

# iWay

Omni-IPF API Services  
Reference Guide  
Version 3.6

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 © 2019, 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

---

<b>Preface</b> .....	<b>7</b>
Documentation Conventions .....	7
Related Publications .....	8
Customer Support .....	8
Help Us to Serve You Better .....	9
User Feedback .....	11
iWay Software Training and Professional Services .....	11
<b>1. Omni-IPF Instance and Master Services</b> .....	<b>13</b>
Load Subject Instance .....	13
PUT: /server/api/v1/server/instance?mode=&timeout=.....	13
Example:.....	14
Get an Instance .....	15
GET: /server/api/v1/server/instance/{subject}/{sourceName}/{sourceInstanceId}.....	15
Example:.....	15
Get a Master .....	16
GET: /server/api/v1/server/master/{masterSubject}/{masterId}.....	16
Example:.....	16
Query for a Master .....	18
POST: /api/v1/server/master?responseType=oid masterId.....	18
Example:.....	18
<b>2. Omni-IPF Change Data Capture</b> .....	<b>21</b>
Subscribe to Changes .....	21
POST: /server/api/v1/server/cdc/subscribe.....	21
Example:.....	22
Cancel a Subscription .....	22
DELETE: /server/api/v1/server/cdc/subscribe/{subscriptionId}.....	22
Example:.....	22
Enable or Disable a Subscription .....	23
PUT: /server/api/v1/server/cdc/subscribe/{subscriptionId}?status=ENABLE DISABLE... ..	23
Example:.....	23
Get List of Subscription .....	24

GET: /server/api/v1/server/cdc/subscribe.....	24
Example:.....	24
Get Change Data Capture Process Status .....	24
GET: /server/api/v1/server/cdc/status.....	24
Example:.....	25
Start the Change Data Capture Process .....	25
GET: /server/api/v1/server/cdc/start?interval=<seconds>.....	25
Example:.....	25
Stop the Change Data Capture Process .....	26
GET: /server/api/v1/server/cdc/stop.....	26
Example:.....	26
<b>3. Omni-IPF Bulk Load .....</b>	<b>29</b>
Load Bulk Subject .....	29
PUT: /server/api/v1/server/bulk/load/{subject}?batchId=.....	29
Example:.....	29
Error Codes:.....	30
Process Bulk Subject .....	31
GET: /server/api/v1/server/bulk/process/{subject}?batchId=&subject=.....	31
Example:.....	31
<b>4. Omni Server Status Operations .....</b>	<b>33</b>
Status .....	33
GET: /server/.....	33
Example:.....	33
Ping Server .....	33
GET: /server/api/v1/ping.....	33
Example:.....	33
Running Product .....	34
GET: /server/api/v1/status/omniProduct.....	34
Example:.....	34
Running Mode.....	34
GET: /server/api/v1/status/omniRunningMode.....	34
Example:.....	34

**5. Omni Controller Deployed Bundle Artifacts .....35**

- Download Bundle Artifact ..... 35
- GET: /api/v1/deploy/bundle/artifact..... 35
- Example.....35



# Preface

---

This documentation provides a reference for Omni Insurance Performance Foundation (Omni-IPF) API Services. This manual is intended for developers and administrators of Omni-IPF.

---

## How This Manual Is Organized

This manual includes the following chapters:

	<b>Chapter/Appendix</b>	<b>Contents</b>
1	Omni-IPF Instance and Master Services	Describes services used for subject instance and master management.
2	Omni-IPF Change Data Capture	Describes the Omni-IPF change data capture (CDC) services, which allow a client to subscribe to changes of instance or master records.
3	Omni-IPF Bulk Load	Describes the Omni-IPF bulk loading API, which supports the ability to load multiple Omni Interface Documents (OID) and process the loaded documents in a single operation.
4	Omni Server Status Operations	Describes the Omni Server status API calls, which allow for server state discovery, its health, and operation mode.
5	Omni Controller Deployed Bundle Artifacts	Describes API calls to download artifacts for a deployed bundle.

## Documentation Conventions

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

<b>Convention</b>	<b>Description</b>
<code>THIS TYPEFACE</code> or <code>this typeface</code>	Denotes syntax that you must type exactly as shown.

<b>Convention</b>	<b>Description</b>
<i>this typeface</i>	Represents a placeholder (or variable), a cross-reference, or an important term. It may also indicate a button, menu item, or dialog box option that you can click or select.
<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.
	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.

## Related Publications

Visit our Technical Documentation Library at <http://documentation.informationbuilders.com>. You can also contact the Publications Order Department at (800) 969-4636.

## Customer Support

Do you have questions about this product?

Join the Focal Point community. Focal Point is our online developer center and more than a message board. It is an interactive network of more than 3,000 developers from almost every profession and industry, collaborating on solutions and sharing every tips and techniques. Access Focal Point at <http://forums.informationbuilders.com/eve/forums>.



You can also access support services electronically, 24 hours a day, with InfoResponse Online. InfoResponse Online is accessible through our website, <http://www.informationbuilders.com>. It connects you to the tracking system and known-problem database at the Information Builders support center. Registered users can open, update, and view the status of cases in the tracking system and read descriptions of reported software issues. New users can register immediately for this service. The technical support section of [www.informationbuilders.com](http://www.informationbuilders.com) also provides usage techniques, diagnostic tips, and answers to frequently asked questions.

Call Information Builders Customer Support Services (CSS) at (800) 736-6130 or (212) 736-6130. Customer Support Consultants are available Monday through Friday between 8:00 A.M. and 8:00 P.M. EST to address all your questions. Information Builders consultants can also give you general guidance regarding product capabilities. Be prepared to provide your six-digit site code (xxxx.xx) when you call.

To learn about the full range of available support services, ask your Information Builders representative about InfoResponse Online, or call (800) 969-INFO.

## Help Us to Serve You Better

To help our consultants answer your questions effectively, be prepared to provide specifications and sample files and to answer questions about errors and problems.

The following table lists the environment information that our consultants require.

<b>Platform</b>	
<b>Operating System</b>	
<b>OS Version</b>	
<b>JVM Vendor</b>	
<b>JVM Version</b>	

The following table lists additional questions to help us serve you better.

<b>Request/Question</b>	<b>Error/Problem Details or Information</b>
Did the problem arise through a service or event?	

Request/Question	Error/Problem Details or Information
Provide usage scenarios or summarize the application that produces the problem.	
When did the problem start?	
Can you reproduce this problem consistently?	
Describe the problem.	
Describe the steps to reproduce the problem.	
Specify the error messages.	
Any change in the application environment: software configuration, EIS/database configuration, application, and so forth?	
Under what circumstance does the problem <i>not</i> occur?	

The following is a list of error and problem files that might be applicable.

- Input documents (XML instance, XML schema, non-XML documents)
- Transformation files
- Error screen shots
- Error output files
- Trace files
- Custom functions and agents in use
- Diagnostic Zip
- Transaction log

## User Feedback

In an effort to produce effective documentation, the Technical Content Management staff welcomes your opinions regarding this document. Please use the Reader Comments form at the end of this document to communicate your feedback to us or to suggest changes that will support improvements to our documentation. You can also contact us through our website, <http://documentation.informationbuilders.com/connections.asp>.

Thank you, in advance, for your comments.

## iWay Software Training and Professional Services

Interested in training? Our Education Department offers a wide variety of training courses for iWay Software and other Information Builders products.

For information on course descriptions, locations, and dates, or to register for classes, visit our website, <http://education.informationbuilders.com>, or call (800) 969-INFO to speak to an Education Representative.

Interested in technical assistance for your implementation? Our Professional Services department provides expert design, systems architecture, implementation, and project management services for all your business integration projects. For information, visit our website, <http://www.informationbuilders.com/consulting>.



## Omni-IPF Instance and Master Services

---

The services described in this section are used for subject instance and master management. These services allow a consumer to read and write instances and read and sync with masters.

### In this chapter:

- [Load Subject Instance](#)
  - [Get an Instance](#)
  - [Get a Master](#)
  - [Query for a Master](#)
- 

### Load Subject Instance

Use the following REST call to initiate the load and mastering process for a subject in Omni-IPF. Omni-IPF will take the subject information in the form of an OID XML and will perform requested cleansing, matching, and merging of the subject.

Omni-IPF can process fully qualified or partial OID XML documents. The required `SourceName` and `SourceInstancelid` elements are used by Omni-IPF to determine whether a subject is new or an update to a previous instance. Omni-IPF will construct the best possible subject document by intelligently combining the current subject instance with previous instance data. This intelligent combination may be configured at the document, source, subject, or source/subject levels.

#### **PUT: /server/api/v1/server/instance?mode=&timeout=**

The data must be a correctly formatted OID XML, which minimally requires a `SourceName` and `SourceInstancelid`. The OID may be fully qualified or partially qualified.

- `mode` – Specify the loading mode of `sync` or `async`. The default is `async`, as `sync` will block the caller until the data is fully processed or a timeout occurs.
- `timeout` – When `mode=sync`, the caller will be blocked until the document is fully processed or until this timeout is exceeded.

**Example:**

PUT https://localhost:9512/server/api/v1/server/instance

Content-Type: application/xml

```
<customer>
  <SourceName>ReadTest</SourceName>
  <SourceInstanceId>ReadTestCustomer</SourceInstanceId>
  <cu_title>DEVELOPER</cu_title>
  <cu_salutation>MR</cu_salutation>
  <cu_suffix>PHD</cu_suffix>
  <cu_first_name>SCOTT</cu_first_name>
  <cu_middle_name>W</cu_middle_name>
  <cu_last_name>BISHOPP</cu_last_name>
  <cu_full_name>MR SCOTT W BISHOPP PHD</cu_full_name>
  <cu_gender>M</cu_gender>
  <cu_dob format="yyyy-MM-dd">1965-02-21</cu_dob>
  <cu_ssn>874-98-4546</cu_ssn>
  <cu_type>P</cu_type>
  <cu_bus_name>IBI</cu_bus_name>
  <cu_dba_name>SCOTTYB</cu_dba_name>
  <cust_addressOmniCollection>
    <customercust_address version="">
      <SourceName>ReadTest</SourceName>
      <SourceInstanceId>ReadTest:address</SourceInstanceId>
      <ad_1>4212 S LIVONIA RD</ad_1>
      <ad_2>ATTN: SCOTT BISHOPP</ad_2>
      <ad_city>LIVONIA</ad_city>
      <ad_state>NY</ad_state>
      <ad_zip>14487</ad_zip>
      <ad_zip4>0212</ad_zip4>
      <ad_country>US</ad_country>
    </customercust_address>
  </cust_addressOmniCollection>
</customer>
```

**- response -**

```
200 OK
Server: Apache-Coyote/1.1
X-Application-Context: application:9500
Content-Type: application/xml; charset=UTF-8
Content-Length: 202
Date: Tue, 08 May 2018 15:34:06 GMT
```

```
<RestResponse>
  <status>0</status>
  <statusText></statusText>
  <responseType>java.lang.String</responseType>
  <response>
    Submitted request for customer:ReadTest:ReadTestCustomer
  </response>
</RestResponse>
```

## Get an Instance

Use the following service to get an instance using its unique identifier.

**GET: /server/api/v1/server/instance/{subject}/{sourceName}/{sourceInstanceId}**

- ❑ {subject} – Required subject name.
- ❑ {sourceName} – Required source name of the instance.
- ❑ {sourceInstanceId} – Required instance identifier.

### Example:

**GET** https://localhost:9512/server/api/v1/server/instance/customer/ReadTest/ReadTestCustomer

**– response –**

```
200 OK
Server: Apache-Coyote/1.1
X-Application-Context: application:9500
Content-Type: application/xml;charset=UTF-8
Content-Length: 1256
Date: Tue, 08 May 2018 15:42:07 GMT
```

```
<RestResponse>
<status>0</status>
<statusText></statusText>
<responseType>java.lang.String</responseType>  <response>
<customer>
  <SourceName>ReadTest</SourceName>
  <SourceInstanceId>ReadTestCustomer</SourceInstanceId>
  <MasterId>26</MasterId>
  <cu_title>Developer</cu_title>
  <cu_salutation>Mr</cu_salutation>
  <cu_suffix>Phd</cu_suffix>
  <cu_first_name>Scott</cu_first_name>
  <cu_middle_name>W</cu_middle_name>
  <cu_last_name>Bishopp</cu_last_name>
  <cu_full_name>Mr Scott W Bishopp Phd</cu_full_name>
  <cu_gender>M</cu_gender>
  <cu_dob format="yyyy-MM-dd">1965-02-21</cu_dob>
  <cu_ssn>874-98-4546</cu_ssn>
  <cu_type>P</cu_type>
  <cu_bus_name>IBI</cu_bus_name>
  <cu_dba_name>SCOTTYB</cu_dba_name>
  <cust_addressOmniCollection>
    <customercust_address>
      <SourceName>ReadTest</SourceName>
      <SourceInstanceId>ReadTest:address</SourceInstanceId>
      <ad_1>4212 S LIVONIA RD</ad_1>
      <ad_2>ATTN: SCOTT BISHOPP</ad_2>
      <ad_city>LIVONIA</ad_city>
      <ad_state>NY</ad_state>
      <ad_zip>14487</ad_zip>
      <ad_zip4>0212</ad_zip4>
      <ad_country>US</ad_country>
    </customercust_address>
  </cust_addressOmniCollection>
</customer>
</response>
</RestResponse>
```

## Get a Master

Use the following service to get a master with a specified master ID.

**GET: /server/api/v1/server/master/{masterSubject}/{masterId}**

- ❑ {masterSubject} – Name of mastered subject.
- ❑ {masterId} – Master ID.

### Example:

**GET** https://localhost:9512/server/api/v1/server/master//customerMaster/26



**- response -**

```

200 OK
Server: Apache-Coyote/1.1
X-Application-Context: application:9500
Content-Type: application/xml;charset=UTF-8
Content-Length: 1146
Date: Tue, 08 May 2018 15:51:20 GMT

<RestResponse>
<status>0</status>
<statusText></statusText>
<responseType>java.lang.String</responseType> <response>
<customerMaster>
  <MasterId>26</MasterId>
  <cu_title>Developer</cu_title>
  <cu_salutation>Mr</cu_salutation>
  <cu_suffix>Phd</cu_suffix>
  <cu_first_name>Scott</cu_first_name>
  <cu_middle_name>W</cu_middle_name>
  <cu_last_name>Bishopp</cu_last_name>
  <cu_full_name>Mr Scott W Bishopp Phd</cu_full_name>
  <cu_gender>M</cu_gender>
  <cu_dob format="yyyy-MM-dd">1965-02-21</cu_dob>
  <cu_ssn>874-98-4546</cu_ssn>
  <cu_type>P</cu_type>
  <cu_bus_name>IBI</cu_bus_name>
  <cu_dba_name>SCOTTYB</cu_dba_name>
  <cust_addressOmniCollection>
    <customercust_addressMaster>
      <MasterChildId>customer:26:cust_address:1</MasterChildId>
      <ad_1>4212 S LIVONIA RD</ad_1>
      <ad_2>ATTN: SCOTT BISHOPP</ad_2>
      <ad_city>LIVONIA</ad_city>
      <ad_state>NY</ad_state>
      <ad_zip>14487</ad_zip>
      <ad_zip4>0212</ad_zip4>
      <ad_country>US</ad_country>
    </customercust_addressMaster>
  </cust_addressOmniCollection>
</customerMaster>
</response>
</RestResponse>

```

## Query for a Master

Use the following REST call to check if subject data matches any currently mastered subject. This service expects data in the form of a subject OID with data used for the defined matching criteria. This service will NOT cause changes in any instance or master data, including the matching indexes.

The implementation of this service will cause execution of defined cleansing plans against the input data to ensure it is standardized before the match attempt. After cleansing, the READ ONLY matching service will be executed to see if a high confidence match and associated master are available.

If a match is available this service can respond with the master ID or the full master XML document based on the responseType parameter. If a match is not found, the service will respond with an HTTP status of 404 (Not Found).

### POST: /api/v1/server/master?responseType=oid|masterId

The body of the post must be a properly formatted XML OID. Only the data used for the matching service is evaluated and other data will simply be ignored.

**Note:** This will NOT create data in Omni-IPF.

#### Example:

**POST** https://localhost:9512/server/api/v1/server/master?responseType=oid

Content-Type: application/xml

```
<customer>
  <cu_first_name>SCOTT</cu_first_name>
  <cu_middle_name>W</cu_middle_name>
  <cu_last_name>BISHOPP</cu_last_name>
  <cu_gender>M</cu_gender>
  <cu_dob format="yyyy-MM-dd hh:mm:ss">1965-02-21 01:34:09</cu_dob>
  <cu_ssn>874-98-4546</cu_ssn>
</customer>
```

#### – response –

```
200 OK
Server: Apache-Coyote/1.1
X-Application-Context: application:9500
Content-Type: application/xml; charset=UTF-8
Content-Length: 1146
Date: Tue, 08 May 2018 16:00:47 GMT
```

```

<RestResponse>
<status>0</status>
<statusText></statusText>
<responseType>java.lang.String</responseType>
<response>
<customerMaster>
  <MasterId>26</MasterId>
  <cu_title>Developer</cu_title>
  <cu_salutation>Mr</cu_salutation>
  <cu_suffix>Phd</cu_suffix>
  <cu_first_name>Scott</cu_first_name>
  <cu_middle_name>W</cu_middle_name>
  <cu_last_name>Bishopp</cu_last_name>
  <cu_full_name>Mr Scott W Bishopp Phd</cu_full_name>
  <cu_gender>M</cu_gender>
  <cu_dob format="yyyy-MM-dd">1965-02-21</cu_dob>
  <cu_ssn>874-98-4546</cu_ssn>
  <cu_type>P</cu_type>
  <cu_bus_name>IBI</cu_bus_name>
  <cu_dba_name>SCOTTYB</cu_dba_name>
  <cust_addressOmniCollection>
    <customercust_addressMaster>
      <MasterChildId>customer:26:cust_address:1</MasterChildId>
      <ad_1>4212 S LIVONIA RD</ad_1>
      <ad_2>ATTN: SCOTT BISHOPP</ad_2>
      <ad_city>LIVONIA</ad_city>
      <ad_state>NY</ad_state>
      <ad_zip>14487</ad_zip>
      <ad_zip4>0212</ad_zip4>
      <ad_country>US</ad_country>
    </customercust_addressMaster>
  </cust_addressOmniCollection>
</customerMaster>
</response>
</RestResponse>

```



# Chapter 2

## Omni-IPF Change Data Capture

---

The Omni-IPF change data capture (CDC) services allow a client to subscribe to changes of instance or master records. When subscribed, changes are detected, and the instance or master OID will be published to a location that the subscriber designated.

### In this chapter:

- [Subscribe to Changes](#)
  - [Cancel a Subscription](#)
  - [Enable or Disable a Subscription](#)
  - [Get List of Subscription](#)
  - [Get Change Data Capture Process Status](#)
  - [Start the Change Data Capture Process](#)
  - [Stop the Change Data Capture Process](#)
- 

### Subscribe to Changes

Use the following REST call to subscribe to changes for a specific subject instance or master. The payload sent to the destination will be the OID that has been modified.

#### POST: /server/api/v1/server/cdc/subscribe

The data body is the configuration information about where to publish, the subscriber and optional the source to watch.

```
<SubscriptionConfiguration>
  <subject/>
  <source/>
  <subscriber/>
  <connectionType/>
  <url/>
</SubscriptionConfiguration>
```

- `subject` – Required subject to subscribe to. This may be an instance or master subject.
- `source` – Optional source when subscribing to instance changes.
- `subscriber` – Required identifier of the subscriber.

- ❑ `connectionType` – Required connection type, which must be set to REST.
- ❑ `url` – Required when the `connectionType` is REST and is used to specify the endpoint that can accept the POST operation.

### Response:

A `RestResponse` will be returned with the `<response>` containing the unique subscription ID.

### Example:

POST <https://localhost:9512/server/api/v1/server/cdc/subscribe>

Content-Type: application/xml

```
<SubscriptionConfiguration>
  <subject>customer</subject>
  <subscriber>Test</subscriber>
  <connectionType>rest</connectionType>
  <url>
    https://localhost:9512/server/api/v1/server/cdc/subscribe/loopback
  </url>
</SubscriptionConfiguration>
```

### – response –

```
200 OK
Server: Apache-Coyote/1.1
X-Application-Context: application:9500
Content-Type: application/xml;charset=UTF-8
Transfer-Encoding: chunked
Date: Thu, 03 May 2018 11:37:45 GMT
```

```
<RestResponse>
  <status>0</status>
  <statusText/>
  <responseType>java.lang.String</responseType>
  <response>271192d4-48d0-4e91-869e-dc1b477311be</response>
</RestResponse>
```

## Cancel a Subscription

Use the following REST call to cancel a subscription.

**DELETE:** [/server/api/v1/server/cdc/subscribe/{subscriptionId}](https://localhost:9512/server/api/v1/server/cdc/subscribe/{subscriptionId})

Specify the subscription ID to cancel.

### Example:

**DELETE** <https://localhost:9512/server/api/v1/server/cdc/subscribe/271192d4-48d0-4e91-869e-dc1b477311be>

**– response –**

```

200 OK
Server: Apache-Coyote/1.1
X-Application-Context: application:9500
Content-Type: application/xml;charset=UTF-8
Transfer-Encoding: chunked
Date: Thu, 03 May 2018 11:39:19 GMT

```

```

<RestResponse>
  <status>0</status>
  <statusText/>
  <responseType>java.lang.String</responseType>
  <response>271192d4-48d0-4e91-869e-dc1b477311be</response>
</RestResponse>

```

**Enable or Disable a Subscription**

Use the following REST call to enable or disable a subscription. When a subscription is disabled no changes will be set to the subscriber.

**PUT: /server/api/v1/server/cdc/subscribe/{subscriptionId}?status=ENABLE|DISABLE**

Specify the subscription ID and the new status of the subscription.

**Example:**

**PUT** https://localhost:9512/server/api/v1/server/cdc/subscribe/5f1021c3-234b-4a05-ae88-86ff6af8aa8e?status=disable

Content-Type: application/xml

**– response –**

```

200 OK
Server: Apache-Coyote/1.1
X-Application-Context: application:9500
Content-Type: application/xml;charset=UTF-8
Transfer-Encoding: chunked
Date: Thu, 03 May 2018 11:41:36 GMT

```

```

<RestResponse>
  <status>0</status>
  <statusText/>
  <responseType>java.lang.String</responseType>
  <response>DISABLED</response>
</RestResponse>

```

## Get List of Subscription

Use the following REST call to the list of current subscriptions

### GET: /server/api/v1/server/cdc/subscribe

Retrieves the list of current subscriptions.

#### Example:

**GET** https://localhost:9512/server/api/v1/server/cdc/subscribe

#### – response –

```
200 OK
Server: Apache-Coyote/1.1
X-Application-Context: application:9500
Content-Type: application/xml;charset=UTF-8
Transfer-Encoding: chunked
Date: Tue, 08 May 2018 11:25:11 GMT

<RestResponse>
  <status>0</status>
  <statusText/>
  <responseType>java.util.ArrayList</responseType>
  <responses>
    <responseItem>
      <CdcSubscriptionInfo>
        <subscriber>Test</subscriber>
        <subject>customer</subject>
        <source/>
        <connectionType>rest</connectionType>
        <connectionInfo/>
        <url>
          http://localhost:9500/controller/api/v1/server/cdc/subscribe/loopback
        </url>
        <createdDate>2016-05-10</createdDate>
        <lastUpdate>2016-05-10</lastUpdate>
        <id>ab529810-e4a8-4718-bb7d-b37240ceeffe</id>
        <status>E</status>
      </CdcSubscriptionInfo>
    </responseItem>
  </responses>
</RestResponse>
```

## Get Change Data Capture Process Status

Use the following REST call to get the current status of CDC process.

### GET: /server/api/v1/server/cdc/status

Get the current status of the CDC poller process.



**Example:****GET** https://localhost:9512/server/api/v1/server/cdc/status**-- response --**

```

200 OK
Server: Apache-Coyote/1.1
X-Application-Context: application:9500
Content-Type: application/xml;charset=UTF-8
Transfer-Encoding: chunked
Date: Tue, 08 May 2018 13:32:06 GMT

<RestResponse>
  <status>0</status>
  <statusText/>

<responseType>com.ibi.omnigen.controller.cdc.NotificationServiceStatus</
responseType>
  <response>
    <NotificationServiceStatus>
      <status>Stopped</status>
      <configurationValid>true</configurationValid>
      <serviceName>cdc</serviceName>
      <lastRun/>
      <pollInterval>300</pollInterval>
    </NotificationServiceStatus>
  </response>
</RestResponse>

```

**Start the Change Data Capture Process**

Use the following REST call to start the CDC process with an optional interval.

**GET: /server/api/v1/server/cdc/start?interval=<seconds>**

Start the CDC process.

**Example:****GET** https://localhost:9512/server/api/v1/server/cdc/start?interval=5**-- response --**

```

200 OK
Server: Apache-Coyote/1.1
X-Application-Context: application:9500
Content-Type: application/xml;charset=UTF-8
Transfer-Encoding: chunked
Date: Tue, 08 May 2018 13:39:53 GMT

```

```
<RestResponse>
  <status>0</status>
  <statusText/>
  <responseType>
    com.ibi.omnigen.controller.cdc.NotificationServiceStatus
  </responseType>
  <response>
    <NotificationServiceStatus>
      <status>Running</status>
      <configurationValid>true</configurationValid>
      <serviceName>cdc</serviceName>
      <lastRun/>
      <pollInterval>5</pollInterval>
    </NotificationServiceStatus>
  </response>
</RestResponse>
```

## Stop the Change Data Capture Process

Use the following REST call to start the CDC process with an optional interval.

**GET: /server/api/v1/server/cdc/stop**

Stop the CDC process.

### Example:

**GET** <https://localhost:9512/server/api/v1/server/cdc/stop>

**- response -**

```
200 OK
Server: Apache-Coyote/1.1
X-Application-Context: application:9500
Content-Type: application/xml;charset=UTF-8
Transfer-Encoding: chunked
Date: Tue, 08 May 2018 13:44:53 GMT

<RestResponse>
  <status>0</status>
  <statusText/>
  <responseType>
    com.ibi.omnigen.controller.cdc.NotificationServiceStatus
  </responseType>
  <response>
    <NotificationServiceStatus>
      <status>Stopped</status>
      <configurationValid>true</configurationValid>
      <serviceName>cdc</serviceName>
      <lastRun>2018-05-08</lastRun>
      <pollInterval>5</pollInterval>
    </NotificationServiceStatus>
  </response>
</RestResponse>
```



# Chapter 3

## Omni-IPF Bulk Load

---

The Omni-IPF bulk loading API supports the ability to load multiple Omni Interface Documents (OID) and process the loaded documents in a single operation.

### In this chapter:

- [Load Bulk Subject](#)
  - [Process Bulk Subject](#)
- 

### Load Bulk Subject

Use the following REST call to load an OID XML into the OnRamp with the specified batchId. This will simply load the data without any further processing. The batchId is generated by the source system and can be used to execute the processing of all the data loaded with it.

#### Caution:

This functionality is focused on the ability to bulk load data into the ramp tables through a REST API. You are not allowed to put duplicate subjects or children on the ramp within the same batch. This is the known design of the ramp and will be enforced through this API.

### PUT: /server/api/v1/server/bulk/load/{subject}?batchId=

The data payload must be a correctly formatted OID XML, which minimally requires a SourceName and SourceInstanceId. The OID may be fully or partially qualified.

- subject** – Specify the name of the subject that is being loaded. This must match the subject that is being loaded in the payload.
- batchId** – Generated that should not collide with any other bulk loading operation. This batchId will define the set of data to process using the bulk processing command.

#### Example:

**PUT** <https://localhost:9512/server/api/v1/server/bulk/load/Facility?batchId=4>

```
<Facility version="1.1.7">
  <SourceName>test_system</SourceName>
  <SourceInstanceId>1499870996285_86_3</SourceInstanceId>
  <Type sourceName="test_system" codeSet="Types">type_code</Type>
  <Name>TestHospitalName</Name>
  ...
</Facility>
```

### **- response -**

```
{
  "status": "OK",
  "code": 0,
  "message": "Added Facility:test_system:1499870996285_86_3 OID XML
document to Ramp.",
  "developerMessage": null,
  "responseType": "com.ibi.omni.rest.support.ServiceOperationDto",
  "response": {
    "service": "RampProcessing",
    "operation": "writeOidToRamp",
    "code": 0,
    "message": "Added Facility:test_system:1499870996285_86_3 OID XML
document to Ramp.",
    "start": 1500059394757,
    "end": 1500059396014
  },
  "exception": null
}
```

Use "code" to determine the result of the request. Any value other than zero (0) should be considered an error and the message will contain the reason.

### **Error Codes:**

- 100 – Missing required SourceName element.
- 110 – Missing required SourceInstanceId element.
- 120 – Fatal error processing the OID.
- 130 – Invalid XML document format.

## Process Bulk Subject

Use the following REST call to process data for a specific subject and batch. This will configure and initiate the processing of the subject and batch. This call will return with status, but the actual execution by the server will begin when its available to do so.

### Note:

While multiple subjects may be loaded using the same batch, you must execute this REST API for each subject in the batch.

### GET: /server/api/v1/server/bulk/process/{subject}?batchId=&subject=

Execute the processing of a loaded subject and batch.

- batchId – Specify the batchId used to load the data.
- subject – Specify the name of the subject to process.

### Example:

**GET** https://localhost:9512/server/api/v1/server/bulk/process/Facility?batchId=4

#### – response –

```
{
  "status": "OK",
  "code": 0,
  "message": "Scheduled ramp load of Facility with count = 4",
  "developerMessage": null,
  "responseType": "com.ibi.omni.rest.support.ServiceOperationDto",
  "response": {
    "service": "RampProcessing",
    "operation": "processRampData",
    "code": 0,
    "message": "Scheduled ramp load of Facility with count = 4",
    "start": 1500059103385,
    "end": 1500059103427
  },
  "exception": null
}
```

Use "code" to determine the result of the request. Any value other than zero (0) should be considered an error and the message will contain the reason.





## Omni Server Status Operations

---

The Omni Server status API calls allow for server state discovery, its health, and operation mode.

### In this chapter:

- Status
  - Ping Server
  - Running Product
- 

### Status

Use the following operation to receive the server status object.

**GET: /server/**

#### Example:

**GET** https://localhost:9512/server/

**–response –**

```
{
  "status": "OK",
  "code": 0,
  "message": null,
  "developerMessage": null,
  "responseType": "java.lang.String",
  "response": "Server Running",
  "exception": null
}
```

### Ping Server

Use the following operation to ping the server and receive string "ok".

**GET: /server/api/v1/ping**

#### Example:

**GET** https://localhost:9512/server/api/v1/ping

**–response –**

ok

## Running Product

Use the following operation to determine edition of the running product.

**GET: /server/api/v1/status/omniProduct**

**Example:**

**GET** https://localhost:9512/server/api/v1/status/omniProduct

**–response–**

```
{
  "status": "OK",
  "code": 0,
  "message": null,
  "developerMessage": null,
  "responseType": "java.lang.String",
  "response": "MD_BASE_EDITION",
  "exception": null
}
```

## Running Mode

Use the following operation to determine the product running mode.

**GET: /server/api/v1/status/omniRunningMode**

**Example:**

**GET** https://localhost:9512/server/api/v1/status/omniRunningMode

**–response–**

```
{
  "status": "OK",
  "code": 0,
  "message": null,
  "developerMessage": null,
  "responseType": "java.lang.String",
  "response": "DEVELOPMENT",
  "exception": null
}
```

## Omni Controller Deployed Bundle Artifacts

---

This section describes API calls to download artifacts for a deployed bundle.

**In this chapter:**

- ❑ [Download Bundle Artifact](#)
- 

### Download Bundle Artifact

Use the following API call to extract different types of artifacts for a deployed bundle.

It could be XSD, documentation, examples, bundle itself.

#### GET: /api/v1/deploy/bundle/artifact

##### Example

**GET** `https://localhost:9500/api/v1/deploy/bundle/artifact`

- ❑ {filename} – Artifact name.
- ❑ {artifact} – Requested artifact type from the list: documentation, xsd, example, ids, bundle

**-request -**

```
https://localhost:9500/api/v1/deploy/bundle/artifact?  
filename=SourceCodeStandard.xml&artifact=example
```

**-response-**

```
<?xml version="1.0" encoding="UTF-8"?>
<OmniInterface>
  <SourceCodeStandard version="3.0.0">
    <BaseCode sourceName="test_system" codeSet="BaseCodes">base_code</
BaseCode>
    <StandardCode sourceName="test_system"
codeSet="StandardCodes">standard_code</StandardCode>
    <Description>description</Description>
    <SourceStatusCode sourceName="test_system"
codeSet="SourceStatusCodes">source_status_code</SourceStatusCode>
    <SourceCreatedDate format="yyyy-MM-dd">2017-10-18</SourceCreatedDate>
    <SourceCreatedBy>source_created_by</SourceCreatedBy>
    <SourceModifiedDate format="yyyy-MM-dd">2017-10-1</SourceModifiedDate>
    <SourceModifiedBy>source_modified_by</SourceModifiedBy>
  </SourceCodeStandard>
</OmniInterface>
```



# Feedback

*Customer success is our top priority. Connect with us today!*

---

Information Builders Technical Content Management team is comprised of many talented individuals who work together to design and deliver quality technical documentation products. Your feedback supports our ongoing efforts!

You can also preview new innovations to get an early look at new content products and services. Your participation helps us create great experiences for every customer.

To send us feedback or make a connection, contact Sarah Buccellato, Technical Editor, Technical Content Management at [Sarah\\_Buccellato@ibi.com](mailto:Sarah_Buccellato@ibi.com).

To request permission to repurpose copyrighted material, please contact Frances Gambino, Vice President, Technical Content Management at [Frances\\_Gambino@ibi.com](mailto:Frances_Gambino@ibi.com).

# iWay

## / Omni-IPF API Services Reference Guide

Version 3.6

DN3502349.0219

Information Builders, Inc.  
Two Penn Plaza  
New York, NY 10121-2898