

The background of the page is white with several abstract, overlapping shapes and lines. There are several thin, curved lines in shades of blue and grey that sweep across the page. At the bottom, there is a large, layered graphic consisting of various shades of blue and grey, creating a sense of depth and movement.

DB2 Web Query
RESTful Web Services
Developer's Guide

Active Technologies, EDA, EDA/SQL, FIDEL, FOCUS, Information Builders, the Information Builders logo, iWay, iWay Software, Parlay, PC/FOCUS, RStat, Table Talk, Web390, WebFOCUS, WebFOCUS Active Technologies, and WebFOCUS Magnify are registered trademarks, and DataMigrator and Hyperstage are trademarks of Information Builders, Inc.

Adobe, the Adobe logo, Acrobat, Adobe Reader, Flash, Adobe Flash Builder, Flex, and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Due to the nature of this material, this document refers to numerous hardware and software products by their trademarks. In most, if not all cases, these designations are claimed as trademarks or registered trademarks by their respective companies. It is not this publisher's intent to use any of these names generically. The reader is therefore cautioned to investigate all claimed trademark rights before using any of these names other than to refer to the product described.

Copyright © 2015 , by Information Builders, Inc. and iWay Software. All rights reserved. Patent Pending. This manual, or parts thereof, may not be reproduced in any form without the written permission of Information Builders, Inc.

Contents

- Preface.....5**
 - Documentation Conventions.....6
- 1. Introducing DB2 Web Query RESTful Web Services.....7**
 - What Is REST?.....8
 - What are RESTful Web Services?.....8
- 2. DB2 Web Query Authentication and Reporting RESTful Web Service Requests.....9**
 - Authenticating DB2 Web Query Sign-On Requests.....10
 - DB2 Web Query Reporting.....12
 - Listing Folders and Subfolders.....12
 - Listing the Parameters for a Report Within DB2 Web Query.....19
 - Running a Report From DB2 Web Query.....23
- 3. Web Query Reporting Server RESTful Web Service Requests.....27**
 - Listing Applications.....28
 - Listing Files Within an Application.....29
 - Listing the Parameters for a Data Flow Within an Application.....32
 - Running a Data Flow Within an Application.....34
- 4. DB2 Web Query Report Broker RESTful Web Service Requests.....37**
 - Running a Schedule.....38
 - Retrieving a Schedule.....38
- A. Alternative Method of Calling DB2 Web Query RESTful Web Service Requests.....43**
 - Calling DB2 Web Query RESTful Web Service Requests.....44
- B. Visual Basic .NET and Java Code Examples.....45**
 - Signing On to DB2 Web Query.....46
 - Visual Basic .NET Example.....46

Java Example.....	47
Listing Folders From DB2 Web Query.....	47
Visual Basic .NET Example.....	48
Java Example.....	49
Running a DB2 Web Query Report.....	50
Visual Basic .NET Example.....	50
Java Example.....	51
Handling Drill Downs, Active Cache, and On-Demand Paging Reports.....	52
Visual Basic .NET Example (signOn.aspx).....	53
Visual Basic .NET Example (WebForm2.aspx).....	55

Preface

This documentation describes how to create RESTful Web Services for DB2 Web Query features. It is intended for experienced developers who will use this capability to expose DB2 Web Query functionality as callable services from a Microsoft Visual Studio .NET or J2EE development platform. Developers should have a knowledge of RESTful web service technology and object oriented programming.

How This Manual Is Organized

This manual includes the following chapters:

	Chapter/Appendix	Contents
1	Introducing DB2 Web Query RESTful Web Services	Provides an introduction to REST and RESTful web services in the context of DB2 Web Query.
2	DB2 Web Query Authentication and Reporting RESTful Web Service Requests	Describes the format and structure of authentication RESTful web service requests.
3	Web Query Reporting Server RESTful Web Service Requests	Describes the format and structure of Web Query Reporting Server RESTful web service requests.
4	DB2 Web Query Report Broker RESTful Web Service Requests	Describes the format and structure of DB2 Web Query Report Broker RESTful web service requests.
A	Alternative Method of Calling DB2 Web Query RESTful Web Service Requests	Describes an alternative method of calling DB2 Web Query RESTful web service requests.

	Chapter/Appendix	Contents
B	Visual Basic .NET and Java Code Examples	Provides Visual Basic .NET and Java code examples on how to create DB2 Web Query RESTful web service requests.

Documentation Conventions

The following table describes the documentation conventions that are used in this manual.

Convention	Description
<code>THIS TYPEFACE</code> or <code>this typeface</code>	Denotes syntax that you must enter exactly as shown.
<i>this typeface</i>	Represents a placeholder (or variable), a cross-reference, or an important term.
<u>underscore</u>	Indicates a default setting.
Key + Key	Indicates keys that you must press simultaneously.
{ }	Indicates two or three choices. Type one of them, not the braces.
[]	Indicates a group of optional parameters. None is required, but you may select one of them. Type only the parameter in the brackets, not the brackets.
	Separates mutually exclusive choices in syntax. Type one of them, not the symbol.
...	Indicates that you can enter a parameter multiple times. Type only the parameter, not the ellipsis (...).
.	Indicates that there are (or could be) intervening or additional commands.

1 Introducing DB2 Web Query RESTful Web Services

This section provides an introduction to REST and RESTful web services in the context of DB2 Web Query.

Topics:

- ❑ What Is REST?
- ❑ What are RESTful Web Services?

What Is REST?

The REST architectural style was developed in parallel with HTTP Version 1.1, based on the existing design of HTTP Version 1.0. The largest implementation of a system conforming to the REST architectural style is the World Wide Web. REST exemplifies how the architecture of the web emerged by characterizing and constraining the macro-interactions of the four components of the web, namely origin servers, gateways, proxies and clients, without imposing limitations on the individual participants. As such, REST essentially governs the proper behavior of participants.

REST-style architectures consist of clients and servers. Clients initiate requests to servers, servers process requests and return appropriate responses. Requests and responses are built around the transfer of representations of resources. A resource can be essentially any coherent and meaningful concept that may be addressed. A representation of a resource is typically a document that captures the current or intended state of a resource.

The client begins sending requests when it is ready to make the transition to a new state. While one or more requests are outstanding, the client is considered to be in transition. The representation of each application state contains links that may be used the next time the client chooses to initiate a new state transition.

REST facilitates the transaction between web servers by allowing loose coupling between different services. REST is less strongly typed than its counterpart, SOAP. The REST language is based on the use of nouns and verbs, and has an emphasis on readability. Unlike SOAP, REST does not require XML parsing and does not require a message header to and from a service provider. This ultimately uses less bandwidth. REST error handling is also different from that used by SOAP.

What are RESTful Web Services?

A RESTful web service (also called a RESTful web API) is a web service that is implemented using HTTP and the principles of REST. It is a collection of resources with four defined aspects:

- ❑ Base URL for the web service, such as:
<http://example.com/resources>
- ❑ Internet media type of the data supported by the web service. This is usually XML, but can be any other valid Internet media type providing that it is a valid hypertext standard.
- ❑ Set of operations supported by the web service using HTTP methods (for example, GET, PUT, POST, or DELETE).
- ❑ The API must be hypertext driven.

2 | DB2 Web Query Authentication and Reporting RESTful Web Service Requests

This section describes the format and structure of authentication RESTful web service requests.

Topics:

- ❑ Authenticating DB2 Web Query Sign-On Requests
- ❑ DB2 Web Query Reporting

Authenticating DB2 Web Query Sign-On Requests

This RESTful web service request can be used to authenticate DB2 Web Query sign-on requests. The XML response that is returned indicates whether the authentication was successful or unsuccessful. The XML response also includes a WQLWI80_Cookie, which exists within the HTTP header. All future DB2 Web Query RESTful web services requests must have the WQLWI80_Cookie in the HTTP header. If an application is required to interact with specific DB2 Web Query components, then the WQLWI80_Cookie is also used when sending the HTTP request to open the component. This eliminates the need to reauthenticate to DB2 Web Query. In addition, if you are already signed on to the DB2 Web Query portal, you are not required to run this sign-on request. The WQLWI80_Cookie is returned in the HTTP header after a successful sign-on attempt is made.

HTTP Method: POST

REST URL Format:

`http://host:port/webquery/rs/ibfs`

where:

host

Is the name of the system where DB2 Web Query is installed.

port

Is the port number used by DB2 Web Query.

Body Format:

`IBIRS_action=signOn&IBIRS_userName=Userid&IBIRS_password=Password`

where:

Userid

Is the user ID that is required to authenticate to DB2 Web Query.

Password

Is the password that is required to authenticate to DB2 Web Query.

Example:

In the following example, a sign-on attempt is made to DB2 Web Query with a user ID value of *admin* and a password value of *admin*.

Post Request URL:

`http://localhost:12331/webquery/rs/ibfs`

Body:

`IBIRS_action=signOn&IBIRS_userName=admin&IBIRS_password=admin`

Response:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ibfsrpc _jt="IBFSResponseObject" language="EN" name="IBIRS_action" returncode="10000"

  returndesc="SUCCESS" subreturncode="0" subsystem="SSYS" type="simple">
  <ibfsparams size="0"/>
  <rootObject _jt="IBFSUserObject" description="" dummy="false" email=""
fullPath="IBFS:/SSYS/USERS/admin" name="admin" password="" type="User">
    <status _jt="IBSSUserStatus" name="UNDEFINED"/>
    <groups _jt="ArrayList" size="0"/>
  </rootObject>
</ibfsrpc>
```

If the value for the *returncode* attribute in the XML response is 10000, then the sign-on attempt to DB2 Web Query was successful.

The following is a sample response trace from an authentication request:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
X-XSS-protection: 0
Set-Cookie: WQLWI80_Cookie=BD61C838569C30474977ACDE3DAD8F54;
Path=/webquery/; HttpOnly
Expires: Mon, 24 Sep 2012 09:12:48 GMT
Cache-Control: private
Set-Cookie: WF_SESSIONID=1932062683094412614; Path=/
IBI_Messages: 2
IBI_Message1: (IBFS10000) SUCCESS
IBI_Message2: <IBIWF_SES_AUTH_TOKEN>=<null>
Content-Type: text/xml;charset=iso-8859-1
Transfer-Encoding: chunked
Date: Mon, 24 Sep 2012 09:07:48 GMT

205
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ibfsrpc _jt="IBFSResponseObject" language="EN" name="signOn"
returncode="10000"
  returndesc="SUCCESS" subreturncode="0" subsystem="SSYS"
type="simple">
  <ibfsparams size="0"/>
  <rootObject _jt="IBFSUserObject" description="" dummy="false" email=""
    fullPath="IBFS:/SSYS/USERS/admin" name="admin" password=""
    rsPath="/webquery/rs/ibfs/SSYS/USERS/admin" type="User">
    <status _jt="IBSSUserStatus" name="UNDEFINED"/>
    <groups _jt="ArrayList" size="0"/>
  </rootObject>
</ibfsrpc>
```

The following is a sample trace of a subsequent request:

```
GET http://localhost:12331/webquery/rs/ibfs/WFC/Repository?IBIRS_action=get
HTTP/1.1
Host: localhost:12331
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:15.0) Gecko/20100101
Firefox/15.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Cookie: WQLWI80_Cookie=BD61C838569C30474977ACDE3DAD8F54;
wcNewPreference=1963156A6FD0D3C6EE81F2C992ED527D;
WF_SESSIONID=1932062683094412614
```

DB2 Web Query Reporting

In this section:

Listing Folders and Subfolders

Listing the Parameters for a Report Within DB2 Web Query

Running a Report From DB2 Web Query

This section describes the format and structure of RESTful web service requests that are used for a variety of DB2 Web Query reporting tasks.

Listing Folders and Subfolders

This RESTful web service request can be used to retrieve a list of folders and subfolders within DB2 Web Query reporting.

HTTP Method: GET

REST URL Format:

```
http://host:port/webquery/rs/ibfs/WFC/Repository/FolderName?IBIRS_action=get
```

where:

host

Is the name of the system where DB2 Web Query is installed.

port

Is the port number used by DB2 Web Query.

FolderName

Is the name of the folder to be used in retrieving a list of its subfolders. To obtain a list of folders, *FolderName* should not be included in the REST URL. To obtain additional levels of subfolders for a particular subfolder, the path to the subfolder name must be included in the REST URL. For example, ParentFolderName/FolderName.

Example 1:

In the following example, a list of folders is retrieved.

Request:

http://localhost:12331/webquery/rs/ibfs/WFC/Repository?IBIRS_action=get

Response:

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ibfsrpc _jt="IBFSResponseObject" language="EN" name="IBIRS_action"
returncode="10000"
  returndesc="SUCCESS" subreturncode="0" subsystem="SSYS" type="simple">
<ibfsparams size="0"/>
<rootObject _jt="IBFSMRObject" binary="false" container="true"
createdBy="Web Query"
  createdOn="1344536982043" defaultLng="en_US" description="Content"
dummy="false"
  effectiveRSName="EDASERVE" expireDate="1344536982047" externalId=""
  fullPath="IBFS:/WFC/Repository" handle="000000000001"
lastModified="1344536982047"
  lastaccessBy="admin" lastaccessOn="1345146849357" lastmodBy="Web Query"
length="0"
  name="Repository" policy="///+f////////9//////////+AAAAA"
returnedLng="en_US"
  summary="Content Root" type="MRRepository">
<children _jt="ArrayList" size="3">
  <item _jt="IBFSMRObject" binary="false" container="true" createdBy="Web
Query"
    createdOn="1344536982083" defaultLng="en_US" description="Public"
dummy="false"
    effectiveRSName="EDASERVE" expireDate="1344536982083" externalId=""
    fullPath="IBFS:/WFC/Repository/Public" handle="000000000004" index="0"

    lastModified="1344536982083" lastaccessBy="admin"
lastaccessOn="1344957209010"
    lastmodBy="Web Query" length="0" name="Public" parent="Repository"
    policy="///+f////////9//////////+AAAAA" returnedLng="en_US"
    summary="Public Folder" type="MRFolder">
<nlsValues _jt="HashMap" loadFactor="0.75" threshold="12">
  <entry>
    <key _jt="string" value="en_US"/>
    <value _jt="ArrayList" size="2">
      <item _jt="string" index="0" value="Public"/>
      <item _jt="string" index="1" value="Public Folder"/>
    </value>
  </entry>
</nlsValues>
<properties size="0"/>
</item>
  <item _jt="IBFSMRObject" binary="false" container="true"
createdBy="admin"
    createdOn="1345146734216" defaultLng="en_US" description="SEC Filings"
dummy="false"
    effectiveRSName="EDASERVE"

```

```

fullPath="IBFS:/WFC/Repository/Financial_Reports"
  handle="75d099c0_163a_46d8_ba25_ec0be965b15d" index="1"
lastModified="1345146734216"
  lastaccessBy="admin" lastaccessOn="1345146755132" lastmodBy="admin"
  length="0"
  name="Financial_Reports" ownerId="10001" ownerName="admin"
ownerType="U"
  parent="Repository" policy="//v+f////////9//////////+AAAAA"
  returnedLng="en_US"
  summary="Quarterly and Yearly Financial Reports reported to the
Securities and Exchange Commission"
  type="MRFolder"> <nlsValues _jt="HashMap" loadFactor="0.75"
threshold="12">
  <entry>
  <key _jt="string" value="en_US"/>
  <value _jt="ArrayList" size="2">
    <item _jt="string" index="0" value="SEC Filings"/>
    <item _jt="string" index="1"
      value="Quarterly and Yearly Financial Reports reported to the
Securities and Exchange Commission"/>
  </value>
  </entry>
  </nlsValues>
  <properties size="0"/>
  </item>
  <item _jt="IBFSMRObject" binary="false" container="true"
createdBy="admin" createdOn="1344607303673"
  defaultLng="en_US" description="RESTful Web Services" dummy="false"
  effectiveRSName="EDASERVE"
  fullPath="IBFS:/WFC/Repository/RESTful_Web_Services"
handle="ac08f200_d2f2_4ab6_9b60_b62d8f2ad345"
  index="2" lastModified="1344957300737" lastaccessBy="admin"
lastaccessOn="1345146071751"
  lastmodBy="admin" length="0" name="RESTful_Web_Services"
ownerId="10001" ownerName="admin"
ownerType="U" parent="Repository"
policy="//v+f////////9//////////+AAAAA" returnedLng="en_US"
  summary="For documenting RESTful Web Services" type="MRFolder">
  <nlsValues _jt="HashMap" loadFactor="0.75" threshold="12">
  <entry>
  <key _jt="string" value="en_US"/>
  <value _jt="ArrayList" size="2">
    <item _jt="string" index="0" value="RESTful Web Services"/>
    <item _jt="string" index="1" value="For documenting RESTful Web
Services"/>
  </value>
  </entry>
  </nlsValues>

```

```

    <properties size="0"/>
  </item>
</children>
<nlsValues _jt="HashMap" loadFactor="0.75" threshold="12">
  <entry>
    <key _jt="string" value="en_US"/>
    <value _jt="ArrayList" size="2">
      <item _jt="string" index="0" value="Content"/>
      <item _jt="string" index="1" value="Content Root"/>
    </value>
  </entry>
</nlsValues>
<properties size="0"/>
</rootObject>
</ibfsrpc>

```

Each folder definition is defined within the opening and closing *item* tag. The *type* attribute for a folder is *MRFolder*. The *name* attribute defines the name of the folder. The *description* attribute defines the title for the folder. The *summary* attribute defines a brief description for the contents of the folder.

In this example, there are three folders, as listed in the following table.

Folder Name	Title	Summary
Public	Public	Public Folder.
RESTful_Web_Services	RESTful Web Services	For documenting RESTful Web Services.
Financial_Reports	SEC Filings	Quarterly and Yearly Financial Reports reported to the Securities and Exchange Commission.

Example 2:

In the following example, a list of subfolders for the SEC Filings (Financial_Reports) folder is retrieved.

Request:

http://localhost:12331/webquery/rs/ibfs/WFC/Repository/Financial_Reports?IBIRS_action=get

Response:

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ibfsrpc _jt="IBFSResponseObject" language="EN" name="IBIRS_action"
returncode="10000"
  returndesc="SUCCESS" subreturncode="0" subsystem="SSYS" type="simple">
<ibfsparams size="0"/>
<rootObject _jt="IBFSMRObject" binary="false" container="true"
createdBy="admin"
  createdOn="1345146734216" defaultLng="en_US" description="SEC Filings"
  dummy="false"
  effectiveRSName="EDASERVE"
fullPath="IBFS:/WFC/Repository/Financial_Reports"
  handle="75d099c0_163a_46d8_ba25_ec0be965b15d"
lastModified="1345146734216"
  lastaccessBy="admin" lastaccessOn="1345147040831" lastmodBy="admin"
length="0"
  name="Financial_Reports" ownerId="10001" ownerName="admin" ownerType="U"

  policy="//v+f////////9//////////+AAAAA" returnedLng="en_US"
  summary="Quarterly and Yearly Financial Reports reported to the
Securities and Exchange Commission"
  type="MRFolder">
  <children _jt="ArrayList" size="1">
    <item _jt="IBFSMRObject" binary="false" container="true"
createdBy="admin" createdOn="1345147005204"
  defaultLng="en_US" description="Quarterly" dummy="false"
effectiveRSName="EDASERVE"
  fullPath="IBFS:/WFC/Repository/Financial_Reports/Quarterly"
handle="a0cfcde1_fb34_4b07_b20d_4144094ec5c2"
  index="0" inheritedPrivacy="true" lastModified="1345147005204"
lastaccessBy="admin"
  lastaccessOn="1345147013034" lastmodBy="admin" length="0"
name="Quarterly" ownerId="10001"
  ownerName="admin" ownerType="U" parent="Financial_Reports"
policy="//v+f////////f9////////9//////////+AAAAA"
  returnedLng="en_US"
  summary="Quarterly Financial Reports reported to the Securities and
Exchange Commission" type="MRFolder">

```

```

<nlsValues _jt="HashMap" loadFactor="0.75" threshold="12">
  <entry>
    <key _jt="string" value="en_US"/>
    <value _jt="ArrayList" size="2">
      <item _jt="string" index="0" value="Quarterly"/>
      <item _jt="string" index="1"
        value="Quarterly and Yearly Financial Reports reported to the
        Securities and Exchange Commission"/>
    </value>
  </entry>
</nlsValues>
<properties size="0"/>
</item>
</children>
<nlsValues _jt="HashMap" loadFactor="0.75" threshold="12">
  <entry>
    <key _jt="string" value="en_US"/>
    <value _jt="ArrayList" size="2">
      <item _jt="string" index="0" value="SEC Filings"/>
      <item _jt="string" index="1"
        value="Quarterly and Yearly Financial Reports reported to the
        Securities and Exchange Commission"/>
    </value>
  </entry>
</nlsValues>
<properties size="0"/>
</rootObject>
</ibfsrpc>

```

Each folder definition is defined within the opening and closing *item* tag. The *name* attribute defines the name of the folder. The *description* attribute defines the title for the folder. The *summary* attribute defines a brief description for the contents of the folder.

In this example, there is one subfolder, as listed in the following table.

Subfolder Name	Title	Summary
Quarterly	Quarterly	Quarterly Financial Reports reported to the Securities and Exchange Commission.

Listing the Parameters for a Report Within DB2 Web Query

This RESTful web service request can be used to retrieve the current parameters for a DB2 Web Query report.

HTTP Method: GET

REST URL Format:

http://host:port/webquery/rs/ibfs/WFC/Repository/FolderName/FexName?IBIRS_action=describeFex

where:

host

Is the name of the system where DB2 Web Query is installed.

port

Is the port number used by DB2 Web Query.

FolderName

Is the the name of the folder used for the stored DB2 Web Query report. If the folder used for the stored DB2 Web Query report exists as a subfolder, then the path to the subfolder name must be included in the REST URL. For example, TopFolderName/SubFolderName.

FexName

Is the name of the DB2 Web Query report as defined in the *name* attribute when listing the content of a folder.

Example:

In the following example, the current parameters for the Sales_for_a_Specific_Country.fex, which exists in the Car_Reports folder, is retrieved. The Car_Reports folder is a subfolder of the RESTful_Web_Services folder. Sales_for_a_Specific_Country.fex is defined in the *name* attribute when listing the content of a folder.

Request:

http://localhost:12331/webquery/rs/ibfs/WFC/Repository/RESTful_Web_Services/Car_Reports/Sales_for_a_Specific_Country.fex?IBIRS_action=describeFex

Response:

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ibfsrpc _jt="IBFSResponseObject" language="EN" name="IBIRS_action" returncode="10000"

  returndesc="SUCCESS" subreturncode="0" subsystem="SSYS" type="simple">
<ibfsparams size="0"/>
<rootObject class="com.ibi.wfrs.IBFSWFDescribe" isSavedParam="false">
  <bindingInfo _jt="HashMap" loadFactor="0.75" threshold="24">
    <entry>
      <key _jt="string" value="SUBSYSTEM"/>
      <value class="com.ibi.wfrs.BindingVar" isReqParm="false" value="Self Service"/>
    </entry>
    <entry>
      <key _jt="string" value="IBI_WF_charset"/>
      <value class="com.ibi.wfrs.BindingVar" isReqParm="false" value="windows-1252"/>
    </entry>
    <entry>
      <key _jt="string" value="IBI_Webapp_Context_Default"/>
      <value class="com.ibi.wfrs.BindingVar" isReqParm="false" value="/webquery"/>
    </entry>
    <entry>
      <key _jt="string" value="SCRIPT_NAME"/>
      <value class="com.ibi.wfrs.BindingVar" isReqParm="false"
value="/webquery/WFServlet"/>
    </entry>
    <entry>
      <key _jt="string" value="IBFS1_action"/>
      <value class="com.ibi.wfrs.BindingVar" isReqParm="true" value="runItem"/>
    </entry>
    .
    .
    .
    <entry>
      <key _jt="string" value="SAVE_PARMRPT"/>
      <value class="com.ibi.wfrs.BindingVar" isReqParm="false"
value="IBFS:/WFC/Repository/RESTful_Web_Services/Car_Reports/Sales_for_a_Specific_Country.fex"/>
    </entry>
  </bindingInfo>
  <amperMap accessOrder="false" class="java.util.LinkedHashMap" loadFactor="0.75"
threshold="12">

```

```

<entry>
  <key _jt="string" value="FOCFOCEXEC"/>
  <value class="com.ibi.wfrs.AmperVar" format="" max="0.0" min="0.0"
name="FOCFOCEXEC" strDef="">
    <type class="com.ibi.wfrs.IBFSAmperVarType" name="system"/>
    <displayType class="com.ibi.wfrs.IBFSAmperDisplayType" name="prompt"/>

    <values accessOrder="false" class="java.util.LinkedHashMap"
loadFactor="0.75" threshold="12"/>
  </value>
</entry>
<entry>
  <key _jt="string" value="FOCEXURL"/>
  <value class="com.ibi.wfrs.AmperVar" format="" max="0.0" min="0.0"
name="FOCEXURL" strDef="">
    <type class="com.ibi.wfrs.IBFSAmperVarType" name="set"/>
    <displayType class="com.ibi.wfrs.IBFSAmperDisplayType" name="prompt"/>

    <values accessOrder="false" class="java.util.LinkedHashMap"
loadFactor="0.75" threshold="12"/>
  </value>
</entry>
<entry>
  <key _jt="string" value="FOHTMLURL"/>
  <value class="com.ibi.wfrs.AmperVar" format="" max="0.0" min="0.0"
name="FOHTMLURL" strDef="">
    <type class="com.ibi.wfrs.IBFSAmperVarType" name="set"/>
    <displayType class="com.ibi.wfrs.IBFSAmperDisplayType" name="prompt"/>

    <values accessOrder="false" class="java.util.LinkedHashMap"
loadFactor="0.75" threshold="12"/>
  </value>
</entry>
<entry>
  <key _jt="string" value="GOOGLEMAPSAPIKEY"/>
  <value class="com.ibi.wfrs.AmperVar" format="" max="0.0" min="0.0"
name="GOOGLEMAPSAPIKEY" strDef="">
    <type class="com.ibi.wfrs.IBFSAmperVarType" name="set"/>
    <displayType class="com.ibi.wfrs.IBFSAmperDisplayType" name="prompt"/>

    <values accessOrder="false" class="java.util.LinkedHashMap"
loadFactor="0.75" threshold="12"/>
  </value>
</entry>
<entry>
  <key _jt="string" value="FOCREL"/>
  <value class="com.ibi.wfrs.AmperVar" format="" max="0.0" min="0.0"
name="FOCREL" strDef="">

```

```

    <type class="com.ibi.wfrs.IBFSAmperVarType" name="system"/>
    <displayType class="com.ibi.wfrs.IBFSAmperDisplayType" name="prompt"/>

    <values accessOrder="false" class="java.util.LinkedHashMap"
loadFactor="0.75" threshold="12"/>
    </value>
  </entry>
  <entry>
    <key _jt="string" value="EXCELSERVURL"/>
    <value class="com.ibi.wfrs.AmperVar" format="" max="0.0" min="0.0"
name="EXCELSERVURL" strDef="">
      <type class="com.ibi.wfrs.IBFSAmperVarType" name="set"/>
      <displayType class="com.ibi.wfrs.IBFSAmperDisplayType" name="prompt"/>

      <values accessOrder="false" class="java.util.LinkedHashMap"
loadFactor="0.75" threshold="12"/>
      </value>
    </entry>
    <entry>
      <key _jt="string" value="COUNTRY"/>
      <value class="com.ibi.wfrs.AmperVar" description="Select Country:"
format="" max="0.0" min="0.0"
      name="COUNTRY" operation="" strDef="">
        <type class="com.ibi.wfrs.IBFSAmperVarType" name="unresolved"/>
        <displayType class="com.ibi.wfrs.IBFSAmperDisplayType"
name="staticType"/>
        <values accessOrder="false" class="java.util.LinkedHashMap"
loadFactor="0.75" threshold="12">
          <entry>
            <key _jt="string" value="ENGLAND"/>
            <value _jt="string" value="ENGLAND"/>
          </entry>
          <entry>
            <key _jt="string" value="JAPAN"/>
            <value _jt="string" value="JAPAN"/>
          </entry>
          <entry>
            <key _jt="string" value="FRANCE"/>
            <value _jt="string" value="FRANCE"/>
          </entry>
        </values>
      </value>
    </entry>
  </amperMap>
</rootObject>
</ibfsrpc>

```

Each parameter definition is defined within the opening and closing *entry* tag.

The XML response that is returned includes many system parameters along with the parameters defined in the DB2 Web Query report. Entries that have a *name* attribute for the *type* element of either *unresolved* or *defaultType* are the DB2 Web Query report parameters, as shown in the following example:

```
<type class="com.ibi.wfrs.IBFSAmperVarType" name="unresolved"/>
```

The *name* attribute within the *value* element defines the parameter that is being used in the selection, as shown in the following example:

```
<value class="com.ibi.wfrs.AmperVar" description="Select Country:" format="" max="0.0" min="0.0" name="COUNTRY" operation="" strDef="">
```

The *description* attribute within the *value* element defines the prompt title for the parameter.

If a parameter definition within a DB2 Web Query report has a list of valid values for the selection, additional *entry* elements will exist in the XML within the parameter definition. The *value* attribute within the *key* element would contain each valid value.

```
<entry><key _jt="string" value="ENGLAND"/><value _jt="string" value="ENGLAND"/></entry>
<entry><key _jt="string" value="JAPAN"/><value _jt="string" value="JAPAN"/></entry>
<entry><key _jt="string" value="FRANCE"/><value _jt="string" value="FRANCE"/></entry>
```

In this example, ENGLAND, JAPAN, and FRANCE are the valid values that can be passed to this parameter.

Running a Report From DB2 Web Query

This RESTful web service request can be used to run a report stored in the DB2 Web Query reporting repository.

HTTP Method: POST

REST URL Format:

```
http://host:port/webquery/rs/ibfs/WFC/Repository/FolderName/ReportName
```

where:

host

Is the name of the system where DB2 Web Query is installed.

port

Is the port number used by DB2 Web Query.

FolderName

Is the the name of the folder used for the stored DB2 Web Query report. If the folder used for the stored DB2 Web Query report exists as a subfolder, then the path to the subfolder name must be included in the REST URL. For example, TopFolderName/SubFolderName.

ReportName

Is the name of the DB2 Web Query report to run. It must include a .fex extension.

Body Format:

```
IBIRS_action=run&IBIRS_clientPath=clientPath&userName=Userid&password=Password&parmNameN=parmValueN
```

where:

clientPath

Is the path to the client application making the RESTful web service calls to DB2 Web Query. For example:

```
http://myapplication.maj.com/Sales/Monthly.aspx
```

The parameter is used when the initial DB2 Web Query report contains drill-down links or is defined as an On-Demand Paging report.

When you click on a drill-down link or pages in an On-Demand Paging report, the request will be routed to the client application, as defined by the *clientPath* value, instead of DB2 Web Query. All of the parameter names and values are sent with the request. The client application will then have to redirect the request to the following URL, which is the DB2 Web Query environment:

```
http://host:port/webquery/rs/ibfs
```

Userid

Is the Reporting Server user ID. If the Reporting Server is running with Security Off or the Reporting Server sign-on credentials are configured in the Reporting Server Client settings, then this paramater does not have to be sent in the REST request.

Password

Is the Reporting Server password. If the Reporting Server is running with Security Off or the Reporting Server sign-on credentials are configured in the Reporting Server Client settings, then this paramater does not have to be sent in the REST request.

parmNameN

Is the name of the defined parameter that will be passed to the Reporting Server.

Note: The number of defined parameters can vary and depend on the number of parameters within the DB2 Web Query report. For example, a DB2 Web Query report that requires two parameters will also require these parameters and corresponding values to be set in the body of this RESTful web service (&parmName1=parmValue1&parmName2=parmValue2). In a different DB2 Web Query report, there could be as many parameters as required (three, four, five, and so on).

parmValueN

Is the value of the defined parameter that will be passed to the Reporting Server.

Example:

In the following example, the Sales_for_a_Specific_Country report is being executed only for Japan.

Request:

http://localhost:12331/webquery/rs/ibfs/WFC/Repository/RESTful_Web_Services/Car_Reports/Sales_for_a_Specific_Country.fex

Body:

IBIRS_action=run&COUNTRY=JAPAN

Response:

The response is a report in either HTML, Excel, PDF, active report, or a graph.

3 | Web Query Reporting Server RESTful Web Service Requests

This section describes the format and structure of Web Query Reporting Server RESTful web service requests. These web service requests are available to support the IBM DB2 Web Query for i DataMigrator ETL Extension. DataMigrator for i data flows reside in an application directory on the Reporting Server. The web service requests found in this chapter can be used to manage data flows.

Topics:

- ❑ Listing Applications
- ❑ Listing Files Within an Application
- ❑ Listing the Parameters for a Data Flow Within an Application
- ❑ Running a Data Flow Within an Application

Listing Applications

This RESTful web service can be used to list the applications for a particular Reporting Server node.

HTTP Method: GET

REST URL Format:

`http://host:port/webquery/rs/ibfs/EDA/NodeName?IBIRS_action=get`

where:

host

Is the name of the system where Web Query is installed.

port

Is the port number used by Web Query.

NodeName

The name of the Reporting Server Node. The NodeName for Web Query is EDASERVE.

Example:

In the following example, the applications within the Web Query Reporting Server called EDASERVE are listed.

Request:

`http://localhost:12331/webquery/rs/ibfs/EDA/EDASERVE?IBIRS_action=get`

Response:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ibfsrpc _jt="IBFSResponseObject" language="EN" name="IBIRS_action" returncode="10000"

  returndesc="SUCCESS" subreturncode="0" subsystem="SSYS" type="simple">
<ibfsparams size="0"/>
<rootObject _jt="IBFSEDAObject" container="true" defaultNode="true" description=""

  dummy="false" fullPath="IBFS:/EDA/EDASERVE" host="REST-COMPUTER" name="EDASERVE"
  nodeClass="CLIENT" policy="///+f////////9//////////+AAAAA" port="12332"
type="EDANode">
  <children _jt="ArrayList" size="14">
    <item _jt="IBFSFolder" container="true" description="foccache" dummy="false"
      fullPath="IBFS:/EDA/EDASERVE/foccache" index="0" lastModified="1345560136000"
      name="foccache" parent="EDASERVE" policy="///+f////////9//////////+AAAAA"
type="IBFSFolder"/>
    .
    .
    .
    <item _jt="IBFSFolder" container="true" description="maintain" dummy="false"
      fullPath="IBFS:/EDA/EDASERVE/maintain" index="12" lastModified="1344546157000"
name="maintain"
      parent="EDASERVE" policy="///+f////////9//////////+AAAAA" type="IBFSFolder"/>

    <item _jt="IBFSFolder" container="true" description="session" dummy="false"
      fullPath="IBFS:/EDA/EDASERVE/session" index="13" lastModified="1344546157000"
name="session"
      parent="EDASERVE" policy="///+f////////9//////////+AAAAA" type="IBFSFolder"/>

  </children>
</rootObject>
</ibfsrpc>
```

Each application definition is defined within the opening and closing *item* tag. The *name* attribute defines the name of the application.

Listing Files Within an Application

This RESTful web service can be used to list all files within a particular application.

HTTP Method: GET

REST URL Format:

http://host:port/webquery/rs/ibfs/EDA/NodeName/AppName?IBIRS_action=get

where:

host

Is the name of the system where Web Query is installed.

port

Is the port number used by Web Query.

nodeName

Is the name of the Reporting Server Node. The nodeName for Web Query is EDASERVE..

appName

Is the name of the application containing the files to be listed. For more information, see [Listing Applications](#) on page 28.

Example:

In the following example, all files within the *ibisamp* application are listed.

Request:

http://localhost:12331/webquery/rs/ibfs/EDA/EDASERVE/ibisamp?IBIRS_action=get

Response:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ibfsrpc _jt="IBFSResponseObject" language="EN" name="IBIRS_action" returncode="10000"

  returndesc="SUCCESS" subreturncode="0" subsystem="SSYS" type="simple">
<ibfsparams size="0"/>
<rootObject _jt="IBFSFolder" container="true" description="ibisamp" dummy="false"
  fullPath="IBFS:/EDA/EDASERVE/ibisamp" lastModified="1345554266" name="ibisamp"
  policy="///+f////////9//////////+AAAAA" type="IBFSFolder">
<children _jt="ArrayList" size="182">
  <item _jt="IBFSFile" description="cargraph.fex" dummy="false"
    fullPath="IBFS:/EDA/EDASERVE/ibisamp/cargraph.fex" index="0"
lastModified="1328583952000"
    length="1471" name="cargraph.fex" parent="ibisamp"
policy="///+f////////9//////////+AAAAA"
    type="IBFSFile"/>
  <item _jt="IBFSFile" description="carinst.fex" dummy="false"
    fullPath="IBFS:/EDA/EDASERVE/ibisamp/carinst.fex" index="1"
lastModified="1328583952000"
    length="2624" name="carinst.fex" parent="ibisamp"
policy="///+f////////9//////////+AAAAA"
    type="IBFSFile"/>
  .
  .
  .
  <item _jt="IBFSFile" description="wfmstart.html" dummy="false"
    fullPath="IBFS:/EDA/EDASERVE/ibisamp/wfmstart.html" index="181"
lastModified="1328619018000"
    length="6364" name="wfmstart.html" parent="ibisamp"
policy="///+f////////9//////////+AAAAA"
    type="IBFSFile"/>
</children>
</rootObject>
</ibfsrpc>
```

Each file definition is defined within the opening and closing *item* tag. The *name* attribute defines the name of the file.

The following list shows the Web Query-specific file name extensions:

- fex.** Data Migrator for i data flow.
- mas.** Master File Description.
- acx.** Access File.
- fof.** FOCUS database.
- etg.** Data Migrator flow.

- ❑ **mnt.** Maintain procedure.
- ❑ **wfm.** Maintain forms.
- ❑ **fcm.** Maintain compiled.
- ❑ **ftm.** Flat file usually used as a temporary file.

The *description* attribute defines the description that was used as input for the file.

Listing the Parameters for a Data Flow Within an Application

This RESTful web service can be used to retrieve the current parameters for a DataMigrator for a data flow stored within an application.

HTTP Method: GET

REST URL Format:

`http://host:port/webquery/rs/ibfs/EDA/NodeName/Appname/FexName?IBIRS_action=describeFex`

where:

host

Is the name of the system where Web Query is installed.

port

Is the port number used by Web Query.

NodeName

Is the name of the Reporting Server Node. The NodeName for Web Query is EDASERVE.

Appname

Is the name of the application containing the files to be listed. For more information, see [Listing Applications](#) on page 28.

FexName

Is the name of the data flow as defined in the *name* attribute when listing files within an application. For more information, see [Listing Files Within an Application](#) on page 29.

Example:

In this example, the current parameters for the data flow `ordtest.fex`, which exists in the `dm_test2` application, are retrieved. The data flow `ordtest.fex` is defined in the *name* attribute when listing the files within an application. For more information, see [Listing Files Within an Application](#) on page 29.

Request:

`http://localhost:12331/webquery/rs/ibfs/EDA/EDASERVE/dm_test2/ordtest.fex?IBIRS_action=describeFex`

Response:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<ibfsrpc _jt="IBFSresponseObject" language="EN" name="describeFex" returncode="10000"
returnndesc="SUCCESS" subreturncode="0" subsystem="SSYS" type="simple">
<ibfsparams size="2">
<entry key="IBIRS_path" value="/EDA/EDASERVE/dm_test2/ordtest.fex" />
<entry key="IBIRS_args" value="__null" />
</ibfsparams>
.
.
.
<entry>
<key _jt="string" value="STARTAT" />
<value class="com.ibi.wfrs.AmperVar" format="" max="0.0" min="0.0" name="STARTAT"
strDef="0">
<type class="com.ibi.wfrs.IBFSAmperVarType" name="defaultType" />
<displayType class="com.ibi.wfrs.IBFSAmperDisplayType" name="prompt" />
<values accessOrder="false" class="java.util.LinkedHashMap" loadFactor="0.75"
threshold="12" />
</value>
</entry>
<entry>
<key _jt="string" value="STOPAT" />
<value class="com.ibi.wfrs.AmperVar" format="" max="0.0" min="0.0" name="STOPAT"
strDef="1000000000">
<type class="com.ibi.wfrs.IBFSAmperVarType" name="defaultType" />
<displayType class="com.ibi.wfrs.IBFSAmperDisplayType" name="prompt" />
<values accessOrder="false" class="java.util.LinkedHashMap" loadFactor="0.75"
threshold="12" />
</value>
</entry>
<entry>
<key _jt="string" value="PLANT" />
<value class="com.ibi.wfrs.AmperVar" format="" max="0.0" min="0.0" name="PLANT"
strDef="">
<type class="com.ibi.wfrs.IBFSAmperVarType" name="unresolved" />
<displayType class="com.ibi.wfrs.IBFSAmperDisplayType" name="prompt" />
<values accessOrder="false" class="java.util.LinkedHashMap" loadFactor="0.75"
threshold="12" />
</value>
</entry>
</amperMap>
</rootObject>
</ibfsrpc>

```

Each parameter definition is defined within the opening and closing *entry* tag.

The XML returned includes many system parameters along with the parameters defined in the data flow. Entries that have a *name* attribute for the *type* element of either *unresolved* or *defaultType* are the data flow parameters.

```
<type class="com.ibi.wfrs.IBFSAmperVarType" name="unresolved"/>
```

The *name* attribute within the *value* element defines the parameter that is being used in the selection.

```
<value class="com.ibi.wfrs.AmperVar" format="" max="0.0" min="0.0"
name="PLANT" strDef="">
<type class="com.ibi.wfrs.IBFSAmperVarType" name="unresolved" />
<displayType class="com.ibi.wfrs.IBFSAmperDisplayType" name="prompt" />
```

If a parameter definition within a data flow has a default value, the *strDef* attribute within the *value* element will contain that value.

If a parameter definition within a data flow has a Prompt title, the *description* attribute within the *value* element will contain the title.

If a parameter definition within a data flow has a list of valid values for the selection, additional *entry* elements will exist in the XML within the parameter definition. The *value* attribute within the *key* element would contain each valid value.

Running a Data Flow Within an Application

This RESTful web service can be used to run a DataMigrator for a data flow stored in an application.

HTTP Method: POST

REST URL Format:

```
http://host:port/webquery/rs/ibfs/EDA/NodeName/Appname/FexName
```

where:

host

Is the name of the system where Web Query is installed.

port

Is the port number used by Web Query.

NodeName

Is the name of the Reporting Server Node. The NodeName for Web Query is EDASERVE.

Appname

Is the name of the application containing the files to be listed. For more information, see [Listing Applications](#) on page 28.

FlexName

Is the name of the data flow as defined in the *name* attribute when listing files within an application. For more information, see [Listing Files Within an Application](#) on page 29.

Body Format:

`IBIRS_action=run&userName=Userid&password=Password&parmNameN=parmValueN&IBIRS_args=Object`

where:

Userid

Is the Reporting Server user ID.

Password

Is the Reporting Server password.

parmNameN

Is the name of the defined parameter that will be passed to the Reporting Server.

Note: The number of defined parameters can vary and depend on the number of parameters within the DataMigrator for i data flow. For example, a data flow that requires two parameters will also require these parameters and corresponding values to be set in the body of this RESTful web service (&parmName1=parmValue1&parmName2=parmValue2).

parmValueN

Is the value of the defined parameter that will be passed to the Reporting Server.

4 DB2 Web Query Report Broker RESTful Web Service Requests

This section describes the format and structure of DB2 Web Query Report Broker RESTful web service requests.

Topics:

- ❑ Running a Schedule
- ❑ Retrieving a Schedule

Running a Schedule

This RESTful web service request can be used to run an existing DB2 Web Query Report Broker Schedule.

HTTP Method: POST

REST URL Format:

`http://host:port/webquery/rs/ibfs/WFC/Repository/FolderName/ScheduleName?IBIRS_action=run`

where:

host

Is the name of the system where DB2 Web Query is installed.

port

Is the port number used by DB2 Web Query.

FolderName

Is the name of the folder used for the stored DB2 Web Query Report Broker Schedule. If the folder used for the stored DB2 Web Query Report Broker Schedule exists as a subfolder, then the path to the subfolder name must be included in the REST URL. For example, TopFolderName/SubFolderName.

ScheduleName

Is the name of the DB2 Web Query Report Broker Schedule to run, which also must have a .sch extension.

Example:

The following example demonstrates how to run a DB2 Web Query Report Broker Schedule called REST_Schedule.

POST Request URL:

`http://localhost:12331/webquery/rs/ibfs/WFC/Repository/RESTful_Web_Services/Car_Reports/REST_Schedule.sch?IBIRS_action=run`

Response:

A job number is returned in HTML format. For example:

`J453ce7a4je11bj48ffj832ej9053e5377495`

Retrieving a Schedule

This RESTful web service request can be used to retrieve an existing DB2 Web Query Report Broker Schedule.

HTTP Method: GET

REST URL Format:

`http://host:port/webquery/rs/ibfs/WFC/Repository/FolderName/ScheduleName?IBIRS_action=get`

where:

host

Is the name of the system where DB2 Web Query is installed.

port

Is the port number used by DB2 Web Query.

FolderName

Is the name of the folder used for the stored DB2 Web Query Report Broker Schedule. If the folder used for the stored DB2 Web Query Report Broker Schedule exists as a subfolder, then the path to the subfolder name must be included in the REST URL. For example, TopFolderName/SubFolderName.

ScheduleName

Is the name of the DB2 Web Query Report Broker Schedule to retrieve, which also must have a .sch extension.

Example:

In the following example, a schedule called REST_Schedule.sch is retrieved from the Car_Reports folder, which is within the RESTful_Web_Services folder.

Request:

`http://localhost:12331/webquery/rs/ibfs/WFC/Repository/RESTful_Web_Services/Car_Reports/REST_Schedule.sch?IBIRS_action=get`

Response:

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ibfsrpc _jt="IBFSResponseObject" language="EN" name="get"
returncode="10000" returndesc="SUCCESS" subreturncode="0" subsystem="SSYS"
  type="simple">
  <ibfsparams size="2">
    <entry key="IBIRS_args" value="__null"/>
    <entry key="IBIRS_"
value="/WFC/Repository/RESTful_Web_Services/Car_Reports/REST_Schedule.sch"/>

  </ibfsparams>
  <rootObject _jt="IBFSBrokerObject" binary="false" createdOn="1356718595487"
defaultLng="en_US" description="Schedule Created through REST"
dummy="false" effectiveRSName="EDASERVE" extension="sch"
externalId="S1995b2ecsa8f6s4096sa62es1867fa2d7a85"
fullPath="IBFS:/WFC/Repository/RESTful_Web_Services/Car_Reports/REST_Schedule.sch"
handle="7c2fd2a3I2dbcI400dIb666I3512e8d8b89f" lastModified="1356718595487"
lastaccessBy="admin" lastaccessOn="1356719962891" length="0"
name="REST_Schedule.sch" ownerId="10001" ownerName="admin" ownerType="U"
policy="//3/D///9+P9///v////////+AAAA=" returnedLng="en_US"
rsPath="/webquery/rs/ibfs/WFC/Repository/RESTful_Web_Services/Car_Reports/REST_Schedule.sch"
type="BrokerSchedule">
  <nlsValues _jt="HashMap" loadFactor="0.75" threshold="12">
    <entry>
      <key _jt="string" value="en_US"/>
      <value _jt="ArrayList" size="2">
        <item _jt="string" index="0" value="Schedule Created through REST"/>
      </value>
    </entry>
  </nlsValues>
  <properties size="2">
    <entry key="id" value="S1995b2ecsa8f6s4096sa62es1867fa2d7a85"/>
    <entry key="tool" value="schedule"/>
  </properties>
  <brokerObject _jt="BrokerSchedule" active="false" compressedReport="false"
deleteJobAfterRun="false" description="Schedule Created through REST"
destinationAddress="OWNER" ibfsId="7c2fd2a3I2dbcI400dIb666I3512e8d8b89f"
ibfsPath="IBFS:/WFC/Repository/RESTful_Web_Services/Car_Reports"
id="S1995b2ecsa8f6s4096sa62es1867fa2d7a85" name="REST_Schedule.sch"
nextRunTime="disabled" notification="INACTIVE" owner="admin"
policy="open,delete,rename,|,run,|,security;makeRules;viewRules"
priority="3" recurrence="0"
scheduleId="S1995b2ecsa8f6s4096sa62es1867fa2d7a85" scheduleTitle="Schedule
Created through REST" sendMethod="LIBRARY" statusLastExecuted=""
taskType="1" traceType="0">
    <notification _jt="BrokerScheduleNotification"

```



```

addressForBriefNotification="" addressForFullNotification="" description=""
  from="" id="" subject="" type="INACTIVE"/>
  <distributionList _jt="array" itemsClass="BrokerScheduleDistribution"
size="1">
    <item accessList="" accessListFullPath="" accessType="OWNER" category=""
class="ibi.broker.api.data.schedule.StorageLibrary"
compressionEnabled="false" counter="0" description="Report Library"
destinationIbfsId="c60b1f9a_05ef_4e72_a737_e869917607db"
destinationPath="IBFS:/WFC/Repository/RESTful_Web_Services/Car_Reports"
disabled="false" expirationData="1" expirationMode="N"
id="D58215579d4885d4b5eda023d9f44d1b4da01" index="0" type="LIBRARY"
valueonly="false">
      <storageLibraryEmail authEnabled="false" authPassword="" authUserId=""
class="ibi.broker.api.data.schedule.StorageLibraryEmail"
libraryURL="http://host:12331/webquery/library/report.rc" mailFrom=""
mailMessage="" mailReplyAddress="" mailServerName="ibismtp.ibi.com"
mailSubject="" sendEmailAfterSaveReport="false" sslEnabled="false"
tlsEnabled="false"/>
    </item>
  </distributionList>
  <timeInfoList _jt="array" itemsClass="BrokerScheduleTimeInfo" size="1">
    <item class="ibi.broker.api.data.schedule.TimeInfoOnce" description=""
disabled="false" id="Ifc777178i1ab0i42faibd06i81df82c234e7" index="0"
name="" type="0">
      <nextRunTime _jt="calendar" time="1355756400000"
timeZone="America/New_York"/>
      <startTime _jt="calendar" time="1355756400000"
timeZone="America/New_York"/>
    </item>
  </timeInfoList>
  <taskList _jt="array" itemsClass="BrokerScheduleTask" size="1">
    <item alertEnabled="false" burst="true"
class="ibi.broker.api.data.schedule.TaskStandardReport" description="Task
1" disabled="false" domainHREF="" execId="guest" execPassword=""
firstPostProcessingProcedure="" firstPreProcessingProcedure="" folderHREF=""
id="Tcdde20bdt3305t436ata200tecd3367ad16f" index="0"
procedureDescription="" procedureId="64e971c8_fd80_4d07_99a7_a2356743010b"

```

```
procedureName="IBFS:/WFC/Repository/RESTful_Web_Services/Car_Reports/Sales_Report_by_Country.fex"
  reportName="car_sales.htm" secondPostProcessingProcedure=""
  secondPreProcessingProcedure="" sendFormat="HTML" serverName="EDASERVE"
  type="1">
  <parameterList _jt="array" itemsClass="BrokerScheduleParameter"
  size="2">
    <item _jt="BrokerScheduleParameter" index="0" name="COUNTRY" type="0"
    useDefaultValue="false" value="ENGLAND"/>
    <item _jt="BrokerScheduleParameter" index="1" name="DEALER_COST"
    type="0" useDefaultValue="false" value="10000"/>
  </parameterList>
  </item>
</taskList>
  <lastTimeExecuted _jt="calendar" time="18000000"
  timeZone="America/New_York"/>
</brokerObject>
</rootObject>
</ibfsrpc>
```

A | Alternative Method of Calling DB2 Web Query RESTful Web Service Requests

This appendix describes an alternative method that can be used to call DB2 Web Query RESTful web service requests.

Topics:

- ❑ Calling DB2 Web Query RESTful Web Service Requests

Calling DB2 Web Query RESTful Web Service Requests

For each DB2 Web Query RESTful web service request, the portion of the URL path following *rs* can be represented as a parameter. *IBIRS_service* represents the parameter for the category and *IBIRS_path* represents the path to the specific functionality that is being performed.

Example:

In the following example, the REST URL for Example 1 from [Listing Folders and Subfolders](#) on page 12 shows the REST URL as:

```
http://localhost:12331/webquery/rs/ibfs/WFC/Repository?IBIRS_action=get
```

This request can also be sent as follows:

```
http://localhost:12331/webquery/rs?IBIRS_action=get&IBIRS_path=/WFC/Repository&IBIRS_service=ibfs
```

B Visual Basic .NET and Java Code Examples

This appendix provides Visual Basic .NET and Java code examples on how to create DB2 Web Query RESTful web service requests.

Topics:

- ❑ Signing On to DB2 Web Query
- ❑ Listing Folders From DB2 Web Query
- ❑ Running a DB2 Web Query Report
- ❑ Handling Drill Downs, Active Cache, and On-Demand Paging Reports

Signing On to DB2 Web Query

In this section:

Visual Basic .NET Example

Java Example

This section provides code examples that demonstrate how to sign on to DB2 Web Query.

Visual Basic .NET Example

```
Imports System.Net
Imports System.IO
Imports System.Text
Dim cookies As New CookieContainer
Dim webStream As Stream
Dim webResponse As String = ""
Dim request As HttpWebRequest
Dim response As HttpWebResponse
Dim postData As String
request = WebRequest.Create("http://localhost:12331/webquery/rs/ibfs")
request.Method = "POST"
postData = "IBIRS_action=signOn&IBIRS_userName=admin&IBIRS_password=admin"
request.CookieContainer = cookies
Dim byteArray As Byte() = Encoding.UTF8.GetBytes(postData)
request.ContentType = "application/x-www-form-urlencoded"
request.ContentLength = byteArray.Length
Dim dataStream As Stream = request.GetRequestStream()
dataStream.Write(byteArray, 0, byteArray.Length)
dataStream.Close()
response = request.GetResponse()
webStream = response.GetResponseStream()
Dim webStreamReader As New StreamReader(webStream)
While webStreamReader.Peek >= 0
    webResponse = webStreamReader.ReadToEnd()
End While
```

Java Example

```
import java.awt.Frame;
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import org.apache.commons.httpclient.*;
import org.apache.commons.httpclient.methods.*;
String request = "http://localhost:12331/webquery/rs/ibfs";
HttpClient client = new HttpClient();
PostMethod method = new PostMethod(request);

method.addParameter("IBIRS_action", "signOn");
method.addParameter("IBIRS_userName", "admin");
method.addParameter("IBIRS_password", "admin");

int statusCode = client.executeMethod(method);
Header[] cookies = null;
InputStream rstream = null;

rstream = method.getResponseBodyAsStream();
cookies = method.getResponseHeaders("Set-Cookie");

BufferedReader br = new BufferedReader(new InputStreamReader(rstream));
String line;
while ((line = br.readLine()) != null) {
    System.out.println(line);
}
br.close();
```

Listing Folders From DB2 Web Query

In this section:

Visual Basic .NET Example

Java Example

This section provides code examples that demonstrate how to retrieve a list of the top-level folders from DB2 Web Query. A successful sign-on request is a prerequisite for running this example, including retrieving the HTTP Header cookies from its response.

Visual Basic .NET Example

```
Imports System.Net
Imports System.IO
Imports System.Text
Dim request3 As HttpWebRequest
Dim response3 As HttpWebResponse
Dim webStream3 As Stream
Dim webResponse3 As String = ""
Dim tempfile As String
request3 = WebRequest.Create("http://localhost:12331/webquery/rs/ibfs/WFC/
Repository?IBIRS_action=get")
request3.Method = "GET"
'cookies is defined as CookieContainer in the Signing-In to DB2 Web Query example
request3.CookieContainer = cookies
response3 = request3.GetResponse()
webStream3 = response3.GetResponseStream()
Dim webStreamReader3 As New StreamReader(webStream3)
tempfile = "c:\temp\Folders.xml"
FileOpen(1, tempfile, OpenMode.Output)
While webStreamReader3.Peek >= 0
    webResponse3 = webStreamReader3.ReadToEnd()
    PrintLine(1, webResponse3)
End While
FileClose(1)
Dim xmlElem = XElement.Parse(webResponse3)
```


Java Example

```

import java.awt.Frame;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import org.apache.commons.httpclient.*;
import org.apache.commons.httpclient.methods.*;
String request3 =
"http://localhost:12331/webquery/rs/ibfs/WFC/Repository?IBIRS_action=get";

GetMethod method_getFolders = new GetMethod(request3);
// cookies is defined as Header[] in the Signing-In to DB2 Web Query example
for(int h=0; h<cookies.length; h++){
    method_getFolders.setRequestHeader(cookies[h].getName(), cookies[h].getValue());
}
// client is defined as HttpClient in the Signing-In to DB2 Web Query example
int statusCode3 = client.executeMethod(method_getFolders);

InputStream rstream3 = null;
rstream3 = method_getFolders.getResponseBodyAsStream();
File tempfile = new File("c:\\temp\\Folders.xml");
FileOutputStream fos = new FileOutputStream(tempfile);
PrintWriter out=new PrintWriter(fos);
BufferedReader br3 = new BufferedReader(new InputStreamReader(rstream3));
String line3;
String newOutput = null;
while ((line3 = br3.readLine()) != null) {
    newOutput = line3;
    out.println(newOutput);
    System.out.println(line3);
}
br3.close();
out.close();

```

Running a DB2 Web Query Report

In this section:

Visual Basic .NET Example

Java Example

This section provides code examples that demonstrate how to run the *Sales_for_a_Specific_Country* DB2 Web Query report, which resides in the *RESTful_Web_Services/Car_Reports* folder. A successful sign-on request is a prerequisite for running this example, including retrieving the HTTP Header cookies from its response.

Visual Basic .NET Example

```
Imports System.Net
Imports System.IO
Imports System.Text
Dim request2 As HttpWebRequest
Dim response2 As HttpWebResponse
Dim webStream2 As Stream
Dim webResponse2 As String = ""
request2 =
WebRequest.Create("http://localhost:12331/webquery/rs/ibfs/WFC/Repository/
RESTful_Web_Services/Car_Reports/Sales_for_a_Specific_Country.fex")
request2.Method = "POST"
'cookies is defined as CookieContainer in the Signing-In to DB2 Web Query example
request2.CookieContainer = cookies
postData = "IBIRS_action=run&COUNTRY=ENGLAND"
Dim byteArray2 As Byte() = Encoding.UTF8.GetBytes(postData)
request2.ContentType = "application/x-www-form-urlencoded"
request2.ContentLength = byteArray2.Length
Dim dataStream2 As Stream = request2.GetRequestStream()
dataStream2.Write(byteArray2, 0, byteArray2.Length)
dataStream2.Close()
response2 = request2.GetResponse()
webStream2 = response2.GetResponseStream()
Dim webStreamReader2 As New StreamReader(webStream2)
While webStreamReader2.Peek >= 0
    webResponse2 = webStreamReader2.ReadToEnd()
End While
WebBrowser1.DocumentText = webResponse2
```

Java Example

```

import java.awt.Frame;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import org.apache.commons.httpclient.*;
import org.apache.commons.httpclient.methods.*;
String request2 = "http://localhost:12331/webquery/rs/ibfs/WFC/Repository/RESTful_Web_
Services/Car_Reports/Sales_for_a_Specific_Country.fex";

PostMethod method_report = new PostMethod(request2);

method_report.addParameter("IBIRS_action", "run");
method_report.addParameter("COUNTRY", "ENGLAND");
// cookies is defined as Header[] in the Signing-in to DB2 Web Query example
for(int h=0; h<cookies.length; h++){
    System.out.println(cookies[h]);
    method_report.addRequestHeader(cookies[h].getName(), cookies[h].getValue());
}

// client is defined as HttpClient in the Signing-In to DB2 Web Query example
int statusCode2 = client.executeMethod(method_report);
InputStream rstream2 = null;

rstream2 = method_report.getResponseBodyAsStream();

File tempfile = new File("c:\\temp\\Report.htm");
FileOutputStream fos = new FileOutputStream(tempfile);
PrintWriter out=new PrintWriter(fos);
BufferedReader br2 = new BufferedReader(new InputStreamReader(rstream2));
String line2;
String newOutput = null;

while ((line2 = br2.readLine()) != null) {
    newOutput = line2;
    out.println(newOutput);
    System.out.println(line2);
}
br2.close();
out.close();

```

Handling Drill Downs, Active Cache, and On-Demand Paging Reports

In this section:

Visual Basic .NET Example (signOn.aspx)

Visual Basic .NET Example (WebForm2.aspx)

This section provides code examples that demonstrate how to run an On-Demand Paging report called ODP_Report.fex, which resides in the *RESTful_Web_Services/Car_Reports* folder.

The examples include:

- ❑ A signOn page, which is used to run the initial request.
- ❑ A WebForm2 page, which is used to make the additional RESTful web services requests required for the paging within the DB2 Web Query report.

The WebForm2 page can also be used as is to handle drill down and Active Cache paging requests.

The signOn page contains the RESTful web service request to run the initial DB2 Web Query report. The IBIRS_clientPath parameter is set so that all additional RESTful web services requests needed, whether paging, image retrieval, or paging will be routed through the client application. For example:

[IBIRS_clientPath=http://localhost:51970/WebForm2.aspx](http://localhost:51970/WebForm2.aspx)

Visual Basic .NET Example (signOn.aspx)

```

Imports System.Net
Imports System.IO
Public Class signOn
    Inherits System.Web.UI.Page
    Dim cookies As New CookieContainer
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
    Dim webStream As Stream
    Dim webResponse As String = ""
    Dim request As HttpWebRequest
    Dim response1 As HttpWebResponse
    Dim postData As String
    request = WebRequest.Create("http://localhost.:12331/webquery/rs/ibfs")

request.Method = "POST"
    postData = "IBIRS_action=signOn&IBIRS_userName=admin&IBIRS_password=admin"
    request.CookieContainer = cookies
    Dim byteArray As Byte() = Encoding.UTF8.GetBytes(postData)
    request.ContentType = "application/x-www-form-urlencoded"
    request.ContentLength = byteArray.Length
    Dim dataStream As Stream = request.GetRequestStream()
    dataStream.Write(byteArray, 0, byteArray.Length)
    dataStream.Close()
    response1 = request.GetResponse()
    webStream = response1.GetResponseStream()
    Dim request2 As HttpWebRequest
    Dim response2 As HttpWebResponse
    Dim webStream2 As Stream
    Dim webResponse2 As String = ""
    Dim uri As New System.Uri("http://localhost.:12331/webquery/rs")

    request2 = WebRequest.Create(uri)
    request2.Method = "POST"
    request2.CookieContainer = cookies
    postData = "IBIRS_action=run" + _
        "&IBIRS_clientPath=http://localhost:51970/WebForm2.aspx" + _
"&IBIRS_path=/WFC/Repository/RESTful_Web_Services/Car_Reports/ODP_Report.fex" + _
        "&IBIRS_service=ibfs" + _
        "&IBIRS_htmlPath=http://localhost:12331/webquery/ibi_html"

Dim byteArray2 As Byte() = Encoding.UTF8.GetBytes(postData)
    request2.ContentType = "application/x-www-form-urlencoded"

```

```
request2.ContentLength = byteArray2.Length
Dim dataStream2 As Stream = request2.GetRequestStream()
dataStream2.Write(byteArray2, 0, byteArray2.Length)
dataStream2.Close()
response2 = request2.GetResponse()
Dim i As Integer
Dim cookieArray As New CookieCollection
cookieArray = cookies.GetCookies(uri)
For i = 0 To cookies.Count - 1
    Dim aCookie As New HttpCookie(cookieArray(i).Name)
    aCookie.Value = cookieArray(i).Value
    Response.Cookies.Add(aCookie)
Next i
webStream2 = response2.GetResponseStream()
Dim webStreamReader2 As New StreamReader(webStream2)
While webStreamReader2.Peek >= 0
    webResponse2 = webStreamReader2.ReadToEnd()
End While
Response.Output.Write(webResponse2)
End Sub
End Class
```

Visual Basic .NET Example (WebForm2.aspx)

```

Imports System.Net
Imports System.IO
Public Class WebForm2
    Inherits System.Web.UI.Page
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
        Dim tDrillURL As String = ""
        Dim i As Integer
        Dim qParm As String
        Dim qValue As String
        Dim IBIRS_path As String = ""
        Dim Clicked_On As String = ""
        Dim cookies As New CookieContainer
        If Request.QueryString.AllKeys.Length = 0 Then
            For i = 0 To Request.Form.AllKeys.Length - 1
                qValue = Request.Form(Request.Form.AllKeys(i))
                If i = 0 Then
                    tDrillURL = tDrillURL + Request.Form.AllKeys(i) + "=" + qValue
                Else
                    tDrillURL = tDrillURL + "&" + Request.Form.AllKeys(i) + "=" +
qValue
                End If
            Next i
        Else
            For i = 0 To Request.QueryString.AllKeys.Length - 1
                qParm = Request.QueryString.AllKeys(i)
                qValue = Request.QueryString(Request.QueryString.AllKeys(i))
                If i = 0 Then
                    If Request.QueryString.AllKeys(i) <> Nothing Then
                        tDrillURL = tDrillURL + Request.QueryString.AllKeys(i) + "="
+ qValue
                    End If
                Else
                    tDrillURL = tDrillURL + "&" + Request.QueryString.AllKeys(i) +
"=" + qValue
                End If
            Next i
        End If
        Dim request3 As HttpWebRequest
        Dim response3 As HttpWebResponse
        Dim webStream3 As Stream
        Dim webResponse3 As String = ""
        Dim getData As String
        Dim uris As String = "http://localhost.:12331/webquery/rs/ibfs"

```

```
Dim uri As New System.Uri(uris)

getData = "http://localhost.:12331/webquery/rs/ibfs?" + _
          tDrillURL + _
          "&IBIRS_clientPath=http://localhost:51970/WebForm2.aspx" + _
          "&IBIRS_htmlPath=http://localhost:12331/webquery/ibi_html"
request3 = WebRequest.Create(getData)
request3.Method = "GET"

Dim j As Integer
For j = 0 To Request.Cookies.Count - 1
    Dim rCookie As New System.Net.Cookie
    rCookie.Name = Request.Cookies(j).Name
    rCookie.Value = Request.Cookies(j).Value
    cookies.Add(uri, rCookie)
    Dim aCookie As New HttpCookie(Request.Cookies(j).Name)
    aCookie.Value = Request.Cookies(j).Value
    Response.Cookies.Add(aCookie)
Next j
request3.CookieContainer = cookies
response3 = request3.GetResponse()
webStream3 = response3.GetResponseStream()
Dim binaryReader3 As New BinaryReader(webStream3)
Dim readData() As Byte = Nothing
Dim byteArray() As Byte = Nothing
Dim byteStart As Integer = 0
Dim byteLength As Integer
While (True)
    readData = binaryReader3.ReadBytes(4096)
    If (readData.Length = 0) Then
        Exit While
    End If
    byteLength = readData.Length
    ReDim Preserve byteArray(byteLength + byteStart - 1)
    Array.Copy(readData, 0, byteArray, byteStart, byteLength)
    byteStart = byteStart + byteLength
End While
Response.OutputStream.Write(byteArray, 0, byteArray.Length)

End Sub
End Class
```