myCalPERS Employer Technical Resources & Toolkit Guide 2019

Version History

Version	Description
v1.0	Initial Draft
v1.1	Revised
v1.2	Initial release of document
v1.3	Revised to include validation files, CalPERS ID File Exchange Layout, updated payroll file, schema and mdi with Division's CalPERS ID and updated enrollment file and schema with Retirement System field added; added change log to track release date of files.
v3	Updated the version of this user guide to coincide with the release of version 3 of the Technical Toolkit (T.T.). Made slight change to font formatting in Glossary (1.3) Added note to Table 3.0 indicating the SampleHealthEnrollment.xml was not ready for the version 3 release of the T.T..
v4	Updated to include information on new T.T. components (State Retirement Enrollment File and State Retirement Enrollment Validations).
v5	Updated Table 3.0 (Contents of Toolkit) to include schema documents.
v6	Added link to Tools Available on the Internet Added reference to Schema Documentation (sec 1.2, sec 3 & sec 3.2). Added Guide to myCalPERS File Readiness to Table 3.0
v7	Updated Table 3.0 to include XML Validator and Guide to XML Validator
v8	Changes not noted.
v9	Changes not noted.
v10	09/04/17 - Revised to remove validation files. Updated to coincide with June 2017 version of the T.T.
v11	01/18/18 – Updated for accessibility.
V12	05/29/19 – Updated for enhanced accessibility requirements.

Employer Technical Resources & Toolkit Guide

Introduction

The Technical Resources page on the CalPERS website consists of various tools to assist external business partners with their CalPERS reporting and file transfer needs. The purpose of this guide is to familiarize you with these resources available.

Contents

Introduction to Technical Resources 4

Unit 1: Employer's Technical Toolkit Guide 5

Unit 2: XML File Development Tips 15

Unit 3: File Testing and the Test Environment 34

Technical Support 40

Appendix 41

Introduction to Technical Resources

Overview

The first step in navigating the Technical Resources page is determining what type of resource you require and for what purpose.

Resources are broken-down into the following business partner types:

- Employers
- Direct Authorization Vendors
- Health & Dental Carriers

Employers

Employers are CalPERS business partners who contract directly with CalPERS for their employee's retirement, health, or Supplemental Income Plan (SIP 457) benefits.

Direct Authorization Vendor (DAV)

A Direct Authorization Vendor (DAV) is an organization authorized to receive monthly deductions from a retirement warrant requested by an annuitant to pay premiums, loans, etc. to a sponsored company such as: Credit Union, Employee Association, Carrier Insurance, etc.

Technical Toolkits

Technical Toolkits are available to assist with the development of XML files. XML files are required for deduction requests for DAVs and reporting of health, retirement enrollment, and contribution information to CalPERS. Additionally, supplemental information is available for payroll, and health/retirement enrollment SFTP procedures.

Making Payments

Instructions for processing payments (via EFT or check) to CalPERS can be found in the *Making Payments* section of the Technical Resources page.

File Readiness Testing & Resources

In the right-hand column of the page you will also find:

- File Readiness Testing Environment access and support
- Quick links to the technical toolkits and reference guides available
- Frequently asked questions

Unit 1: Employer's Technical Toolkit Guide

Introduction

The Employer Technical Toolkit provides external business partners development solutions to utilize the required XML File Upload or Secure File Transfer Protocol (SFTP) for reporting of the following events in myCalPERS:

- Health enrollments
- Retirement enrollments
- Payroll contributions (including SIP contributions)

This guide will describe the resources included in the Technical Toolkit and how to utilize them for your reporting needs.

Important Note

In addition to employers, some business partners may also interact with CalPERS as a direct authorization vendor (DAV), a health carrier, dental carrier or an association.

To assist CalPERS' business partners with their interactions, three separate *Technical Toolkit Guides* have been developed. If your organization falls into multiple categories, it is recommended that you review each of the appropriate guides which can be found on the Technical Resources page of the CalPERS website.

Contents

Glossary of Terms 6

Methods for Transmitting Data 3

Technical Toolkit Download Instructions 10

Employer Toolkit Contents 11

Employer Toolkit Quick Reference 12

Glossary of Terms

Data Element Definitions (DEDs)

A Data Element Definition (DED) document provides a table listing of data elements included in a particular reporting file to be submitted by an employer. The table includes the name of the data element, whether it is required, conditional or optional to be included in the file, the level of the hierarchy within the file where the data element fits, the type of data, and any valid values or particular format that needs to be used when reporting the data.

Note: The Data Element Definitions do not describe the file structure. Please refer to the appropriate XML Schema Definitions (XSDs) found in the Technical Toolkit for file structure specifications.

Encryption

Encryption is the cryptographic transformation of "plaintext" data into "ciphertext" data 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.

File Transfer Protocol (FTP)

FTP is a TCP/IP protocol specifying the transfer of text or binary files across the network. FTP is the most secure method for transferring data to CalPERS for reporting.

HyperText Transfer Protocol (HTTP)

HTTP is the protocol in the Internet Protocol (IP) family used to transport hypertext documents across an internet.

HyperText Transfer Protocol/ Secure (HTTPS)

When used in the first part of a URL (the part that precedes the colon and specifies an access scheme or protocol), this term specifies the use of HTTP enhanced by a security mechanism, which is usually SSL.

Public Key

A Public Key is the publicly-disclosed component of a pair of cryptographic keys used for asymmetric cryptography.

Response File

A Response File is the file returned to external business partners that indicates the success or failure of the transactions submitted to CalPERS via Secure File Transfer Protocol (SFTP).

Sample XML

Sample XML is a file containing an example of XML for a particular reporting file to demonstrate the XML content and format of the file.

Secure File Transfer Protocol (SFTP)

SFTP is a communications protocol used to transfer files without compromise of data. Secure FTP provides extra security by encrypting the files before transmission. It encrypts both the commands and the data, preventing passwords and sensitive information from being transmitted in clear text over the network.

Simple Object Access Protocol (SOAP)

This is the formal set of conventions governing the format and processing rules of a SOAP message. These conventions include the interactions among SOAP nodes generating and accepting SOAP messages to exchange information along a SOAP message path.

Validation File

A validation file is specific to a given transaction set describing the error codes and error messages that are generated for each data element validation performed.

Web Service

Web Service is a software system designed to support interoperable machine-to-machine interaction over a network.

XML Schema

An XML schema is a way to describe and validate data in an XML environment. A schema is a model for describing the structure of information. XML Schema (XSD) is a recommendation of the W3C.

Methods for Transmitting Data

Interface Transmission Methods

CalPERS offers two methods for employers to transmit interface files to CalPERS.

1. Manual File Upload while logged into myCalPERS
2. Secure File Transfer Protocol (SFTP) where files are exchanged between a client and a CalPERS server through a secured FTP connection.

All files are expected to be formatted per the Extensible Markup Language (XML) and CalPERS interface reporting standards.

Manual File Upload

With File Upload, the employer manually uploads a file into myCalPERS. As myCalPERS is accessed via secure HTTP (HTTPS), file encryption is not required when submitting files using this method. A process flow of the File Upload process is shown in Appendix 1.

It is recommended that employers familiarize themselves with the steps to log into myCalPERS, upload a file, review data, and use the preprocessing area to identify and, when necessary, correct erred records for reprocessing.

For additional support with validating your XML file or identifying test case scenarios, refer to the *Technical Support* section of this document for more information.

Secure File Transfer Protocol (SFTP)

Files transmitted to myCalPERS contain sensitive information and must be encrypted when transmitting an XML file via SFTP. Response files delivered as acknowledgements of SFTP file processing will be in XML format and encrypted.

Each employer will have a login and password to a specific directory where they will submit their encrypted XML files to a CalPERS inbound folder and retrieve the encrypted response file from a CalPERS outbound folder.

Employers will use a CalPERS public key to encrypt their files and will use their own private key to "sign" the file. Once CalPERS receives the file, we will use the employer's public key to verify the signature and use CalPERS private key to decrypt the file.

As testing of the file transmission process begins, CalPERS will assign technical resources to validate the test files submitted and provide guidance to employers for troubleshooting.

Employers looking to submit files via SFTP should refer to the SFTP Onboarding Guide found on the CalPERS website. In addition, it is recommended that employers familiarize themselves with the steps to log into myCalPERS, upload a file, review data, and use the preprocessing area to identify and, when necessary, correct erred records for reprocessing.

A flow of the SFTP transmission process is shown in Appendix 2.

Technical Toolkit Download Instructions

Downloading the File

The Employer Technical Toolkit file is available on the CalPERS website. To locate it, navigate to the myCalPERS Technical Requirements page under the **Employers** global navigation tab. This page includes further instructions for downloading the file.

Note: The contents of the Technical Toolkit are contained in a ZIP file for ease of downloading and to preserve the format of the XML schema files during download.

See Figure 1.1 for an example of contents found within the Technical Toolkit, Health folder.

Extracting the Contents of the File

After downloading the file, you will need to extract the contents. CalPERS recommends the WinZip program for extracting. WinZip is a file compression utility that allows several files to be compressed and contained within a single file.

Use of Contents

The contents within the Technical Toolkit will assist in the development and validation of the appropriate XML file(s) required to report data to CalPERS. Depending on the nature of the reporting relationship you have with CalPERS, you will use either a subset of the files located in the ZIP file or all of them.

Technical Toolkit Contents

Encryption/ Decryption Requirement

For employers reporting to CalPERS using XML, there are file transmission requirements which are outlined in the Encryption/Decryption Requirement document. This document details the file transmission requirements with respect to encrypting/decrypting files submitted/retrieved using SFTP, a description of the schema files (.xsd), and how they relate to the structure of the files transmitted.

Schema Documentation

For SFTP and File Upload reporting, the XML files require conformance to the XML Schema Definitions (XSD) which define the structure of each file. The XML schema file names and the files they correspond with are shown in the Toolkit Quick Reference Table. A graphical representation of the schema relationships is shown in Appendix 3.

Data Element Definitions

The Data Element Definitions (DEDs) are found within the Payroll, Health, and Retirement folders in the Employer Technical Toolkit. Each element in the definition file is numbered for reference and elements that are logically related are grouped together.

Note: These documents do not represent the order or relationship of data elements within the XML files. Also, for health and retirement enrollment, there are separate DED documents for State agencies (includes CSU) and public agencies and schools.

SFTP Onboarding Steps File

This file outlines the steps to ensure you have a myCalPERS compatible XML file developed prior to considering the SFTP exchange.

Response Files

After processing a file submission via SFTP, CalPERS will generate an XML response file indicating the success or failure of each submitted transaction. The schema for the XML format is included to assist the employer in building a process to retrieve and process the response file.

Sample Files

Sample XML files are provided as an example of what the resulting XML should look like (i.e. SamplePayroll.xml, SampleEnrollment.xml).

Employer Technical Toolkit Quick Reference

The following outlines the content of the Employer Technical Toolkit and the relevant files based on the employer's contracting relationship with CalPERS.

Documentation for Getting Started

Guide to the Technical Toolkit (PDF) – This guide defines the content within the Toolkit with instructions and guidelines specific to payroll, retirement and health transactions for the development of XML files and requesting SFTP access for business partners interested in submitting files via SFTP.

Encryption Decryption File Naming (PDF) – This document describes the Encryption Decryption Service, designed to allow an external entity (such as an employer) to interact with CalPERS using encrypted data files.

Common Schemas/Utilities

These XSD files are used when creating all XML files received by/sent from myCalPERS.

CommonUtilities (XSD) – This file defines the specifications for the data elements that are common across all of the XML files used by myCalPERS (i.e., addresses, phone numbers, dates, currency amounts, etc.).

SoapEnvelope (XSD) – This file is common across all files transmitted to CalPERS using SFTP or file upload.

Schemas for Creating Payroll Contribution XML Files

Schema Documentation (PDF) – This document provides the technical details for the payroll retirement transactions schema (PayrollRetirementV1.xsd).

PayrollRetirementV1 (XSD) – This file defines the structural requirements for building the payroll contribution XML file.

PayrollReportResponseV1 (XSD) – This file defines the structure of the file CalPERS will return to employers who submit payroll contributions in an XML file using SFTP. (SFTP submission only)

Documents for Creating Payroll XML Files

Payroll Contribution Data Element Definitions (PDF) – This document defines the data elements and business rules needed to transmit information within the payroll contribution XML file. These data elements are embedded into the structure found in the payroll contribution schema.

Payroll Response Data Element Definitions (PDF) – This document defines the data elements CalPERS will return in the response file when employers submit payroll contribution files using SFTP. (SFTP submission only)

Common Payroll Errors (PDF) – This document provides a list of common level 1 validation payroll error messages and resolution tips.

Creating a Payroll Adjustment Report (PDF) – This document provides the steps to manually convert a regular payroll earned period XML report into an adjustment report. This should be referenced by business partner's whose payroll software is limited to generating regular reports only.

20130123000000_123_10006_SamplePayroll.xml – A sample payroll contribution XML file, which can be used as a reference.

20160101000000_000_10006_SamplePayrollResponse.xml – A sample payroll contribution response XML file, which can be used as a reference for SFTP submitters. (SFTP submission only)

Schemas for Creating Retirement and Health XML Files

The XSD files listed below are used for building both retirement and health XML files for public agencies, State agencies and schools. You must refer to the appropriate DEDs, also listed below, for determining which data elements should be included in the XML file you are creating.

Schema Documentation (PDF) – This document provides technical details for the retirement and health transactions schema (RetirementHealthTransactionsV1.xsd)

RetirementHealthTransactionsV1 (XSD) – This file defines the structural requirements for building the retirement and health enrollment XML files.

RetirementHealthResponseV1 (XSD) – This file defines the structure of the file CalPERS will return to employers who submit retirement and health enrollments in an XML file using SFTP. (SFTP submission only)

Documents for Creating Retirement/Health XML Files

[Retirement/Health] Enrollment Data Element Definitions (PDF) – This document is for public agencies and non-CSU schools only and defines the data elements and business rules for inclusion in the retirement/health enrollment XML file. These data elements are embedded into the structure found in the retirement/health enrollment schema.

State - [Retirement/Health] Enrollment Data Element Definition (PDF) – This document is for State agencies and CSUs only and defines the data elements and business rules for inclusion in the retirement/health enrollment file. These data elements are embedded into the structure found in the retirement/health enrollment schema.

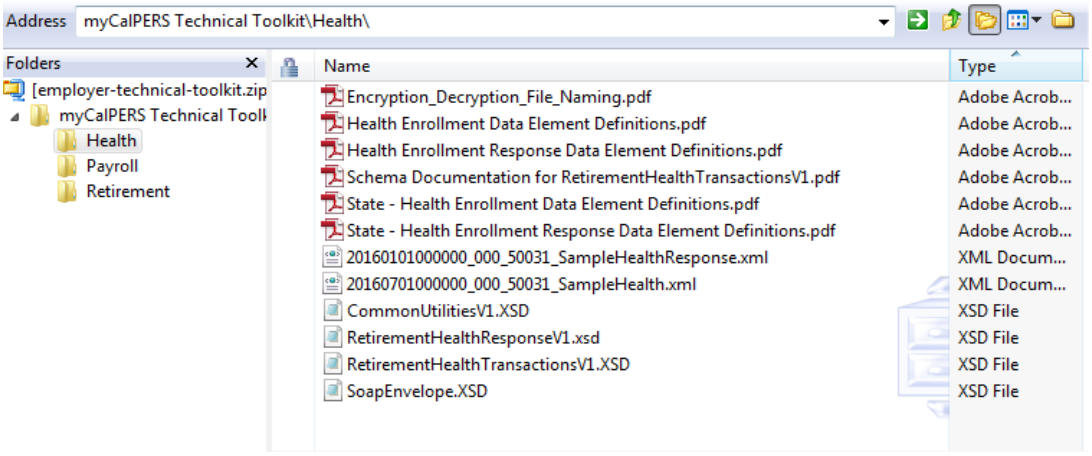
[Retirement/Health] Enrollment Response Data Element Definitions (PDF) – This document defines the data elements CalPERS will send to employers after processing the retirement/health enrollment data submitted through SFTP. (SFTP submission only)

State - [Retirement/Health] Enrollment Response Data Element Definitions (PDF) – This document defines the data elements CalPERS will send to State agencies/representatives after processing the retirement/health enrollment data submitted through SFTP. (SFTP submission only)

20160701000000_001_00007_Sample.xml – A sample retirement/health enrollment XML file, which can be used as a reference.

20160101000000_000_00007_SampleRespose.xml – A sample retirement/health enrollment response XML file, which can be used as reference for SFTP submitters. (SFTP submission only)

Figure 1.1 Employer Technical Toolkit Contents, Health Folder



Unit 2: XML File Development Tips

Introduction

This unit is designed to assist CalPERS' employer's technical staff with some of the nuances associated with developing the XML file as specified by myCalPERS.

These tips are meant to serve as supplemental guidelines to XML coding standards and principles that are readily available online. More information on various XML tools and resources is available in the Employer Technical Toolkit and the *Technical Support* section of this document.

Contents

Two Step File Validation 16

XML/XSD Background 16

XML Validation Tool 16

XML Development Guidelines 17

XML File Guidelines 25

Responsibilities by Phase 29

XML Structure Overview 31

Two-Step File Validation

The myCalPERS solution incorporates a two-step file validation concept.

File Level 1 Validation

Level 1 validation is looking to enforce the XML interface file specification or XML Schema Document (XSD) against the file that is being submitted. The main function of Level 1 validation is to make sure the file naming conventions are followed and ensure all required fields are present and the file being submitted is a proper XML file, meaning that start tags and end tags are present, no empty tags are submitted and maximum length of a data element is enforced.

File Level 2 Validation

Level 2 validation consists of enforcing the defined CalPERS business rules against the data contained in the XML file. Level 2 errors are well described by the error messages. For SFTP file submissions, Level 2 errors will be returned through a response file that will identify the errors. These can then be corrected in the employer's own system and the records can be resubmitted. Alternatively, Level 2 errors can be displayed on a user screen within myCalPERS where they can be corrected directly.

Essentially, Level 1 is checking that the file can be consumed by myCalPERS and Level 2 is checking the actual data contained in the XML file. This section focuses on Level 1 validation nuances.

XML/XSD Background

CalPERS requires all file reporting in XML format. The use of XML allows data transfer or document exchange within or across organizations with different platforms.

XSD defines syntax and shows how elements and attributes in an XML file should be contained. Some of the XSDs that CalPERS uses have nuances and idiosyncrasies that this guide will clarify to inform developers of how to interact with the system.

XML Validation Tool

Employers are encouraged to obtain an XML validation tool to ensure their file structure is compliant with CalPERS XSD schema specifications. These are readily available online with a quick search.

XML Development Guidelines

Introduction

This section reviews some of the more common problem areas and questions that arise when developing a CalPERS-defined XML file and recommendations for how to address those problems. Additionally, an *XML Structure Overview* can be found later in this document.

Contents

File Header Information 18

Order of Fields 19

Required Fields 20

No Data to Include in a Data Element 22

Schema Location Data Attribute 22

Boolean Values 23

XSD Sequence 23

XSD Max Length Value 24

File Header Information

Every file has an element defined as Interface Type ID. This ID is specific to the internal CalPERS-defined interface specifications and can be found in the file header along with other identifiers such as Business Partner ID and Schema Version.

Table 2.1 shows the Interface Type ID values related to each file that is being submitted or received. The Interface Number values should be used for the Interface Type ID for inbound and outbound files.

Table 2.1 – Interface Type ID

Interface	Interface Description	Interface Direction	Interface Number
Payroll Contribution	Used to process payroll transactions	Inbound to myCalPERS	10006
Payroll Response	Used to correct payroll errors when file is submitted and returned via SFTP	Outbound from myCalPERS	10006
Retirement Enrollment	Used to process retirement enrollment transactions	Inbound to myCalPERS	00007
Retirement Enrollment Response	Used to correct health enrollment errors when file is submitted and returned via SFTP	Outbound from myCalPERS	00007
Health Enrollment	Used to process health enrollment transactions	Inbound to myCalPERS	50031
Health Enrollment Response	Used to correct health enrollment errors when file is submitted and returned via SFTP	Outbound from myCalPERS	50031

Order of Fields

There is a set order (hierarchy) for the XML and data elements contained in the interface files. The hierarchy for each interface file is defined in its associated XSD; therefore, the XSD is the driving set of specifications when developing the XML file.

For example:
Payroll contribution with data elements in the correct order:

```
<n1:ParticipantInfo>
  <n1:ParticipantsCalPERSId>001234567</n1:ParticipantsCalPERSId>
  <n1:FirstName>James</n1:FirstName>
  <n1:LastName>Smith</n1:LastName>
</n1:ParticipantInfo>
```

Payroll contribution with data elements NOT in the correct order:

```
<n1:ParticipantInfo>
  <n1:FirstName>James</n1:FirstName>
  <n1:LastName>Smith</n1:LastName>
  <n1:ParticipantsCalPERSId>001234567</n1:ParticipantsCalPERSId>
</n1:ParticipantInfo>
```

In this example, the data element title in bold is *not in the correct* order which would cause the XML file to be rejected.

Start/End Tags

It is important to note that every data element must have a “Start Tag” and an “End Tag”:

- Start tag = <n1:Element name>
- End tag = </n1:Element name>

Start tags always open with a less than sign “<” whereas end tags always open with a less than sign followed by a slash “</”. The start and end tags contain the element name and surround the element value as shown in Example 1 and 2 above.

Required Fields

If a field is defined as "Required" and its parent element is also required, or it has no parent element, then it must always be present in the file. Otherwise the file will fail Level 1 validation.

For example:
In the payroll contribution file, the data elements <ReportPeriodBeginDate> and <ReportPeriodEndDate> are required fields and must always be included in the XML file:

<n1:ReportPeriodBeginDate>**2009-05-01**</n1:ReportPeriodBeginDate>
<n1:ReportPeriodEndDate>**2009-05-31**</n1:ReportPeriodEndDate>

If a field is defined as "Required" and its parent element is optional, then it must be present if its parent element is used, otherwise the file will fail Level 1 validation.

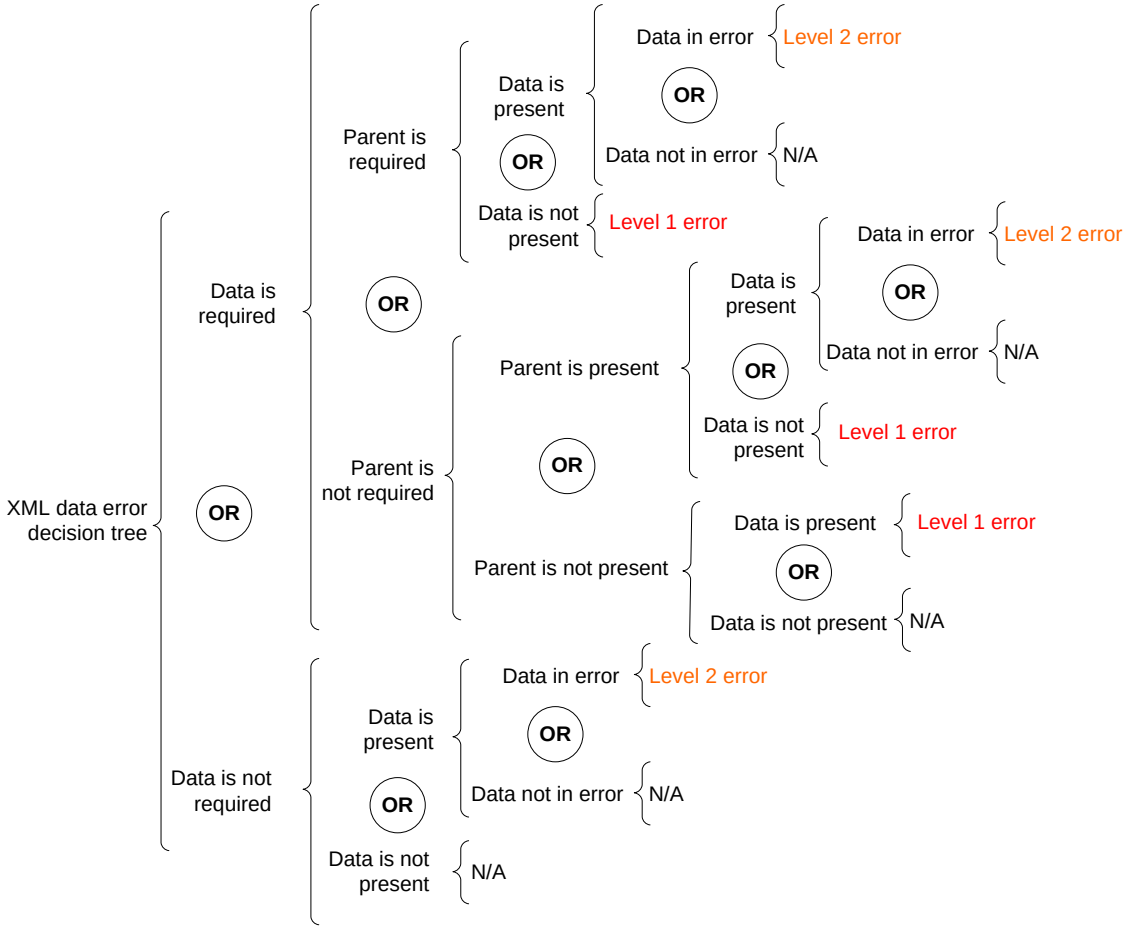
For example:
In the payroll contribution file, the following fragment of the XSD indicates that the element <SupplementalIncomePlanDetails> is optional (minOccurs ="0") but the child elements <SIPPlanID>, <SIPCount>, and <SIPTotal> are required.

```
<xs:element name="SupplementalIncomePlanDetails" minOccurs="0"
maxOccurs="unbounded">
        <xs:complexType>
                <xs:sequence>
                        <xs:element name="SIPPlanID">
                                <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                                <xs:maxLength value="10"/>
                                                <xs:pattern value="[0-9]*"/>
                                        </xs:restriction>
                                </xs:simpleType>
                        </xs:element>
                        <xs:element name="SIPCount" type="xs:int"/>
                        <xs:element name="SIPTotal" type="cuns:MoneyType"/>
                </xs:sequence>
        </xs:complexType>
</xs:element>
```

This means that if the element "SupplementalIncomePlanDetails" is not used, then the child elements should not be used either. If the element "SupplementalIncomePlanDetails" is used, then all three child elements must be present. If "SupplementalIncomePlanDetails" is used and, for instance, "SIPPlanID" is not present, a Level 1 error will occur.

Figure 2.1

This diagram provides a graphical representation of the scenarios under which Level 1 and Level 2 errors may be generated when required fields are missing the XML files.



No Data to Include in a Data Element

When using XML, if a data element is not going to contain data and it is not a required or conditional field, then the tags must be left out. If a tag is included and there is no data for the data element, then the XML file may fail Level 1 Validation. Using the payroll contribution XSD as an example below, if an individual does not have a middle name, the MiddleName data element tags (<n1:MiddleName> </n1:MiddleName>) are not to be included.

For example:

```
<n1:ParticipantInfo>
   <n1:ParticipantsCalPERSId>001234567</n1:ParticipantsCalPERSId>
   <n1:FirstName>James</n1:FirstName>
   <n1:LastName>Smith</n1:LastName>
</n1:ParticipantInfo>
```

Schema Location Data Attribute

The XML schema defines the location for which the schemaLocation attribute must be provided. This attribute uses pairs of values where the first URL reference is a namespace, and the second is the location of a schema that describes that namespace.

Note: The schemaLocation attribute is a location pointer. It's important to make sure that a space is entered between the first and second namespace (i.e. after V1 and before PayrollRetirementV1).

Using the sample payroll XML file as an example:

```
<n1:RetirementAndPayrollTransactions
xsi:schemaLocation="http://calpers.ca.gov/PSR/PayrollRetirementV1
PayrollRetirementV1.xsd" xmlns:cuns="http://calpers.ca.gov/PSR/CommonUtilitiesV1"
xmlns:n1="http://calpers.ca.gov/PSR/PayrollRetirementV1"
xmlns:rhtns="http://calpers.ca.gov/PSR/RetirementHealthTransactionsV1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

Boolean Values

Boolean data elements in all CalPERS XML files must be specified in all lower case. XML is case sensitive; therefore, it is very important to set the data elements to the proper case. For required Boolean fields, the tags must be included and the value must be “true” or “false”. Omitting the field does not equate to “false.”

For example:
The data element <TestReport>, which indicates whether or not a payroll contribution file is a test file, should contain the value of “true” if it is a test file:
<n1:TestReport>true</n1:TestReport>

XSD Sequence

The XSD defines the sequence in which data elements are to be included in the XML file. When an <xs:sequence> entry is shown in the XSD, the XSD will then follow with the order in which the elements need to appear in the XML file. If the order is changed, the XML file will fail Level 1 validation.

For example:

```
<xs:sequence>
        <xs:element name="RecordPeriodBeginDate" type="cuns:LocalDateType" />
        <xs:element name="RecordPeriodEndDate" type="cuns:LocalDateType" />
        <xs:element name="PayrollRecordMemo" type="cuns:UniqueTransactionIdentifierType"
        minOccurs="0" />
</xs:sequence>
```

In the example above, if the order is changed to where <RecordPeriodEndDate> comes before <RecordPeriodBeginDate>, the file will fail validation.

XSD Max Length Value

The Max Length value specification defines the maximum number of characters or digits that are allowed in the data field. This Max Length definition does not mean that the data element must always be the specified length but rather that a value cannot be more than the defined maximum length. If the XML schema specifies a Max Length of three but the code value(s) are two characters in length, report the two-character value without any padding or extra spaces.

For example:
A “MaxLengthExample” is specified as having a maximum length of three but a valid code value for this element is "AA" this would be represented in the XML file as:
<cuns:MaxLengthExample>AA</cuns:MaxLengthExample>

Specified Length Value

There are some data elements that must always be a specified length. This is indicated by the xs:length specification in the Common Utilities XSD.

For example:
The SSN-type data elements in the XSDs must always have a length of nine characters:

```
<xs:simpleType name="SSNType">
<xs:restriction base="xs:string">
<xs:length value="9"/>
```

Note: Since some SSNs have leading zeroes, when the value is added to the XML file, it should contain the leading zeroes.

<cuns:SSN>**001234567**</cuns:SSN>

Because the SSN must always be a nine-digit value, this value may be stored as a string in the source system or, if stored as a numeric value, should always be zero padded to ensure it is nine digits in length. In the example above, the SSN would be either be stored as “001234567” or have logic in place to take the numeric value of 1234567 and pad it with zeros to become the nine-digit length.

XML File Guidelines

Introduction

In XML, a well-formed file must conform to the following rules, among others.

XML Files Must Have a Root Element

XML files must contain one element that is the parent of all other elements. This element is called the root element.

```
<root>
        <child>
                <subchild>.....</subchild>
        </child>
</root>
```

For example:

```
<n1:Participant>
        <n1:ParticipantInfo>
        <n1:ParticipantsCalPERSId>1111219789</n1:ParticipantsCal
PERSId>
                <n1:FirstName>Lawrence</n1:FirstName>
                <n1:MiddleName>Trace</n1:MiddleName>
                <n1:LastName>Fermin</n1:LastName>
        </n1:ParticipantInfo>
```

In the example above <n1:Participant> is the root element, <n1:ParticipantInfo> is a child element, while <n1:ParticipantsCalPERSId> is a sub child element.

All XML Elements Must Have a Closing Tag

In XML, it is illegal to omit the closing tag. All elements must have a closing tag.

For example:

```
<p>This is a paragraph</p>
```

Although in XML, empty elements may be marked with an empty-element (self-closing) tag (such as <Gender/>), in myCalPERS, empty or null, data elements should be removed as null values are not processed. Only data elements with values should be included in the files.

XML Tags Are Case Sensitive

XML elements are defined using XML tags which are case sensitive. With XML, the tag <EmployerCalPERSId> is different from the tag <EmployercalPERSId>.

Opening and closing tags must be written with the same case:
<BirthDate>**1982-01-12**</BirthDate> Correct!
<BirthDate>**1982-01-12**</Birthdate> Incorrect!

Note: "Opening and closing tags" are often referred to as "Start and end tags." Use whatever you prefer; They are the same thing.

XML Elements Must Be Properly Nested

Tags may be nested but must not overlap. Each non-root element must be completely contained in another element. In XML, all elements must be properly nested within each other.

For example:

```
<ParticipantInfoType>
    <ParticipantsCalPERSId>0001234567</ParticipantsCalPERSId>
    <FirstName>John</FirstName>
    <LastName>Doe</LastName>
</ParticipantInfoType>
```

In the example above, properly nested simply means that since the <FirstName> element is opened inside the <ParticipantInfoType> element, it must be closed inside the <ParticipantInfoType> element.

```
<CommunicationInfoType>
    <Phone>HOM</Phone>
</CommunicationInfoType>
```

In the example above, properly nested simply means that since the <Phone> element is opened inside the <CommunicationInfoType> element, it must be closed inside the <CommunicationInfoType>

XML Attribute Values Must Be Quoted

XML elements can have attributes in name/value pairs just like in HTML.

In XML the attribute value must always be quoted.

For example:
#1 - Correct

```
<note date=”12/11/2007”>
        <to>Tove</to>
        <from>Jani</from>
</note>
```

#2 - Incorrect

```
<note date=12/11/2007>
        <to>Tove</to>
        <from>Jani</from>
</note>
```

The error in #2 is the date attribute in the note element is not quoted.

All Attribute Values Are Quoted With Single (‘) or Double (“) Quotes

Single quotes (‘) close a single quote, and double quotes (“) close a double quote.

To include a double quote inside an attribute value that is double quoted, or a single quote inside an attribute value that is single quoted, escape the inner quote mark using entity references.

Comments in XML

The syntax for writing comments in XML is:

```
<!-- This is a comment -->
```

Note: Never use the two dashes (--) anywhere but at the beginning and end of your comments.

Ampersand Use

An ampersand should always be represented as “&.”

Entity References

Some characters have a special meaning in XML. If a character like "<" is placed inside an XML element, it will generate an error because the parser interprets it as the start of a new element; this will generate an XML error: <message>if salary < 1000 then</message>

To avoid this error, replace the "<" character with an entity reference:
<message>if salary < 1000 then</message>

There are 5 predefined entity references in XML:

1. <
   <
   less than
2. >
   >
   greater than
3. &
   &
   ampersand
4. '
   '
   apostrophe
5. "
   "
   quotation mark

Note: Only the characters "<" and "&" are strictly illegal in XML. The greater than character is legal, but it is a good habit to replace it.

White Space Preservation in XML

With XML, the white space in a file is not truncated.

Sequence]]>

This sequence is not allowed in the XML content.

Responsibilities by Phase

Overview

Employer and CalPERS responsibilities follow schedule phases for employers preparing to exchange files with CalPERS. These include:

- Discover
- Design
- Develop
- Test
- Deploy

Table 3.1 –Responsibility & Success Criteria by Phase

Phase	Who	Responsibility	Success Criteria
Discover	Employer	Understand the interfaces to be used with CalPERS as well as methods for interacting with the CalPERS system	Ability to design internal system changes and extract files for exchange with CalPERS as well as the business processes to interact with CalPERS
	CalPERS	Share information on interfaces to be used by employer	Employer understands CalPERS interface designs
Design	Employer	Design system changes to accommodate data for exchanges and to interact successfully with CalPERS	Complete designs to ensure mandatory data elements for file exchanges will be available under defined conditions in file format specifications
	CalPERS	Share design information on interfaces to be used by employer; notify employer of design changes	Employer is clear on what data is mandatory under which conditions; design changes are communicated as soon as possible
Develop	Employer	Implement designed changes to internal systems to allow future file exchanges with CalPERS	Changes are implemented and allow the system to operate as intended for exchanging files with CalPERS
	CalPERS	Clarify design questions from employers	Employers can complete implementation for valid file exchanges

Phase	Who	Responsibility	Success Criteria
Test	Employer	Test file exchanges with CalPERS to verify operability and error processing requirements	Satisfied that the file exchange is operational and that errors are known and can be corrected
	CalPERS	Provide a test platform and guides for employers to test their interfaces	Provide responses to questions and an operational test platform for use by employers
Deploy	Employer	Deploy system and process changes to operational status when coordinated by CalPERS for production deployment	Systems and business processes are deployed and operational; file exchanges can continue with myCalPERS
	CalPERS	Coordinate production deployment timing and conditions with employers	Deployment dates and conditions are communicated appropriately to coordinate operational status

XML Structure Overview

Introduction

Two XSDs are common across all XML files being exchanged between file reporting business partners and CalPERS:

- CommonUtilitiesV1.xsd
- SoapEnvelope.xsd

The remaining XSD(s) define the body of the XML files.

The following information describes these specific XSDs and how to structure the content of the Payroll, Retirement, and Health files sent by file reporting employers and CalPERS. Files exchanged via File Upload or SFTP will be structured as a SOAP Envelope.

Payroll

SOAP Envelope

- The interface file must contain a root element named “Envelope” with the namespace identifier of “http://schemas.xmlsoap.org/soap/envelope/”
- An envelope MUST have
    - exactly one child element called soap:Header
    - exactly one child element called soap:Body
- An envelope MUST NOT have any element children of soap:Envelope following the soap:Body element

SOAP Header

- This element must have one child element named HeaderInfo with a Namespace identifier of “http://calpers.ca.gov/PSR/CommonUtilitiesV1”

SOAP Body

- This element of a **file inbound** to CalPERS must have one child element named RetirementAndPayrollTransactions with the namespace identifier of “http://calpers.ca.gov/PSR/PayrollRetirementV1”
- The soap:Body element of a **file outbound** from CalPERS must have one child element named EmployerPayrollReportResponse with the namespace identifier of “http://calpers.ca.gov/PSR/EmployerPayrollReportResponseV1”

XSD

- Inbound from Employers to CalPERS – PayrollRetirementV1.xsd
- Outbound from CalPERS to Employers – PayrollReportResponse.xsd

XML Structure Overview, continued

Retirement

Envelope

- The interface file must contain a root element named “Envelope” with the namespace identifier of “http://schemas.xmlsoap.org/soap/envelope/”
- An envelope MUST have
    - exactly one child element called soap:Header
    - exactly one child element called soap:Body
- An envelope MUST NOT have any element children of soap:Envelope following the soap:Body element

SOAP Header

- The soap:Header element must have one child element named HeaderInfo with the namespace identifier of “http://calpers.ca.gov/PSR/CommonUtilitiesV1”

SOAP Body

- The soap:Body element of a **file inbound** to CalPERS must have one child element named RetirementHealthEnrollment with the namespace identifier of “http://calpers.ca.gov/PSR/RetirementHealthTransactionsV1”
- The soap:Body element of a **file outbound** from CalPERS must have one child element named RetirementHealthResponse with the namespace identifier of “http://calpers.ca.gov/PSR/RetirementHealthTransactionsV1”

XSD

- Inbound from Employers to CalPERS – RetirementHealthTransactionsV1.xsd
- Outbound from CalPERS to Employers – RetirementHealthResponseV1.xsd

XML Structure Overview, continued

Health

Envelope

- The interface file must contain a root element named “Envelope” with the namespace identifier of “http://schemas.xmlsoap.org/soap/envelope/”
- An envelope MUST have
    - exactly one child element called soap:Header
    - exactly one child element called soap:Body
- An envelope MUST NOT have any element children of soap:Envelope following the soap:Body element

SOAP Header

- The soap:Header element must have one child element named HeaderInfo with the namespace identifier of “http://calpers.ca.gov/PSR/CommonUtilitiesV1”

SOAP Body

- The soap:Body element of a **file inbound** to CalPERS must have one child element named RetirementHealthEnrollment with the namespace identifier of “http://calpers.ca.gov/PSR/RetirementHealthTransactionsV1”
- The soap:Body element of a **file outbound** from CalPERS must have one child element named RetirementHealthResponse with the namespace identifier of “http://calpers.ca.gov/PSR/RetirementHealthTransactionsV1”

XSD

- Inbound from Employers to CalPERS – RetirementHealthTransactionsV1.xsd
- Outbound from CalPERS to Employers – RetirementHealthResponseV1.xsd

Unit 3: File Testing & the Test Environment

Introduction

This document describes the considerations and activities for preparing and testing your files. The testing process will allow you to validate your files, data, and processes for submitting to and receiving data from CalPERS.

Test Environment

CalPERS provides a test environment for employers to validate their file structure, conformance to the schema definitions, and validate data contained in the file to ensure it meets the myCalPERS business rules and requirements for file sharing.

The test environment can be accessed on the Technical Resources page in the *File Readiness Testing* section.

Important Considerations

The data contained in the myCalPERS test and production environments is **confidential** and should be treated accordingly.

Business partners' security practices, policies, and procedures should apply and be considered when designating a System Access Administrator (SAA) or assigning user(s) access privileges to myCalPERS. Refer to the System Access Administration Student Guide for more information.

The *File Testing Readiness Checklist* at the end of this unit is included so that you can monitor your progress throughout these activities.

Contents

File Testing Approach 35

Preparing Test Files 36

Suggested Test Scenarios 37

General Notes on Error Correction 38

File Testing Readiness Checklist 39

File Testing Approach

Overview

Prior to testing, consider having a strategy to prepare for testing scenarios and plan for appropriate resources you may need.

Recommendation: Begin testing with files consisting of fewer records, slowly building up to larger files and more complex transactions and/or scenarios.

Testing Resources

Based on the number of test files and transactions you plan to test, you will need to ensure you have the appropriate testing resources identified and scheduled to participate. This may include:

- Appropriate IT staff available to create and/or submit test files
- Program staff to help submit test files and resolve business rule errors
- Any third-party vendors to coordinate secure resources throughout the process
- Identification of payroll and retirement enrollment transactions you are interested in testing based on how your organization typically conducts business with CalPERS

For example:
When planning to test Payroll files, determine what report types (e.g., Payroll Earned Period or an Adjustment Report), record types (e.g., Payroll Record or Service Credit Purchase), and transaction types (e.g., Earned Period Report, Prior Period Adjustment, etc.) need to be tested. Then, you will know how many different test files may need to be created.

Test File Submission

After validating your XML file structure is correct, submit a test file with data resembling your standard day-to-day functions to ensure the file meets the new data requirements.

For example:
First test a payroll file with five participants and run it through myCalPERS. Resolve any errors and run through the process again. Once you confirm you've met both file structure and data requirements, submit a test file with all participants and repeat the process.

Preparing Test Files

Test Report Indicator

The Test Report indicator options are “true” and “false.”

When preparing to submit files for testing, it is important that you set the Test Report indicator of your files to “true” to prevent records from posting. This is useful for testing because you will see all the errors generated without actually posting the records to the system. You can delete this file and fix the errors on your source system and resubmit the file until you are satisfied with the processing results.

Once you are ready to submit the files for posting, you can then change the Test Report indicator to “false.”

File Name

After you have tested the file and determined you are ready to submit it for posting, you will first need to change the file name to a unique name before the system will allow you to submit the file again.

For example:
Test report = 20170714254612_000_1006.xml
Non-Test report = 20170714254612_111_1006.xml

Note: You can change one digit of the timestamp in the filename to make it unique.

Special Characters

myCalPERS code strips away special characters from the data that you submit.

For example:
Hyphens, apostrophes, etc., which are common in name fields (e.g., Reeves-Wong or O’Malley) will not cause a problem in the system. The field data will “match.”

Spacing

myCalPERS code does NOT strip out extra spaces in file fields. Therefore, any differences in spaces between CalPERS production and business partner native systems will error out.

For example:
McGeorge will not match Mc George

Suggested Test Scenarios

Introduction

Below are scenarios of typical transactions our business partners submit to CalPERS. The type of transactions you submit will be relative to the contract(s) you have with CalPERS. The scenarios below cover a variety of report types, record types, and transaction types but are not meant to be a complete list of what you may need or may want to test. You may select those that are appropriate for your contract(s) and represent typical scenarios you might encounter.

Payroll Contributions

- Earned period
- Supplemental Income Plan (SIP and/or 457)
- Service credit purchase
- Prior period adjustments
- Earned period, no contribution, no service (retired annuitant)
- Retroactive salary adjustment (covering multiple pay periods)

Retirement Enrollment

- New enrollment
- Begin/End Leave
- Correct Event
- Permanent Separation

Health Enrollment

- New enrollment
- Cancel coverage
- Open enrollment
- Add dependent
- Delete dependent
- Change dependent address

General Notes on Error Correction

Monitoring File Status

After uploading your file successfully, the *File Upload History* section will display the status of Accepted and indicate the number of records that were validated and the count of those that were found to have errors. You can then select the *View Preprocessing Areas* link to access the results of an accepted file. Refer to Figure 3.1 for an example of where to find these links.

You will not receive any automated notification that your file has processed successfully during testing. You will need to log in to the system to determine the processing status of your test files.

SFTP Versus File Upload Errors

Files submitted via SFTP will generate a response file if there are no Level 1 errors in the file.

Files submitted using File Upload may generate errors, but you will need to log in to the system to review them. No response file will be generated for files submitted by File Upload within mylCalPERS.

Payroll Error Corrections

If you submit a payroll file via File Upload or SFTP and the test report indicator in the file is set to “true”, you will be able to review your errors in the online system and determine the changes that are required to fix the file so that it will post.

The system will perform the validations on the fixed records in the test report, but it will not post those records. You will need to fix the errors in the file and set the test report indicator to “false” and resubmit the file to post the records.

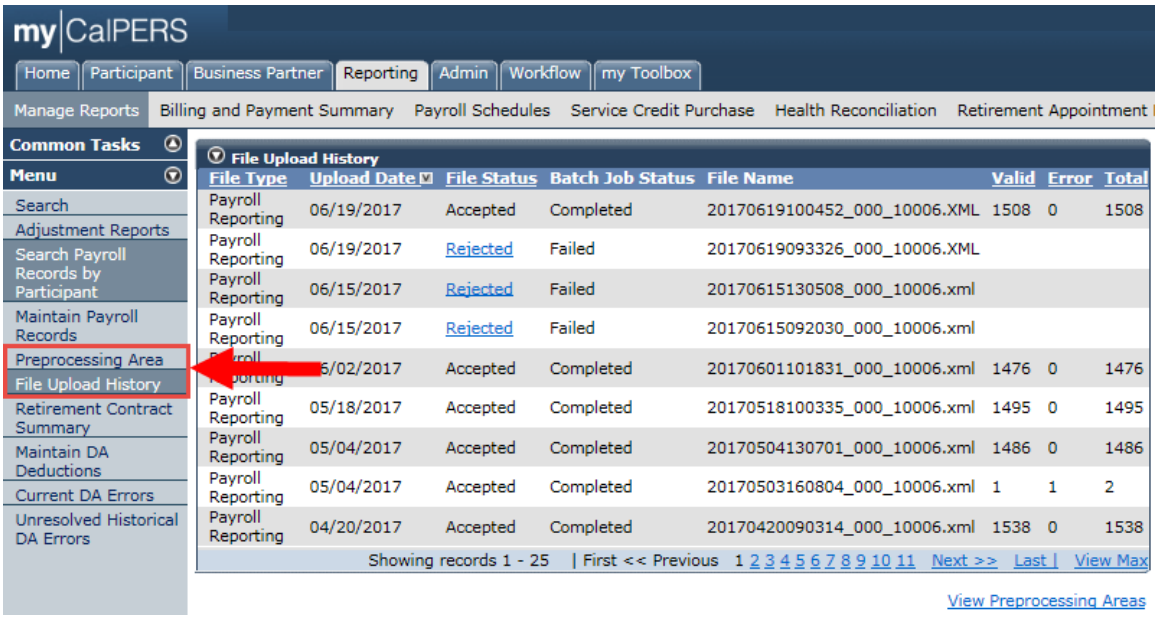
Health and Retirement Enrollment Error Corrections

When you submit a health or retirement enrollment file using SFTP or File Upload, all records without errors will automatically post to the system. You can correct the records with errors in your system and resubmit a file to post these records again or you can correct them in the preprocessing area and process the report using the online system.

File Testing Readiness Checklist

Step	Action	Complete
1	If submitting files using SFTP, contact CalPERS to ensure SFTP access has been established.	
2	Develop your XML files.	
3	Coordinate with your third-party vendor.	
4	Plan your test resources.	
5	Download required data from mylCalPERS via the Cognos report utility. **Note:** A list of available reports can be found on the mylCalPERS Employer Reports (Cognos) Web page.	
6	SAA registers in the registration environment.	
7	SAA creates test users in the test environment.	
8	Submit your XML files to myCalPERS using your chosen method: File Upload or SFTP.	
10	Check the "Accepted" files for Level 2 errors and correct as needed, either via the online process or correct in the file and resubmit.	
11	Post correct/corrected records to myCalPERS.	

Figure 3.1: Monitoring Status Links



Technical Support

File Testing Troubleshooting

Should you have difficulty during the file testing process, reference the Technical Resources page for information that may be helpful in resolving your issue.

Also consider these additional tips:

- Verify that test data submitted in the file corresponds with the data date, and those participants who had active status in your organization at that time.
- Ensure the correct values from the seed data are included in the file.
- Ensure that the file submitted has a unique file name (one that you have not submitted previously).

Resources

There are many resources available on the CalPERS website to help developers with understanding XML and XML development tools, schemas and related topics.

Key references include:

- Frequently Asked Questions (FAQ)
- Employer Technical Resources
- Payroll Level 1 Error Validations (PDF)

Contact

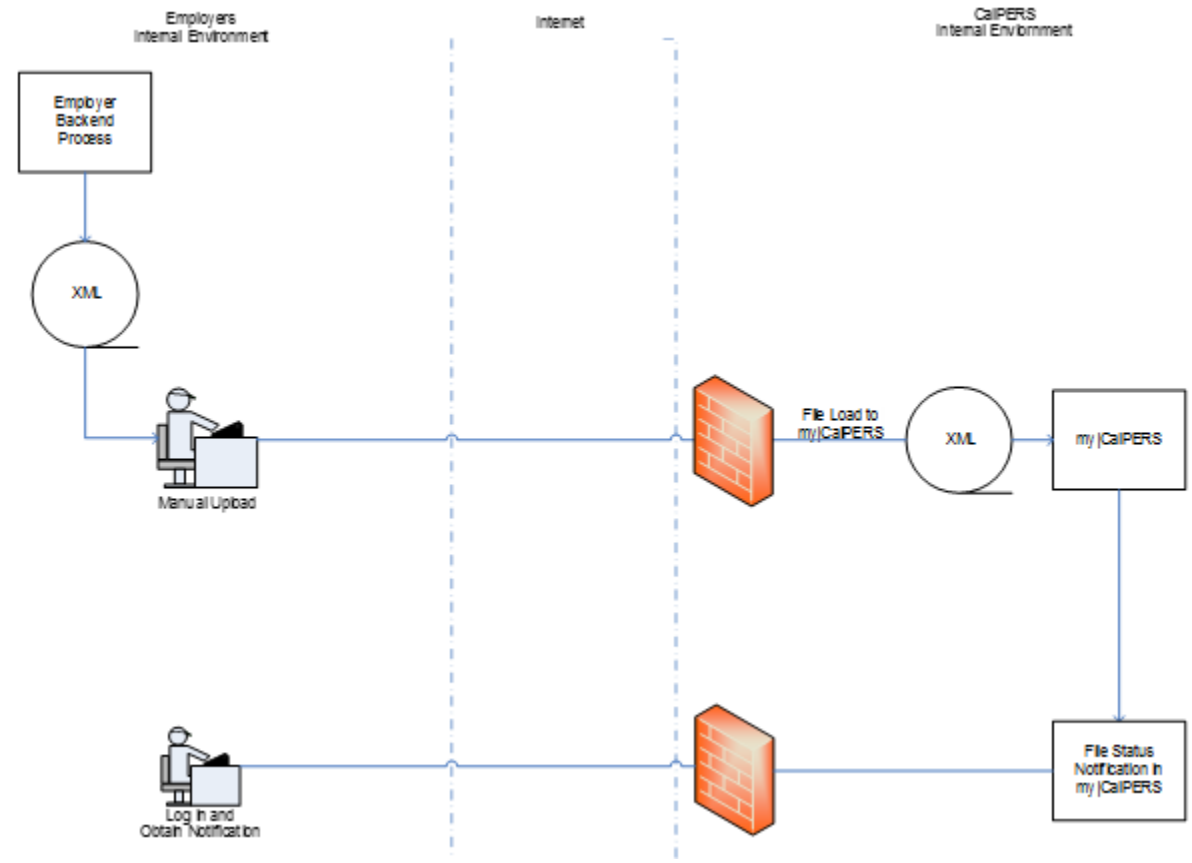
For assistance with the CalPERS Technical Requirements, Business Partners can contact the Employer Technical Support Team via this email or call the Employer Contact Center (ECC) at **888** CalPERS (or **888**-225-7377).

Helpful Information to Have When You Call

- Your organization's full name and CalPERS ID
- If you are having trouble with a file, have the interface ID available:
    - 10006 for payroll contributions
    - 00007 for retirement enrollment
    - 50031 for health enrollment
- For inquiries regarding an error message—have the error code and text available, as well as the actions taken to generate the error.
- For inquiries regarding the status of a previously reported issue—have the past Call Reference Number available.

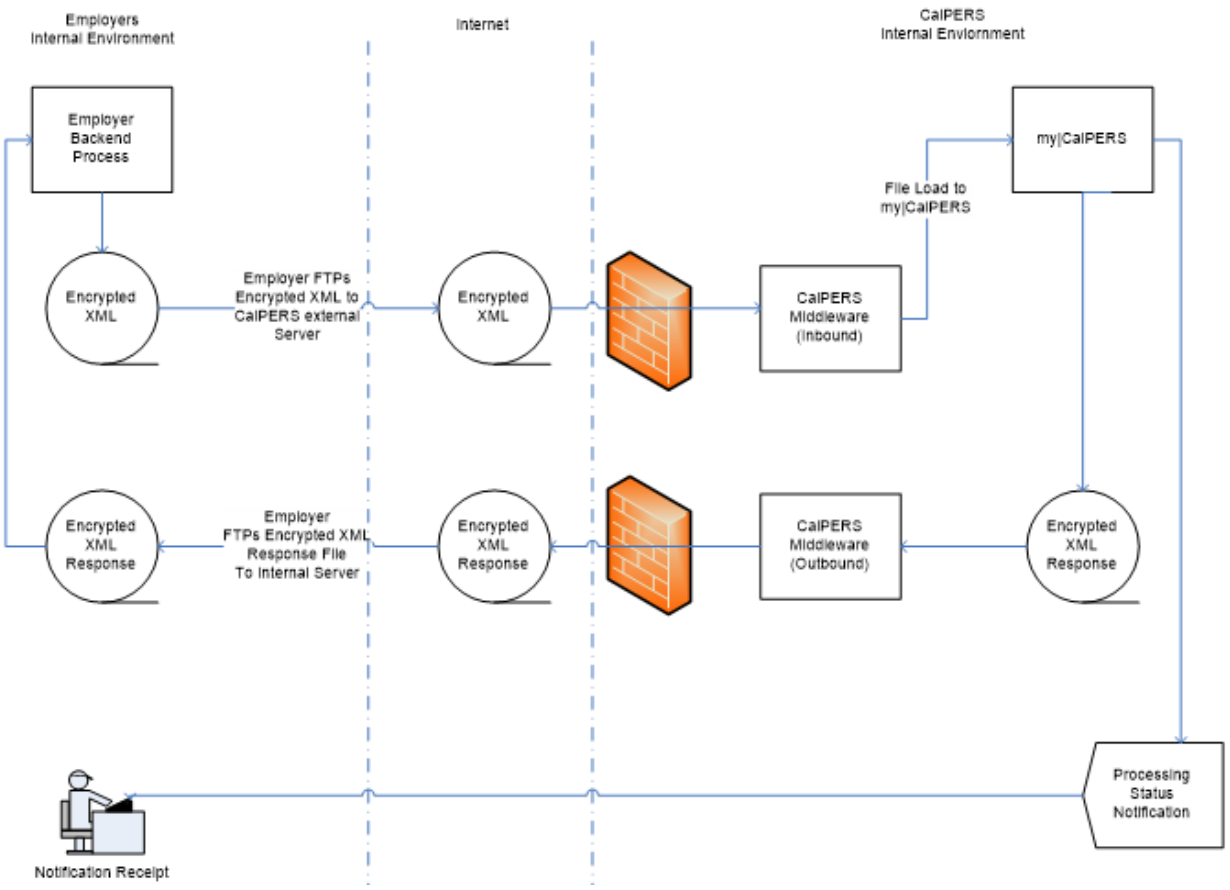
Appendix 1

File Upload Overview



Appendix 2

FTP Overview



Appendix 3

File Schema Relationship Overview