

ellucian.

Banner Advancement Web Services Handbook

Release 8.8
December 2015



Without limitation: Ellucian®, Banner®, Colleague®, and Luminis® are trademarks of the Ellucian group of companies that are registered in the U.S. and certain other countries; and Ellucian Advance™, Ellucian Course Signals™, Ellucian Degree Works™, Ellucian PowerCampus™, Ellucian Recruiter™, Ellucian SmartCall™, are also trademarks of the Ellucian group of companies. Other names may be trademarks of their respective owners.

© 2009, 2015 Ellucian.

Contains confidential and proprietary information of Ellucian and its subsidiaries. Use of these materials is limited to Ellucian licensees, and is subject to the terms and conditions of one or more written license agreements between Ellucian and the licensee in question.

In preparing and providing this publication, Ellucian is not rendering legal, accounting, or other similar professional services. Ellucian makes no claims that an institution's use of this publication or the software for which it is provided will guarantee compliance with applicable federal or state laws, rules, or regulations. Each organization should seek legal, accounting, and other similar professional services from competent providers of the organization's own choosing.

Ellucian
4375 Fair Lakes Court
Fairfax, VA 22033
United States of America

Revision History

Publication Date	Summary
December 2015	New version that supports Banner Advancement 8.8 software.

Contents

Introduction	5
What is a web service?	5
What is the GetGivingHistory web service?	5
What is the Banner Advancement Web Services Adapter?	5
Install Banner Advancement Web Service Adapter	7
Requirements	7
Recommended configuration	7
Installation steps	8
<i>Step 1 - Configure the Oracle WebLogic Server</i>	8
<i>Step 2 - Configure logging (optional)</i>	9
<i>Step 3- Create a Managed Server</i>	10
<i>Step 4- Configure the Managed Server</i>	10
<i>Step 5- Define the adapter data source</i>	11
<i>Step 6- Define the Translation Service data source</i>	12
<i>Step 7- Install the adapter</i>	14
<i>Step 8 - Configure the security group and user</i>	15
<i>Step 9 - Enable schema validation (optional)</i>	16
<i>Step 10 - Update the WSDL endpoint URL (optional)</i>	17
<i>Step 11 - Test the Banner Advancement web service (optional)</i>	18
GetGivingHistory Web Service	19
Message exchange	19
GivingHistoryRequest	19
GivingHistoryResponse	19
SOAP fault messages	20
Message mapping to Banner	20
GivingHistoryRequest	21
GivingHistoryResponse	21

Intended usage	21
Setup requirements	22
Test cases	22
Advancement individual with giving history	22
Advancement individual without giving history	23
Invalid advancement individual ID	24

Introduction

This chapter introduces web services, describes the GetGivingHistory web service, and describes the Banner® Advancement Web Services Adapter.

What is a web service?

A web service exposes an application's processing logic to support a service-oriented architecture and to facilitate integration with external systems. A web service allows an external system or business process to invoke the application's logic without having to understand the application's internal structure.

Web services are based on open, Internet-based standards. This makes them relevant to application integration within an organization and with external organizations. Standards such as XML, SOAP, WSDL, and UDDI provide cross-platform compatibility that does not depend on a single programming language or network transport.

What is the GetGivingHistory web service?

The GetGivingHistory web service allows external systems to request giving history information for a donor who is set up in Banner. The web service returns summary giving information and a list of gifts for the donor.

What is the Banner Advancement Web Services Adapter?

The Banner Advancement Web Services Adapter is a Java-based application that exposes the GetGivingHistory web service. This exposure makes the Banner Advancement functions available to external systems using the SOAP protocol over HTTP/HTTPS. External systems interact with the web service, which in turn is supported by Banner APIs. This layered approach provides an insulating buffer between external systems and Banner. External systems do not interact with Banner directly, but rather exchange XML messages with the exposed web service.

The Banner Advancement Web Services Adapter supports the synchronous, request/reply message exchange pattern as follows:

1. The external system requests a service of Banner by sending an XML message to the web service endpoint that is exposed by the adapter. The message contains the information required for Banner to service the request.
2. The Banner Advancement Web Services Adapter invokes the appropriate Banner API.
3. The Banner API performs the necessary Banner processing logic.
4. One of the following occurs:
 - 4.1. If the action is completed successfully, the API provides a response message, which the adapter forwards to the external system.
 - 4.2. If the action is not completed successfully, the adapter sends an error message (called a SOAP fault) to the external system.

Install Banner Advancement Web Service Adapter

The Banner® Advancement Web Services Adapter exposes the GetGivingHistory web service. This chapter gives instructions for installing the adapter on Oracle WebLogic Server 11g.

Requirements

The Banner Advancement Web Services Adapter has the following requirements:

- Oracle WebLogic Server 11g with Java 1.7
- Oracle Database 11g
- Banner Translation Service data source



Note: The adapter does *not* require the Banner Translation Service. The adapter, however, requires the definition of a Translation Service data source. This chapter includes a step for creating the Translation Service data source. If you are deploying the adapter into the same container with other Banner web service adapters, the data source should already be defined.

- Banner Advancement 8.4 or higher (contains objects that create and expose a Banner Advancement API as a web service)

Recommended configuration

The Banner Advancement Web Services Adapter must be installed in an Oracle WebLogic Basic Domain. It must not be installed using any other Oracle WebLogic template, especially the Oracle WebLogic Classic Domain that supports Oracle Forms and Reports.

The recommended configuration is to establish a separate physical or virtual domain for the adapter and other middle-tier components. This domain would run a separate installation of Oracle WebLogic Server, configured using the Basic Domain template (not the Classic Domain template) that is provided by Oracle.

The Oracle WebLogic Server domain should consist of the default Admin Server and at least one Managed Server for the deployment of the adapter.

If a domain based on the Basic Domain template already exists for middle-tier applications, the adapter can be installed in a separate Managed Server in that domain.

Installation steps

The Banner Advancement Web Services Adapter is packaged as a J2EE compatible enterprise archive file named `advancement_services.ear`. Use the following steps to install the adapter on Oracle WebLogic Server 11g:

- [Step 1 - Configure the Oracle WebLogic Server](#)
- [Step 2 - Configure logging \(optional\)](#)
- [Step 3- Create a Managed Server](#)
- [Step 4- Configure the Managed Server](#)
- [Step 5- Define the adapter data source](#)
- [Step 6- Define the Translation Service data source](#)
- [Step 7- Install the adapter](#)
- [Step 8 - Configure the security group and user](#)
- [Step 9 - Enable schema validation \(optional\)](#)
- [Step 10 - Update the WSDL endpoint URL \(optional\)](#)
- [Step 11 - Test the Banner Advancement web service \(optional\)](#)

The following sections give detailed instructions for each step.

Step 1 - Configure the Oracle WebLogic Server



Note: The Oracle WebLogic Server needs to be configured only once. If the server was previously configured, you can skip this step.

The Oracle WebLogic Server must be configured to use the *DD only* option selected from the **Security Model Default** drop-down list. This step pertains to the realm configuration. It applies to the entire domain. (Although you can create a totally new realm for the domain, only one realm can be active at a time for the entire domain.) This security configuration protects all server resources for the domain.

Use the following steps to configure the server.

1. Connect to the Oracle WebLogic Server administration console:

```
http://<host>:<port>/console
```
2. In the Change Center pane, click **Lock & Edit**.
3. In the Domain Structure pane, click **Security Realms**.

4. Click **myrealm**.
5. On the Settings for myrealm page:
 - 5.1. Select `DD only` from the **Security Model Default** drop-down list.
 - 5.2. Click the **Advanced** link.
6. Select `All Web Applications and EJBs` in the **Check Roles and Policies** drop-down list.
7. Click **Save**.
8. In the Change Center pane, click **Activate Change**.

Step 2 - Configure logging (optional)

The Banner Advancement Web Services Adapter uses Apache's log4j to log the activities performed by the application at runtime. The log file is located at the following location:

```
Oracle\Middleware\user_projects\domains\<<domain_name>\
servers\<<Managed_Server_name>\logs
```

where `<domain_name>` is the name of the domain where the Banner Advancement Web Services Adapter will be installed. This location cannot be changed.

A property in the `log4j.properties` file determines the logging level. The default logging level is `DEBUG`, resulting in a large amount of information (`INFO`, `WARNING`, `ERROR`, and `FATAL` level statements) being stored in log files. The logging level should be `DEBUG` for initial operations only. To provide more limited logging after initial operations, you should change the logging level to `INFO`.

Use the following steps to modify the logging level if you want less detailed logging.

1. Extract `pcr-000114584_alu8070103`. The extract directory is referred to as `<ZIP_HOME>`.
2. Navigate to `<ZIP_HOME>/advancement_services.ear`.
3. Copy `advancement_services.ear` to a temporary location. This location is referred to as `<EAR_HOME>`.
4. Navigate to `<EAR_HOME>` and execute the following command.

```
jar xvf advancement_services.ear
```

The extract contains `advancement_web.war`.
5. Create a folder under `<EAR_HOME>` and name it `war_home`.
6. Navigate to `war_home` and execute the following command.

```
jar xvf <EAR_HOME>/advancement_web.war
```
7. Open `war_home/WEB-INF/classes/log4j.properties`.

8. Edit the `log4j.category.com.sungardsct` property as follows:

Original value:	DEBUG
New value:	INFO

9. Save the change.

10. From `war_home` execute the following command to rebuild the `.war` file.

```
jar cvf <EAR_HOME>/advancement_web.war META-INF/* WEB-INF/* ui/* index.jsp
```

11. From `<EAR_HOME>` execute the following command to rebuild the `.ear` file.

```
jar cvf advancement_services.ear *.war META-INF/* legal/* APP-INF/*
```

The rebuilt `advancement_services.ear` is used for installation.

Step 3- Create a Managed Server

The Managed Server to which the application is deployed should be dedicated to the Advancement so it can be managed independently from other applications.

Create a new Managed Server for the Banner Advancement Web Service Adapter so that it can be independently managed. Refer to Oracle WebLogic Server Documentation Library for details.



Note: This step only needs to be performed once. If you previously performed this step, do not perform the steps below.

Step 4- Configure the Managed Server

Use the following steps to configure the managed server:



Note: This step only needs to be performed once. If you previously performed this step, do not perform the steps below.

1. Connect to the Oracle WebLogic Server Administration Console.

The home page is displayed.

2. Click **Lock & Edit** from the Change Center.

3. In the Domain Structure pane, click **Environment**, then Servers.

The Summary of Servers is displayed.

4. Click the name of the managed server.

The Settings for `<servername>` is displayed, where `<servername>` is the name of the managed server.

5. Click **SSL** tab under Configuration tab.
6. Open Advanced Section.
At the top of the Advanced section, item **Hostname Verification**: use the drop-down to change the value from **BEA Hostname Verifier** to **None**.
7. Click **Save**.
8. Click the **Server Start** tab under the **Configuration** tab.
9. In the item **Arguments**: add the following line:

`-Xmx2048m -XX:PermSize=100m -XX:MaxPermSize=150m`
10. Click **Save**.
11. In the Change Center pane, click **Activate Changes**.

Step 5- Define the adapter data source

A data source provides the connection properties to the Banner database. By default, the adapter needs a data source with lookup name `jdbc/bannerws`. If you previously installed a Banner web services adapter in the instance, then you can skip this step. Otherwise, use the following steps to define the data source.

1. Connect to the Oracle WebLogic Server administration console:

`http://<host>:<port>/console`
2. In the Change Center pane, click **Lock & Edit**.
3. In the Domain Structure pane, expand and click **Services > JDBC > Data Sources**.
4. Click **New**.
5. Enter the following data source properties:

Name	BannerWS
JNDI Name	<code>jdbc/bannerws</code>
Database Type	Oracle

6. Click **Next**.
7. Select the appropriate database driver that is used to create database connections:
 - If your database is RAC-based, select `*Oracle's Driver (Thin) for RAC Service-Instance connections; Versions:10,11`.
 - Otherwise, select `*Oracle's Driver (Thin) for Instance connections; Versions:9.0.1,9.2.0,10,11`.
8. Click **Next**.
9. If the Connection Properties page is displayed, go to step 10.

If the Transaction Options page is displayed, clear the **Supports Global Transactions** check box and click **Next**. Then go to step 10.

10. Enter the following properties on the Connection Properties page:

Service Name	Service name of the database to which you are connecting. This field is displayed and is required if you selected *Oracle's Driver (Thin) for RAC Service-Instance connections; Versions:10,11 as the database driver.
Database Name	Name of the database to which you are connecting
Host Name	IP address or name of the database server
Port	Port on the database server that is used to connect to the database
Database User Name	integmgr
Password	Password for the integmgr user
Confirm Password	Confirmation of the password

11. Click **Next** to display the properties that you entered.
12. Verify the property values.
13. Click **Test Configuration**. The page is redisplayed with a success or failure message.
 - 13.1. If the test succeeds, continue with the next step.
 - 13.2. If the test fails, ensure that the connection URL and credentials are correct. Continue testing until the connection is successful.
14. Click **Next**.
15. In the Servers list, check the managed server created to deploy the new data source.
16. Click **Finish** to display the new data source.
17. Verify that the new data source is associated with the server.
18. In the Change Center pane, click **Activate Change**.

Step 6- Define the Translation Service data source

The adapter does *not* require the Banner Translation Service. The adapter, however, requires the definition of a Translation Service data source with lookup name `jdbc/transsvc`.

If you are deploying the adapter into the same container with other Banner web service adapters, the Translation Service data source should already be defined and you can skip this step. Otherwise, use the following steps to define the Translation Service data source.

1. In the Change Center pane, click **Lock & Edit**.
2. In the Domain Structure pane, expand and click **Services > JDBC > Data Sources**.

3. Click **New**.
4. Enter the following data source properties:

Name	transsvc
JNDI Name	jdbc/transsvc
Database Type	Oracle

5. Click **Next**.
6. Select the appropriate database driver that is used to create database connections:
 - If your database is RAC-based, select `*Oracle's Driver (Thin) for RAC Service-Instance connections; Versions:10,11`.
 - Otherwise, select `*Oracle's Driver (Thin) for Instance connections; Versions:9.0.1,9.2.0,10,11`.
7. Click **Next**.
8. If the Connection Properties page is displayed, go to step 9.
If the Transaction Options page is displayed, clear the **Supports Global Transactions** check box and click **Next**. Then go to step 9.
9. Enter the following properties on the Connection Properties page:

Service Name	Service name of the database to which you are connecting. This field is displayed and is required if you selected <code>*Oracle's Driver (Thin) for RAC Service-Instance connections; Versions:10,11</code> as the database driver.
Database Name	Name of the database to which you are connecting
Host Name	IP address or name of the database server
Port	Port on the database server that is used to connect to the database
Database User Name	integmgr
Password	Password for the integmgr user
Confirm Password	Confirmation of the password

10. Click **Next** to display the properties that you entered.
11. Verify the property values.

12. Click **Test Configuration**. The page is redisplayed with a success or failure message.
 - 12.1. If the test succeeds, continue with the next step.
 - 12.2. If the test fails, ensure that the connection URL and credentials are correct. Continue testing until the connection is successful.
13. Click **Next**.
14. In the Servers list, check the managed server created to deploy the new data source.
15. Click **Finish** to display the new data source.
16. Verify that the new data source is associated with the server.
17. In the Change Center pane, click **Activate Change**.

Step 7- Install the adapter

Use the following steps to install the adapter to the Oracle WebLogic Server.

1. If `pcr-000114584_alu8070103` was previously extracted, continue with step 2 below.

If `pcr-000114584_alu8070103` was not previously extracted, extract `pcr-000114584_alu8070103`. The extract directory is referred to as `<ZIP_HOME>`.
2. In the Change Center pane of the Oracle WebLogic administration console, click **Lock & Edit**.
3. In the Domain Structure pane, click **Deployments**.
4. Click **Install**.
5. Click **upload your file(s)**.
6. Select the file to be uploaded as follows:
 - 6.1. In the **Deployment Archive** field, click **Browse**.
 - 6.2. Navigate to the `advancement_services.ear` file in `<ZIP_HOME>`.
 - 6.3. Select the file and click **Open**.
7. Click **Next**.
8. Select the adapter ear file from the list.
9. Click **Next**.
10. Select **Install this deployment as an application**.
11. Click **Next**.



Note: The Select Targets page or the Optional Settings page is displayed, depending on the domain.

If the Select Targets page is displayed, go to step 12.

If the Optional Settings page is displayed, check your WebLogic server configuration to ensure that a Managed Server is available for

deployment of applications. If a Managed Server is not available, the adapter will be deployed to the Admin Server, which is not a recommended configuration. For more information, consult the Oracle WebLogic Server Documentation Library. Then go to step 14.

12. On the Select Targets page, select the server where the adapter should be deployed.
13. Click **Next**.
14. On the Optional Settings page, enter the following information:

Name	Name for the application (for example, Advancement Services)
DD Only: Use only roles and policies that are defined in the deployment descriptors.	Select the radio button.
Copy this application onto every target for me	Select the radio button.

15. Click **Next**.
16. Select **No, I will review the configuration later**.
17. Click **Finish** to start the deployment. When deployment is completed, the Summary of Deployments page is redisplayed with the newly deployed adapter.
18. In the Change Center pane, click **Activate Changes**.
19. Start the newly deployed application as follows:
 - 19.1. Select the newly deployed adapter.
 - 19.2. Click **Start > Servicing all requests**.
 - 19.3. Click **Yes** to start the application.

Step 8 - Configure the security group and user

Use the following steps to add the `bannerws` group and an administrative user to the adapter. This group and user protect the defined endpoint.

1. In the Change Center pane, click **Lock & Edit**.
2. In the Domain Structure pane, click **Security Realms**.
3. Click **myrealm**.
4. Select the **Users and Groups > Groups** tab.
5. Click **New**.

6. Enter the following information to create a group:

Name	bannerwsGroup
Description	Banner Web Services Administrative Group
Provider	DefaultAuthenticator

7. Click **OK**. The table of groups is redisplayed with the new group.
8. Select the **Users** tab.
9. Click **New**.
10. Enter the following information to create a user:

Name	admin (This is an example. Enter the name of your choice.)
Description	Banner Web Services Administrator
Provider	DefaultAuthenticator
Password	Password for the user being created
Confirm Password	Confirmation of the password

11. Click **OK**. The table of users is redisplayed with the new user.
12. Click the name of the user you just created.
13. Select the **Groups** tab.
14. In the Parent Groups section, select `bannerwsGroup` in the **Available** list and move it to the **Chosen** list.
15. Click **Save**.
16. In the Change Center pane, click **Activate Changes**.

Step 9 - Enable schema validation (optional)

Validating XML request and response messages for each web service invocation degrades system performance. For this reason, schema validation is turned off by default. To enable schema validation, you must set system property `BANNERWS_SCHEMA_VALIDATION` with a value of `true` for the instance where the adapter is installed. Use the following steps to enable schema validation.

1. In the Change Center pane, click **Lock & Edit**.
2. In the Domain Structure pane, click the name of the domain.
3. Select the **EJBs** tab and add the following value in the **Append Java Compiler Options** field:

```
-DBANNERWS_SCHEMA_VALIDATION=true
```


4. Click **Save**.
5. In the Change Center pane, click **Activate Changes**.

Step 10 - Update the WSDL endpoint URL (optional)

SOAP-based web services are described via a WSDL (Web Services Definition Language) document. WSDL is an XML format for describing network services as endpoints. A WSDL document contains information about the messages that can be exchanged with the web service, the network protocol supported, and the location of the service expressed as a URL.

The WSDL for the Banner Advancement Web Services Adapter (`banner_giving_history_service.wsdl`) is provided in the `pcr-000114584_alu8070103` patch. This file can be used to generate client-side code for accessing the web service. Third-party testing tools, such as soapUI (www.soapui.org), can also use this file to generate XML-based test cases and test suites for testing the web service.

If you plan to use the WSDL document for development or testing, you must update the document to reflect the actual location of the web service (that is, the service endpoint). Once updated, the file can be transferred to a local computer or shared network location to be used, as needed, by development and test tools that require the WSDL document. The WSDL document should *not* be deployed to the application server.

Use the following steps to configure the WSDL document so it references the proper endpoint URL.

1. Locate `banner_giving_history_service.wsdl` in the `pcr-000114584_alu8070103` patch.
2. Open the file with a text editor or XML editor.
3. Locate the `<service>` tag in the WSDL document. The `<soap:address location>` attribute under this tag contains the following tokenized dummy URL string:

```
http://@@@hostname@@@:@@@port@@@/advancement/v8_2
```
4. Modify the dummy URL to reflect the correct host name and port for your installation.
5. If you changed the default URL string (`/advancement`) during installation, make the corresponding change in the dummy URL.
6. Save the file.

Example

Updating the dummy URL to reflect a deployment to a server named `OAS_Test` that listens on port 443 has the following result:

```
http://oas_test.somewhere.edu:443/advancement/v8_2
```

If the default URL string was changed from `/advancement` to `/advancement/bws_test`, the URL would also need to be updated as follows:

```
http://oas_test.somewhere.edu:443/advancement/  
bws_test/v8_2
```

With these modifications in place, testing and development tools can use the WSDL document to generate code for accessing the specific URL.

Step 11 - Test the Banner Advancement web service (optional)

Once the adapter is installed and the appropriate WSDL file is modified, you should test the exposed Banner Advancement web service. This step is optional, but highly recommended. The degree of testing depends on the data and tools that your institution uses. For this reason, this document provides testing guidelines rather than specific steps.

You should use an interactive testing tool that provides visibility into web service request and response messages. Such a tool helps functional and technical users understand the options that are available for a Banner web service and their effect on the content of response messages. You can create schema-compliant XML messages for your specific environment, quickly inspect the results, modify web service settings in Banner (if necessary), and execute the same request to see the resulting differences.

Several commercial and open source web service testing tools provide these capabilities. One open source tool is eviware soapUI (www.soapui.org). This tool uses a WSDL document and associated XML schema as input to generate compliant request message stubs to which data can be added for calling the associated service. Response messages are displayed in an adjacent window for easy visibility. The tool also allows individual service requests to be tied together to form larger test suites.

Manually creating web service requests and visually inspecting responses provide practical insight into the data returned by a web service under specific conditions. In addition, issues with the underlying Banner configuration for a web service are noticed more readily.

Refer to [“Test cases” on page 22](#) for sample test cases for the GetGivingHistory web service.

GetGivingHistory Web Service

The GetGivingHistory web service allows external systems to request giving history information for a donor who is set up in Banner®. The web service returns summary giving information and a list of individual gifts for the donor. The results are filtered as follows:

- The Outstanding Pledges total includes those pledges that are allowed to be displayed on the web. The **Web Indicator** defined for each pledge type on the Pledge Type Validation (ATVPLDG) page determines which pledges are allowed to be displayed on the web.
- Details are displayed for those gifts that are allowed to be displayed on the web. The **Web Indicator** defined for each gift type on the Gift/Payment Type Validation (ATVGIFT) page determines which gifts are allowed to be displayed on the web.

This web service supports integration with iModules Encompass, but can be used for other purposes.

Message exchange

The GetGivingHistory web service exchanges the following messages:

- GivingHistoryRequest
- GivingHistoryResponse

GivingHistoryRequest

An external system uses the GivingHistoryRequest message to request giving history information from Banner. The only input parameter is the `ConstituentID`.

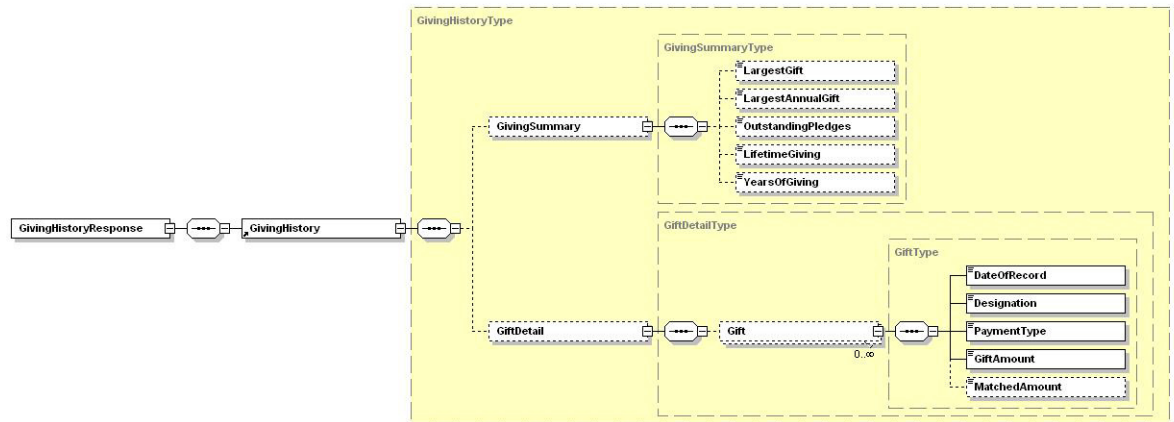
The following diagram shows the structure of the GivingHistoryRequest message schema.



GivingHistoryResponse

If the `ConstituentID` is found, the GetGivingHistory web service returns giving history in the GivingHistoryResponse message. This message contains both summary and

detailed information. The following diagram shows the structure of the GivingHistoryResponse message schema.



If the `ConstituentID` is found but no giving history exists, the web service returns an empty `GivingHistory` element.

SOAP fault messages

If a valid response cannot be created as a `GivingHistoryResponse` message, a SOAP fault message is returned. Situations that might cause a SOAP fault message include the following:

- The `ConstituentID` provided in the `GivingHistoryRequest` message is not a valid Banner PIDM (`SPRIDEN_PIDM`).
- A `ConstituentID` is not supplied in the `GivingHistoryRequest` message.
- A network, database, or other technical issue occurs.

Message mapping to Banner

The following tables provide a mapping between the message elements/attributes and Banner columns. The vertical lines represent the nesting of the attributes inside the elements. Elements can also nest inside other elements.

GivingHistoryRequest

Element/Attribute	Database Mapping
GivingHistoryRequest	
ConstituentID	SPRIDEN_PIDM

GivingHistoryResponse

Element/Attribute	Database Mapping
GivingHistoryResponse	
GivingHistory	
GivingSummary	
LargestGift	APBGHIS_HIGH_GIFT_AMT
LargestAnnualGift	Calculated
OutstandingPledges	Calculated
LifetimeGiving	Calculated
YearsOfGiving	Calculated
GiftDetail	
Gift	
DateOfRecord	AGBGIFT_GIFT_DATE
Designation	AGRGDES_DESG
PaymentType	ATVGIFT_DESC (AGBGIFT_GIFT_CODE)
GiftAmount	AGRGDES_AMT
MatchedAmount	SUM(AGRMDES_AMT)

Intended usage

Partner applications such as iModules Encompass use the GetGivingHistory web service to display giving history to donors in a portal-based environment.

Setup requirements

The following setups determine which pledge and gift information is displayed to a donor:

- The **Web Indicator** flag defined for each pledge type on the Pledge Type Validation (ATVPLDG) page determines which pledges are included in the Outstanding Pledges total that is displayed to a donor. If the indicator is selected for a type code, the web service includes pledges with that type code in the total. If the indicator is not selected for a type code, the web service excludes pledges with that type code from the total.
- The **Web Indicator** flag defined for each gift type on the Gift/Payment Type Validation (ATVGIFT) page determines which gift details are displayed to a donor. If the indicator is selected for a type code, the web service displays details for gifts with that type code. If the indicator is not selected for a type code, the web service excludes details for gifts with that type code.

Test cases

After the Banner Advancement Web Services Adapter is installed, you should test the exposed GetGivingHistory web service. Testing is optional but highly recommended. The degree of testing depends on the data and tools that your institution uses.

The GetGivingHistory web service has one input parameter, the `SPRIDEN_PIDM` of the advancement individual or advancement organization. In the web service request, the XML tag is titled `ConstituentID`.

The following tests can help you understand the web service. These tests are examples only. They do not reflect actual data in your database.

Advancement individual with giving history

Test GetGivingHistory using the `SPRIDEN_PIDM` of an advancement individual with known giving history.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:sungardhe:enterprise:advancement:messages">
  <soapenv:Header/>
  <soapenv:Body><urn:GivingHistoryRequest>
    <urn:ConstituentID>301</urn:ConstituentID>
  </urn:GivingHistoryRequest>
</soapenv:Body>
</soapenv:Envelope>
```

The response message should look similar to the following:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
```

```

    <GivingHistoryResponse
xmlns="urn:sungardhe:enterprise:advancement:messages">
    <GivingHistory>
    <GivingSummary>
    <LargestGift>0</LargestGift>
    <LargestAnnualGift>0</LargestAnnualGift>
    <OutstandingPledges>937.5</OutstandingPledges>
    <LifetimeGiving>1250</LifetimeGiving>
    <YearsOfGiving>1</YearsOfGiving>
    </GivingSummary>
    <GiftDetail>
    <Gift>
    <DateOfRecord>2005-10-10</DateOfRecord>
    <Designation>Allen Family Scholarship</Designation>
    <PaymentType>Check</PaymentType>
    <GiftAmount>312.5</GiftAmount>
    <MatchedAmount>0</MatchedAmount>
    </Gift>
    <Gift>
    <DateOfRecord>2005-10-10</DateOfRecord>
    <Designation>University Campaign Fund</Designation>
    <PaymentType>Check</PaymentType>
    <GiftAmount>312.5</GiftAmount>
    <MatchedAmount>0</MatchedAmount>
    </Gift>
    </GiftDetail>
    </GivingHistory>
    </GivingHistoryResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Advancement individual without giving history

Test GetGivingHistory using the SPRIDEN_PIDM of an advancement individual without giving history.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/
envelope/" xmlns:urn="urn:sungardhe:enterprise:advancement:messages">
  <soapenv:Header/>
  <soapenv:Body><urn:GivingHistoryRequest>
    <urn:ConstituentID>50005</urn:ConstituentID>
  </urn:GivingHistoryRequest>
</soapenv:Body>
</soapenv:Envelope>

```

The response message should look similar to the following:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <GivingHistoryResponse
xmlns="urn:sungardhe:enterprise:advancement:messages">
      <GivingHistory/>
    </GivingHistoryResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Invalid advancement individual ID

Test GetGivingHistory using an invalid SPRIDEN_PIDM.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:sungardhe:enterprise:advancement:messages">
  <soapenv:Header/>
  <soapenv:Body><urn:GivingHistoryRequest>
    <urn:ConstituentID>303</urn:ConstituentID>
  </urn:GivingHistoryRequest>
</soapenv:Body>
</soapenv:Envelope>
```

The response message should look similar to the following:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>Client</faultcode>
      <faultstring>*ERROR* Invalid Constituent ID.
ORA-06512</faultstring>
      <faultactor>/advancement/v8_2</faultactor>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```