DonorPerfect Online XML API

User Manual

SofterWare, Inc.

Version 4.0: September 11, 2014

Table of Contents

1. Executive Summary 6

Intended Audience 6

Version Control 6

2. API Overview 7

What is the DPO API? 7

Common API Uses 7
How do I use the DPO API? 7

API Example Call Breakdown 8

API Connection Credentials 9

Username/Password (&login=) 9
API Key (&apikey=) 9

XML Response Format 10

API Administration and Limitations 11

Logging 11
Restrictions 11
No Blocking 11

3. Dynamic Queries 12

SELECT Statements 12

Example: 12

DPO Table List 14

4. Predefined Procedures 15

dp_donorsearch 15

Parameters: 15
Sample Call: 15
Returns: 15
Notes: 16

dp_savedonor 17

Parameters: 17
Sample Call: 18
Returns: 18
Notes: 18

dp_gifts 19

Parameters: 19
Sample Call: 19
Returns: 19
Notes: 19

dp_savegift 20

Parameters: 20
Sample Call: 21
Returns: 21
Notes: 21

dp_savepledge 22

Parameters: 22
Sample Call: 23
Returns: 23
Notes: 23

dp_saveotherinfo 24

Parameters: 24
Sample Call: 24
Returns: 24
Notes: 24

dp_save_udf_xml 25

Parameters: 25
Sample Call: 25
Returns: 25
Notes: 25

dp_saveflag_xml 26

Parameters: 26
Sample Call: 26
Returns: 26
Notes: 26

dp_delflags_xml 28

Parameters: 28
Sample Call: 28
Returns: 28
Notes: 28

dp_savecontact 29

Parameters: 29
Sample Call: 29
Returns: 30
Notes: 30

dp_PaymentMethod_Insert 31

Parameters: 31
Sample Call: 31

Returns: 32
Notes: 32

5. Support and Implementation Services 34

1. Executive Summary

The DonorPerfect Online Application Programming Interface (DPO API) makes it possible for you to integrate DonorPerfect Online with your existing web environment. You can, for example, access donor and gift information on secure (login protected) portions of your intranet.

The API proves secure connections to the DPO database, allowing read and write access in real time.

Common uses of the API allow developers to create content rich applications for web sites to display donor and gift information. For example, a religious organization may choose to display a list of church contact records from the DonorPerfect Online database on their website. Then, when the contact records are updated in the DonorPerfect Online database, the website will automatically display the updated records. Extending the same scenario, the API could also be configured to allow users to update the contact records from their website.

Another example of the use of the API is the updating of DPO contact information by the ConstantContact®[1] application. When ConstantContact® sends out an email, the DPO donor contact record is updated with the email title, status and date information. This particular API interaction is set up automatically when you purchase the ConstantContact add-on feature in DPO.

For more information, please contact your Account Manager at 800-220-8111.

Intended Audience

This document is written for customers who currently have an existing DonorPerfect Online license, or for those developers that wish to integrate their third party applications with the overall DPO CRM solution.

Version Control

This document has been updated to Version 4.0 to correspond to DonorPerfect Online Version 2014.03.

- This update corrects error in dp_savecontact example to show use of 'null' in this field

[1] ConstantContact is a registered trademark of ConstantContact Inc.

2. API Overview

What is the DPO API?

The DPO API is a set of calls and parameters that one can use to interface external applications with DPO. The API uses XML to return data back to the client, and these XML data 'streams' can be easily parsed using many popular XML parsers. Simply put, the DPO API is a way for programmers and webmasters to make their existing websites or any other applications synchronize with DPO on a variety of levels.

Common API Uses

The DPO API includes the most commonly requested functions such as:

- Saving Donor/Constituent Information – includes demographic Name, Address, Email, Phone Numbers, etc.
- Saving a Donor/Constituent's extended information (User Defined Fields) – Unlimited ability to save to any field in the DPO system at the donor/constituent level- including ones that are created by the client.
- Retrieving a Donor/Constituent's record from the database – retrieves all demographic information and User Defined Fields.
- Searching for a Donor/Constituent, and returning the Donor_ID.
- Saving a Gift – saves all aspects of a gift or pledge transaction, including Date, Start Date, Amount, Gift Type, General Ledger, Solicitation, Thank You Letter Type, Memo/Comments, etc.
- Saving a Gift's extended information (User Defined Fields) – unlimited ability to save any field at the Gift/Pledge level.
- Listing Gifts for a selected Donor/Constituent – retrieves Gift or Pledge Transactions for the selected Donor.

The list of these functions is constantly expanding as clients request more commonly used functions exposed to the API. It is also possible to have a custom call designed for your organization, and it has to be arranged through your Account Manager.

How do I use the DPO API?

The DPO API can be accessed via HTTPS (Hyper Text Transfer Protocol Secure) requests made to a specific page located on the secure DPO web server. Anyone who has familiarity with XML and has possibly used an outside credit card processor (which requires an HTTPS request) should be comfortable with using the DPO API. XML is a structured data set that is sent from our DPO system to your application (website, etc.) that is then translated to a meaningful form whether for the web user to view or to be stored by your application (database, file, email, etc.). Some calls are used exclusively to save information in DPO such as gift, pledge or event information.

API Example Call Breakdown

A sample HTTPS request is analyzed below. This particular one returns all the gifts that are in the DPO system for a donor with id=1269525:

```
https://www.donorperfect.net/prod/xmlrequest.asp?action=dp_
gifts&params=1269525&login=xxx&pass=yyy
```

Each XML API call has a number of key components:

1. ***https://www.donorperfect.net/prod/xmlrequest.asp*** - a constant location of the page that accepts the parameters and returns the requested XML information.
2. ***action=dp_gifts*** – action (function) that one would like to perform.
3. ***params=1269525*** – parameters listing. This particular example only has one parameter but in most cases there will be more than one for other functions (especially that deal with saving information into DPO). The parameters must be listed in order and separated by a comma (,). Some parameters are numeric, some are date/time and some are text.
4. ***login=xxx*** – where “xxx” is the username/login provided to you by your account manager for communication with DPO via XML API
5. ***pass=yyy*** – where “yyy” is the password provided to you by your account manager for communication with DPO via XML API

NOTE: See following section on API Connection Credentials for information on the alternative API Key connection method.

API Connection Credentials

There are two methods for supplying connection credentials to DonorPerfect Online (DPO). Either connection method will work, but the new API Key method has the benefit of enhanced security through the use of long (over 100 characters) and very strong key values:

1. Using a username and corresponding password value from a user created within the DPO user interface (Settings > User Security)
2. Using an API Key value provided by DonorPerfect

Username/Password (&login=)

This is the traditional method for supplying connection credentials. A sample HTTPS request is analyzed below. This particular one returns all the gifts that are in the DPO system for a donor with id=1269525:

```
https://www.donorperfect.net/prod/xmlrequest.asp?action=dp_
gifts&params=1269525&login=xxx&pass=yyy
```

Note that the login credentials can be supplied either at the beginning (i.e.; after *… https://www.donorperfect.net/prod/xmlrequest .asp?*) or the end of the command. This second example shows the &login credentials at the beginning of the command specific part of the string. :

```
https://www.donorperfect.net/prod/xmlrequest.asp?login=xxx&
pass=yyy&action=dp_gifts&params=1269525
```

API Key (&apikey=)

This connection method has the benefit of providing enhanced security through the use of very strong passwords of over one hundred characters in length.

A sample HTTPS request is analyzed below. This particular one returns all the gifts that are in the DPO system for a donor with id=1269525. The actual key values are much longer than the partial key shown here:

```
https://www.donorperfect.net/prod/xmlrequest.asp?apikey=
95LTf7cfVoBJnHqILHv1jN2ZYbdBR5U……
&action=dp_gifts&params=1269525
```

XML Response Format

Each XML API call has a number of key components:

If you were to copy and paste the complete HTTP request found above into the Address area in a browser and press Enter (Assuming the login and password was correct), you would see the actual XML data being returned from DPO. This will only work if all the items in the call (1 through 5) are valid and current.

Note that the returned values are quoted within the attributes **value** parameter.

```
- <result>
  - <record>
      <field name="gift_date2" id="gift_date2" value="09/30/2005" />
      <field name="amount" id="amount" value="5000" />
      <field name="total" id="total" value="0" />
      <field name="sub_solicit_code" id="sub_solicit_code" value="" />
      <field name="campaign" id="campaign" value="KA0905" />
      <field name="balance" id="balance" value="0" />
      <field name="gl" id="gl" value="" />
      <field name="solicit_code" id="solicit_code" value="0223AF" />
      <field name="reference" id="reference" value="56724" />
      <field name="record_type" id="record_type" value="G" />
      <field name="gift_id" id="gift_id" value="384441" />
      <field name="donor_id" id="donor_id" value="1269525" />
    </record>
  - <record>
      <field name="gift_date2" id="gift_date2" value="09/28/2005" />
      <field name="amount" id="amount" value="5000" />
      <field name="total" id="total" value="0" />
      <field name="sub_solicit_code" id="sub_solicit_code" value="" />
      <field name="campaign" id="campaign" value="KA0905" />
      <field name="balance" id="balance" value="0" />
      <field name="gl" id="gl" value="" />
      <field name="solicit_code" id="solicit_code" value="0223AF" />
      <field name="reference" id="reference" value="56724" />
      <field name="record_type" id="record_type" value="G" />
      <field name="gift_id" id="gift_id" value="384382" />
      <field name="donor_id" id="donor_id" value="1269525" />
    </record>
  <result>
```

API Administration and Limitations

Logging

DonorPerfect Online logs all API requests for record keeping and administration purposes. These logs are kept to help troubleshoot common problems as well as to identify fraudulent or suspicious activity.

Restrictions

In addition, the calls are limited to return a maximum amount of data so that overall application performance will not be affected. Since some calls allow multiple transactions to be returned, these calls are limited to 5000 transactions per call. The limit is set as a default, though can be changed on an individual client basis depending on need and scheduling concerns. Finally, there is no limit to the number of consecutive calls to the server.

SmartAction triggers are not activated by any API actions (e.g.; donor/gift additions or updates).

No Blocking

The API is designed such that the API user is exempted from the DPO restriction on maximum number of concurrent users, so API calls are never blocked regardless of the number of concurrent users connected to the DPO system.

3. Dynamic Queries

SELECT Statements

This query format is based on the ANSI[2] SQL implementation of Structured Query Language (SQL). SQL is the universal language used to communicate with databases.

The dynamic query feature of the DPO API allows users to build API calls using SQL syntax.

Example:

DPO contains a table called DP which contains donor information including people's first and last names and their donor ID numbers. The column names are DONOR_ID, FIRST_NAME and LAST_NAME. The values are capitalized for clarity, but the query does not expect capitalization.

An SQL query to retrieve DONOR_ID, FIRST_NAME and LAST_NAME from the DP table for people with the last name 'Bacon'would look like this:

```
Select donor_id, first_name, last_name from dp where
last_name = 'bacon'
```

This SQL gets placed into an XML API call and looks like this:

```
https://www.donorperfect.net/prod/xmlrequest.asp?action=
Select donor_id, first_name, last_name from dp where
last_name = 'bacon' &login=xxx &pass=yyy
```

If there is a donor in your system named Kevin Bacon and the above command was submitted through a web browser with the appropriate username and password, the raw (unprocessed) XML result will look like this:

```
- <result>
  - <record>
      <field name="donor_id" id="donor_id" value="68" />
      <field name="first_name" id="first_name"
        value="Kevin" />
      <field name="last_name" id="last_name"
        value="Bacon" />
    </record>
  </result>
```

[2] ANSI is the American National Standards Institute. MSSQL uses ANSI SQL.

To retrieve all fields from the DP table, you would use the asterisk wildcard (*) instead of field names.

NOTE: This can be a good way to identify all the field names in a table BUT make sure you limit the result set by specifying a particular donor_id or gift_id as appropriate. Otherwise your system will be burdened with a query retrieving thousands of data values. The query to retrieve all DP table field data for Kevin (donor_id 68) would look like this:

```
https://www.donorperfect.net/prod/xmlrequest.asp?action=
Select * from dp where donor_id = 68 &login=xxx &pass=yyy
```

DPO Table List

This is a list of tables in DPO you can run Dynamic Queries On

Table Name	Comments
DP	Contains donor information
DPUDF	User Defined Fields associated with the DP table
DPUSERMULTIVALUES	Checkbox field values
DPGIFT	Contains gift information
DPGIFTUDF	User Defined Fields associated with the DPGIFT table
DPADDRESS	Allows storages of additional addresses for donors (in DPO Address tab)
DPLINK	Associated with values seen in the DPO Link tab
DPOTHERINFO	Associated with values seen in the DPO Other Info tab
DPOTHERINFOUDF	User Defined fields added to the DPO Other Info tab
DPPAYMENTMETHOD	Contains payment information associated with EFT Transactions for systems where this feature has been enabled. Two examples of the data returned from this table are shown in the section on the dp_paymentMethodInsert command.

4. Predefined Procedures

dp_donorsearch

Searching for a Donor—used to search for the donor based on a number of search criteria (similar to the search functionality offered in DPO). Use "%" for wildcards.

Parameters:

Parameter	Type	Notes
@donor_id	numeric	
@last_name	NVarchar(100)	
@first_name	NVarchar(50)	
@opt_line	NVarchar(100)	
@address	NVarchar(100)	
@city	NVarchar(50)	
@state	NVarchar(20)	
@zip	NVarchar(50)	
@country	NVarchar(50)	
@filter_id	numeric	
@user_id	NVarchar(20)	

Sample Call:

```
https://www.donorperfect.net/prod/xmlrequest.asp?action=dp_
donorsearch&params=null,'Pa%','Ori%',null, null, null,
null, null, null, null, null  &login=xxx &pass=yyy
```

Returns:

```
<result>
-<record>
<field name="donor_id" id="donor_id" value="147"/>
<field name="first_name" id="first_name" value="Orianthi"/>
<field name="last_name" id="last_name" value="Panagaris"/>
<field name="title" id="title" value=""/>
<field name="suffix" id="suffix" value=""/>
<field name="address" id="address" value="4240 Main St."/>
<field name="opt_line" id="opt_line" value=""/>
```

```
<field name="city" id="city" value="North Woodstock"/>
<field name="state" id="state" value="NH"/>
<field name="zip" id="zip" value="12345"/>
<field name="gifts" id="gifts" value="0"/>
<field name="gift_total" id="gift_total" value="0"/>
<field name="address2" id="address2" value=""/>
<field name="donor_name" id="donor_name" value="Orianthi
Panagaris"/>
<field name="city_state_zip" id="city_state_zip"
value="North Woodstock, NH 12345"/>
</record>
</result>
```

Notes:

- If you require different fields to be returned, if you need the city_state_zip or donor_name fields to appear as separate fields then use a Dynamic Query SELECT statement instead to achieve the desired result set.

dp_savedonor

Saving a New/Existing Donor—used to save changes to the existing donor/constituent or save the new donor/constituent into the DPO system.

Parameters:

Parameter	Type	Notes
@donor_id	numeric	Enter 0 (zero) to create a new donor/constituent record or an existing donor_id
@first_name	NVarchar(100)	
@last_name	NVarchar(150)	
@middle_name	NVarchar(100)	
@suffix	NVarchar(100)	
@title	NVarchar(100)	
@salutation	NVarchar(100)	
@prof_title	NVarchar(100)	
@opt_line	NVarchar(100)	
@address	NVarchar(100)	
@address2	NVarchar(100)	
@city	NVarchar(75)	
@state	NVarchar(50)	
@zip	NVarchar(50)	
@country	NVarchar(50),	
@address_type	NVarchar(30),	
@home_phone	NVarchar(75),	
@business_phone	NVarchar(75),	
@fax_phone	NVarchar(75),	
@mobile_phone	NVarchar(75),	
@email	NVarchar(100),	
@org_rec	NVarchar(1),	
@donor_type	NVarchar(30),	

@nomail	NVarchar(1),	
@nomail_reason	NVarchar(30),	
@narrative	text,	
@user_id	NVarchar(20)	

Sample Call:

```
https://www.donorperfect.net/prod/xmlrequest.asp?action=dp_
savedonor&params=0,'Orianthi', 'Panagaris', null, null,
null, null, null, null, '4240 Main St.', null, 'North
Woodstock', 'NH', '12345', 'US', null, '205-555-1212',
null, null, '205-987-6543', 'orianthi@bigstar.com', null,
null,'N', null, null,'API User' &login=xxx &pass=yyy
```

Returns:

This command returns the donor ID value of the new/updated donor.

```
<result>
      <record>
            <field name="" id="" value="147" />
      </record>
</result>
```

Notes:

- 147 is the donor_id of the created donor
- Do not have any spaces in the first part of your command (https://www.donorperfect.net/prod/xmlrequest.asp?action=dp_savedonor&params=) otherwise the API will return an error indicating that the parameter @donor_id was not supplied.
- The null values act as placeholders and must be included in your command
- Character values and date values must be enclosed in single quotes (e.g; '11/01/2010')
- The @user_id value is just there for transaction reference purposes and is not validated against the list of DPO users.

dp_gifts

This procedure returns a predefined set of fields associated with all gifts given by the specified donor.

Parameters:

Parameter	Type	Notes
@donor_id	numeric	

Sample Call:

```
https://www.donorperfect.net/prod/xmlrequest.asp?action=dp_
gifts&params=147 &login=xxx &pass=yyy
```

Returns:

```
<result>
-<record>
<field name="gift_date2" id="gift_date2"
value="10/27/2010"/>
<field name="amount" id="amount" value="14.98"/>
<field name="total" id="total" value="0"/>
<field name="sub_solicit_code" id="sub_solicit_code"
value="TS"/>
<field name="campaign" id="campaign" value=""/>
<field name="balance" id="balance" value="0"/>
<field name="gl" id="gl" value="NO Rcpt: BOOKS & TAPES"/>
<field name="solicit_code" id="solicit_code" value="BQ10"/>
<field name="reference" id="reference" value=""/>
<field name="record_type" id="record_type" value="G"/>
<field name="gift_id" id="gift_id" value="10230"/>
<field name="donor_id" id="donor_id" value="147"/>
<field name="anongift" id="anongift" value=""/>
<field name="gift_aid_date" id="gift_aid_date" value=""/>
</record>
</result>
```

Notes:

- If you need different or additional fields returned from the system, use a Dynamic Query SELECT command.

dp_savegift

This procedure is used to save changes to an existing gift or to save a new gift into the DPO system

Parameters:

Parameter	Type	Notes
@gift_id	numeric	Enter 0 in this field to create a new gift or the gift ID of an existing gift.
@donor_id	numeric	
@record_type	NVarchar(30)	'G' for Gift, 'P' for Pledge
@gift_date	datetime	
@amount	money	
@gl_code	NVarchar(30)	
@solicit_code	NVarchar(30)	
@sub_solicit_code	NVarchar(30)	
@gift_type	NVarchar(30)	
@split_gift	NVarchar(1)	
@pledge_payment	NVarchar(1)	
@reference	NVarchar(25)	
@memory_honor	NVarchar(30)	
@gfname	NVarchar(50)	
@glname	NVarchar(75)	
@fmv	money	
@batch_no	numeric	
@gift_narrative	NVarchar(3000)	
@ty_letter_no	NVarchar(30)	
@glink	numeric	
@plink	numeric	
@nocalc	NVarchar(1)	

Parameter	Type	Notes
@receipt	NVarchar(1)	
@old_amount	money	
@user_id	NVarchar(20)	
@campaign	NVarchar(30) = NULL	
@membership_type	NVarchar(30) = NULL	
@membership_level	NVarchar(30) = NULL	
@membership_enr_date	datetime = NULL	
@membership_exp_date	datetime = NULL	
@membership_link_ID	numeric = NULL	
@address_id	numeric = NULL	

Sample Call:

```
https://www.donorperfect.net/prod/xmlrequest.asp?action=dp_savegift&params=0,147,'G','10/27/2010',14.98,'4540-N','BQ10','TS','VISAIC','N','N',null,null,null,null,0,0,'In memory of my friend', null, null, null, 'N', 'N', null, 'API User'&login=xxx&pass=yyy
```

Returns:

```
<result>
      <record>
            <field name="" id="" value="10230" />
      </record>
</result>
```

Notes:

- 10230 is the gift_id of the created gift

dp_savepledge

This procedure is used to create, or save changes to, a pledge. It is not used for pledge payments. In DPO, there is a parent pledge (which this command is used to create) that shows up in the DPO pledges tab. Then, when pledge payments are made, they are created as gifts (record_type='G') using the dp_savegift procedure with a gift_type of 'G' like a regular gift, but add in a '**plink'** value with the gift_id of the parent pledge.

Parameters:

Parameter	Type	Notes
@gift_id	numeric	Enter 0 in this field to create a new pledge or the gift ID of an existing pledge.
@donor_id	numeric	Enter the donor_id of the person for whom the pledge is being created/updated
@gift_date	datetime	
@start_date	datetime	
@total	money	Enter either the total amount to be pledged (the sum of all the expected payment amounts) or enter 0 (zero) if the pledge amount is to be collected ad infinitum
@bill	money	Enter the individual monthly/quarterly/annual billing amount
@frequency	NVarchar (30)	Enter one of: M (monthly), Q (quarterly), S (semi-annually), A (annually)
@reminder	NVarchar (1)	Sets the pledge reminder flag
@gl_code	NVarchar(30)	
@solicit_code	NVarchar(30)	
@initial_payment	NVarchar (1)	Set to ''Y' for intial payment, otherwise 'N'
@sub_solicit_code	NVarchar(30)	
@writeoff_amount,	money	
@writeoff_date	datetime	
@user_id	NNVarchar(20),	
@campaign	NVarchar(30)	Or NULL
@membership_type	NVarchar(30)	Or NULL
@membership_level	NVarchar(30)	Or NULL

Parameter	Type	Notes
@membership_enr_date	datetime	Or NULL
@membership_exp_date	datetime	Or NULL
@membership_link_ID	numeric	Or NULL
@address_id	numeric	Or NULL
@gift_narrative	NVarchar(3000)	Or NULL
@ty_letter_no	**NVarchar(30)**	Or NULL
@vault_id	NVarchar(55)	Or NULL
@receipt_delivery_g	NVarchar(30)	'E' for email, 'B' for both email and letter, 'L' for letter, 'N' for do not acknowledge or NULL
@contact_id	numeric	Or NULL

Sample Call:

```
https://www.donorperfect.net/prod/xmlrequest.asp?action=dp_
savepledge&params=0, 147,'10/12/2012',
'10/15/2012',600.00,30.00,'M','Y','4540-N','BQ10','N','TS',
0.00,'12/29/2013','API User','REG', NULL, NULL, NULL, NULL,
NULL, NULL,'Gift narrative', 'TY', '123321','E',NULL
&login=xxx&pass=yyy
```

Returns:

```
<result>
      <record>
            <field name="" id="" value="10230" />
      </record>
</result>
```

Notes:

- 10230 is the gift_id of the created gift

dp_saveotherinfo

This procedure saves fields to the dpotherinfo table. It will create a new or updated '**Other Info'** record for the specified donor_id.

Parameters:

Parameter	Type	Notes
@other_id	numeric	Enter 0 to create a new record or the other_id record number of an existing dpotherinfo record
@donor_id	numeric	Enter the donor_id for whom the record is to be created / updated.
@other_date	Date_time	Format as date('m\/d\/Y,time())
@comments	NVarchar(500)	
@user_id	NVarchar(20)	

Sample Call:

```
https://www.donorperfect.net/prod/xmlrequest.asp?action=dp_
saveotherinfo&params=0,147,'11/18/2010','my comment', 'API
User' &login=xxx &pass=yyy
```

Returns:

```
<result>
-<record>
<field name="" id="" value="160" />
</record>
</result>
```

Notes:

- 160 is the other_id associated with the updated value
- This other_id value can now be used as a @matching_id value to save additional fields associated with the dpotherinfoudf table associated with this entry.

dp_save_udf_xml

This procedure saves a Donor's extended information (User Defined Fields) —used to save changes to the user-defined fields that are custom for each client and are not part of the standard DPO system.

This procedure will save a single parameter for a specified User Defined Field (UDF).

Parameters:

Parameter	Type	Notes
@matching_id	numeric	Specify either a donor_id value if updating a donor record, a gift_id value if updating a gift record or an other_id value if updating a dpotherinfo table value (see dp_saveotherinfo)
@field_name	NVarchar(20)	
@data_type	NVarchar(1)	C- Character, D-Date, N- Numeric
@char_value	NVarchar(2000)	Null if not a Character field
@date_value	datetime	Null if not a Date field
@number_value	numeric (18,4)	Null if not a Number field
@user_id	NVarchar(20)	

Sample Call:

```
https://www.donorperfect.net/prod/xmlrequest.asp?action=dp_
save_udf_xml&params=16013,'ALT_PHONE','C','555-
1212',null,null,'API User' &login=xxx &pass=yyy
```

Returns:

```
<result>
-<record>
<field name="" id="" value=" 16013" />
</record>
</result>
```

Notes:

- 16013 is the donor_id associated with the updated value
- This example assumes that a user defined field called ALT_PHONE already exists in your system

dp_saveflag_xml

This procedure allows you to set flags as shown in the top section of the Main tab. The Flags field is on the Main tab in the DPO user interface.

Flags must have been previously created in Settings > Code Maintenance and the value you set corresponds to the Code value (not the description value).

).

Parameters:

Parameter	Type	Notes
@donor_id	numeric	Specify either a donor_id value if updating a donor record, a gift_id value if updating a gift record or an other_id value if updating a dpotherinfo table value (see dp_saveotherinfo)
@flag	varchar(20)	Use the **code** value associated with the flag. For example, the 'AL', flag in this example had a description value of 'Alumni'.
@user_id	varchar(20)	

Sample Call:

```
https://www.donorperfect.net/prod/xmlrequest.asp?action=dp_
saveflag_xml&params=135,'AL','xmluser'&login=xxx &pass=yyy
```

Returns:

```
<result>
-<record>
<field name="" id="" value=" 135" />
</record>
</result>
```

Notes:

- 135 is the donor_id associated with the updated value
- This example assumes that a user defined field called ALT_PHONE already exists in your system
- If the flag was already set, the system returns an error like this: [Microsoft][ODBC SQL Server Driver][SQL Server]Violation of PRIMARY KEY constraint 'PK_dpflags'. Cannot insert duplicate key in object
- To view the flags set for a specified donor, use this command format:

https://www.donorperfect.net/prod/xmlrequest.asp?action=select * from dpflags where donor_id=135&login='xxx'&pass='yyy'

dp_delflags_xml

This procedure removes (deletes) all flags for the specified donor. Flags are shown on the main donor screen in DPO.

Parameters:

Parameter	Type	Notes
@donor_id	numeric	Specify the donor_id of the donor for whom the flags (all of them) are to be deleted
@user_id	varchar(20)	

Sample Call:

```
https://www.donorperfect.net/prod/xmlrequest.asp?action=dp_
delflags_xml&params=135,'xmluser'&login=xxx &pass=yyy
```

Returns:

```
<result>
<record>
<field name="" id="" value="135"/>
</record>
</result>
```

Notes:

- 135 is the donor_id associated with the deleted flags
- It is not currently possible to delete individual flags for a specified donor. This command deletes all flags set for the specified donor_id.
- To view the flags set for a specified donor, use this command format:

  https://www.donorperfect.net/prod/xmlrequest.asp?action=select * from dpflags where donor_id=135&login='xxx'&pass='yyy'

dp_savecontact

This procedure saves fields to the DPCONTACT table. It will create a new or updated Contact record for the specified donor_id.

Parameters:

Parameter	Type	Notes
@contact_id	numeric	Enter 0 to create a new record or the other_id record number of an existing dpcontact record
@donor_id	Numeric	Enter the Donor ID of the donor for whom the contact record is to be created or retrieved
@activity_code	NVarchar (30)	CODE value for the Activity Code field. See DPO Settings > Code Maintenance > Activity Code / Contact Screen. The required values will be listed in the Code column of the resulting display.
@mailing_code	NVarchar (30)	CODE value for Mailing Code field
@by_whom	NVarchar (30)	CODE value for the By Whom/Contact Screen field in DPO Description value of selected code shows in the 'Assigned To' field of the contact record.
@contact_date	Datetime	Contact / Entry Date field in DPO
@due_date	Datetime	Due Date field in DPO
@due_time	NVarchar	Time field in DPO
@completed_date	Datetime	Completed Date field in DPO
@comment	NVarchar (3000)	Contact Notes field in DPO
@document_path	NVarchar (200)	Type a URL/File Path field in DPO
@user_id	NVarchar (20)	Created by value – not shown in DPO user interface

Sample Call:

```
https://www.donorperfect.net/prod/xmlrequest.asp?action=dp_
savecontact&params=0, 135, 'TE', 'BW02', 'ELF',
'09/13/2012', '09/18/2012', '12:24:00 PM', '09/28/2012',
'Your comment on this contact item.',
'http://api.markwarren.ca', 'APIUser' &login=xxx &pass=yyy
```

Returns:

```
<result>
-<record>
<field name="" id="" value="123 />
</record>
</result>
```

Notes:

- 123 is the contact_id associated with the created value
- In this example, the contact_date is 09/13/2012. Typically the current date. The due_date is 09/18/2012 and the completed_date is 09/28/2012 but you would typically leave this field blank by entering null in this field to allow the DPO User to whom the activity was assigned to mark the item as completed when they performed the required activity: e.g.;

```
https://www.donorperfect.net/prod/xmlrequest.asp?action=dp_savecontact&params=0, 135, 'TE', 'BW02', 'ELF', '09/13/2012', '09/18/2012', '12:24:00 PM', null, 'Your comment on this contact item.', 'http://api.markwarren.ca', 'API User' &login=xxx &pass=yyy
```

dp_PaymentMethod_Insert

This procedure allows insertion of DPO Payment Method values. This table is used on systems with the EFT Transactions feature enabled.

This procedure will save a single parameter for a specified User Defined Field (UDF).

Parameters:

Parameter	Type	Notes
@CustomerVaultID	NVarchar(55)	Enter -0 to create a new Customer Vault ID record
@donor_id	int)	
@IsDefault bit	Bit	Enter 1 if this is will be the default EFT payment method
@AccountType	NVarchar(256)	e.g. 'Visa'
@dpPaymentMethodTypeID	NVarchar(20)	e.g.; 'creditcard'
@CardNumberLastFour	NVarchar(16)	e.g.; '4xxxxxxxxxx1111
@CardExpirationDate	NVarchar(10)	e.g.; '0810'
@BankAccountNumberLastFour	NVarchar(50)	
@NameOnAccount	NVarchar(256)	
@CreatedDate	datetime	
@ModifiedDate	datetime	
@import_id	int	
@created_by	NVarchar(20)	
@modified_by	NVarchar(20)	
@selected_currency	NVarchar(3)	

Sample Call:

```
https://www.donorperfect.net/prod/xmlrequest.asp?action=dp_
paymentmethod_insert&params=0,147,1,'Visa','creditcard','4x
xxxxxxxx1234','0412',null,'Oriana',null,null,null,null,null
,'USD' &login=xxx &pass=yyy
```

Returns:

```
<result>
-<record>
<field name="DpPaymentMethodID" id="DpPaymentMethodID"
value="2"/>
</record>
</result>
```

Notes:

- This table would normally only be populated from an ecommerce API where the DPO system has EFT Transactions enabled.

Here are two samples of XML data retrieved from a Dynamic SELECT Query on the DpPaymentMethod table from a system with EFT Transactions enabled. :

Sample Check Item:

```
- <record>
  <field name="DpPaymentMethodID" id="DpPaymentMethodID"
value="402" />
  <field name="donor_id" id="donor_id" value="123456" />
  <field name="IsDefault" id="IsDefault" value="True" />
  <field name="AccountType" id="AccountType" value="Bank
Account" />
  <field name="dpPaymentMethodTypeID"
id="dpPaymentMethodTypeID" value="check" />
  <field name="CardNumberLastFour" id="CardNumberLastFour"
value="" />
  <field name="CardExpirationDate" id="CardExpirationDate"
value="" />
  <field name="BankAccountNumberLastFour"
id="BankAccountNumberLastFour" value="7xxxx6543" />
  <field name="NameOnAccount" id="NameOnAccount"
value="Arthur Roundtable" />
  <field name="CreatedDate" id="CreatedDate"
value="10/15/2010 12:01:48 PM" />
  <field name="ModifiedDate" id="ModifiedDate"
value="10/15/2010 12:01:46 PM" />
  <field name="CustomerVaultID" id="CustomerVaultID"
value="1742923032" />
  <field name="import_id" id="import_id" value="" />
  <field name="created_by" id="created_by" value="" />
  <field name="modified_by" id="modified_by" value="" />
  <field name="selected_currency" id="selected_currency"
value="CAD" />
  </record>
```

Sample Credit Card Item:

```
- <record>
  <field name="DpPaymentMethodID" id="DpPaymentMethodID"
value="392" />
  <field name="donor_id" id="donor_id" value="21245" />
  <field name="IsDefault" id="IsDefault" value="True" />
  <field name="AccountType" id="AccountType"
value="MasterCard" />
  <field name="dpPaymentMethodTypeID"
id="dpPaymentMethodTypeID" value="creditcard" />
  <field name="CardNumberLastFour" id="CardNumberLastFour"
value="1xxxxxxxxxxx2345" />
  <field name="CardExpirationDate" id="CardExpirationDate"
value="0614" />
  <field name="BankAccountNumberLastFour"
id="BankAccountNumberLastFour" value="" />
  <field name="NameOnAccount" id="NameOnAccount" value=""
/>
  <field name="CreatedDate" id="CreatedDate"
value="10/14/2010 1:11:30 PM" />
  <field name="ModifiedDate" id="ModifiedDate"
value="10/14/2010 1:11:29 PM" />
  <field name="CustomerVaultID" id="CustomerVaultID"
value="1234567890" />
  <field name="import_id" id="import_id" value="" />
  <field name="created_by" id="created_by" value="Diane
Warner" />
  <field name="modified_by" id="modified_by" value="" />
  <field name="selected_currency" id="selected_currency"
value="USD" />
  </record>
```

5. Support and Implementation Services

How to contact DPO API Support:

Mark Warren – XML API Consultant

Email: mwarren@softerware.com (Fastest path for resolution.)

Phone: 1-855-896-5100

- Normal Working Hours: Monday - Friday, 8:30 a.m. to 5:00 p.m. EST.
- Acknowledgement response time for support calls is not guaranteed and is independent of any other support guidelines. Calls are answered in the order they are received and it's reasonable to expect a response within one business day. For emergencies, send an additional email to support@donorperfect.com.

Your purchase of the API/XML service allows you access to our support staff and we can offer the following type of services at no additional charge:

- Assistance with API Call syntax.
- Recommendations or suggestions to accomplish your task.
- Error message explanations
- Restoring API service due to our own outages.

Provision of the DPO API by SofterWare includes:

- The DPO API toolkit (commands)
- The DPO API documentation
- API support as described above

API Support does not include assistance or consulting services on applications developed by the client organization or an authorized third party integrator. These remain the sole responsibility of the client organization.

As a result, additional chargeable services not covered above include:

- Creating, Reviewing or Testing API calls on behalf of the client.
- Debugging API Code created by the client or the client's authorized third party.
- Creating or editing website code to utilize the DPO API.
- Creating or editing any API documentation

Depending on the availability of SofterWare development staff and on the evaluation of any such proposed contract project offers, SofterWare may provide these types of

services to a client by prior agreement with the client. Any such services would be based on a rate of **$100 / hour**.

Please contact your Account Manager for any additional questions regarding our support for the DPO API.