



Inland Revenue

## Build pack: Software Intermediation service

**Date:** 2/02/2018  
**Version:** v0.80



## Contents

<b>1 Overview</b>	<b>4</b>
1.1 This solution	4
1.2 Intended audience	4
1.3 Onboarding packs for supported processes	4
1.3.1 TDS Overview and Transition Build Pack—onboarding section	4
1.4 Related build packs	5
1.4.1 Transaction Data Services Overview and Transition build pack	5
1.4.2 Identity and access services build pack	5
1.5 Prerequisites	5
<b>2 Solution design</b>	<b>6</b>
2.1 Architecture	6
2.2 Service scope	6
2.3 Messaging	6
2.4 Security	8
<b>3 Operations</b>	<b>11</b>
3.1 Link	12
3.2 Delink	13
3.3 RetrieveClientList	15
<b>4 End points, schemas and WSDLs</b>	<b>18</b>
4.1 End points	18
4.2 Schemas	19
4.3 WSDLs	19
<b>5 Responses</b>	<b>20</b>
5.1 Generic gateway response codes	20
5.2 Generic software intermediation response codes	21
<b>6 Use cases and scenarios</b>	<b>22</b>
6.1 Scenarios	22
6.2 Use cases	24
<b>7 Appendix A—Glossary</b>	<b>26</b>
<b>8 Appendix B—Change log</b>	<b>29</b>



## List of figures

Figure 1 : Flow of data from user to Inland Revenue .....	6
Figure 2: SOAP request .....	7
Figure 3: Soap response.....	8
Figure 4: Schema aliases.....	11
Figure 5: General response structure .....	12
Figure 6: Link operation structure.....	13
Figure 7: Delink operation structure.....	14
Figure 8: RetrieveClientList operation structure.....	16
Figure 9: Retrieve List Response structure.....	17
Figure 10: WSDL naming conventions .....	19

## List of tables

Table 1: Prerequisites .....	5
Table 2: Ciphers and key strengths.....	8
Table 3: Endpoints.....	10
Table 4: Data fields .....	11
Table 5: Link operation data .....	13
Table 6: Delink operation data .....	15
Table 7: RetrieveClientList operation data.....	16
Table 8: Sliced data end points .....	18
Table 9: Unsliced data end points .....	18
Table 10: Production end points .....	18
Table 11: Generic Gateway Service response codes .....	21
Table 12: Software intermediation response codes .....	21
Table 13: Software Intermediation scenarios .....	24
Table 14: Use case link/delink software provider .....	25



## 1 Overview

### 1.1 This solution

Inland Revenue has a suite of digital services available for consumption by software providers that supports efficient, electronic business interactions with Inland Revenue. The Software Intermediation service described in this build pack document forms part of a suite of Gateway Services.

The Software Intermediation service provides the ability for software providers to link to their Tax Agents or directly to their Customers in order to be provided an initial full data load and daily bulk file updates of their transaction data. The Software Intermediation service allows these links to be established, queried, or removed. There is a separate Intermediation service that deals with links between business intermediaries like tax agents and their client lists and their customer accounts. The actual dataset resulting in bulk files from a software intermediation link will be impacted by those intermediation links but the links themselves are maintained separately. This is explained in more detail in the [TDS Overview build pack](#).

This is a stand-alone document intended to provide the technical details required to support the consumption of this Gateway Service. It describes the architecture of the technical solution, schemas, endpoints and also its interaction with other build packs that cover different aspects of Gateway Services. The associated onboarding and overview documents describe the end-to-end business level solution, of which this build pack forms part. There is a zip file with development versions of schemas, and another one with sample requests and responses that are released with this document.

### 1.2 Intended audience

The solution outlined in this document is intended to be used by technical teams and development staff. It describes the technical interactions, including responses, provided by the Software Intermediation service.

The reader is assumed to have a suitable level of technical knowledge in order to comprehend the information provided. A range of technical terms and abbreviations are used throughout this document, and while most of these will be understood by the intended readers, a [glossary](#) is provided at the end.

### 1.3 Onboarding packs for supported processes

Before using this build pack, ensure the relevant onboarding pack or overview pack has been consulted to provide business-level context. The Inland Revenue onboarding packs listed below are supported by this build pack, all of which are available on Inland Revenue's Gateway Services GitHub site:

<https://github.com/InlandRevenue/Gateway-Services>

#### 1.3.1 TDS Overview and Transition Build Pack—onboarding section

The Onboarding section of the TDS Overview Build Pack provides a guide for how consumers can onboard the various TDS components. It gives details of prerequisites, setup requirements, testing, contact lists and more. It is intended to help an organisation start using the TDS solution as quickly and easily as possible.



## 1.4 Related build packs

The following Gateway Services build packs complement this one.

### 1.4.1 Transaction Data Services Overview and Transition build pack

The [Transaction Data Services Overview and Transition build pack](#) was created to support software providers in their transition from Tax Agent Web Services to the use of TDS. It provides an overview of TDS, describes the data which will be made available through the services and the processes, as well as giving use cases for how these services will be employed.

### 1.4.2 Identity and access services build pack

The [Identity and access \(IAS\) services build pack](#) describes the operations provided under Identity and Access services, which is another part of the Gateway Services suite. These services are used to authenticate access.

This Software Intermediation service build pack was written using information from the Identity and Access services build pack.

## 1.5 Prerequisites

Party	Requirement	Description
<b>Inland Revenue</b>	Provide the Inland Revenue public certificate for mutual TLS	Inland Revenue's public X.509 certificate to support TLS will be provided as part of connectivity testing.
<b>Software pProvider</b>	Acquire a X.509 certificate from a certificate authority for the Test and Production environments	This is required when using mutual TLS with cloud-based software providers.

**Table 1: Prerequisites**

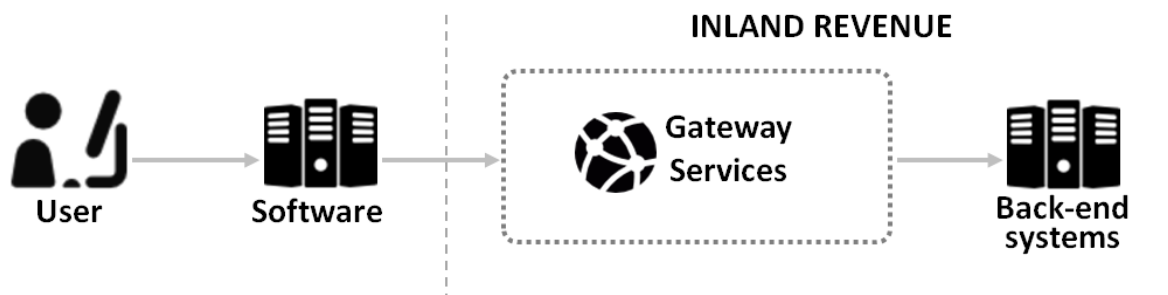


## 2 Solution design

### 2.1 Architecture

Inland Revenue is offering a suite of web services in order to facilitate interactions with Inland Revenue via software packages. The Gateway Services suite will be used by approved software vendors to facilitate everything from registration activities, filing returns, making payments and other service offerings in order to allow customers to interact with Inland Revenue.

The diagram below illustrates the flow of data from the customer to Inland Revenue.



**Figure 1 : Flow of data from user to Inland Revenue**

The online WSDLs for the Gateway Services define an 'any' XML request and response structure, which then relies on a group of XSDs to define the data structure of those requests and responses. Each request and response type will define a lower, 'wrapper' element. To simplify analysis and code generation, a development oriented version of the WSDL and XSDs is provided with the build pack that has the any elements replaced with relevant types.

Any malformed XML will instantly be rejected by the Gateway Services prior to any schema validation.

### 2.2 Service scope

The Software Intermediation service supports the following operations:

- **Link:** This service is used to create a link between a software intermediary and a client.
- **Delink:** This service is used to cease a link between the above parties.
- **RetrieveClientList:** This service is used to retrieve a list of the software intermediary's clients.

### 2.3 Messaging

All SOAP messages require a SOAP header containing the **Action:** parameter, as well as a SOAP body containing a structured XML payload. Please refer to the WSDL for the correct addresses.

The online WSDLs for the Gateway Services define an 'any' XML request and response structure, which then relies on a group of XSDs to define the data structure of those requests and responses. Each request and response type will define a lower, 'wrapper' element. To simplify analysis and code generation, a development oriented version of the WSDL and XSDs is provided with the build pack that has the any elements replaced with relevant types.



The Gateway Services allow the consumption of any structured XML payload but will be validated against the Inland Revenue-published XSDs.

This is a late binding validation, performed after authentication has been reviewed. The message structure of these services is a simple request/response. The XML request will be checked for well-formed XML before the schema validation. Responses to these requests will be in XML format as well and will be defined in the same schemas that define the requests.

Any XML submissions in the SOAP body that do not meet the provided schemas will not be accepted by the Gateway Services. Incorrect namespaces will also fail validation against the published schemas.

### Example SOAP request structure

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:sft="https://services.ird.govt.nz/GWS/SoftwareIntermediation/"
  xmlns:gcl="https://services.ird.govt.nz/GWS/SoftwareIntermediation/:types/RetrieveClientListRequest"
  xmlns:a="http://www.w3.org/2005/08/addressing">
  <soap:Header>
    <a:Action>https://services.ird.govt.nz/GWS/SoftwareIntermediation/SoftwareIntermediation/Operation</a:Action>
  </soap:Header>
  <soap:Body>
    <sft:RetrieveClientList>
      <sft:RetrieveClientListRequestMsg>
        <rcl:RetrieveClientListRequestWrapper>
          <RetrieveClientListRequest xmlns:xsi...
            <...XML payload...>
          </RetrieveClientListRequest>
        </rcl:RetrieveClientListRequestWrapper>
      </sft:RetrieveClientListRequestMsg>
    </sft:RetrieveClientList>
  </soap:Body>
</soap:Envelope>
```

**Figure 2: SOAP request**

### Example SOAP response structure

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:a="http://www.w3.org/2005/08/addressing"
  xmlns:si="https://services.ird.govt.nz/GWS/SoftwareIntermediation/"
  xmlns:b="https://services.ird.govt.nz/GWS/SoftwareIntermediation/:types/RetrieveClientListResponse"
  xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cmn="urn:www.ird.govt.nz/GWS:types/Common.v1">
  <s:Header>
    <a:Action s:mustUnderstand="1">
      https://services.ird.govt.nz/GWS/SoftwareIntermediation/SoftwareIntermediation/RetrieveClientListResponse
    </a:Action>
  </s:Header>
  <s:Body>
    <si:RetrieveClientListResponse >
      <si:RetrieveClientListResult>
        <b:RetrieveClientListResponseWrapper>
          <cmn:RetrieveClientListResponse>
            <cmn:statusMessage>
              <cmn:statusCode>0</statusCode>
              <cmn:errorMessage/>
            </cmn:statusMessage>
          </cmn:RetrieveClientListResponse>
        </b:RetrieveClientListResponseWrapper>
      </si:RetrieveClientListResult>
    </si:RetrieveClientListResponse>
  </s:Body>
</s:Envelope>
```



```

    </cmn:RetrieveClientListResponse>
  </b:RetrieveClientListResponseWrapper>
</si:RetrieveClientListResult>
</si:RetrieveClientListResponse>
</s:Body>
</s:Envelope>

```

**Figure 3: Soap response**

## 2.4 Security

Gateway Services requests are access-controlled using an OAuth token that identifies the user making the request. Users will authenticate using their Inland Revenue myIR credentials. For instructions on how to acquire an OAuth token, review the Identity and access build pack. For TDS Real Time web service requests, an OAuth access token is required in the HTTP header. Authorisation for using the Gateway Services is defined in the permissions set in myIR.

Permissions will reflect those granted in myIR. For example, if a user does not have permission to file a return online, they will not be able to file a return via Gateway Services either. This applies to users who are granted access as staff inside an organisation or as staff in a tax agency.

The Gateway Services use an HTTPS transport layer, with HTTP1.1 transport protocol supported.

The Gateway Services also use the SOAP version 1.2 protocol.

The SOAP service contract is published using WSDL version 1.1.

Transport layer encryption is mandatory and Gateway Services generally use the TLS version 1.2 specification.

Inland Revenue requires the following ciphers and key strengths to be used:

<b>Encryption:</b>	Advanced Encryption Standard (AES)	FIPS 197	256-bit key
<b>Hashing:</b>	Secure Hash Algorithm (SHA-2)	FIPS 180-3	SHA-256

**Table 2: Ciphers and key strengths**

There will be two endpoints, which are summarised in the bullet points below (the table immediately afterwards provides more detail):

1. There is an endpoint to which software providers' centralised **cloud** locations can connect. This will involve mutual TLS certificates that need to be exchanged during the onboarding phase. On the cloud endpoint Inland Revenue has controls to shield software providers from issues caused by heavy usage from other providers.
2. For software providers connecting from **desktops**, there is a separate endpoint that does not use mutual TLS. For this service, certificates do not need to be exchanged during onboarding. On the desktop endpoint Inland Revenue has less ability to shield consumers of the service from heavy usage by other Inland Revenue partners.





	Endpoint for cloud-based connections	Endpoint for desktop connections
<b>Purpose</b>	<ul style="list-style-type: none"> <li>This is the default endpoint to connect software providers to the Gateway Services.</li> </ul>	<ul style="list-style-type: none"> <li>Additional endpoint provided to facilitate connecting from desktops which might be high volumes of sources addresses, transient DHCP addresses, not realistically associated with client side TLS certificates, not individually onboarded to setup certificate trust.</li> </ul>
<b>Client application type</b>	<ul style="list-style-type: none"> <li>Cloud applications.</li> </ul>	<ul style="list-style-type: none"> <li>Desktop/native applications.</li> <li>For connecting from multiple decentralised clients.</li> </ul>
<b>Constraints</b>	<ul style="list-style-type: none"> <li>Only for source locations with client side TLS certificates.</li> <li>On the cloud endpoint Inland Revenue has controls to shield software providers from issues caused by heavy usage from other providers.</li> </ul>	<ul style="list-style-type: none"> <li>Less scalable.</li> <li>Subject to tighter security controls.</li> <li>On the desktop endpoint Inland Revenue has less ability to shield consumers of the service from heavy usage by others.</li> <li>OAuth2 refresh tokens will not be offered to desktop clients.</li> </ul>
<b>Mutual TLS</b>	<ul style="list-style-type: none"> <li>Inland Revenue explicitly trusts the certificate the software provider associates with the TLS connection as client for Mutual TLS connections and uses it to identify the software provider in conjunction with the web service identification below.</li> </ul>	<ul style="list-style-type: none"> <li>Server side certificates only.</li> </ul>
<b>Minimum TLS version</b>	<ul style="list-style-type: none"> <li>1.2</li> </ul>	<ul style="list-style-type: none"> <li>1.0(+)</li> </ul>
<b>URL</b>	<ul style="list-style-type: none"> <li>Contains .../gateway/..</li> </ul>	<ul style="list-style-type: none"> <li>Contains .../gateway2/..</li> </ul>
<b>Port</b>	<ul style="list-style-type: none"> <li>4046</li> </ul>	<ul style="list-style-type: none"> <li>443 (Default https port)</li> </ul>
<b>Web service consumer identification</b>	<ul style="list-style-type: none"> <li>Each software provider is given a software platform ID during onboarding. This ID is of type Customer ID and is independent of the endpoint.</li> <li>To be identified in web service calls related to getting an OAuth token each cloud</li> </ul>	<ul style="list-style-type: none"> <li>Each software provider is given a software platform ID during onboarding. This ID is of type Customer ID and is not specific to the endpoint.</li> <li>Desktop clients will be given different identity and access system</li> </ul>



	Endpoint for cloud-based connections	Endpoint for desktop connections
	application will be given identity and access system client_id/client_secret credentials during onboarding to allow it to get OAuth tokens to call this endpoint. <ul style="list-style-type: none"> <li>The mutual TLS certificate is used to identify the service provider</li> </ul>	client_id/client_secret credentials to cloud application clients.
<b>Firewalling in production</b>	<ul style="list-style-type: none"> <li>No IP address restrictions.</li> <li>Access limited by certificate enrolment.</li> </ul>	<ul style="list-style-type: none"> <li>No IP address restrictions.</li> </ul>
<b>Firewalling in non-production environments</b>	<ul style="list-style-type: none"> <li>No IP address restrictions.</li> <li>Access limited by certificate enrolment.</li> </ul>	<ul style="list-style-type: none"> <li>Firewalled—IP whitelisting needed.</li> </ul>

**Table 3: Endpoints**

**Delegated permissions:** These services will allow a user to retrieve only the data of customers that their credential (as represented by the OAuth token) has access to. If an account or its data is targeted by the request parameters but the user does not have permission, an error will be returned. This access will depend on delegation permissions set up in myIR.

Gateway services like these typically have a 60 second timeout configured, although this may be adjusted after testing.



### 3 Operations

**IMPORTANT:** The schemas listed here are subject to change. For the authoritative definitions, please refer to the information provided on the Inland Revenue Gateway Services GitHub site: <https://github.com/InlandRevenue/Gateway-Services>

The structures of all Gateway Service operations are intended to produce the most efficient requests and responses. Any common structures and fields will be used across many schemas and tax types through an intentional inheritance method. The section below describes the structure of each operation and the scenarios in which certain fields will be used in XML requests and responses.

This section contains schema aliases:

- Cmn: Common.xsd
- Sft: SoftwareIntermediation.xsd

All requests and responses live in the SoftwareIntermediation.xsd.

All operations for the Software Intermediation service will contain two standard header fields: **softwareProviderData** and **identifier**. The identifier value type will contain either "IRD", "ACCIRD" or "CST". This parameter will be different for each operation, as defined below.

The **identifier** field is common across all gateway services but refers to different parties in different services. In all cases it is the party with delegated permissions to whom an OAuth token is provided. If the value cannot be resolved to a known context, or if it can but the provided OAuth token does not have the necessary delegated permissions then the error code 4 "unauthorised delegation" is returned. Please refer to individual operations for the nature of the identifier expected in this parameter in any given context.

For example:

```
<cmn:softwareProviderData>
  <cmn:softwareProvider>SoftwareProvider</cmn:softwareProvider>
  <cmn:softwarePlatform>SoftwarePlatform</cmn:softwarePlatform>
  <cmn:softwareRelease>v1</cmn:softwareRelease>
</cmn:softwareProviderData>
<cmn:identifier IdentifierValueType="ACCIRD">0123456789</cmn:identifier>
```

**Figure 4: Schema aliases**

Field	Description
<b>softwareProvider</b>	The company that developed the software
<b>softwarePlatform</b>	The software package that is making the request
<b>softwareRelease</b>	The version of the software package
<b>IdentifierValueType</b>	The ID type being submitted which will be either "ACCIRD", "IRD" or "CST" depending on context and operation
<b>identifier</b>	The value submitted for this field should contain only digits, with no dashes. IRD Numbers that are eight digits must be padded with a leading zero.

**Table 4: Data fields**



Proper use:

- The only SoftwareProvider Data values that will be accepted are the ones that were provided to Inland Revenue at the time of onboarding.

The response structure for all requests will use the two default service response fields: **statusCode** and **errorMessage**. For the link/delink responses you will only receive these fields. For the retrieveList you will receive these fields as well as the client list being requested.

For example:

```
<linkDelinkResponse xmlns="urn:www.ird.govt.nz/GWS:types/ReturnCommon.v1">
  <StatusMessage xmlns="urn:www.ird.govt.nz/GWS:types/Common.v1">
    <statusCode>0</statusCode>
    <errorMessage></errorMessage>
  </StatusMessage>
</linkDelinkResponse>
```

**Figure 5: General response structure**

For a list of possible error codes and messages, see the ['Responses'](#) section of this document.

### 3.1 Link

The Link operation will be used to link a software package to its purchaser. This operation will be called once per relationship upon first use. The request for this operation is defined in the SoftwareIntermediation schema and is called **LinkRequest**.

The Identifier field for the Link operation will have the IdentifierValueTypes of either "ACCIRD" or "IRD". This value indicates the party to be linked to the software platform. This party will be either of the following:

- i. Tax Agency – IdentifierValueType IRD - in which case AccountType will be ignored
- ii. Customer Account – IdentifierValueType ACCIRD - in which case Account Type is required

The provided OAuth Token needs to have delegated permissions to this party.

Base structure:

```
<linkRequest
  xmlns="urn:www.ird.govt.nz/GWS:types/SoftwareIntermediation"
  xmlns:cmn="urn:www.ird.govt.nz/GWS:types/Common.v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:www.ird.govt.nz/GWS:types/SoftwareIntermediation">
  <cmn:softwareProviderData>
    <cmn:softwareProvider>SoftwareProvider</cmn:softwareProvider>
    <cmn:softwarePlatform>SoftwarePlatform</cmn:softwarePlatform>
    <cmn:softwareRelease>v1</cmn:softwareRelease>
  </cmn:softwareProviderData>
  <cmn:identifier IdentifierValueType="ACCIRD">0123456789</cmn:identifier>
  <cmn:accountType>GST</cmn:accountType>
  <softwarePlatformID>1500068629</softwarePlatformID>
</linkRequest>
```

**Figure 6: Link operation structure**

Field	Description
<b>identifier</b>	Required. This value indicates the party to be linked to the software platform. The value submitted for this field should contain only digits, with no dashes. IRD Numbers that are eight digits must be padded with a leading zero.
<b>AccountType</b>	Depending on context this field is required (for CUSTOMER) or ignored (for TAX AGENT) The account type in this payload will specify what account type to link to. This will only be done for links to CUSTOMERs and not AGENTs. (For distinction between the two see the RetrieveList operation)
<b>SoftwarePlatformID</b>	Required. This is the value provided by Inland Revenue during the registration process. This type of identifier is called a customer ID (a generic technical key for external parties in the Inland Revenue system) and is unique for every software package. It will be compared to the submitted software platform field.

**Table 5: Link operation data**

NOTE: When this operation is being called the user of the software **must** have owner or administrator access to the IRD number they are submitting for. The delegations are determined by the access currently granted in myIR. The Link operation will not allow the same link to be created twice. If there is any uncertainty that a link exists there is no harm in calling the operation again, just ensure the first Link call has had time to process. The generic response fragment documented above in figure 5 is the only response since the processing happens asynchronously.

### 3.2 Delink

The Delink operation will be used to delink a software user from a software package. This operation will be called once per link upon final use. The removal of a link should only occur upon cessation of a relationship, while the renewal of a subscription does not require this operation.



The Identifier field for the Delink operation will have the IdentifierValueTypes of either "IRD" or "CST" depending on who is submitting the Delink call. If the token is for someone with permissions on the software platform then the identifier value will contain the software platform ID (of type customer ID so IdentifierValueType="CST"), and the targetID field will contain the agent/account to be unlinked. If the token is for a person with permissions on a Tax Agent or Customer Account then this field will contain the corresponding ID and the targetId will contain the platform ID.

The provided OAuth Token needs to have delegated permissions on this party.

Base structure:

```
<delinkRequest
  xmlns="urn:www.ird.govt.nz/GWS:types/SoftwareIntermediation"
  xmlns:cmn="urn:www.ird.govt.nz/GWS:types/Common.v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:www.ird.govt.nz/GWS:types/SoftwareIntermediation">
  <cmn:softwareProviderData>
    <cmn:softwareProvider>SoftwareProvider</cmn:softwareProvider>
    <cmn:softwarePlatform>SoftwarePlatform</cmn:softwarePlatform>
    <cmn:softwareRelease>v1</cmn:softwareRelease>
  </cmn:softwareProviderData>

  <!-- Either -->
  <cmn:identifier IdentifierValueType="CST">1500123456</cmn:identifier>
  <targetId>123123123</targetId>
  <targetAccount>GST</targetAccount>

  <!-- OR -->
  <cmn:identifier IdentifierValueType="IRD">123123123</cmn:identifier>
  <targetId>1500123456</targetId>
  <targetAccount>GST</targetAccount>

</delinkRequest>
```

**Figure 7: Delink operation structure**

Field	Description
<b>identifier</b>	<p>If the token is for someone with permissions on the software platform then the identifier value will contain the software platform ID (of type customer ID so IdentifierValueType="CST"), and the targetID field must contain the IRD number of the Tax Agent/Customer Account to be delinked.</p> <p>If the token is for a person with permissions on a Tax Agent or Customer Account then this field will contain the corresponding ID and IdentifierValueType MUST be "IRD", and the targetId must contain the platform ID.</p> <p>The value submitted for this field should contain only digits, with no dashes. IRD Numbers that are eight digits must be padded with a leading zero.</p>



Field	Description
<b>AccountType</b>	Not used.
<b>targetId</b>	This field will be the other party to the link not already provided in the identifier. If the identifier is for the platform then the targetId will be that of software user. If the identifier is that of the agent of customer account then the targetId will be the CST of the software package.
<b>targetAccount</b>	This field will be used when the link being removed is an account-level link. If this field is not included in the payload and a customer-level link could not be found, account-level links will not be searched for. If the link was to a Tax Agent then this field will be ignored.

**Table 6: Delink operation data**

Proper Use:

There are two ways that this payload can be submitted, via the software provider or the software user. If the software user is requesting to delink, the software can submit this request one of two ways:

- A payload with "IRD" as the identifier type and the CST identifier for the platform as the targetID.
- Alternatively, the request can be made later by the software provider with the platform of "CST" type as the identifier and the other party IRD as the targetID.

The delinking of an account-level link requires the TargetAccount field to be populated. This means that any client with multiple account-level links will require multiple Delink calls for the different links.

**NOTE:** For special cases the Delink operation can be used to reissue bulk feed files. This operation will be called followed by the link operation. There is a short processing time required for the Delink operation to finish before the Link operation can be called. The generic response fragment documented above in figure 5 is the only response since the processing happens asynchronously.

### 3.3 RetrieveClientList

The RetrieveClientList operation will be used to retrieve all purchasers of a software package subscribed for bulk updates. This will only return active links and will not return delinked or expired relationships.

Base request structure:

```
<RetrieveListRequest
  xmlns="urn:www.ird.govt.nz/GWS:types/SoftwareIntermediation"
  xmlns:cmn="urn:www.ird.govt.nz/GWS:types/Common.v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:www.ird.govt.nz/GWS:types/SoftwareIntermediation">
  <cmn:softwareProviderData>
    <cmn:softwareProvider>SoftwareProvider</cmn:softwareProvider>
    <cmn:softwarePlatform>SoftwarePlatform</cmn:softwarePlatform>
    <cmn:softwareRelease>v1</cmn:softwareRelease>
  </cmn:softwareProviderData>
  <cmn:identifier IdentifierValueType="CST">0123456789</cmn:identifier>
```





```

<clientType>CUSTOMER</clientType>
<clientAccountType>GST</clientAccountType>
</RetrieveListRequest>
  
```

**Figure 8: RetrieveClientList operation structure**

Field	Description
<b>identifier</b>	<p>Indicates the software platform</p> <p>If this party cannot be resolved as a valid party, or if it can but the OAuth token is for someone who does not have permissions, in both cases a 4 unauthorised delegation status code will be returned.</p> <p>The value submitted for this field should contain only digits, with no dashes. IRD Numbers that are eight digits must be padded with a leading zero.</p>
<b>AccountType</b>	Not Used.
<b>clientType</b>	This type is to distinguish between AGENT lists and CUSTOMER lists. AGENT lists will return the links to Tax Agents and CUSTOMER lists will return the links directly to Customer Accounts.
<b>clientAccountType</b>	Optional: This field will allow you to filter by account type when clientType above is CUSTOMER.

**Table 7: RetrieveClientList operation data**

Base response structure:

```

<retrieveListResponse
  xmlns="urn:www.ird.govt.nz/GWS:types/SoftwareIntermediation"
  xmlns:cmn="urn:www.ird.govt.nz/GWS:types/Common.v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:www.ird.govt.nz/GWS:types/SoftwareIntermediation">
  <cmn:statusMessage>
    <cmn:statusCode>0</cmn:statusCode>
    <cmn:errorMessage/>
  </cmn:statusMessage>
  <softwareUser>
    <client>
      <clientType>AGENT</clientType>
      <idType>IRD</idType>
      <id>123088456</id>
    </client>

    <!-- OR -->

    <client>
      <clientType>CUST</clientType>
      <idType>ACCIRD</idType>
      <id>123089456</id>
      <accountType>GST</accountType>
    </client>
  
```





```

    <client>
      <clientType>CUST</clientType>
      <idType>ACCIRD</idType>
      <id>123089456</id>
      <accountType>INC</accountType>
    </client>
  </softwareUser>
</retrieveListResponse>

```

**Figure 9: Retrieve List Response structure**

Field	Description
<b>clientType</b>	Depending on the request, only a list of AGENTS or only a list of CUSTOMERS will be returned.
<b>idType</b>	This value will either be an Account IRD (ACCIRD) or IRD. Tax Agencies will be the only ones with IRD as the identifier.
<b>Id</b>	This is the value based on the idType.
<b>accountType</b>	This value only applies to ACCIRD idTypes.

## 4 End points, schemas and WSDLs

**IMPORTANT:** The end points, schemas and WSDLs listed here are subject to change. For the authoritative definitions, please refer to the information provided on the Inland Revenue Gateway Services GitHub site—<https://github.com/InlandRevenue/Gateway-Services>

### 4.1 End points

The end points for the Digital Test Environment XZT (sliced data):

Service	Environment	URL
<b>Authentication</b>	Cloud	<a href="https://q.services.ird.govt.nz">https://q.services.ird.govt.nz</a>
	Desktop/native app	<a href="https://q.services.ird.govt.nz">https://q.services.ird.govt.nz</a>
<b>Gateway Services</b>	Cloud	<a href="https://xzt.services.ird.govt.nz:4046/gateway/gws/SoftwareIntermediation/">https://xzt.services.ird.govt.nz:4046/gateway/gws/SoftwareIntermediation/</a>
	Desktop/native app	<a href="https://xzt.services.ird.govt.nz:4046/gateway/2/gws/SoftwareIntermediation/">https://xzt.services.ird.govt.nz:4046/gateway/2/gws/SoftwareIntermediation/</a>

**Table 8: Sliced data end points**

The end points for the Digital Test Environment XZS (unsliced data):

Service	Environment	URL
<b>Authentication</b>	Cloud/desktop/native apps	<a href="https://q.services.ird.govt.nz">https://q.services.ird.govt.nz</a>
<b>Gateway Services</b>	Cloud	<a href="https://xzs.services.ird.govt.nz:4046/gateway/gws/SoftwareIntermediation/">https://xzs.services.ird.govt.nz:4046/gateway/gws/SoftwareIntermediation/</a>
	Desktop/native app	<a href="https://xzs.services.ird.govt.nz:4046/gateway/2/gws/SoftwareIntermediation/">https://xzs.services.ird.govt.nz:4046/gateway/2/gws/SoftwareIntermediation/</a>

**Table 9: Unsliced data end points**

The end points for Production are as follows:

Service	Environment	URL
<b>Authentication</b>	Cloud/desktop/native apps	<a href="https://services.ird.govt.nz:443">https://services.ird.govt.nz:443</a>
<b>Gateway Services</b>	Cloud	<a href="https://services.ird.govt.nz:4046/gateway/gws/SoftwareIntermediation/">https://services.ird.govt.nz:4046/gateway/gws/SoftwareIntermediation/</a>
	Desktop/native app	<a href="https://services.ird.govt.nz:4046/gateway2/gws/SoftwareIntermediation/">https://services.ird.govt.nz:4046/gateway2/gws/SoftwareIntermediation/</a>

**Table 10: Production end points**



## 4.2 Schemas

All schemas for the Software Intermediation service import a common.xsd which has some data types specific to Inland Revenue. This common.xsd will be used in other gateway services outside of the /SoftwareIntermediation/ namespace so it must be kept up-to-date, without numerous redundant versions remaining.

The schemas for all operations will import SoftwareIntermediation.xsd for the request and response.

## 4.3 WSDLs

The Software Intermediation Gateway Service has one WSDL, which has a target namespace of <https://services.ird.govt.nz/GWS/SoftwareIntermediation/> and can be found at

<https://services.ird.govt.nz/GWS/SoftwareIntermediation/?singleWSDL>.

As explained in section **Error! Reference source not found. "Error! Reference source not found."** the online WSDLs have any elements, the development version of the WSDLs shipped with the build pack are more strongly typed to facilitate development and analysis and testing.

All WSDL messages follow this naming convention:

```
SoftwareIntermediation_<operation>_InputMessage

<wsdl:portType name="SoftwareIntermediation">
  <wsdl:operation name="Link">
  <wsdl:operation name="Delink">
  <wsdl:operation name="RetrieveList">
  <wsdl:service name="SoftwareIntermediation">
```

**Figure 10: WSDL naming conventions**

## 5 Responses

The response message from the Gateway Services always includes a status code and status message that describes how successfully the gateway service call was carried out. Following the status message will be the responseBody, which will return the operations response.

### 5.1 Generic gateway response codes

The following response codes are common to all gateway service calls:

Standard codes	Standard message	Description
<b>-1</b>	An unknown error has occurred	This is generally what will be returned for internal errors that are not due to the service request
<b>0</b>	Success	Standard success code is 0
<b>1</b>	Authentication failure	General authentication failure status
<b>2</b>	Missing authentication token(s)	Tokens were not included in the HTTP header as expected
<b>3</b>	Unauthorised access	Access is not permitted for the requester to use the gateway services. Access could not be confirmed due to OAuth token validation failing. This could be due to invalid format of the token or infrastructure being unavailable.
<b>4</b>	Unauthorised delegation	<p>Access is not permitted for the requester to perform this operation for the submitted identifier. This code will be returned in any of these situations:</p> <ul style="list-style-type: none"> <li>The submitted cmn:identifier has an invalid value.</li> <li>The identifier type (IdentifierValueType attribute on cmn:identifier) supplied is invalid.</li> <li>The AccountType supplied does not exist for that identifier.</li> </ul> <p>All the values above are valid but the provided OAuth token does not have delegated access to that Customer or Account.</p>
<b>5</b>	Unauthorised vendor	Vendor is not permitted access – has not been onboarded for this operation
<b>6</b>	Authentication Expired	Token authentication has expired and needs to be refreshed.
<b>20</b>	Unrecognised XML request	This could be due to the external sender sending in incorrect XML or it could be due to bad/poor/missing configuration
<b>21</b>	XML request failed validation	The external requestor submitted XML that is not formatted according to our defined



Standard codes	Standard message	Description
		schemas.
		In some scenarios where the request message does not have a well formed XML structure or is not valid or does not adhere to the SOAP protocol formats, the framework generates a parsing exception that is not wrapped in XML nor has a response status code.

**Table 11: Generic Gateway Service response codes**

## 5.2 Generic software intermediation response codes

The following response codes are specific to Software intermediation service calls:

Standard codes	Standard message	Description
<b>100</b>	Could not extract data from xml payload	Could not extract data from xml payload
<b>101</b>	Link already exists	A link already exists between the parties attempting to link.
<b>102</b>	No active link	A link does not exist between the parties attempting to delink.
<b>103</b>	Unable to save request	A request was not created. Submit request again.
<b>104</b>	Missing account type	An account IRD was submitted without an account type.
<b>105</b>	Cannot create customer level link	An IRD was submitted without an account type and the IRD does not belong to an agency
<b>106</b>	Missing target account	A target account type was not provided to find the account to delink from.
<b>108</b>	No account IRD provided for account	No account IRD provided for the given account type.
<b>109</b>	The specified customer has no links of that type.	The specified customer has no links of that type. Type is put in as either AGENT or CUSTOMER.
<b>110</b>	The CST provided could not be found.	The CST ID identifier could not be found.
<b>111</b>	No software registration found.	No software intermediary found for the given Software Platform Customer ID.
<b>112</b>	No agent or customer found.	No agent or customer found with provided identifier.
<b>113</b>	No account found.	No account found for the given account type.
<b>114</b>	The provided account type is not supported.	The provided account type is not authorized in the current operation.

**Table 12: Software intermediation response codes**



## 6 Use cases and scenarios

This section outlines possible scenarios or business use cases where Tax Agents or Customers might want to use this service (see [TDS Overview Build Pack for details of business use cases](#)).

### 6.1 Scenarios

Scenario	Typical sequence
<p><b>A. Link to subscribe for updates:</b> A Tax Agent as a client of a software intermediary wishes to receive TDS Bulk File Data for their Clients through their software provider software.</p> <p>Alternatively a direct customer of a software intermediary wishes to receive TDS Bulk File Data for themselves through their Software Provider software.</p> <p>This requires that they create a link between the Tax Agent/Customer and the software provider software through the Software Intermediation Service.</p>	<ol style="list-style-type: none"> <li>1. User representing the Tax Agent/Customer of a software intermediary signs onto software provider software and navigates to use Inland Revenue Gateway Services.</li> <li>2. Software Provider software user starts an independent browser session for the user to log onto the Inland Revenue site.</li> <li>3. At the end of this logon sequence an OAuth token is returned for use in further calls to the Gateway Services. See the <a href="#">Identity and Access Build Pack</a> for more information.</li> <li>4. The software provider software uses this token in a call to the Inland Revenue Software Intermediation Gateway Service <b>Link</b> Operation to request the creation of a link.</li> <li>5. Upon receipt of the request, the Software Intermediation service Link operation is invoked.</li> <li>6. The user receives confirmation the link has been created.</li> <li>7. User might do other work in the software provider software and eventually logs off and terminates session.</li> </ol>
<p><b>B. Delink:</b> A Tax Agent or Customer of the software intermediary wishes to stop receiving TDS Bulk File Data through their Software Provider software</p> <p>Alternately Software Providers wishes to break the link for a Customer or Tax Agent no longer using their software</p> <p>This requires that the Tax Agent or Customer delinks – i.e. removes the link between the client and the software intermediary through the Software Intermediation Service.</p> <p><b>Alternatively the user is no longer using the software</b></p>	<ol style="list-style-type: none"> <li>1. User representing the Tax Agent/Customer of a software intermediary signs onto software provider software and navigates to use Inland Revenue Gateway Services. Alternately an administrator of the software provider might log in.</li> <li>2. Software provider software starts an independent browser session for the user to log onto the Inland Revenue site.</li> <li>3. At the end of this logon sequence an OAuth token is returned for use in further calls to the Gateway Services. See the <a href="#">Identity and Access Build Pack</a> for more information.</li> <li>4. Alternatively to 1-3 if the delink is being done by the platform administrator when the user is already not using the software anymore then the administrator might go through a logon sequence to get an OAuth token to use.</li> <li>5. Alternately an automated batch process at the Software Provider might identify accounts of expired ex-users to delink. An OAuth token for an administrator might be kept perpetually active through refresh tokens for use in such jobs.</li> <li>6. The software provider software uses this token in a call to Inland Revenue Software Intermediation Gateway</li> </ol>



Scenario	Typical sequence
and an administrator for the software platform needs to delink,	<p>Service <b>Delink</b> Operation to request delinking.</p> <ol style="list-style-type: none"> <li>7. Upon receipt of the request, the Software Intermediation service Delink operation is invoked.</li> <li>8. The user receives confirmation the delink has been completed.</li> <li>9. User might continue working in the software.</li> </ol>
<p>C. <b>Retrieve Client List:</b> The Software Provider software wishes to see the list of all their Clients linked through this service according to Inland Revenue records</p>	<ol style="list-style-type: none"> <li>1. Software provider software starts an independent browser session for an administrator to log onto the Inland Revenue site. This user needs permissions on the Software Platform in order to retrieve its links, it doesn't need any permissions on the linked Customer Accounts or Tax Agents.</li> <li>2. At the end of this logon sequence an OAuth token is returned for use in further calls to the Gateway Services. See the <a href="#">Identity and Access Build Pack</a> for more information</li> <li>3. Alternately an automated batch process at the software provider might use an OAuth token. This might be an extended session using the Refresh token to facilitate automated batch jobs</li> <li>4. The Software Provider software uses this token in a call to Inland Revenue Software Intermediation Gateway Service <b>RetrieveClientList</b> Operation</li> <li>5. Upon receipt of the request, the Software Intermediation service RetrieveClientList operation is invoked</li> <li>6. The user receives a list of linked Clients</li> </ol>

Scenario	Typical sequence
<p>D. <b>Relink:</b> The Tax Agent, Customer or the Software Provider Software wishes to receive a full Bulk File for a Tax Agent/Customer who is already linked.</p> <p>As the Tax Agent/Customer is already linked the Bulk File received will only contain changes. If for some reason, such as data corruption the Tax Agent/Customer wishes to receive all their data again this can be accommodated by delinking the Tax Agent/Customer and then relinking.</p> <p>TDS will recognise the relinking as a new Tax Agent/Customer and will produce a file with all transaction data for that Tax Agent/Customer.</p>	<ol style="list-style-type: none"> <li>1. User representing the Tax Agent/Customer of a software intermediary signs onto Software Provider software and navigates to use Inland Revenue Gateway Services.</li> <li>2. Software provider software starts an independent browser session for the user to log onto the Inland Revenue site.</li> <li>3. At the end of this logon sequence an OAuth token is returned for use in further calls to the Gateway Services. See the <a href="#">Identity and Access Build Pack</a> for more information.</li> <li>4. Alternatively (to steps 1-3) Software Provider software administrator logon can be used. This might be an extended session using the Refresh token to facilitate automated batch jobs</li> <li>5. The Software Provider software uses this token in a call to Inland Revenue Software Intermediation Gateway Service <b>Delink</b> Operation to request delinking.</li> <li>6. Upon receipt of the request, the Software Intermediation service Delink operation is invoked.</li> <li>7. The user receives confirmation the delink has been completed.</li> <li>8. It may be necessary to wait for up to 10 minutes before completing this process and relinking. If action is taken before then it may result in a message saying the link is already present – i.e. the delinking has not taken full effect as yet.</li> <li>9. The Software Provider software again uses this token in a call to the Inland Revenue Software Intermediation Gateway Service <b>Link</b> Operation to request the creation of a link.</li> <li>10. Upon receipt of the request, the Software Intermediation service Link operation is invoked.</li> <li>11. The user receives confirmation the link has been created.</li> <li>12. The Bulk File Feed the next morning will contain a full file for the Tax Agency or Customer concerned.</li> </ol>

**Table 13: Software Intermediation scenarios**

## 6.2 Use cases

Systems Use Case	Operation
SUC041.Link Software Provider	Software Intermediation Build Pack, operation SoftwareIntermediation.Link
SUC042.Delink Software Provider	Software Intermediation Build Pack, operation SoftwareIntermediation.Delink
SUC043.Query Software Provider Links	Software Intermediation Build Pack, operation SoftwareIntermediation.RetrieveClientList



Summary Systems Use Case Link/Delink/Query Software Provider	
<b>User/Actors</b>	Software Provider software
<b>Secondary Actor</b>	
<b>Description</b>	The use case goal is to link/delink or request a Linked Client List and return the relevant response.
<b>Inland Revenue systems</b>	START
<b>Pre-Conditions</b>	
<b>Triggers</b>	Request received from Software Provider software to link, to delink or to receive a list of linked Clients.
<b>Constraints</b>	It is expected that the Software Provider has explicit consent from the Tax Agent or Customer to create a link between them.
<b>Post-Conditions</b>	Software Provider software will be sent a response from Inland Revenue that a subscription link is in place between the Software Provider software and the Tax Agency or Customer using their software, or a subscription link is no longer in place or a list of Clients for whom the link is in place is provided.
Use Case Scenarios	
1. Normal Flow	<ol style="list-style-type: none"> <li>1. Request received by the Software Intermediation service.</li> <li>2. Inland Revenue validates that the OAuth token presented is for a user that has the necessary delegated authority to administer the party in the cmd:Identifier field (see individual operation)</li> <li>3. The Software Intermediation Service creates the link between Software Provider Software and Tax Agent or Customer Account OR delinks OR extracts a list of Clients for whom the link is already in place.</li> <li>4. Inland Revenue Responds to request from software provider with completion status.</li> <li>5. Use case ends.</li> </ol>
2. Exception Flows	See Section 5 above for Error Codes.
3. Alternatives	For initial transition of existing Software Provider consumers a bulk linking process will be used. This is summarised in Use Case PUC202 in the <a href="#">TDS Overview Build Pack</a> .

**Table 14: Use case link/delink software provider**

## 7 Appendix A—Glossary

Acronym/term	Definition
<b>Authentication</b>	The process that verifies the identity of the party attempting to access Inland Revenue.
<b>Authorisation</b>	The process of determining whether a party is entitled to perform the function or access a resource.
<b>Build Pack</b>	Details the technical requirements and specifications, processes and sample payloads for the specified activity
<b>Client</b>	As used in this build pack client generally refers to the party licensing and using the software intermediary / software provider's software
<b>Credentials</b>	Information used to authenticate identity, for instance an account username and password.
<b>Customer</b>	<p>A Customer is the party who is a tax payer or a participant in the social policy products that are operated by Inland Revenue. The Customer might be a person (an "individual") or a non-individual entity such as a company, trust, society etc.</p> <p>Practically all of the service interactions with Inland Revenue are about a Customer (e.g. their returns, accounts, entitlements etc.) even though these interactions might be undertaken by an Intermediary such as a tax agent on their behalf.</p>
<b>Encryption</b>	Cryptographic transformation of data (called "plaintext") into a form (called "cipher text") that conceals the data's original meaning to prevent it from being known or used. If the transformation is reversible, the corresponding reversal process is called "decryption", which is a transformation that restores encrypted data to its original state. [RFC 2828]
<b>Endpoints</b>	A term used to describe a web service that has been implemented.
<b>GWS</b>	Gateway Services—the brand name for the suite of web services that Inland Revenue is providing. The Software intermediation service is a Gateway Service.
<b>HTTP, HTTPS</b>	Hyper Text Transmission Protocol (Secure)—the protocol by which web browsers and servers interact with each other. When implemented over TLS1.2 HTTP becomes HTTPS.
<b>IP</b>	Internet Protocol—the principal communication protocol in the Internet protocol suite for relaying datagrams across networks.
<b>NZISM</b>	NZ Information Security Manual—the security standards and best practices for Government agencies. Maintained by the NZ Government Communications Security Bureau (GCSB).
<b>OAuth 2.0</b>	OAuth 2.0 is an industry-standard protocol for authorization
<b>Pattern</b>	A constraint on data type values that require the string literal used in the data type's lexical space to match a specific pattern.
<b>Payloads</b>	This refers to the data contained within the messages that are exchanged when a web service is invoked. Messages consist of a header and a payload.

Acronym/term	Definition
<b>Schemas</b>	An XML schema defines the syntax of an XML document, in particular of a payload. The schema specifies what a valid payload (such as a GST return) must/can contain, as well as validating the payload.
<b>SHA</b>	Secure Hashing Algorithm. There is a family of them that provide different strengths. SHA-2 is currently favoured over SHA-1, which has been compromised.
<b>Software provider</b>	The organisation developing the software connecting to Inland Revenue gateway services Also known as software intermediary Also known as software developer Also known as service provider
<b>Software provider software</b>	A Client Application is an operating instance of Software that is deployed in one or more sites. A number of deployment patterns are possible: <ol style="list-style-type: none"> <li>1. A single cloud based instance with multiple tenants and online users,</li> <li>2. An on premise instance (e.g. an organisation's payroll system)</li> <li>3. A desktop application with an online user.</li> </ol> This is the computer software that contains interfaces to consume the services that Inland Revenue exposes. Software is developed and maintained by a software developer and subsequently deployed as one or more Client applications.
<b>SFTP</b>	Secure File Transport Protocol. SFTP 3.0 is used.
<b>Solution</b>	The technology components, systems and interface specifications constituting the Tax Agent Web Services capability which enables integration and communication across the Gateway channel between Inland Revenue and Tax Agents for the purpose of providing the Service.
<b>SOAP</b>	Simple Object Access Protocol—a set of standards for specifying web services. GWS uses SOAP version 1.2.
<b>SSL</b>	Secure Sockets Layer certificates—used to establish an encrypted connection between a browser or user's computer and a service or website.
<b>START</b>	Simplified Taxation and Revenue Technology— Inland Revenue's new core tax processing application. It is an implementation of the GenTax product from FAST Enterprises.
<b>Tax agent</b>	A Tax Agent who is formally registered as such with Inland Revenue.
<b>TDS</b>	Transaction Data Services
<b>TLS1.2</b>	Transport Layer Security version 1.2—the protocol that is observed between adjacent servers for encrypting the data that they exchange. Prior versions of TLS and all versions of SSL have been compromised and are superseded by TLS1.2.
<b>URL</b>	Universal Resource Locator—also known as a web address.
<b>User</b>	The user referred to in this document is the user of the software provider accounting or tax package. This user needs delegated



Acronym/term	Definition
	permissions on Customer tax accounts (potentially via a tax agency or other intermediary) in order to use TDS. The web logon used in eServices needs to be used in making Inland Revenue queries. This web logon must be granted permission there to access Customer Accounts
<b>WSDL</b>	Web Service Definition Language—an XML definition of a web service interface.
<b>X.509 certificate</b>	An international standard for encoding and describing a digital certificate. In isolation a public key is just a very large number, the X509 certificate to which it is bound identifies whose key it is, who issued it, when it expires etc. When a counterparty's X509 digital certificate is received, the recipient takes their public key out of it and store the key in their own keystore. The recipient can then use this key to encrypt and sign the messages that they exchange with this counterparty.
<b>XIAMS</b>	External IAMS—an instance of IAMS that authenticates and authorises access by external parties, for example customers, trading partners etc, as opposed to internal parties such as staff.
<b>XML</b>	Extensible Mark-up Language—a language used to define a set of rules used for encoding documents in a format that can be read by humans and machines.
<b>XSD</b>	XML Schema Definition—the current standard schema language for all XML data and documents.



## 8 Appendix B—Change log

This table lists all changes that have been made to this build pack document since the release of version 0.5.

Version	Date of change	Document section	Description
<b>0.8</b>	24/01/2018	1.1	Described purpose of service
		2.1	Described development WSDL
		2.4	Updated endpoint consumer identification
		3	Described identifier parameter and permissions in more detail
		5.1	Updated response codes 3-6 and added 104-109,114 Added new structure for Link and Delink request as well as linkDelinkResponse Response codes updated
<b>0.5</b>	04/12/2017		Draft created