HMRC Data Provisioning Service API

Andy Greener, Business Tax Programme, HMRC
Graham Jenkins, Capgemini

Version 3.0 – 20/10/2018

Version History

Version	*Date*	*Author*	*Comments*
1.0	1/08/2005	Andy Greener	Baselined (derived from internal API definition v1.1)
1.1	27/9/05	Andy Greener	Removed DPSlogoff, DPSlogon renamed to DPSrequestToken to better reflect functionality, additional returned data wrapper elements. Minor text modifications and clarification of difference between insertion date and issue date
1.2	25/10/05	Andy Greener	Section 5 heavily modified as a result of decision not to use password digests
1.3	22/06/11	Andy Greener	Add RTI message
1.4	30/10/12	Graham Jenkins	Added Generic Notifications to RTI
1.5	06/05/14	Graham Jenkins	Revised section 2.7 message size note
1.6	25/09/2014	Fraz Ahmad	Revised Section 6.3 SL1 Data Item Schema – To include the new PlanType this is an optional element either 01, 02 or “”
1.7	04/12/2014	Dave Joss	Updated Schema diagrams in Sections 6.1 and 6.2
1.8	11/01/2015	Troy Newton	Added Generic Notifications for CIS
1.9	02/03/2018	Troy Newton	Section 2.1 updated to refer to TLS; section 3.3 references to Nitems corrected to DPSqueryResult; section 3.5 DPSqueryResult entry added for clarity
1.10	25/09/2018	Troy Newton	Addition of Post Graduate Loan PGL1 and PGL2 Data Items
2.0	04/10/2018	Troy Newton	Document updated following review comments from the SDS Team.
3.0	20/10/2018	Troy Newton	Baselined

Contents

1. Introduction

This document defines the client view of the Application Programming Interface (API) for HMRC's Data Provisioning Service (DPS) in abstract terms – that is, in terms of the data that passes between client applications and the service itself, and the modes of interaction. The exact programming "look and feel" will depend upon the SOAP toolkit (if any) employed in the construction of the client application – most will be able to present a remote method invocation or remote procedure call interface given an appropriate Web Services Description Language (WSDL) file describing the service interface. However, there is sufficient detail in this document (and the WSDL file) to allow the low-level construction and handling of SOAP messages if necessary. The selected interaction style is RPC ("remote procedure call") with no (i.e. "literal") encoding. This style implies a synchronous, client-server model of interaction.

2. Major Aspects of the Interface

2.1 Transport and Security

The service will be accessible only via SOAP/HTTP over a TLS (formerly SSL) connection ("secure HTTP" or HTTPS) to ensure the confidentiality of requests and returned data. The DPS will be implemented using SOAP v1.1, which will support v1.1 SOAP clients.

Client authentication will be based on Govt Gateway credentials (user id and password) and will conform to the WS-I WS-Security standard, requiring prior registration on the Gateway or HMRC Portal and enrolment for a service supported by the DPS (currently only PAYE and CIS).

2.2 Stateless Operation

The DPS provides a stateless API for third party applications to retrieve data items made available by the service. In essence this means that each call to the interface is entirely independent (ignoring authentication) and contains enough information to define exactly the type and extent of the data to be retrieved – no "state" or temporary information concerning a sequence of retrieval calls need be held server-side, improving the resilience of the service in the face of unexpected client or server failure.

2.3 Session-Oriented Operation

Consideration was given to the ability of the three core methods of the API – DPSretrieve, DPSquery and DPSdate2index - to be used in "sessionless" mode by providing Govt Gateway credentials (user id and password) on each call. The overhead inherent in this mode of operation is that each call must be authenticated with the Govt Gateway, which requires a synchronous call from the DPS to the Gateway Authentication and Authorisation (A&A) API. For occasional or light use of the DPS API this might be acceptable from a client application point of view, but it is considered as an unacceptable risk to the overall load on the service at the DPS-Gateway interface. Sessionless operation may be introduced at a later date.

Consequently, the DPS will operate in "session-oriented" mode – an unauthenticated WS-Security UsernameToken is converted into a signed SAML security "token" at the cost of a single authentication call to the Gateway A&A API. From then on (until the Token times out) the token is presented, unaltered, as the credential for each call. So long as the token remains valid this will avoid any further calls to the Gateway A&A API during that "session". The "state" held by the DPS relates to the identity and authorisation of the caller only (the API itself remains stateless with respect to the data being retrieved).

Sessions are supported by the method call DPSrequestToken, which must be the first method call in any interaction with the DPS API (the token returned by DPSrequestToken will time out after four hours).

2.4 Data Item Indexing

Each data item (e.g. each individual employee coding notice or user reminder notice) within a set of related items has an associated unique index number (a 32-bit unsigned integer). These index numbers are assigned sequentially when data items arrive in the DPS database from HMRC back-end systems and within a set of related items will never repeat (a 32-bit unsigned integer gives over 4 billion possible index values, which is sufficient to last many tens of years).

For PAYE there are nine sets of related data items and therefore nine independent indexes, all starting at 1 and monotonically increasing (index number 0 is deliberately not used). The nine PAYE sets are:

- P6 and P6b
- P9
- SL1
- SL2
- PGL1
- PGL2
- Annual Reminders (ARs)
- Notifications (P35, P11Db, Incentive Letter)
- RTI Notifications

For CIS there will be one set

- CIS (GEN)

The index numbers for a set of related data items for a particular user are not guaranteed to be contiguous but will be unique and chronologically ordered because of the way they are assigned on arrival in the DPS database (i.e. a higher index number indicates a concurrent or later data item than a lower index number).

The uniqueness (across all data items of a particular type, for all users) and chronological ordering of the index numbers are fundamental to the way the API works. Data items are retrieved based on their index numbers and "high water marks" – the "high water mark" is the only piece of "state" that needs to persist on the client side (aside from credentials and designatory information) during the normal operation of the DPS interface.

2.5 Normal Operation

Initially, before any call is made to the service, the client's high water mark for each index range is assumed to be zero (this is why 0 is not used as a data item index number). After a successful retrieval call for a number of data items of a particular type (made by indicating that the caller has "got" up to "index number" 0) the returned high water mark will indicate the highest index number of the data items returned. This value should be retained and used in subsequent retrieval calls as the "got" value – only data items with higher index numbers than this will be returned, and again the high water mark will be updated (this cycle is the crux of the normal client interaction procedure over time).

This mechanism works whether there is one user's data or a million users' data residing in the DPS – in the latter case index numbers are not likely to be contiguous, except where a batch of data items for a single user is added to the DPS database in an atomic operation, but the chronological ordering guarantees that items with an index value higher than a particular high water mark, arrived after the high water mark item and are therefore more recent additions to the database available for retrieval.

2.6 Unusual Operations

The utility of the index and high water mark mechanism permits unusual operations to be performed where necessary. Where a client application needs to repopulate a database from scratch (after a disaster for instance), all the data items of a particular type for an employer, contractor or Agent's clients can be retrieved by adjusting the "got" value to zero, even after old data is archived from the service after several years this will still retrieve all the remaining un-archived data in the service database. Similarly, when a client moves from one Agent to another, or joins an Agent for the first time, the new Agent immediately has access to all the data held for that client, assuming the client elected to receive electronic output previously, and can retrieve everything held on the DPS relating to that client (the old Agent immediately loses access to all the data for that client of course, except for any already downloaded – this is a consequence of run-time checking of Agent authorisations).

For more sophisticated retrievals, based on business events or key dates, an API method (DPSdate2index) is provided to convert a date (meaningful to the client or Agent) into a high water mark prior to that date, so that using that high water mark as a "got" value, a retrieval will download any relevant data items that appeared (i.e. were inserted into the DPS database, which is not necessarily the same as the date of issue attached to a data item by the issuing system) in the service on that day or subsequently, since they will have higher index numbers. This makes it natural to use a date on which something significant happened (e.g. the start of a tax year as 6th April) to obtain an appropriate high water mark for a subsequent retrieval inclusive of that date (in practice this might result in the retrieval of some items that were inserted on that date but which were issued a day or two earlier – if necessary these can be filtered out by examination of item issue dates or item index numbers – see below).

Where data is re-retrieved, the data item index number (which is returned as an attribute of the data item) can be used to accurately identify duplicated data items and discard them (such as when overlapping ranges of items are downloaded when client application errors occur, or date-based retrieval returns items issued just prior to the period of interest, or client move Agents and regular and one-off retrievals are not synchronised properly). This should help avoid the accidental re-use of potentially obsolete data, assuming client applications record index numbers for comparison.

2.7 Managing Large Retrievals

For large users, Agents, and in the case of unusual operations for small employers such as re-synchronising an application database, it is possible that the amount of data to retrieve will exceed the in-built message-size limits of the service (each data type will have a maximum number of data items defined for a single retrieval designed to keep message size below a pre-defined limit – see table below). In these cases, a retrieval will result in the return of only some of the data items waiting (always the lowest index numbers). Where this occurs, a "more data" flag is set by the service and the returned high water mark is set to the index number of the highest index value actually returned. The "more data" flag should be tested in the normal course of events after every DPSretrieve call in case this situation occurs. If it does, one or more further calls are made, each time using the returned high water mark and updating the "got" value on each call. In this manner, large downloads can be accommodated without prior arrangement. If a retrieval times-out, it is only necessary to re-start that DPSretrieve call with the same parameters.

However, there may be situations where it is known in advance that a large number of data items will likely be waiting (e.g. a large user or Agent). In this case, a client application can enquire to obtain the actual number of items waiting by using the DPSquery method call. Based on this information, the client application can limit retrievals to a set number of items, calculated as a proportion of the total. By repeating retrievals an appropriate number of times, all the waiting data can be retrieved in chunks that are manageable. One significant advantage of this method is that the client application can attempt to predict the duration of a download and report progress in real time (by updating a status bar or 'percent complete' indicator).

Data Item Type	Maximum Data Items per Retrieve
P6	1,500
P9	1,500
SL1	2,000
SL2	2,000
PGL1	2,000
PGL2	2,000
Annual Reminders	600
Notifications	500
RTI Notifications	500
CIS Notifications	600

Sizing Notes:

These maxima were chosen to restrict the returned XML to approximately 1MB and should be used as a sizing guide.

The Maximum Data Items per Retrieve may be further adjusted to maintain the 1MB approximate XML message size restriction. Any adjustment to these maxima will be re-published in this API Specification.

3. API Calls

3.1 Session Authentication

DPSrequestToken is the method used to acquire a security token for the sole purpose of avoiding the credential authentication overhead on a number of subsequent calls to the API. The token returned as a string of XML is passed to all subsequent calls to identify the requestor to the service. DPSrequestToken should be the first method call made to the API during a "session".

Method: DPSrequestToken				
Parameters - all mandatory				
Call	Name	Type	Example value	Comments
	Credentials	String	<wsse:UsernameToken> ……. </wsse:UsernameToken>	A WS-Security Username Token with password in clear text
	Version	Integer	1	Always 1 for now
	VendorID	String	"0132"	SDST-allocated Vendor ID
Return				
	Token	String	<!--Token XML -->	A serialised WS-Security token

Reported error conditions resulting in failure of method call (custom errors raised as SOAP faults):

- Missing or badly formed Credentials
- Unable to authenticate credentials
- Missing or unknown version (1 is the only version currently supported)
- Missing or badly formed VendorID
- Data Provisioning Service unavailable
- Data Provisioning Service unable to respond

In all other circumstances (aside from internal server or SOAP errors) a valid WS-Security token is returned. This may be passed to subsequent API calls unaltered to identify the caller. The exact content of this token and its XML syntactic representation are irrelevant to the client application and is only meaningful to the Data Provisioning Service – for this reason it should be treated by the client application as a fixed and opaque sequence of characters.

See Section 5 below for a description of the construction of a UsernameToken.

3.2 Data Retrieval

DPSretrieve is the method called to retrieve data from the DPS. It must be called using a security token returned by DPSrequestToken. The call parameters contain all the information necessary to identify a specific and homogenous set of data items of a particular type and relating to a particular entity or set of entities, and in conjunction with the returned data and associated indicators can be used to manage large retrievals with repeated calls.

Method: DPSretrieve				
Parameters - all mandatory				
Call	Name	Type	Example value	Comments
	Credentials	String	<!--Token XML -->	A serialised WS-Security token returned from a previous DPSrequestToken call
	Version	Integer	1	Always 1 for now
	VendorID	String	"0132"	SDST-allocated Vendor ID
	Service	String	"PAYE", "CIS"	Supported services
	EntityType	String	"EmpRef" or "UTR"	Always "EmpRef" for PAYE and CIS
	Entity	String	"123/A45678"	An emp ref value or nil for all Agent's emp refs
	DataType	String	"P6", "P9", etc	The DataType 'CIS' is specific to the service 'CIS', and the other nine DataTypes relate to service 'PAYE'.
	Got	Integer	21422437	Highest item index already retrieved
	Nitems	Integer	1000	Max number of items to retrieve, nil = unlimited
Return				
	HighWaterMark	Integer	342149	Highest item index retrieved in this call
	MoreData	Boolean	true or false	True if more data to retrieve, false if not
	Data	String	<!-- data serialised as XML -->	Serialised XML structure representing retrieved data

Parameter enumerations supported by the service at this release:

Version	1
Service	"PAYE", "CIS"
EntityType	"EmpRef"
DataType	"P6", "P9", "SL1", "SL2", "PGL1", "PGL2", "AR", "NOT", "RTI", "CIS"

Note: The "P6" data type includes P6 and P6b; the "AR" data type includes AR6, AR1n, AR2n, AR1mn and AR2mn; the "NOT" data type includes P35Notification, P11DbNotification and Incentive Notification; the "RTI" data type includes RTI Notifications (NOT and NVR) and Generic Notifications (GEN); the "CIS" data type includes Generic Notifications (GEN).

Reported error conditions resulting in failure of method call (custom errors raised as SOAP faults):

- Missing or badly formed Credentials
- Unable to authenticate credentials
- Missing or unknown version (1 is the only version currently supported)
- Missing or badly formed VendorID
- Data Provisioning Service unavailable

- Data Provisioning Service unable to respond
- Missing or unknown Service name
- Missing or unknown EntityType
- Unrecognised or unauthorised Entity value
- Missing or unrecognised DataType value
- Negative or excessive (>4294967295, i.e. 2^{32}-1) Got value
- Negative or excessive (>4294967295, i.e. 2^{32}-1) Nitems value

If a retrieve legitimately results in no data items then an “empty” Data wrapper is returned (XML containing only header elements). This circumstance is not an error condition.

In the case of an EntityType defined as “EmpRef”, the format of the string used in the Entity parameter will be a three digit office number, a forward slash (“/”) and up to ten alphanumeric characters. This format exactly matches the permitted format for EmployerRef element content returned in data items associated with the PAYE and CIS services.

3.3 Data Query

DPSquery is the method called to obtain the number of data items waiting to be retrieved, based on the values of the Service, EntityType, Entity, DataType and Got parameters. It must be called using a security token returned by DPSrequestToken. The call parameters contain all the information necessary to identify a specific and homogenous set of data items of a particular type and relating to a particular entity or set of entities. The return value is intended to be used to compute the size, if necessary, or likely time to complete, of subsequent DPSretrieve method calls.

Method: DPSquery				
Parameters - all mandatory				
Call	Name	Type	Example value	Comments
	Credentials	String	<!--Token XML -->	A serialised WS-Security token returned from a previous DPSrequestToken call
	Version	Integer	1	Always 1 for now
	VendorID	String	“0132”	SDST-allocated Vendor ID
	Service	String	“PAYE”, “CIS”	Supported services.
	EntityType	String	“EmpRef” or “UTR”	Always “EmpRef” for PAYE and CIS
	Entity	String	“123/A45678”	An emp ref value or nil for all Agent’s emp refs
	DataType	String	“P6”, “P9”, etc	The DataType 'CIS' is specific to the service 'CIS', and the other nine DataTypes relate to service 'PAYE'.
	Got	Integer	21422437	Highest item index already retrieved
Return				
	DPSqueryResult	Integer	123	Number of data items available to be retrieved

Parameter enumerations supported by the service at this release:

Version	1
Service	“PAYE”, “CIS”
EntityType	“EmpRef”
DataType	“P6”, “P9”, “SL1”, “SL2”, “PGL1”, “PGL2”, “AR”, “NOT”, “RTI”, “CIS”

Note: The “P6” data type includes P6 and P6b; the “AR” data type includes AR6, AR1n, AR2n, AR1mn and AR2mn; the “NOT” data type includes P35Notification, P11DbNotification and Incentive Notification; the “RTI” data type includes RTI Notifications (NOT and NVR) and Generic Notifications (GEN); the “CIS” data type includes Generic Notifications (GEN).

Reported error conditions resulting in failure of method call (custom errors raised as SOAP faults):

- Missing or badly formed Credentials
- Unable to authenticate credentials
- Missing or unknown version (1 is the only version currently supported)
- Missing or badly formed VendorID
- Data Provisioning Service unavailable
- Data Provisioning Service unable to respond
- Missing or unknown Service name
- Missing or unknown EntityType
- Unrecognised or unauthorised Entity value
- Missing or unrecognised DataType value
- Negative or excessive (>4294967295, i.e. 2^{32}-1) Got value

A DPSqueryResult value of zero (i.e. no data fitting the description is available) is a legitimate response, not an error condition.

3.4 Date Conversion

DPSdate2index is the method used to convert a calendar date into a high water mark value for subsequent use in DPSretrieve method calls. It must be called using a security token returned by DPSrequestToken. The call parameters contain all the information necessary to identify a specific and homogenous set of data items of a particular type and relating to a particular entity or set of entities. The returned value is a data item index for a data item with an insertion date & time immediately prior to the given date (note: insertion date is not necessarily the same as issue date, and may be a day or two later). The item index is only meaningful for use in subsequent DPSretrieve calls that relate to the same Service, Entity and DataType.

Method: DPSdate2index				
Parameters - all mandatory				
Call	Name	Type	Example value	Comments
	Credentials	String	<!--Token XML -->	A serialised WS-Security token returned from a

				previous DPSrequestToken call
	`Version`	`Integer`	`1`	Always 1 for now
	`VendorID`	`String`	`“0132”`	SDST-allocated Vendor ID
	`Service`	`String`	`“PAYE”, “CIS”`	Supported services.
	`EntityType`	`String`	`“EmpRef” or “UTR”`	Always “EmpRef” for PAYE and CIS
	`Entity`	`String`	`“123/A45678”`	An emp ref value or nil for all Agent’s emp refs
	`DataType`	`String`	`“P6”, “P9”, etc`	The DataType 'CIS' is specific to the service 'CIS', and the other nine DataTypes relate to service 'PAYE'.
	`Date`	`Date`	`2006-04-06`	Calendar date for conversion to an index
`Return`				
	`HighWaterMark`	`Integer`	`342149`	Highest inserted item index prior to given date

Parameter enumerations supported by the service at this release:

Version	1
Service	“PAYE”, “CIS”
EntityType	“EmpRef”
DataType	“P6”, “P9”, “SL1”, “SL2”, “PGL1”, “PGL2”, “AR”, “NOT”, “RTI”, “CIS”

Note: The “P6” data type includes P6 and P6b; the “AR” data type includes AR6, AR1n, AR2n, AR1mn and AR2mn; the “NOT” data type includes P35Notification, P11DbNotification and Incentive Notification; the “RTI” data type includes RTI Notifications (NOT and NVR) and Generic Notifications (GEN); the “CIS” data type includes Generic Notifications (GEN).

Reported error conditions resulting in failure of method call (custom errors raised as SOAP faults):

- Missing or badly formed Credentials
- Unable to authenticate credentials
- Missing or unknown version (1 is the only version currently supported)
- Missing or badly formed VendorID
- Data Provisioning Service unavailable
- Data Provisioning Service unable to respond
- Missing or unknown Service name
- Missing or unknown EntityType
- Unrecognised or unauthorised Entity value
- Missing or unrecognised DataType value
- Missing or badly-formed Date
- Date prior to 2005-08-01 (to catch obviously erroneous dates)
- Date in future

3.5 Parameter Data Type Definitions

The following table summarises the type definitions (in terms of W3C XML Schema data types) for the parameters of the above API calls.

Parameter	Type	Min	Max	Pattern or Enumeration	Nil Allowed
Credentials/Token	xsd:string			*Well-formed XML*	No
Version	xsd:positiveInteger	1	99		No
VendorID	xsd:string	len 2	len 4	[0-9]{2,4}	No
Service	xsd:string	len 1	len 20	[A-Za-z0-9\-_]*	No
EntityType	xsd:string	len 1	len 10	[A-Za-z0-9\-_]*	No
Entity	xsd:string	len 1	len 20	[A-Za-z0-9\-_/#\(\)\[\]\+\.:;]*	Yes[1]
DataType	xsd:string	len 1	len 10	[A-Za-z0-9\-_]*	No
Got	xsd:positiveInteger	0	2^{32}-1		No
Nitems	xsd:positiveInteger	0	2^{32}-1		Yes[2]
DPSqueryResult	xsd:positiveInteger	0	2^{32}-1		Yes
HighWaterMark	xsd:positiveInteger	0	2^{32}-1		N/A
MoreData	xsd:boolean			true, false	N/A
Data	xsd:string			*Well-formed XML*	N/A
Date	xsd:date	2005-08-01	Today		No

[1] A nil *Entity* signifies all entities relating to the authenticated user.

[2] A nil *Nitems* indicates no client-imposed limit on the number of data items retrieved (per-data item limits apply- see section 2.7).

4. Error Handling

DPS-specific errors identified for each method above are raised by the DPS SOAP service as custom errors, and communicated to the client application using the SOAP <Fault> element in the SOAP envelope. Access to SOAP faults will depend on the error handling provisions of the SOAP client library in use.

After making a DPS method call, the client application is expected to check for errors – custom and other (e.g. SOAP, object instantiation, etc.). If no errors of any kind are raised, the response is expected to contain the normal output from the call, and processing can continue. If errors were raised, further checks for custom errors can be made to determine the cause and the appropriate remedial action to be undertaken.

Where a DPSretrieve legitimately results in no data items being identified for retrieval, an “empty” Data XML wrapper is returned. This is not an error condition.

5. Credentials

Credentials can take one of two forms (both XML-based, and both conforming to the WS-Security standard): a username token (provided to DPSrequestToken) or a SAML security token (returned from DPSrequestToken and provided in all other API method calls). A username token is recognised by its root element name (including the “standard” namespace prefix): <wsse:UsernameToken>. The username token is defined in the OASIS Web Services Security UsernameToken Profile 1.0:

http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf

The WS-Security specification supports two different methods of passing the password in the UsernameToken, either ‘PasswordText’ or ‘PasswordDigest’. The ‘PasswordText’ attribute allows the password to be included in the security token in clear text. This is not supported directly by the Government Gateway but it will be the only method supported by the DPS API since all transactions are encrypted for transit as a result of using HTTPS.

In its simplest form the UsernameToken looks like this:

```
<wsse:UsernameToken>
    <wsse:Username>[Gateway-user-id]</wsse:Username>
    <wsse:Password Type="…#PasswordText">[Password]</wsse:Password>
</wsse:UsernameToken>
```

The highlighted values would be replaced by the actual user-id and the clear-text password. Note that the value of the Password element Type attribute has been shortened for clarity in the example – in full it should be:

http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText

though since this is the default value for the Type attribute it may be omitted.

The namespace name for the conventional prefix ‘wsse’ is

http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd

The UsernameToken should be placed inside a wsse:Security element within the header of a SOAP message. Recent SOAP toolkits that support WS-Security may provide explicit support for the addition of security tokens. In all other cases, an explicit insert of a wsse:Security section containing the UsernameToken into the SOAP header will be required by whatever means is available.

6. DataType Data Item Schemas

The DataTypes fall into three categories: employee data (P6, P9, SL1, SL2, PGL1, PGL2, RTI/NOT, RTI/NVR), employer-only data (Reminders and Notifications) and Generic Data that can consist of employee or employer data (RTI/GEN) or contractor data (CIS/GEN) – see table below. When DPSretrieve returns a stream of data items as XML, the employee data items contain (aside from the necessary employee data) an <EmployerRef>. For a single user retrieval all the <EmployerRef> values will be identical – for an Agent retrieval each <EmployerRef> value may be different. No other User, Agent or HMRC Office details are included in the data stream, either in individual data items or in the message “wrapper”.

Category	DataTypes	Comments
Employee Data	P6, P9, SL1, SL2, PGL1, PGL2, RTI/NOT, RTI/NVR	Data item contains Employer Reference only
Employer-only Data	AR, NOT	Data item contains Employer, Agent (if appropriate) and HMRC Office details
Generic Data (Employee/Employer data)	RTI/GEN	The “RTI” GEN Form Type contains Employer details and is optionally extensible to provide additional content details.
Generic Data (Contractor data)	CIS\GEN	The “CIS” GEN Form Type contains appeals information.

When DPSretrieve returns a stream of employer-only category data items, each data item is complete with User, Agent (if appropriate) and HMRC Office details (if required) as necessary.

All data items, of whichever category, carry a sequence number, an issue date, and where appropriate an effective date.

Whilst the DPS API is very different from the SMS API it replaces, the data item content they both handle is very similar. As a result, and where possible, Schema structures in the old SMS outgoing service message Schemas (not RTI) have been retained to minimise changes to existing rendering stylesheets and code already in place to handle data. In particular, attributes have been retained even where it might make more sense to use elements (for instance the Issue Date is retained as an attribute on the root element across the board). The one exception to this rule applies to the employee-category data items, where instead of multi-employee “form” structures, each employee notice is a separate data item.

In all cases, the root element of the data item Schemas described below carry attributes for the Issue Date and Sequence Number. Item-specific attributes are described below.

Please note that all schemas are published as part of the Data Provisioning Service technical specification pack

Appendix A. Usage Scenarios

A.1 Small to medium user, regular cycle

Within the context of a “session” created by a call to DPSrequestToken, which returns a security token that will expire after 4 hours, the one and only regular interaction with the API will be a single DPSretrieve for each data type of interest to the user, using the security token returned by DPSrequestToken and the user’s one and only Employer Reference, on a regular cycle to suit the user and the level of traffic expected (perhaps once a week for P6s, and larger intervals for other data types, initiated by user intervention with only occasional need to make an Internet connection).

The information saved between calls amounts to a single data value per data type – the current “high water mark” returned from the last retrieval. This value is used in a subsequent call and then overwritten by the newly returned “high water mark”.

The data items retrieved in each call invariably represent all the data waiting at that point in time – if, in extreme circumstances, the volume of data exceeds the message size limit for the service (see section 2.7) further retrieval calls may be made to collect any outstanding data.

A.2 Large user regular cycle

A large user may well require multiple retrievals and/or queries for each data type (at least for some of the high traffic employee-oriented data types). Prior to a series of retrievals or queries, a “session” is initiated using the user’s Gateway credentials in a call to DPSrequestToken – subsequent API calls use the returned security token.

Repeated DPSretrieve calls with the user’s Employer Reference are made for each data type in turn, checking the ‘more data’ flag returned in each call until it is false. The “high water mark” returned by the final DPSretrieve for each data type is stored for subsequent use (intermediate “high water marks” are used to advance the retrieval process as it proceeds). A high water mark value for each data type must be persisted between retrieval sessions.

Large users may have multiple PAYE/CIS Schemes and therefore multiple Employer References. However, PAYE/CIS service enrolment is per-Scheme, so separate Gateway credentials are required for each Scheme. Consequently, a separate session must be established with the DPS for each Employer Reference, unless a user has arranged for “agent” access, in which case scenario A.3 below applies.

Frequency of access is likely to be of the order of days rather than weeks, and may be under program control rather than initiated by a user on demand.

A.3 Agent, regular cycle

An Agent is similar to a large user in terms of the volume of data (more so in the case of notices and reminders where there will be one for each client) and will behave similarly as a result. Whereas an individual user makes API calls with a single Employer Reference or repeatedly initiates sessions with each of their Employer References, an Agent can elect to retrieve data or query the service based on a single Employer Reference or all the Employer References of all their clients. In the latter case, data items can be distinguished by the value of the <EmployerRef> element they carry. Agent software applications written to make multi-Employer Reference retrievals will have to be able to distinguish data for each client in the returned data stream. However, only one high water mark per data type need be persisted between sessions, unless a mixture of single and multiple Employer Reference retrievals is required, in which case it is necessary to maintain a high water mark per Employer Reference and synchronise these when making multi-Employer Reference retrievals.

Frequency of access is likely to be high (perhaps once a day), especially for large Agents.

A.4 Pre-arranged large download

In the case of a large user or Agent, or a retrieval of an entire user's history (see A.5 below), a client application may wish to structure the entire download into a series of retrievals of known size. The number of items of a particular data type waiting to be retrieved can be obtained using DPSquery. Using this number, and the known maximum-items-per-message as an upper limit (see section 2.7), a number of DPSretrieve calls may be made with a fixed number of items to retrieve. As each retrieval completes the user can be informed of progress, in the form of simple messages (e.g. “Got 1 of 5”, “Got 2 of 5”, etc) or graphically (e.g. as a dynamically updating progress bar). In addition, a prediction of time taken can be made based on the number of items to retrieve and then updated as it progresses.

A.5 Recover/reconstruct download history

The DPS persists all provisioned data until it is archived (see non-functional requirements for archival and retention periods, ref. [B.1]), so it is always possible to re-retrieve data so long as it is still resident in the DPS's database. The most common requirement is likely to be the retrieval of data from some known and significant date, such as the start of the present tax year.

It is possible for client applications to record high water marks at significant historic points, in which case a retrieve can be initiated using the appropriate value (whether a single retrieval, an ad hoc multiple retrieval or a planned multi-part retrieval). However, the API provides a method call specifically to convert a known calendar date into a high water mark value – this is then used as the “Got” value in a subsequent DPSretrieve (or as the starting point for a series of DPSretrieve calls). The date relates to the insertion of data items into the DPS database, and is not necessarily the same as the issue date assigned to data items by the issuing system (the issue date might be a day or two earlier, depending upon the frequency of DPS updates from the relevant back-end system).

In the unlikely event that all the data of a specific data type and relating to one or a set of Employer References must be retrieved, the "Got" value of the appropriate DPSretrieve can simply be set to zero.

A.6 Client transfers from Agent to Agent

The authentication mechanism provided by the DPS ensures that only a Client or their authorised Agent can retrieve data relating to their Employer Reference. When a Client leaves an Agent and their relationship is no longer valid, the Agent will automatically lose access to that Client's data, past and future. Correspondingly, an Agent who is newly authorised by the Client immediately gains access to that Client's data, past and future. This means that regular multi-client retrievals will begin to include the Client's data, as it is issued.

Specific action may be required to retrieve past data (from the start of the tax year for instance) if the transfer happens in-year (see A.5 above). In this case the index numbers of individual data items can be used to recognise and ignore duplicate data items and infer order of issue.

Appendix B. Non-Functional Requirements

B.1 DataType Data Item Retention

The HMRC Retention Policy defines, for each Data Item Type, how long HMRC will store the data for the DPS. Whilst HMRC retain the data for the DPS, it may be returned in response to valid DPSretrieve requests.

Data Item Type	HMRC Retention Policy
P6	LEQ[1] current tax year – 3
P9	LEQ current tax year – 3
SL1	LEQ current tax year – 3
SL2	LEQ current tax year – 3
PGL1	LEQ current tax year – 3
PGL2	LEQ current tax year – 3
Annual Reminders	LEQ current tax year – 3
Notifications	LEQ current tax year – 3
RTI Notifications (NOT & NVR)	LEQ 30 days (from date of issue)
CIS Notifications (GEN)	LEQ 30 days (from date of issue)

The retention policy for RTI and CIS Notifications of Form Type “GEN” allows different retention periods to be defined for different Notification Types identified by <MessageTypeID>. For further details please refer to the DPS User Guide.

[1] LEQ: “Less than or Equal to”.