Inland Revenue
Te Tari Taake

Inland Revenue

# Build pack: Software Intermediation service

**Date:** 28/09/2018

UNCLASSIFIED

# Contents

## List of figures



## List of tables

# 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 create links to tax agencies and/or to individual customer accounts. These links are then used by the Transaction Data Service (TDS) Bulk File process to determine what data gets sent to each software provider. In the case of tax agencies, the link is created at the client list level—all accounts within the list will then be returned in the bulk file process.

This document is intended to provide the technical details required to support the consumption of this Gateway Service. It describes the architecture of the technical solution, schemas, end points 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. Development versions of schemas, and sample requests and responses are also available with this build pack.

## 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.



UNCLASSIFIED

## 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 |
|---|---|---|
| **Inland Revenue** | 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. |
| **Software Provider** | 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 providers 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.

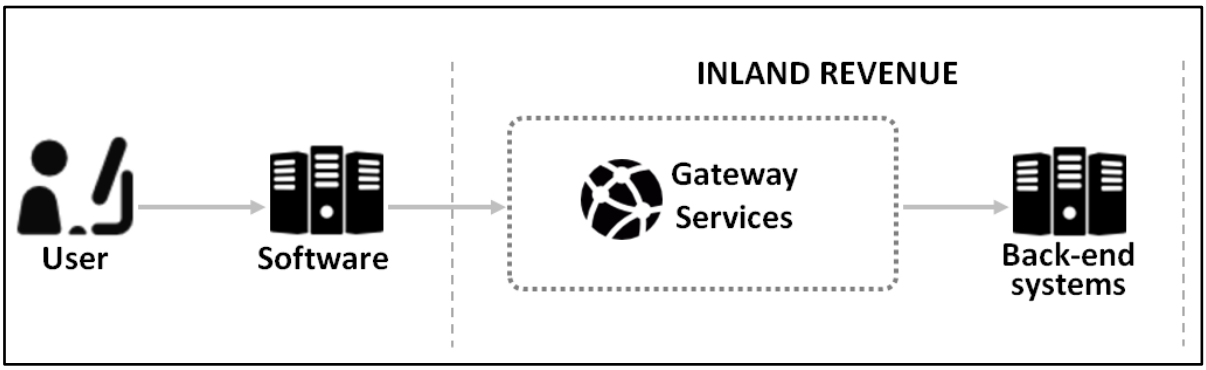


**Figure 1: Flow of data from user to Inland Revenue**

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 either a Tax Agency Client List or an individual Customer Account.
- **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 links.

## 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/RetrieveClientListRespo
        nse
        </a:Action>
    </s:Header>
    <s:Body>
        <si:RetrieveClientListResponse >
        <si:RetrieveClientListResult>
            <b:RetrieveClientListResponseWrapper>
                <cmn:RetrieveClientListResponse>
```



UNCLASSIFIED

```
                    <cmn:statusMessage>
                       <cmn:statusCode>0</statusCode>
                       <cmn:errorMessage/>
                    </cmn:statusMessage>
                 </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:

| | | | |
|---|---|---|---|
| **Encryption:** | Advanced Encryption Standard (AES) | FIPS 197 | 256-bit key |
| **Hashing:** | Secure Hash Algorithm (SHA-2) | FIPS 180-3 | SHA-256 |

**Table 2: Ciphers and key strengths**

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

1. There is an end point 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 end point, 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 end point that does not use mutual TLS. For this service, certificates do not need to be exchanged

during onboarding. On the desktop end point Inland Revenue has less ability to shield consumers of the service from heavy usage by other Inland Revenue partners.

| | End point for cloud-based connections | End point for desktop connections |
|---|---|---|
| **Purpose** | • This is the default end point to connect software providers to the Gateway Services. | • Additional end point 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. |
| **Client application type** | • Cloud applications. | • Desktop/native applications.<br>• For connecting from multiple decentralised clients. |
| **Constraints** | • Only for source locations with client side TLS certificates.<br>• On the cloud end point Inland Revenue has controls to shield software providers from issues caused by heavy usage from other providers. | • Less scalable.<br>• Subject to tighter security controls.<br>• On the desktop end point Inland Revenue has less ability to shield consumers of the service from heavy usage by others.<br>• OAuth2 refresh tokens will not be offered to desktop clients. |
| **Mutual TLS** | • 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. | • Server-side certificates only. |
| **Minimum TLS version** | • 1.2 | • 1.0(+) |
| **URL** | • Contains …/gateway/.. | • Contains …/gateway2/.. |
| **Port** | • 4046 | • 443 (Default https port) |
| **Web service consumer identification** | • Each software provider is given a software platform ID during onboarding. This ID is | • Each software provider is given a software platform ID during onboarding. This ID is of type Customer ID and |



UNCLASSIFIED

| | End point for cloud-based connections | End point for desktop connections |
|---|---|---|
| | of type Customer ID and is independent of the end point.<br>• To be identified in web service calls related to getting an OAuth token each cloud 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 end point.<br>• The mutual TLS certificate is used to identify the service provider | is not specific to the end point.<br>• Desktop clients will be given different identity and access system client_id/client_secret credentials to cloud application clients. |
| **Firewalling in production** | • No IP address restrictions.<br>• Access limited by certificate enrolment. | • No IP address restrictions. |
| **Firewalling in non-production environments** | • No IP address restrictions.<br>• Access limited by certificate enrolment. | • Firewalled—IP whitelisting needed. |

**Table 3: End points**

**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 a requestor but the user does not have permission, an error will be returned. This access will depend on the 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.v1.xsd
- Sft: SoftwareIntermediation.v1.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 |
|---|---|
| **softwareProvider** | The company that developed the software |
| **softwarePlatform** | The software package that is making the request |
| **softwareRelease** | The version of the software package |
| **IdentifierValueType** | The ID type being submitted which will be either "ACCIRD", "IRD" or "CST" depending on context and operation (see operations below for proper use) |
| **identifier** | 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: Software platform 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**. These will be the only two fields returned for the link/delink responses. For the retrieveList, these two fields will be returned as well as the requested client list.

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 'Response codes' 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:

- **Tax Agency Client List ID—IdentifierValueType IRD**, in which case AccountType will be ignored.
- **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 |
|---|---|
| **identifier** | *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. |
| **AccountType** | Depending on context this field is required (for CUSTOMER) or ignored (for TAX AGENT).<br>The account type in this payload will specify the account type to which to link. This will only be done for links to CUSTOMERs and not AGENTs. (For distinction between the two see the RetrieveList operation) |
| **SoftwarePlatformID** | *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 for which they are submitting. 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 delinked. 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 |
|---|---|
| **identifier** | 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.<br><br>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 |

| Field | Description |
|---|---|
| | IdentifierValueType **must** be "IRD", and the targetId **must** contain the platform ID.<br><br>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. |
| **AccountType** | Not used. |
| **targetId** | 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. |
| **targetAccount** | 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 |
|---|---|
| **identifier** | Indicates the software platform.<br>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.<br>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. |
| **AccountType** | Not used. |
| **clientType** | This type is to distinguish between AGENT lists and CUSTOMER lists. AGENT lists will return the links to tax agents client lists and CUSTOMER lists will return the links directly to customer accounts. |
| **clientAccountType** | Optional: This field will allow filtering 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 |
|---|---|
| **clientType** | Depending on the request, only a list of AGENT List IDs or only a list of CUSTOMERs will be returned. |
| **idType** | This value will either be an Account IRD (ACCIRD) or IRD. Tax agencies will be the only ones with IRD as the identifier. |
| **Id** | This is the value based on the idType. |
| **accountType** | This value only applies to ACCIRD idTypes. |



UNCLASSIFIED

# 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 |
|---|---|---|
| **Authentication** | Cloud | https://q.services.ird.govt.nz |
| | Desktop/native app | https://q.services.ird.govt.nz |
| **Gateway Services** | Cloud | https://xzt.services.ird.govt.nz:4046/gateway/gws/SoftwareIntermediation/ |
| | Desktop/native app | https://xzt.services.ird.govt.nz:4046/gateway2/gws/SoftwareIntermediation/ |

**Table 8: Sliced data end points**

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

| Service | Environment | URL |
|---|---|---|
| **Authentication** | Cloud/desktop/native apps | https://q.services.ird.govt.nz |
| **Gateway Services** | Cloud | https://xzs.services.ird.govt.nz:4046/gateway/gws/SoftwareIntermediation/ |
| | Desktop/native app | https://xzs.services.ird.govt.nz:4046/gateway2/gws/SoftwareIntermediation/ |

**Table 9: Unsliced data end points**

The end points for Production are as follows:

| Service | Environment | URL |
|---|---|---|
| **Authentication** | Cloud/desktop/native apps | https://services.ird.govt.nz:443 |
| **Gateway Services** | Cloud | https://services.ird.govt.nz:4046/gateway/gws/SoftwareIntermediation/ |
| | Desktop/native app | https://services.ird.govt.nz:4046/gateway2/gws/SoftwareIntermediation/ |

**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.v1.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 the Solution design section of this document, the online WSDLs have 'any' elements underneath the wrapper elements. There is a development version of the WSDL available with this build pack that replaces the 'any' element with an imported reference to the schema to facilitate initial development and testing. To consume the actual service, the binding will need to be done at the hosted end point. However, for initial development this static WSDL can be used.

All WSDL messages follow this naming convention:

```
SoftwareIntermdiation_<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 Response codes

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. The operations for the Software Intermediation service all apply customer-level security validation at the framework level and the descriptions for these codes reflect that.

| Standard codes | Standard message | Description |
|---|---|---|
| **-1** | An unknown error has occurred | This is generally what will be returned for internal errors that are not due to the service request |
| **0** | Success | Standard success code is 0 |
| **1** | Authentication failure | General authentication failure status |
| **2** | Missing authentication token(s) | Tokens were not included in the HTTP header as expected |
| **3** | 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. |
| **4** | Unauthorised delegation | 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:<br>• The submitted cmn:identifier has an invalid value.<br>• The identifier type (IdentifierValueType attribute on cmn:identifier) supplied is invalid.<br>• All the values above are valid but the provided OAuth token does not have delegated access to that Customer.<br>• **The Delink operation will NOT return this code but will return 113 as per below**. |
| **5** | Unauthorised vendor | Vendor is not permitted access—has not been onboarded for this operation |

| Standard codes | Standard message | Description |
|---|---|---|
| **6** | Authentication Expired | Token authentication has expired and needs to be refreshed.<br>Note that this will only be provided for a token that has been successfully used before. For an expired unused token 3 above will be returned. |
| **20** | 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 |
| **21** | XML request failed validation | The external requestor submitted XML that is not formatted according to our defined schemas |
| **(none)** | (non XML) | 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 |
| **(none)** | (SOAP fault)<br>UnAuthorised | When maximum concurrency has been exceeded by the service provider this SOAP fault will be returned |

**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 | Link | Delink | Retrieve |
|---|---|---|---|---|---|
| **100** | Could not extract data from xml payload | Could not extract data from XML payload | X | X | X |
| **101** | Link already exists | A link already exists between the parties attempting to link | X | | |
| **102** | No active link | A link does not exist between the parties attempting to delink | | X | |
| **103** | Unable to save request | A request was not created. Submit request again. | X | X | |
| **104** | Missing account type | An account IRD number was submitted without an account type | | X | |
| **105** | Cannot create customer level link | An IRD number was submitted without an account type and the IRD number does not belong to a tax agency | X | | |

| Standard codes | Standard message | Description | Link | Delink | Retrieve |
|---|---|---|---|---|---|
| **106** | Missing target account | A target account type was not provided to find the account from which to delink | | X | |
| **108** | No account IRD provided for account | No account IRD number (ACCIRD) provided for the given account type | X | | |
| **109** | The specified software has no links of that type | The specified software intermediary has no links of that type. Link types are either AGENT or CUSTOMER. | | | X |
| **111** | No software registration found | No software intermediary found for the given Software Platform Customer ID | X | X | |
| **112** | No agent or customer found | No agent or customer found with provided identifier | | X | |
| **113** | No account found | No account found for the given account type | | X | |
| **114** | The provided account type is not supported | The provided account type is not authorised in the current operation | X | | |

**Table 12: Generic 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 (please refer to the TDS Overview Build Pack for details of business use cases).

## 6.1 Scenarios

| Scenario | Typical sequence |
|---|---|
| **A. Link to subscribe for updates:**<br>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.<br><br>Alternatively a software provider wishes to receive TDS bulk file data for a user of their software.<br><br>This requires that they create a link between the tax agent/customer and the software provider software through the Software Intermediation Service. | 1. User representing the tax agents /customer of a software intermediary signs onto software provider software and navigates to use Inland Revenue Gateway Services.<br>2. Software provider software user starts an independent browser session for the user to log onto the Inland Revenue site.<br>3. At the end of this logon sequence an OAuth token is returned for use in further calls to the Gateway Services. (See the Identity and Access Build Pack for more information.)<br>4. The software provider software uses this token in a call to the Inland Revenue Software Intermediation Gateway Service **Link** Operation to request the creation of a link.<br>5. Upon receipt of the request, the Software Intermediation service Link operation is invoked.<br>6. The user receives confirmation the link has been created.<br>7. User might do other work in the software provider software and eventually logs off and terminates session. |
| **B. Delink:**<br>A tax agent of the software intermediary wishes to stop receiving TDS bulk file data through their software provider software.<br><br>Alternately software providers wishes to break the link for a customer or tax agent no longer using their software.<br><br>This requires that the tax agent or customer delinks (for example, removes the link between the client and the software intermediary through the Software Intermediation Service. | 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.<br>2. Software provider software starts an independent browser session for the user to log onto the Inland Revenue site.<br>3. At the end of this logon sequence an OAuth token is returned for use in further calls to the Gateway Services. (See the Identity and Access Build Pack for more information.)<br>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.<br>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 |

| Scenario | Typical sequence |
|---|---|
| Alternatively the user is no longer using the software and wants to delink from the software platform. | kept perpetually active through refresh tokens for use in such jobs.<br>6. The software provider software uses this token in a call to Inland Revenue Software Intermediation Gateway Service **Delink** Operation to request delinking.<br>7. Upon receipt of the request, the Software Intermediation service Delink operation is invoked.<br>8. The user receives confirmation the delink has been completed.<br>9. User might continue working in the software. |
| **C. Retrieve Client List:** The software provider software wishes to see the list of all their Clients linked through this service according to Inland Revenue records | 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.<br>2. At the end of this logon sequence an OAuth token is returned for use in further calls to the Gateway Services. (See the Identity and Access Build Pack for more information.)<br>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.<br>4. The software provider software uses this token in a call to Inland Revenue Software Intermediation Gateway Service **RetrieveClientList** Operation.<br>5. Upon receipt of the request, the Software Intermediation service RetrieveClientList operation is invoked.<br>6. The user receives a list of linked clients. |

| Scenario | Typical sequence |
|---|---|
| **D. Relink:**<br>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.<br><br>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.<br><br>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. | 1. User representing the tax agent/customer of a software intermediary signs onto software provider software and navigates to use Inland Revenue Gateway Services.<br>2. Software provider software starts an independent browser session for the user to log onto the Inland Revenue site.<br>3. At the end of this logon sequence an OAuth token is returned for use in further calls to the Gateway Services. (See the Identity and Access Build Pack for more information.)<br>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.<br>5. The software provider software uses this token in a call to Inland Revenue Software Intermediation Gateway Service **Delink** Operation to request delinking.<br>6. Upon receipt of the request, the Software Intermediation service Delink operation is invoked.<br>7. The user receives confirmation the delink has been completed.<br>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 (for example, the delinking has not yet taken full effect).<br>9. The software provider software again uses this token in a call to the Inland Revenue Software Intermediation Gateway Service **Link** Operation to request the creation of a link.<br>10. Upon receipt of the request, the Software Intermediation service Link operation is invoked.<br>11. The user receives confirmation the link has been created.<br>12. The bulk file feed the next morning will contain a full file for the tax agency or customer concerned. |

**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 | |
|---|---|
| **User/actors** | Software provider software |
| **Secondary actor** | |
| **Description** | The use case goal is to link/delink or request a linked client list and return the relevant response. |
| **Inland Revenue systems** | START |
| **Pre-conditions** | |
| **Triggers** | Request received from software provider software to link, to delink or to receive a list of linked clients. |
| **Constraints** | It is expected that the software provider has explicit consent from the tax agent or customer to create a link between them. |
| **Post-conditions** | 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** | 1. Request received by the Software Intermediation service.<br>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)<br>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 ink is already in place.<br>4. Inland Revenue Responds to request from software provider with completion status.<br>5. Use case ends. |
| **2. Exception flows** | See Section 5 above for response 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 TDS Overview Build Pack. |

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

# 7 Glossary

| Acronym/term | Definition |
|---|---|
| **Authentication** | The process that verifies the identity of the party attempting to access Inland Revenue |
| **Authorisation** | The process of determining whether a party is entitled to perform the function or access a resource |
| **Build pack** | Details the technical requirements and specifications, processes and sample payloads for the specified activity |
| **Client** | As used in this build pack client generally refers to the party licensing and using the software intermediary/software provider's software |
| **Credentials** | Information used to authenticate identity, for instance an account username and password |
| **Customer** | 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.<br>Practically all of the service interactions with Inland Revenue are about a customer (such as their returns, accounts, entitlements etc) even though these interactions might be undertaken by an Intermediary such as a tax agent on their behalf. |
| **Encryption** | 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 (see RFC 2828). |
| **End points** | A term used to describe a web service that has been implemented |
| **GWS** | Gateway Services—the brand name for the suite of web services that Inland Revenue is providing. The Software intermediation service is a Gateway Service. |
| **HTTP, HTTPS** | 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. |
| **IP** | Internet Protocol—the principal communication protocol in the Internet protocol suite for relaying datagrams across networks |
| **NZISM** | NZ Information Security Manual—the security standards and best practices for Government agencies. Maintained by the NZ Government Communications Security Bureau (GCSB). |
| **OAuth 2.0** | OAuth 2.0 is an industry-standard protocol for authorisation |
| **Pattern** | A constraint on data type values that require the string literal used in the data type's lexical space to match a specific pattern |
| **Payloads** | 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. |



UNCLASSIFIED

| Acronym/term | Definition |
|---|---|
| **Schemas** | 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. |
| **SHA** | Secure Hashing Algorithm. There is a family of these that provide different strengths. SHA-2 is currently favoured over SHA-1, which has been compromised. |
| **Software provider** | The organisation developing the software connecting to Inland Revenue gateway services—also known as software intermediary, software developer or service provider |
| **Software provider software** | A client application is an operating instance of software that is deployed in one or more sites. A number of deployment patterns are possible:<br>1. A single cloud based instance with multiple tenants and online users,<br>2. An on premise instance (eg an organisation's payroll system)<br>3. A desktop application with an online user.<br>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. |
| **SFTP** | Secure File Transport Protocol. SFTP 3.0 is used. |
| **Solution** | 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 |
| **SOAP** | Simple Object Access Protocol—a set of standards for specifying web services. GWS uses SOAP version 1.2 |
| **SSL** | Secure Sockets Layer certificates—used to establish an encrypted connection between a browser or user's computer and a service or website |
| **START** | Simplified Taxation and Revenue Technology—Inland Revenue's new core tax processing application. It is an implementation of the GenTax product from FAST Enterprises. |
| **Tax agent** | A tax agent who is formally registered as such with Inland Revenue |
| **TDS** | Transaction Data Services |
| **TLS1.2** | 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. |
| **URL** | Universal Resource Locator—also known as a web address |
| **User** | The user referred to in this document is the user of the software provider accounting or tax package. This user needs delegated permissions on customer tax accounts (potentially via a tax agency or other intermediary) in order to use TDS. The web logon used in |

| Acronym/term | Definition |
|---|---|
| | eServices needs to be used in making Inland Revenue queries. This web logon must be granted permission there to access customer accounts. |
| **WSDL** | Web Service Definition Language—an XML definition of a web service interface |
| **X.509 certificate** | 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. |
| **XIAMS** | External IAMS—an instance of IAMS that authenticates and authorises access by external parties, for example customers or trading partners, as opposed to internal parties such as staff. |
| **XML** | 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. |
| **XSD** | XML Schema Definition—the current standard schema language for all XML data and documents. |

# 8 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 |
|---|---|---|---|
| **VERSION NUMBERS REMOVED HENCEFORTH** | 28/09/2018 | Various | • Design change relating to linking at Client List level rather than Agency level |
| **1.00** | 13/04/2018 | Entire doc | • Version 1.0<br>• Minor formatting/cosmetic changes etc. |
| | 11/04/2018 | 4.3 | • Information about WSDLs updated |
| **0.82** | 08/03/2018 | 5.2 | • Added three columns to table of generic software intermediation response codes, plus data:<br>  ○ Link<br>  ○ Delink<br>  ○ RetrieveList |
| | | 6.1 and 6.2 | • Scenarios and use cases updated |
| **0.81** | 05/03/2018 | 5 | • Added status code description detail for code 4 and maximum concurrency |
| **0.80** | 2/02/2018 | Title page, footers | • Corrected classification to UNCLASSIFIED |
| | 24/01/2018 | 1.1 | • Described purpose of service |
| | | 2.1 | • Described development WSDL |
| | | 2.4 | • Updated end point consumer identification |
| | | 3 | • Described identifier parameter and permissions in more detail<br>• Updated response codes 3-6 and added 104-109,114 |
| | | 5.1 | • Added new structure for Link and Delink request as well as linkDelinkResponse<br>• Response codes updated |
| **0.5** | 04/12/2017 | | • Draft created |