Inland Revenue
Te Tari Taake

Business Transformation

Inland Revenue

Build pack: Software intermediation service

Date:	04/12/2017
Version:	v0.5

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 our service providers that support 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.

This is a stand-alone document intended to provide the technical details required to support the end-to-end onboarding Gateway Services. It describes the architecture of the technical solution, schemas, endpoints, sample payloads to use in non-production environments, 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.

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 Transaction data services (TDS) onboarding section 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 service 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 version 1.5 of 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.
Service provider	Acquire a X.509 certificate from a competent authority for the Test and Production environments	This is required when using mutual TLS with cloud-based service providers.

Table 1: Prerequisites

Refer to the onboarding documents and sections mentioned above for more details.

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.

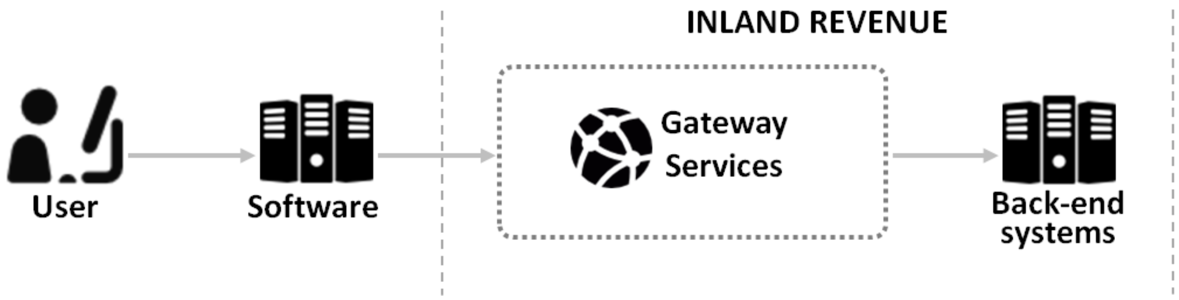

Figure 1 : Flow of data from user to Inland Revenue

The 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.

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 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">
    <s:Header>
        <a:Action s:mustUnderstand="1">
        https://services.ird.govt.nz/GWS/SoftwareIntermediation/SoftwareIntermediation/RetrieveClientListResponse
        </a:Action>
    </s:Header>
    <s:Body>
         <RetrieveClientListResponse xmlns="https://services.ird.govt.nz/GWS/SoftwareIntermediation/">
          <RetrieveClientListResult xmlns:b=https://services.ird.govt.nz/GWS/SoftwareIntermediation
        /:types/RetrieveClientListResponse xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
              <b:RetrieveClientListResponseWrapper>
                 <RetrieveClientListResponse xmlns="urn:www.ird.govt.nz/GWS:types/Common.v1">
                   <statusMessage>
                      <statusCode>0</statusCode>
                      <errorMessage/>
                   </statusMessage>
                 </RetrieveClientListResponse>
               </b:RetrieveClientListResponseWrapper>
           </RetrieveClientListResult>
        </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 endpoints, which are summarised in the bullet points below (the table immediately afterwards provides more detail):

1. There is an endpoint to which service 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 service providers from issues caused by heavy usage from other providers.

2. For service 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 others.

	Endpoint for cloud-based connections	**Endpoint for desktop connections**
Purpose	• Primary preferred endpoint to connect to from service providers for Gateway Services.	• Additional transitory 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.
Client application type	• Cloud applications.	• Desktop/native applications. • For connecting from multiple decentralised clients.
Constraints	• Only for source locations with client side TLS certificates. • On the cloud endpoint Inland Revenue has controls to shield service providers from issues caused by heavy usage from other providers.	• Less scalable. • Subject to tighter security controls. • On the desktop endpoint Inland Revenue has less ability to shield consumers of the service from heavy usage by others. • OAuth2 refresh tokens will not be offered to desktop clients.
Mutual TLS	• Inland Revenue explicitly trusts the certificate the service provider associates with the TLS	• Server side certificates only.

	Endpoint for cloud-based connections	Endpoint for desktop connections
	connection as client for Mutual TLS connections and uses it to identify the service provider in conjunction with the web service identification below.	
Minimum TLS version	• 1.2	• 1.0(+)
URL	• Contains …/gateway/..	• Contains …/gateway2/..
Port	• 4046	• 443 (Default https port)
Web service consumer identification	• To be identified in web service calls—each cloud application will be given client_id/client_secret credentials during onboarding to allow it to call this endpoint.	• Desktop clients will be given different client_id/client_secret credentials to cloud application clients.
Firewalling in production	• No IP address restrictions. • Access limited by certificate enrolment.	• No IP address restrictions.
Firewalling in non-production environments	• No IP address restrictions. • Access limited by certificate enrolment.	• Firewalled—IP whitelisting needed.

Table 3: Endpoints

Delegated permissions: The services will allow one to retrieve all of the data for a customer that the calling user (as represented by the OAuth token) has access to. There may be additional accounts this identity does not have access to, those will not be mentioned. If an account or data in it 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. If the token represents a user in a tax agency or other intermediary, then the agent-client linking is also considered.

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 "CST" and the value will be the Inland Revenue-provided customer ID. This customer ID is unique for every software package and will be compared to the submitted software platform field.

For example:

```
<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>
```

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 should be "CST". The value submitted for this field should contain only digits, with no dashes.

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 relationshipManager 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:

```
<relationshipManagerResponse 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>
</relationshipManagerResponse>
```

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 **RelationshipManagerRequest**.

Base structure:

```
<relationshipManagerRequest
           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>
   <cmn:accountType></cmn:accountType>
   <clientIRD>123068629</clientIRD>
</relationshipManagerRequest>
```

Figure 5: Link operation structure

Field	Description
AccountType	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)
clientIRD	This field defines who the software package should be linked to. This will most likely be the IRD number of the purchaser of the software. For tax agencies this will be the IRD of the agency.

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. This access is 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.

3.2 Delink

The Delink operation will be used to delink a software package to its purchaser. This operation will be called once per relationship 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 request for this operation is the same as Link and is called **RelationshipManagerRequest**.

Base structure:

```
<relationshipManagerRequest
            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>
   <clientIRD>123068629</clientIRD>
</relationshipManagerRequest>
```

Figure 6: Delink operation structure

Field	Description
clientIRD	This field defines who the software package should be delinked from. This will be the IRD that is currently linked to the software. For tax agencies this will be the IRD of the agency.
AccountType	If the IRD number is for a Customer then this field is needed to define the customer account to delink from, similar to how it was used in the link operation. When delinking form an AGENT this field has no meaning

Table 6: Delink operation data

NOTE: For special cases this 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.

3.3 RetrieveClientList

The RetrieveClientList operation will be used to retrieve all purchasers of a software package. This will only return active links and will not return delinked 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>AGENT</clientType>
</RetrieveListRequest>
```

Figure 7: RetrieveClientList operation structure

Field	Description
clientType	This type is to distinguish between AGENT lists and CUSTOMER lists. AGENT lists will return the agents. CUSTOMER lists will return Customer Accounts

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>
```

Field	Description
clientType	Depending on the request, only a list of AGENTs or only a list of CUSTOMERs will be returned.
idType	This value will either be an account IRD or an IRD. 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.

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.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.

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 8: 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
-1	An unknown error has occurred	This error will be logged by the Gateway Services and evaluated the next business day.
0	Success	This resembles a successful web service call.
1	Authentication failure	Authentication failure means the token provided is not a valid token.
2	Missing authentication token(s)	No oAuth token in HTTP header.
3	Unauthorised access	The logon making the call does not have access to make the request on behalf of the client or agency.
5	Unauthorised vendor	The vendor provided is not authorised to use these suite of services.
20	Unrecognised XML request	The XML submitted is not recognisable and no schema can be determined.
21	XML request failed validation	The XML structure did not meet the definition laid out by the schemas published by Inland Revenue.

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
100	Invalid request data	Could not extract data from xml payload. Review payload and try again.
101	Link already exists	A link already exists between the parties attempting to link.
102	No active link	A link does not exist between the parties attempting to delink.
103	Unable to save request	A request was not created. Submit request again.

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
A. **Link to subscribe for updates:** A Tax Agent as a client of a Software Intermediary wishes to receive TDS Bulk File Data for their Clients through their Service Provider or Accounting software. Alternatively a direct Customer of a Software Intermediary wishes to receive TDS Bulk File Data for themselves through their Service Provider or Accounting software. This requires that they create a link between the Tax Agent/Customer and the Service Provider or Accounting software through the Software Intermediation Service.	1. User representing the Tax Agent/Customer of a Software Intermediary signs onto Service Provider or Accounting software and navigates to use Inland Revenue Gateway Services. 2. Service Provider or Accounting software user starts an independent browser session for the user to log onto the Inland Revenue site. 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. 4. The Service Provider or Accounting software uses this token in a call to the Inland Revenue Software Intermediation Gateway Service **Link** Operation to request the creation of a link. 5. Upon receipt of the request, the Software Intermediation service Link operation is invoked. 6. The user receives confirmation the link has been created. 7. User might do other work in the Service Provider or Accounting software and eventually logs off and terminates session.
B. **Delink:** A Tax Agent or Customer of the software intermediary wishes to stop receiving TDS Bulk File Data through their Service Provider or Accounting software Alternately Service providers wishes to break link for customer or tax agent no longer using their software 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.	1. User representing the Tax Agent/Customer of a Software Intermediary signs onto Service Provider or Accounting software and navigates to use Inland Revenue Gateway Services. Alternately an administrator of the service provider might log in. 2. Service Provider or Accounting software starts an independent browser session for the user to log onto the Inland Revenue site. 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. 4. The Service Provider or Accounting software uses this token in a call to Inland Revenue Software Intermediation Gateway Service **Delink** Operation to request delinking. 5. Upon receipt of the request, the Software Intermediation service Delink operation is invoked. 6. The user receives confirmation the delink has been completed. 7. User might continue working in the software.

Scenario	Typical sequence
C. **Retrieve Client List:** The Service Provider or Accounting software wishes to see the list of all their Clients linked through this service according to Inland Revenue records	1. Service Provider or Accounting software starts an independent browser session for a user to log onto the Inland Revenue site – typically an administrator 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 3. This might be an extended session using the Refresh token to facilitate automated batch jobs 4. The Service Provider software uses this token in a call to Inland Revenue Software Intermediation Gateway Service **RetrieveClientList** Operation 5. Upon receipt of the request, the Software Intermediation service RetrieveClientList operation is invoked 6. The user receives a list of linked Clients
D. **Relink:** The Tax Agent, Customer or the Service Provider or Accounting software wishes to receive a full Bulk File for a Tax Agent/Customer who is already linked. 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 Agency/Customer wishes to receive all their data again this can be accommodated by delinking the Tax Agency/Customer and then relinking. TDS will recognise the relinking as a new Tax Agent/Customer and will produce a file with all START data for that Tax Agent/Customer.	1. User representing the Tax Agent/Customer of a Software Intermediary signs onto Service Provider or Accounting software and navigates to use Inland Revenue Gateway Services. 2. Service Provider or Accounting software starts an independent browser session for the user to log onto the Inland Revenue site. 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. 4. Alternatively (to steps 1-3) Service Provider or Accounting software administrator logon can be used. This might be an extended session using the Refresh token to facilitate automated batch jobs 5. The Service Provider or Accounting software uses this token in a call to Inland Revenue Software Intermediation Gateway Service **Delink** Operation to request delinking. 6. Upon receipt of the request, the Software Intermediation service Delink operation is invoked. 7. The user receives confirmation the delink has been completed. (It may be necessary to wait for 10 mins 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). 8. The Service Provider or Accounting 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. 9. Upon receipt of the request, the Software Intermediation service Link operation is invoked. 10. The user receives confirmation the link has been created. 11. The Bulk File Feed the next morning will contain a full file for the Tax Agency or Customer concerned.

Table 13: Real time scenarios and use cases

6.2 Use cases

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

Summary Systems Use Case Link Service Provider	
User/Actors	Service Provider or Accounting 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 Service Provider or Accounting software to link, to delink or to receive a list of linked Clients.
Constraints	It is expected that the Service Provider has explicit consent from the Tax Agent or Customer to create a link between them.
Post-Conditions	Service Provider or Accounting Software will be sent a response from Inland Revenue that a subscription link is in place between the Service Provider or the Accounting software and the Tax Agency or Customer using their software, a subscription link is in place or no longer in place or a list of Clients for whom the link is provided.
Use Case Scenarios	
1. Normal Flow	1. Request received by the Software intermediation service. 2. Inland Revenue validates that the OAuth token presented is for a user that has the necessary delegated authority to see all the data for the Tax Agency ID or Customer ID presented 3. The Software Intermediation Service creates the link between Service Provider or Accounting Software and Tax Agent or Customer Account OR delinks or extracts a list of Clients for whom the ink is already in place. 4. Inland Revenue Responds to request from Service Provider with completion status. 5. Use case ends.
2. Exception Flows	See Section 5 above for Error Codes.
3. Alternatives	For initial transition of existing service provider consumers a bulk linking process will be used. This is summarised in Use

Summary Systems Use Case Link Service Provider	
	Case PUC202 in the TDS Overview Build Pack.

Table 14: Use case link/delink service provider

7 Appendix A—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. 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.
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. [RFC 2828]
Endpoints	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 authorization
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.

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 them that provide different strengths. SHA-2 is currently favoured over SHA-1, which has been compromised.
Service Provider	The organisation developing the software connecting to Inland Revenue gateway services Also known as Software Intermediary Also known as Software Developer Also known as Software Provider
Service 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: 1. A single cloud based instance with multiple tenants and online users, 2. An on premise instance (e.g. an organisation's payroll system) 3. A desktop application with an online user. 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

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
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, trading partners etc, 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 Appendix B—Change log

This table lists all changes that have been made to this build pack document since [DATE].

Version	Date of change	Document section	Description
0.5	04/12/2017		Draft created