Location Intelligence
Infrastructure Asset Management

Confirm®

Financial Interface XML Agent Specification

v21.20b.AM

Open Source Attribution Notice

The Confirm suite of products contain the following open source software:

- Feature Data Objects v 3.5.0, which is licensed under GNU Lesser General Public License, Version 2.1, February 1999 with the unRAR restriction. The license can be downloaded from: **http://fdo.osgeo.org/licenceAndGovernance.html**. The source code for this software is available from **http://fdo.osgeo.org/content/fdo-350-downloads**
- MrSID software (specifically the mrsid32.dll) is used under license and is Copyright © 1995-2002, LizardTech, Inc., 1008 Western Ave., Suite 200, Seattle, WA 98104. All rights reserved. MrSID is protected by U.S. Patent No. 5,710,835. Foreign patents are pending. Unauthorized use or duplication prohibited.

  Patented technology in the Software was developed in part through a project at the Los Alamos National Laboratory, funded by the U.S. Government and managed by the University of California. The U.S. Government has reserved rights in the technology, including a non-exclusive, nontransferable, irrevocable, paid-up license to practice or have practiced throughout the world, for or on behalf of the United States, inventions covered by the patent, and has other rights under 35 U.S.C. § 200-212 and applicable implementing regulations.

  For further information, contact Lizardtech.

- NodaTime, version number 1.3.10, which is licensed under the Apache license, version number 2.0. The license can be downloaded from http://www.apache.org/licenses/LICENSE-2.0 . The source code for this software is available from http://nodatime.org/.
- Chromium Embedded Framework, version 3, which is licensed under the New BSD License. The license can be downloaded from http://opensource.org/licenses/BSD-3-Clause. The source code for this software is available from http://code.google.com/p/chromiumembedded/downloads/list.
- Xilium.CefGlue, version 3, which is licensed under the MIT License (with portions licensed under the New BSD License). The licenses can be downloaded from http://opensource.org/licenses/MIT and http://opensource.org/licenses/BSD-3-Clause. The source code for this software is available from http://xilium.bitbucket.org/cefglue/.
- D3 Data Driven Documentation, version 3.4.1, which is licensed under the New BSD License. The license can be downloaded from from https://github.com/mbostock/d3/blob/master/LICENSE. The source code for this software is available from http://d3js.org/.
- OpenLayers, version 2.12, which is licensed under the Modified BSD License. The license can be downloaded from http://svn.openlayers.org/trunk/openlayers/license.txt. The source code for this software is available from http://trac.osgeo.org/openlayers/browser.
- OpenLayers, version 3, which is licensed under the BSD 2-Clause Licence. The license which can be downloaded from https://github.com/openlayers/ol3/blob/master/LICENSE.md. The source code for this software is available from https://github.com/openlayers/ol3.
- Proj4js, version 1+, which is licensed under the Apache License, Version 2, January 2004. The license can be downloaded from http://www.apache.org/licenses/LICENSE-2.0.html. The source code for this software is available from http://trac.osgeo.org/proj4js/.
- requireJS, version 2.1.2, which is licensed under the MIT License or the New BSD License. The license can be downloaded from https://github.com/jrburke/requirejs/blob/master/LICENSE. The source code for this software is available from http://requirejs.org/.

- Apache Cordova, version 8.1.2, which is licensed under the Apache License, Version 2, January 2004. The license can be downloaded from http://www.apache.org/licenses/LICENSE-2.0.html. The source code for this software is available from http://phonegap.com/download/.
- Xilium.CefGlue, version 75.1, which is unlicensed. The source code for this software is available from https://gitlab.com/xiliumhq/chromiumembedded/cefglue.
- Chromium Embedded Framework, version 75.0, which is licensed according to the following criteria:

  Copyright (c) 2008-2014 Marshall A. Greenblatt. Portions Copyright (c) 2006-2009 Google Inc. All rights reserved.

  Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

  * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

  * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

  * Neither the name of Google Inc. nor the name Chromium Embedded Framework nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

  THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

  The source code for this software is available from http://opensource.spotify.com/cefbuilds/index.html#

July 06, 2021

Contents

Specifications

Financial Interface XML Agent 6

- Overview 6
- Format of the XML 6
- Import Format 14

Specifications

The following sections outline all the Specifications that exist within the Confirm functionality.

In this section

Financial Interface XML Agent 6

Financial Interface XML Agent

Introduction

The Confirm Financial XML Agent, is a plug-in module for the Confirm Task Processor. Its functionality differs from that of the current Financial Message Transfer Agent (MTA) in that it may export information from Confirm in XML or HTML as well as a text format.

This document aims to provide a full specification of the necessary XML file format.

Reference Documents

XML Schema Part 2: Datatypes. W3C Recommendation 02 May 2001

Overview

Scope

The Financial XML Agent extracts information from Confirm about works orders, variation orders and payments, creates an XML file, transforms this using an XSL stylesheet and exports the newly created file.

The interface is designed to run automatically, at pre-defined intervals from the Confirm Task Processor, but may also be initiated manually from the Scheduled Task window in the Confirm Client.

Format of the XML

OVERALL STRUCTURE AND FORMAT

The data is incorporated into an XML file that uses the tree structure and elements described in the following tables.

FILE NAMING AND DIRECTORIES

The Agent may be configured to poll the Confirm database at set intervals defined by the user.

The Agent generates one file for each Confirm Data Set for which it is configured. The path name of the target file is configurable within the software and existing files are not overwritten.

SIMPLE DATA TYPES

The simple XML schema data types use definitions given by W3C (**www.w3.org**).

Data Type	Format	Example	Comments
boolean	true, false, 1, 0	true	
date-Time	CCYY-MM-DDThh:mm:ss	2002-01-31T14:45:34	
date	CCYY-MM-DD	2002-01-31	
decimal	p,s	a 6,2 decimal could be up to 9999.99	Positive and negative numbers with decimal point if appropriate. Size shown in the form p,s where p = precision and s = scale
string	character value	High Street	Max size shows maximum length

COMPLEX DATA TYPES

The structure of the schema is relatively flat and complex data types only exist where there are repeating groups. The composition of these complex data types is given in the next section entitled 'Defintion of Element Types'.

DEFINITION OF ELEMENT TYPES

In the following tables the “No. of” column indicates whether the element is mandatory and whether repeating groups may occur.

financialInterface

Element	Type	No. of	Max Size	Confirm Screen	Confirm Label
batchDate	dateTime	1		Ledger Batch Lookup	Date
todaysDate	dateTime	1		n/a	n/a
batchNumber	decimal	1	6,0	Ledger Batch Lookup	Batch No.
dataSetID	string	1	10	System Registration Details	Data Set ID
worksOrder	**worksOrder**	0-n		Order Header	
variationOrder	**vari-ationOrder**	0-n		Variation Header	
paymentBatch	**payment-Batch**	0-n		Payment Batch	

worksOrder

Element	Type	No. of	Max Size	Confirm Screen	Confirm Label
contractCode	string	1	6	Order Header	Contract Code
contractName	string	1	30	Contract	Name

Element	Type	No. of	Max Size	Confirm Screen	Confirm Label
creditorCostCode	string	0-1	15	Contract - Payments	Cost Code
vendorId	string	0-1	20	Contractor – Additional	Vendor ID
vendorDivisionId	string	0-1	20	Contractor - Additional	Vendor Division
contractExternalFIS	string	0-1	30	Cost Code	External FIS
orderNotes	string	0-1	2000	Order Header	Notes
orderNumber	decimal	1	8,0	Order Header	Order Number
commitDate	dateTime	1		n/a	n/a
userLoginSurname	string	0-1	20	User Security	Surname
workTypeCode	string	1	4	Work Type	Code
priorityCode	string	0-1	4	Priority	Code (from Work Type)
orderDate	dateTime	1		Order Header	Date/Time
commitUserId	string	0-1	50	User Security	User ID
contractorCode	string	1	6	Contractor	Code
orderDetail	**orderDetail**	0-n		Order Job	
orderDetailRoutine	**orderDetail-Routine**	0-n		Routine Maintenance – Order Header	

orderDetail

Element	Type	No. of	Max Size	Confirm Screen	Confirm Label
jobNumber	decimal	1	8	Order Job	Job Number
costCodeCode	string	1	15	Order Item	Cost Code
externalFIS	string	1	30	Cost Code	External FIS
costCodeCatCode	string	1	4	Cost Code Category	Code
jobDescription	string	0-1	2000	Order Job	Description
jobLocation	string	0-1	2000	Order Job	Location
siteCode	string	1	10	Order Job	Site
siteName	string	1	40	Site	Name
areaName	string	1	30	Area	Name
plotNumber	decimal	1	10,2	Order Job	Plot Number
featureId	string	0-1	15	Order Job	Feature ID
featureLocation	string	0-1	2000	Order Job	Locn
estCompletionDate	dateTime	0-1		Order Job	Est Completion
orderValue	decimal	1	10,2	Order Header	Order Value
officerName	string	1	30	Order Job	Allocated To
jobPriorityCode	string	1	4	Priority	Code (from Job)

Element	Type	No. of	Max Size	Confirm Screen	Confirm Label
jobStartDate	dateTime	0-1		Job	Estimated Start Date
budgetCode	string	1	15	Cost Code	Budget Code
budgetExternalFIS	string	1	30	Cost Code	External FIS
budgetCatCode	string	1	4	Cost Code Category	Code
jobTypeCode	string	1	4	Job	Job Type

orderDetailRoutine

Element	Type	No. of	Max Size	Confirm Screen	Confirm Label
costCodeCode	string	1	15	Routine maintenance - Order Job	Cost Code
externalFIS	string	1	30	Cost Code	External FIS
costCodeCatCode	string	1	4	Cost Code Category	Code
regimeCode	string	1	6	Routine Maintenance – Order Job	Regime Code
regimeName	string	1	50	Maintenance regime	Description
siteCode	string	1	10	Routine Maintenance - Order Job	Site
siteName	string	1	40	Site	Name
areaName	string	1	30	Area	Name
plotNumber	decimal	0-1	10,2	Routine Maintenance - Order Job	Plot Number
featureId	string	0-1	15	Routine Maintenance - Order Job	Feature ID
featureLocation	string	0-1	2000	Routine Maintenance - Order Job	Locn
orderValue	decimal	1	10,2	Routine Maintenance – Order Job	Value

variationOrder

Element	Type	No. of	Max Size	Confirm Screen	Confirm Label
contractCode	string	1	6	Variation Order	Contract Code
contractName	string	1	30	Contract	Name
creditorCostCode	string	0-1	15	Contract – Payments	Cost Code
vendorId	string	0-1	20	Contractor – Additional	Vendor ID
vendorDivisionId	string	0-1	20	Contractor - Additional	Vendor Division
variationNumber	decimal	1	8,0	Variation Header	Variation Number

Element	Type	No. of	Max Size	Confirm Screen	Confirm Label
commitDate	dateTime	1		n/a	n/a
workTypeCode	string	1	4	Work Type	Code (from Variation Header)
priorityCode	string	0-1	4	Priority	Code (from Work Type)
contractExternalFIS	string	0-1	30	Cost Code	External FIS
variationDate	dateTime	0-1		Variation Order	Date/Time
commitUserId	string	0-1	50	User Security	User ID
contractorCode	string	1	6	Contractor	Code
variationDetail	**variationDetail**	0-n		Variation Job	
variationDetailRoutine	**variationDetailRoutine**	0-n		Routine maintenance - Variation Job	

variationDetail

Element	Type	No. of	Max Size	Confirm Screen	Confirm Label
orderNumber	decimal	1	8,0	Variation Job	Order Number
jobNumber	decimal	1	8	Variation Job	Job Number
costCodeCode	string	0-1	15	Variation Item	Cost Code
jobDescription	string	0-1	2000	Variation Job	Description
siteCode	string	1	10	Variation Job	Site
siteName	string	1	40	Site	Name
plotNumber	decimal	1	10,2	Variation Job	Plot Number
featureId	string	0-1	15	Variation Job	Feature ID
variationValue	decimal	1	10,2	Variation Header	Variation Value
jobPriorityCode	string	1	4	Priority	Code (from Job)
externalFIS	string	1	30	Cost Code	External FIS
costCodeCatCode	string	1	4	Cost Code Category	Code
jobStartDate	dateTime	0-1		Job	Estimated Start Date
budgetCode	string	1	15	Cost Code	Budget Code
budgetExternalFIS	string	1	30	Cost Code	External FIS
budgetCatCode	string	1	4	Cost Code Category	Code
jobTypeCode	string	1	4	Job	Job Type

variationDetailRoutine

Element	Type	No. of	Max Size	Confirm Screen	Confirm Label
orderNumber	decimal	1	8,0	Routine maintenance - Variation Job	Order Number
costCodeCode	string	0-1	15	Routine maintenance - Variation Item	Cost Code
regimeCode	string	1	6	Routine Maintenance – Order Job	Regime Code
regimeName	string	1	50	Maintenance regime	Description
siteCode	string	1	10	Routine maintenance - Variation Job	Site
siteName	string	1	40	Site	Name
plotNumber	decimal	0-1	10,2	Routine maintenance - Variation Job	Plot Number
featureId	string	0-1	15	Routine maintenance - Variation Job	Feature ID
variationValue	decimal	1	10,2	Variation Header	Variation Value
externalFIS	string	1	30	Cost Code	External FIS
costCodeCatCode	string	1	4	Cost Code Category	Code

paymentBatch

Element	Type	No. of	Max Size	Confirm Screen	Confirm Label
contractCode	string	1	6	Payment Batch	Contract Code
contractName	string	1	30	Contract	Name
creditorCostCode	string	0-1	15	Contract - Payments	Cost Code
contractorCode	string	1	6	Contractor	Code
vendorId	string	0-1	20	Contractor – Additional	Vendor ID
vendorDivisionId	decimal	0-1	8,0	Contractor - Additional	Vendor Division
contractExternalFIS	string	0-1	30	Cost Code	External FIS
paymentText	string	0-1	2000	Payment Batch	Payment Text
paymentNumber	decimal	1	10	Payment Batch	Payment Number
ledgerBatchNumber	decimal	1	8	Assigned by Agent	Ledger Batch
commitDate	dateTime	1		Assigned by Payment Commit	Payment Commit Date/Time
commitLogin	string	1	10	Assigned by Payment Commit	Payment Commit Login
finYearStart	date	0-1		Finance Year	Start

Element	Type	No. of	Max Size	Confirm Screen	Confirm Label
finYearEnd	date	0-1		Finance Year	End
taxStatusCode	string	1	4	Contractor	Tax Status
commitUserId	string	0-1	50	User Security	User ID
invoiceNumber	string	0-1	20	Payment Batch	Invoice Number
invoiceReceivedDate	dateTime	0-1		Payment Batch	Received
paymentDetail	**paymentDetail**	0-n			
routinePaymentDetail	**paymentDetailRoutine**	0-n			

paymentDetail

Element	Type	No. of	Max Size	Confirm Screen	Confirm Label
orderNumber	decimal	1	8,0	Payment Job	Order No
purchaseOrderNumber	string	0-1	15	Assigned when authorisation imported	PO Number
orderNotes	string	0-1	2000	Order Header	Notes
orderCommitDate	dateTime	0-1		n/a	n/a
orderExternalRef	String	0-1	20	Order Header	External Ref.
jobNumber	decimal	1	8	Payment Job	Job Number
costCodeCode	string	0-1	15	Cost Code	Cost Code
externalFIS	string	1	30	Cost Code	External FIS
costCodeCatCode	string	1	4	Cost Code Category	Code
jobDescription	string	0-1	2000	Payment Job	Description
estCompletionDate	dateTime	0-1		Job, Additional	Est. Completion (terminology set in Job System Settings)
siteCode	string	1	10	Payment Job	Site
siteName	string	1	40	Site	Name
plotNumber	decimal	1	10,2	Payment Job	Plot No
featureId	string	0-1	15	Payment Job	Feature ID
taxPercentage	decimal	1	5,2	Tax Rate	Percentage
taxValue *1	decimal	1	10,2	Payment Item	Job Payment Tax Total
taxTypeCode	string	1	4	Tax Type	Code
jobClose	boolean	1		Payment Job	Job Complete
jobCustRef	String	0-1	20	Job	Customer Reference
requestRef	string	0-1	20	Payment Request Header	Request Ref.

Element	Type	No. of	Max Size	Confirm Screen	Confirm Label
requestText	string	0-1	2000	Payment Request Header	Text
requestDate	dateTime	0-1		Payment Request Header	Request Date
requestLogDate	dateTime	0-1		Payment Request Header, Status Log	Log Date where Log No. is 1
paymentDate	dateTime	1		Payment Batch	Payment Date
periodEndDate	dateTime	1		Finance Period	End
paymentValue	decimal	1	12,2	Payment Item	This Payment
orderOriginalValue	decimal	1	12,2	Order Item	Original Order Value
priorityCode	string	0-1	4	Priority	Code (from Work Type)
workTypeCode	string	1	4	Work Type	Code (from Order Header)
jobPriorityCode	string	1	4	Priority	Code (from Job)
requestClassCode	string	0-1	6	Request Classification	Code
requestClassName	string	0-1	50	Payment Request Header	Request Classification
jobStartDate	dateTime	0-1		Job	Estimated Start Date
budgetCode	string	1	15	Cost Code	Budget Code
budgetExternalFIS	string	1	30	Cost Code	External FIS
budgetCatCode	string	1	4	Cost Code Category	Code
jobTypeCode	string	1	4	Job	Job Type

*1 The taxValue includes the tax adjustment value. In Confirm this is on the Payment Item screen.

paymentDetailRoutine

Element	Type	No. of	Max Size	Confirm Screen	Confirm Label
orderNumber	decimal	1	8,0	Routine maintenance - Payment Job	Order No
purchaseOrderNumber	string	0-1	15	Assigned when author-isation imported	PO Number
orderNotes	string	0-1	2000	Routine maintenance - Order Header	Notes
orderCommitDate	dateTime	0-1		n/a	n/a
orderExternalRef	String	0-1	20	Routine maintenance - Order Header	External Ref.
costCodeCode	string	0-1	15	Cost Code	Cost Code
externalFIS	string	1	30	Cost Code	External FIS
costCodeCatCode	string	1	4	Cost Code Category	Code

Element	Type	No. of	Max Size	Confirm Screen	Confirm Label
regimeCode	string	1	6	Routine Maintenance – Payment Job	Regime Code
regimeName	string	1	50	Maintenance regime	Description
siteCode	string	1	10	Routine maintenance - Payment Job	Site
siteName	string	1	40	Site	Name
plotNumber	decimal	0-1	10,2	Routine maintenance - Payment Job	Plot No
featureId	string	0-1	15	Routine maintenance - Payment Job	Feature ID
taxPercentage	decimal	1	5,2	Tax Rate	Percentage
taxValue	decimal	1	10,2		
taxTypeCode	string	1	4	Tax Type	Code
paymentDate	dateTime	1		Payment Batch	Payment Date
periodEndDate	dateTime	1		Finance Period	End
paymentValue	decimal	1	12,2	Routine maintenance - Payment Item	This Payment
orderOriginalValue	decimal	1	12,2	Routine Maintenance - Order Item	Original Order Value
workTypeCode	string	1	4	Work Type	Code (from Order Header)

Import Format

FILE FORMAT

General

The format of the file will be comma delimited at field level with each record delimited by the appropriate new-line character(s).

Text qualifiers (“”) may be supplied if the data itself contains one of the delimiting characters.

Extensible Markup Language (XML) can also be used as the files format. The XML file has a XSL Stylesheet applied to it that converts the file into a composite comma format.

Delimited Files

Field names must be specified, i.e. the first line of each file should include the field names of those fields being supplied, as per the import definition for the entity concerned.

Composite Delimited Files

Each row contains several delimited items. The first is a number identifying the row type and the second identifies the Data Type. The file can contain any or all of the Data Types mentioned in the Data Type Descriptions section of this document (the format is called Composite Delimited as data of different types are contained in a single file). The different row types are described here.

The Header Definition row (type 1) contains the field names of all data being supplied, in the order in which it is supplied. There should be only one header for each Data Type.

A Data row (type 2) contains the actual values to go into the fields. The data should be in the order defined by the header record. If the file contains any data before the header record, an error will be reported.

The Record Count row (type 9) is optional. If supplied, this will contain a single numeric item, which is checked against the number of rows of this Data Type in the file. If there is a mismatch, it is assumed that the file is incomplete or otherwise invalid, and an error is reported. There should be no more than one record count for each Data Type.

Key Fields

Key fields must be unique where specified or the whole file will be rejected and an error generated. Generally alpha key fields will be capitalised.

Lookup Fields

Where fields in a file being imported reference the database as lookups, the appropriate lookup record must exist to maintain database integrity. These can usually be supplied as a code or as a name (the tables below will say “6 or 50” where something can be supplied as a 6 character code or a 50 character name).

Checks for these lookups are case sensitive. Where a lookup is referenced which does not already exist an error will be reported and the file rejected.

Field Details

Where “CAPS” is shown in the definition tables it means that the relevant field will have its contents converted to capitals during the import, regardless of the case of the supplied data.

- Numeric fields are specified in the format 6 (4.2), meaning 6 digits in total, 4 to the left of the decimal point, 2 digits to the right. e.g. 1234.56.
- To prevent problems when files are imported on a different system to the one where the files were prepared, fields specified as date should be supplied in the format DDMMYYYY or DD/MM/YYYY, except when importing XML files.
- All fields specified as time should be supplied in the format HH:MM, e.g. 18:30.
- The field lengths need not be as shown in the definition tables but any characters in excess of those specified will be truncated. A warning will be given if this happens.

Validation

This facility will accept a PO number allocated by a Financial System, and will accept multiple files for import.

If there are any syntactical errors in the file, such as invalid data types used or an invalid record count, then the entire file will be rejected.

Any semantic errors, such as an Authorisation being received for a non-existent Order, will cause the items in error to be ignored. All complete records that are acceptable will be updated even though other records in the same input file are not accepted.

The Task Processor will email the error log (see below) to a designated systems administrator if there are any problems with a file.

File Processing

Import Transformations are stored in Confirm detailing:

- The directory where the files to be imported are located.
- File mask to be applied to limit the files in the directory to be selected (e.g. *.* will cater for selecting all files in the specified directory.
- The field delimiter.
- The location and name of the stylesheet to be used to convert the files to the SBSCSV format detailed below.

Files successfully imported will be written to a “READ” sub-directory of the Import Directory with no warning given before overwriting existing files. If the sub-directory does not exist then an error / alert will be generated and the Financial Interface XML Agent Scheduled Task will stop.

Error Logging

The import will detect various errors in the Import data. When a file of data is imported an error file will be generated containing any errors found in the data, this will happen whether or not the update takes place. An indication of the line in the text file where the problem has been detected will also be included. This will allow location and correction of such errors.

The error file will take the same name as the imported file but will take the extension “.ERR”.

Where errors mean that the entire file cannot be read (e.g. incorrect format or inconsistent totals), then the original import file and error file will be placed in an “INVALID” sub-directory of the import directory. If the “INVALID” sub-directory does not exist, then an error / alert will be generated, and the Financial Interface XML Agent Scheduled Task will stop.

If the error(s) only applied to certain records in the file, but others may have been imported, then the error file is placed in the “READ” sub-directory with the main file.

FILE DEFINITION

This import facility caters for an external financial system issuing a Purchase Order number to be held against Works Orders.

Works Orders need to be in a pending state, i.e. Uncommitted with the Pending state flag set, in order to be valid for this process.

When a Works Order number is imported with a Purchase Order number, the Works Order will be deemed to be authorised, and will, therefore, be moved on to a Committed Status, and the Cost Code commitments will be updated.

When a Works Order number is imported with no Purchase Order number, the Works Order will be deemed to have failed the authorisation process, and will, therefore, have its Pending state flag reset, making it an amendable Uncommitted Works Order.

Works Order Authorisation

Field	Size	Type	Comment
Contract Code	6	ASCII / CAPS	Mandatory
Works Order Number	8 (8.0)	Numeric	Mandatory
Purchase Order Number	15	Alpha-Numeric	If blank, Work Order will be set to uncommitted

Composite File Example

```
1,FIXML,contract code,order number,po number
2,FIXML,C1,150,PO Number 150
2,FIXML,C1,255,PO Number 255
```