July 1, 2008
Kapil Kaushik

# GFPD FormData Structure

**Receiver/Listener Point of View**

This document includes tables that describe the elements of the XML documents that are used in Form Data responses [soap envelope GFPD point of view]. The schema is the actual authority, but it can be difficult to read, and these tables attempt to show the structure of the final documents, skipping some of the intermediate type definitions.

## 1 Element Summary

This document describes the XML in the SOAP FormDataResponse (body and header).The PGS delivery agent [GFPD] will send the form response in soap envelope to the clients. The soap envelope will include following elements:

| | |
|---|---|
| **SOAP Header Elements** | |
| `Response` | Header element with internal processing information. |
| **Function Body Element** | |
| `FormDataResult` | Body element which contains the FormDataResult |

Special attention should be paid to the "#" column of these tables, which indicates the number of the particular element that may occur. The values in this column should be interpreted as follows:

| # of Elements | Description |
|---|---|
| **1** | required element, and only one is allowed |
| **1-** | required element, there must be at least one, but there may be more |
| 0-1 | optional element, but no more than one allowed |
| 0- | optional element, any number may appear |

The elements are arranged in indented columns to indicate the hierarchy within the XML document – child elements appear indented under the parent element.
Choices in the XML are indicated by a shaded column to the immediate left of the elements which may be selected. Only one of the choice elements may be used.

## 1.1 SOAP Header Element Response

The Response element will be included in responses sent to the client.

| *Tag Name* | | | *#* | *Description* |
|---|---|---|---|---|
| **Response** | | | | |
| | **SenderKey** | | **1** | URI associated with the sender. This currently has no practical significance. |
| | **ReceiverKey** | | 0-1 | URI associated with the receiver. |
| | **RequestID** | | 0-1 | Echoes the value supplied in the Request, if any. |
| | **ProcessingID** | | 0-1 | Internal processing id. This will not ordinarily be of any interest, although it may help PGS research processing errors. |

## 1.2 FormDataResult

The FormDataResult element. Please see FormDataResult in XOA.wsdl.

| *Tag Name* | | | *#* | *Description* |
|---|---|---|---|---|
| **FormDataResult** | | | | |
| | **Status** | | **1** | Status of processing. See Status |
| | **SubmitId** | | 0-1 | Echoes the `SubmitId` in the request, if one was provided, and if possible. |
| | **NumToRetrieve** | | 0-1 | The number of records retrieved. If the requested `NumToRetrieve` was greater than the `TotalCount` value, this value will be the same as `TotalCount`. |
| | **TotalCount** | | 0-1 | The total number of responses. |
| | **StartSequence** | | 0-1 | The start sequence of the response. |
| | **EndSequence** | | 0-1 | The end sequence of the response. |
| | **CampaignName** | | 0-1 | Campaign name associated with the form. |
| | **FormName** | | 0-1 | Name associated with the form |
| | **CharacterSet** | | 0-1 | Characterset associated with the form |
| | **FormFieldInfo** | | 0- | This element will contain the information about the fields of form |
| | | **FieldName** | 1 | Field Name of the form which is encodable string. For details. EncodableStringType) |
| | | **FieldDataType** | 1 | The type of the value that a field can hold. |
| | | **AdditionalInfo** | 0- | Additional information related with form fields |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | **attribute** name | 1 | Name of associated with additional info. |
| | **Response** | | | | 0- | Response data and information about response |
| | | **attribute** sequenceNumber | | | 1 | The sequence number of the response. |
| | | **attribute** lastModifiedTime | | | 1 | Modified time in date Time format |
| | | **FormField** | | | 1- | Form field with values |
| | | | **Name** | | 1 | Form Field Name |
| | | **Choice** | **ExtString** | | 0-1 | Used when the value is represented as string. See EncodableStringType |
| | | | **ExtNumber** | | 0-1 | Used when the value is represented as number.(Integer or floating point) |
| | | | **ExtDateTime** | | 0-1 | Used when the value is represented as date/time. See section DateTime format |
| | | | **ExtBoolean** | | 0-1 | Used if extended data is known to be a Boolean. |
| | | | **ExtBinary** | | 0-1 | Used if data is known to be binary data. |
| | | | | **FileName** | 0-1 | Name of the data file if any.Encodable String see EncodableStringType |
| | | | | **Data** | 1 | The binary data in base64 format. |

## *1.3 Status*

| *Tag Name* | | | | *#* | *Description* |
|---|---|---|---|---|---|
| **Status** | | | | | |
| | **StatusCode** | | | **1** | Contents will be an integer value. The value 0 indicates successful processing of the message, and any non-zero value indicates something less than complete successful processing. So, data should be ignored if its StatusCode value is not 0.<br>Success -0<br>Failure -1<br>Partial -2<br>In general, a value of 1 will indicate a failure or problem, and 2 may indicate partial success – although again the status of each part must be examined. |
| | **StatusMessage** | | | 0-1 | If present, this will contain a text message associated with the `StatusCode`. This is intended as an aid only – users should rely only on the `StatusCode`, and should not count on any particular text string or pattern in `StatusMessage`. |
| | **ErrorList** | | | 0-1 | If present, this will contain one or more *`Error`* elements with more information about errors that were encountered. |
| | | **Error** | | **1-** | Contains error information |
| | | | **ErrorCode** | **1** | An integer value associated with the particular error. |

| | | | | | |
|---|---|---|---|---|---|
| | | | **ErrorMessage** | 0-1 | Text relating to the error. |
| | **SubmissionTime** | | | **1** | The time the request (or part of the request) was received for processing. |
| | **CompletionTime** | | | **1** | The time that processing of the request (or part of the request was completed. Note that there may be a delay in communicating the result of the task back to the user. |

# 2 EncodableStringType

PGS is working to support UTF-8 for all data. A legacy feature is that certain data fields associated with jobs and destinations may be represented in specific character sets, and have that character set information stored with them These fields are now accessible as EncodableStringType in the schema, which includes an attribute “b64charset”. The value of the field must be supplied in base64-encoded form, and the b64charset attribute specifies the character set the data is actually in. A special b64charset value “binary” simply specifies that no character set or conversion is to be used on the data – after being decoded from base64, the data is used as is.

# 3 DateTime format

Many elements in the schema are defined using the xs:dateTime format. This format is specified in the W3C XML Schema language. The following description is from the W3C Recommendation, ‘XML Schema Part 2: Datatypes Second Edition’:
The ·lexical space· of dateTime consists of finite-length sequences of characters of the form: '-'? yyyy '-' mm '-' dd 'T' hh ':' mm ':' ss ('.' s+)? (zzzzzz)?, where
'-'? yyyy is a four-or-more digit optionally negative-signed numeral that represents the year; if more than four digits, leading zeros are prohibited, and '0000' is prohibited (see the Note above (§3.2.7); also note that a plus sign is not permitted);
the remaining '-'s are separators between parts of the date portion;
the first mm is a two-digit numeral that represents the month;
dd is a two-digit numeral that represents the day;
'T' is a separator indicating that time-of-day follows;
hh is a two-digit numeral that represents the hour; '24' is permitted if the minutes and seconds represented are zero, and the dateTime value so represented is the first instant of the following day (the hour property of a dateTime object in the ·value space· cannot have a value greater than 23);
':' is a separator between parts of the time-of-day portion;
the second mm is a two-digit numeral that represents the minute;
ss is a two-integer-digit numeral that represents the whole seconds;
'.' s+ (if present) represents the fractional seconds;
zzzzzz (if present) represents the timezone (as described below).
For example, 2002-10-10T12:00:00-05:00 (noon on 10 October 2002, Central Daylight Savings Time as well as Eastern Standard Time in the U.S.) is 2002-10-10T17:00:00Z, five hours later than 2002-10-10T12:00:00Z.
And from the description of “timezones”:
Time zones are durations with (integer-valued) hour and minute properties (with the hour magnitude limited to at most 14, and the minute magnitude limited to at most 59, except that if the hour magnitude is 14, the minute value must be 0); they may be both positive and both negative.
The lexical representation of a timezone is a string of the form: (('+' | '-') hh ':' mm) | 'Z', where
hh is a two-digit numeral (with leading zeros as required) that represents the hours,
mm is a two-digit numeral that represents the minutes,
'+' indicates a nonnegative duration,
'-' indicates a nonpositive duration.
The mapping so defined is one-to-one, except that '+00:00', '-00:00', and 'Z' all represent the same zero-length duration timezone, UTC; 'Z' is its canonical representation.

So date/time values in FormDataResult will look something like this:
2002-10-10T12:00:00-05:00
2002-10-10T17:00:00Z
The “timezone” as used in this format is not quite the same as the common meaning of the term, which entails some geographic association and additional rules like Daylight Savings Time. Consequently, the dateTime values are basically used to specify a point in time, and may use any appropriate “timezone” value as long as it turns out to represent the right time.