <i>strMapName</i>: Name of the map, <i>strKey</i>: Key to look up in this map</p> <p>Returns: Value associated with the key from the map if it exists, otherwise null.</p>
static Boolean	addUpdateKey(String strMapName, String strKey, String strValue)	<p>This method fetches the global map of the given name from the global cache. If an existing map is found, the given key value pairs are added to the map (if the key is not present) or the value is updated in the map (if the key is already present).</p> <p>Parameters: <i>elmMap</i>: Map element</p> <p>Returns: Boolean True if all keys refreshed successfully, otherwise False.</p>

Conclusion

This article showed a sample implementation of the new WebSphere Message Broker global cache feature in an enterprise environment to store and access reference data. The global cache is useful in many other scenarios, and you can write similar ESQL and Java routines to access the global cache depending on your requirements.

Downloadable resources

Description	Name	Size
Code sample	CodeSample.zip	11 KB

Related topics

- **WebSphere Message Broker resources**

- [WebSphere Message Broker V8 information center](#)
A single Web portal to all WebSphere MQ V7 documentation, with conceptual, task, and reference information on installing, configuring, and using WebSphere Message Broker V8.
- [WebSphere Message Broker product page](#)
Product descriptions, product news, training information, support information, and more.
- [Download free trial version of WebSphere Message Broker](#)
WebSphere Message Broker is an ESB built for universal connectivity and transformation in heterogeneous IT environments. It distributes information and data generated by business events in real time to people, applications, and devices throughout your extended enterprise and beyond.
- [WebSphere Message Broker documentation library](#)
WebSphere Message Broker specifications and manuals.
- [WebSphere Message Broker forum](#)
Get answers to technical questions and share your expertise with other WebSphere Message Broker users.

- **IBM Integration Bus resources**

- [IBM Integration Bus V9 information center](#)
A single Web portal to all IBM Integration Bus documentation, with conceptual, task, and reference information on installing, configuring, migrating to, and using BM Integration Bus.
- [IBM Integration Bus product page](#)
Product features, use cases, and resources.
- [Download IBM Integration Bus Developer Edition](#)
A lightweight edition that you can use for evaluation, development, unit test, and other scenarios.
- [What's new in IBM Integration Bus V9](#)
Introductory topic in the IBM Integration Bus Information Center.
- [What's new for WebSphere ESB users](#)
Description of key differences between IBM Integration Bus and WebSphere ESB.
- [IBM Integration Bus V9 announcement letter](#)
Official announcement information, including prerequisites, terms and conditions, and ordering information.
- [IBM Integration Bus V9 introductory video](#)
A short YouTube video showing key IBM Integration Bus features.
- [Follow IBM Integration Bus on Twitter](#)
Latest IBM Integration Bus news and announcements,
- [IBM Integration Bus forum](#)
Forum on mqseries.net for user questions, answers, and tips.
- [Track IBM Integration Bus user requirements](#)
Create, view, and track IBM Integration Bus user requirements.

© Copyright IBM Corporation 2013

(www.ibm.com/legal/copytrade.shtml)

[Trademarks](#)

(www.ibm.com/developerworks/ibm/trademarks/)