Oracle Enterprise Taxation and Policy Management - Self Service

Integration Guide

Release 1.0.0.0

E36013-01

October 2012

ORACLE

Oracle Enterprise Taxation and Policy Management - Self Service Integration Guide

Release 1.0.0.0

E36013-01

October 2012

Contents

Chapter 1

Self Service Integration

Oracle Enterprise Taxation and Policy Management ("the product") provides an integration with Oracle Enterprise Taxation and Policy Management Self Service ("the self service product"). Refer to the self service product documentation for an overview of the implementation for that application. The following sections describes the functionality provided in the product to support the integration and outlines the configuration required to support the functionality.

The Big Picture

This section describes high level topics related to the integration.

Message Processing Overview

Oracle Enterprise Taxation and Policy Management interacts with the self service product via web services. Inbound web services are processed using XAI. An appropriate XAI inbound service is invoked from the self service product via BPEL. The XAI inbound service is configured with a processing service script that knows how to process the request. Requests fall into one of the following categories:

- A taxpayer request. These types of requests fall into one or more sub-categories:
  - A taxpayer inquiry. If the taxpayer requests information like to find out the status of a refund, the XAI inbound service references a service script that performs the appropriate logic. Most of the time the service script associated with the XAI inbound service can perform the logic needed to retrieve the information to present to the taxpayer. However, there may be cases where the service script creates a real time Service Task to retrieve the requested information and return it.

    Fastpath: Refer to the *Refund Status* for an example of a request for information that results in the creation of a service task.

  - A request to perform some transaction processing. For example, the taxpayer may submit a payment via self service. In this case, the service script associated with the XAI inbound service creates a Service Task record that will subsequently perform the necessary validation and processing. The taxpayer does not wait for the service task to complete its lifecycle.

- An internal web service request submitted by the self service product to get information. For example, before submitting a payment the self service product may require the taxpayer to provide proof of identity to verify that the taxpayer is known to the system. The taxpayer identification information is sent to this product via a web service and is processed real time.

> **Note:** Naming convention. The product XAI inbound services follow a naming convention of an XAI inbound service name of TS%. Specific XAI inbound services are mentioned throughout the supported functionality.

Confirmation Message

All web services that represent a taxpayer request and result in the creation of a service task within the product are designed to provide the taxpayer with a confirmation ID and a confirmation message. Additional confirmation details may be added by logic in the product that perform actions. This may be used when a taxpayer wants to inquire about the status of an action or request using the confirmation ID. A web service inquiring on the status of a task can retrieve the confirmation details to display to the taxpayer. In addition, if the taxpayer has questions at a later date about a request, the confirmation ID may be used when contacting the tax authority about the request.

The common structure captured with each message (stored in the data area **C1-SelfServiceConfirmMessage**) is as follows:

- Confirmation ID. This may be generated prior to sending the web service to the product. For example, in many cases BPEL generates a confirmation ID and adds it to the message.
- Confirmation Header. This is populated with the message category / message number used as the acknowledgement message. For example: 'Your request has been submitted successfully, reference ID 1238098'. It is defined on the service task's task type.
- Confirmation Details. This is a collection of messages (message category / message number) that may be added while processing the transaction. For example: 'Tax Clearance Certificate approved', and 'Mailed to the taxpayer on 01-02-2012'

Task Confirmation Inquiry

A taxpayer may inquire on the status of a transaction or request using a confirmation number. If so, a web service is sent to the product. The product provides the XAI Inbound Service **TSGetConfirmationInformation** to process this web service. It uses the script **C1-GetCnfmID** to process the request. This service script finds the service task associated with the confirmation, retrieves the confirmation details stored with the task and returns the information to the calling system.

Confirmation Email

The product provides an algorithm to allow a tax authority to send an email to the taxpayer once a task that is performing maintenance actions has completed. For example, when a taxpayer submits a payment, the service task business object that processes the payment can be configured to send an email to the taxpayer once the payment is successfully created. The base algorithm includes the confirmation details from the service task in the email along with general email text that is configurable.

> **Fastpath:** Refer to the algorithm section of *General Configuration Tasks* for more information about this algorithm.

Error Handling

All web services includes standard error message elements. If any errors are found in processing a web service, a response is returned with the error message.

> **Note:** The system error messages may not be appropriate to display to the web self service user. The message can be translated to one that is more appropriate in the BPEL layer.

Configuration Errors

If the processing script used to process any of the messages finds a configuration problem or unrecognized information from the message, it returns a generic error to the self service application like "Your request can not be processed at this time, please try again later". In addition, a To Do entry and / or an email is sent to an appropriate responsible user.

> **Fastpath:** Refer to the algorithm section of *General Configuration Tasks* for more information about this algorithm.

Extending the Base Logic

All web services include a "custom information" area that allows an implementation to configure up to 10 additional fields using field name and value pairs that may be populated using a custom service script.

> **Fastpath:** Refer to the master configuration section of *General Configuration Tasks* for more information.

Standard Message Format

All web services used to communicate with the self service application follow a common structure that includes the following information. For web services that require a record to be instantiated in the system, a service task is created. The base product business objects for service task capture this information.

- Common data used to track the web user access details: User ID, Name, IP Address and email address. Note that in the base business objects this information is captured in the business object data area as well as in characteristics to allow searching by these values.
- Access keys. The web service supports capturing up to 5 key fields defined with field name and value pairs.
- A custom information area for implementation specific values.
- For the requests that perform transaction processing, there is also a standard data area that defines Confirmation Details.

Viewing Raw XML

The 'raw XML' of the web service is captured for audit purposes on any resulting service task. It's possible that a user may need to review the raw XML to troubleshoot problems. Because the data may contain personal and sensitive information from the taxpayer, such as a person identifier, the product provides additional security configuration for viewing the raw XML on service tasks.

> **Note:** Refer to the business object descriptions for the base parent business objects **C1-StandardDeferredSSTask** and **C1-StandardRealTimeSSTask** for more information about securing this logic.

Settings in Master Configuration

Many implementation wide settings are needed when configuring the product to work with the web self service application. The product captures these settings in the Self Service Master Configuration record.

In the sections in this document that explain how to configure the system to use the base self service functionality, any settings in the self service master configuration record are noted.

Inquiry Audit

The product supports the ability for implementations to capture audit records of what data users are viewing. The integration with a self service application extends this capability so that audit records may be captured when the system receives requests for information from the self service application.

A special inquiry audit business object (**C1-WebAuditInquiry**) has been provided to capture audits of information requests from the self service product. The business object captures the web user id, the web user name, the email address, the IP address and, if configured, a snapshot of the actual request XML including any response.

Note: The user interface for this type of inquiry audit allows a user to view the request / response XML for those inquiry audit records that capture it. Because the data may contain personal and sensitive information from the taxpayer, such as a person identifier, the product provides additional security configuration for viewing the raw XML on inquiry audits. Refer to the business object description for the base business objects **C1-WebAuditInquiry** for more information about securing this logic.

Configuring Inquiry Audit

An implementation configures which web services should cause an inquiry audit record to be created. For example, perhaps web services for a refund status inquiry are audited, but web services for inquiring on the status of a task via the confirmation ID are not. The self service master configuration record allows an implementation to configure the web services that should be audited by indicating an appropriate audit service script to invoke when the web service is processed.

Fastpath: Refer to the master configuration section of *General Configuration Tasks* for more information.

General Configuration Tasks

This section describes the general tasks required to configure the system to support the integration. It also describes the service task type portal.

Refer to the Supported Functionality section for addition details about configuration required for specific business functionality.

Messages

Review the base messages provided for communication of positive and negative responses to the self service user. These are configured on self service task types.

- Confirmation Header message. The product provides a base message that may be used, with message category 11126 and message number 11723. The base message text may be overwritten. Or a new message in a message category reserved for implementations may be created.
- Error Header message. The product provides a base message that may be used, with message category 11126 and message number 11010. The base message text may be overwritten. Or a new message in a message category reserved for implementations may be created.

Note: In the BPEL layer the product's Message Category / Message Number combination is translated into message codes understood by the self service product.

Managed Content

Create a managed content record to define the HTML to use for the various general emails.

- Create one for emails routed to a responsible user when problems are found with the master configuration table.
- If confirmation emails should be sent to taxpayers when tasks with multiple steps are complete, create a managed content entry for that email text. If different service tasks should have different general text in the confirmation email, create a separate record for each variation.

The Managed Content Type should be HTML. Navigate to the Schema tab and enter the following tags and type the appropriate text for the email within the "" tags.

```
<html>
  <body>
    <span> </span>
```

```
  </body>
</html>
```

XAI Sender

Verify that an XAI sender is configured for routing email real time.

- Invocation Type should be set to **Real-time**
- XAI Class should be set to **RTEMAILSNDR**
- On the Context tab, the context type **SMTP Host name** should be defined with a valid context value.

> **Note:** The XAI Options includes an option for a default Sender for real-time emails that can be configured with this XAI sender.

Field

Define a field that includes the text for the email subject for the notification of master configuration errors email.

Define one or more fields that includes the text for the email subject for the confirmation emails to send for the various service tasks.

Algorithm

If any service task should send a confirmation email after all steps are complete, for example once a payment is processed, configure one or more appropriate algorithms for the **C1-SSCONFREM** algorithm type.

- Set the Email Subject field created above for the confirmation email.
- Set the Predefined Text to the Managed Content created above for the confirmation email.
- Enter the From Address that should be used in the generated email
- Enter the XAI Sender for Email defined above.

> **Note:** This algorithm must be plugged into appropriate business objects as an enter algorithm on a state where actions done by the task are complete.

Self Service Master Configuration

Create a Self Service Master Configuration record that holds system wide configuration needed for the self service integration.

The Configuration Support section defines information needed to alert a user that problems were found with configuration. The alert may be sent using a To Do entry or an email or both.

- Define the user that is responsible for technical issues related to self service who should receive an email if there is an issue.
- To Do Type should be set to a standard error To Do type to use when issues are found with the master configuration table. The base product To Do Type **Self Service Master Configuration Issues (C1-SSMCI)** is provided for this purpose.
- To Do Role should be configured to an appropriate default To Do role for the above To Do type.

Populate the following fields if an email should be generated when configuration problems exist.

- Enter the From Email Address that should be used in the email.
- Enter the From Name that should be used in the email.
- Enter the XAI Sender created above or leave it blank to use the default from XAI options.

- Set the Email Content to the Managed Content record created above.
- Set the Email Subject field created above.
- Provide the Configuration Support Email Script that produces the email. The product has supplied **C1-SSIssueEm** which may be used here.

The Request Processing Control information allows an implementation to define special logic that should occur when a web service is processed. The special logic may be causing an inquiry audit record to be created or it could be to indicate a custom extension to a base product service by retrieving additional details to return in a web service call.

- To Do Type for Audit Script Errors should be set to an appropriate To Do type. The base product To Do Type **Self Service Task Audit Issues (C1-SSAI)** is provided for this purpose.
- To Do Role for Audit Script Errors should be configured to an appropriate default To Do role for the above To Do type.
- To Do Type for Custom Extension Errors should be set to an appropriate To Do type. The base product To Do Type **Self Service Task Custom Extension Issues (C1-SSCSI)** is provided for this purpose.
- To Do Role for Custom Extension Errors should be configured to an appropriate default To Do role for the above To Do type.

For any web service that requires inquiry audit or custom extension, determine the schema associated with the XAI inbound service that processes the web service.

- Schema Type and Schema Name should be set to the appropriate object that the XAI inbound service references.
- For enabling inquiry audit, check the Inquiry Audit Enabled button. If the raw XML of the request and response should be captured, check the Store Response Data checkbox. Indicate the Audit Script. The base product provides the script **C1-AddWebAI** that may be used.
- For providing additional data in a web service response, provide a Custom Extension Script. The schema for the custom extension script should match the schema that it is extending. It populates data in the data area **C1-SSCustomExtensionFields**.

Setting Up Service Task Types

Service Task Types contain the rules that control how service tasks are processed.

To set up a service task type, select **Admin Menu** > **Self Service Task Type** .

The topics in this section describe the base-package zones that appear on the Service Task Type portal.

Self Service Task List

The Service Task Type List Zone lists every self service task type. The following functions are available:

- Click the broadcast button to open other zones that contain more information about the adjacent service task type.
- The standard actions of **Edit, Delete** and **Duplicate** are available for each service task type.
- Click the **Add** link in the zone's title bar to add a new service task type.

Self Service Task Type

The Self Service Task Type zone contains display-only information about a service task type. This zone appears when a service task type has been broadcast from the Self Service Task Type List zone or if this portal is opened via a drill down from another page.

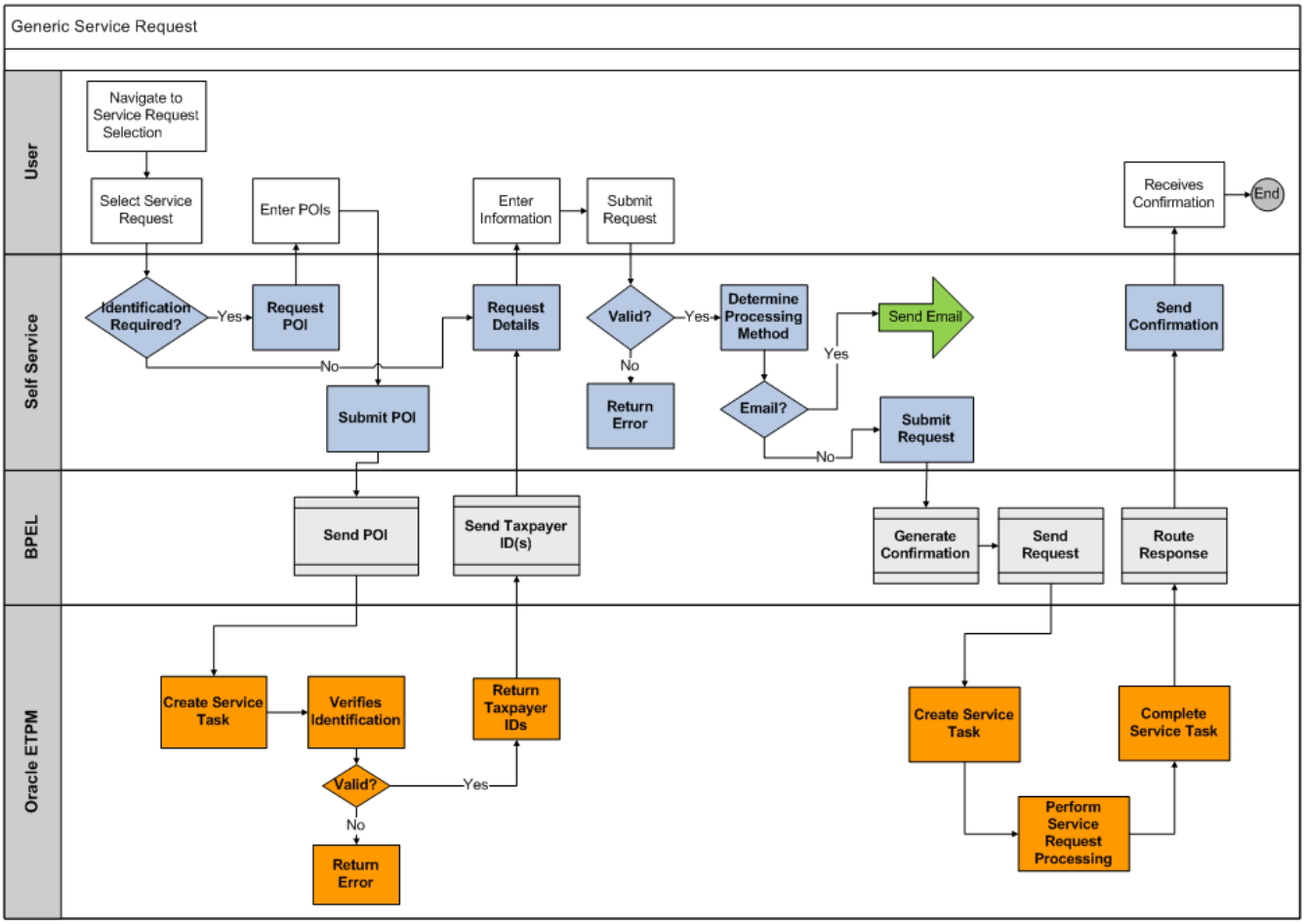
Please see the zone's help text for information about this zone's fields.

Supported Functionality

This section describes the out of the box functionality provided for the integration along with the required configuration tasks.

Taxpayer Service Request

The self service product supports service requests by the taxpayer. Some requests may be fulfilled by the self service product, such as information about when to file or technical support requests. Other requests may require information from this product. The following diagram illustrates an expected process flow.



The following points highlight the process

- The service request may or may not require up front taxpayer identification. If it does, the product receives a web service request to identify the taxpayer. The product provides the XAI Inbound Service **TSTaxpayerIdentification** to process the message. It uses the script **C1-IdentTaxp** to process the request. This should cause an appropriate service task to be created based on the type and populate the relevant data for the service task. The service task should include algorithms to identify the taxpayer and immediately respond to the web service call.

  Note:
  Refer to *Configuration Tasks for Taxpayer Service Request* for more information.

- For any service request that requires action by the product, another web service request is received with information about the request, including the task type to create. The product provides the XAI Inbound Service **TSTaxpayerServiceRequest** which uses the script **C1-TaxSvcReq** to process the request. This should cause an appropriate service task to be created based on the type and populate the relevant data for the service task. Depending on the requirements for a particular service request, the service task may be designed to do the following:
  - Give real-time feedback / results (if the request simply involves information retrieval)
  - Perform the task as a separate step without the taxpayer waiting for a reply. In this case the taxpayer should receive a confirmation ID related to the request that is provided in the web service so that at a later date the progress of the service request can be queried by the taxpayer.

    Note: Refer to *Implementing A New Service Request* for more information.

The base product provides functionality out of the box for a taxpayer's request for a tax clearance certificate. The topics below describe more information about the provided functionality, followed by configuration tasks.

Configuration Tasks for Taxpayer Identification Service Request

The following sections list the configuration tasks required to implement the base taxpayer identification functionality.

Characteristic Type

If your implementation captures the taxpayer's birth date on the person record and would like that to be part of the taxpayer identification for service requests, create a characteristic type for Birth Date. It should be defined as ad-hoc and should reference Person as the characteristic entity.

Note: The product does not provide any functionality for populating the birth date characteristic on the person.

Algorithms

Create the following algorithms:

- Create an algorithm for algorithm type **C1-IDTXPSR**. Populate the Birth Date Required parameter as per business rules. Define the characteristic type for the birth date that was created above.

Business Object

Update the business object **C1-TaxpayerIdentSvcReqSSTask** to include this algorithm. Navigate to the Lifecycle tab and on the **In Progress** state, add an entry to the Algorithm collection: System Event is **Enter**, Algorithm is the one created above.

Service Task Type

Create a service task type for taxpayer identification to use for service requests. Use the business object **Taxpayer Service Request Self Service Task Type**.

- The related transaction BO should be set to **C1-TaxpayerIdentSvcReqSSTask**.
- Service task class should be set to **Service Request**.
- To Do Type should be set to a standard error To Do type to use when transitioning to error. The base product To Do Type **Self Service Task Issues (C1-SSTTD)** is provided for this purpose.
- To Do Role should be configured to an appropriate default To Do role for the above To Do type. This is optional. If no value is provided the default role for the To Do type is used.
- Configure the Confirmation Header Message Category and Message Number and the Error Header Message Category and Message Number to use for communication to the self service user.

Fastpath: Refer to the Messages section of the *General Configuration Tasks* for more information.

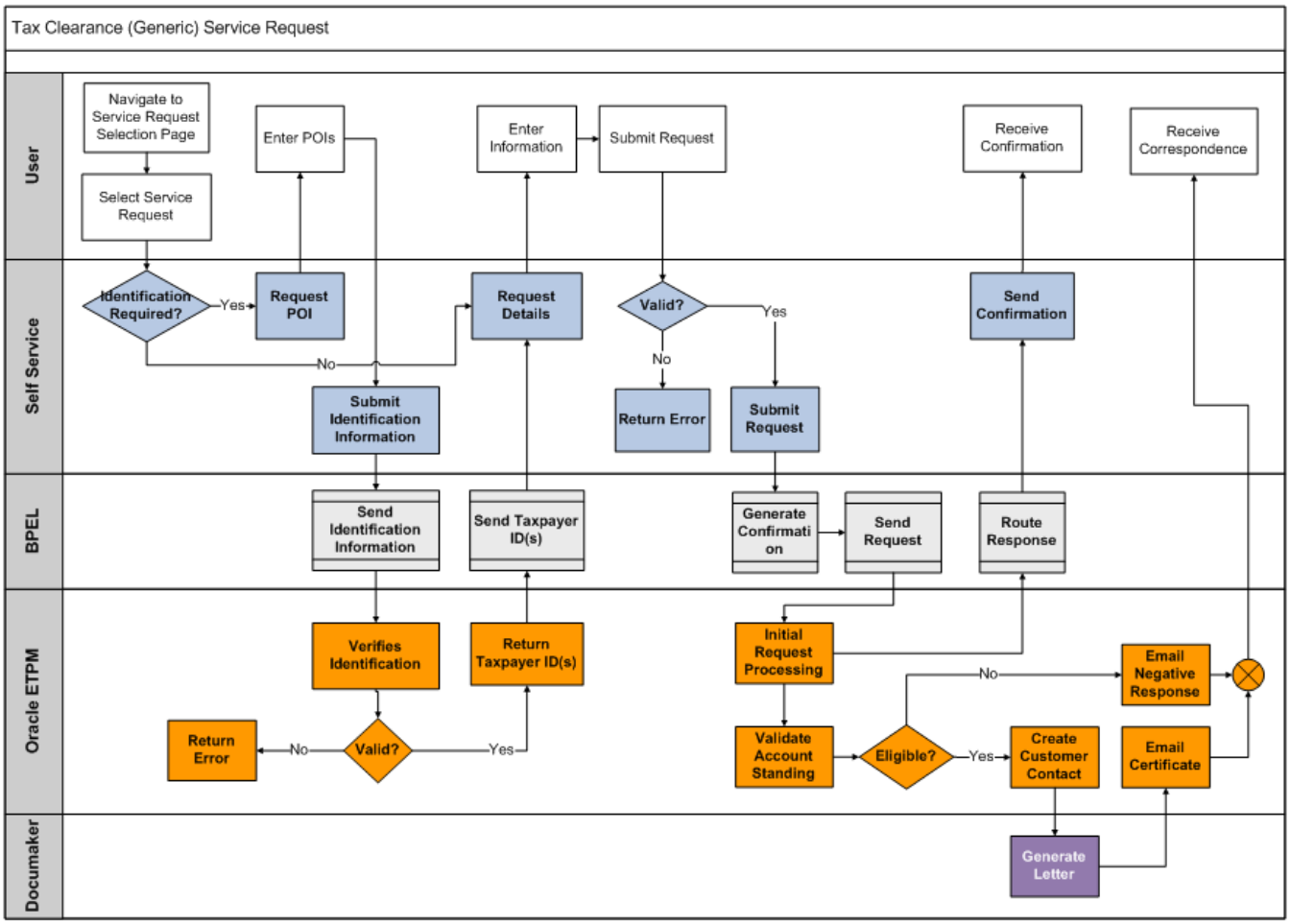
- The service request fields mapping is used when creating the target service task to correctly populate the requestField list from the list of fields passed in on the web service request XML. For the base transaction business object, the following fields are expected:

Sequence	Transaction BO XPath
1	requestData/legalName
2	requestData/birthDate
3	requestData/personType
4	requestData/idType
5	requestData/personIdNumber

Fastpath: Service Task Type Mapping. The service task type defined here must be configured in the service task domain value mapping in the SOA Composer application. Refer to the self service product documentation for details.

Tax Clearance Certificate

On occasion a taxpayer may be required to produce a Tax Clearance Certificate to a third party. This must be requested from the tax authority as a written confirmation that a taxpayer is considered in good standing as of the date of issue of the Certificate. The following diagram illustrates the process flow supported for issuing a tax clearance certificate.



The following points describe in more detail the functionality provided

- This type of service request requires taxpayer identification first.
- The product provides a base service task business object called **C1-TaxClearCertSvcReqSSTask** that processes the request. It calls an algorithm to determine if the account is in good standing. If so it creates a customer contact to issue the certificate, that may be emailed. If not, it creates a customer contact to inform the taxpayer that a certificate is not possible.
- The product provides a letter extract for a sample tax clearance certificate. In addition a letter template for Documaker is included in this functionality.

Configuration Tasks for Tax Clearance Certificate

The following sections list the configuration tasks needed to implement the tax clearance certificate functionality.

Lookup

Using the lookup **C1_TAX_CLEARANCE_PURPOSE_FLG** define the values that are valid for your implementation.

Characteristic Types

Create the following characteristic types:

- Tax Clearance Purpose

. This char type is used to capture the tax clearance purpose on the customer contact for audit purposes. Define Customer Contact as a characteristic entity. The tax clearance purpose is defined in a lookup field (**C1_TAX_CLEARANCE_PURPOSE_FLG**). The characteristic type should be defined as a predefined characteristic with the valid values from the lookup repeated.

Letter Template

Create a letter template to represent the tax clearance certificate. Reference the extract algorithm **C1-LTR-TCSDC**. Batch control is required and **LTRPRT** can be entered here.

Customer Contact Class / Type

Define or identify an appropriate Customer Contact Class for the tax clearance certificate customer contact types.

Create a Customer Contact Type for issuing the tax clearance certificate.

- Customer Contact Business Object is not required.
- Contact Shorthand is not applicable.
- Set the Contact Action to **Send Letter** and indicate the letter template created above.
- For the template characteristics, define the Characteristic type created above for Tax Clearance Purpose. In addition, define the (base) characteristic type Expiration Date.

Create a Customer Contact Type for declining the Tax Clearance Certificate.

- Customer Contact Business Object is not required.
- Contact Shorthand is not applicable.
- Leave Contact Action blank.
- For the template characteristics, define the Characteristic type created above for Tax Clearance Purpose.

Managed Content

Create managed content records to define the HTML to use for the email related to the tax clearance certificate. Define one for the email that is sent when issuing the certificate. Define another for the email that is sent when the request is denied.

For both records, the Managed Content Type should be HTML. Navigate to the Schema tab and enter the following tags and type the appropriate text for the email within the "" tags.

```
<html>
  <body>
    <span> </span>
  </body>
</html>
```

Field

Define a field that includes the text for the email subject.

Algorithms

Create the following algorithms:

- Evaluate Outstanding Debt. Create an algorithm for the algorithm type **C1-EVALDEBT**. Populate the Max Outstanding Debt Allowed parameter.
- Evaluate Gap in Filing. Create an algorithm for the algorithm type **C1-EVALGAP**. Populate the Overdue Process Template and the No-gap Filing Period parameters.

- Email Tax Clearance Certificate Letter. Create an algorithm for the algorithm type **C1-EMTXCL**. Populate the parameters:
  - Set the Customer Contact Class created above.
  - Set the Customer Contact Types for Issue Tax Clearance Certificate and for Decline Tax Clearance Certificate created above.
  - Set the Characteristic Type for Tax Clearance Certificate Purpose
  - Set the Predefined Email Text for the Issuing Certificate Email and the Decline Request Email configured as Managed Content above.
  - Set the Email Subject field created above.
  - Enter the Characteristic Type for Certificate Expiration Date: C1-EXPDT
  - Enter the number of days that is the Certificate Valid Period
  - Enter the XAI Sender for Email created as part of the general configuration options.
  - Enter the From Email Address that should be used in the generated email
  - Enter the From Name that should be used in the generated email

Business Object

Update the business object **C1-TaxClearCertSvcReqSSTask**. Navigate to the Lifecycle tab.

- On the **In Progress** state, add an entry to the Algorithm collection: System Event is **Enter**, Algorithm is set to the one created above for Evaluate Outstanding Debt.
- On the **In Progress** state, add an entry to the Algorithm collection: System Event is **Enter**, Algorithm is set to the one created above for Evaluate Gap in Filing.
- On the **Complete** state, add an entry to the Algorithm collection: System Event is **Enter**, Algorithm is set to the one created above to Email the Tax Clearing Certificate Letter.

Service Task Type

Create a service task type for the tax clearance certificate service task with the business object **Taxpayer Service Request Self Service Task Type**.

- The related transaction BO should be set to **C1-TaxClearCertSvcReqSSTask**.
- Service task class should be set to **Service Request**.
- To Do Type should be set to a standard error To Do type to use when transitioning to error. The base product To Do Type **Self Service Task Issues (C1-SSTTD)** is provided for this purpose.
- To Do Role should be configured to an appropriate default To Do role for the above To Do type. This is optional. If no value is provided the default role for the To Do type is used.
- Configure the Confirmation Header Message Category and Message Number and the Error Header Message Category and Message Number to use for communication to the self service user.

  Fastpath: Refer to the Messages section of the *General Configuration Tasks* for more information.

- The service request fields mapping is used when creating the target service task to correctly populate the requestField list from the list of fields passed in on the web service request XML. For the base transaction business object, the following fields are expected:

Sequence	Transaction BO XPath
1	requestData/emailAddress
2	requestData/phoneNumber

3	requestData/taxClearancePurpose
4	requestData/isPaperCopyReq

Fastpath: Service Task Type Mapping. The service task type defined here must be configured in the service task domain value mapping in the SOA Composer application. Refer to the self service product documentation for details.

Refund Status

In this release, the integration supports the ability for a taxpayer to inquire on the status of a tax refund. This is a special kind of taxpayer service request that defines its own web service. The integration has the following steps:

- The self service application captures information about the taxpayer, a way of identifying the filing period for the refund in question and a "shared secret" that ideally only the taxpayer and the tax authority would know (such as the expected refund amount).
- This information is sent to the system, which creates a Refund Status Inquiry service task.
- The task's lifecycle is designed to process the information immediately with no deferred monitor process so that the information can be returned to the self service application real-time.
- The base business object for the Refund Status Inquiry service task includes an algorithm to verify the taxpayer identification, confirm that a form has been received for the filing period, determine the status of the refund and compose an appropriate message to send back to the taxpayer immediately.

Note that the implementation of the refund status inquiry creates a service task for the following reasons:

- The creation of a service task with its business object architecture provides many plug-in spots to allow implementations to easily customize the logic for identifying the taxpayer and determining the refund status.
- The status of a refund is based on conditions in the system that can change over time. Instantiating a service task for the refund status inquiry provides a way of capturing the point in time that the request was received and being able to reconcile this with the information sent to the taxpayer.
- Creating a service task enables the tax authority to provide the taxpayer with a confirmation ID for possible follow up in the future. For example, imagine a taxpayer is told "Your refund has been processed. You should receive it shortly." and several weeks later the taxpayer has still not received the refund. If the taxpayer contacts the tax authority and provides the confirmation ID, the user can view the service task, verify the details and more easily investigate the situation.

The topics below describe more information about the provided functionality, followed by configuration tasks.

Refund Inquiry Service Task

The web service requesting the refund status is processed by the XAI Inbound Service **TSGetRefundStatus**, which invokes the service script **C1-RSISvcReq**. The appropriate service task type to use must be passed in with the other information received from the self service system.

Identifying the Shared Secret Amount

The logic provided in the base product assumes that the shared secret is an amount on the taxpayer's form. Form definition in the system is configurable and each form may be defined with different form lines that capture the amount used for the shared secret. For each form type, the system needs to know which form line is the one used for the shared secret. This information is defined in the self service master configuration.

Fastpath: Refer to *Configuration Tasks for Refund Status* for information about configuring the system to support these payment destinations.

Determining the Status of the Refund

The lifecycle of creating a tax refund in the system includes the following steps.

- The taxpayer files the form, which is received by the system. The form gets processed and appropriate adjustments are created for the obligation for the filing period to record the affect on the balance.
- Once the taxpayer's obligation has a credit balance, an overpayment process is used to process the refund. How the overpayment process is created depends on the business practice. An account debt monitor may be used to detect obligations with a credit and can create the overpayment process. The form can include logic to immediately create the overpayment process based on the existence of a credit balance based on a form rule.

  Note: The base product does not currently supply a form rule to create the overpayment process at posting time. A custom form rule may be developed.

- The overpayment process may require approval. Once approvals are processed, the overpayment process may have logic to use the credit to apply to other debt. Or it may carry the credit forward to a future filing period (typically upon the taxpayer's request). When the overpayment process completes, it has a record of carry forward, offset and refund amounts. If there is any refund, the overpayment creates an accounts payable (A/P) adjustment if the refund should be provided using a paper check. It creates a credit payment with appropriate bank information to process the refund as a direct deposit.

An algorithm plugged into the refund status inquiry service task business object as an enter plug-in on the **In Progress** state is responsible for determining the status of the refund. Refer to the description of the algorithm for more information about the base logic provided.

Configuration Tasks for Refund Status

The following sections list the configuration tasks needed to implement the refund status functionality.

Service Task Type

Create a service task type for the refund status inquiry service task with the business object **Standard Self Service Task Type**.

- The related transaction BO should be set to **C1-RefundStatusInquirySSTask**.
- To Do Type should be set to a standard error To Do type to use when transitioning to error. The base product To Do Type **Self Service Task Issues (C1-SSTTD)** is provided for this purpose.
- To Do Role should be configured to an appropriate default To Do role for the above To Do type. This is optional. If no value is provided the default role for the To Do type is used.
- Configure the Confirmation Header Message Category and Message Number and the Error Header Message Category and Message Number to use for communication to the self service user.

  Fastpath: Refer to the Messages section of the *General Configuration Tasks* for more information.

Fastpath: Service Task Type Mapping. The service task type defined here must be configured in the service task domain value mapping in the SOA Composer application. Refer to the self service product documentation for details.

Master Configuration

The self service master configuration record includes several settings required for refund status functionality.

- Shared Secret Form Line. For each form type that is supported for a taxpayer to inquire about a refund, indicate the XPath of the form line reference. Note that there is an ability to view the schema of the form business object associated with the form type to find the correct XPath.

  Note: Reported vs. Current. Because the configuration is related to information that the taxpayer has reported, consider whether the "current" element (asCurrent) or "reported" element (asReported) is used. If the taxpayer has made calculation mistakes, form rules or users may adjust the current value of the shared secret field. In this case the reported value is what the taxpayer knows and would enter.

- Overpayment Process Amounts. In order to be able to provide details related to the refund, it is necessary to report amounts that may have been used to offset other debt and amounts that are carried forward to future filing periods. The information is defined as elements in overpayment process business object. For each Overpayment Process Type, indicate the XPath values for the Refund Amount, the Offset Amount and the Carry Forward Amount. Note that there is an ability to view the schema of the overpayment process business object associated with the overpayment process type to find the correct XPath.

Implementing A New Service Request

Your implementation may wish to provide support for additional service requests for your taxpayers. The following information outlines the steps required in the product to support a new service request.

Taxpayer Identification

First decide if the taxpayer must provide identification information as part of the service request. If so,

- Should the system validate the taxpayer before continuing with the actual request? In this case an initial web service call must be sent to the system with appropriate information to validate the taxpayer.
- Can the identification information be provided with the other service request details so that the taxpayer identification can be done with the service request processing?

If the taxpayer identification is done as a separate step, refer to *Configuration Tasks for Taxpayer Service Request* for the base taxpayer identification business object provided with the product. That may be used or a new one based on appropriate business rules may be introduced with this one as a sample.

Business Object

Each service request must have its own Service Task business object. This business object defines the information required for this type of request and defines the algorithms that validate and process the request. The product provides two business objects that may be used as a parent business object for a new service request.

- The Standard Deferred Self Service Task business object (**C1-StandardDeferredSSTask**) supports the ability to defer the full processing of the request to a subsequent step. Defining a service task business object as a child of this business object is recommended if any of the following are true:
  - A user needs to review the taxpayer's request.
  - The request is high volume such that processing the request real-time for many taxpayers may cause a lot of system traffic.
  - The logic required to process the request is such that it doesn't make sense for the taxpayer to wait until the processing is complete.
- The Standard Real Time Self Service Task business object (**C1-StandardRealTimeSSTask**) supports immediate processing of the service request to give a response to the taxpayer real time.

Create the new business object defining the appropriate parent business object and navigate to the schema tab. Include the parent BO's schema. Define the additional elements required for this business object's functionality. Refer to the Tax Clearance Certificate business object for an example.

The business object must also include the appropriate algorithms to satisfy the request.

Service Task Type

Create a service task type for the new service request with the business object **Taxpayer Service Request Self Service Task Type**.

- The related transaction BO should be set to the business object created above.
- Service task class should be set to **Service Request**.
- To Do Type should be set to a standard error To Do type to use when transitioning to error. The base product To Do Type **Self Service Task Issues (C1-SSTTD)** is provided for this purpose.
- To Do Role should be configured to an appropriate default To Do role for the above To Do type. This is optional. If no value is provided the default role for the To Do type is used.
- Configure the Confirmation Header Message Category and Message Number and the Error Header Message Category and Message Number to use for communication to the self service user.

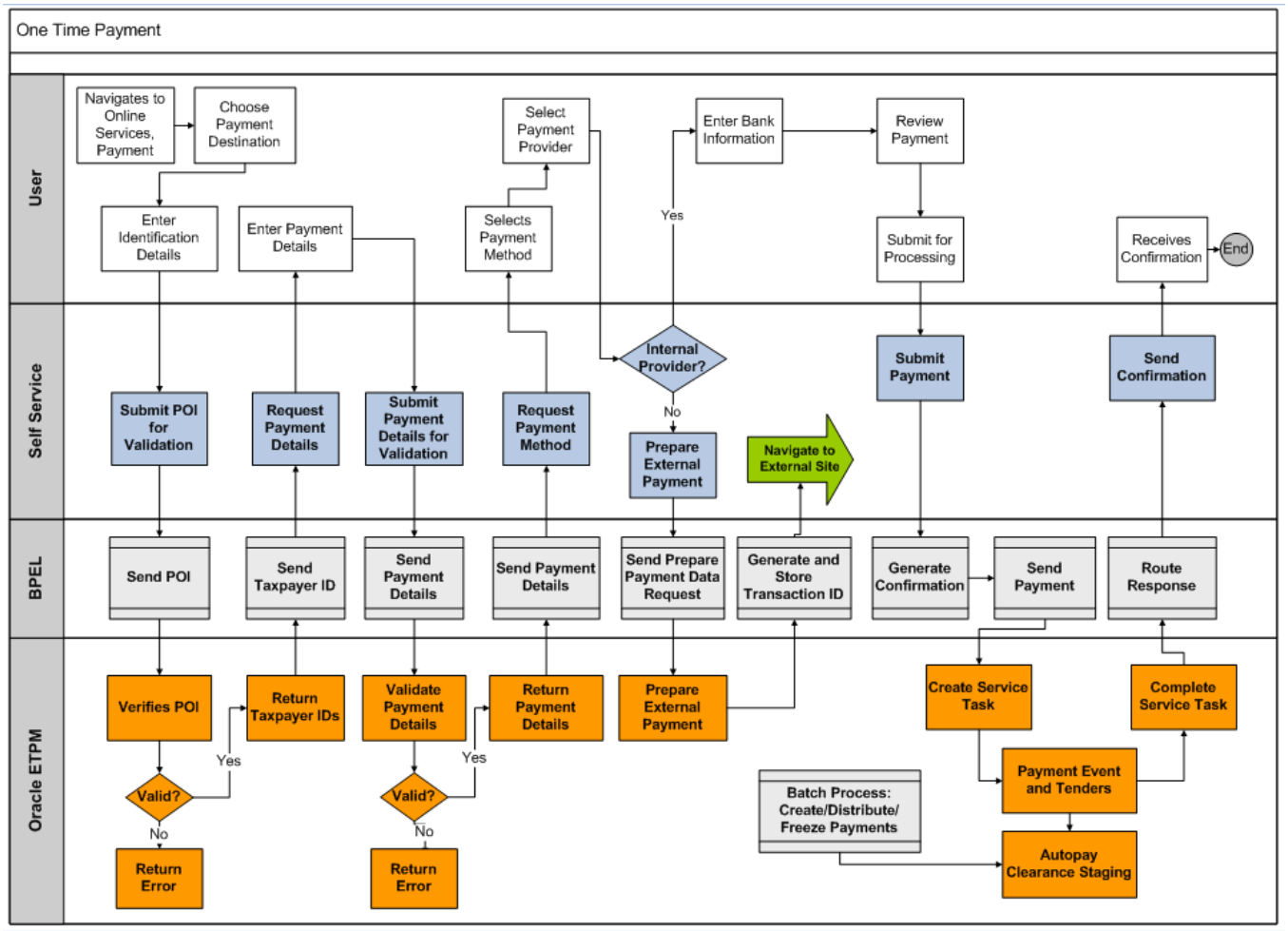
  Fastpath: Refer to the Messages section of the *General Configuration Tasks* for more information.

- The service request fields mapping is used when creating the target service task to correctly populate the request data list from the list of fields passed in on the web service. Define the mappings appropriate for this service request as needed.

  Fastpath: Additional Configuration is required in the self service product and in the integration layer for new service requests. Refer to the self service product documentation for more information.

Online Payment

In this release, the integration supports payment by a taxpayer using a bank or credit card number. The following diagram illustrates an expected process flow.



The integration has the following steps:

- The self service application captures information about the taxpayer and sends a web service request to this product to validate the taxpayer (proof of identity - POI).
- Once identified, the taxpayer indicates what the payment is for. This is referred to as the target of the payment or the payment destination or the payment details. For example, is this a payment for a pay plan scheduled payment? If so, the taxpayer must provide the pay plan identifier. Once the taxpayer indicates the payment target details, another web service request is sent to this product to verify the details.
- Next the taxpayer indicates the method of payment.
  - If the method is via a credit card and the implementation uses an external payment vendor, the self service application sends a web service request to this product to retrieve any additional information about the taxpayer that may need to be sent to the external vendor (such as the taxpayer's address) and to determine if the external vendor's fee should be passed on to the taxpayer. At that point the taxpayer is taken to the external vendor's website to further process the payment details. Once that step is complete a final web service request is sent to this product to process the payment.
  - If the method is via a bank account or if the implementation does not use an external payment vendor for credit card payments, the self service application sends a web service to this product to create the payment. The standard Auto Pay logic in the product is used to process the payment.
- If the external vendor supports payment reconciliation, the subsequent reconciliation report is processed by the product.

The topics below describe more information about the provided functionality, followed by configuration tasks.

Identifying the Taxpayer

In this release the integration supports payments by an 'unregistered' taxpayer. The taxpayer does not need to log in using a user name and password in order to submit a payment. In addition, the product supports a person paying on behalf of another taxpayer.

The taxpayer must provide enough information to be identified by the system prior to proceeding. A special service task business object is provided for identifying a taxpayer for one-time payments (**C1-TaxpayerIdentifySSTask**).

> **Fastpath:**
> Refer to *Configuration Tasks for Online Payment* for more information.

Payment Destination

When creating a payment, the product supports using distribution rules to determine where the payment should be targeted. The integration provides out of the box support for the following payment destinations.

- Pay a filing period. This method allows the taxpayer to provide their taxpayer ID, a tax type and a filing period to direct the payment to.
- Pay a pay plan. This method allows the taxpayer to provide a reference to a pay plan to pay one or more of the scheduled payments.
- Pay a collection notice. This method assumes that a collection notice has been issued from a customer contact in the system and that the customer contact ID has been provided to the taxpayer as an identifier.

The business object provided by the product for processing one time payments has been designed to support different payment destinations. To do this, the business object must support receiving different types of information (for example, pay plan ID or collection notice ID) and use plug-ins that are specific to each payment destination that knows the type of data expected, how to use that data to identify the correct account and obligation(s) and process the payment successfully. At a high level it works as follows:

- The web service requests sent by the self service application to validate the payment target and to process the payment include an indication of the payment destination and a field/value pair collection to hold the values that represent the 'target' for the payment.
- The self service master configuration record includes a mapping between the payment destination and a self service task type.
- The service task type for each unique payment destination references a "processor" service, which can be a service script or a business service. This service should be specific for a given payment destination and expects the data appropriate for it. For example, the process to "pay a pay plan" expects the service task's destination details to specify a pay plan ID.
- The service task type includes a field mapping section. This mapping indicates which field value in the destination details list to populate in which target XPath in the "processor" service.
- The web service called to validate the destination details provided by the taxpayer invokes the "processor" to ensure that the data can be found.
- When processing the payment, the service task that processes payments includes an algorithm that uses the configuration on the task type to call the "processor" service, passing the details of the payment target.

> **Fastpath:** Refer to *Configuration Tasks for Online Payment* for information about configuring the system to support these payment destinations. Refer to *Implementing New Payment Destinations* for information about defining additional payment destinations for your implementation.

Validating the Payment Destination

The web service used to validate the payment destination information entered by the taxpayer is processed by the XAI Inbound Service **TSPrepareExtPaymentData**, which invokes the service script **C1-PrExPyDta** with an action of **VALIDATEONLY**. The service script determines the service task type based on the payment destination (via master configuration). It then calls the payment destination details processor with an action of "validate only". The processor should return an error if the information provided by the taxpayer does not identify an appropriate related object for the payment destination.

Creating the Appropriate Service Task

The web service request to create a payment is processed by the XAI Inbound Service **TSOneTimePayment**, which invokes the service script **C1-PyUnrgUsr**. The service script is responsible for creating a Service Task of the correct service task type based on the payment destination.

Processing Auto Pay Payments In the System

When a taxpayer provides bank account information for payment in self service, the payment detail is sent to this system and the integration with the customer's bank is handled through standard auto pay processing. This method may also be used to process credit cards. However, it's expected that an integration with an external vendor will most often be used for that. Refer to the section below for more information on external payment vendors.

Requiring an Auto Payment Agreement

Many tax authorities require taxpayers to sign an automatic payment agreement prior to allowing them to submit tax payments online. The product supports checking for a reference to an ACH agreement that is defined as a characteristic linked to the taxpayer's record.

Payment Processing Lifecycle

The one-time payment service task provided by the base product includes the following basic lifecycle states:

- **Pending**. When the web service request is processed, a service task is created in pending status. Initial validation is performed, including verifying that the information provided for the payment destination and determining the appropriate account to associate with the payment. If any issues are found, the service task is not stored and an error is returned. Once the record is validated, no further processing occurs at this time. When the service task monitor is run all pending records are further processed.
- **In Progress**. When transitioning to this status, the payment is processed. The data related to the payment destination is mapped to appropriate payment event distribution details and the payment is created. If any errors are encountered, the service task transitions to **Error**. If the payment was processed through an external vendor that requires reconciliation, it transitions to **Pending Reconciliation**. Otherwise, it transitions to **Complete**.
- **Error**. Any errors detected in payment processing or during reconciliation processing causes the service task to transition to this state and creates a To Do entry. A user must review and resolve the problem.
- **Error Resolved**. When a user resolves the error reported the service task can be transitioned to this state. Note that the assumption is that the user has manually corrected the error. This state does not cause any logic to occur.
- **Pending Reconciliation**. When a payment is made through an external vendor that requires reconciliation, the service task transitions to this state and waits for the reconciliation process. Refer to the external payment vendor section below for more information about reconciliation. Note that this state is configured with time out logic. The payment vendor can be configured with a number of days to wait for the reconciliation report. If that number of days pass, a To Do entry is created using an appropriate To Do type defined for the payment vendor.
- **Complete**. Service tasks transition to this state when the payment is fully processed.

External Payment Vendor

Many websites integrate with an external payment vender to process credit cards real-time. In this case, the following functionality is provided:

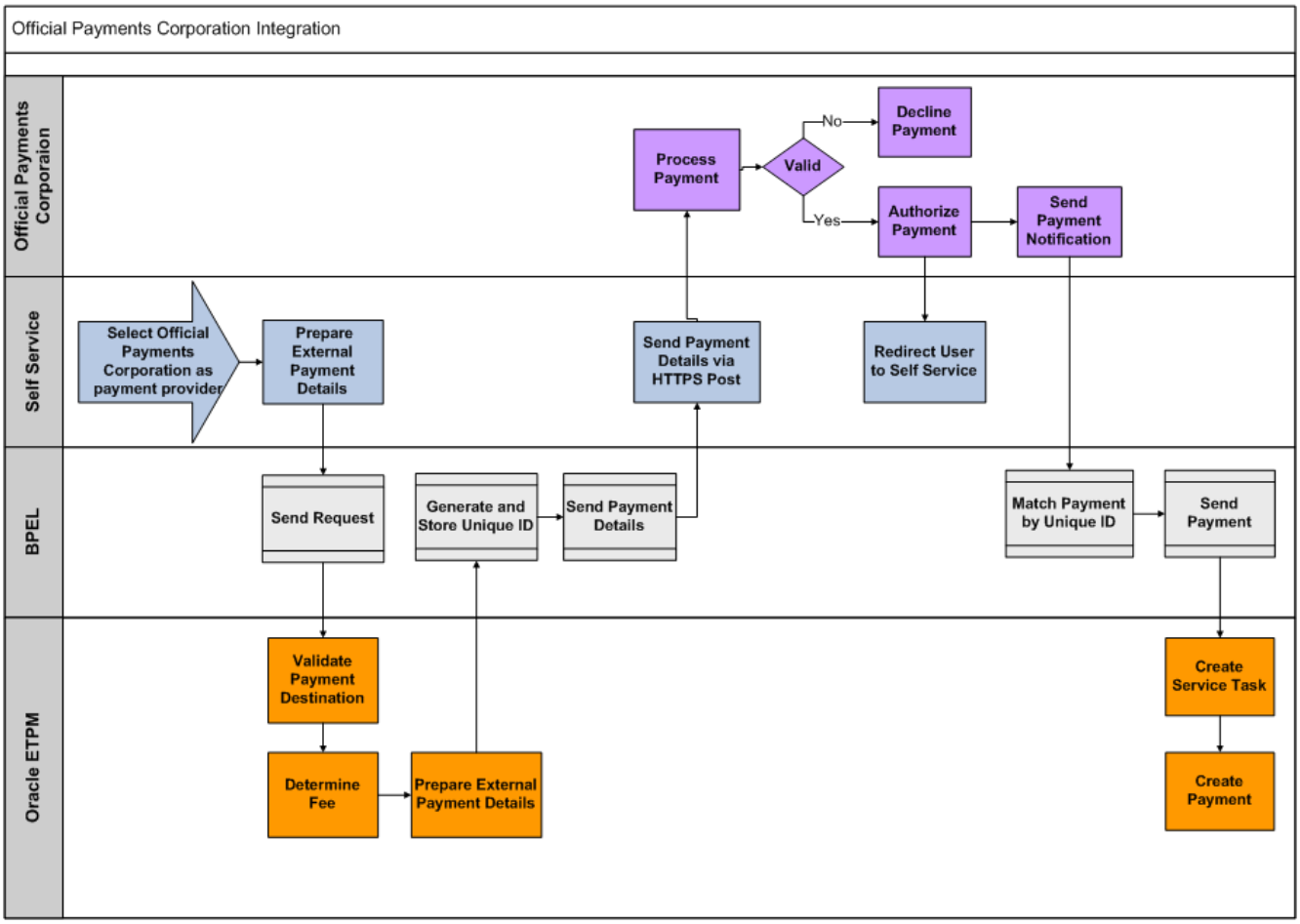
- Preparing External Payment Data. After the standard steps of identifying the taxpayer and validating the payment destination, when a taxpayer indicates payment using a credit card, the self service system sends a web service request to this product to determine whether a fee is applicable and to return additional payment data.
- Payment processing. Once the credit card information is processed by the external vendor, a web service request with the payment information is sent to this system. This causes an appropriate service task to be created to process the payment and apply it to the taxpayer's balance appropriately.
- Reconciliation. External vendors may supply reconciliation information for all the payments that were processed in a given time period. The product supports the ability to receive the reconciliation information, compare it against our records and highlight any discrepancies.

> **Note:** Integration with Official Payments Corporation (OPC). The product provides functionality to integrate with OPC as an external vendor. The logic provided in the base product is based on that vendor's requirements.

Refer to the self service documentation for more information about configuring the system to work with external vendors.

The following topics provide additional detail about the integration functionality provided.

The diagram illustrates the prepare external payment data and payment processing components for the integration with OPC.



Prepare External Payment Data

Once the taxpayer has indicated that payment will be made by credit card, the self service application sends a web service request to determine the fee requirement and to retrieve detail about the taxpayer to provide to the external vendor. These are processed by the XAI Inbound Service **TSPrepareExtPaymentData**, with an action of **PREPARE**, which invokes the service script **C1-PrExPyDta**. This script does the following:

- It uses the payment destination to determine the appropriate service task type (via the master configuration) and uses this information to determine the fee requirements. External vendors often charge fees for using their service. The tax authority may to choose to pass the fee onto the taxpayer or to absorb the cost of the fee. The product supplies configuration to allow the tax authority to define for each service task type whether taxpayers should be charged the fee based on the person's person type.

  > **Note:** Refer to the Service Task Types section of *Configuration Tasks for Online Payments* for more information.

- It retrieves additional information about the taxpayer, if required by the external vendor. It looks for a Prepare Data service script linked to the external vendor's record. If one is defined it is called. Refer to *Configuration Tasks for Payment Vendors* for more information.

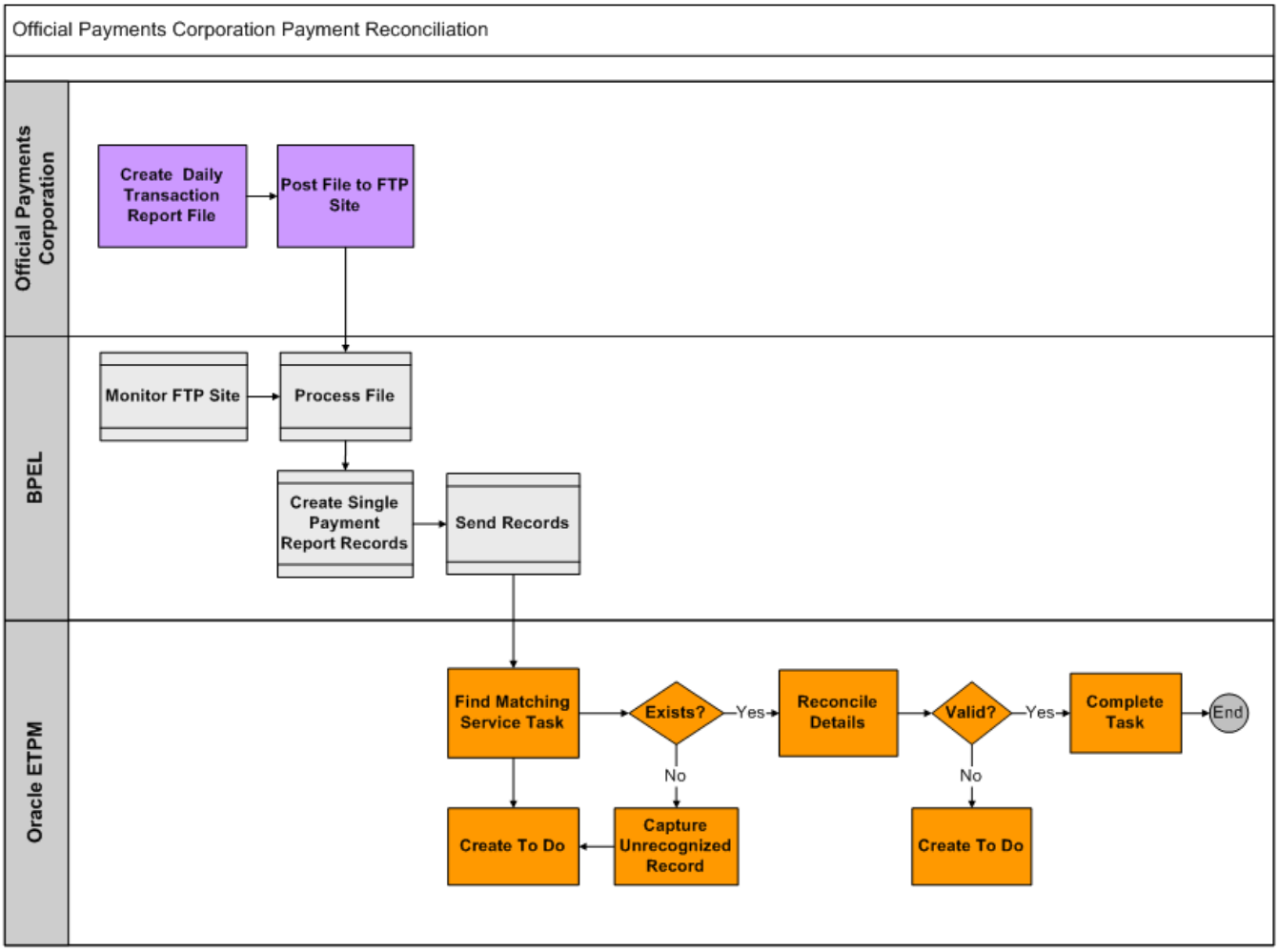
Payment Processing

For the most part, the payment processing for these payments is analogous to the processing done for bank payments. The tender type used should be one that is not configured to integrate bank details. And an external reference to the payment vendor's record should be recorded with the service task using a characteristic. This allows the reconciliation process to identify the record.

The product supports capturing additional information as defined by the requirements for the external vendor. The data is stored in a "raw" element on the service task called externalPaymentData. The structure of the information captured must be defined using the External Payment Data Area linked to the payment vendor lookup.

Reconciliation

The following diagram illustrates the reconciliation component for the integration with OPC.



Once the service task successfully creates the payment, it checks whether reconciliation with the payment vendor is applicable and if so, it transitions to the **Pending Reconciliation** state to wait for the reconciliation details.

When the reconciliation details are processed, individual web service calls are sent to the system for each payment. These are processed by the XAI Inbound Service **TSProcessExtPayReportRecord**, which invokes the service script **C1-PrcPyRpRc**. This script uses the external reference to find an existing payment service task in the **Pending Reconciliation** state with this reference linked as a characteristic.

- If one is found, it is updated with a snapshot of the payment report from the vendor. The payment report data is stored on the service task in a "raw" element called externalPaymentReport. The structure of the information captured must be defined using the External Payment Report Data Area linked to the payment vendor lookup. In addition, it is marked as ready to reconcile. The next time the reconciliation monitor runs the service task's reconciliation algorithm is executed. The algorithm is configured to call logic that is specific for the payment vendor. It calls the Payment Reconciliation Script defined on the payment vendor lookup. If there are any issues it transitions to **Error**, otherwise it transitions to **Complete**.
- If one is not found, the system creates a new service task record using a special task type configured for logging an unrecognized external payment in the payment report. The service task BO provided by the product for this scenario **C1-UnrecognizedExtPymtRptTsk** provides a simple lifecycle that creates the record in the **Pending** state. The next time the service task deferred monitor runs, the record creates a To Do entry for a user to research and resolve the issue and sets this record to **Complete**.

Fastpath:

Refer to *Configuration Tasks for Payment Vendors* for more information.

Configuration Tasks for Online Payments

The following sections list the configuration tasks needed to implement the online payment functionality.

Standard Payment and Automatic Payment Options

When the one-time payment service task processes the payments it relies on standard payment logic in the product. Refer to the payment configuration documentation for configuration options required for general payment processing as well as configuration options for automatic payment processing.

> **Fastpath:** Refer to the Defining Financial Transaction Options > Managing Payment Setup and Automatic Payment Options in the administration guide for more information.

Obligation Type for Excess Credit

The base product payment distribution rule algorithms that create payments support the ability to direct excess credit to a special obligation rather than crediting the obligation used to levy the taxes. If your implementation uses the base product algorithms, define an obligation type to use when directing excess credit to a special obligation for the account.

Suspense Obligation

In some cases when an account cannot be found for directing a payment, a payment is created in suspense, meaning that it is created for a suspense obligation. If your implementation does not already have a suspense obligation set up, create one.

Match Types

Define a match type for Obligation. This is used by the Pay a Pay Plan distribution rule.

Algorithms

Create a Distribution Rule - Create Payment algorithm for each supported payment destination.

- **Pay a Filing Period**. If your implementation supports the ability to direct a payment to a filing period, define an algorithm for the algorithm type **C1-PAYFRM**. In the parameters define the division / obligation type for the excess credit obligation along with the suspense obligation. Finally, define characteristic types for each of the payment details that are required by the algorithm. Note that the product supplies characteristic types that may be used for each one.
  - C1-DLN for document locator number. Note that this is not expected to be supplied by a taxpayer for web self service payments, but the parameter is required for the algorithm.
  - C1-TXPID for taxpayer ID.
  - C1-TAXTY for tax type.
  - C1-FLNGP for filing period.
- **Pay a Collection Notice**. If your implementation supports the ability to direct a payment to a collection notice, define an algorithm for the algorithm type **C1-DR-PAYCC**. In the parameters define the division / obligation type for the excess credit obligation. In addition, define the characteristic type that the overdue process uses to log the creation of the customer contact. This allows the algorithm to determine the related overdue process to find the covered obligations.
- **Pay a Pay Plan**. If your implementation supports the ability to direct a payment to a pay plan, define an algorithm for the algorithm type **C1-DR-PAYPP**. In the parameters define the division / obligation type for the excess credit obligation. In addition, define the match type for referencing the pay plan's obligation.

Distribution Rules

Create a distribution rule for each supported payment destination.

- **Pay a Filing Period**. If your implementation supports the ability to direct a payment to a filing period, multiple distribution rules must be created, one for Taxpayer ID, one for Tax Type and one for Filing Period. Each should reference the appropriate characteristic type. Refer to the create payment algorithm above for details of the characteristic types to use. Note the following points related to configuring these distribution rules:
  - The payment distribution processing expects that the create payment algorithm is only linked to the last distribution rule.
  - The payment distribution processing expects that the Determine Tender Account is linked to all the distribution rules.
  - The payment details processor business service used by the payment service task may expect the definition of the distribution rules to be in a certain order. Refer to the business service description for **C1-PaymentDestinationTxTyFilPd** for more information.
- **Pay a Collection Notice**. If your implementation supports the ability to direct a payment to a collection notice, create a distribution rule referencing the characteristic type **C1-CUSCN** Customer Contact. It should also reference the Create Payment algorithm created above for paying a collection notice.
- **Pay a Pay Plan**. If your implementation supports the ability to direct a payment to a pay plan, create a distribution rule referencing the characteristic type **C1-PAYPL** Pay Plan. It should also reference the Create Payment algorithm created above for paying a pay plan.

Business Object

Decide whether or not an email confirmation should be sent to the taxpayer once the payment is processed. If so, add an Enter plug-in to the In Progress state after the payment creation algorithm or perhaps the Complete state to generate the confirmation email.

> **Fastpath:** Refer to the algorithm section of the *General Configuration Tasks* for more information about configuring the confirmation algorithm.

Service Task Types

Taxpayer Identification Task Type

Define a task type to use for taxpayer identification with the business object **Taxpayer Service Request Self Service Task Type**.

- The Related Transaction BO should be set to **C1-TaxpayerIdentifySSTask**.
- Service Task Class should be set to **Taxpayer Identification**.
- To Do Type should be set to a standard error To Do type to use when transitioning to error. The base product To Do Type **Self Service Task Issues (C1-SSTTD)** is provided for this purpose. Note that the algorithm plugged in on the error state for the base business object checks that there is not already a To Do for the record before creating one of this type. This allows algorithms that detect an error to also create a To Do with a specific type if desired.
- To Do Role should be configured to an appropriate default To Do role for the above To Do type. This is optional. If no value is provided the default role for the To Do type is used.
- Configure the Confirmation Header Message Category and Message Number and the Error Header Message Category and Message Number to use for communication to the self service user.

> **Fastpath:** Refer to the Messages section of the *General Configuration Tasks* for more information.

- The service request fields mapping is used when creating the target service task to correctly populate the requestField list from the list of fields passed in on the web service request XML. For the base transaction business object, the following fields are expected:

Sequence	Transaction BO XPath
1	requestData/legalName
2	requestData/idType
3	requestData/personIdNumber
4	requestData/emailAdd
5	requestData/addressLine1
6	requestData/addressLine2
7	requestData/city
8	requestData/county
9	requestData/state
10	requestData/postal
11	requestData/onBehalfOfFlag
12	requestData/secondaryLegalName
13	requestData/secondaryIdType
14	requestData/secondaryPersonIdNumber
15	requestData/secondaryEmailAdd

Fastpath: Service Task Type Mapping. The service task type defined here must be configured in the service task domain value mapping in the SOA Composer application. Refer to the self service product documentation for details.

Payment Destination Task Types

Create a service task type for each payment destination with the business object **One-Time Payment Self Service Task Type**. Each service task configured should include the following common configuration.

- The Related Transaction BO should be set to **C1-StandardOneTimePaySSTask**.
- Service Task Class should be set to **Self Service Payment**.
- To Do Type should be set to a standard error To Do type to use when transitioning to error. The base product To Do Type **Self Service Task Issues (C1-SSTTD)** is provided for this purpose. Note that the algorithm plugged in on the error state for the base business object checks that there is not already a To Do for the record before creating one of this type. This allows algorithms that detect an error to also create a To Do with a specific type if desired.
- To Do Role should be configured to an appropriate default To Do role for the above To Do type. This is optional. If no value is provided the default role for the To Do type is used.
- Configure the Confirmation Header Message Category and Message Number and the Error Header Message Category and Message Number to use for communication to the self service user.

Fastpath: Refer to the Messages section of the *General Configuration Tasks* for more information.

The following configuration differs based on the payment destination:

- Payment Distribution Rules. Link the appropriate distribution rule(s) created above for the payment destination represented by this service task type.

- Link an appropriate Processor business service or service script that knows how to receive the detail related to the payment destination and create the payment event distribution details correctly. The base product provides one Processor business service for each supported payment destination. Using the search look for business services that begin with **C1-PaymentDestination%**
- Field mappings

  . For each payment destination detail captured on the service task, indicate the target XPath value in the above processor.

Finally, if external vendors are supported, indicate whether or not fees are appropriate based on the taxpayer type.

> **Fastpath:** Refer to *Implementing A Fee Requirement Script* for information.

Master Configuration

The self service master configuration record includes several settings required for payment related processing.

- Char Type for Overdue Process Log. If your implementation supports the Pay a Collection Notice payment destination, define the characteristic type that is used to link the customer contact to the overdue process. The base product provides a value that may be used.
- Refer to the ACH Agreement Validation section below for information about the Char Type for ACH Agreement Verification.
- Tender Types. Indicate the tender types that are valid for paying online.
- Supported Payment Destinations. For each supported payment destination (as defined in the **C1_PAYMENT_TARGET_FLG** lookup), indicate the corresponding Service Task Type.
- Define the customer contact types that represent the collection notices that a taxpayer may pay online using the Pay a Collection Notice payment destination.

ACH Agreement Validation

If your implementation requires a pre-signed agreement from the taxpayer when submitting a payment through their bank account, define a characteristic type on the taxpayer, the following configuration is needed:

- Create an appropriate characteristic type for **ACH Agreement**. Define a valid characteristic entity of **Person**.
- Update the master configuration to indicate this characteristic type.
- Update the business object **C1-StandardOneTimePaySSTask** to plug in the validation algorithm to check for the existence of this characteristic. Navigate to the **Lifecycle** tab and expand the configuration for the **Pending** state. Add an entry in the Algorithms collection with a System Event of **Enter**, an appropriate sequence and the algorithm **C1-CHKACHAGR**.

> **Note:** The product does not provide any functionality for populating the characteristic on the taxpayer record. This functionality is custom and must be provided by your implementation.

Configuration Tasks for Payment Vendors

The following sections list the additional configuration tasks needed to support an external payment vendor.

External Payment Report Task Type

Create a service task type to use when a payment in the reconciliation report for an external vendor is not found.

- The service task type BO should be **C1-StandardSelfServiceTaskType**
- The related transaction BO should be set to **C1-UnrecognizedExtPymtRptTsk**.

- To Do Type is not applicable. The service task BO has an algorithm that creates a To Do based using the unrecognized payment To Do Type on the payment vendor lookup.
- To Do Role is not applicable
- Confirmation Header Message Category and Message Number and Error Header Message Category and Message Number are not applicable.

External Vendor Extendable Lookup

Create a payment vendor extendable lookup record for the external vendor that is used by the self service application.

- If additional information is required by the external vendor to process the payment (such as the taxpayer's address), indicate an appropriate **Prepare Payment Data Script** that populates the data. The script should use the data area **C1-ExternalPaymentDetails** as its schema. The data area defines a paymentDetails list, which is a collection of field name and value pairs, to capture the additional information.

  Note: Integration with Official Payments. The product provides a service script **C1-BldExPyDt** that builds the additional data required for an integration with Official Payments.

- If additional information is included in the payment processed by the external vendor, indicate the **External Payment Data Area** that describes the structure of the data. This information is captured in the service task created to process the payment in the raw element externalPaymentData.

  Note: Integration with Official Payments. The product provides the data area **C1-OPCExtPaymentData** that supports the additional data sent with payments made via integration with Official Payments.

If the vendor supports a reconciliation report, check the **Reconciliation Required** checkbox and fill in the remaining detail.

- Indicate the **External Payment Report Task Type** created above.
- Define the **External Payment Report Data Area** that describes the structure of the data supplied with the reconciliation report.

  Note: Integration with Official Payments. The product provides the data area **C1-OPCExtPaymentData** that defines the data provided in the reconciliation report from Official Payments.

- Indicate the **Number Of Days To Wait For Timeout** appropriate for your implementation.
- Define the **Payment Reconciliation Script** that includes the vendor specific logic invoked by the reconciliation algorithm. The script should reference the business object **C1-StandardOneTimePaySSTask** as its schema.

  Note: Integration with Official Payments. The product provides a service script **C1-PymtRecon** that provides the additional payment reconciliation logic for an integration with Official Payments.

Additional Topics

The following sections provide some additional information related to online payment functionality.

Implementing New Payment Destinations

Your implementation may wish to support additional payment options for your taxpayers. For example, perhaps a certain tax type is assigned an external identifier on its tax role and your taxpayers are able to use that identifier to "pay a given tax type". For the purposes of this section, the assumption is that the external identifier is unique across all tax roles in the system. The following information outlines the steps required in this product to support an additional payment destination.

Note: Additional steps are required in the self service product. Refer to the self service product documentation for more information.

Lookup

Define a new value in the **C1_PAYMENT_TARGET_FLG** lookup to represent the new payment destination.

Create Payment Algorithm

Each payment destination requires a create payment algorithm that understands how to direct the payment to the appropriate obligation(s). In our example of paying a filing period via external id, the create payment algorithm must use a provided External ID to determine an appropriate Tax Role. From that tax role, the algorithm must determine the obligation(s) for the tax role that have outstanding debt and direct the payment towards the obligations appropriately.

The algorithm must be coded and implemented using an Algorithm Type and related Algorithm. The base product Create Payment algorithm types may be used as examples.

Characteristic Type

If there is not already a base characteristic type for each distribution detail value that must be provided for the payment distribution to work, one must be created. In this example, an Ad-hoc characteristic type to capture the External ID must be defined.

Distribution Rule

An appropriate distribution rule for each distribution detail required by the new payment destination must be created. In our example a distribution rule for "pay a given tax type" is required configuring the above created characteristic type and create payment algorithm.

> **Note:** If the payment destination requires multiple pieces of information, such as a ID plus a filing period, then multiple distribution rules are required, each referring the appropriate characteristic type. Only the last distribution rule should reference the Create Payment algorithm.

Payment Processor Service

The processor service is used for the following purposes:

- Validation. When the payment service task is first created, the processor is called to validate the details. The processor should use the payment destination details to verify that an appropriate record is found and that the appropriate account for that record can be determined. For our example, the processor should take the input External ID and find one and only one tax role with that ID and determine the tax role's account. The processor should provide appropriate errors if the data cannot be found.
- Validate Only. This is a variation of validation and is used for checking payment details when using an external vendor. The only difference between this and the validation logic is that the account ids are not retrieved for this logic.
- Build. When getting ready to process the payment, the service task BO first uses this processor to build the appropriate collection of Payment Distribution Details expected by the creation of a payment. The distribution details built by the processor are captured in the service task in the paymentDistributionList collection. A subsequent algorithm uses this information to create the payment with this distribution detail.

When creating a new custom Processor, use any of the base business services provided as an example of the type of functionality needed. All base business services provided begin with **C1-PaymentDestination%**. The processor may also be implemented as a service script, if preferred.

The custom service, whether it be a service script or a business service should include the data area **C1-PayDestProcessorCommon** along with a destination details group that includes the list of payment details provided by the taxpayer. These elements may be defined with specific identifiers. The service task type will contain mapping to populate the elements from the input information.

Service Task Type

Each additional payment destination requires its own service task type. Refer to *Configuration Tasks for Online Payments* for more information.

Implementing A Fee Requirement Script

When integrating with an external payment vendor, the product provides the ability to indicate for each service task type (i.e., payment destination) if the fee should be passed onto the taxpayer or not based on taxpayer type. Your implementation may conditionally pass on the fee to certain kinds of taxpayers. For example, maybe when a taxpayer is paying a filing period, the fee is passed on if the taxpayer is not paying the filing period on time. The product provides a service script (**C1-FeeReqFPr**) that includes this logic.

If your implementation has rules for when a fee should be passed onto the taxpayer based on specific conditions, a service script with the appropriate rules must be implemented. The above base script should be used as a sample, including the input and output that is expected.

Maintaining Self Service Tasks

Use the Service Task transaction to view and maintain service tasks. Navigate using **Main Menu** > **Self Service** > **Service Task** .

Self Service Task Query

Use the query portal to search for a service task. Once a service task is selected, you are brought to the maintenance portal to view and maintain the selected record.

Self Service Task Portal

This page appears when viewing the detail of a specific self service task.

Self Service Task - Main

This portal appears when a Self Service Task has been selected from the Self Service Task Query portal.

The topics in this section describe the base-package zones that appear on this portal.

Self Service Task

The Self Service Task zone contains display-only information about the selected Self Service Task, including the following:.

- Audit information for the task, including the web user ID and name, the email address and IP address.
- The list of related objects associated with the task
- Any error information
- Confirmation ID and message and any confirmation details
- Other information that is specific for the type of task. Refer to the in-line help text for more information about the fields.

Self Service Task - Log

Navigate to the Log tab to view the logs for a self service task.