

Abstract

Almost every store built in IBM WebSphere shares the common premise of having a location. The intent of this document is to provide a way to integrate a store locator feature in any store website by providing live Maps of stores available near your requested location. As an instance we have implemented a Store locator in WebSphere Commerce using MapQuest Site Advantage™ Web Service [1].

Introduction

In this article, we will demonstrate how an existing IBM WebSphere Commerce application can provide Store locator functionality and their corresponding Maps for stores near a user's requested location directly by connecting to MapQuest Site Advantage™ Web Service. By virtue of that one can incorporate the zip locator functionality in their store.

IBM WebSphere Commerce automates and integrates online marketing and sales processes across multiple channels. Moreover, it is very helpful so that you can easily develop a highly secure, scalable online store. WebSphere Commerce can be used for any or all of a company's business models and touch points. It is the single, unified platform you need to do business directly with consumers, with businesses, indirectly through channel partners - or all of the above simultaneously [2].

Site Advantage Web service facilitates developers to integrate maps, driving directions and location searching into applications. The nucleus of the system is a MapQuest-hosted server that provides features for maps, driving directions, and location searching. The Web service's interface enables integration with any enterprise system. In this case, we are using standard TCP/IP network connections and HTTP protocol to exchange data and images in standard formats.

Requirements

Since the basic idea is about integrating store locator with WebSphere Commerce server, we will show how to deploy store locator on Commerce Server. In order to understand this document the user must have some knowledge about J2EE, XML, Web Service and web development and IBM WebSphere Commerce. However, you can use other J2EE servers such as tomcat, BEA or others.

We are using WebSphere Commerce Express Developer Edition 6. We require an existing Commerce application and then all you need is an account with MapQuest Server. After opening a MapQuest account you will receive a ClientID that helps in authorizing your account over MapQuest server for each request sent. Moreover, a DNS entry is required to resolve server called from WebSphere Commerce Server (WCS). Furthermore, WC server IP must also be registered with MapQuest as well for security purpose.

Assumptions

Before we get started, let's go over the technologies and other requirements needed to set up our Store Locator. The assumption in this document is that our Stores locations data is loaded into MapQuest hosted server in advance and Store Locator act as client to that server that uses MapQuest Site Advantage APIs to search and list our Stores locations, maps and driving directions. Moreover, we are avoiding the intrinsic pains for transforming XML using XSLT. Therefore, we need a framework for binding XML data to Java objects that can lets us work with data from XML documents using our own class structures. We have used open source framework called JiBX that is downloadable from <http://jibx.sourceforge.net/>. This has circumvented the need for handling XML translation and encapsulated all the details of converting data to and from XML based on your instructions.

Developing the client side

As stated earlier, we are using WebSphere Commerce Developer Edition 6 that can only be installed over WebSphere Studio Application Developer. However, one can use other Integration Development Environments (IDE) of choice.

Now for establishing communication with the MapQuest Web service we need to write a binding xml to understand the XML received. Below we have provided the snippet of the binding file for our received XML decided with MapQuest.

```
<?xml version="1.0" encoding="UTF-8" ?>
<binding>
<mapping class="com.royalcyber.commerce.mapquest.Distance" abstract="true">
  <value style="attribute" name="units" field="units" usage="optional" />
  <value style="text" field="distance" usage="required" />
</mapping>
<mapping name="point" class="com.royalcyber.commerce.mapquest.PointType"> <value
  name="x" field="x" usage="required" />
  <value name="y" field="y" usage="required" />
  <value name="data" field="data" usage="required" />
  <value name="recordId" field="recordId" usage="required" />
</mapping>
<mapping name="rollover" class="com.royalcyber.commerce.mapquest.Rollover">
```

```

< collection field="pointList" usage="optional">
  < structure map-as="com.royalcyber.commerce.mapquest.PointType" />
</collection>
< value name="count" field="count" usage="optional" style="attribute" />
</mapping>
< mapping class="com.royalcyber.commerce.mapquest.Map" abstract="true">
  < value name="height" field="height" usage="required" />
  < value name="width" field="width" usage="required" />
  < value name="type" field="type" usage="required" />
  < value name="zoomLevel" field="zoomLevel" usage="optional" />
  < value name="latitude" field="latitude" usage="optional" />
  < value name="longitude" field="longitude" usage="optional" />
  < value name="mapStyle" field="mapStyle" usage="optional" />
  < value name="request" field="request" usage="required" />
  < value name="mapSessionId" field="mapSessionId" usage="optional" />
  < structure field="rollover" usage="optional" map-
    as="com.royalcyber.commerce.mapquest.Rollover" />
</mapping>
< mapping name="userFields" class="com.royalcyber.commerce.mapquest.UserFields">
  < value name="user1" field="user1" usage="optional" />
  < value name="user10" field="user10" usage="optional" />
  < value name="user2" field="user2" usage="optional" />
  < value name="user3" field="user3" usage="optional" />
  < value name="user4" field="user4" usage="optional" />
  < value name="user5" field="user5" usage="optional" />
  < value name="user6" field="user6" usage="optional" />
  < value name="user7" field="user7" usage="optional" />
  < value name="user8" field="user8" usage="optional" />
  < value name="user9" field="user9" usage="optional" />
</mapping>
< mapping name="searchFields" class="com.royalcyber.commerce.mapquest.SearchFields" />
< mapping name="categories" class="com.royalcyber.commerce.mapquest.CategoryFields" />
< mapping name="addData" class="com.royalcyber.commerce.mapquest.AddDataFields" />
< mapping name="location" class="com.royalcyber.commerce.mapquest.Location">
  < value name="number" field="number" usage="optional" />
  < value name="name" field="name" usage="optional" />
  < value name="address" field="address" usage="optional" />
  < value name="city" field="city" usage="optional" />
  < value name="county" field="county" usage="optional" />
  < value name="stateProvince" field="stateProvince" usage="optional" />
  < value name="postalCode" field="postalCode" usage="optional" />
  < value name="country" field="country" usage="optional" />
  < value name="recordId" field="recordId" usage="optional" />
  < value name="latitude" field="latitude" usage="optional" />
  < value name="longitude" field="longitude" usage="optional" />
  < structure name="distance" field="distance" usage="optional"
    type="com.royalcyber.commerce.mapquest.Distance" />
  < value name="dataSource" field="dataSource" usage="optional" />
  < value name="geocodeQuality" field="geocodeQuality" usage="optional" />
  < value name="iconId" field="iconId" usage="optional" />
  < structure field="userFields" usage="optional" map-
    as="com.royalcyber.commerce.mapquest.UserFields" />
  < structure field="searchFields" usage="optional" map-
    as="com.royalcyber.commerce.mapquest.SearchFields" />
  < structure field="categories" usage="optional" map-
    as="com.royalcyber.commerce.mapquest.CategoryFields" />
  < structure field="addData" usage="optional" map-
    as="com.royalcyber.commerce.mapquest.AddDataFields" />
  < structure name="map" field="map" usage="optional"
    type="com.royalcyber.commerce.mapquest.Map" />
</mapping>
< mapping class="com.royalcyber.commerce.mapquest.LocationCollection" abstract="true">

```

```

< value name="status" field="status" usage="optional" />
< collection field="locationList">
< structure map-as="com.royalcyber.com merce.mapquest.Location" />
</collection>
< value name="count" field="count" usage="optional" style="attribute" />
< value name="number" field="number" usage="optional" style="attribute" />
</mapping>
< mapping class="com.royalcyber.com merce.mapquest.LocationCollectionCollection"
  abstract="true">
< collection field="locationsList">
< structure name="locations" type="com.royalcyber.com merce.mapquest.LocationCollection" />
</collection>
< value name="count" field="count" usage="optional" style="attribute" />
</mapping>
< mapping name="maneuver" class="com.royalcyber.com merce.mapquest.Maneuver">
< value name="number" field="number" usage="required" />
< value name="text" field="text" usage="required" />
< value name="time" field="time" usage="required" />
< structure name="distance" field="distance" usage="required"
  type="com.royalcyber.com merce.mapquest.Distance" />
< structure name="thumbnailMap" field="thumbnailMap" usage="optional"
  type="com.royalcyber.com merce.mapquest.Map" />
< structure name="searchResults" field="searchResults" usage="optional"
  type="com.royalcyber.com merce.mapquest.LocationCollection" />
</mapping>
< mapping class="com.royalcyber.com merce.mapquest.ManeuverCollection" abstract="true">
< collection field="maneuverList">
  < structure map-as="com.royalcyber.com merce.mapquest.Maneuver" />
</collection>
< value name="totalTime" field="totalTime" usage="optional" />
< structure name="totalDistance" field="totalDistance" usage="optional"
  type="com.royalcyber.com merce.mapquest.Distance" />
< value name="prequel" field="prequel" usage="optional" />
< value name="sequel" field="sequel" usage="optional" />
< value name="count" field="count" usage="optional" style="attribute" />
< value name="latw Count" field="latw Count" usage="optional" style="attribute" />
</mapping>
< mapping name="leg" class="com.royalcyber.com merce.mapquest.RouteLeg">
< value name="number" field="number" usage="required" />
< collection field="maneuversList">
< structure name="maneuvers" type="com.royalcyber.com merce.mapquest.ManeuverCollection"
  /> </collection>
< value name="time" field="time" usage="required" />
< structure name="distance" field="distance" usage="required"
  type="com.royalcyber.com merce.mapquest.Distance" />
< structure name="map" field="map" usage="optional"
  type="com.royalcyber.com merce.mapquest.Map" />
< value name="count" field="count" usage="optional" style="attribute" />
</mapping>
< mapping name="legs" class="com.royalcyber.com merce.mapquest.RouteLegCollection">
< collection field="legList">
< structure map-as="com.royalcyber.com merce.mapquest.RouteLeg" /> </collection>
< value name="count" field="count" usage="optional" style="attribute" /> </mapping>
< mapping class="com.royalcyber.com merce.mapquest.ParametersCollection" abstract="true">
< value name="count" field="count" usage="optional" style="attribute" />
< value name="clientid" field="clientI d" usage="optional" style="attribute" />
< value name="recordid" field="recordI d" usage="optional" style="attribute" />
< value name="transaction" field="transaction" usage="optional" style="attribute" />
</mapping>
< mapping class="com.royalcyber.com merce.mapquest.InputCollection" abstract="true">
< structure name="locations" field="locations" usage="optional"
  type="com.royalcyber.com merce.mapquest.LocationCollection" />

```

```

< structure name="poiResults" field="poiResults" usage="optional"
  type="com.royalcyber.com merce.mapquest.LocationCollection" />
< structure name="origin" field="origin" usage="optional"
  type="com.royalcyber.com merce.mapquest.LocationCollection" />
< structure name="interDests" field="interDests" usage="optional"
  type="com.borders.com merce.mapquest.LocationCollectionCollection" />
< structure name="destination" field="destination" usage="optional"
  type="com.royalcyber.com merce.mapquest.LocationCollection" /> </mapping>
< mapping name="locMap" class="com.royalcyber.com merce.mapquest.LocMapResponse">
< structure name="parameters" />
< structure name="input" field="input" usage="required"
  type="com.royalcyber.com merce.mapquest.InputCollection" />
< structure name="locations" field="locations" usage="required"
  type="com.royalcyber.com merce.mapquest.LocationCollection" />
< value name="status" field="status" usage="optional" style="attribute" /> </mapping>
< mapping name="route" class="com.royalcyber.com merce.mapquest.RouteResponse">
< structure name="parameters" />
< structure name="input" field="input" usage="required"
  type="com.royalcyber.com merce.mapquest.InputCollection" />
< structure name="origin" field="origin" usage="required"
  type="com.royalcyber.com merce.mapquest.LocationCollection" />
< structure name="destination" field="destination" usage="required"
  type="com.royalcyber.com merce.mapquest.LocationCollection" />
< structure name="overviewMap" field="overviewMap" usage="optional"
  type="com.royalcyber.com merce.mapquest.Map" />
< structure name="latwWidth" field="latwWidth" usage="optional"
  type="com.royalcyber.com merce.mapquest.Distance" />
< value name="proxl conl d" field="proxl conl d" usage="optional" />
< structure name="maneuvers" field="maneuvers" usage="optional"
  type="com.royalcyber.com merce.mapquest.ManeuverCollection" />
< value name="status" field="status" usage="optional" style="attribute" /> </mapping>
< mapping name="m proute" class="com.royalcyber.com merce.mapquest.MpRouteResponse">
< structure name="parameters" field="parameters" usage="required"
  type="com.royalcyber.com merce.mapquest.ParametersCollection" />
< structure name="input" field="input" usage="required"
  type="com.royalcyber.com merce.mapquest.InputCollection" />
< structure name="origin" field="origin" usage="required"
  type="com.royalcyber.com merce.mapquest.LocationCollection" />
< structure name="interDests" field="interDests" usage="required"
  type="com.royalcyber.com merce.mapquest.LocationCollectionCollection" />
< structure name="destination" field="destination" usage="required"
  type="com.royalcyber.com merce.mapquest.LocationCollection" />
< structure name="overviewMap" field="overviewMap" usage="optional"
  type="com.royalcyber.com merce.mapquest.Map" />
< structure name="latwWidth" field="latwWidth" usage="optional"
  type="com.royalcyber.com merce.mapquest.Distance" />
< value name="proxl conl d" field="proxl conl d" usage="optional" />
< structure field="legs" usage="optional" map-
  as="com.royalcyber.com merce.mapquest.RouteLegCollection" />
< value name="totalTime" field="totalTime" usage="optional" />
< structure name="totalDistance" field="totalDistance" usage="optional"
  type="com.royalcyber.com merce.mapquest.Distance" />
< value name="status" field="status" usage="optional" style="attribute" /> </mapping>
< mapping name="search" class="com.royalcyber.com merce.mapquest.SearchResponse">
< structure name="parameters" />
< structure name="input" field="input" usage="required"
  type="com.royalcyber.com merce.mapquest.InputCollection" />
< structure name="locations" field="locations" usage="required"
  type="com.royalcyber.com merce.mapquest.LocationCollection" />
< structure name="map" field="map" usage="optional"
  type="com.royalcyber.com merce.mapquest.Map" />
< structure name="radius" field="radius" usage="optional"
  type="com.royalcyber.com merce.mapquest.Distance" />

```

```

< value name="proxIconId" field="proxIconId" usage="optional" />
< structure name="searchResults" field="searchResults" usage="optional"
  type="com.royalcyber.commerce.mapquest.LocationCollection" />
< value name="nextData" field="nextData" usage="optional" />
< value name="prevData" field="prevData" usage="optional" />
< value name="status" field="status" usage="optional" style="attribute" /> </mapping>
< mapping name="poiMap" class="com.royalcyber.commerce.mapquest.PoiMapResponse">
< structure name="parameters" field="parameters" usage="required"
  type="com.royalcyber.commerce.mapquest.ParametersCollection" />
< structure name="input" field="input" usage="required"
  type="com.royalcyber.commerce.mapquest.InputCollection" />
< structure name="locations" field="locations" usage="optional"
  type="com.royalcyber.commerce.mapquest.LocationCollection" />
< structure name="poiResults" field="poiResults" usage="optional"
  type="com.royalcyber.commerce.mapquest.LocationCollection" />
< structure name="map" field="map" usage="optional"
  type="com.royalcyber.commerce.mapquest.Map" />
< value name="status" field="status" usage="optional" style="attribute" /> </mapping>
< mapping name="error" class="com.royalcyber.commerce.mapquest.ErrorResponse">
< structure name="parameters" />
< structure name="input" field="input" usage="optional"
  type="com.royalcyber.commerce.mapquest.InputCollection" />
< value name="code" field="code" usage="required" />
< value name="text" field="text" usage="required" />
< value name="querystring" field="querystring" usage="optional" />
< value name="status" field="status" usage="optional" style="attribute" />
</mapping>
< mapping name="advantage"
  class="com.royalcyber.commerce.mapquest.AdvantageResponse">
< structure field="locMap" usage="optional" map-
  as="com.royalcyber.commerce.mapquest.LocMapResponse" />
< structure field="route" usage="optional" map-
  as="com.royalcyber.commerce.mapquest.RouteResponse" />
< structure field="mproute" usage="optional" map-
  as="com.royalcyber.commerce.mapquest.MpRouteResponse" />
< structure field="search" usage="optional" map-
  as="com.royalcyber.commerce.mapquest.SearchResponse" />
< structure field="poiMap" usage="optional" map-
  as="com.royalcyber.commerce.mapquest.PoiMapResponse" />
< structure field="error" usage="optional" map-
  as="com.royalcyber.commerce.mapquest.ErrorResponse" />
</mapping>
</binding>

```

For binding, first we need to compile our classes made for the binding then run the following command from the command prompt location where the above binding class which is MapQuestBinding.xml reside.

```

java -cp .; C:\VCToolkit\EE60\workspace\bin; C:\lib; C:\lib\jibx-binding.jar
org.jibx.binding.CompileMapQuestBinding.xml

```

Here we are setting the class path in the -cp flag for .class file location for mapping classes and libraries of JiBX available.

After Binding the XML we could easily communicate with the MapQuest web service by calling the server for store map, list of available store for a zip code and driving directions. So let's begin, assuming that you have published the sample store ConsumerDirect from CommerceAccelerator and imported some sample data. So right click Stores\Web Content\ConsumerDirect and make a folder called "ZipCodeLocator" and create in it. The following JSP snippet can be use as sample.

CODE 1 (StoresLocatorForm.jsp)

```
<html> <!-- Created By Danish Najam -->
<head><title>Stores Locator</title></head>
<body>
  <form name="SearchLocationForm" METHOD="POST" ACTION="StoresLocationRequest">
    Keywords &nbsp;
    <input name="textStoreName" type="text" />
    <br/>City &nbsp;
    <input name="textCity" type="text" />
    <br/>State &nbsp;
    <select name="textState">
      <option value="">select a state</option>
      <option value="AL">Alabama</option>
      <option value="AK">Alaska</option>
      <option value="AZ">Arizona</option>
    </select>
    <br/>
    <input name="textZipCode" type="text" />
    <br/>
    <a href="#" onclick="document.SearchLocationForm.submit();">
      Search Store Location
    </a>
    <input type="hidden" name="radius" value="10" />
    <input type="hidden" name="units" value="1" />
    <input type="hidden" name="city" value="1" />
    <input type="hidden" name="stateProvince" value="1" />
    <input type="hidden" name="postalCode" value="1" />
    <input type="hidden" name="searchtype" value="1" />
  </form>
</body>
</html>
```

The above code send the request to our SmartDataBean which internally called the setMapQuestRequestURL method to form a MapQuest URL that would be place to call the MapQuest Web service and return an XML document which needed to be parsed with JiBX binding API.

```
// Created By Danish Najam
private String setMapQuestRequestURL(int requestType, String extraURLParameters) {
  this.requestType = requestType;
  StringBuffer buffer = new StringBuffer();
  buffer.append( "http://xml.sa.mapquest.com/?clientid=YourIdGotFromMapQuest" );
  try {
    switch(requestType) {
      case SEARCH: // declared as 1 fetched from searchtype
        buffer.append(
          "&transaction=search&mapStyle=european&maxSearchResults=100&pageResults=100");
        buffer.append( "&radius=" + originLocation.getDistance().getDistance() + "&units="
          + originLocation.getDistance().getUnits() );
        if(originLocation.getCity() != "")
          buffer.append( "&city=" + URLEncoder.encode(originLocation.getCity(), "UTF-8"));
        if(originLocation.getStateProvince() != "")
```

```

        buffer.append("&stateProvince=" +
            URLEncoder.encode(originLocation.getStateProvince(), "UTF-8") );
        if(originLocation.getPostalCode() != "")
            buffer.append("&postalCode=" +
                URLEncoder.encode(originLocation.getPostalCode(), "UTF-8"));
        break;
        // . . . Some other breaks conditions which will be shown later as required
        case MAP:
            default:
                buffer.append( "&transaction=locMap&mapStyle=european&recordId=" + storeId );
    }
} catch (UnsupportedEncodingException e) {
    e.printStackTrace();
}
return buffer.toString();
}

```

Later when we need the store list, we would get the list of available stores using following method `getMapQuestAdvantageResponse()` after the call dispatched we will going to have an object of `AdvantageResponse` class, which in turn give us `Location` class's Objects collection that does contains all the data saved over MapQuest server for our store.

```

// Created By Danish Najam
public AdvantageResponse getMapQuestAdvantageResponse () throws JibXException, IOException
{
    AdvantageResponse response = null;
    //getting XML Response from Map Quest
    HttpClient client = new HttpClient() ;
    PostMethod method = null;
    try {
        // This URL is set after the setMapQuestRequestURL
        method = new PostMethod(mapQuestURL);
        //configuring MapQuest Communication client);
        String envVariable = System.getProperty ("bs.env.type");
        envVariable = "dev";
        if (envVariable.equals("dev")) {
            HostConfiguration hostConfig = new HostConfiguration();
            // SENDING THE MESSAGE AND GETTING THE RESPONSE
            client.setHostConfiguration(hostConfig);
        } //Un Marshal the response XML
        int statusCode = client.executeMethod (method);
        if (statusCode == 200) {
            String responseBody = method.getResponseBodyAsString() ;
            StringReader reader = new StringReader (responseBody);
            IUnmarshallingContext unmctx =
                BindingDirectory.getFactory(SearchResponse.class).createUnmarshallingContext();
            response =(AdvantageResponse) unmctx.unmarshalDocument (reader, null) ;
        }
    }
    catch (Exception e){ return null; }
    finally { if (method != null) method.releaseConnection(); }
    //Because MapQuest can not return more then 20 search results at a time,
    //Therefore we manually get them.
    try {
        //check for more results
        if ( !response.getSearch().getNextData().equals(null) ||
            !response.getSearch().getNextData().equals("") ) {
            AdvantageResponse advantageResponse = getMapQuestAdvantageResponse();
            if (advantageResponse.getSearch().getSearchResults().sizeLocationList()>0) {

                for (Iterator itr= advantageResponse.getSearch().getSearchResults()
                    .getLocationList().iterator(); itr.hasNext();) {
                    Location location = (Location) iter.next();
                    response.getSearch().getSearchResults().addLocation(location);
                }
            }
        }
    }
}

```



```
    }  
    }  
    }catch (Exception e) {}  
    return response ;  
}
```

After receiving all Location in AdvantageResponse we would render those locations over our StoreList.jsp. When a user clicks on particular record s/he will be redirected to StoreDetail.jsp page which renders the Map image by getting the Map URL associated with that particular location by calling getMap() that in essence provide a URL that would be placed over the .jsp pages required to render results.

Conclusion

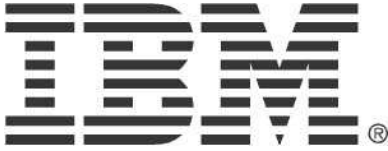
This article was an attempt to provide the concept for the building a store locator module. As a result, we had come up with a flexible and extensible application that can be used in a number of ways within a wide range of B2C application domains. The intent of this project was to provide a bridge between the WebSphere Commerce and Mapping services available to for a Store Locator. Since both platforms share a common premise of location navigation. By using this premise, our application not only provides the mapping services for particular store client but also enhances business perspective and more personal shopping experience and boosts the requirements of transportation and shopping decision. With respect to our original goal we have built and demonstrated the Store Locator service as a off the shelf module. As a first step for assessing the feasibility of our goal we have provided the mapping service using MapQuest successfully. This application serves the purpose of our endeavour and shows how a request can be handled and depicted over WebSphere Commerce.

References

[1]<http://company.mapquest.com/mqbs/4.html>

[2]<http://www-06.ibm.com/software/info1/websphere/index.jsp?tab=products/commerce>

[3] <http://jibx.sourceforge.net/>



© Copyright IBM Corporation 2010
IBM Global Services
Route 100
Somers, NY 10589
U.S.A.
Produced in the United States of America
08-10
All Rights Reserved

IBM, the IBM logo, ibm.com, Lotus®, Rational®, Tivoli®, DB2® and WebSphere® are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. 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 "Copyright and trademark information" at ibm.com/legal/copytrade.shtml Other company, product and service names may be trademarks or service marks of others. The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided "as is" without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software. This document illustrates how one organization uses IBM products. Many factors have contributed to the results and benefits described; IBM does not guarantee comparable results elsewhere.