

1. DSD - Section 19	2
1.1 DSD 19/Payroll - Warrant Management	4
1.1.1 DSD 19/Payroll – Warrant Management/Warrant Fund and Contractor Homemaker Management Topic Area	5
1.1.2 DSD 19/Payroll – Warrant Management/Business Process	6
1.1.2.1 DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions	7
1.1.2.1.1 DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Create and Maintain Fund Information	8
1.1.2.1.2 DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Vendor Maintenance	12
1.1.2.1.3 DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Warrant Processing	16
1.1.2.1.4 DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Electronic Claim File	24
1.1.2.1.5 DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Daily Payroll Reconciliation	26
1.1.2.1.6 DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Reconciling of Tax Filing	29
1.1.2.1.7 DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Replacement and Reissue Warrants	32
1.1.2.1.8 DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Stop Payment and Redeposit Warrants	36
1.1.2.1.9 DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Stale Dated Warrants	40
1.1.2.1.10 DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Warrant/Funds Mgt Warrant Cross Reference	43
1.1.2.1.11 DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/County Contractor	46
1.1.2.1.12 DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Homemakers	47
1.1.2.1.13 DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/General Ledger	48
1.1.2.1.14 DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Funding Management	49
1.1.2.1.15 DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Fiscal Year Fund Maintenance	50
1.1.2.2 DSD 19/Payroll – Warrant Management/Business Process/Screen Designs	51
1.1.2.2.1 DSD 19/Payroll – Warrant Management/Business Process/Screen Designs/Financial Data Elements	52
1.1.2.3 DSD 19/Payroll – Warrant Management/Business Process/Navigation Elements	59
1.1.2.4 DSD 19/Payroll – Warrant Management/Business Process/Error Messages	60
1.1.2.5 DSD 19/Payroll – Warrant Management/Business Process/Business Rules	61
1.1.2.6 DSD 19/Payroll – Warrant Management/Business Process/Tasks/Notifications	62
1.1.2.7 DSD 19/Payroll – Warrant Management/Business Process/Internal Interfaces	63
1.1.2.7.1 DSD 19/Payroll – Warrant Management/Business Process/Internal Interfaces/Warrant Information Table Create	64
1.1.2.7.2 DSD 19/Payroll – Warrant Management/Business Process/Internal Interfaces/Warrant Information Table Update	71
1.1.2.7.3 DSD 19/Payroll – Warrant Management/Business Process/Internal Interfaces/Send Void and Reissue Request	73
1.1.2.7.4 DSD 19/Payroll – Warrant Management/Business Process/Internal Interfaces/County Contractor and Homemaker Journal Voucher	75
1.1.2.8 DSD 19/Payroll – Warrant Management/Business Process/External Interfaces	77
1.1.2.8.1 DSD 19/Payroll – Warrant Management/Business Process/External Interfaces/EFT Pre-Note – PRWS109B	78
1.1.2.8.2 DSD 19/Payroll – Warrant Management/Business Process/External Interfaces/Daily Payment – PRDS109A	84
1.1.2.8.3 DSD 19/Payroll – Warrant Management/Business Process/External Interfaces/SCO Warrant Number Assignment – PRDR109D and PRDR109H	97
1.1.2.8.4 DSD 19/Payroll – Warrant Management/Business Process/External Interfaces/Warrant Paid – PRDR110A	100
1.1.2.8.5 DSD 19/Payroll – Warrant Management/Business Process/External Interfaces/Undeliverable Warrants – PRDR111A	102
1.1.2.8.6 DSD 19/Payroll – Warrant Management/Business Process/External Interfaces/Vendor Listing to SCO – PRMS109I	106
1.1.2.9 DSD 19/Payroll – Warrant Management/Business Process/Batch Processing	108
1.1.2.10 DSD 19/Payroll – Warrant Management/Business Process/Applicable Security Roles	113
1.1.2.11 DSD 19/Payroll – Warrant Management/Business Process/Reporting	114
1.1.2.12 DSD 19/Payroll – Warrant Management/Business Process/Forms	115

DSD - Section 19



CMIPS

D-4.2-03 – IHSS CMIPS Detailed System Design (DSD) (R2025.03.01) Section 19

Version 1.0

03/28/2025

Table of Contents

- DSD 19/Payroll - Warrant Management
 - DSD 19/Payroll – Warrant Management/Warrant Fund and Contractor Homemaker Management Topic Area
 - DSD 19/Payroll – Warrant Management/Business Process
 - DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions
 - DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Create and Maintain Fund Information
 - DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Vendor Maintenance
 - DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Warrant Processing
 - DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Electronic Claim File
 - DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Daily Payroll Reconciliation
 - DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Reconciling of Tax Filing
 - DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Replacement and Reissue Warrants
 - DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Stop Payment and Redeposit Warrants
 - DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Stale Dated Warrants
 - DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Warrant/Funds Mgt Warrant Cross Reference
 - DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/County Contractor
 - DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Homemakers

- DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/General Ledger
- DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Funding Management
- DSD 19/Payroll – Warrant Management/Business Process/Business Process Functions/Fiscal Year Fund Maintenance
- DSD 19/Payroll – Warrant Management/Business Process/Screen Designs
 - DSD 19/Payroll – Warrant Management/Business Process/Screen Designs/Financial Data Elements
- DSD 19/Payroll – Warrant Management/Business Process/Navigation Elements
- DSD 19/Payroll – Warrant Management/Business Process/Error Messages
- DSD 19/Payroll – Warrant Management/Business Process/Business Rules
- DSD 19/Payroll – Warrant Management/Business Process/Tasks/Notifications
- DSD 19/Payroll – Warrant Management/Business Process/Internal Interfaces
 - DSD 19/Payroll – Warrant Management/Business Process/Internal Interfaces/Warrant Information Table Create
 - DSD 19/Payroll – Warrant Management/Business Process/Internal Interfaces/Warrant Information Table Update
 - DSD 19/Payroll – Warrant Management/Business Process/Internal Interfaces/Send Void and Reissue Request
 - DSD 19/Payroll – Warrant Management/Business Process/Internal Interfaces/County Contractor and Homemaker Journal Voucher
- DSD 19/Payroll – Warrant Management/Business Process/External Interfaces
 - DSD 19/Payroll – Warrant Management/Business Process/External Interfaces/EFT Pre-Note – PRWS109B
 - DSD 19/Payroll – Warrant Management/Business Process/External Interfaces/Daily Payment – PRDS109A
 - DSD 19/Payroll – Warrant Management/Business Process/External Interfaces/SCO Warrant Number Assignment – PRDR109D and PRDR109H
 - DSD 19/Payroll – Warrant Management/Business Process/External Interfaces/Warrant Paid – PRDR110A
 - DSD 19/Payroll – Warrant Management/Business Process/External Interfaces/Undeliverable Warrants – PRDR111A
 - DSD 19/Payroll – Warrant Management/Business Process/External Interfaces/Vendor Listing to SCO – PRMS109I
- DSD 19/Payroll – Warrant Management/Business Process/Batch Processing
- DSD 19/Payroll – Warrant Management/Business Process/Applicable Security Roles
- DSD 19/Payroll – Warrant Management/Business Process/Reporting
- DSD 19/Payroll – Warrant Management/Business Process/Forms

DSD 19/Payroll - Warrant Management

This section will describe the CGI Advantage® Commercial-Off-The-Shelf (COTS) Human Resource Management (HRM) Payroll and Financial packages utilized to satisfy the payroll and fund source requirements.


DSD 19/Payroll – Warrant Management/Warrant Fund and Contractor Homemaker Management Topic Area

This document will provide the high-level view of CMIPS Payroll – Warrant/Fund Management. The CMIPS solution will utilize the CGI Advantage® COTS HRM Payroll and Financial packages to satisfy the payroll and fund source requirements.

DSD 19/Payroll – Warrant Management/Business Process

DSD 19/Payroll – Warrant Management/Business Process /Business Process Functions

DSD 19/Payroll – Warrant Management/Business Process /Business Process Functions/Create and Maintain Fund Information

CI	Document Name
 CI-69268 - DSD BF Create and Maintain Fund Information IMPLEMENTED	DSD_BF_Create_and_Maintain_Fund_Information.doc

The actual financial tables and batch processes that must be established and maintained for the CMIPS funding structure are identified in this section. These tables are shared between CMIPS HRM and Financial but are maintained in Financial. Also, the tables and batch processes are shown in the order that Financial component of CMIPS should be populated.

Begin Day

Begin Day is a daily batch job that is available to update the system date parameter through the batch interface. The system date parameter is defined as "APPL_SYS_DT" in the Application Parameters table. This batch job will be the first job that is run in the day to set the "APPL_SYS_DT" parameter value to the current system date. The result of this batch job is the Application Parameters table is updated with the current system date.

New Year Table Initialization (NYTI)

A yearly batch process, New Year Table Initialization (NYTI), exists to populate the tables with values for a coming year so the California Department of Social Services (CDSS) does not have to enter them manually unless they choose to modify them.

This batch process should be run to bring the day zero data up to the current fiscal year (FY) before configuration of the tables. This batch process is also used as part of the year-end process.

CDSS will run this process each year to establish the new FY tables. This process will be run yearly at some time before any activity begins against a new FY. This process will populate or delete records from tables, which have records that are unique by fiscal year or budget FY, so that these records do not have to be populated or deleted manually.

The result of this batch job is a NYTI report that shows the tables updated and list of errors. If the batch job is successful then new FY tables are established.

Fiscal Year (FY)

The Fiscal Year (FY) table is the first table in the configuration of the Financial component of CMIPS system. This table is populated when the NYTI process is run before the beginning of the next FY. CDSS should check the FY table to ensure the data is correct.

The FY table allows CDSS to establish and verify the valid FYs and budget FYs in Financial component of CMIPS. All transactions and processes must be executed within a valid year. Verification of the FY used on a transaction not only ensures a valid FY is entered on transactions but also ensures that the year is open for transactions.

Accounting Period (APD)

The Accounting Period (APD) table allows CDSS to establish, maintain and validate accounting periods used. This table is used to define the valid accounting periods for each FY. CDSS can also use this table to provide period-end, period closing and adjustment information that effects transaction processing. This table is populated when the NYTI process is run; however, CDSS should check the table to ensure the data is correct before beginning a new FY.

Calendar Date (CLDT)

The Calendar Date (CLDT) table allows CDSS to define every Calendar Date for a Fiscal Year. Information on this table is usually automatically updated when the Financial component of CMIPS is installed; however, CDSS can revise or update the information online (e.g. state holidays, federal bank holidays, etc.). CDSS must update the information on this table before beginning a new fiscal year. The Calendar Date table allows the Financial component of CMIPS to convert dates for reporting purposes and to validate dates entered on documents.

Contact (CNTAC)

The Contact (CNTAC) table data is used on the Bank table and during the disbursement cycle. There are two sections to the CNTAC table: General Information and Address. CDSS is not using the Address, so this section will not be populated. The General Information section must be configured manually.

Balance Sheet Type (BST)

The Balance Sheet Type (BST) table is used to capture additional detail for payroll and financial reporting. The balance sheet types were developed to meet payroll reporting needs which are used only in the financial component of the CMIPS system.

Balance Sheet Account (BSA)

The Balance Sheet Account (BSA) table is the first table in the configuration of the Fund Structure elements (chart of accounts). The balance sheet accounts are used on other configuration tables. There are three sections to the BSA table: General Information, Rollups and 1099 Info. CDSS is not using BSA rollups or 1099 Info, so these sections will not be populated. The General Information section must be configured manually.

The cash balance sheet account must be established in order to record payment information. Other balance sheet accounts that will be established are accounts payable, disbursements payable, warrant payable, state dated payable and escheat payable.

Bank (BANK)

The Bank (BANK) table must be configured before the Fund table, since the bank account is a required field on the Fund table. There are three sections to the Bank table: General Information, Disbursement Tracking, and Next Available Information. The General Information, Disbursement Tracking, and Next Available Information sections must be configured manually.

Fund (FUND)

The Fund (FUND) table has four sections: General Information, Year End Options, Rollups, and Description/Contact. Only two sections must be configured: General Information and Year End Options. Only one field in the Year End Options section will be populated and the other fields are not required for the financial component of CMIPS. The Rollups section and the Description/Contact section will not be used by CDSS. The General Information and Year End Options sections must be configured manually.

In CDSS current budget structure, the General Fund 0001 is used for CDSS programs payments and reimbursements.

Appropriation (APPR)

The Appropriation (APPR) table has two sections: General Information and Rollups. Only one section must be configured: General Information. The Rollups section will not be used by CDSS. The General Information section must be configured manually. In CMIPS, an appropriation refers to a set of budget lines, all governed by the same legislative authorization.

In the current budget structure, CDSS uses Reference 111, Program 25 and Element 15 to reference a budget line. In CMIPS, the Reference, Program and Element will be combined to create an appropriation unit (for example: 1112515).

Object Category (OCAT)

CDSS currently is only using object 706 to record CMIPS activities into CALSTARS. In order to capture additional detail for payroll and financial reporting, new objects were developed to meet payroll reporting needs which are used only in the financial component of the CMIPS system. The new objects were not established in the California State Accounting and Reporting System (CALSTARS). The new objects will roll up to CDSS' CALSTARS object 706. For example, wages are established with object code 5001 and will roll up to object category code 706.

Object (OBJ)

The Object (OBJ) table has four sections: General Information, General Options, 1099 Info/1042-S Info and Rollups. Only three sections must be configured manually (General Information, General Options and Rollups).

CDSS Accounting is only using object 706 for payroll reporting into CALSTARS. In order to capture additional detail for reporting, new objects were developed to meet payroll and financial reporting needs which are used only in the financial component of the CMIPS system. The new objects were not established in CALSTARS. The new objects will roll up to CDSS' CALSTARS object 706. The CMIPS object category 706 is equivalent to CDSS CALSTARS object.

For example, wages are established with the object code 5001, which will roll up to object category code 706 (CALSTARS object).

Government Branch (GOVBR)

The Government Branch (GOVBR) table represents the top of the organizational tree and usually reflects the responsible executive, legislative, or judicial body. The Government Branch is a required field on the Department Fiscal Year Controls table. A government branch place holder will be established for use on the Department Fiscal Year Controls table since it is a required field.

Cabinet (CAB)

The Cabinet (CAB) table is a portion of the government branch. The cabinet element rolls up to the government branch. The Cabinet is a required field on the Department Fiscal Year Controls table. A cabinet place holder will be established for use on the Department Fiscal Year Controls table since it is a required field.

Department (DEPT)

The Department (DEPT) table is used for establishing department controls that are not fiscal year based. The Department table has three sections: General Information, Buyers, and Description/Contacts. Only one section must be configured: General Information. The Buyer and Description/Contacts sections will not be used by CDSS. The General Information section must be configured manually.

In the CDSS current budget structure, the Organization 5180 is used to indicate CDSS. The Department 5180 in CMIPS will be the equivalent to CDSS' Organization 5180.

Unit (UNIT)

The Unit (UNIT) table has four sections: General Information, Rollups, Flexible Reporting and Buyer. Only one section must be configured: General Information and it will be manually configured. This table will display the CMIPS programs currently known as fund source and sub-sources. For example, IPW has six sub-sources, which would be displayed as IPW1, IPW2, IPW3, IPW4, IPW5, and IPW6 in CMIPS.

Sub Unit (SUNIT)

The Sub Unit (SUNIT) table is used to define the most detailed element in the organizational chart of accounts structure. CDSS needs to be able to attribute funding structure to a county and district office within the county. In CMIPS, the county and the district office will be combined to create a sub-unit. For example, Alameda County 01 and District Office 21 will be sub-unit 0121.

Reporting Code (RPT)

The Reporting Code (RPT) table has two sections which are: General Information and Rollups. Only one section must be configured: General Information. The General Information must be manually configured. The Rollups section will not be used by CDSS. This table will display the different types of reporting elements: County, State and Federal to help define the cost for each program.

Department Fiscal Year (DEPTFY)

The Department Fiscal Year (DEPTFY) table is to define controls and setting that can vary by fiscal year. This table must be established in order for a financial transaction to process.

Revenue Source (RVRS)

The Revenue Source (RSRC) table has five sections which are: General Information, Rollups, Revenue Options, Revenue Accounts and Applied Fee/WO Accounts. Only one section must be configured: General Information. The Rollups section, Revenue Options sections, Revenue Accounts section and the Applied Fee/WO Accounts section will not be used by the CDSS.

Vendor ABA (VABA)

This table allows CDSS to establish banking institutions where electronic funds transfer (EFT) data will be sent. The records on this table represent the bank's American Bankers Association (ABA) routing number. Only one record needs to be established on this table for each unique ABA number.

Appropriation Inference (APPRINF)

This table allows CDSS to define by budget fiscal year, fund, department, unit, and object code an appropriation code that will be inferred when they are used.

Disbursement Format (DISF)

The Disbursement Format (DISF) table is used to define each electronic or hard copy that may be produce by the disbursement process. This table contains the value indication the warrant format to be generated for disbursement. The Disbursement Format table has two sections: General Information and ACH Information. Only one section will be configured: General Information. The General Information section must be configured manually.

Disbursement Priority (DISP)

The Disbursement Priority (DISP) table lists all the available disbursement priorities. The Auto Disbursement process looks to the disbursement priority to determine the order in which to process payments. In addition to "regular payment," CDSS could decide to set up "critical payment" as a priority in case the State is experiencing cash restrictions. All payments will default to regular and can be overridden if necessary. The default value will be "99" for Regular Payment.

Disbursement Category (DISC)

The Disbursement Category (DISC) table is used to classify and/or consolidate payments.

System Options (SOPT)

The System Options (SOPT) table provides CDSS with a set of characteristics that guide the budget and accounting activities. Each control or option that is set on SOPT is associated with a business rule that drives events/actions throughout the system. However, those rules do not exist on SOPT. This table contains the collection of flags, rules, and controls that guide processes that read these options and controls as guidelines. The records on SOPT are unique by fiscal year. The table is divided into the following sections: Expense, General, Revenue, Fixed Assets and Internal Cost. CDSS will only use the Expense section, which must be configured manually.

Special Accounts (SPEC)

The Special Accounts (SPEC) table allows CDSS to define the default accounts and some special-purpose accounts by FY that are used throughout the financial component of CMIPS. In general, the accounts set on the Special Accounts table are the default accounts that will be posted for the event types that are defined. Records on SPEC are unique by FY in the financial component of CMIPS. The table is divided into the following sections: Accounts Payable, Purchasing and Procurement, Revenue, Miscellaneous, Fixed Assets and Internal Costing. CDSS will only use the Accounts Payable and Miscellaneous sections for CMIPS.

Auto Numbering (ADNT)

This table allows CDSS to set up the format for automatic document numbering for each Document Code, Year and Department combination that exists in CMIPS.

Extended Department (DEPTX)

The Extended Department (DEPTX) table is used to reduce data entry. This table is used to assign accounting elements and percentage splits to an Individual Provider (IP) in the Payroll component of CMIPS. The accounting information is then used to post the accounting information in the financial component of CMIPS.

Labor Distribution Profile (LDPR)

The Labor Distribution Profile (LDPR) table is used to set up accounting distributions to which pay resulting from events can be charged. The labor distribution profile code reduces data entry since a single code can be used in place of entering the full accounting distribution, and it also allows CMIPS to distribute an expense between different accounting lines.

Budget Inquiry Tables

The Budget Inquiry tables list all budget lines that have been established for the CMIPS budget structure and level that is indicated in the page caption on the inquiry. The Chart of Accounts elements and name of each budget are displayed and searchable. The budget inquiry tables also display budget tracking amount fields associated with the selected budget as well as non-budgetary information that pertain to the selected budget. The fields displayed on these tables were defined during implementation of the financial component of CMIPS.

For each standalone budget tracking amount field (those directly updated by accounting and budget documents) you can access field details and transaction details. The field details show the actual bucket amount and totals for how much is in pending for the bucket for both increases and decreases. This is done by a mouse click directly on the label for the standalone bucket. The icon to the right of the bucket opens a window where documents, which have updated the budget bucket, are listed. Each record has a transition to allow a document to be opened. The Download action found just below the grid of documents will create a Microsoft Excel file of the documents for the budget amount being investigated. That file can then be saved locally with a name provided by the system user. When the Detailed Transaction Listing window is closed, a user is returned to the transaction details.

Business Process Flow - Create and Maintain Fund Information

This section is not applicable for Warrant Management.

Initiation/Triggers, Pre-Conditions and Post Conditions

This section is not applicable for Warrant Management.

Process/Screen Flow

This section is not applicable for Warrant Management.

DSD 19/Payroll – Warrant Management/Business Process /Business Process Functions/Vendor Maintenance

Create Vendor

CI	Document Name
CI-69261 - DSD BF Vendor Maintenance IMPLEMENTED	DSD_BF_Vendor_Maintenance.doc

A Vendor record will be created using the Vendor Customer Creation document (VCC) to add the organization to the Vendor Customer (VCUST) table. Only one vendor may be created on a single document.

The VCC document is created through the document catalog. Once created, the VCC document may not be modified or cancelled once in Final Status. It cannot be copied or reference another document.

All document tabs are children of the Header, and most subsequent tabs are children of the Vendor Customer tab. At a high-level, the VCC document follows the structure illustrated in the figure below.

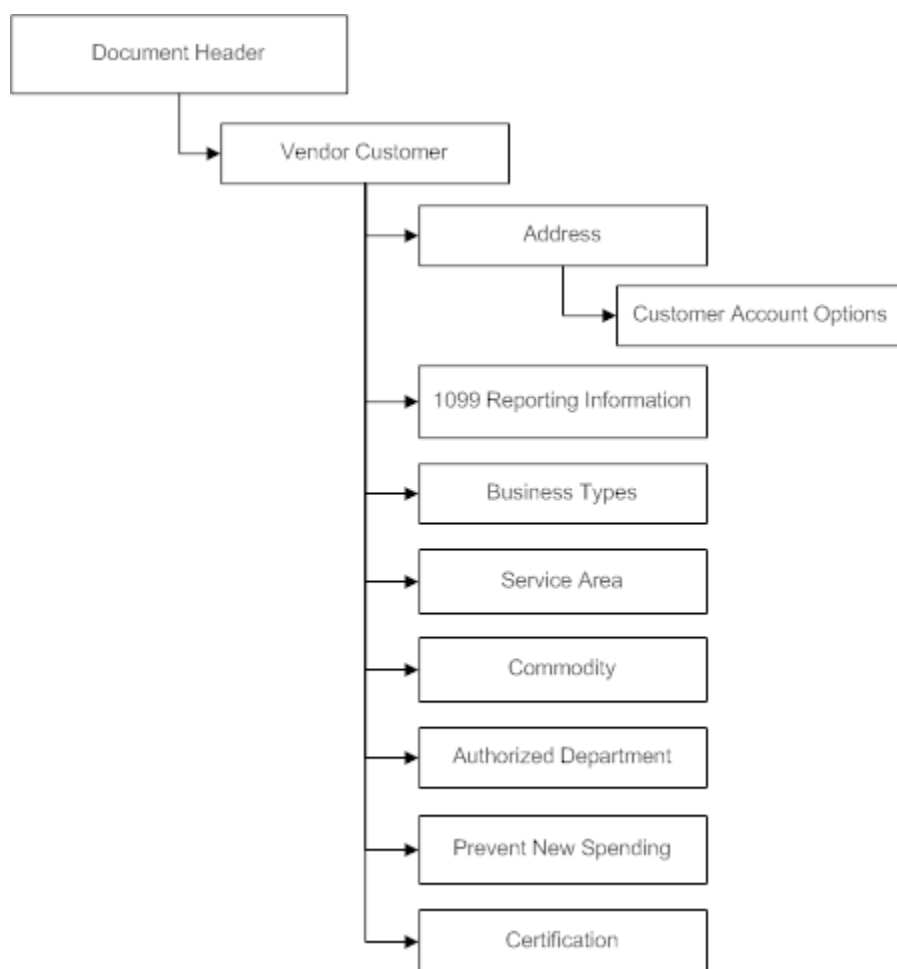


Figure – VCC Document Structure – Create Vendor

The VCC document is the primary data entry vehicle for three types of actions: create vendor receiving paper warrant, create vendor receiving Electronic Funds Transfer (EFT) and to create the pre-note.

A pre-note is a record containing EFT information about a vendor, such as bank account number and bank name, which is recorded in the Pre-note/EFT section of the VCC document. This information is sent along with the nightly payments to the financial institution, and notifies the Financial Institution of the intent to electronically deposit payments to one of its accounts and verifies that all the account information is correct.

The vendor will receive a paper warrant until the 20 days lapse for the EFT pre-note or the pre-note is turned off.

The VCC document is primarily data entry vehicle that supports the ability to update table data collected in document components. As evident in the above diagram, the document components mirror the makeup of the VCUST table. The VCC document supports workflow when approvals are required prior to table updates taking effect.

Each document component is mapped to a corresponding data object. Upon validate or submit the collected document data will be inserted as a new record into the corresponding data object, as if it was manually entered on the table. The existing table constraints will be fired as normal, therefore ensuring that the data is valid. Additional constraints may exist on the document, but the majority of the controls reside within the tables.

The VCC document is "smart" in one respect that once the Header information is entered, then upon saving the document (or navigating to another component), the system will create a line on the Vendor Customer document tab and Certification document tab if one does not already exist.

Modify Vendor

If a Vendor/Customer record already exists on the VCUST table, then a user may create a Vendor Customer Modification document (document code VCM) to modify data on Vendor Customer tables. Only one vendor may be modified on a single document.

The Vendor Customer Modification document may be created via document creation links on the VCUST table. A user may create a VCM document by using the Modify Existing Record or Add New Values links (specific labels vary by tab) on the VCUST table. Once created, the document will open to the Header, having pulled forward the corresponding Vendor Customer code in addition to the values for the record being accessed within VCUST to the corresponding page (e.g., Address, Business Types, etc.). The system will set the default Line Action to Modify on the related document tab.

All document tabs are children of the Header. Unlike the VCC document, there are no additional parent/child relationships among the VCM components. At a high-level, the Vendor Customer Modification document follows the structure illustrated in the figure below.

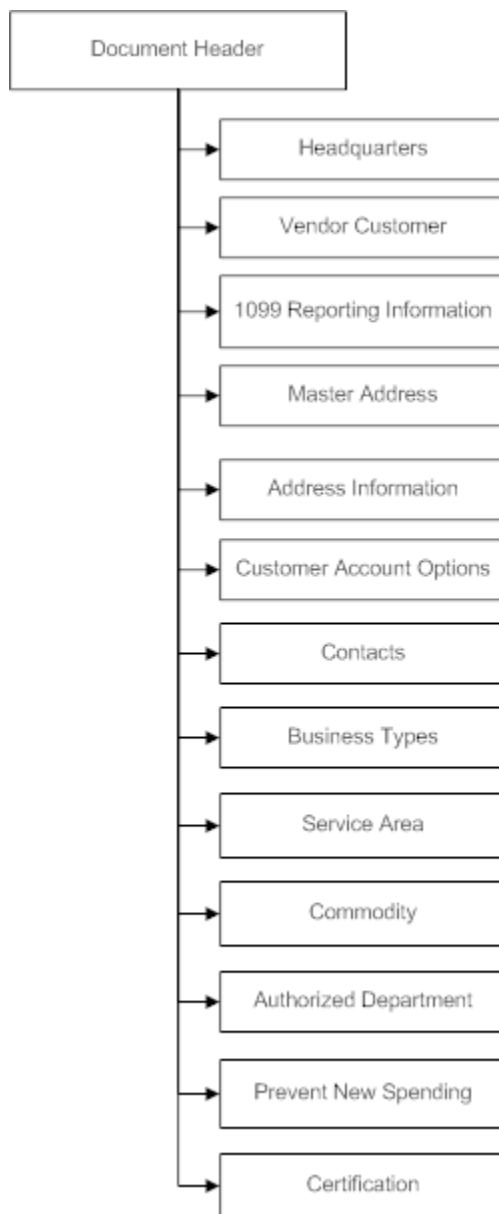


Figure – VCM Document Structure – Modify Vendor

With the exception of the Vendor Customer, Headquarters, and Certification tabs, which allow only one line per record, other VCUST pages also include a link to add new values for that specific tab. When this link is selected, the system will create a VCM document, but the only corresponding information to be inferred will be the Vendor Customer Code on the Header. The system will set the default Line Action on the corresponding page to New.

As with the VCC document, each document component is mapped to a corresponding data object. Depending on the Line Action, upon validate or submit the collected document data will either update or delete an existing record, or be inserted as a new record, on the corresponding data object as if it was manually entered on the table. The permissible line actions vary per tab, but may include New, Modify or Delete. Existing table constraints will be triggered as normal, therefore ensuring that the data is valid. Additional constraints may exist on the document, but the majority of the controls reside within the tables.

Following is additional detail related to how processing varies by line action:

- When the Line Action is set to New, the system will attempt to insert a new line into the corresponding table. If, upon validate or submit, the system finds a match with existing table's key values (e.g., Vendor Customer Code, Address ID and Address Type on the Address Information table), then the table's unique constraint violation will prevent the update from processing and the system will return an error.
- When the Line Action is set to Modify, the system will attempt to update the existing record by replacing the existing table values with the document values. If, upon validate or submit, the system does not find a match on the corresponding table's key values, then the system will return an error. This constraint will exist on the document. (NOTE: In order to modify an existing record, the user must first click on the Load Existing Values link. This will infer the current values for that record and allow the user to modify as needed.)
- When the Line Action is set to Delete, the system will attempt to delete the corresponding table record; however, existing table referential integrity logic will be enforced and deletions will only be allowed if no active child or related records exist. For example, a Master Address may not be deleted if active Address Types are associated with it. Therefore, it may be necessary to process several documents (one [1] to delete the Address Types, and another to delete the Master Address record) in order to complete a record deletion.

The AMS Row Version Number value is used to ensure that conflicting updates cannot be made to the same record. For example, a document may be created in order to modify an address but, prior to the document becoming final, a change to that same address is made directly to the table. In this case, the next time the document is validated or submitted (following the table change) the user will receive an error indicating, "This record has been modified by another user." The user may then select the row in error and click the Load Values link on the tab, which will refresh the page with the most current table information for that record. The user may then determine if additional modifications are still necessary, and if so, update and submit the document. The AMS Row Version Number will be reloaded when the Load Values link is clicked, when a document is created using the "Modify" link from VCUST table, or when returning from a pick (provided the pick is to the whole record and not only a component).

The Header is where the user determines the Vendor Customer Code to which all document modifications will be applied. If the document is created via the Document Catalog, then the user may select the Vendor Customer Code via a pick to the VCUST table. Picks on the subsequent tabs within the document will be filtered based on this value. The Header includes standard document fields for the Record Date, Document Description, and Extended Description. Information entered into the Extended Description field will be inferred to the Vendor Customer tab upon saving the document and then update the Comments field on the Vendor Customer table when the document is submitted. The document record date will default when the document is accepted.

The Header also includes a section for the user to change a Vendor Customer's Legal Name. This will be the only place on the VCM document where the legal name may be changed for either a Headquarters or non-Headquarters record. The Legal Name on the Header will be the name inferred to all subsequent tabs when new lines are inserted. However, the system must enforce that the correct data object is updated with the Legal Name, depending upon whether the record is a Headquarters Account. To control this, upon validate or submit, if the Header Legal Name does not equal the Legal Name on the Vendor Customer table, then the system will require at least one line to exist on either the Headquarters tab or the Vendor Customer tab. Tab-specific document logic will verify that any line on the Headquarters tab is a Headquarters account. Additionally, if upon validate or submit the Organization Type on the Header does not equal the Vendor Customer Organization Type, and then the system will require a line on the Vendor Customer document tab. The Vendor Customer tab will infer the Organization Type value from the Header, and it will be a protected field on the VCUST tab. When the document is processed, the Vendor Customer data object will be updated with the new Organization Type.

Finally, Document Information (e.g. change management) values on the Header will update the corresponding Vendor Customer, 1099 and Customer Account tables. If the document is not subject to approval, then the Last Approved By and On fields will be populated with the Last Modified By and On values. If the document is subject to approval, then the Last Approved By and On fields will be set based on the user who approved the document and submitted it to Final status.

Similar to the VCC, once created the VCM document may not be modified or cancelled once Final, and it cannot be copied or reference another document.

Business Process Flow – Vendor Maintenance

This section is not applicable for Warrant Management.

Initiation/Triggers, Pre-Conditions and Post Conditions

This section is not applicable for Warrant Management.

Process/Screen Flow

This section is not applicable for Warrant Management.

DSD 19/Payroll – Warrant Management/Business Process /Business Process Functions/Warrant Processing

CI	Document Name
CI-69662 - DSD BF Warrant Processing IMPLEMENTED	DSD_BF_Warrant_Processing.doc

Payroll Warrant Processing

Payroll will calculate pay based on the output from Time-to-Gross (TTG) and Gross-to-Net (GTN). TTG translates the employee's hours worked to gross pay based on the employee's assigned pay rate. GTN accumulates the employees' gross pay data and does additional processing such as deduction and tax processing. The system, once configured, establishes the current pay period and verifies that the date of the timesheets fall within the current pay period, then it verifies the employee is eligible for the pay type, and verifies the hours entered. The system then makes the conversion from time, multiplied by the hourly wage, to gross pay, and updates the summary tables for pay, tax, and deduction activity.

In the following example, Provider 1 has a timesheet for 10 hours at \$10 per hour. The pay is for the Personal Care Services Program (PCSP) for Yolo County (57) in District Office 1 (01.) Once the net pay is known, the Employee Status Maintenance (ESMT) table is used to lookup the Provider's Home Dept. and Home Unit. The Extended Department (DEPTX) table is used to lookup the correct Labor Distribution (LDPR) based on the Home Dept. and Home Unit. The LDPR on the DEPTX table is used to look up the accounting. The Pay Events table will tell whether there are any overrides to the LDPR.

Provider 1: Pay check example

Hours	Rate	Gross Pay
10	\$10	\$100
Federal Income Tax		\$10
Personal Income Tax		\$10
Labor Organization Dues		\$10
Net Pay		\$70

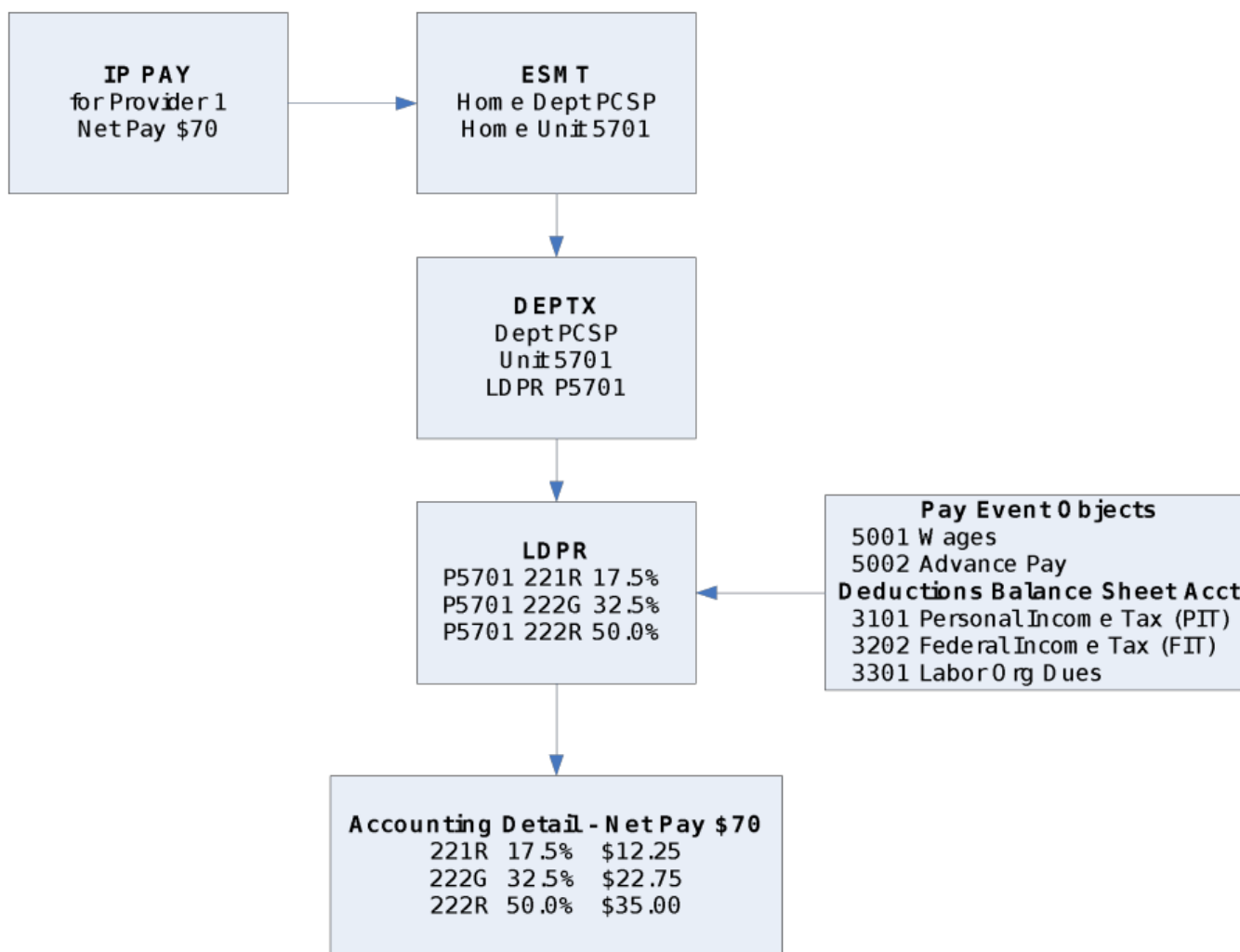


Figure – IP Pay Example

Internal Interface between CMIPS HRM and Financial component of CMIPS.

After the payroll nightly cycle is run in the Payroll component of CMIPS, the payroll expenses and liabilities (State and Federal taxes) and vendor payables (i.e., health benefits, labor organization dues and lien payments) are posted to the financial components of CMIPS so that the proper accounts can be debited and credited and vendors can be paid. The Payroll Accounting Management (PAM) process is a two-step process. The steps are:

- Paycheck Processor - Processes the pay details and deduction details to create payroll expense journal, payroll liability journal and payroll fringe journal records.
- Document Generator - Payroll expense journal and payroll liability journal records are used to create the following payroll accounting (PYRL) documents.
- Net pay liability journal records are used to create Payroll Net Payment (PRLNP) documents.
- Payroll fringe journal records with the vendor information are used to create Payroll Vendor Payment Request (PRLVP) documents. These are disbursement documents for health benefits, Labor Organization dues, and lien holders.
- Payroll Liability Documents (PRLIA) which record liabilities associated with the payroll.
- Payroll Expenditure Documents (PREXP) which accrued expenditures associated with the payroll.

Note: The journals that are posted in the financial component of CMIPS are a summary of the payroll run, not detail by individual. Payroll details for individual Providers or Recipients can be accessed in Case Management (CM).

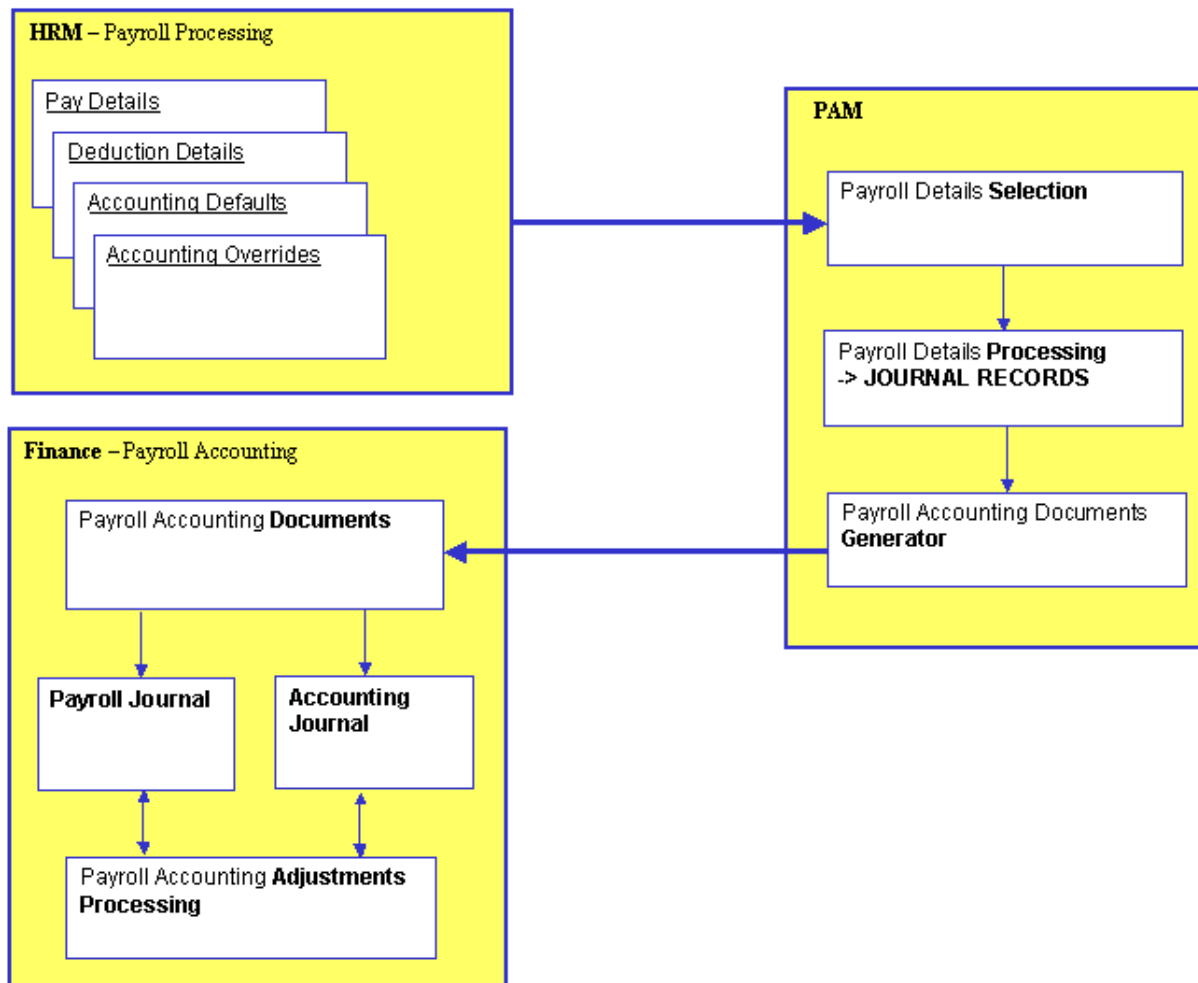


Figure – Internal Interface between HRM and Financial

Vendor Warrant Processing

Following the PAM process, the PRLVP documents are now viewable on the Disbursement Request (DISRQ) table. The Automated Disbursement Chain (AD Chain) job in the Financial component of CMIPS is a group of jobs that work together to create disbursement documents from the payment request documents. The Automated Disbursement process selects authorized payments, edits payments for validity, and post payment transactions.

The AD Chain job will be run daily as part of the nightly cycle.

Lien holder disbursements will be run daily.

Payments to Public Authorities and labor organizations will be made on the last day of the month.

The Disbursement Parameters (DISPA) table defines the parameters for non-EFT warrants and EFT warrants. This table must be updated before each Automated Disbursement cycle. The DISPA table is defined in the Financial Data Elements section of this document.

Non-EFT Warrant Processing

The instructions in the table below describe the Automated Disbursement process for non-EFT Warrants.

Table – Automated Disbursement Processing Steps for non-EFT Warrants

Step	Role	Description
------	------	-------------

1	<p>Verify that the Payment requests eligible for disbursement are present on the Disbursement Request (DISRQ) table. From the secondary panel on the left side of the screen, select Search then select the Page Search link. In the Page Search screen, enter the page code DISRQ in the Page Code field and select Browse. The results are displayed in a table below the Page Code field. Select the link to take you to the Disbursement Request table. The Disbursement Request table displays showing the payment requests that are eligible for disbursement processing. Verify that the desired payment requests are present by checking the following fields in the Vendor Section:</p> <ul style="list-style-type: none"> • Document Code (e.g. PRLVP in the document code) • Document Department • Document ID • Vendor Code <p>In the Disbursement control section verify the following fields:</p> <ul style="list-style-type: none"> • Scheduled Payment Date • Handling Code • Disbursement Format • Disbursement Type • Hold Payment Reason (if applicable) <p>Note: If there is a hold, the Hold Payment Reason will be displayed.</p>
2	<p>Set up the job parameters on the Disbursement Parameters table (DISPA): From the secondary panel on the left side of the screen, select Search then select the Page Search link. In the Page Search screen, enter the page code DISPA in the Page Code field and select Browse. The results are displayed in a table below the Page Code field. Select the link to take you to the Disbursement Parameters table. The Disbursement Parameters table is displayed. Enter the following fields:</p> <ul style="list-style-type: none"> • Document Code • Documents Department • Disbursement Type • Disbursement Priority • Bank Account Code • Active check box • From Date • To Date • Check Date

3	<p>Execute the AD Chain job:</p> <p>From the secondary panel on the left side of the screen, select Search then select the Page Search link.</p> <p>In the Page Search screen, enter the page code BATRUN in the Page Code field and select Browse.</p> <p>The results are displayed in a table below the Page Code field. Select the link to take you to the Job Manager.</p> <p>On the secondary panel, on the left side of the screen select Financial, then select Accounts Payable, then select Chain Jobs, and then select the AD Chain link.</p> <p>The Chain Job Summary for AD Chain page is displayed. Enter a job name. At the bottom of this page, select the Schedule New Chain Job link.</p> <p>The Job Steps for new Chain Job page is displayed. If the State of California wants to disable a job then select the check box for that job. At the bottom of this page, select the Continue Scheduling Chain Job link.</p> <p>The New Chain Job page is displayed.</p> <p>Enter a job name</p> <p>Select the App Server ID</p> <p>Select Save</p> <p>Select the Setup Parameters for Job Steps link</p> <p>The Job Steps page is displayed.</p> <p>Define the batch parameters for each job shown below:</p> <ul style="list-style-type: none"> • AD XML creation • AD Upload • AD Submitter • AD Clean Up and IT XML creation • Check Number Assignment • IT Upload • IT Submit <p>Note: Select the Edit Job Parameters link to enter the parameters for each job.</p> <p>Select OK at the bottom of the screen.</p> <p>The New Chain Job is displayed.</p> <p>Select the Submit Request link at the bottom of the page.</p> <p>Verify that the job completes successfully:</p> <p>In the Chain Job Summary for AD Chain enter the Job ID or Job Name and select Browse until the job submitted by the user appears. Alternatively, the user may monitor the job status in the View Pending Chain Jobs page (select the View Pending Chain Jobs link at the bottom).</p> <p>If needed select the Next or Last link to get to the end of the job listing. Alternatively you can search for a job by entering Job ID or Job Name and then selecting Browse.</p> <p>The job should have a return code = Successful.</p> <p>Review job results and report output.</p> <p>To see the logs, select on the View Log link.</p> <p>Select the View Report(s) link for the Job ID (within the Chain job) that completed successfully by the user.</p> <p>Choose the preferred format of viewing the report; HTML vs. PDF.</p> <p>AD Chain – AD Exception report output.</p> <p>If the AD Chain was run during the nightly cycle then the AD Exception report should be reviewed the following business day by a business user. The business user should then conduct research and take corrective action.</p> <p>Review the Disbursement Document (AD) created on the DISRQ table:</p> <p>From the secondary panel on the left side of the screen, select Search then select the Page Search link.</p> <p>In the Page Search screen, enter the page code DISRQ in the Page Code field and select Browse.</p> <p>The results are displayed in a table below the Page Code field. Select the link to take you to the Disbursement Request table.</p> <p>From the Disbursement Request table search for the payment request document, Select the Search link and use the following criteria:</p> <ul style="list-style-type: none"> • Document Code (e.g. PRLVP in the document code) • Document Department • Document ID • Vendor Code <p>From the Payment Request Document, select the Document Reference link in the secondary navigation panel.</p> <p>From the Document Reference Query page</p> <p>Select the Forward Reference link from the secondary panel at the left side of the screen, the Document Reference Query page display, select the AD Document record to review the AD Document:</p> <p>The AD document should be submitted to FINAL.</p> <p>The Warrant/EFT Number is posted.</p>
---	---

4	<p>Execute the Check Register Report</p> <p>From the secondary panel on the left side of the screen, select Search then select the Page Search link.</p> <p>In the Page Search screen, enter the page code BATRUN in the Page Code field and select Browse.</p> <p>The results are displayed in a table below the Page Code field. Select the link to take you to the Job Manager.</p> <p>On the secondary panel, on the left side of the screen select Financial, then select Accounts Payable, then select Reports-AP, and then select the ADCheckRegister link.</p> <p>The Report Summary for ADCheckRegister page is displayed. Enter a job name. At the bottom of this page, select the Schedule New Job link.</p> <p>The New Job page is displayed.</p> <p>Enter a job name</p> <p>Select the App Server ID</p> <p>Select Save</p> <p>Select the Setup Parameters</p> <p>The Job Parameters for ADCheckRegister is displayed. Enter the following fields:</p> <ul style="list-style-type: none"> • Bank Account Code • Client Name for Report • Disbursement Type • End Date • Run ID • Start Date <p>Select OK at the bottom of the screen.</p> <p>The New Job is displayed.</p> <p>Select the Submit Request link at the bottom of the page.</p> <p>Verify that the job completes successfully:</p> <p>In the Report Summary for ADCheckRegister, enter the Job ID or Job Name and select browse until the job submitted by the user appears. Alternatively, the user may monitor the job status in the View Pending Chain Jobs page (select the View Pending Chain link at the bottom).</p> <p>If needed select the Next or Last to get to the end of the job listing. Alternatively you can search for a job by entering Job ID or Job Name and then selecting Browse.</p> <p>The job should have a return code = Successful.</p> <p>Review job results and report output.</p>
---	--

System Updates/Expected Results for non-EFT Warrant Processing

The Automated Disbursement Document (AD) is created by the AD Chain. The payment request (PRLVP) record is no longer on Disbursement Request table and reflects a closed amount. The Check Reconciliation (CHREC) table in the financial component of CMIPS is updated by the disbursement process.

Check Reconciliation (CHREC)

A record is added to CHREC when the AD document is processed to "Final." The Status is then set to "Paid."

EFT Warrant Processing

The instructions in Table 2 below describe the Automated Disbursement process for EFT Warrants.

Table – Automated Disbursement Processing Steps for EFT Warrants

Step	Role	Description
1		<p>Verify that the Payment requests documents eligible for disbursement are present on the Disbursement Request (DISRQ) table.</p> <p>From the secondary panel on the left side of the screen, select Search then select the Page Search link.</p> <p>In the Page Search screen, enter the page code DISRQ in the Page Code field and select Browse.</p> <p>The results are displayed in a table below the Page Code field. Select the link to take you to the Disbursement Request table.</p> <p>The Disbursement Request table displays showing the payment requests that are eligible for disbursement processing.</p> <p>Verify that the desired payment requests are present by checking the following fields in the Vendor Section:</p> <ul style="list-style-type: none"> • Document Code • Document Department • Document ID • Vendor Code (e.g., PRLVP in the document code) <p>In the Disbursement control section verify the following fields:</p> <ul style="list-style-type: none"> • Scheduled Payment Date • Handling Code • Disbursement Format • Disbursement Type • Hold Payment Reason <p>Note: If there is a hold, the Hold Payment Reason will be displayed.</p>

2	<p>Set up the job parameters on the Disbursement Parameters table (DISPA): From the secondary panel on the left side of the screen, select Search then select the Page Search link. In the Page Search screen, enter the page code DISPA in the Page Code field and select Browse. The results are displayed in a table below the Page Code field. Select the link to take you to the Disbursement Parameters table. The Disbursement Parameters table is displayed. Enter the following fields:</p> <ul style="list-style-type: none"> • Document Code • Documents Department • Disbursement Type • Disbursement Priority • Bank Account Code • Active check box • From Date • To Date • Check Date
3	<p>Execute the AD Chain job: From the secondary panel on the left side of the screen, select Search then select the Page Search link. In the Page Search screen, enter the page code BATRUN in the Page Code field and select Browse. The results are displayed in a table below the Page Code field. Select the link to take you to the Job Manager. On the secondary panel, on the left side of the screen select Financial, then select Accounts Payable, then select Chain Jobs, and then select the AD Chain link. The Chain Job Summary for AD Chain page is displayed. Enter a job name. At the bottom of this page, select the Schedule New Job link. The Job Steps for new Chain Job page is displayed. If the State of California wants to disable a job then select the check box for that job. At the bottom of this page, select the Continue Scheduling Chain Job link. The New Chain Job page is displayed. Enter a job name Select the App Server ID Select Save Select the Setup Parameters for Job Steps link The Job Steps page is displayed. Define the batch parameters for each job shown below:</p> <ul style="list-style-type: none"> • AD XML creation • AD Upload • AD Submitter • AD Clean Up and IT XML creation • Check Number Assignment • IT Upload • IT Submit <p>Note: Select the Edit Job Parameters link to enter the parameters for each job. Select OK at the bottom of the screen. The New Chain Job page is displayed. Select the Submit Request link at the bottom of the page. Verify that the job completes successfully: In the Chain Job Summary for AD Chain page enter the Job ID or Job Name and select Browse until the job submitted by the user appears. Alternatively, the user may monitor the job status in the View Pending Chain Jobs page (select the View Pending Chain Jobs link at the bottom). If needed select the Next or the Last link to get to the end of the job listing. Alternatively you can search for a job by entering Job ID or Job Name and then selecting Browse. The job should have a return code = Successful. Review job results and report output. To see the logs, select on the View Log link. Select the View Report(s) link for the Job ID (within the Chain job) that completed successfully by the user. Choose the preferred format of viewing the report; HTML vs. PDF. AD Chain – AD Exception report output. If the AD Chain was run during the nightly cycle then the AD Exception report should be reviewed the following business day by a business user. The business user should then conduct research and take corrective action. Review the Disbursement Document (EFT) created on the DISRQ table. From the secondary panel on the left side of the screen, select Search then select the Page Search link. In the Page Search screen, enter the page code DISRQ in the Page Code field and select Browse. The results are displayed in a table below the Page Code field. Select the link to take you to the Disbursement Request table. From the Disbursement Request table search for the payment request document, Select the Search link and use the following criteria:</p> <ul style="list-style-type: none"> • Document Code • Document Department • Document ID • Vendor Code (e.g., PRLVP in the document code) <p>From the Payment Request Document, select the Document Reference link in the secondary navigation panel. From the Document Reference Query page Select the Forward Reference link from the secondary panel at the left side of the screen, the Document Reference Query page display, select the EFT Document record to review the EFT Document: The EFT document should be submitted to FINAL. The Check/EFT Number is posted.</p>

4	<p>Execute the EF ACH chain job.</p> <p>In the Page Search screen, enter the page code BATRUN in the Page Code field and select Browse.</p> <p>The results are displayed in a table below the Page Code field. Select the link to take you to the Job Manager.</p> <p>On the secondary panel, on the left side of the screen select Financial, then select Accounts Payable, then select Chain Jobs, and then select the EF ACH Transaction link.</p> <p>The Chain Job Summary for EF ACH Transaction page is displayed. Enter a job name. At the bottom of this page, select the Schedule New Chain Job link.</p> <p>The Job Steps for New Chain Job page is displayed. If the State of California wants to disable a job then select the check box for that job. At the bottom of this page, select the Continue Scheduling Chain Job link.</p> <p>The New Chain Job page is displayed.</p> <p>Enter a job name</p> <p>Select the App Server ID</p> <p>Select Save</p> <p>Select the Setup Parameters for Job Steps link</p> <p>The Job Steps page is displayed.</p> <p>Define the batch parameters for each job shown below:</p> <ul style="list-style-type: none"> • Build XML File • Build Flat File <p>Note: Select the Edit Job Parameters link to enter the parameters for each job.</p> <p>Select OK at the bottom of the screen.</p> <p>The New Chain Job is displayed.</p> <p>Select the Submit Request link at the bottom of the page.</p> <p>Verify that the job completes successfully:</p> <p>In the Chain Job Summary for EF ACH Transaction enter the Job ID or Job Name and select Browse until the job submitted by the user appears. Alternatively, the user may monitor the job status in the View Pending Chain Jobs page (select the View Pending Chain Jobs link at the bottom).</p> <p>If needed select the Next or the Last link to get to the end of the job listing. Alternatively you can search for a job by entering Job ID or Job Name and then selecting Browse.</p> <p>The job should have a return code = Successful.</p> <p>Review job results and report output.</p> <p>To see the logs, select on the View Log link.</p> <p>Select the View Report(s) link for the job ID (within the Chain job) that completed successfully by the user.</p> <p>Choose the preferred format of viewing the report; HTML vs. PDF.</p> <p>Review the XML file output and the DAT file output.</p> <p>Output files are available in the ExportImport directory.</p> <p>EFACHXMLFormatFile.xml is the XML Output file used as input to create the flat files (i.e., DAT files) that may then be sent to a client's partner bank.</p> <p>Review and transmit the EFT output.</p> <p>EFACHPPDFFile.DAT, EFACHCCDFFile.DAT and EFACHCTXFile.DAT are the output files that may be sent to a partner bank to record the disbursement of EFT transactions.</p>
---	---

System Updates/Expected Results

The payment request (PRLVP) record is no longer on Disbursement Request table and reflects a closed amount. The Check Reconciliation (CHREC) table in the financial component of CMIPS is updated by the disbursement process.

Note: The Check/EFT number is present on Check Reconciliation even though it is not present on the Accounting Journal.

Also note that the cleared date is the Record Date plus the numbers of days for EFT lag defined on DISPA (this excludes weekends and holidays).

Business Process Flow – Warrant Processing

This section is not applicable for Warrant Management.

Initiation/Triggers, Pre-Conditions and Post Conditions

This section is not applicable for Warrant Management.

Process/Screen Flow

This section is not applicable for Warrant Management.

DSD 19/Payroll – Warrant Management/Business Process /Business Process Functions/Electronic Claim File

CI	Document Name
CI-69658 - DSD BF Electronic Claims File IMPLEMENTED	DSD_BF_Electronic_Claims_File.doc

A number of electronic claim files will be sent from CMIPS to outside entities as well as accepting files back from them. The layout of the files summarized in this section can be found in the Interfaces section of this document.

Electronic Claim Files from CMIPS

Two files will be sent to the Operations Unit at the SCO on a daily basis. These files are:

- EFT payments including vendor and payroll warrants.
- Non-EFT payments including vendor and payroll warrants.

The IHSS-R, PCSP, WPCS, IPO and IPW1-6 programs will be included in these files. Cancelled warrants are not included in these files. The files shall include final payment warrants for those Providers who have quit or have been discharged. The files will include the following types of payments. Not all types of payments will be included each day but will be sent according to the schedule below.

- Recipient Advance Payments (without Share of Cost) – Once per month per Recipient
- Recipient Advance Payment (with Share of Cost) - Once per month per Recipient
- Restaurant Meal Allowance - Twice per month
- State Hearing Payments – When requested
- Individual Provider Payments – Daily, when requested
- Warrant Re-Issue Requests – Daily, when requested
- Retro-Active or One-Time Payments – Daily, when requested
- SDI Refund Payments - Once per quarter
- Lien Holder Payments - Daily, when requested
- Public Authority Payments (health benefits) – Monthly
- Labor Organization Due Payments – Monthly

A separate file containing FICA Refund Payment information will be submitted to SCO once per year and no other payments will be included in this file. This file will have the same layout as the above electronic claim files.

A separate file containing pre-note information will be included weekly with the daily files submitted to SCO. This information notifies SCO of the intent to electronically deposit payments to one of its accounts and verifies that all the account information is correct. SCO will respond within 20 days only if the pre-note request has been rejected. The vendor or Provider/Recipient will receive a paper warrant until the 20 days lapse for the EFT pre-note or the pre-note flag is turned off. This file will have the same layout as the above electronic claim files. In CMIPS, lien holders will only receive paper warrants, not EFT, as in legacy CMIPS.

A daily Audit File will be sent to the Audit Unit at SCO. This file will contain all the data necessary for SCO to be able to audit the CDSS programs. This file includes both EFT and non-EFT payments.

Electronic Claim Files to CMIPS

Two files will be returned from the Operations unit at SCO on a daily basis. These files will be used to update CMIPS with the SCO warrant numbers. These files are:

- EFT payments including vendor and payroll warrants updated with SCO Warrant Number and SCO issue date.
- Non-EFT payments including vendor and payroll warrants updated with SCO Warrant Number and SCO issue date.

An exception report will be produced following the updates for those instances where the record in the SCO update file cannot be matched to a record in CMIPS.

Two additional files will be returned from the SCO Operations unit. These files will be used update the CMIPS warrant status.

- All Redeposited and Replacement warrants.
- All Stale dated warrants.

An exception report will be produced following the updates for those instances where the record in the SCO update file cannot be matched to a record in CMIPS.

One file will be returned from the State Treasurer's Office (STO) on a daily basis. This file will include a listing of all warrants that have cleared and will be uploaded to CMIPS to update the warrant status. An exception report will be produced following the updates for those instances where the record in the STO update file cannot be matched to a record in CMIPS.

Business Process Flow – Electronic Claim File

This section is not applicable for Warrant Management.

Initiation/Triggers, Pre-Conditions and Post Conditions

This section is not applicable for Warrant Management.

Process/Screen Flow

This section is not applicable for Warrant Management.

DSD 19/Payroll – Warrant Management/Business Process /Business Process Functions/Daily Payroll Reconciliation

CI	Document Name
CI-69270 - DSD BF Daily Payroll Reconciliation IMPLEMENTED	DSD_BF_Daily_Payroll_Reconciliation.doc

CMIPS will process the Provider timesheets in the nightly cycle. Resulting from the daily payroll run, CMIPS will generate Daily Payment Voucher reports to be sent to CDSS Accounting and posted to On-line reports.

These reports will contain detail and summary totals for In-Home Supportive Services-Residual (IHSS-R), Personal Care Services Program (PCSP), Waiver Personal Care Services (WPCS), In-Home Plus Option (IPO) and IHSS Plus Waiver 1-6 (IPW 1-6) programs. CDSS Accounting will use the Daily Payment Voucher Report to generate the entry to be posted to CALSTARS.

CMIPS will produce the Daily Claim Schedule Transmittal report. The Contractor will use the totals on this report to complete the STD 218 Claim Schedule Face Sheet EFT and non-EFT, which will be sent to CDSS Accounting.

After the warrants are generated, SCO will return a file that will be uploaded to CMIPS daily. The file will update the following warrant information:

Provider/Recipient Warrants

For payroll warrants:

- Update the Warrant Status field to "Paid" in the PAY_CHECK table.
- Update the Replace Check field to "N/A" in the PAY_CHECK table.
- Update the Issue Date field based on the date the warrant was processed by SCO in the PAY_CHECK table.
- Update the Last Action Date field based on the date the warrant was processed by SCO in the PAY_CHECK table.
- Update the SCO Warrant Number in the PAY_CHECK table.
- The Issue Date field based on the date the warrant was processed by SCO and the SCO warrant number will then be viewable on the Warrant Cross Reference (WCRE) table.

Vendor Warrants

For vendor warrants:

- Update the Warrant Status field to "Paid" in the Check Reconciliation (CHREC) table.
- Update the SCO Warrant number in the CHREC table.
- Update the Issue Date field based on the date the warrant was processed by SCO in the CHREC table.
- Update the Last Action Date field based on the date the warrant was processed by SCO in the CHREC table.
- An exception report will be produced following the updates for those instances where the record in the SCO update file cannot be matched to a record in CMIPS.
- The STO will also return a file that will be uploaded to CMIPS daily. The file will update the following warrant information:
- Update the Warrant Status field to "Cleared" and update the Cleared Date in the Check Reconciliation (CHREC) table for vendor warrants. This update will move the record from the CHREC table to the Paid Checks (PDCHK) table.

Or for Provider/Recipient warrants:

- Update the Warrant Status field to "Cleared" in the PAY_CHECK table for payroll warrants.
- An exception report will be produced following the updates for those instances where the record in the STO update file cannot be matched to a record in CMIPS.

CDSS Billing Processes

CMIPS will generate Daily, Weekly, Monthly, Quarterly and Annual Payment Voucher reports to be sent to CDSS and posted to online reports. These reports will have the same format as the Daily Payment Voucher reports but the content will vary based on the time period. CDSS Accounting will use the Weekly Payment Voucher report to produce the invoice to bill the Federal government for its share of the program cost. Monthly Payment Voucher reports will be used to produce the monthly invoices to bill the counties for their share of the program costs and taxes. Quarterly Payment Voucher reports will be used to produce the quarterly invoice to bill the Federal government for its share of the taxes. All the Payment Voucher reports (Daily, Weekly, Monthly, Quarterly and Annual) shall contain sufficient data for CDSS Accounting to be able to post the daily payroll and the county and Federal invoices to CALSTARS.

Business Process Flow – Daily Payroll Reconciliation

This section is not applicable for Warrant Management.

Initiation/Triggers, Pre-Conditions and Post Conditions

This section is not applicable for Warrant Management.

Process/Screen Flow

The following diagrams describe the process flow for Daily Payroll Balancing.

CI	Document Name
CI-117885 - DSD BP Daily Payroll Balancing IMPLEMENTS	DSD_BP_Daily_Payroll_Balancing.doc

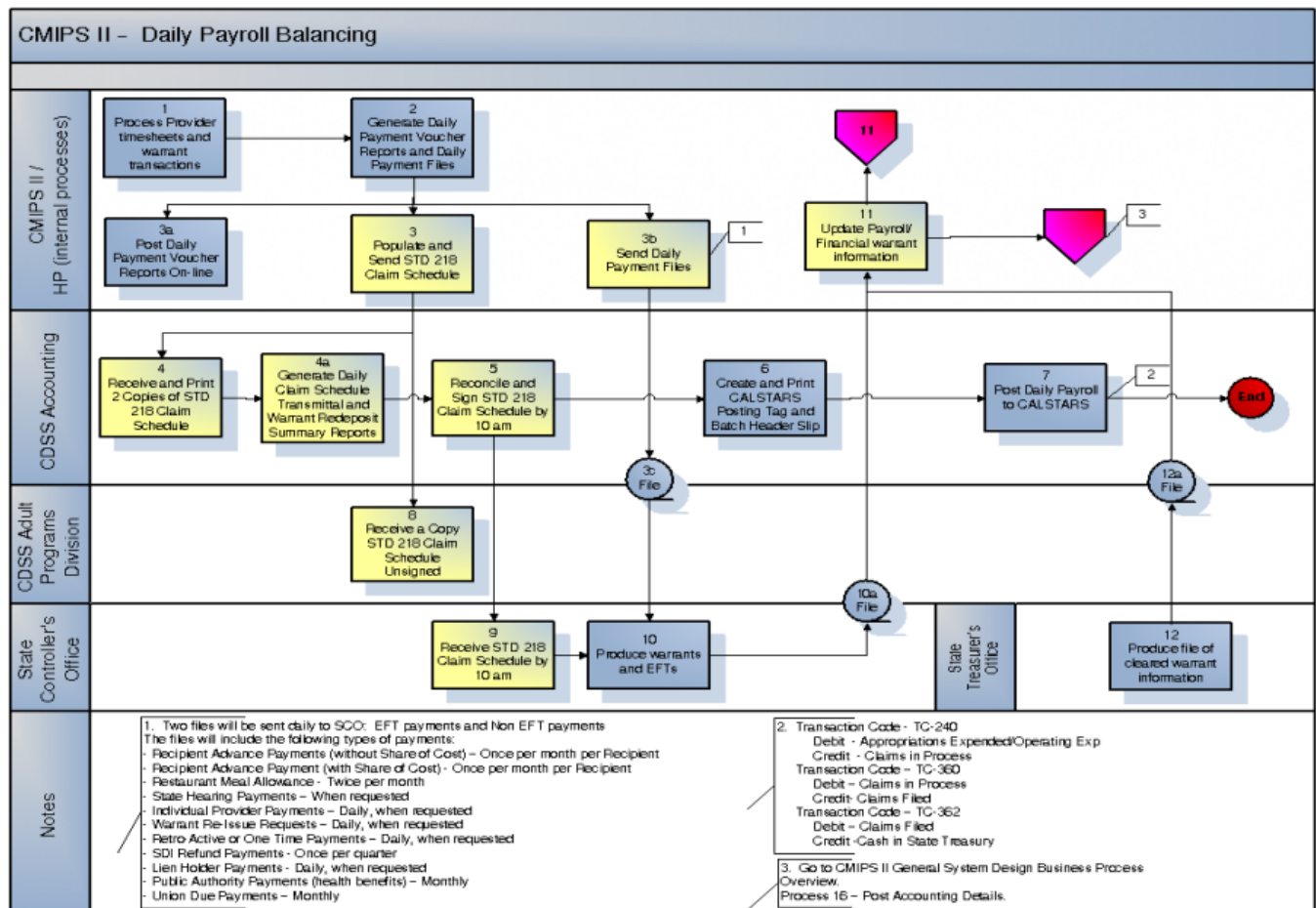


Figure – To Be Daily Payroll Processing – Part 1

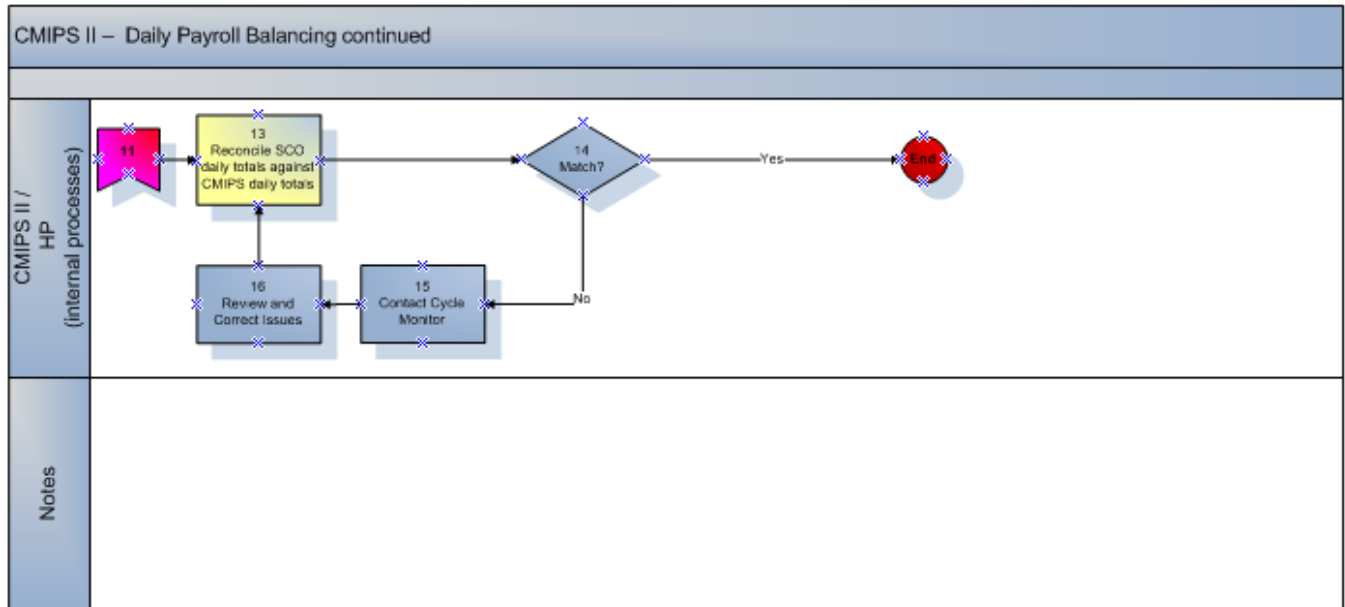


Figure – To Be Daily Payroll Processing – Part 2

DSD 19/Payroll – Warrant Management/Business Process /Business Process Functions/Reconciling of Tax Filing

CI	Document Name
CI-69263 - DSD BF Reconciling of Tax Filing IMPLEMENTED	DSD_BF_Reconciling_of_Tax_Filing.doc

Federal and State taxes are paid quarterly per statutory requirements for CMIPS. The Contractor initiates the process at the end of the quarter by creating a Quarterly Federal Tax Disbursement Summary Report for the Federal Taxes and Quarterly State Tax Disbursement Summary Report for the State taxes and then preparing a Quarterly Tax Memo for CDSS Accounting. A copy of the Tax Memo is sent to the CDSS Adult Program Division (APD) for its reference and input in the process.

Processing the Federal Tax Payments

Once CDSS Accounting receives the Tax Memo along with the Quarterly Federal Tax Disbursement Summary Report from the Contractor, staff prepares and posts the relevant Journal Entry into CALSTARS.

- The Journal Entry for Federal Tax is TC-240:
- Debit – App expend/operating expend & expenses
- Credit – Claims in Process

Simultaneously, CDSS Accounting prepares and sends a Claim Schedule Face Sheet to the SCO for further action. The SCO deposits the taxable amount into a zero balance account for onward sweep by the Contractor. This is known as "Special Handling. Approximately two weeks after APD receives the Tax Memo, they call the bank where the funds are swept by SCO and verify the amount. Then APD informs the Contractor regarding the funds availability. Once the Contractor is informed regarding the availability of funds, they send the quarterly tax payment to the IRS through a system known as Electronic Federal Tax Payment System (EFTPS). Once the funds are transferred, then a Journal Voucher will be created in the Financial component of CMIPS to reduce the tax liability accounts and to reduce the cash account. The CMIPS Journal Voucher entry should reconcile with the Quarterly Federal Tax Disbursement Summary Report and the amount of the warrant issued.

The following are the Federal Taxes:

- Federal Income Tax (FIT)
- Federal Insurance Contributions Act (FICA)
- Medicare Taxes
- Federal Unemployment Tax Act (FUTA)

An example of the Federal Tax Memo that will be sent to CDSS Accounting is provided in the Forms section. The tax memo and tax disbursement report should reconcile before the warrant is issued.

Processing the State Tax Payments

Once CDSS Accounting Department receives the Tax Memo along with the Quarterly State Tax Disbursement Summary Report from the Contractor, they prepare and post the relevant Journal Entry into CALSTARS.

- The Journal Entry for State Tax is TC-242:
- Debit – App expend/operating expend & expenses
- Credit – Cash in State Treasury

Simultaneously, CDSS Accounting prepares and sends an Inter-Agency Fund Transfer Request (CA 504) to the SCO for further action. A copy of the CA 504 is also sent to Employment Development Department (EDD), the proponent for the State Taxes and the Recipient of the inter-agency fund transfer. The SCO processes the fund transfer and generates a Notice of Transfer, also known as the SCO Journal Entry. SCO follows through by sending the Journal Entry to CDSS Accounting. Once the funds are transferred, then a Journal Voucher will be created in the financial component of CMIPS to reduce the tax liability accounts and to reduce the cash account. The CMIPS Journal Voucher entry should reconcile with the Quarterly State Tax Disbursement Summary Report and the amount transferred to EDD.

The following are the State Taxes:

- Personal Income Tax (PIT)
- State Unemployment Insurance (SUI)
- Employer Training Tax (ETT)
- State Disability Insurance (SDI)
- Disability Insurance Elective Coverage (DIEC)

An example of the State Tax Memo that will be sent to CDSS Accounting is provided in the Forms section of this document. The tax memo and tax disbursement report should reconcile before the warrant is issued.

Business Process Flow – Reconciling of Tax Filing

This section is not applicable for Reconciling of Tax Filing Management.

Initiation/Triggers, Pre-Conditions and Post Conditions

This section is not applicable for Reconciling of Tax Filing Management.

Process/Screen Flow

The diagrams in the figures below describe the Federal and State Tax payments processes.

CI	Document Name
CI-116696 - DSD BP Quarterly Federal Taxes IMPLEMENTED	DSD_BP_Quarterly_Federal_Taxes.doc

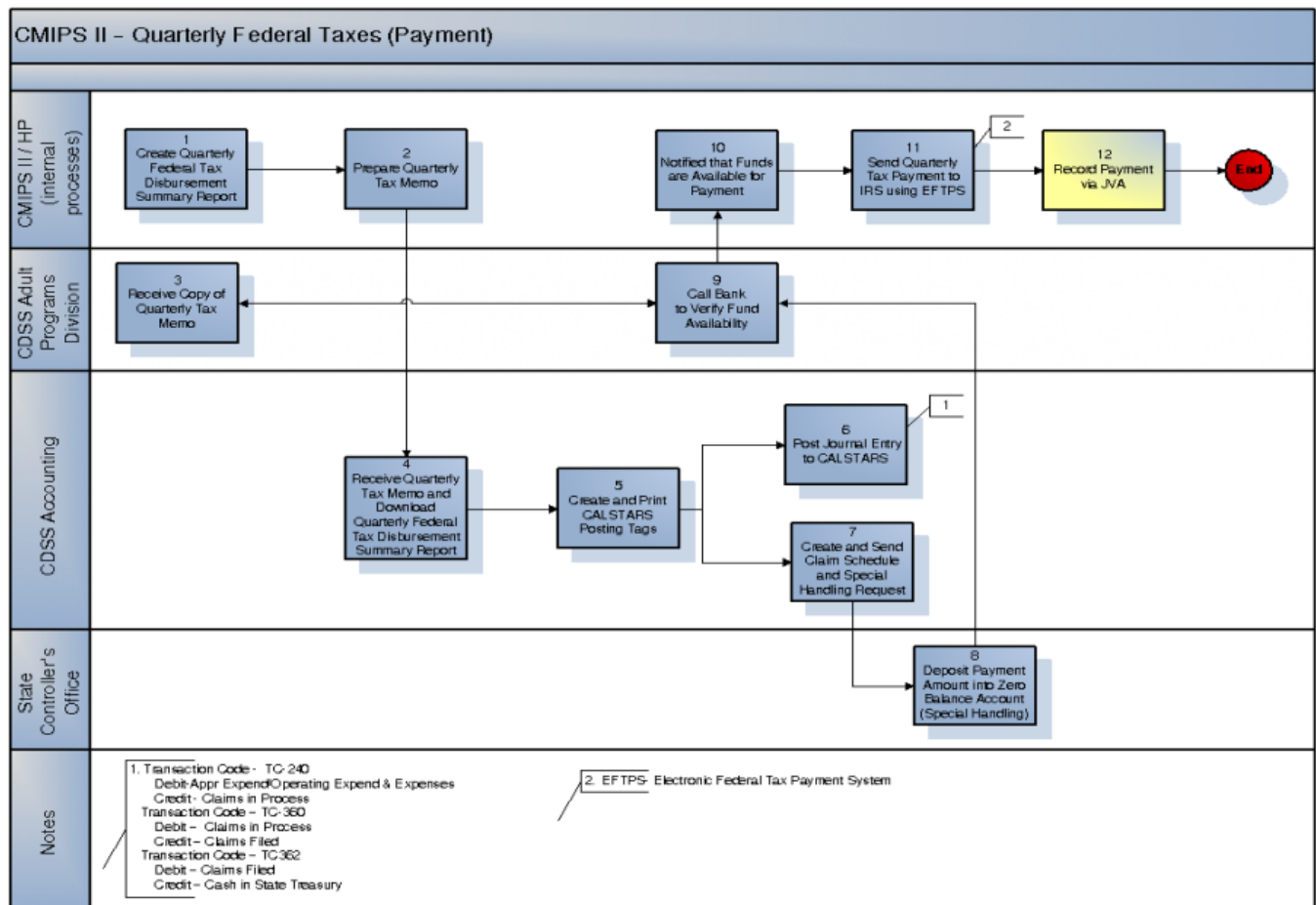


Figure – To Be Quarterly Federal Taxes

CI	Document Name
CI-116697 - DSD BP Quarterly State Taxes IMPLEMENTED	DSD_BP_Quarterly_State_Taxes.doc

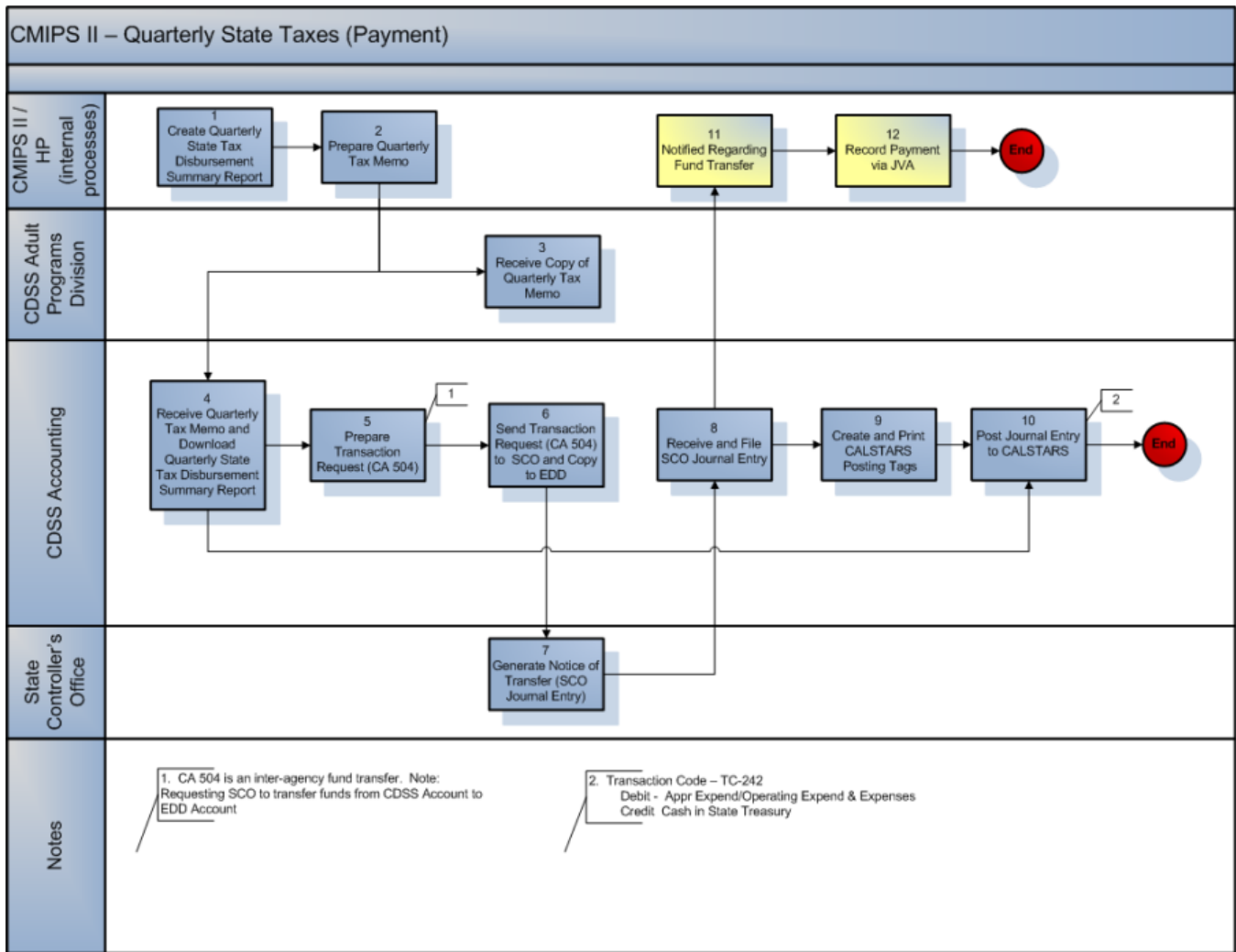



Figure – To Be Quarterly State Taxes

DSD 19/Payroll – Warrant Management/Business Process /Business Process Functions/Replacement and Reissue Warrants

CI	Document Name
 CI-69260 - DSD BF Replacement and Reissue Warrants IMPLEMENTED	DSD_BF_Replacement_and_Reissue_Warrants.doc

Void and Replacement - Provider/Recipient Payroll Warrants

Replacement warrant transactions are processed when a Provider/Recipient notifies the county that his/her warrant has been lost, stolen, destroyed or never received. The County Staff keys the transactions that will both stop payment on the original warrant and request a replacement warrant. The replacement warrant that is issued is an exact duplicate of the original warrant with the same warrant number and a new date.

The Case Management (CM) screens that will be used to initiate the void and replacement of the warrant can be found in the Application Screens section of this document. In order for the warrant to be replaced, a Request of Duplicate Controller's Warrant (STD435) must be submitted to the SCO. The STD435 is an existing form that will not be changed in CMIPS. Once SCO has issued the replacement warrants, a report is returned to the Contractor listing the replaced warrants and replacement date.

For payroll warrants that are voided and replaced, no updates are made in Advantage.

Void and Reissue - Provider/Recipient Payroll Warrants

Reissue warrant transactions are processed when a Provider/Recipient notifies the County that their warrant has been damaged and provides the evidence of the damaged warrant to the County Staff. The County staff keys a request for a void and subsequent reissue on the CM screen. The request for the warrant to be voided goes to SCO on the Void/Stop Payment report. The reissue transaction goes through the normal nightly cycle. The resulting reissued warrant is issued from SCO with a new warrant number.

For payroll warrants that are voided and reissued, the following updates will be made via an internal interface from CM. A CHCK document will be generated to make the following updates:

- Update the original Warrant Status field to "Redeposited-Stopped" in the PAY_CHECK table.
- Update the Replace Check field on the original warrant to "Replace" in the PAY_CHECK table.
- The original accounting entry will be reversed.
- A new warrant is issued to the Provider/Recipient with a new warrant number.
- SOC is taken again (against the reissued warrant).
- The following updates will be made via an external interface from SCO:
- Update the warrantStatusDate field in the Undeliverable Update Warrant file that is sent to CM based on the date the void was processed by SCO.

Void and Reissue - Vendor warrants

Vendor warrants can be returned to the Contractor due to a bad address. For vendor warrants that need to be voided and reissued, a Disbursement Cancellation (DC) document will be generated via an external interface from SCO. When necessary, the Contractor will manually reissue the warrant in the Financial component of CMIPS.

For vendor warrants that are voided via the DC document, the following updates will occur:

- Update the original Warrant Status field to "Void" in the Check Reconciliation (CHREC) table.
- Update Last Action Date in the CHREC table based on the date the original warrant was processed by SCO.
- The original accounting entry will be reversed.
- When necessary, a warrant request is manually generated and a warrant is issued to the vendor with a new warrant number.

FICA refund warrants are processed once per year. There are instances where these warrants may need to be replaced. For example, if the payee has moved or their bank EFT information has changed, they would need to communicate the new information to the County and request a replacement warrant. The request for replacement FICA refund warrant is made in the same way a Provider or Recipient would request a replacement for a regular paycheck. The Provider or Recipient's address and/or bank information needs to be updated and then the replacement request is made on the CM screen.

Business Process Flow – Replacement and Reissue Warrants


This section is not applicable for Warrant Management.

Initiation/Triggers, Pre-Conditions and Post Conditions

This section is not applicable for Warrant Management.

Process/Screen Flow

The following diagrams describe the process flow for Replacement Warrants.

CI	Document Name
  CI-116703 - DSD BP Void Replacement Warrant Payroll Warrants IMPLEMENTED	DSD_BP_Void_Replacement_Warrant_Payroll_Warrants.doc

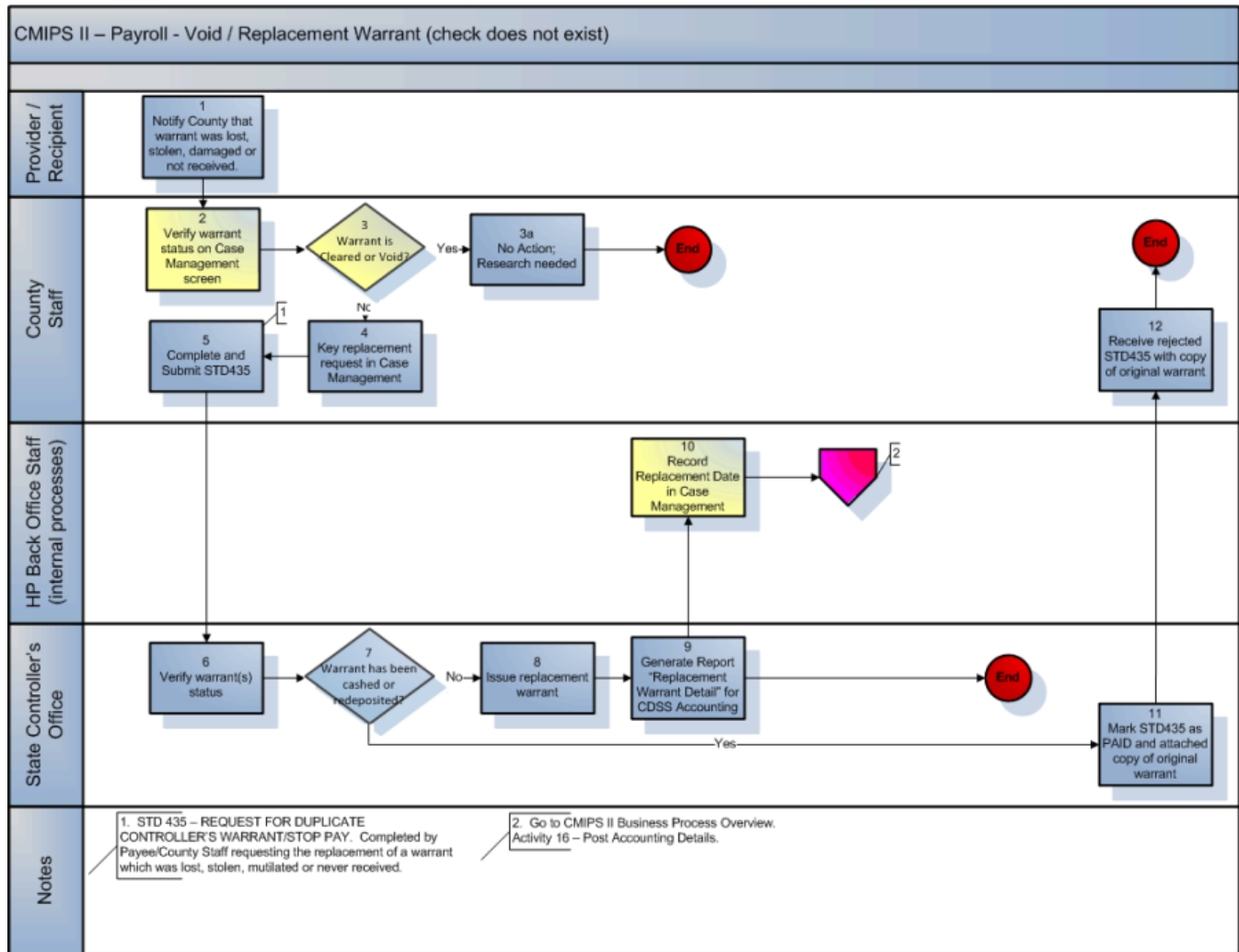



Figure – Void/Replacement Warrant – Payroll Warrants

CI	Document Name
<div> <div>  CI-116702 - DSD BP Void Reissue Warrant Payroll Warrants </div> <div>IMPLEMENTED</div> </div>	DSD_BP_Void_Reissue_Warrant_Payroll_Warrants.doc

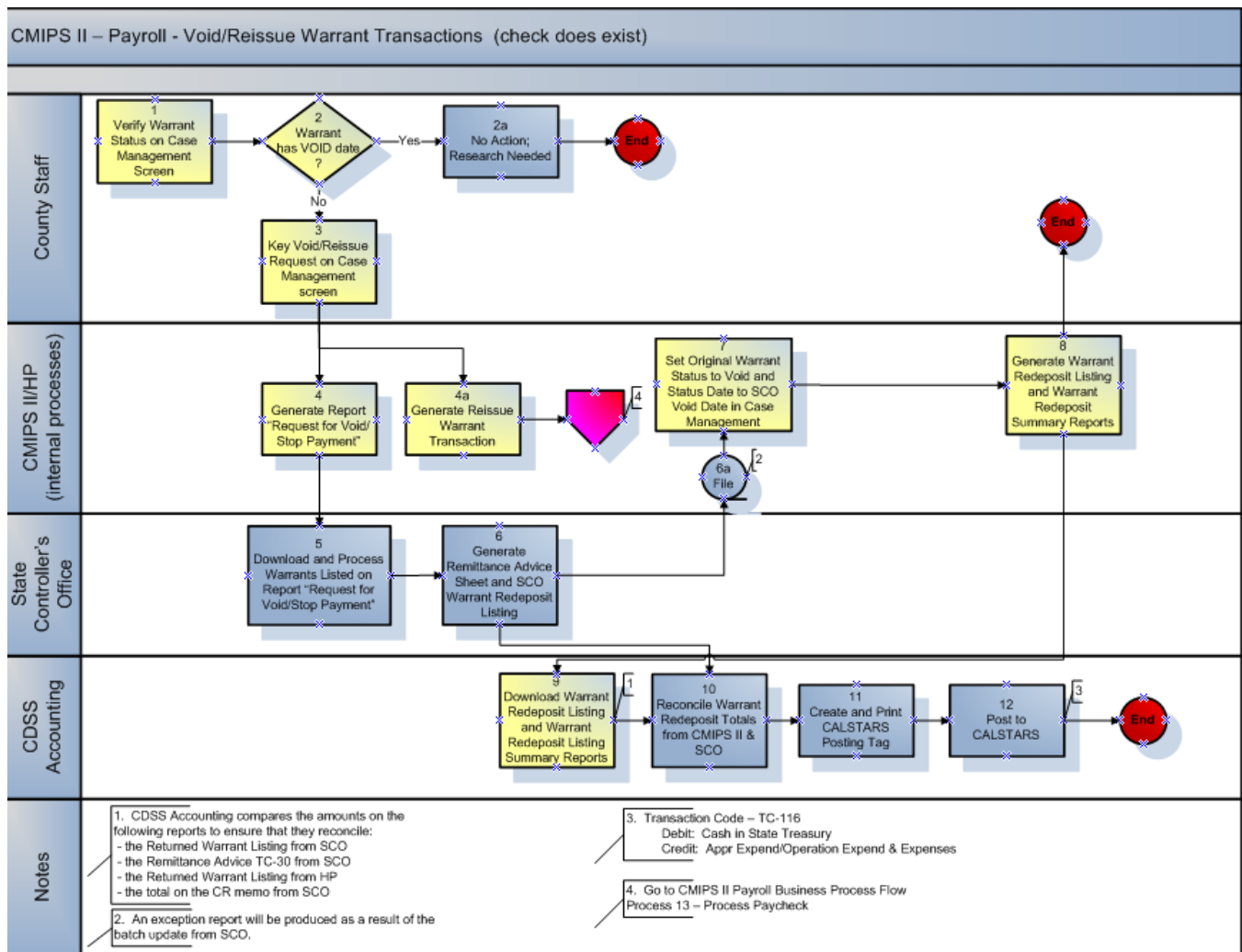


Figure – Void/Reissue Warrant – Payroll Warrants

CI	Document Name
<div> <div>CI-116704 - DSD BP Void Replacement Warrant Vendor Warrants</div> <div>IMPLEMENTED</div> </div>	DSD_BP_Void_Replacement_Warrant_Vendor_Warrants.doc

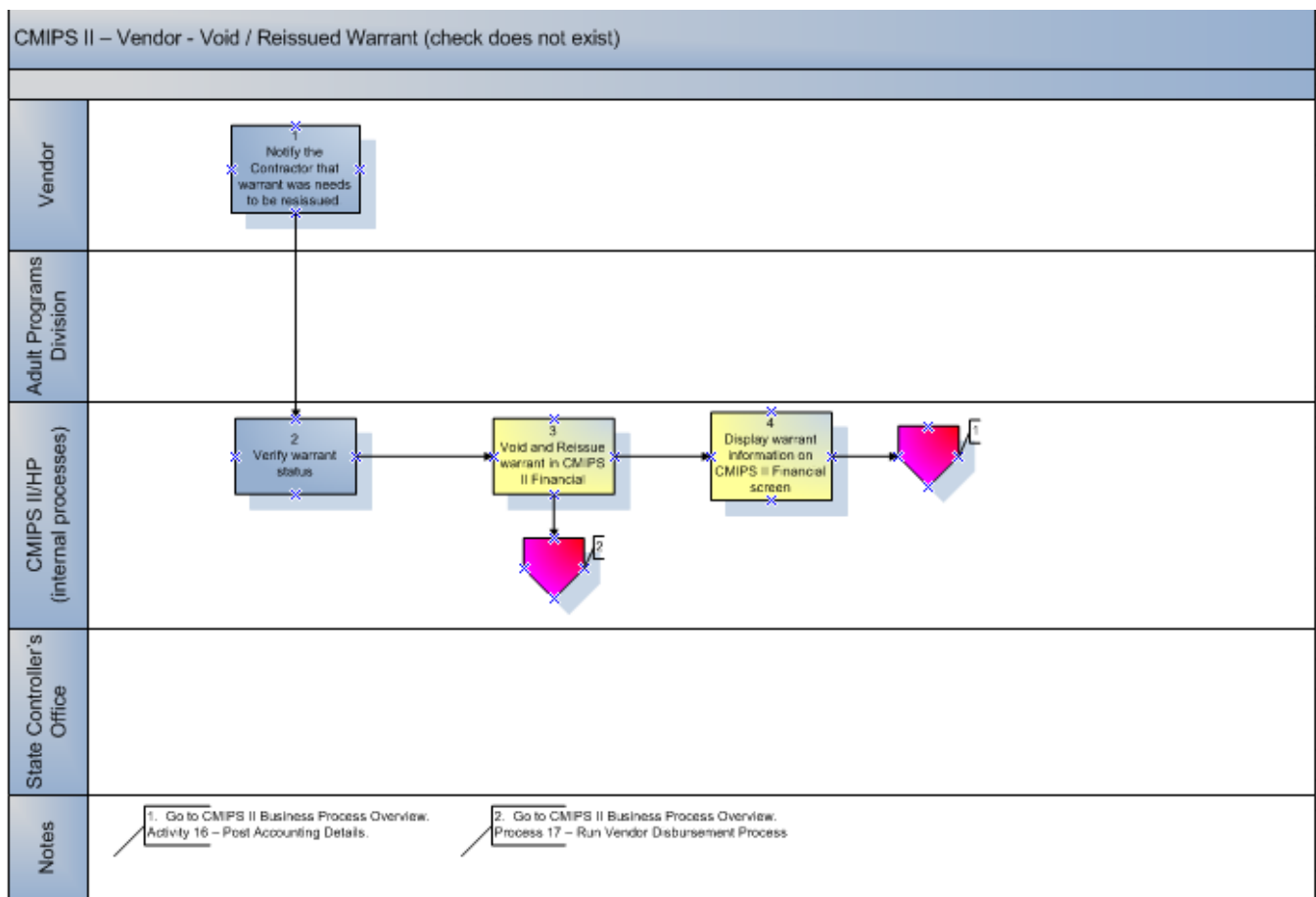


Figure – Void/Replacement Warrant – Vendor Warrants

DSD 19/Payroll – Warrant Management/Business Process /Business Process Functions/Stop Payment and Redeposit Warrants

CI	Document Name
CI-69272 - DSD BF Stop Payment and Redeposit Warrants IMPLEMENTED	DSD_BF_Stop_Payment_and_Redeposit_Warrants.doc

Stop payment transactions are processed at the request of Counties for warrants which have been returned because the payee is ineligible. In this business process, the warrant will not be replaced and SCO will redeposit the funds. Undeliverable warrants are also returned directly to SCO and the funds are redeposited. These warrants may be reissued at some point in the future if the payee submits a corrected address.

The CM screens that will be used to initiate the stop payment of the warrant can be found in the Application Screens section of this document. The Request for Void/Stop Payment report, which lists all warrants that need a stop payment issued, will be sent to SCO. SCO will void the warrants on the Request for Void/Stop Payment report as well as the warrants that are returned directly to SCO as undeliverable. A file will be returned from SCO that will be used to update the warrant information in CMIPS. If after the warrant is voided it becomes necessary to reissue the warrant, the request will be keyed in CM and the warrant will be reissued through the normal nightly cycle.

The content and design of the report "Request for Void/Stop Payment" can be found in the Reports section of this document.

Provider/Recipient Warrants – Stop Payment Initiated by County.

Payroll warrants will be voided via an external interface from SCO. A CHCK document will be generated to make the following updates:

For Service Hours:

- Update the Warrant Status field to "Redeposited-Cancelled" in the PAY_CHECK table.
- Update the Replace Check field to "N/A" in the PAY_CHECK table.
- Update the Last Action Date field in the PAY_CHECK table based on the date the void was processed by SCO.
- The accounting entry will be reversed.
- The hours are returned to the case.
- SOC is sent back to CM. If the recipient is uncertified, the SOC reversal is attempted. If the recipient is certified or reversal is not allowed, then the SOC is held for any provider for that recipient in that service month.

For Travel Claim or Travel Claim – Supplemental:

- Update the Warrant Status field to "Redeposited-Cancelled" in the PAY_CHECK table.
- Update the Replace Check field to "N/A" in the PAY_CHECK table.
- Update the Last Action Date field in the PAY_CHECK table based on the date the void was processed by SCO.
- The accounting entry will be reversed.
- The travel hours are returned to the provider/case.
- Update the Travel Claim Status

There will need to be an update if there is a change in the processing of voids.

Provider/Recipient Warrants – Undeliverable Warrants Redeposited by SCO

Redeposited payroll warrants will be voided via an external interface from SCO. A CHCK document will be generated to make the following updates:

For Service Hours:

- Update the Warrant Status field to "Redeposited-Cancelled" in the PAY_CHECK table
- Update the Replace Check field to "N/A" in the PAY_CHECK table
- Update the Last Action Date field in the PAY_CHECK table based on the date the void was processed by SCO
- Reverse the accounting entry
- Send SOC back to CM
- Return the hours to the case

For Travel Claim or Travel Claim – Supplemental:

- Update the Warrant Status field to "Redeposited-Cancelled" in the PAY_CHECK table
- Update the Replace Check field to "N/A" in the PAY_CHECK table
- Update the Last Action Date field in the PAY_CHECK table based on the date the void was processed by SCO
- Reverse the accounting entry
- Return the travel hours to the provider/case
- Update the Travel Claim Status

Vendor Warrants

Vendor warrants will be voided via an external interface from SCO.

When the DC document is submitted via the interface to redeposit a warrant without replacing it, the following occurs:

- Updates the warrant status field in the Check Reconciliation table (CHREC) with a status of 'Void.'

- Update Last Action Date in the CHREC table based on the date the warrant was processed by SCO.
- Updates the Cancellation Type and Cancellation Reason in the CHREC table. The Cancellation Type will be "Close" and the Cancellation Reason will be set to "Cancel."
- Reverses the original accounting entry.

If the vendor warrant that was redeposited was a lien holder warrant, the Contractor must complete an additional step that will return the amount to the Provider or Recipient from which it was deducted. In CMIPS HRM, an OTDED must be processed on the deduction screen to return that amount to the Provider or Recipient's net pay. The situation also could occur when the lien holder has cashed the warrant, realized that the money is no longer owed and writes a warrant back to CDSS to return the money. In this case, the Contractor must process an OTDED to return the money to the Provider or Recipient from which it was deducted. No further updates are required in the financial component of CMIPS.

The SCO will send a listing of all warrants that have been redeposited to CDSS Accounting. CMIPS will produce a report of all warrants that have been redeposited. CDSS Accounting staff will reconcile the two reports to verify that all redeposited warrants have been accounted for. See the Reports section of this document for the layout of the Warrant Redeposit Report.

Business Process Flow – Stop Payment and Redeposit Warrants

This section is not applicable for Warrant Management.

Initiation/Triggers, Pre-Conditions and Post Conditions

This section is not applicable for Warrant Management.

Process/Screen Flow

The following diagrams describe the process flow for Warrant Management.

CI	Document Name
CI-116700 - DSD BP Void Redeposit Warrant Payroll Warrants IMPLEMENTED	DSD_BP_Void_Redeposit_Warrant_Payroll_Warrants.doc

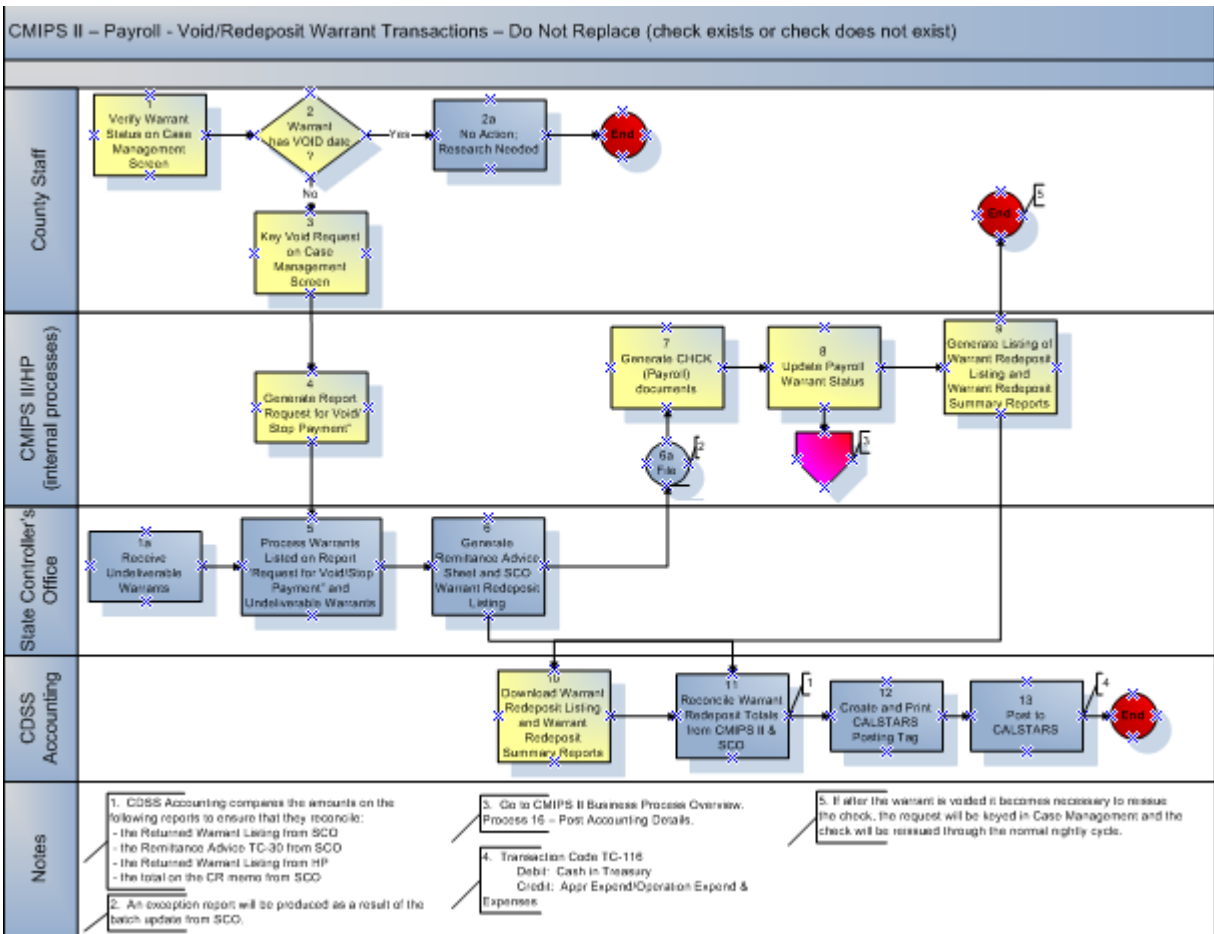


Figure – To Be Void/Redeposit Warrant – Payroll Warrants

CI	Document Name
CI-116701 - DSD BP Void Redeposit Warrant Vendor Warrant IMPLEMENTED	DSD_BP_Void_Redeposit_Warrant_Vendor_Warrant.doc

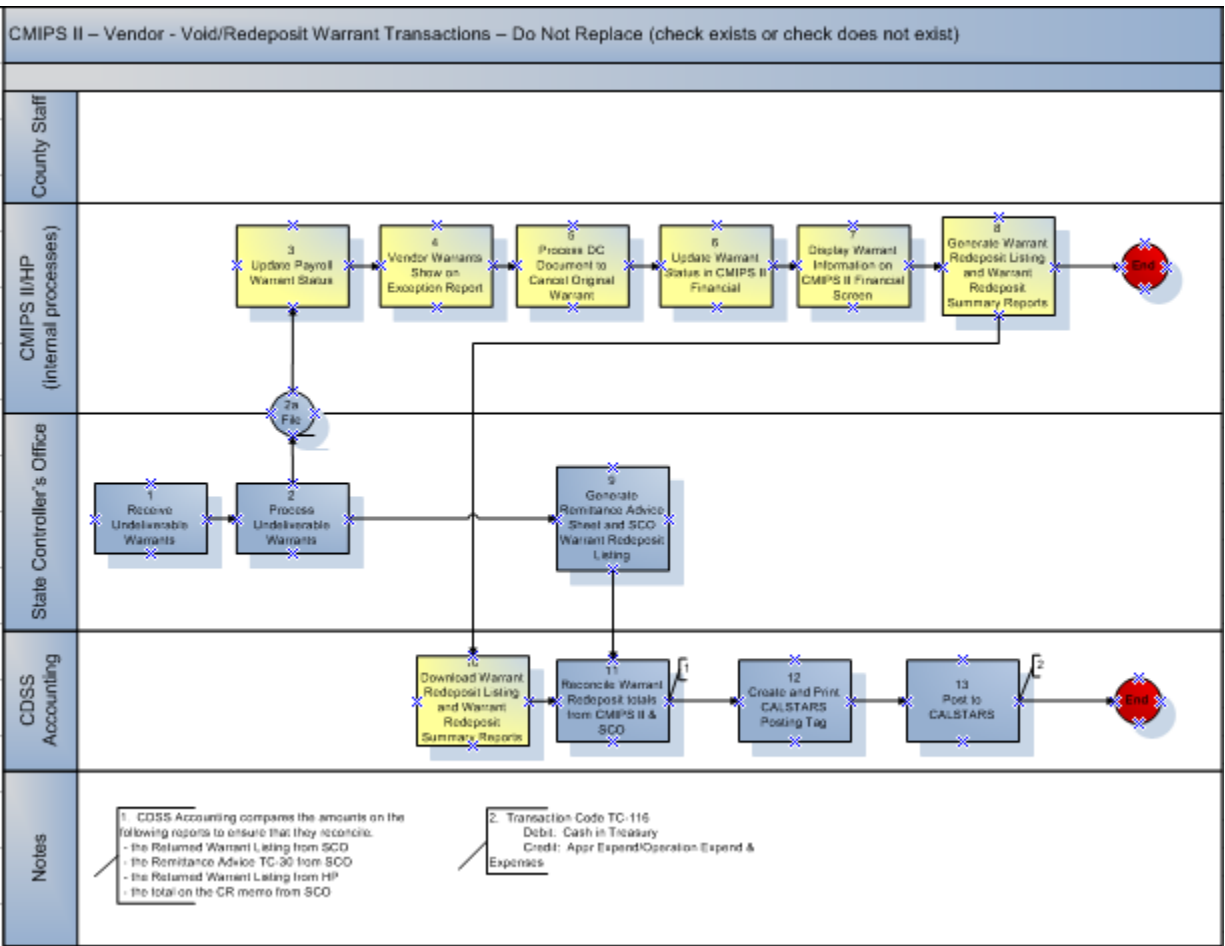


Figure – To Be Void/Redeposit Warrant – Vendor Warrant

DSD 19/Payroll – Warrant Management/Business Process /Business Process Functions/Stale Dated Warrants

CI	Document Name
CI-69265 - DSD BF Stale Dated Warrants IMPLEMENTED	DSD_BF_Stale_Dated_Warrants.doc

When a warrant is issued but not cashed within 13 months, it is considered a stale dated warrant. SCO will stale date these warrants and return a file that will update the warrant information. If after the warrant is stale dated it becomes necessary to reissue the warrant, the Provider/Recipient must submit a STD435 to the Contractor. If the warrant has not been cashed, the Contractor will notify CDSS Accounting that a replacement warrant is needed for the stale dated warrant. CDSS Accounting will produce a claim schedule for SCO and SCO will generate a warrant and send directly to the Provider/Recipient. If the replacement of a stale dated warrant is requested more than two years after the warrant was stale dated, then a request needs to be filed with the Crime Victims Compensation and Government Claims Board.

Warrants are stale dated by SCO when they are 13 months old. The details of this interface can be found in the Interfaces/Batch Processes section of this document.

Provider/Recipient warrants

For payroll warrants that are stale dated via an external interface from SCO, the new warrant status of "Stale Dated" will be passed to Case Management along with the date the warrant was stale dated. Case Management will store the updated status and date but display the status of "Paid."

Vendor warrants

Vendor warrants will be stale dated with a Disbursement Correction (DC) document which will be generated via an external interface from SCO. Vendor warrants exist only in Payroll and are not loaded to Case Management.

When the DC document is submitted via the interface to stale date a warrant, the following occurs:

- Updates the warrant status field in the Check Reconciliation table (CHREC) with a status of "Stale Dated."
- Update the Last Action Date field in the CHREC table based on the date the stale date was processed by SCO.
- Updates the Cancellation Type and Cancellation Reason in the CHREC. The Cancellation Type will be "Closed" and the Cancellation Reason will be "Stale Dated."
- The accounting entry will be reversed.

Business Process Flow – Stale Dated Warrants

This section is not applicable for Warrant Management.

Initiation/Triggers, Pre-Conditions and Post Conditions

This section is not applicable for Warrant Management.

Process/Screen Flow

The following diagrams describe the process flow for Stale Dated Warrants.

CI	Document Name
CI-116698 - DSD BP Stale Date Warrants Interface from SCO IMPLEMENTED	DSD_BP_Stale_Date_Warrants_Interface_from_SCO.doc

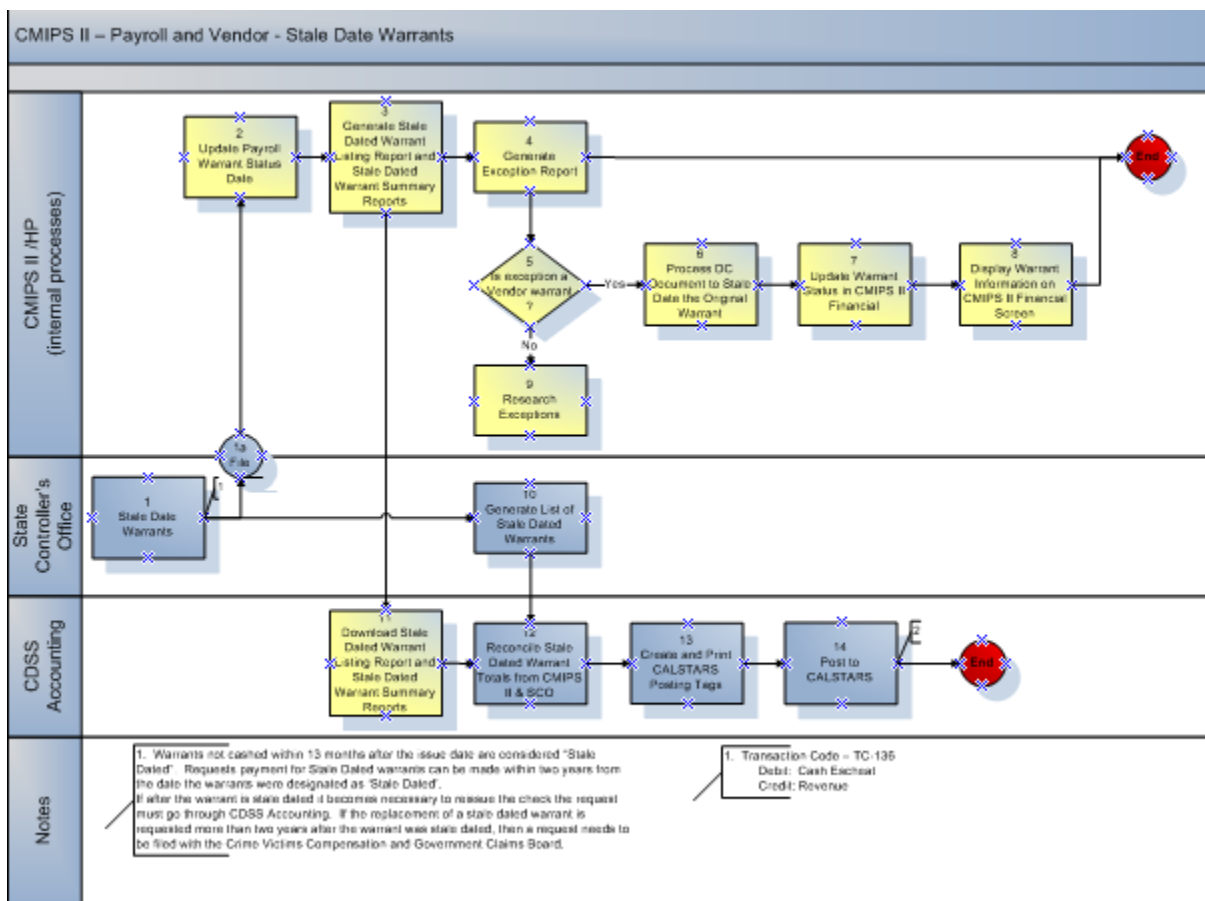


Figure – To Be State Date Warrants – Interface from SCO

CI	Document Name
CI-116699 - DSD BP State Date Warrants Replacement Request from Payee IMPLEMENTED	DSD_BP_State_Date_Warrants_Replacement_Request_from_Payee.doc

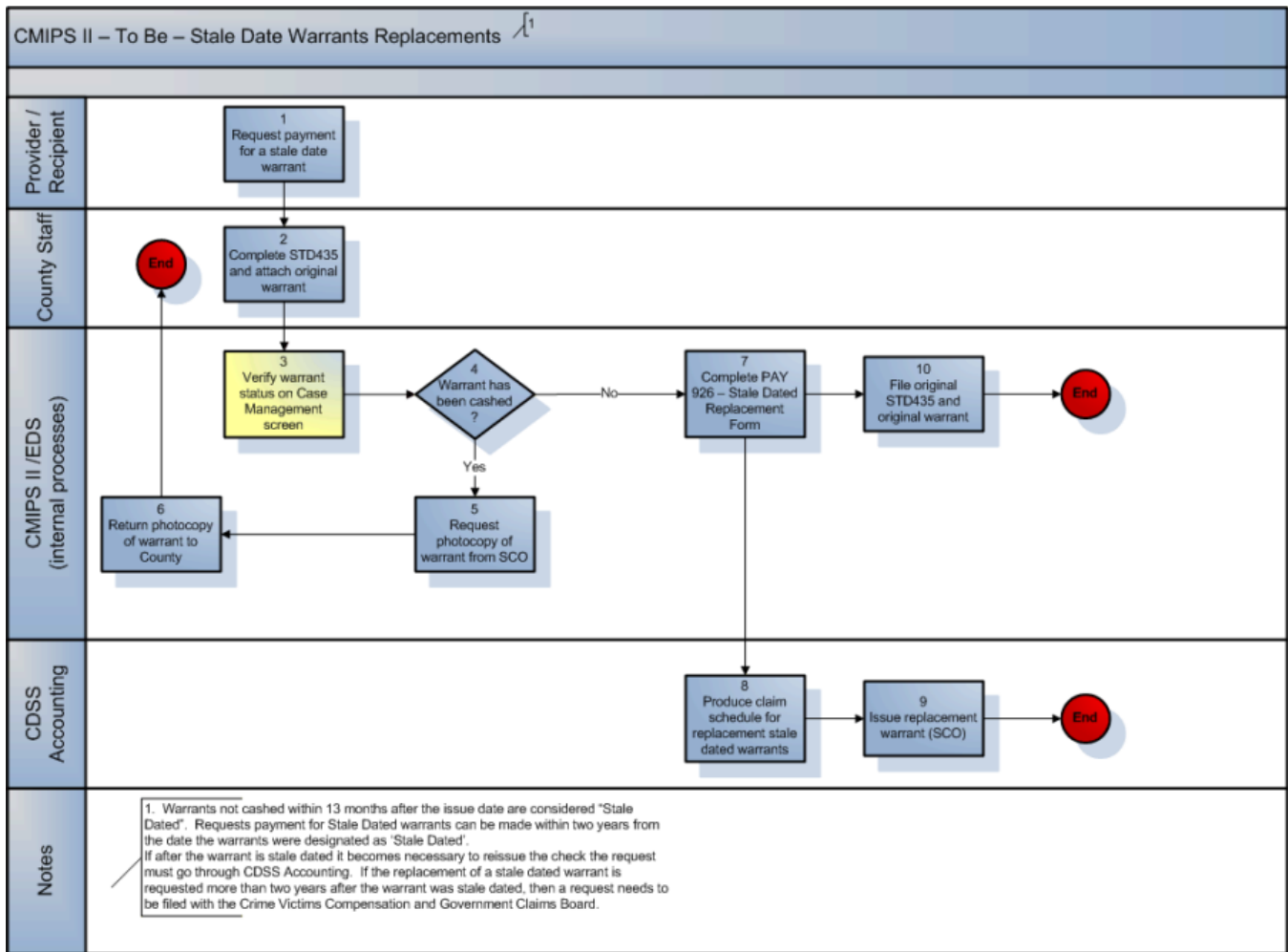



Figure – To Be Stale Date Warrants – Replacement Request from Payee

DSD 19/Payroll – Warrant Management/Business Process /Business Process Functions/Warrant/Funds Mgt Warrant Cross Reference

CI	Document Name
 CI-69266 - DSD BF Warrant Funds Mgt Warrant Cross Reference IMPLEMENTED	DSD_BF_Warrant_Funds_Mgt_Warrant_Cross_Reference.doc

Advantage Financial

To accommodate the requirements for CMIPS, the following changes were incorporated into the baseline Advantage Financial:

- Modify the Check Reconciliation table - add two new fields (SCO Warrant Number and SCO Warrant Date).
- Modify the Paid Check table - add two new fields (SCO Warrant Number and SCO Warrant Date).
- Create a new query page that will display the Advantage Warrant Number and Advantage Warrant Date that will be mapped to the SCO Warrant Number and SCO Warrant Date.
- Modify the Check Reconciliation process to move the SCO Warrant Number and SCO Warrant Date from the Check Reconciliation Table to the Paid Check table.
- Modify the Paid Check Restore functionality to move the values from the Paid Check table to the Check Reconciliation table.

The new local interface discussed in this paragraph will be supported by the CMIPS on-site interface team. One new local interface will process the data file sent from SCO. This file will contain the SCO Warrant Numbers, Advantage Warrant Numbers, and Advantage Bank Account Numbers that will allow the local interface to identify the warrant records in the Financial Check Reconciliation table. The local interface will then populate the SCO Warrant Number and SCO Warrant Date in the Financial Check Reconciliation table (CHREC).

Advantage Payroll

To accommodate the requirements for CMIPS, baseline Advantage HRM was modified to store the cross-reference between SCO and Advantage Warrant Numbers. The Paycheck (warrant) Data table in HRM will be modified to add the SCO Warrant Number and SCO Warrant Date. The same two fields will also be added to the cloned Paycheck Data tables (C_PAY_CHECK and P_PAY_CHECK) to keep the tables in sync. It should be noted that the Advantage Warrant Number consists of the Bank Account Number and the Check Number.

There will be a new online page, Warrant Cross Reference (WCRE) to display the Advantage Warrant Number (defined as Varchar2 15), Advantage Warrant Date, SCO Warrant Number (defined as Varchar2 30), and SCO Warrant Date. Users may navigate to this new page by inputting "WCRE" in the Navigational box or via Page Search. Users may search for the Advantage Warrant Number using the SCO Warrant Number and SCO Warrant Date. The search will return an exact match using the entered SCO Warrant Number and SCO Warrant Date. If an exact match is not found, an error will display. If either the SCO Warrant Number or SCO Warrant Date is not entered, an error will also display.

Currently, baseline HRM Check Print job produces the paycheck data file (.DAT file) and FTPs it to an FTP server. This functionality will stay the same. The local interface (discussed below) can pick up this file and forwards it to SCO.

The new local interface discussed in this paragraph will be supported by the CMIPS on-site interface team. One new local interface will process data file sent from SCO. This file will contain the SCO Warrant Numbers and Advantage Warrant Numbers, which will allow the local interface to identify the warrant records in the HRM Paycheck Data table. The local interface will then populate the SCO Warrant Number and SCO Warrant Date in the HRM Paycheck Data table (PAY_CHECK table).

The baseline Check Archiving job already archives the Paycheck Data table and hence, no modification is needed for this job.

Advantage Financial will have similar processing and a new inquiry page as in Advantage HRM. There will be no shared table between Advantage HRM and Advantage Financial. Please refer to the Advantage Financial functional design for this enhancement for further details.

Business Process Flow – Warrant/Funds Mgt Warrant Cross Reference

This section is not applicable for Warrant Management.

Initiation/Triggers, Pre-Conditions and Post Conditions

This section is not applicable for Warrant Management.

Process/Screen Flow

Advantage Financial

The following Business Scenarios are examples of how the current processes can be impacted or improved by this modification.

Provider or Recipient Warrant information sent to SCO

In this scenario, after the Check Print job completes, the Check Reconciliation table will have the following fields:

Check Reconciliation Table with Warrant Information	
Field Name	Value
Bank Account Code	9099999
Warrant Number	100001
Warrant Amount	2500.00
Warrant Date	11-01-2008
Status	Warranted
Issue Date	<system date>
SCO Warrant Number	<blank>
SCO Warrant Date	<blank>

The local interface will pick up this file and send it to SCO. SCO will proceed with sending warrants to Recipients and Providers.

SCO Warrant information sent from SCO

Continuing from above scenario: after SCO sends warrants to Recipients and Providers, the local interface will update the Check Reconciliation table with the SCO Warrant Number and SCO Warrant Date. The table will have the updated fields as follows:

Check Reconciliation Table with Warrant Information	
Field Name	Value
Bank Account Code	9099999
Warrant Number	100001
Warrant Amount	2500.00
Warrant Date	11-01-2008
Status	Warranted
Issue Date	<system date>
SCO Warrant Number	3302938098990
SCO Warrant Date	11-02-2008

Advantage Payroll

The following Business Scenarios are examples of how the current processes can be impacted or improved by this modification.

Provider or Recipient Warrant information sent to SCO

In this scenario, after daily Payroll processing completes, the Paycheck Data file will have the following fields:

Paycheck Data File with Paycheck Information	
Field Name	Value
Employee ID	123456789
SSN	123-45-6789
First Name	John
Last Name	Doe
Warrant Number	100001
Warrant Amount	2500.00

Warrant Date	11-01-2008
Disposition	Outstanding
Disposition Date	<system date>
SCO Warrant Number	<blank>
SCO Warrant Date	<blank>

The local interface will pick up this file and send it to SCO. SCO will proceed with sending warrants to Recipients and Providers.

SCO Warrant information sent from SCO

Continuing from above scenario, after SCO sends warrants to Recipients and Providers, the local interface will update the Paycheck Data file with the SCO Warrant Number and SCO Warrant Date. The table will have the updated fields as follows:

Paycheck Data File with Paycheck and SCO Information	
Field Name	Value
Employee ID	123456789
SSN	123-45-6789
First Name	John
Last Name	Doe
Warrant Number	100001
Warrant Amount	2500.00
Warrant Date	11-01-2008
Disposition	Outstanding
Disposition Date	<system date>
SCO Warrant Number	3302938098990
SCO Warrant Date	11-02-2008

Locate Advantage Warrant Number using the WCRE Screen

Continuing from above scenario: the table now has the fields populated for check number 100001.

The user can locate the Advantage Warrant Number by navigating to the new page WCRE and searching for SCO Warrant Number 3302938098990 and SCO Warrant Date of 11-02-2008. The page will display the following:

Warrant Cross Reference Page	
Field Name	Value
Warrant Number	100001
Warrant Date	11-01-2008
SCO Warrant Number	3302938098990
SCO Warrant Date	11-02-2008

DSD 19/Payroll – Warrant Management/Business Process /Business Process Functions/County Contractor

Once the County Contractor invoice data is recorded into CMIPS, the financial details are posted in CMIPS Financial via journal voucher during the nightly cycle. Once the journal voucher is processed, the County Contractor financial data becomes available for reporting needs.

Business Process Flow – County Contractor

This section is not applicable for Warrant Management.

Initiation/Triggers, Pre-Conditions and Post Conditions

This section is not applicable for Warrant Management.

Process/Screen Flow

This section is not applicable for Warrant Management.

DSD 19/Payroll – Warrant Management/Business Process /Business Process Functions/Homemakers

CI	Document Name
CI-117898 - DSD BF Homemakers IMPLEMENTED	DSD_BF_Homemakers.doc

Once the Homemakers' data is recorded into CMIPS, the financial details are posted in CMIPS Financial via journal voucher during the nightly cycle. Once the journal voucher is processed, the Homemakers' financial data becomes available for reporting needs.

Business Process Flow - Homemakers

This section is not applicable for Warrant Management.

Initiation/Triggers, Pre-Conditions and Post Conditions

This section is not applicable for Warrant Management.

Process/Screen Flow

This section is not applicable for Warrant Management.

DSD 19/Payroll – Warrant Management/Business Process /Business Process Functions/General Ledger

CI	Document Name
CI-69277 - DSD BF General Ledger IMPLEMENTED	DSD_BF_General_Ledger.doc

Financial component of CMIPS is Generally Accepted Accounting Principles (GAAP) compliance.

The General Ledger stores the payroll and warrant details at the summarization level. For each payroll run, the General Ledger will have one record for all payroll details with the same financial elements. For example, every payroll record for that pay cycle that has the same Fiscal Year, Fund, Department, Unit, Sub Unit, Appropriation, Object and Reporting Code will be summarized and posted as one entry. The financial elements allow CMIPS to capture data at the Federal, State, county, district and program levels and store the data on the General Ledger.

All types of transactions will be stored on the General Ledger whether it is a payroll warrant, adjustment to payroll, correction of overpayment, advance pay, lien holder payments, public authorities payments, labor organization payments, tax payments, FICA refunds or County Contractor and Homemakers payments.

The General Ledger will store all the changes made to programs including the movement of activities between programs, the reducing of program cost due to overpayment recovery, redeposit, stale date warrant or reversing incorrect program cost.

The CMIPS data is stored on the General Ledger at the Federal, State, county, district and program levels, which allows for support of an internal or external program audit and detailed reporting.

Business Process Flow – General Ledger

This section is not applicable for Warrant Management.

Initiation/Triggers, Pre-Conditions and Post Conditions

This section is not applicable for Warrant Management.

Process/Screen Flow

This section is not applicable for Warrant Management.

DSD 19/Payroll – Warrant Management/Business Process /Business Process Functions/Funding Management

CI	Document Name
CI-69659 - DSD BF Funding Management IMPLEMENTED	DSD_BF_Funding_Management.doc

Fund Management will assist CDSS in easily tracking and managing the CMIPS programs since the program data will be stored at the Federal, State, county, district and program levels. CDSS will have the ability to track and report on program sources and program sub-sources and to provide detailed reports on the programs at whatever level is necessary. For example, CDSS bills the county for its SOC for PCSP. This data is now easily attainable since the data is captured at the program level, county level and at the correct funding percentage for each program. CDSS Accounting and the County staff will have detailed information about what was charged to each program to assist with any reporting needs.

CDSS will have the ability to easily reconcile the payments and deductions. Below are several examples:

Quarterly payments made to IRS and EDD will reconcile to the Tax Disbursement Report and the Tax Disbursement Report will reconcile to the Payment Voucher Report that is created quarterly.

Monthly health benefits payments to Public Authorities will reconcile with the detail report listing the Provider and amount. Also, this payment will reconcile to the Payment Voucher Report that is created monthly.

Monthly payments to labor organizations will reconcile with the detail report listing the Provider and amount. Also, this payment will reconcile to the Payment Voucher Report that is created monthly.

The CDSS billings to the Federal government and county governments will have detailed backup to support each billing. The Payment Voucher Report will provide the exact amount due from each government based on the program and funding percentage.

The wealth of information now available will ensure that CDSS has the tools to assist in developing the program budget, reporting and supporting any audit request.

Business Process Flow – Funding Management

This section is not applicable for Funding Management.

Initiation/Triggers, Pre-Conditions and Post Conditions

This section is not applicable for Funding Management.

Process/Screen Flow

This section is not applicable for Funding Management.

DSD 19/Payroll – Warrant Management/Business Process /Business Process Functions/Fiscal Year Fund Maintenance

CI	Document Name
CI-69278 - DSD BF Fiscal Year Fund Maintenance IMPLEMENTED	DSD_BF_Fiscal_Year_Fund_Maintenance.doc

The New Year Table Initialization (NYTI) batch process needs to be run in April each year to populate the tables with values for a coming fiscal year. This allows time for new programs elements to be established and to verify that the new table data is correct for the new fiscal year before July 1.

Annual Closing Process is run each year after the end of a year and after the documents from the prior years is finalized and the Accounting Periods have been closed for that year. The purpose of the Annual Closing Process is to facilitate the movement of account balances from the accounting fiscal year that is being closed to the subsequent accounting fiscal year.

Business Process Flow – Fiscal Year Fund Maintenance

This section is not applicable for Fiscal Year Fund Management.

Initiation/Triggers, Pre-Conditions and Post Conditions

This section is not applicable for Fiscal Year Fund Management.

Process/Screen Flow

This section is not applicable for Fiscal Year Fund Management.

DSD 19/Payroll – Warrant Management/Business Process /Screen Designs

The screens that will be used for this functionality are described in [Recip CM & OS – Payroll within Case Management](#).



DSD 19/Payroll – Warrant Management/Business Process /Screen Designs/Financial Data Elements

The data elements described in the subsections that follow are specific to financial configuration and have been customized for CMIPS. Refer to Appendix C for baseline Advantage configuration elements.

If, in the table layout, the Value column is left empty, it is inferred that the entered value will be based on business rules. Refer to the Comments column for additional information regarding the valid choices or examples for the Value.

The Comments column will start with one of the following five items: Required, Rule Driven, Blank, Default or Optional. These are to be considered when creating a new transaction.

Check Reconciliation with SCO Warrant Information

CI	Document Name
 CI-124524 - DSD CFGN Check Reconciliation with SCO Warrant Information 	DSD_CFGN_Check_Reconciliation_with_SCO_Warrant_Information.doc

The Check Reconciliation table was customized for CMIPS to include SCO warrant information.

Table – Check Reconciliation (CHREC)

Table	Field	Value	Source	Comments
CHREC	Bank Account	1000	System Inferred	Default. The field indicates the bank account from which the check/EFT payments were issued.
CHREC	Check\ EFT	0010000 00000000	System Inferred	Default. This field provides the Disbursement instrument number. If the Disbursement is made through a check, then this field provides the Check number. If the Disbursement is made through Electronic Fund Transfer, then this field provides the EFT number. On the Check Reconciliation (CHREC) table, this field is updated by the Disbursement documents.
CHREC	Amount	\$302.26	System Inferred	Default. The dollar amount of the check or electronic fund transfer.
CHREC	Record Date	12/16 /2008	System Inferred	Default. The date the disbursement is recorded in CMIPS.
CHREC	Issue Date	12/16 /2008	System Inferred	Default. The day that represents when the check or EFT was issued for disbursement.
CHREC	Last Action	12/16 /2008	System Inferred	Default. The system-generated date indicating the date the last time this record was updated.
CHREC	Status	Paid	System Inferred	Default. Indicates the status of the check. Valid values include: Paid (issued, not cashed), Cleared (cashed), Void or Stale Dated.
CHREC	Cancellation Reason		System Inferred	Rule Driven. Indicates the reason for the cancellation of a check. Valid values in the drop down are Lost, Stolen, Never Received, Destroyed, Incorrect Information, Stale, Payee Ineligible, Cancel, Redeposit (system generated) or Damaged.
CHREC	Cleared Date		System Inferred	Rule Driven. In the case of a check, the Cleared Date is the date that the check cleared the bank. In the case of an EFT, the Cleared Date is the date the electronic payment became available at the payee's bank.
CHREC	Comments			Blank.
CHREC	Doc Code	AD	System Inferred	Default. The document code will be AD for paper warrant and EFT for electronic fund transfers.
CHREC	Doc Dept	5180	System Inferred	Default. The department code assigned to this document
CHREC	Doc ID	AD0000 0000000 0000015	System Inferred	Default. The document code and number that is either manually assigned or automatically generated by the system if you do not enter this information.
CHREC	SCO Warrant Number	74701423	Data Entry	Optional. Warrant Number assigned by SCO.
CHREC	SCO Warrant Date	10/04 /2010	Data Entry	Optional. Date the warrant was issued by SCO.


Illustrations – Budget and Disbursement in Advantage Financial

The next three figures illustrate the budget inquiry tables.

Appropriation Budget

	BFY	Fund	Department	Appr Unit	Current Budget	Encumbered	Actual Expenses	Unobligated
✓	2009	0001	5180	1112515	\$0.00	\$0.00	\$5,025.52	(\$5,025.52)

First Prev Next Last

[Search](#) 

▼Budget Actuals

Pre-Encumbered :	<input type="text" value="\$0.00"/>		Uncommitted :	<input type="text" value="(\$5,025.52)"/>
Encumbered :	<input type="text" value="\$0.00"/>		Unobligated :	<input type="text" value="(\$5,025.52)"/>
Accrued Expenses :	<input type="text" value="\$2,517.90"/>		Actual Expenses :	<input type="text" value="\$5,025.52"/>
Cash Expenses :	<input type="text" value="\$2,507.62"/>			

▼Budgeted Amounts

Adopted :	<input type="text" value="\$0.00"/>		Current Budget :	<input type="text" value="\$0.00"/>
Carry Forward :	<input type="text" value="\$0.00"/>		Amendments :	<input type="text" value="\$0.00"/>
Allocated :	<input type="text" value="\$0.00"/>		Transfers :	<input type="text" value="\$0.00"/>

▼Linked Revenues

[Link Collected Earned Revenue :](#)

▼General Information

BFY :	<input type="text" value="2009"/>	Name :	<input type="text" value="IHSS"/>
Fund :	<input type="text" value="0001"/>	Description :	<input type="text"/>
Department :	<input type="text" value="5180"/>	Active :	<input checked="" type="checkbox"/>
Appr Unit :	<input type="text" value="1112515"/>		

[Top](#)

[Modified Budget Line Controls](#) [Expense Budgets](#)

Figure – OC Appropriation Budget (BQ29LV1)

Detailed Transaction Listing

	Doc Code	Doc Dept	Doc ID	Document Identifier	Accrued Expenses
✓	GAX	5180	A TEST 5	GAX 5180 A TEST 5	\$1,999.99
	GAX	5180	A TEST 4	GAX 5180 A TEST 4	\$73.73
	GAX	5180	A-TEST 3	GAX 5180 A-TEST 3	\$250.50
	GAX	5180	A-TEST 2	GAX 5180 A-TEST 2	\$118.18
	GAX	5180	A-TEST1	GAX 5180 A-TEST1	\$75.50
	AD	5180	MAN000000000000000015	AD 5180 MAN000000000000000015	(\$302.26)
	GAX	5180	TEST 11	GAX 5180 TEST 11	\$250.75
	GAX	5180	TEST10	GAX 5180 TEST10	\$51.51
	AD	5180	MAN000000000000000014	AD 5180 MAN000000000000000014	(\$442.41)
	GAX	5180	TEST 7	GAX 5180 TEST 7	\$166.66
	GAX	5180	TEST 6	GAX 5180 TEST 6	\$275.75
	AD	5180	MAN000000000000000013	AD 5180 MAN000000000000000013	(\$388.88)
	AD	5180	MAN000000000000000012	AD 5180 MAN000000000000000012	(\$998.69)
	GAX	5180	TEST5	GAX 5180 TEST5	\$388.88
	GAX	5180	TEST-4	GAX 5180 TEST-4	\$732.71
	GAX	5180	TEST-3	GAX 5180 TEST-3	\$265.98

First Prev [Next](#) [Last](#) [Download](#)

[Search](#)

Doc Code :

Doc Dept :

Accrued Expenses

Doc ID :

Figure – Detail Transaction Listing (Appropriation Budget)


Expense Budget

BFY	Fund	Department	Appr Unit	Unit	Object	Current Budget	Encumbered	Actual Expenses	Unobligated
2009	0001	5180	1112515	IHSS	3005	\$0.00	\$0.00	\$0.00	\$0.00
2009	0001	5180	1112515	IHSS	4001	\$0.00	\$0.00	\$0.00	\$0.00
2009	0001	5180	1112515	IHSS	4002	\$0.00	\$0.00	\$0.00	\$0.00
2009	0001	5180	1112515	IHSS	4003	\$0.00	\$0.00	\$0.00	\$0.00
✓ 2009	0001	5180	1112515	IHSS	5001	\$0.00	\$0.00	\$3,651.45	(\$3,651.45)
2009	0001	5180	1112515	IHSS	5002	\$0.00	\$0.00	\$0.00	\$0.00
2009	0001	5180	1112515	IHSS	5003	\$0.00	\$0.00	\$0.00	\$0.00
2009	0001	5180	1112515	IHSS	6001	\$0.00	\$0.00	\$0.00	\$0.00

[First](#) [Prev](#) [Next](#) [Last](#)

[Search](#) 

▼Budget Actuals

Pre-Encumbered :	<input type="text" value="\$0.00"/>		Uncommitted :	<input type="text" value="(\$3,651.45)"/>
Encumbered :	<input type="text" value="\$0.00"/>		Unobligated :	<input type="text" value="(\$3,651.45)"/>
Accrued Expenses :	<input type="text" value="\$2,517.90"/>		Actual Expenses :	<input type="text" value="\$3,651.45"/>
Cash Expenses :	<input type="text" value="\$1,133.55"/>			

▼Budgeted Amounts

Adopted :	<input type="text" value="\$0.00"/>		Current Budget :	<input type="text" value="\$0.00"/>
Carry Forward :	<input type="text" value="\$0.00"/>		Amendments :	<input type="text" value="\$0.00"/>
Allocated :	<input type="text" value="\$0.00"/>		Transfers :	<input type="text" value="\$0.00"/>

▼Linked Revenues

[Link Collected Earned Revenue](#) :

▼General Information

BFY :	<input type="text" value="2009"/>	Name :	<input type="text" value="Wages"/>
Fund :	<input type="text" value="0001"/>	Description :	<input type="text" value=""/>
Department :	<input type="text" value="5180"/>	Active :	<input checked="" type="checkbox"/>
Appr Unit :	<input type="text" value="1112515"/>		
Unit :	<input type="text" value="IHSS"/>		
Object :	<input type="text" value="5001"/>		

[Top](#)

[Modified Budget Line Controls](#) [Appropriation Budget](#) [Linked Revenues](#)

Figure – OC Expense Budget (BQ29LV2)

Detailed Transaction Listing

	Doc Code	Doc Dept	Doc ID	Document Identifier	Accrued Expenses
✓	GAX	5180	A TEST 5	GAX 5180 A TEST 5	\$1,999.99
	GAX	5180	A TEST 4	GAX 5180 A TEST 4	\$73.73
	GAX	5180	A-TEST 3	GAX 5180 A-TEST 3	\$250.50
	GAX	5180	A-TEST 2	GAX 5180 A-TEST 2	\$118.18
	GAX	5180	A-TEST1	GAX 5180 A-TEST1	\$75.50
	AD	5180	MAN000000000000000015	AD 5180 MAN000000000000000015	(\$302.26)
	GAX	5180	TEST 11	GAX 5180 TEST 11	\$250.75
	GAX	5180	TEST10	GAX 5180 TEST10	\$51.51
	AD	5180	MAN000000000000000014	AD 5180 MAN000000000000000014	(\$442.41)
	GAX	5180	TEST 7	GAX 5180 TEST 7	\$166.66
	GAX	5180	TEST 6	GAX 5180 TEST 6	\$275.75
	AD	5180	MAN000000000000000013	AD 5180 MAN000000000000000013	(\$388.88)
	GAX	5180	TEST5	GAX 5180 TEST5	\$388.88

First Prev Next Last [Download](#)

[Search](#)

Doc Code :

Doc Dept :

Accrued Expenses

Doc ID :

Figure – Detail Transaction Listing (Expense Budget)

Disbursement Request

Menu Quick Search

Document	Vendor Code	Legal Name	Accounting Line Amount
✓ GAX.5180 A-TEST 4	00COUNTY01	TREASURER ALAMEDA COUNTY	\$73.73
GAX.5180 A-TEST 5	00COUNTY02	TREASURER ALPINE COUNTY	\$1,999.99
GAX.5180 A-TEST 2	00COUNTY05	TREASURER CALAVERAS COUNTY	\$118.18
GAX.5180 A-TEST 3	00COUNTY06	TREASURER COLUSA COUNTY	\$250.50
GAX.5180 A-TEST 1	00COUNTY04	TREASURER BUTTE COUNTY	\$75.50

First Prev Next Last

Save Undo Search

▼Vendor

Doc Code: GAX	Legal Name: TREASURER ALAMEDA	Payee Code:
Doc Dept: 5180	Vendor Alias Name:	Payee Legal Name:
Doc ID: A-TEST 4	Address Code: AD003	Payee Address ID:
Vendor Line Number: 1	Address 1: 1221 OAK ST	Payee Alias Name:
Date of Record: 12/16/2008	Address 2:	Address 1:
Vendor Code: 00COUNTY01	City: OAKLAND	Address 2:
	State: California	City:
	Zip Code: 94612	State:
	Taxpayer ID Number:	Zip Code:
	Taxpayer ID Type:	
	Miscellaneous Vendor Flag: false	

▼Disbursement Control

Scheduled Payment Date: 12/16/2008	Disbursement Format: REG
Single Payment: <input type="checkbox"/>	Disbursement Category: REG
Handling Code:	Disbursement Type: Check
Online Disb Request: <input type="checkbox"/>	Consolidation Object 1:
Disbursement Management: <input type="checkbox"/>	Consolidation Object 2:
User Hold: <input type="checkbox"/>	Consolidation Object 3:
Hold Payment Reason:	Consolidation Object 4:
System Hold Flag: false	Consolidation Object 5:
System Hold Reason: Blank	Consolidation Object 6:
Priority: 99	Sort Object 1:
	Sort Object 2:

Figure – Disbursement Request (DISRQ) - Part 1

▼Commodity

Commodity Code :		Invoice Number :	TEST 4
Commodity Line Number :	0	Invoice Line :	1
Commodity Description :		Invoice Date :	12/08/2008
Unit :		Tracking Date :	
Unit Price :		Service From Date :	12/16/2008
Contract Amount :	0.00	Service To Date :	12/16/2008
Calculated Line Total :	0.00	PCard Record ID :	
Tax Amount :	0.00	PCard Administrator ID :	
		Procurement Card Payment :	No

▼Accounting

Accounting Line Number :	1	Fund :	0001	Location :	
Fiscal Year :	2009	Sub Fund :		Sub Location :	
Accounting Period :	6	Object :	5001	Reporting :	221R
Event Type :	AP01	Sub Object :		Sub Reporting :	
Bank Account Code :	1000	Revenue :		Dept Object :	
Posting Line Number :		Sub Revenue :		Dept Revenue :	
Accounting Line Amount :	\$73.73	BSA :		Major Program :	
Retainage Line Amount :	0.00	Sub BSA :		Program :	
Special Instructions Code :		Department :	5180	Program Period :	
		Unit :	IHSS	Funding Profile :	
		Sub Unit :	0122	Task :	
		Appr Unit :	1112515	Sub Task :	
		Activity :		Task Order :	
		Sub Activity :			
		Function :		Check Description :	
		Run Time Date :	12/18/2008		
		Sub Function :			

▼Reference Information

Reference Document Code :		Reference Document Vendor Line :	
Reference Document Department :		Reference Commodity Line :	
Reference Document ID :		Reference Accounting Line :	

▼Discount Information

Discount Always 1 :	false	Discount Days 1 :		Discount Percent 1 :	
Discount Always 2 :	false	Discount Days 2 :		Discount Percent 2 :	
Discount Always 3 :	false	Discount Days 3 :		Discount Percent 3 :	
Discount Always 4 :	false	Discount Days 4 :		Discount Percent 4 :	
		Replacement :	<input type="checkbox"/>		
		Advance Type :			
		Dyna Bank :			

[Top](#)

Figure – Disbursement Request (DISRQ) - Part 2

DSD 19/Payroll – Warrant Management/Business Process /Navigation Elements

This section is not applicable for Warrant Management.

DSD 19/Payroll – Warrant Management/Business Process /Error Messages

This section is not applicable for Warrant Management.

DSD 19/Payroll – Warrant Management/Business Process /Business Rules

This section is not applicable for Warrant Management.

DSD 19/Payroll – Warrant Management/Business Process /Tasks/Notifications

This section is not applicable for Warrant Management.

DSD 19/Payroll – Warrant Management/Business Process /Internal Interfaces

The following are the internal interfaces for Warrant Management:

- Warrant Information Table Create
- Warrant Information Table Update
- Send Void and Reissue Request
- County Contractor and Homemaker Journal Voucher

DSD 19/Payroll – Warrant Management/Business Process /Internal Interfaces/Warrant Information Table Create

CI	Document Name
CI-69631 - DSD INTF Warrant Information Table Create IMPLEMENTED	DSD_INTF_Warrant_Information_Table_Create.doc

This is the interface between CMIPS and CM. Interface will extract warrant-specific information from the Advantage HRM tables for recipient and provider warrants issued on specified cycle dates and for specified GTN run numbers. This information will be stored in the file and will contain the data fields required by CM to store in CM-specific Warrant table(s). The following records will be excluded from processing into CM:

- Regular cycle \$0 warrants that were not reduced to \$0 by deductions
- Supplemental cycle warrants with \$0 gross and \$0 net where there are no deductions or all deductions are \$0
- The warrants where only fringe deductions exist and warrant is for \$0 gross and \$0 net
- Prenote \$0 checks that were not reduced to \$0 by deductions

Characteristics of Interface

- Type (Interface Method) – Batch
- Format – Text file (ASCII)
- Priority – Critical
- Frequency – Daily
- Timing – Sequencing – Execute after Daily Payment File to SCO process is completed.
- Security & Privacy Classification: N/A – Interface will be internal to CMIPS
- Source(s) (System) – Advantage Systems
- Target(s) (System) – CM
- Data Transformation(s) – none
- Media – Electronic
- Interface Method – Web Services over HTTPS

Data Elements of Interface

Warrant Information Create File

Source: Payroll		Target: Warrant Information Create File			
Table Name/ File Name/ WSDL Name	Field Name	Field Name	Data Type (Format- length)	Description(Data Transformation rules)	R/O(Required or optional)
TAXING_ENTITY	CNTAC_LAST_NM	recipientName.lastName	String(20)		R
TAXING_ENTITY	CNTAC_MIDDLE_NM	recipientName.middleName	String(15)		O
TAXING_ENTITY	CNTAC_FIRST_NM	recipientName.firstName	String(15)		R
TAXING_ENTITY	CNTAC_NM_SUFFIX	recipientName.suffix	String(5)		O
PAY_CHECK_EXT	EMPL_LAST_NM	payeeName.lastName	String(20)		R
PAY_CHECK_EXT	EMPL_MIDDLE_NM	payeeName.middleName	String(15)		O
PAY_CHECK_EXT	EMPL_FIRST_NM	payeeName.firstName	String(15)		R
PAY_CHECK_EXT	EMPL_SUFFIX_CD	payeeName.suffix	String(5)		O
PAY_CHECK_EXT	STR_1_NM + " STR_2_NM	street	String(150)		R
PAY_CHECK_EXT	CITY_NM	city	String(28)		R
PAY_CHECK_EXT	ST_CD	state	String(2)		R
PAY_CHECK_EXT	ZIP (first 5 characters)	zip	String(5)		R
PAY_CHECK_EXT	ZIP (last 4 characters, if not empty)	zip4	String(4)		O
PAY_CHECK	CHK_DT	processedDate	Date		R
PAY_CHECK	PPRD_STRT_DT	paymentFromDate	Date	See the logic in CMWarrantRecordProcessorImpl. populateWarrantInfo()	R

PAY_CHECK	PPRD_END_DT	paymentToDate	Date	See the logic in CMWarrantRecordProcessorImpl. populateWarrantInfo()	R
PAY_DETL	INPUT_AM	hoursOverAuth	Numeric	Represents hours cutback due to exceeding monthly authorization. See the logic in CMWarrantRecordProcessorImpl. populateWarrantInfo() 'CBHP'	O
PAY_DETL	INPUT_AM	hoursPaidMin (provide in minutes only)	Numeric	Represents time recorded providing services. See the logic in CMWarrantRecordProcessorImpl. populateWarrantInfo() Where substr(evtnt_tpy_cd, 1, 1) = 'G' or 'A' or 'N' or (substr(evtnt_tpy_cd,1,1) = 'C' and payAdjustType = 'CVDR')	R
PAY_DETL	INPUT_AM	hoursPaidMinTrvl	Numeric	Represents the Travel hours that were paid. Where substr(evtnt_tpy_cd, 1, 1) = 'T' or 'V'	O
PAY_DETL	INPUT_AM	hoursOverMaxTrvl	Numeric	Represents travel hours reported but not paid due to exceeding 14 hours in a pay period, evtnt_tpy_cd = 'HCBT'	O
PAY_DETL	INPUT_AM	hoursPaidMinSkiv	Numeric	Represents the Sick Leave hours that were paid. Where substr(evtnt_tpy_cd, 1, 1) = 'S' or 'K'	O
PAY_DETL	INPUT_AM	hoursOverMaxSkiv	Numeric	Represents sick leave hours claimed but not paid due to exceeding available hours, evtnt_tpy_cd = 'CBHS'	O
PAY_DETL	INPUT_AM	hoursPaidMinTrn	Numeric	Represents Training hours that were paid (in minutes) Where substr(evtnt_tpy_cd,1,1) = 'N' and input_am <> cntrct_py_am	
PAY_DETL	INPUT_AM	hoursOverMaxTrn	Numeric	Represents Training hours that were claimed but not paid , evtnt_tpy_cd = 'CBHN'	
PAY_DETL	PAY_RT_AM	payRate	Numeric(11,2)	See the logic in CMWarrantRecordProcessorImpl. populateWarrantInfo() Where substr(evtnt_tpy_cd, 1, 1) = 'N' if EVENT_TYPE_CD = G,A,T,V,S,N,K,C then the highest pay rate from the HoursWageRate query should be for Payrate field.	R
PAY_DETL	PAY_RT_AM	payRateOT	Numeric(11,2)	See the logic in CMWarrantRecordProcessorImpl. populateWarrantInfo() if EVENT_TYPE_CD = F or L then the highest pay rate from the HoursWageRate query should be for PayrateOT field	R
PAY_CHECK	PPRD_CHK_GROS_AM	PayStubItemtype.amount - "TOTAL GROSS" line item type code	Numeric(9,2)	PAY_CHECK.PPRD_CHK_GROS_AM	R
PAY_DETL	CNTRCT_PY_AM	PayStubItemtype.amount - "PAYREG" line item type code	Numeric(9,2)	Sum of payment lines associated with service hours provided. Where substr(evtnt_tpy_cd, 1, 1) = 'G' or 'N'	O
PAY_DETL	CNTRCT_PY_AM	PayStubItemtype.amount - "PAYADJ" line item type code	Numeric(9,2)	Sum of payment lines associated with payment corrections or special transactions Where substr(evtnt_tpy_cd, 1, 1) = 'A' or (substr(evtnt_tpy_cd,1,1) = 'C' and payAdjustType = 'CVDR')	O
PAY_DETL	CNTRCT_PY_AM	PayStubItemtype.amount - "PAYTRV" line item type code	Numeric(9,2)	Sum of payment lines associated with travel Where substr(evtnt_tpy_cd, 1, 1) = 'T' or 'V'	O
PAY_DETL	CNTRCT_PY_AM	PayStubItemtype.amount - "PAYOVT" line item type code	Numeric(9,2)	Sum of payment lines associated with overtime Where substr(evtnt_tpy_cd, 1, 1) = 'F' or 'L'	O

PAY_DETL	CNTRCT_PY_AM	PayStubItem.amount - "PAYSKLV" line item type code	Numeric(9,2)	Sum of payment lines associated with sick leave Where substr(evtntyp_cd, 1, 1) = 'S' or 'K'	O
DED_DETL	DED_AM	PayStubItem.amount - "FEDEIC" line item type code	Numeric(9,2)	Sum of ded_detl.ded_am values for a check with ded_detl.dedtyp_cd corresponding to DED_DETL_EXT.CHK_prnt_seq_no = 4	O
DED_DETL	DED_AM	PayStubItem.amount - "ADDFEDTX" line item type code	Numeric(9,2)	Sum of ded_detl.ded_am values for a check with ded_detl.dedtyp_cd corresponding to DED_DETL_EXT.CHK_prnt_seq_no = 5	O
DED_DETL	DED_AM	PayStubItem.amount - "STTX" line item type code	Numeric(9,2)	Sum of ded_detl.ded_am values for a check with ded_detl.dedtyp_cd corresponding to DED_DETL_EXT.CHK_prnt_seq_no = 6	O
DED_DETL	DED_AM	PayStubItem.amount - "ADDSTTX" line item type code	Numeric(9,2)	Sum of ded_detl.ded_am values for a check with ded_detl.dedtyp_cd corresponding to DED_DETL_EXT.CHK_prnt_seq_no = 7	O
DED_DETL	DED_AM	PayStubItem.amount - "FICA" line item type code	Numeric(9,2)	Sum of ded_detl.ded_am values for a check with ded_detl.dedtyp_cd corresponding to DED_DETL_EXT.CHK_prnt_seq_no = 8	O
DED_DETL	DED_AM	PayStubItem.amount - "MEDICARE" line item type code	Numeric(9,2)	Sum of ded_detl.ded_am values for a check with ded_detl.dedtyp_cd corresponding to DED_DETL_EXT.CHK_prnt_seq_no = 9	O
DED_DETL	DED_AM	PayStubItem.amount - "SDIDIEC" line item type code	Numeric(9,2)	Sum of ded_detl.ded_am values for a check with ded_detl.dedtyp_cd corresponding to DED_DETL_EXT.CHK_prnt_seq_no = 10	O
DED_DETL	DED_AM	PayStubItem.amount - "SOC" line item type code	Numeric(9,2)	Sum of ded_detl.ded_am values for a check with ded_detl.dedtyp_cd corresponding to DED_DETL_EXT.CHK_prnt_seq_no = 11	O
DED_DETL	DED_AM	PayStubItem.amount - "LIEN" line item type code	Numeric(9,2)	Sum of ded_detl.ded_am values for a check with ded_detl.dedtyp_cd corresponding to DED_DETL_EXT.CHK_prnt_seq_no = 13	O
DED_DETL	DED_AM	PayStubItem.amount - "HEALTH" line item type code	Numeric(9,2)	Sum of ded_detl.ded_am values for a check with ded_detl.dedtyp_cd corresponding to DED_DETL_EXT.CHK_prnt_seq_no = 14	O
DED_DETL	DED_AM	PayStubItem.amount - "DUES" line item type code	Numeric(9,2)	Sum of ded_detl.ded_am values for a check with ded_detl.dedtyp_cd corresponding to DED_DETL_EXT.CHK_prnt_seq_no = 15	O
DED_DETL	DED_AM	PayStubItem.amount - "HLTHTRST" line item type code	Numeric(9,2)	Sum of ded_detl.ded_am values for a check with ded_detl.dedtyp_cd corresponding to DED_DETL_EXT.CHK_prnt_seq_no = 16	O
DED_DETL	DED_AM	PayStubItem.amount - "COPEPL" line item type code	Numeric(9,2)	Sum of ded_detl.ded_am values for a check with ded_detl.dedtyp_cd corresponding to DED_DETL_EXT.CHK_prnt_seq_no = 17	O
DED_DETL	DED_AM	PayStubItem.amount - "INIT" line item type code	Numeric(9,2)	Sum of ded_detl.ded_am values for a check with ded_detl.dedtyp_cd corresponding to DED_DETL_EXT.CHK_prnt_seq_no = 18	O
DED_DETL	DED_AM	PayStubItem.amount - "OTINS" line item type code	Numeric(9,2)	Sum of ded_detl.ded_am values for a check with ded_detl.dedtyp_cd corresponding to DED_DETL_EXT.CHK_prnt_seq_no = 19	O
DED_DETL	DED_AM	PayStubItem.amount - "APDED" line item type code	Numeric(9,2)	Sum of ded_detl.ded_am values for a check with ded_detl.dedtyp_cd corresponding to DED_DETL_EXT.CHK_prnt_seq_no = 20	O
DED_DETL	DED_AM	PayStubItem.amount - "RECOVERY" line item type code	Numeric(9,2)	Sum of ded_detl.ded_am values for a check with ded_detl.dedtyp_cd corresponding to DED_DETL_EXT.CHK_prnt_seq_no = 12	O
DED_DETL	DED_AM	PayStubItem.amount - "CART" line item type code	Numeric(9,2)	Sum of ded_detl.ded_am values for a check with ded_detl.dedtyp_cd corresponding to DED_DETL_EXT.CHK_prnt_seq_no = 21	O
DED_DETL	DED_AM	PayStubItem.amount - the "NETPAY" line item type code	Numeric(9,2)	DED_DETL.DED_AM where DEDTYP_CD = '99999'	R
PAY_CHK_PAY_SUMM	PPRD_CHK_GROS_AM	PayStubItem.ytdAmt - "TOTAL GROSS" line item type code	Numeric(9,2)	Sum of all wages earned within the current calendar year. See the logic in CMWarrantRecordProcessorImpl.populateWarrantInfo()	R

PAY_CHK_PAY_S UMM	QTR_1_CASH_PAY_AM QTR_2_CASH_PAY_AM QTR_3_CASH_PAY_AM QTR_4_CASH_PAY_AM	PayStubItemtype.ytdAmt - "PAYREG" line item type code	Numeric(9,2)	sum of wages earned for providing services to the Recipient. See the logic in CMWarrantRecordProcessorImpl. populateWarrantInfo() sum (pay_chk_pay_summ. qtr_1_cash_pay_am, qtr_2_cash_pay_am, qtr_3_cash_pay_am, qtr_4_cash_pay_am) where frng_py_typ_fl = 'N' and chk_prnt_seq_no = ('1' , '6')	O
PAY_CHK_PAY_S UMM	QTR_1_CASH_PAY_AM QTR_2_CASH_PAY_AM QTR_3_CASH_PAY_AM QTR_4_CASH_PAY_AM	PayStubItemtype.ytdAmt - "PAYADJ" line item type code	Numeric(9,2)	Sum of wages earned as adjustments when providing services to the Recipient See the logic in CMWarrantRecordProcessorImpl. populateWarrantInfo() sum (pay_chk_pay_summ. qtr_1_cash_pay_am, qtr_2_cash_pay_am, qtr_3_cash_pay_am, qtr_4_cash_pay_am) where frng_py_typ_fl = 'N' and chk_prnt_seq_no = '2'	O
PAY_CHK_PAY_S UMM	QTR_1_CASH_PAY_AM QTR_2_CASH_PAY_AM QTR_3_CASH_PAY_AM QTR_4_CASH_PAY_AM	PayStubItemtype.ytdAmt - "PAYRTRV" line item type code	Numeric(9,2)	sum of wages earned traveling from Recipient A to Recipient B within the current calendar year. sum (pay_chk_pay_summ. qtr_1_cash_pay_am, qtr_2_cash_pay_am, qtr_3_cash_pay_am, qtr_4_cash_pay_am) where frng_py_typ_fl = 'N' and chk_prnt_seq_no = '4'	O
PAY_CHK_PAY_S UMM	QTR_1_CASH_PAY_AM QTR_2_CASH_PAY_AM QTR_3_CASH_PAY_AM QTR_4_CASH_PAY_AM	PayStubItemtype.ytdAmt - "PAYOVT" line item type code	Numeric(9,2)	sum (pay_chk_pay_summ. qtr_1_cash_pay_am, qtr_2_cash_pay_am, qtr_3_cash_pay_am, qtr_4_cash_pay_am) where frng_py_typ_fl = 'N' and chk_prnt_seq_no = '3'	O
PAY_CHK_PAY_S UMM	QTR_1_CASH_PAY_AM QTR_2_CASH_PAY_AM QTR_3_CASH_PAY_AM QTR_4_CASH_PAY_AM	PayStubItemtype.ytdAmt - "PAYSKLV" line item type code	Numeric(9,2)	sum (pay_chk_pay_summ. qtr_1_cash_pay_am, qtr_2_cash_pay_am, qtr_3_cash_pay_am, qtr_4_cash_pay_am) where frng_py_typ_fl = 'N' and chk_prnt_seq_no = '5'	O
PAY_CHK_DED_ SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	PayStubItemtype.ytdAmt - "FEDEIC" line item type code	Numeric(9,2)	Sum of pay_chk_ded_summ. ded_1st_qtr_am, ded_2nd_qtr_am, ded_3rd_qtr_am, ded_4th_qtr_am values for a current year where chk_prnt_seq_no = 4	O
PAY_CHK_DED_ SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	PayStubItemtype.ytdAmt - "ADDFTDX" line item type code	Numeric(9,2)	Sum of pay_chk_ded_summ. ded_1st_qtr_am, ded_2nd_qtr_am, ded_3rd_qtr_am, ded_4th_qtr_am values for a current year where chk_prnt_seq_no = 5	O
PAY_CHK_DED_ SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	PayStubItemtype.ytdAmt - "STTX" line item type code	Numeric(9,2)	Sum of pay_chk_ded_summ. ded_1st_qtr_am, ded_2nd_qtr_am, ded_3rd_qtr_am, ded_4th_qtr_am values for a current year where chk_prnt_seq_no = 6	O
PAY_CHK_DED_ SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	PayStubItemtype.ytdAmt - "ADDSTTX" line item type code	Numeric(9,2)	Sum of pay_chk_ded_summ. ded_1st_qtr_am, ded_2nd_qtr_am, ded_3rd_qtr_am, ded_4th_qtr_am values for a current year where chk_prnt_seq_no = 7	O
PAY_CHK_DED_ SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	PayStubItemtype.ytdAmt - "FICA" line item type code	Numeric(9,2)	Sum of pay_chk_ded_summ. ded_1st_qtr_am, ded_2nd_qtr_am, ded_3rd_qtr_am, ded_4th_qtr_am values for a current year where chk_prnt_seq_no = 8	O
PAY_CHK_DED_ SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	PayStubItemtype.ytdAmt - "MEDICARE" line item type code	Numeric(9,2)	Sum of pay_chk_ded_summ. ded_1st_qtr_am, ded_2nd_qtr_am, ded_3rd_qtr_am, ded_4th_qtr_am values for a current year where chk_prnt_seq_no = 9	O
PAY_CHK_DED_ SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	PayStubItemtype.ytdAmt - "SDIDIEC" line item type code	Numeric(9,2)	Sum of pay_chk_ded_summ. ded_1st_qtr_am, ded_2nd_qtr_am, ded_3rd_qtr_am, ded_4th_qtr_am values for a current year where chk_prnt_seq_no = 10	O
PAY_CHK_DED_ SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	PayStubItemtype.ytdAmt - "SOC" line item type code	Numeric(9,2)	Sum of pay_chk_ded_summ. ded_1st_qtr_am, ded_2nd_qtr_am, ded_3rd_qtr_am, ded_4th_qtr_am values for a current year where chk_prnt_seq_no = 11	O

PAY_CHK_DED_SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	PayStubItemtype.ytdAmt - "LIEN" line item type code	Numeric(9,2)	Sum of pay_chk_ded_summ. ded_1st_qtr_am, ded_2nd_qtr_am, ded_3rd_qtr_am, ded_4th_qtr_am values for a current year where chk_prnt_seq_no = 13	O
PAY_CHK_DED_SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	PayStubItemtype.ytdAmt - "HEALTH" line item type code	Numeric(9,2)	Sum of pay_chk_ded_summ. ded_1st_qtr_am, ded_2nd_qtr_am, ded_3rd_qtr_am, ded_4th_qtr_am values for a current year where chk_prnt_seq_no = 14	O
PAY_CHK_DED_SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	PayStubItemtype.ytdAmt - "DUES" line item type code	Numeric(9,2)	Sum of pay_chk_ded_summ. ded_1st_qtr_am, ded_2nd_qtr_am, ded_3rd_qtr_am, ded_4th_qtr_am values for a current year where chk_prnt_seq_no = 15	O
PAY_CHK_DED_SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	PayStubItemtype.ytdAmt - "HLTHTRST" line item type code	Numeric(9,2)	Sum of pay_chk_ded_summ. ded_1st_qtr_am, ded_2nd_qtr_am, ded_3rd_qtr_am, ded_4th_qtr_am values for a current year where chk_prnt_seq_no = 16	O
PAY_CHK_DED_SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	PayStubItemtype.ytdAmt - "COPEPPL" line item type code	Numeric(9,2)	Sum of pay_chk_ded_summ. ded_1st_qtr_am, ded_2nd_qtr_am, ded_3rd_qtr_am, ded_4th_qtr_am values for a current year where chk_prnt_seq_no = 17	O
PAY_CHK_DED_SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	PayStubItemtype.ytdAmt - "INIT" line item type code	Numeric(9,2)	Sum of pay_chk_ded_summ. ded_1st_qtr_am, ded_2nd_qtr_am, ded_3rd_qtr_am, ded_4th_qtr_am values for a current year where chk_prnt_seq_no = 18	O
PAY_CHK_DED_SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	PayStubItemtype.ytdAmt - "OTINS" line item type code	Numeric(9,2)	Sum of pay_chk_ded_summ. ded_1st_qtr_am, ded_2nd_qtr_am, ded_3rd_qtr_am, ded_4th_qtr_am values for a current year where chk_prnt_seq_no = 19	O
PAY_CHK_DED_SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	PayStubItemtype.ytdAmt - "APDED" line item type code	Numeric(9,2)	Sum of pay_chk_ded_summ. ded_1st_qtr_am, ded_2nd_qtr_am, ded_3rd_qtr_am, ded_4th_qtr_am values for a current year where chk_prnt_seq_no = 20	O
PAY_CHK_DED_SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	PayStubItemtype.ytdAmt - "RECOVERY" line item type code	Numeric(9,2)	Sum of pay_chk_ded_summ. ded_1st_qtr_am, ded_2nd_qtr_am, ded_3rd_qtr_am, ded_4th_qtr_am values for a current year where chk_prnt_seq_no = 12	O
PAY_CHK_DED_SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	PayStubItemtype.ytdAmt - "CART" line item type code	Numeric(9,2)	Sum of pay_chk_ded_summ. ded_1st_qtr_am, ded_2nd_qtr_am, ded_3rd_qtr_am, ded_4th_qtr_am values for a current year where chk_prnt_seq_no = 21	O
PAY_CHECK	PAY_CHK_AM	PayStubItemtype.ytdAmt - "NETPAY" line item type code	Numeric(9,2)	Sum of the PAY_CHECK.PAY_CHK_AM for all pay periods in the current calendar year.	R
PAY_CHECK	CHK_NO	advWarrantNumber	String(13)		R
PAY_CHECK_EXT	EMPLOYEE_ID	employeeID	String(10)		R
TAXING_ENTITY	ST_TAX_ID_2	sein	String(8)	See the logic in CMWarrantRecordProcessorImpl. populateWarrantInfo()	R
PAY_DETL	DOC_ID	payType	String(4)	See the logic in CMWarrantRecordProcessorImpl. populateWarrantInfo()	R
PAY_CHECK_EXT	CIVIL_SVC_STA_CD	taxRelationshipCode	String(2)	See the logic in CMWarrantRecordProcessorImpl. populateWarrantInfo()	R
PAY_CHECK_EXT	FED_TAX_EXMP_CT	w4Allowances	Numeric	Null value	R
PAY_CHECK_EXT	FED_TAX_MAR_STA_CD	w4StatusCode	String(1)	Null value	O
EMPL_TAX_PARM	TAX_CLS_CD	eicStatusCode	String(1)	Null value	O
PAY_CHECK_EXT	ST_LC_TAX_EXMP_CT	de4Allowances	Numeric	Null value	R
PAY_CHECK_EXT	ST_LC_TX_MAR_STA_CD	de4StatusCode	String(1)	Null value	O
PAY_CHECK_EXT	HOME_UNIT_CD	countyCode	String(2)	home_unit_cd first two characters	R
PAY_CHECK_EXT	HOME_UNIT_CD	districtOfficeCode	String(2)	Home_unit_cd last two characters	R
PAY_CHECK_EXT	HOME_DEPT_CD	fundingSource	String(5)	HOME_DEPT_CD	R
MICR	EFT_BANK_ACCT_NM	isEFT	String(1)	If MICR. EFT_BANK_ACCT_NM is empty then "N" else "Y"	R
DED_DETL	DED_AM	EmployerPaidTaxItemType. amount - "FICA" line item type code	Numeric(9,2)	Sum of ded_detl records for a check with dedtyp_cd = "CFICR".	O

DED_DETL	DED_AM	EmployerPaidTaxItemType. amount - "MEDICARE"	Numeric(9,2)	Sum of ded_detl records for a check with dedtyp_cd = "CMEDR".	O
DED_DETL	DED_AM	EmployerPaidTaxItemType. amount - "FUTA"	Numeric(9,2)	Sum of ded_detl records for a check with dedtyp_cd = "CFU1R" and "CFU2R".	O
DED_DETL	DED_AM	EmployerPaidTaxItemType. amount - "SUI"	Numeric(9,2)	Sum of ded_detl records for a check with dedtyp_cd = "CSUIR".	O
DED_DETL	DED_AM	EmployerPaidTaxItemType. amount - "ETT"	Numeric(9,2)	Sum of ded_detl records for a check with dedtyp_cd = "CETTR".	O
PAY_CHK_DED_ SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	EmployerPaidTaxItemType. ytdAmt - "FICA" line item type code	Numeric(9,2)	Sum of pay_chk_ded_summ records with catg_cd = "CFICR" for a current calendar year.	O
PAY_CHK_DED_ SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	EmployerPaidTaxItemType. ytdAmt - "MEDICARE"	Numeric(9,2)	Sum of pay_chk_ded_summ records with catg_cd = "CMEDR" for a current calendar year.	O
PAY_CHK_DED_ SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	EmployerPaidTaxItemType. ytdAmt - "FUTA"	Numeric(9,2)	Sum of pay_chk_ded_summ records with catg_cd = "CFU1R" and "CFU2R" for a current calendar year.	O
PAY_CHK_DED_ SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	EmployerPaidTaxItemType. ytdAmt - "SUI"	Numeric(9,2)	Sum of pay_chk_ded_summ records with catg_cd = "CSUIR" for a current calendar year.	O
PAY_CHK_DED_ SUMM	DED_1ST_QTR_AM, DED_2ND_QTR_AM, DED_3RD_QTR_AM, DED_4TH_QTR_AM	EmployerPaidTaxItemType. ytdAmt - "ETT"	Numeric(9,2)	Sum of pay_chk_ded_summ records with catg_cd = "CETTR" for a current calendar year.	O
PAY_DETL	DOC_ID	timesheetNumber	String(16)	See the logic in CMWarrantRecordProcessorImpl. populateTimesheetPayRequestList ()	O
PAY_DETL. DOC_ID	DOC_ID	specialTransactionNumber	String(16)	See the logic in CMWarrantRecordProcessorImpl. populateTimesheetPayRequestList ()	O
PAY_DETL. DOC_ID	DOC_ID	sickLeaveClaimNumber	String(16)	See the logic in CMWarrantRecordProcessorImpl. populateTimesheetPayRequestList ()	O
PAY_DETL. DOC_ID	DOC_ID	TimeAdjustmentNumber	String(16)	See the logic in CMWarrantRecordProcessorImpl. populateTimesheetPayRequestList ()	O
PAY_DETL	CNTRCT_PY_AM	MultiplePayRequests. PayRequest.grossAmt	Numeric (9,2)	See the logic in CMWarrantRecordProcessorImpl. populateTimesheetPayRequestList ()	O
PAY_DETL	DOC_ID	MultiplePayRequests. PayRequest.PayType	String(4)	See the logic in CMWarrantRecordProcessorImpl. populateTimesheetPayRequestList ()	O
PAY_DETL	CNTRCT_PAY_AM	MultiplePayRequests. PayRequest.grossAmt			O
PAY_DETL	INPUT_AM	MultiplePayRequests. PayRequest. hoursPaidMin			O
PAY_DETL	INPUT_AM	MultiplePayRequests. PayRequest. hoursPaidOvertime			O
PAY_DETL	INPUT_AM	MultiplePayRequests. PayRequest.hoursPaidMinTrvl			O
PAY_DETL	INPUT_AM	MultiplePayRequests. PayRequest.hoursPaidMinSkvl			O
PAY_DETL	INPUT_AM	MultiplePayRequests. PayRequest.hoursPaidMinTrn		New element - include hours from Training in this element	O
PAY_DETL	CNTRCT_PAY_AM	MultiplePayRequests. PayRequest.payReg		Include gross wages from training in this element.	O
PAY_DETL	CNTRCT_PAY_AM	MultiplePayRequests. PayRequest.payAdj			O
PAY_DETL	CNTRCT_PAY_AM	MultiplePayRequests. PayRequest.payTrv			O

PAY_DETL	CNTRCT_PAY_AM	MultiplePayRequests. PayRequest.payOvt			O
PAY_DETL	CNTRCT_PAY_AM	MultiplePayRequests. PayRequest.paySkvl			O
PAY_DETL	DOC_ID	MultiplePayRequests. PayRequest. transactionNumber		See the logic in CMWarrantRecordProcessorImpl. populateWarrantInfo() If no pay_detl record exists, set pay type to 'TXRF'	O
PAY_DETL	STEP_CD	backUpProviderInd	String(1)	Set to "Y" when STEP_CD = "02", else to "N"	R
PAY_DETL	DOC_ID	cpClaims.cpClaimNumber		See the logic in CMWarrantRecordProcessorImpl. populateTimesheetPayRequestList ()	O
PAY_DETL	EVNT_TYP_CD	hoursOverMaxCp		Populate where evnt_typ_cd = "CBHN"	O

Error Codes

Error No.	Error Code	Description
1	WARRCREATE1001	Current Taxing Entity record was not found
2	WARRCREATE1007	Gross YTD and Net YTD warrant totals were not found
3	WARRCREATE1009	Invalid length of Civil Status Code
4	WARRCREATE1010	Invalid Civil Service Status Code
5	WARRCREATE1011	Invalid length of County Code

Triggering Events

This section is not applicable for Warrant Management.

Errors

The error processing will be following the same process as other asynchronous errors. SMU errors will be captured and relayed back to CM. Error log files will be written for the Operations staff.

Edits and Audits

The standard COTS package edits will be performed on the Advantage application. Any failures from a synchronous call will send back the failure on the response to CM. Any failures from an asynchronous call will be written to an error log, and when the error is from a document create or update the document will remain in the Advantage application with a rejected status.

DSD 19/Payroll – Warrant Management/Business Process /Internal Interfaces/Warrant Information Table Update

CI	Document Name
CI-69630 - DSD INTF Warrant Information Table Update IMPLEMENTED	DSD_INTF_Warrant_Information_Table_Update.doc

This interface will be an internal interface between Payroll and CM that will be run on a nightly basis. In order for CM to be in sync with Payroll, Payroll will need to create a file that CM can process with the updates that occurred during the Payroll nightly batch cycle. This file will be used to provide CM with SCO warrant numbers and dates, warrant status changes and Advance Pay payment/timesheet reconciliation status information.

Characteristics of Interface

- Type (Interface Method) – Batch
- Format – Text file (ASCII)
- Priority – Critical
- Frequency – Nightly
- Timing – Sequencing – Executed after
- Warrant Assignment File is processed
- Warrant Paid File is processed
- Undeliverable/Cancelled File is processed
- Advance Pay Reconciliation Batch job is processed
- Security & Privacy Classification: N/A – Interface will be internal to CMIPS
- Source(s) (System) – Advantage Systems
- Target(s) (System) – CM
- Data Transformation(s) – none
- Media – Electronic
- Interface Method – Web Services over HTTPS

Data Elements of Interface

Table – SCO Return File for Cross Reference Warrants

Source : Payroll		Target: Case Management			
Table Name/ File Name/WSDL Name	Field Name	Field Name	Data Type (Format-length)	Description (Data Transformation rules)	R/O (Required or optional)
PAY_CHECK	CHK_NO	AdvWarrantNumber	As defined in XSD		R
PAY_CHECK	SCO_WARRANT_NUMBER	scoWarrantNumber	As defined in XSD		R
PAY_CHECK	SCO_WARRANT_DATE	Issuedt	As defined in XSD		R

Table – STO Return File for Cleared Warrants

Source : Payroll		Target: Case Management			
Table Name/ File Name/WSDL Name	Field Name	Field Name	Data Type (Format-length)	Description (Data Transformation rules)	R/O (Required or optional)
PAY_CHECK	CHK_NO	AdvWarrantNumber	As defined in XSD		R
PAY_CHECK	BANK_CHK_DISP_ID	Warrant Status Code	As defined in XSD		R
PAY_CHECK	CHK_DISP_DT	Warrant Status Date	As defined in XSD		R

Table – SCO Return File for Returned/Cancelled Warrants

Source : Payroll	Target: Case Management
------------------	-------------------------

Table Name/ File Name/WSDL Name	Field Name	Field Name	Data Type (Format-length)	Description (Data Transformation rules)	R/O (Required or optional)
PAY_CHECK	CHK_NO	AdvWarrantNumber	As defined in XSD		R
PAY_CHECK	BANK_CHK_DISP_ID	Warrant Status Code	As defined in XSD		R
PAY_CHECK	CHK_DISP_DT	Warrant Status Date	As defined in XSD		R

Table – Advance Pay Recipient Payments/Timesheet Reconciliation

Source : Payroll		Target: Case Management			
Table Name/ File Name/WSDL Name	Field Name	Field Name	Data Type (Format-length)	Description (Data Transformation rules)	R/O (Required or optional)
PAY_CHECK	CHK_NO	AdvWarrantNumber	As defined in XSD		R
ADVANCE_PY_RECON	Reconciliation Status	reconstatuscode	As defined in XSD		R
ADVANCE_PY_RECON	Reconciliation Date	reconstatusdt	As defined in XSD		R

Triggering Events

This section is not applicable for Warrant Management.

Errors

The error processing will be following the same process as other asynchronous errors. SMU errors will be captured and relayed back to CM. Error log files will be written for the Operations staff.

Edits and Audits

The standard COTS package edits will be performed on the Advantage application. Any failures from a synchronous call will send back the failure on the response to CM. Any failures from an asynchronous call will be written to an error log, and when the error is from a document create or update the document will remain in the Advantage application with a rejected status.

Table – Error Codes

Error No.	Error Code	Description
1	UPDWARR1001	No employees with recon status found

DSD 19/Payroll – Warrant Management/Business Process /Internal Interfaces/Send Void and Reissue Request

CI	Document Name
CI-69628 - DSD INTF Send Void and Reissue Request IMPLEMENTED	DSD_INTF_Send_Void_and_Reissue_Request.doc

The Payment Void/Reissue/Replacement Activity page in CM allows the user to request "Void and Reissue" to void a warrant which was damaged and to reissue another warrant to the payee. This section provides the data elements that CM sends to Payroll for Void and Reissue processing in Payroll.

Characteristics of Interface

- Type – Batch
- Format – XML data structures
- Priority – Critical
- Frequency – One-time at the end of the day, but before Payroll Payment processing job runs
- Timing/Sequencing – Void/Reissue request is generated from CM screen
- Security & Privacy Classification – No encryption is required on the data elements specified in this document –assumption as this is an internal interface
- Source(s) (System) – CM
- Target(s) (System) – Payroll
- Data Transformation(s) – None
- Media – Electronic
- Interface Method – Batch process

Data Elements of Interface

Table – Send Void and Reissue Request

Source:		Target: CHCK Document			
Batch Process	Field Name	Field Name	Data Type (Format-length)	Description (Data Transformation rules)	R/O (Required or optional)
Warrantreissuereq	advWarrantNo	CHK_NO	String (20)	CM will send the Advantage Warrant No.	R
Warrantreissuereq	employeeId	EMPLOYEE_ID	String (1)	Employee ID	R
		APPOINTMENT_ID	String (1)	Look up into PAY_CHECK table based on CHK_NO and employee_id	R
		BANK_ACCT_CD	String (4)	Defaulted to '1000'	R
		CHK_STA	String (1)	Set to 'S' for Void	R
		REPL_CHK	String (1)	Set to 'Y' for Replace	R

Table – Error Codes

Error No.	Error Code	Description
1	REISSCHCK1001	Transaction Failed since Check is not in Paid Status
2	REISSCHCK1002	Transaction Failed – Check Not Found in Advantage
3	REISSCHCK1003	CHCK Document submission failed

Triggering Events

This section is not applicable for Warrant Management.



Errors

The error processing will be following the same process as other asynchronous errors. SMU errors will be captured and relayed back to CM. Error log files will be written for the Operations staff.

Edits and Audits

The standard COTS package edits will be performed on the Advantage application. Any failures from a synchronous call will send back the failure on the response to CM. Any failures from an asynchronous call will be written to an error log, and when the error is from a document create or update the document will remain in the Advantage application with a rejected status.

DSD 19/Payroll – Warrant Management/Business Process /Internal Interfaces/County Contractor and Homemaker Journal Voucher

CI	Document Name
 CI-69629 - DSD INTF County Contractor and Homemaker Journal Voucher 	DSD_INTF_County_Contractor_and_Homemaker_Journal_Voucher.doc

The County Contractor and Homemaker are modes in CMIPS that account for services provided to Recipients by County employees and/or county-approved contractors. The County Contractors and Homemakers are paid directly by the counties for these services. Once the County Contractor invoice data is recorded into CMIPS, the financial details are posted in CMIPS Financial via journal voucher during the nightly cycle. The Homemaker hours are entered online into CMIPS and processed during the nightly cycle. Once the journal voucher is processed, the County Contractor and Homemaker financial data becomes available for reporting needs.

Characteristics of Interface

- Type (Interface Method) – Batch
- Format – XML
- Priority – Critical
- Frequency – Monthly
- Timing – Sequencing – No sequencing issues
- Security & Privacy Classification – None
- Source(s) (System) – CM
- Target(s) (System) – Financial
- Data Transformation(s) – Not applicable
- Media – Electronic
- Interface Method – XML data transfer

Data Elements of Interface

Table – County and Homemaker Journal Voucher

Source: Case Management		Target: FIN JVA Document			
Name of the Batch	Field Name	Field Name	Data Type(Format-length)	Description(Data Transformation rules)	R/O(Required or optional)
CountyContractHHorJVReq	Program	UNIT_CD	String (4)	Program code such as IHSS or WPCS	R
CountyContractHHorJVReq	County District	SUNIT_CD	String (4)	County District	R
CountyContractHHorJVReq	Amount	DR_AM	String (14)	Amount to be debited	R
CountyContractHHorJVReq	CCorHHflag	DOC_ID	String (20)	Process "HH" if HH, process "CC" if CC	R
CountyContractHHorJVReq	MonthEffective	ACTG_NM_DSCR	String (7)	Month effective; FIN does not use it for processing, but reporting will use it.	R

Table – Error Codes

Error No.	Error Code	Description
1	CNTYCNTRCTJVA1001	Invalid amount provided
2	CNTYCNTRCTJVA1002	Invalid County provided
3	CNTYCNTRCTJVA1003	Invalid District provided
4	CNTYCNTRCTJVA1004	Invalid Month Effective provided
5	CNTYCNTRCTJVA1005	Invalid CC or HM flag provided
6	CNTYCNTRCTJVA1006	Report Code/Split Percentage not found

7	CNTYCNTRCTJVA1007	Initial JVA failed on import
8	CNTYCNTRCTJVA1008	Adjusting JVA failed on import
9	CNTYCNTRCTJVA1009	Initial JVA failed on Validation/Submit
10	CNTYCNTRCTJVA1010	Adjusting JVA failed on Validation/Submit
11	CNTYCNTRCTJVA1011	Invalid Program Provided

Triggering Events

This section is not applicable for Warrant Management.

Errors

The error processing will be following the same process as other asynchronous errors. SMU errors will be captured and relayed back to CM. Error log files will be written for the Operations staff.

Edits and Audits

The standard COTS package edits will be performed on the Advantage application. Any failures from a synchronous call will send back the failure on the response to CM. Any failures from an asynchronous call will be written to an error log, and when the error is from a document create or update the document will remain in the Advantage application with a rejected status.

DSD 19/Payroll – Warrant Management/Business Process /External Interfaces

The following external interfaces will be covered in Warrant Management:

- EFT Pre-Note
- Daily Payment
- SCO Warrant Number Assignment
- Warrant Paid
- Undeliverable Warrants

DSD 19/Payroll – Warrant Management/Business Process /External Interfaces/EFT Pre-Note – PRWS109B

CI	Document Name
CI-116366 - DSD EINTF EFT Pre-Note PRWS109B IMPLEMENTED	DSD_EINTF_EFT_PreNote_PRWS109B.doc

This is the interface between CMIPS and the bank processing the EFT direct deposits. Interface will extract EFT data from CMIPS tables, format the data and send to SCO. SCO then forwards the information to the bank. The bank will review this information and contact SCO regarding incomplete or incorrect EFT direct deposit data. SCO sends this information back to CMIPS. CMIPS will wait 15 days from the date when direct deposit information was sent to the bank for validation. After this time, it will start issuing direct deposit payments to the Provider or Recipient unless information about incomplete or incorrect EFT information was received from the bank.

Internal Layout/System Mapping

Table – HRM EFT Pre-Note

Source: Payroll		Target: File to BPM			
Table Name/ File Name/ WSDL Name	Field Name	Field Name	Data Type (Format-length)	Description(Data Transformation rules)	R/O(Required or optional)
		RA_AU_RECIP_LAST	N/A	Not populated	N/A
		RA_AU_RECIP_MIDDLE	N/A	Not populated	N/A
		RA_AU_RECIP_FIRST	N/A	Not populated	N/A
		RA_AU_RECIP_SUFFIX	N/A	Not populated	N/A
MICR	EMPL_LAST_NM	RA_AU_PROV_LAST_OR_LGL_NM	String (20)		R
MICR	EMPL_MIDDLE_NM	RA_AU_PROV_MIDDLE_OR_LGL_IND	String (15)		O
MICR	EMPL_FIRST_NM	RA_AU_PROV_FIRST	String (15)		R
PAY_CHECK_EXT	EMPL_SUFFIX_CD	RA_AU_PROV_SUFFIX	String (5)		O
		RA_AU_PAYEE_GUARDIAN_STREET_ADDRESS	N/A	Not populated	N/A
		RA_AU_PAYEE_GUARDIAN_CITY	N/A	Not populated	N/A
		RA_AU_PAYEE_GUARDIAN_STATE	N/A	Not populated	N/A
		RA_AU_PAYEE_GUARDIAN_ZIP	N/A	Not populated	N/A
		RA_AU_PAYEE_GUARDIAN_ZIP_PLUS_4	N/A	Not populated	N/A
		RA_AU_PAYMENT_DT	N/A	Not populated	N/A
		RA_AU_CHECK_PAY_FROM	N/A	Not populated	N/A
		RA_AU_CHECK_PAY_TO	N/A	Not populated	N/A
		RA_AU_HOURS_SUBMITTED	N/A	Not populated	N/A
		RA_AU_HOURS_OVER_AUTH	N/A	Not populated	N/A
		RA_AU_CHECK_HOURS_PAID (provide in hours and minutes - HH:MM)	N/A	Not populated	N/A
		RA_AU_IHSS_R_RATE	N/A	Not populated	N/A
		RA_AU_CHECK_GROSS	N/A	Not populated	N/A
		RA_AU_PAYMENT_ADJUSTMENT	N/A	Not populated	N/A
		RA_AU_CHECK_FEDERAL_EIC	N/A	Not populated	N/A
		RA_AU_CHECK_FED_ADD	N/A	Not populated	N/A
		RA_AU_CHECK_STATE_TAX	N/A	Not populated	N/A
		RA_AU_CHECK_STATE_ADD	N/A	Not populated	N/A

		RA_AU_CHECK_FICA	N/A	Not populated	N/A
		RA_AU_CHECK_MEDI	N/A	Not populated	N/A
		RA_AU_CHECK_SDI_DIEC	N/A	Not populated	N/A
		RA_AU_CHECK_SOC	N/A	Not populated	N/A
		RA_AU_CHECK_LIEN	N/A	Not populated	N/A
		RA_AU_CHECK_HEALTH	N/A	Not populated	N/A
		RA_AU_CHECK_UDUES	N/A	Not populated	N/A
		RA_AU_CHECK_UTRST	N/A	Not populated	N/A
		RA_AU_CHECK_UPOLC	N/A	Not populated	N/A
		RA_AU_CHECK_UINIT	N/A	Not populated	N/A
		RA_AU_CHECK_UOTHR	N/A	Not populated	N/A
		RA_AU_CHECK_RECOVERY	N/A	Not populated	N/A
		RA_AU_CHECK_NET_PAY	String (1)	Always defaulted to "0"	R
		RA_AU_CHECK_GROSS_YTD	N/A	Not populated	N/A
		RA_AU_PAYMENT_ADJUSTMENT_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_FEDERAL/EIC_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_FED_ADD_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_STATE_TAX_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_STATE_ADD_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_FICA_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_MEDI_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_SDI_DIEC_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_SOC_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_LIEN_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_HEALTH_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_UDUES_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_UTRST_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_UPOLC_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_UINIT_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_UOTHR_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_RECOVERY_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_NET_PAY_YTD	N/A	Not populated	N/A
		RA_AU_ADV_WARRANT_NO	N/A	Not populated	N/A
MICR	EFT_ROUTING_NO	RA_ABA_ROUTING_NO	String (9)		R
MICR	SAVING_CHECKING_ID	RA_SAVING_CHECKING_ID	String (1)		R
MICR	EFT_BANK_ACCOUNT_NM	RA_EFT_BANK_ACCT_NM	String (17)		R
		AU_RECIPIENT_ID	N/A	Not populated	N/A
		AU_PROVIDER_ID	N/A	Not populated	N/A
		AU_PAYEE_ID	N/A	Not populated	N/A
		AU_EMPLOYEE_ID	N/A	Not populated	N/A
		AU_OFFICE	N/A	Not populated	N/A
		AU_ELIG_PAY_TYPE	N/A	Not populated	N/A
		AU_RECIP_ADDRESS	N/A	Not populated	N/A
		AU_RECIP_CITY	N/A	Not populated	N/A
		AU_RECIP_STATE	N/A	Not populated	N/A

		AU_RECIP_ZIP	N/A	Not populated	N/A
		AU_RECIP_ZIP_PLUS_4	N/A	Not populated	N/A
		AU_RECIP_SSN	N/A	Not populated	N/A
		AU_PROVIDER_TYPE	N/A	Not populated	N/A
		AU_PROV_SSN	N/A	Not populated	N/A
		AU_SOC_WORKER_NUM	N/A	Not populated	N/A
		AU_IHSS_P_AUTHHRS	N/A	Not populated	N/A
		AU_PROGRAM_CODE	N/A	Not populated	N/A
		AU_CIVIL_SVC_STA_CD	N/A	Not populated	N/A
		TIMESHEET_NUMBER	N/A	Not populated	N/A
		RA_PRE_NOTE_IND	As defined in XSD	"Y"	R

Table 14 – FIN EFT Prenote

Source: Payroll		Target: File to BPM			
Table Name/ File Name/ WSDL Name	Field Name	Field Name	Data Type (Format-length)	Description(Data Transformation rules)	R/O(Required or optional)
		RA_AU_RECIP_LAST	N/A	Not populated	N/A
		RA_AU_RECIP_MIDDLE	N/A	Not populated	N/A
		RA_AU_RECIP_FIRST	N/A	Not populated	N/A
		RA_AU_RECIP_SUFFIX	N/A	Not populated	N/A
PRENOTE_PRG_VW	LGL_NM	RA_AU_PROV_LAST_OR_LGL_NM	String (60)		R
		RA_AU_PROV_MIDDLE_OR_LGL_IND	String (10)	"LEGAL NAME"	R
		RA_AU_PROV_FIRST	N/A	Not populated	N/A
		RA_AU_PROV_SUFFIX	N/A	Not populated	N/A
		RA_AU_PAYEE_GUARDIAN_STREET_ADDRESS	N/A	Not populated	N/A
		RA_AU_PAYEE_GUARDIAN_CITY	N/A	Not populated	N/A
		RA_AU_PAYEE_GUARDIAN_STATE	N/A	Not populated	N/A
		RA_AU_PAYEE_GUARDIAN_ZIP	N/A	Not populated	N/A
		RA_AU_PAYEE_GUARDIAN_ZIP_PLUS_4	N/A	Not populated	N/A
		RA_AU_PAYMENT_DT	N/A	Not populated	N/A
		RA_AU_CHECK_PAY_FROM	N/A	Not populated	N/A
		RA_AU_CHECK_PAY_TO	N/A	Not populated	N/A
		RA_AU_HOURS_SUBMITTED	N/A	Not populated	N/A
		RA_AU_HOURS_OVER_AUTH	N/A	Not populated	N/A
		RA_AU_CHECK_HOURS_PAID (provide in hours and minutes - HH:MM)	N/A	Not populated	N/A
		RA_AU_IHSS_R_RATE	N/A	Not populated	N/A
		RA_AU_CHECK_GROSS	N/A	Not populated	N/A
		RA_AU_PAYMENT_ADJUSTMENT	N/A	Not populated	N/A
		RA_AU_CHECK_FEDERAL_EIC	N/A	Not populated	N/A
		RA_AU_CHECK_FED_ADD	N/A	Not populated	N/A
		RA_AU_CHECK_STATE_TAX	N/A	Not populated	N/A
		RA_AU_CHECK_STATE_ADD	N/A	Not populated	N/A
		RA_AU_CHECK_FICA	N/A	Not populated	N/A
		RA_AU_CHECK_MEDI	N/A	Not populated	N/A
		RA_AU_CHECK_SDI_DIEC	N/A	Not populated	N/A

		RA_AU_CHECK_SOC	N/A	Not populated	N/A
		RA_AU_CHECK_LIEN	N/A	Not populated	N/A
		RA_AU_CHECK_HEALTH	N/A	Not populated	N/A
		RA_AU_CHECK_UDUES	N/A	Not populated	N/A
		RA_AU_CHECK_UTRST	N/A	Not populated	N/A
		RA_AU_CHECK_UPOLC	N/A	Not populated	N/A
		RA_AU_CHECK_UINIT	N/A	Not populated	N/A
		RA_AU_CHECK_UOTHR	N/A	Not populated	N/A
		RA_AU_CHECK_RECOVERY	N/A	Not populated	N/A
		RA_AU_CHECK_NET_PAY	String (1)	Always defaulted to "0"	R
		RA_AU_CHECK_GROSS_YTD	N/A	Not populated	N/A
		RA_AU_PAYMENT_ADJUSTMENT_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_FEDERAL/EIC_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_FED_ADD_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_STATE_TAX_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_STATE_ADD_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_FICA_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_MEDI_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_SDI_DIEC_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_SOC_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_LIEN_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_HEALTH_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_UDUES_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_UTRST_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_UPOLC_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_UINIT_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_UOTHR_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_RECOVERY_YTD	N/A	Not populated	N/A
		RA_AU_CHECK_NET_PAY_YTD	N/A	Not populated	N/A
		RA_AU_ADV_WARRANT_NO	N/A	Not populated	N/A
PRENOTE_PRG_VW	ABA_NO	RA_ABA_ROUTING_NO	String (9)		R
PRENOTE_PRG_VW	ACCT_T YP	RA_SAVING_CHECKING_ID	String (1)	Set to "C" if "2" Set to "S" if "1"	R
PRENOTE_PRG_VW	ACCT_NO	RA_EFT_BANK_ACCT_NM	String (17)		R
		AU_RECIPIENT_ID	N/A	Not populated	N/A
		AU_PROVIDER_ID	N/A	Not populated	N/A
		AU_PAYEE_ID	N/A	Not populated	N/A
		AU_EMPLOYEE_ID	N/A	Not populated	N/A
		AU_OFFICE	N/A	Not populated	N/A
		AU_ELIG_PAY_TYPE	N/A	Not populated	N/A
		AU_RECIP_ADDRESS	N/A	Not populated	N/A
		AU_RECIP_CITY	N/A	Not populated	N/A
		AU_RECIP_STATE	N/A	Not populated	N/A
		AU_RECIP_ZIP	N/A	Not populated	N/A
		AU_RECIP_ZIP_PLUS_4	N/A	Not populated	N/A
		AU_RECIP_SSN	N/A	Not populated	N/A
		AU_PROVIDER_TYPE	N/A	Not populated	N/A

		AU_PROV_SSN	N/A	Not populated	N/A
		AU_SOC_WORKER_NUM	N/A	Not populated	N/A
		AU_IHSS_P_AUTHHRS	N/A	Not populated	N/A
		AU_PROGRAM_CODE	N/A	Not populated	N/A
		AU_PAY_EVNT_TYPE	N/A	Not populated	N/A
		TIMESHEET_NUMBER	N/A	Not populated	N/A
		RA_PRE_NOTE_IND	As defined in XSD	"Y"	R

Trigger Events

Pre-Note jobs will be triggered by the script called by AutoSys. They will be run weekly (every Friday).

Processing Criteria

There will be two files created by two separate batch jobs and sent to BPM:

- HRM Pre-Note file created by HRM Pre-Note interface
- Financial Pre-Note file created by FIN Pre-Note interface.

The above files will have the same format and will be built based on the same XML schema.

HRM Pre-Note interface will extract the necessary information from MICR and PAY_CHECK_EXT tables. It will extract all records where the Prenote Effective Date is greater than the current application data minus seven days, Net Pay Distribution Type ID is Prenote and bank information is populated.

FIN Pre-Note interface will extract necessary information from PRENOTE_PRG_VW table. The job will extract all records where the EFT Status is Prenote.

Error Processing

Error No.	Error Code	Description
1	UTLS1002	Batch Processing Error
2	GEN1006	General Exception Occurred

Related Components

Business Process Functions

N/A

Business Flows

N/A

Business Rules

N/A

Tasks/Notifications

N/A

Screens

N/A

Reports

N/A

DSD 19/Payroll – Warrant Management/Business Process /External Interfaces/Daily Payment – PRDS109A

CI	Document Name
CI-116363 - DSD EINTF Daily Payment PRDS109A IMPLEMENTED	DSD_EINTF_Daily_Payment_PRDS109A.doc

This is the interface between CMIPS and the Business Automation Workflow (BAW) file server. The interface will extract all warrant-specific information from Advantage (HRM) tables for warrants/payments issued on specified cycle dates and for specified GTN run numbers. This information will be stored in the file and will include Remittance Advice (RA) and Audit information for payments processed for recipients and providers and is sorted by zip code. The file will be sent to BAW which will later transform the data into the format specified by SCO and combine it with a vendor payment files (EFT and Non-EFT) received from Advantage Financial and transmit it to the SCO.

Data Elements of Interface

Table – HRM (EFT and Non-EFT) Daily Payment

Source: Payroll		Target: File to BPM			
Table Name/ File Name/ WSDL Name	Field Name	Field Name	Data Type (Format-length)	Description(Data Transformation rules)	R/O (Required or optional)
TAXING_ENTITY	CNTAC_LAST_NM	RA_AU_RECIP_LAST	String (20)		R
TAXING_ENTITY	CNTAC_MIDDLE_NM	RA_AU_RECIP_MIDDLE	String (15)		O
TAXING_ENTITY	CNTAC_FIRST_NM	RA_AU_RECIP_FIRST	String (15)		R
TAXING_ENTITY	CNTAC_NM_SUFFIX	RA_AU_RECIP_SUFFIX	String (5)		O
PAY_CH_ECK_EXT	EMPL_LAST_NM	RA_AU_PROV_L AST_OR_LGL_NM	String (20)		R
PAY_CH_ECK_EXT	EMPL_MIDDLE_NM	RA_AU_PROV_MIDDLE_OR_LG L_IND	String (15)		O
PAY_CH_ECK_EXT	EMPL_FIRST_NM	RA_AU_PROV_F IRST	String (15)		R
PAY_CH_ECK_EXT	EMPL_SUFFIX_CD	RA_AU_PROV_S UFFIX	String (5)		O
PAY_CH_ECK_EXT	STR_1_NM	RA_AU_PAYEE_GUARDIAN _STREET_ADDR ESS_1	String (75)		R
PAY_CH_ECK_EXT	STR_2_NM	RA_AU_PAYEE_GUARDIAN _STREET_ADDR ESS_2	String (75)		O
PAY_CH_ECK_EXT	CITY_NM	RA_AU_PAYEE_GUARDIAN_CITY	String (28)		R
PAY_CH_ECK_EXT	ST_CD	RA_AU_PAYEE_GUARDIAN_STA TE	String (2)		R
PAY_CH_ECK_EXT	ZIP (first 5 characters)	RA_AU_PAYEE_GUARDIAN_ZIP	String (5)		R
PAY_CH_ECK_EXT	ZIP (last 4 characters, if not empty)	RA_AU_PAYEE_GUARDIAN_ZIP PLUS_4	String (4)		O
PAY_CH_ECK	CHK_DT	RA_AU_PAYME NT_DT	Date (CCY Y-MM-DD)		R

PAY_CHECK	PPRD_STRT_DT	RA_AU_CHECK_PAY_FROM	Date (CCY Y-MM-DD)		R
PAY_CHECK	PPRD_END_DT	RA_AU_CHECK_PAY_TO	Date (CCY Y-MM-DD)		R
PAY_DETL	INPUT_AM	RA_AU_HOURS_SUBMITTED	String (HHH:MM)	Represents all hours submitted on the timesheet/payment request. RA_AU_HOURS_OVER_AUTH + RA_AU_CHECK_HOURS_PAID	R
PAY_DETL	INPUT_AM	RA_AU_HOURS_OVER_AUTH	String (HHH:MM)	Represents the hours that were not paid (cutback service hours, cutback overtime exemption limit and cutback travel hours). Sum of input_am from pay_detl for a paycheck where evtnt_type_cd first character = "C" and payAdjustType <> "CVDR". That will represent the cutback minutes. Convert it to hours and minutes. If PAY_DETL.INPUT_AM (calculated as HHH:MM) for where EVNT_TYP_CD = "CBHN".	O
PAY_DETL	INPUT_AM	RA_AU_CHECK_HOURS_PAID (provide in hours and minutes - HH:MM)	String (HHH:MM)	Represents the hours that were paid. Sum of input_am from pay_detl for a paycheck where the evtnt_typ_cd first character = "G", "A", "T", "V", "N", "S", or "K" or first character = "C" and payAdjustType = "CVDR" If PAY_DETL.INPUT_AM (calculated as HHH:MM) in line where DOC_CD = 'TADJ' and EVNT_TYP_CD LIKE 'N%' and EVNT_TYP_CD <> 'CBHN' and PAY_DETL_EXT.PAY_INPUT_DEF_ID = 'I'	R
PAY_DETL	PAY_RT_AM	RA_AU_IHSS_RATE	String (11,4)	Determine the rate amount based on Pay Detail records for the paycheck. IF a timesheet paid at the Step 02 rate is combined with a CP Time Claim paid at the Step 01 rate, the ihssRRate reflects the Step 02 rate in the xml. OR IF the transaction is a Training Time Claim only, the ihssRRate will be set to the Step 01 rate, and the eligPayType will be set to "H" OR IF the transaction is a Training Incentive Claim only, the ihssRRate will be set to 0, and the eligPayType will be set to "X" Note: The highest value from the PAY_RATE of HoursWageRate query will be displayed in the xml tags based on the EVENT_TYPE_CD.	R
PAY_DETL	INPUT_AM	RA_AU_HOURS_PAID_OVERTIME	String (HHH:MM)	Represents overtime hours	
PAY_CHECK	PPRD_CHK_GROS_AM	RA_AU_CHECK_GROSS	Numeric (9,2)		R
PAY_DETL	PAY_CHK_PAY_SUMM	RA_AU_PAYMNT_REGULAR	Numeric (9,2)	Determine based on pay detail, event type and event category records for the paycheck. evtnt type	O
DED_DETL DED_DETL_EXT	DED_AM_CHK_PRNT_SEQ_NO	RA_AU_CHECK_FEDERAL_EIC	Numeric (9,2)	Sum of DED_DETL.DED_AM where CHK_PRNT_SEQ_NO = 4 on corresponding DED_DETL_EXT records or where DEDTYP_CD in ('2FDTX', '2FDTR') Based on assumption that one of these fields will always be 0.	O
DED_DETL DED_DETL_EXT	DED_AM_CHK_PRNT_SEQ_NO	RA_AU_CHECK_FED_ADD	Numeric (9,2)	Sum of DED_DETL.DED_AM where CHK_PRNT_SEQ_NO = 5 on corresponding DED_DETL_EXT records or where DEDTYP_CD in ('2FDTA', '2FDAR')	O
DED_DETL DED_DETL_EXT	DED_AM_CHK_PRNT_SEQ_NO	RA_AU_CHECK_STATE_TAX	Numeric (9,2)	Sum of DED_DETL.DED_AM where CHK_PRNT_SEQ_NO = 6 on corresponding DED_DETL_EXT records or where DEDTYP_CD in ('2STTX', '2STTR')	O
DED_DETL DED_DETL_EXT	DED_AM_CHK_PRNT_SEQ_NO	RA_AU_CHECK_STATE_ADD	Numeric (9,2)	Sum of DED_DETL.DED_AM where CHK_PRNT_SEQ_NO = 7 on corresponding DED_DETL_EXT records or where DEDTYP_CD in ('2STXA', '2STAR')	O
DED_DETL DED_DETL_EXT	DED_AM_CHK_PRNT_SEQ_NO	RA_AU_CHECK_FICA	Numeric (9,2)	Sum of DED_DETL.DED_AM where CHK_PRNT_SEQ_NO = 8 on corresponding DED_DETL_EXT records or where DEDTYP_CD in ('2FICE', '2FICA', '2FICR', '2FICF')	O
DED_DETL DED_DETL_EXT	DED_AM_CHK_PRNT_SEQ_NO	RA_AU_CHECK_MEDI	Numeric (9,2)	Sum of DED_DETL.DED_AM where CHK_PRNT_SEQ_NO = 9 on corresponding DED_DETL_EXT records or where DEDTYP_CD in ('2MEDE', '2MEDA', '2MEDR', '2MEDF')	O

DED_DE TL DED_DE TL_EXT	DED_AM CHK_PRNT_SEQ_NO	RA_AU_CHECK_ SDI_DIEC	Numeric (9,2)	Sum of DED_DETL.DED_AM where CHK_PRNT_SEQ_NO = 10 on corresponding DED_DETL_EXT records or where DEDTYP_CD in ('2SDIE', '2SDIA', '2SDIF', '2SDIR', '2SDIEC')	O
DED_DE TL DED_DE TL_EXT	DED_AM CHK_PRNT_SEQ_NO	RA_AU_CHECK_ SOC	Numeric (9,2)	Sum of DED_DETL.DED_AM where CHK_PRNT_SEQ_NO = 11 on corresponding DED_DETL_EXT records or where DEDTYP_CD in ('ROBC', 'SOC', 'SOCAD', 'SOCC')	O
DED_DE TL DED_DE TL_EXT	DED_AM CHK_PRNT_SEQ_NO	RA_AU_CHECK_ LIEN	Numeric (9,2)	Sum of DED_DETL.DED_AM where CHK_PRNT_SEQ_NO = 13 on corresponding DED_DETL_EXT records or where DEDTYP_CD like 'LVB%', 'SLO%', 'IRS%', 'SSU%', 'CSU%', 'FTB%', 'LVF%', 'CRT%', 'LIE%', 'SS1%', 'SS2%', 'BAN%', 'EDD%', 'MSA%', 'MSU%', 'SSA%', 'WSU%', 'SHF%', 'WCD%', 'CSA%', 'NCD%', 'SLN%', 'VRG%'	O
DED_DE TL DED_DE TL_EXT	DED_AM CHK_PRNT_SEQ_NO	RA_AU_CHECK_ HEALTH	Numeric (9,2)	Sum DED_DETL.DED_AM where CHK_PRNT_SEQ_NO = 14 on corresponding DED_DETL_EXT records or where DEDTYP_CD in ('HBADJ', 'VISN', 'DENT', 'HLTH', 'MED')	O
DED_DE TL DED_DE TL_EXT	DED_AM CHK_PRNT_SEQ_NO	RA_AU_CHECK_ UDUES	Numeric (9,2)	Sum of DED_DETL.DED_AM where CHK_PRNT_SEQ_NO = 15 on corresponding DED_DETL_EXT records or where DEDTYP_CD in ('UDADJ', 'DUESC', 'UDUES')	O
DED_DE TL DED_DE TL_EXT	DED_AM CHK_PRNT_SEQ_NO	RA_AU_CHECK_ UTRST	Numeric (9,2)	Sum of DED_DETL.DED_AM where CHK_PRNT_SEQ_NO = 16 on corresponding DED_DETL_EXT records or where DEDTYP_CD in ('UTRST', 'UTADJ')	O
DED_DE TL DED_DE TL_EXT	DED_AM CHK_PRNT_SEQ_NO	RA_AU_CHECK_ UPEOP	Numeric (9,2)	Sum of DED_DETL.DED_AM where CHK_PRNT_SEQ_NO = 17 on corresponding DED_DETL_EXT records or where DEDTYP_CD in ('UCOPE', 'UPADJ', 'UPEOP', 'UCADJ')	O
DED_DE TL DED_DE TL_EXT	DED_AM CHK_PRNT_SEQ_NO	RA_AU_CHECK_ UNIT	Numeric (9,2)	Sum of DED_DETL.DED_AM where CHK_PRNT_SEQ_NO = 18 on corresponding DED_DETL_EXT records or where DEDTYP_CD in ('UIADJ', 'UNIT')	O
DED_DE TL DED_DE TL_EXT	DED_AM CHK_PRNT_SEQ_NO	RA_AU_CHECK_ UOTHR	Numeric (9,2)	Sum of DED_DETL.DED_AM where CHK_PRNT_SEQ_NO = 19 on corresponding DED_DETL_EXT records or where DEDTYP_CD in ('UOADJ', 'UOTHR')	O
DED_DE TL DED_DE TL_EXT	DED_AM CHK_PRNT_SEQ_NO	RA_AU_CHECK_ RECOVERY	Numeric (9,2)	Sum of DED_DETL.DED_AM where CHK_PRNT_SEQ_NO = 12 on corresponding DED_DETL_EXT records or where DEDTYP_CD LIKE 'OP%' or 'TR%'	O
DED_DE TL DED_DE TL_EXT	DED_AM CHK_PRNT_SEQ_NO	RA_AU_CHECK_ CART	Numeric (9,2)	Sum of DED_DETL.DED_AM where CHK_PRNT_SEQ_NO = 21 on corresponding DED_DETL_EXT records	O
PAY_CH ECK	PAY_CHK_AM	RA_AU_CHECK_ NET_PAY	Numeric (9,2)		R
PAY_CH K_PAY_S UMM	QTR_1_CASH_PAY_AM + QTR_2_CASH_PAY_AM + QTR_3_CASH_PAY_AM + QTR_4_CASH_PAY_AM	RA_AU_CHECK_ GROSS_YTD	Numeric (9,2)	where FRNG_PY_TYP_FL = 'N'	R
PAY_CH K_PAY_S UMM	QTR_1_CASH_PAY_AM + QTR_2_CASH_PAY_AM + QTR_3_CASH_PAY_AM + QTR_4_CASH_PAY_AM	RA_AU_PAYME NT_REGULAR_Y TD	Numeric (9,2)	Where CHK_PRNT_SEQ_NO in (1,6) and FRNG_PY_TYP_FL = 'N'	O
PAY_CH K_PAY_S UMM	QTR_1_CASH_PAY_AM + QTR_2_CASH_PAY_AM + QTR_3_CASH_PAY_AM + QTR_4_CASH_PAY_AM	RA_AU_PAYME NT_ADJUSTMEN T_YTD	Numeric (9,2)	Determine based on pay detail, event type and event category records for the paycheck. where CHK_PRNT_SEQ_NO = 2 and FRNG_PY_TYP_FL = 'N'	O
PAY_CH K_DED_S UM	DED_1ST_QTR_AM + DED_2ND_QTR_AM + DED_3RD_QTR_AM + DED_4TH_QTR_AM	RA_AU_CHECK_ FEDERAL /EIC_YTD	Numeric (9,2)	where CHK_PRNT_SEQ_NO = 4	O

PAY_CHK_DESUM	DED_1ST_QTR_AM + DED_2ND_QTR_AM + DED_3RD_QTR_AM + DED_4TH_QTR_AM	RA_AU_CHECK_FED_ADD_YTD	Numeric (9,2)	where CHK_PRNT_SEQ_NO = 5	O
PAY_CHK_DESUM	DED_1ST_QTR_AM + DED_2ND_QTR_AM + DED_3RD_QTR_AM + DED_4TH_QTR_AM	RA_AU_CHECK_STATE_TAX_YTD	Numeric (9,2)	where CHK_PRNT_SEQ_NO = 6	O
PAY_CHK_DESUM	DED_1ST_QTR_AM + DED_2ND_QTR_AM + DED_3RD_QTR_AM + DED_4TH_QTR_AM	RA_AU_CHECK_STATE_ADD_YTD	Numeric (9,2)	where CHK_PRNT_SEQ_NO = 7	O
PAY_CHK_DESUM	DED_1ST_QTR_AM + DED_2ND_QTR_AM + DED_3RD_QTR_AM + DED_4TH_QTR_AM	RA_AU_CHECK_FICA_YTD	Numeric (9,2)	where CHK_PRNT_SEQ_NO = 8	O
PAY_CHK_DESUM	DED_1ST_QTR_AM + DED_2ND_QTR_AM + DED_3RD_QTR_AM + DED_4TH_QTR_AM	RA_AU_CHECK_MEDI_YTD	Numeric (9,2)	where CHK_PRNT_SEQ_NO = 9	O
PAY_CHK_DESUM	DED_1ST_QTR_AM + DED_2ND_QTR_AM + DED_3RD_QTR_AM + DED_4TH_QTR_AM	RA_AU_CHECK_SDI_DIEC_YTD	Numeric (9,2)	where CHK_PRNT_SEQ_NO = 10	O
PAY_CHK_DESUM	DED_1ST_QTR_AM + DED_2ND_QTR_AM + DED_3RD_QTR_AM + DED_4TH_QTR_AM	RA_AU_CHECK_SOC_YTD	Numeric (9,2)	where CHK_PRNT_SEQ_NO = 11	O
PAY_CHK_DESUM	DED_1ST_QTR_AM + DED_2ND_QTR_AM + DED_3RD_QTR_AM + DED_4TH_QTR_AM	RA_AU_CHECK_LIEN_YTD	Numeric (9,2)	where CHK_PRNT_SEQ_NO = 13	O
PAY_CHK_DESUM	DED_1ST_QTR_AM + DED_2ND_QTR_AM + DED_3RD_QTR_AM + DED_4TH_QTR_AM	RA_AU_CHECK_HEALTH_YTD	Numeric (9,2)	where CHK_PRNT_SEQ_NO = 14	O
PAY_CHK_DESUM	DED_1ST_QTR_AM + DED_2ND_QTR_AM + DED_3RD_QTR_AM + DED_4TH_QTR_AM	RA_AU_CHECK_UDUES_YTD	Numeric (9,2)	where CHK_PRNT_SEQ_NO = 15	O
PAY_CHK_DESUM	DED_1ST_QTR_AM + DED_2ND_QTR_AM + DED_3RD_QTR_AM + DED_4TH_QTR_AM	RA_AU_CHECK_UTRST_YTD	Numeric (9,2)	where CHK_PRNT_SEQ_NO = 16	O
PAY_CHK_DESUM	DED_1ST_QTR_AM + DED_2ND_QTR_AM + DED_3RD_QTR_AM + DED_4TH_QTR_AM	RA_AU_CHECK_UPOLC_YTD	Numeric (9,2)	where CHK_PRNT_SEQ_NO = 17	O
PAY_CHK_DESUM	DED_1ST_QTR_AM + DED_2ND_QTR_AM + DED_3RD_QTR_AM + DED_4TH_QTR_AM	RA_AU_CHECK_UNIT_YTD	Numeric (9,2)	where CHK_PRNT_SEQ_NO = 18	O
PAY_CHK_DESUM	DED_1ST_QTR_AM + DED_2ND_QTR_AM + DED_3RD_QTR_AM + DED_4TH_QTR_AM	RA_AU_CHECK_UOTHR_YTD	Numeric (9,2)	where CHK_PRNT_SEQ_NO = 19	O
PAY_CHK_DESUM	DED_1ST_QTR_AM + DED_2ND_QTR_AM + DED_3RD_QTR_AM + DED_4TH_QTR_AM	RA_AU_CHECK_RECOVERY_YTD	Numeric (9,2)	where CHK_PRNT_SEQ_NO = 12	O
PAY_CHK_DESUM	DED_1ST_QTR_AM + DED_2ND_QTR_AM + DED_3RD_QTR_AM + DED_4TH_QTR_AM	RA_AU_CHECK_CART_YTD	Numeric (9,2)	where CHK_PRNT_SEQ_NO = 21	O
PAY_CHK_DESUM	PDED_1ST_QTR_AM + DED_2ND_QTR_AM + DED_3RD_QTR_AM + DED_4TH_QTR_AM	RA_AU_CHECK_NET_PAY_YTD	Numeric (9,2)	where CHK_PRNT_SEQ_NO = 99 and FRNG_PY_TYP_FL = 'N'	R
PAY_CHECK	CHK_NO	RA_AU_ADV_WARRANT_NO	String (15)		R
MICR	EFT_ROUTING_NO	RA_ABA_ROUTING_NO	String (9)		O

MICR	SAVING_CHECKING_ID	RA_SAVING_CHECKING_ID	String (1)		O
MICR	EFT_BANK_ACCT_NM	RA_EFT_BANK_ACCT_NM	String (17)		O
TAXING_ENTITY	TAXING_ENTITY_CD	AU_RECIPIENT_ID	String (7)		R
PAY_CHECK_EXT	EMPL_ALT_ID_NO	AU_PROVIDER_ID	String (9)		R
PAY_CHECK_EXT	EMPL_ALT_ID_NO	AU_PAYEE_ID	String (9)		R
PAY_CHECK_EXT	EMPLOYEE_ID	AU_EMPLOYEE_ID	String (10)		R
PAY_CHECK_EXT	HOME_UNIT_CD	AU_OFFICE	String (4)		R
TAXING_ENTITY	ST_TAX_ID_9	AU_ELIG_TYPE	String (1)	Recipient Impairment. Values are 'N' = Non-Severely 'S' = Severely 'O' = Not applicable	R
PAY_CHECK	APPOINTMENT_ID	AU_ELIG_PAY_TYPE	String (1)	Values are: W = WPCS payment M = Meals payment H = Hourly (train, travel, service, sick leave hrs) X = Adjustments R = Reissues A = Advance Pay V = Vendor payments	R
TAXING_ENTITY	STR_1_NM + " " + STR_2_NM	AU_RECIP_ADDRESS	String (150)	From current timeslice TAXING_ENTITY.STR_1_NM + " " + TAXING_ENTITY.STR_2_NM (for taxing_entity_cd representing the Recipient)	R
TAXING_ENTITY	CITY	AU_RECIP_CITY	String (28)	From current timeslice TAXING_ENTITY.CITY (for taxing_entity_cd representing the Recipient)	R
TAXING_ENTITY	ST_CD	AU_RECIP_STATE	String (2)	From current timeslice TAXING_ENTITY.ST_CD (for taxing_entity_cd representing the Recipient)	R
TAXING_ENTITY	ZIP (first 5 characters)	AU_RECIP_ZIP	String (5)	From current timeslice TAXING_ENTITY.ZIP (first 5 characters) (for taxing_entity_cd representing the Recipient)	R
TAXING_ENTITY	ZIP (last 4 characters)	AU_RECIP_ZIP_PLUS_4	String (4)	From current timeslice TAXING_ENTITY.ZIP (last 4 characters) (for taxing_entity_cd representing the Recipient)	O
TAXING_ENTITY	ST_TAX_ID_1	AU_RECIP_SSN	String (9)	From current timeslice TAXING_ENTITY.ST_TAX_ID_1 (for taxing_entity_cd representing the Recipient)	R
PAY_CHECK_EXT	EMPL_ALT_ID_NO and TAXING_ENTITY_CD	AU_PROVIDER_TYPE	String (1)	If PAY_CHECK_EXT.EMPL_ALT_ID_NO = PAY_CHECK_EXT.TAXING_ENTITY_CD Then "R" Else "I"	R
PAY_CHECK_EXT	EMPL_SSN	AU_PROV_SSN	String (9)		R
TAXING_ENTITY	ST_TAX_ID_10	AU_SOC_WORKER_NUM	String (4)		R
TAXING_ENTITY	ST_TAX_ID_8	AU_IHSS_P_AU_THHRS	String (HHH:MM)	If Training Time Claim (TTC) pay period is prior to IHSS or WPCS Authorization Date, then set value to '00:00'	R
PAY_CHECK_EXT	HOME_DEPT_CD	AU_PROGRAM_CODE	String (4)	Set to "WPCS" if appointment ID = "W." Otherwise, set to pay_check_ext.home_dept_cd	R
PAY_DETL	DOC_ID	AU_PAY_EVNT_TYPE	String (4)	Derived from first for characters of DOC_ID For 'Meals' it is defaulted to 'GM.' Set do "MULTI" if multiple DOC_IDs exist for the paycheck.	R
PAY_DETL	DOC_ID	TIMESHEET_NUMBER	String (16)	Derived from DOC_ID if the payment is for IP or Advance Pay	O

N/A	N/A	RA_PRE_NOTE_IND	String (1)	For Daily Payment file this is set to "N"	R
PAY_CHECK_EXT	EMPL_FIRST_NM, EMPL_MIDDLE_NM, EMPL_LAST_NM,	RA_PAYEE_NAME	String (60)	Payee's Last Name + First Name + Middle Name	R
PAY_CHECK_EXT	STR_1_NM	RA_PAYEE_STREET_ADDRESS_1	String (75)	Street address 1. If payment is to a recipient who has a Designee (contact type conservator or guardian), then STR_1_NM source field will contain "C/O <designeeFullName>"	R
PAY_CHECK_EXT	STR_2_NM	RA_PAYEE_STREET_ADDRESS_2	String (75)	Street address 2. If payment is to a recipient who as a Designee (contact type conservator or guardian), then STR_2_NM source field will contain the complete Designee address. If Designee address is long enough to contain street 1 and street 2 elements, then they will be concatenated with "^" in between (i.e. Street1+"^"+Street2)	O
PAY_CHECK_EXT	CITY_NM	RA_PAYEE_CITY	String (28)		R
PAY_CHECK_EXT	ST_CD	RA_PAYEE_STATE	String (2)		R
PAY_CHECK_EXT	ZIP (all 10 characters)	RA_PAYEE_ZIP	String (10)	Add "-" after the first 5 digits, if there are more than 5 digits in the zip.	R
PAY_DETL	INPUT_AM	RA_CHECK_TRAVEL_HOURS (provide in hours and minutes - HH:MM)	String (HHH:MM)	Represents the Travel hours that were paid. Where substr(evtnt_tpy_cd, 1, 1) = 'T' or 'V'	O
PAY_DETL	INPUT_AM	RA_CHECK_OT_HOURS (provide in hours and minutes - HH:MM)	String (HHH:MM)	Represents the OT hours that were paid. Where substr(evtnt_tpy_cd, 1, 1) = 'F' or 'L'	O
PAY_DETL	INPUT_AM	RA_CHECK_SICKLEAVE_HOURS (provide in hours and minutes - HH:MM)	String (HHH:MM)	Represents the Sick Leave hours that were paid. Where substr(evtnt_tpy_cd, 1, 1) = 'S' or 'K' • Note: Reflects hours in the warrant that are from Accrued or Emergency Sick Leave.	O
PAY_DETL	CNTRCT_PY_AM	RA_CHECK_TRAVEL	Numeric (9,2)	Sum where substr(evtnt_tpy_cd, 1, 1) = 'T' or 'V'	O
PAY_CHECK_PAY_SUMM	Calculation	RA_CHECK_TRAVEL_YTD	Numeric (9,2)	Sum where frng_py_tpy_fl = 'N' and chk_prnt_seq_no = 5	O
PAY_DETL	CNTRCT_PY_AM	RA_CHECK_OVERTIME	Numeric (9,2)	Sum where substr(evtnt_tpy_cd, 1, 1) = 'F' or 'L'	O
PAY_CHECK_PAY_SUMM	QTR_1_CASH_PAY_AM + QTR_2_CASH_PAY_AM + QTR_3_CASH_PAY_AM + QTR_4_CASH_PAY_AM	RA_CHECK_OVERTIME_YTD	Numeric (9,2)	Sum where frng_py_tpy_fl = 'N' and chk_prnt_seq_no = 4	O
PAY_DETL	CNTRCT_PY_AM	RA_CHECK_SICKLEAVE	Numeric (9,2)	Sum where substr(evtnt_tpy_cd, 1, 1) = 'S' or 'K' • Note: Reflects wages in the warrant that are generated from Accrued or Emergency Sick Leave hours.	O
PAY_CHECK_PAY_SUMM	QTR_1_CASH_PAY_AM + QTR_2_CASH_PAY_AM + QTR_3_CASH_PAY_AM + QTR_4_CASH_PAY_AM	RA_CHECK_SICKLEAVE_YTD	Numeric (9,2)	where frng_py_tpy_fl = 'N' and chk_prnt_seq_no = 5 • Note: Reflects YTD wages from both Accrued and Emergency Sick Leave hours (if paid).	O
DED_DETL_DETL_EXT	DED_AM CHK_PRNT_SEQ_NO	RA_CHECK_TOTAL_DED	Numeric (9,2)	Sum DED_DETL.DED_AM where CHK_PRNT_SEQ_NO <> 99 on corresponding DED_DETL_EXT records	O
DED_DETL_EXT	YTD_DED_AM	RA_CHECK_TOTAL_DED_YTD	Numeric (9,2)	where CHK_PRNT_SEQ_NO <> 99	O

SICK_LEAVE_AVERAGE_CREDITS	HOURS_PAID	RA_CHECK_SICK_LEAVE_FYTD	String (HHH:MM)	Reflects only the Accrued Sick Leave hours paid for the current fiscal YTD	O
SICK_LEAVE_AVERAGE_CREDITS	HOURS_AVAILABLE	RA_CHECK_SICK_LEAVE_AVAIL	String (HHH:MM)	Reflects only the Accrued Sick Leave balance for the Fiscal Year.	O

Table – FIN EFT Daily Payment

Source: EFACHXML File		Target: File to BPM			
Table Name/ File Name/ WSDL Name	Field Name	Field Name	Data Type (Format-length)	Description(Data Transformation rules)	R/O (Required or optional)
		RA_AU_RECIP_LAST	N/A	Not populated	N/A
		RA_AU_RECIP_MIDDLE	N/A	Not populated	N/A
		RA_AU_RECIP_FIRST	N/A	Not populated	N/A
		RA_AU_RECIP_SUFFIX	N/A	Not populated	N/A
EFACHXML file	PYEE_LGL_NM	RA_AU_PROV_LAST_OR_LGL_NM	String (60)		R
		RA_AU_PROV_MIDDLE_OR_LGL_IND	N/A	Not populated	N/A
		RA_AU_PROV_FIRST	N/A	Not populated	N/A
		RA_AU_PROV_SUFFIX	N/A	Not populated	N/A
EFACHXML file	PYEE_AD_LN_1	RA_AU_PAYEE_GUARDIAN_STREET_ADDRESS_1	String (75)		R
EFACHXML file	PYEE_AD_LN_2	RA_AU_PAYEE_GUARDIAN_STREET_ADDRESS_2	String (75)		O
EFACHXML file	PYEE_CITY	RA_AU_PAYEE_GUARDIAN_CITY	String (28)		R
EFACHXML file	PYEE_ST	RA_AU_PAYEE_GUARDIAN_STATE	String (2)		R
EFACHXML file	PYEE_ZIP (first 5 characters)	RA_AU_PAYEE_GUARDIAN_ZIP	String (5)		R
EFACHXML file	PYEE_ZIP (last 4 characters)	RA_AU_PAYEE_GUARDIAN_ZIP_PLUS_4	String (4)		R
EFACHXML file	CLR_DT	RA_AU_PAYMENT_DT	Date	"CCYY-MM-DD"	R
		RA_AU_CHECK_PAY_FROM	N/A	Not populated	N/A
		RA_AU_CHECK_PAY_TO	N/A	Not populated	N/A
		RA_AU_HOURS_SUBMITTED	String (4)	Always defaulted to "0:00"	R
		RA_AU_HOURS_OVER_AUTH	String (4)	Always defaulted to "0:00"	R
		RA_AU_CHECK_HOURS_PAID (provide in hours and minutes - HH:MM)	String (4)	Always defaulted to "0:00"	R
		RA_AU_IHSS_R_RATE	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_GROSS	String (4)	Always defaulted to "0.00"	R
		RA_AU_PAYMENT_ADJUSTMENT	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_FEDERAL_EIC	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_FED_ADD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_STATE_TAX	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_STATE_ADD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_FICA	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_MEDI	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_SDI_DIEC	String (4)	Always defaulted to "0.00"	R

		RA_AU_CHECK_SOC	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_LIEN	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_HEALTH	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_UDUES	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_UTRST	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_UPOLC	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_UNIT	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_UOTHR	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_RECOVERY	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_CART	String (4)	Always defaulted to "0.00"	R
EFACHXML file	AMOUNT	RA_AU_CHECK_NET_PAY	Number(9,2)		R
		RA_AU_CHECK_GROSS_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_PAYMENT_ADJUSTMENT_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_FEDERAL/EIC_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_FED_ADD_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_STATE_TAX_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_STATE_ADD_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_FICA_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_MEDI_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_SDI_DIEC_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_SOC_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_LIEN_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_HEALTH_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_UDUES_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_UTRST_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_UPOLC_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_UNIT_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_UOTHR_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_RECOVERY_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_CART_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_NET_PAY_YTD	String (4)	Always defaulted to "0.00"	R
EFACHXML file	CHK_EFT_NO	RA_AU_ADV_WARRANT_NO	String (15)		R
EFACHXML file	RECV_DFI_ID	RA_ABA_ROUTING_NO	String (9)		R
EFACHXML file	TRAN_CD	RA_SAVING_CHECKING_ID	String (1)	"C" if "22" "S" if "32"	R
EFACHXML file	DFI_ACCT_NO	RA_EFT_BANK_ACCT_NM	String (17)		R
		AU_RECIPIENT_ID	N/A	Not populated	N/A
		AU_PROVIDER_ID	N/A	Not populated	N/A
EFACHXML file	RECV_NM	AU_PAYEE_ID	String (10)		R
EFACHXML file	RECV_NM	AU_EMPLOYEE_ID	String (10)		R
		AU_OFFICE	N/A	Not populated	N/A
		AU_ELIG_TYPE	String (1)	Not Populated	N/A
		AU_ELIG_PAY_TYPE	String (1)	"V"	R

		AU_RECIP_ADDRESS	N/A	Not populated	N/A
		AU_RECIP_CITY	N/A	Not populated	N/A
		AU_RECIP_STATE	N/A	Not populated	N/A
		AU_RECIP_ZIP	N/A	Not populated	N/A
		AU_RECIP_ZIP_PLUS_4	N/A	Not populated	N/A
		AU_RECIP_SSN	N/A	Not populated	N/A
EFACHXML file	PYEE_ALIAS_NM	AU_PROVIDER_TYPE	String (1)	"L"	R
		AU_PROV_SSN	N/A	Not populated	N/A
		AU_SOC_WORKER_NUM	N/A	Not populated	N/A
		AU_IHSS_P_AUTHHRS	String (4)	Always defaulted to "0:00"	R
		AU_PROGRAM_CODE	N/A	Not populated	N/A
		AU_PAY_EVNT_TYPE	N/A	Not populated	N/A
		TIMESHEET_NUMBER	N/A	Not populated	N/A
		RA_PRE_NOTE_IND	As defined in XSD	"N"	R
EFACHXML file	PYEE_LGL_NM	RA_PAYEE_NAME	String (60)	Name on the warrant	R
EFACHXML file	PYEE_AD_LN_1	RA_PAYEE_STREET_ADDRE SS_1	String (75)	This will be the address of the vendor	R
EFACHXML file	PYEE_AD_LN_2	RA_PAYEE_STREET_ADDRESS_2	String (75)		O
EFACHXML file	PYEE_CITY	RA_PAYEE_CITY	String (28)		R
EFACHXML file	PYEE_ST	RA_PAYEE_STATE	String (2)		R
EFACHXML file	PYEE_ZIP (first 5 characters)+ PYEE_ZIP (last 4 characters)	RA_PAYEE_ZIP	String (10)	Combine the zip and zip plus four. Place a – in between the two fields if there is a value in the PAYEE_ZIP)last 4 characters)	R

Table – FIN Non-EFT Daily Payment

Source: Offline Disbursement Check Printing File		Target: File to BPM			
Table Name/ File Name/ WSDL Name	Field Name	Field Name	Data Type (Format-length)	Description(Data Transformation rules)	R/O (Required or optional)
		RA_AU_RECIP_LAST	N/A	Not populated	N/A
		RA_AU_RECIP_MIDDLE	N/A	Not populated	N/A
		RA_AU_RECIP_FIRST	N/A	Not populated	N/A
		RA_AU_RECIP_SUFFIX	N/A	Not populated	N/A
Offline Disbursement Check Printing file	PYEE_NM_ON_CHK	RA_AU_PROV_LAST_OR_L GL_NM	String (60)		R
		RA_AU_PROV_MIDDLE_OR _LGL_IND	N/A	Not populated	N/A
		RA_AU_PROV_FIRST	N/A	Not populated	N/A
		RA_AU_PROV_SUFFIX	N/A	Not populated	N/A
Offline Disbursement Check Printing file	PYEE_AD_LN_1	RA_AU_PAYEE_GUARDIAN _STREET_ADDRESS_1	String (75)		R
Offline Disbursement Check Printing file	PYEE_AD_LN_2	RA_AU_PAYEE_GUARDIAN _STREET_ADDRESS_2	String (75)		O

Offline Disbursement Check Printing file	PYEE_CITY	RA_AU_PAYEE_GUARDIAN_CITY	String (28)		R
Offline Disbursement Check Printing file	PYEE_ST	RA_AU_PAYEE_GUARDIAN_STATE	String (2)		R
Offline Disbursement Check Printing file	PYEE_ZIP (first 5 characters)	RA_AU_PAYEE_GUARDIAN_ZIP	String (5)		R
Offline Disbursement Check Printing file	PYEE_ZIP (last 4 characters)	RA_AU_PAYEE_GUARDIAN_ZIP_PLUS_4	String (4)		O
Offline Disbursement Check Printing file	CHK_ISS_DT	RA_AU_PAYMENT_DT	Date (CCY Y-MM-DD)		R
		RA_AU_CHECK_PAY_FROM	N/A	Not populated	N/A
		RA_AU_CHECK_PAY_TO	N/A	Not populated	N/A
		RA_AU_HOURS_SUBMITTED	String (4)	Always defaulted to "0:00"	R
		RA_AU_HOURS_OVER_AUTH	String (4)	Always defaulted to "0:00"	R
		RA_AU_CHECK_HOURS_PAID (provide in hours and minutes - HH:MM)	String (4)	Always defaulted to "0:00"	R
		RA_AU_IHSS_R_RATE	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_GROSS	String (4)	Always defaulted to "0.00"	R
		RA_AU_PAYMENT_ADJUSTMENT	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_FEDERAL_EIC	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_FED_ADD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_STATE_TAX	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_STATE_ADD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_FICA	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_MEDI	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_SDI_DIEC	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_SOC	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_LIEN	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_HEALTH	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_UDUES	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_UTRST	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_UPOLC	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_UINIT	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_UOTHR	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_RECOVERY	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_CART	String (4)	Always defaulted to "0.00"	R
Offline Disbursement Check Printing file	CHK_AM	RA_AU_CHECK_NET_PAY	Numeric (9,2)		R
		RA_AU_CHECK_GROSS_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_PAYMENT_ADJUSTMENT_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_FEDERAL/EIC_YTD	String (4)	Always defaulted to "0.00"	R

		RA_AU_CHECK_FED_ADD_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_STATE TAX_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_STATE_ADD_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_FICA_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_MEDI_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_SDI_DIEC_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_SOC_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_LIEN_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_HEALTH_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_UDUES_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_UTRST_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_UPOLC_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_UNIT_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_UOTHR_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_RECOVER_Y_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_CART_YTD	String (4)	Always defaulted to "0.00"	R
		RA_AU_CHECK_NET_PAY_YTD	String (4)	Always defaulted to "0.00"	R
Offline Disbursement Check Printing file	CHK_NO	RA_AU_ADV_WARRANT_NO	String (15)		R
		RA_ABA_ROUTING_NO	N/A	Not populated	N/A
		RA_SAVING_CHECKING_ID	N/A	Not populated	N/A
		RA_EFT_BANK_ACCT_NM	N/A	Not populated	N/A
		AU_RECIPIENT_ID	N/A	Not populated	N/A
		AU_PROVIDER_ID	N/A	Not populated	N/A
Offline Disbursement Check Printing file	PYEE_CD	AU_PAYEE_ID	String (10)		R
Offline Disbursement Check Printing file	PYEE_CD	AU_EMPLOYEE_ID	String (10)		R
		AU_OFFICE	N/A	Not populated	N/A
		AU_ELIG_TYPE	String (1)	Not Populated	N/A
		AU_ELIG_PAY_TYPE	String (1)	"V"	R
		AU_RECIP_ADDRESS	N/A	Not populated	N/A
		AU_RECIP_CITY	N/A	Not populated	N/A
		AU_RECIP_STATE	N/A	Not populated	N/A
		AU_RECIP_ZIP	N/A	Not populated	N/A
		AU_RECIP_ZIP_PLUS_4	N/A	Not populated	N/A
		AU_RECIP_SSN	N/A	Not populated	N/A
Offline Disbursement Check Printing file	PYEE_ALIAS_NM	AU_PROVIDER_TYPE	String (1)	"L"	R
		AU_PROV_SSN	N/A	Not populated	N/A

		AU_SOC_WORKER_NUM	N/A	Not populated	N/A
		AU_IHSS_P_AUTHHRS	String (4)	Always defaulted to "0:00"	R
		AU_PROGRAM_CODE	N/A	Not populated	N/A
		AU_PAY_EVENT_TYPE	N/A	Not populated	N/A
		TIMESHEET_NUMBER	N/A	Not populated	N/A
		RA_PRE_NOTE_IND	String (1)	"N"	
Offline Disbursement Check Printing file	PYEE_NM_ON_CHK	RA_PAYEE_NAME	String (60)	Name on the warrant.	R
Offline Disbursement Check Printing file	PYEE_AD_LN_1	RA_PAYEE_STREET_ADDR ESS_1	String (75)	This will be the address of the vendor	R
Offline Disbursement Check Printing file	PYEE_AD_LN_2	RA_PAYEE_STREET_ADDRESS_2	String (75)		O
Offline Disbursement Check Printing file	PYEE_CITY	RA__PAYEE_CITY	String (28)		R
Offline Disbursement Check Printing file	PYEE_ST	RA_PAYEE_STATE	String (2)		R
Offline Disbursement Check Printing file	PYEE_ZIP (first 5 characters)+ PYEE_ZIP (last 4 characters)	RA_PAYEE_ZIP	String (10)	Combine the ZIP code and ZIP code plus four. Place a – in between the two fields if there is a value in the PAYEE_ZIP) last four characters)	R

Logical Layout/Transaction: Claim Header Record

Pos	Len	Name	Format	PIC Clause	Values
1	20	EFT ASCII as defined by SCO Blue Book			
21	10	CLAIM ID	AN	PIC X(10)	Value = IHSSCMIPSE
31	8006	EFT ASCII as defined by SCO Blue Book			

Trigger Events

Daily Payment jobs will be triggered by the script called by AutoSys. They will be run daily.

Processing Criteria

There are multiple files created by the Advantage Daily Payment batch jobs that will be reformatted into three (3) files and sent to the BPM:

- Multiple HRM Daily Payment Files that contain all non-vendor payments are created by the HRM Daily Payment File batch job and will be reformatted into one (1) file with the payment records sorted by zip code (ascending).
- One (1) FIN EFT Daily Payment File is created by the FIN EFT Daily Payment File batch job that contains vendor EFT payments
- One (1) FIN non-EFT Daily Payment File is created by the FIN non-EFT Daily Payment File batch job that contains vendor non-EFT payments

These files will have the same format and are built based on the same XML schema.

HRM Daily Payment batch job will extract the necessary information from Advantage HRM tables and send it to the BPM. Records will be extracted from paycheck-related tables where the following conditions are met:

GTN Run Number is within the series of numbers passed to the job as a parameter.

Check Status is "Paid". It can also be "Cleared" if the Net Pay Distribution Type is EFT.

The following warrants will be excluded from the set of warrants selected based on the above criteria:

- Regular Cycle \$0 checks with gross amount = \$0 and net amount = \$0.
- Supplemental Cycle \$0 with gross amount = \$0 and net amount = \$0 and no deductions.
- Checks reduced to \$0 by APDED deduction (Advance Pay reconciliation related checks that get sent to Case Management, not to SCO)

- Pre-note \$0 checks with gross amount = \$0 and net amount = \$0.

FIN EFT Daily Payment File and FIN non-EFT Daily Payment File batch jobs will extract the necessary information from the files generated by AD Chain Advantage Financial job. There will be no additional selections performed by the FIN EFT Daily Payment File and FIN non-EFT Daily Payment File jobs. All records provided to these jobs on the files will be reformatted and placed in the output file.

All three (3) files created by the Daily Payment File batch jobs will be sent to the BPM.

Table – Error Processing

Error No.	Error Code	Description
1	WARRCREATE1001	Current Taxing Entity record was not found
2	WARRCREATE1007	Gross YTD and Net YTD warrant totals were not found
3	WARRCREATE1011	Invalid length of County Code
6	WARRCREATE1014	Specified GTN Run Number(s) is not numeric
7	WARRCREATE1016	GTN Run Numbers are not specified
8	UTLS1002	Batch processing error
9	GEN1006	General Exception

Related Components

Business Process Functions

N/A

Business Flows

N/A

Business Rules

N/A

Tasks/Notifications

N/A

Screens

N/A

Reports

N/A

DSD 19/Payroll – Warrant Management/Business Process /External Interfaces/SCO Warrant Number Assignment – PRDR109D and PRDR109H

CI	Document Name
CI-116398 - DSD EINTF SCO Warrant Number Assignment IMPLEMENTED	DSD_EINTF_SCO_Warrant_Number_Assignment.doc

This is an interface between the SCO and CMIPS. The SCO provides the warrant and payment numbers for daily claim and daily EFT requests sent to the SCO. Advantage HRM-specific warrant numbers (for provider warrants) and their corresponding SCO warrant numbers will be stored in Advantage HRM tables. Advantage Financial-specific warrant numbers (for Vendor warrants) and their corresponding SCO warrant numbers will be stored in Advantage Financial tables. Having access to SCO-specific warrant numbers for each warrant will allow CMIPS users to view warrant information based on these numbers. It will also allow warrant status update requests received from SCO and STO.

Data Elements of Interface

This section will detail the data elements of this SCO Warrant Number Assignment interface.

Table – HRM SCO Warrant Number Assignment

Source: BPM			Target: PAY_CHECK Table in HRM		
Table Name/ File Name /WSDL Name	Field Name	Field Name	Data Type (Format-length)	Description (Data Transformation rules)	R/O (Required or optional)
Warrant Assignment File	CMIPS Warrant Number (Advantage Warrant Number – Will only be the right 5 bytes of the Adv. Warrant Number)			Compare to PAY_CHECK/CHK_NO (last 5 bytes only) when looking for a corresponding warrant.	R
Warrant Assignment File	Payee ID			Compare to PAY_CHECK /EMPL_ALT_ID_NO	R
Warrant Assignment File	SCO Warrant Amount			N/A	O
Warrant Assignment File	SCO Warrant Number	PAY_CHECK: SCO_WARRANT_NO	String	Update PAY_CHECK/ SCO_WARRANT_NO on the retrieved warrant row.	R
Warrant Assignment File	SCO Warrant Issue Date	PAY_CHECK: SCO_WARRANT_DT	Date	Update PAY_CHECK/ SCO_WARRANT_DT on the retrieved warrant row	R

Table – FIN SCO Warrant Number Assignment

Source: BPM		Target: R_AP_CHK_RECON			
Table Name/ File Name /WSDL Name	Field Name	Field Name	Data Type (Format-length)	Description (Data Transformation rules)	R/O (Required or optional)
Warrant Assignment File	CMIPS Warrant Number (Advantage Warrant Number – Will be the right 5 bytes of the Adv. Warrant Number only)			Compare to R_AP_CHK_RECON /CHK_EFT_NO (right 5 bytes) when looking for a corresponding warrant.	R
Warrant Assignment File	Payee Id			Compare to DOC_VEND/VEND_CUST_CD (last 10 bytes only)	R
Warrant Assignment File	SCO Warrant Amount			N/A	O
Warrant Assignment File	SCO Warrant Number	R_AP_CHK_RECON: SCO_WARRANT_NO	String	Update R_AP_CHK_RECON/ SCO_WARRANT_NO on the retrieved warrant row.	R
Warrant Assignment File	SCO Warrant Issue Date	R_AP_CHK_RECON: SCO_WARRANT_DT	Date	Update R_AP_CHK_RECON/ SCO_WARRANT_DT on the retrieved warrant row	R

Trigger Events

SCO Warrant Number Assignment jobs will be triggered by the script called by AutoSys. The jobs will be run daily.

Processing Criteria

Information about the warrant status (cleared, void) has to be maintained in CMIPS. These statuses will be received from SCO and STO through the interfaces. Since SCO and STO will not store Advantage Warrant Numbers, the unique identifier for each warrant in these interfaces will be the SCO warrant number and, along with the Provider ID, will need to be cross-referenced to the Advantage Warrant Number when received by CMIPS.

Paper warrants and EFT warrants are processed together in the same file.

Both Advantage HRM and Advantage Financial components of CMIPS will have to store the cross-reference between SCO and Advantage Warrant Numbers in the form of the SCO Warrant Number and SCO Warrant Date.

The file SCO Warrant Assignment file received from BPM will have the Advantage Warrant Number, SCO Warrant Number, SCO Warrant Date, Provider ID and Warrant Amount for each warrant created by SCO the previous day.

SCO Warrant Number Assignment interface will consist of two separate batch jobs:

- HRM Warrant Assignment job will process the entire input file. It will perform a look up to the Paycheck tables based on the SCO Warrant Number and Provider ID to see if the advantage warrant number exists. If there is a match, it will update the Paycheck tables with the correct SCO warrant number and SCO warrant date. If there is not a match, the job will write the record to a mismatch file.
- FIN Warrant Assignment job will process the mismatch file and perform a look up to the Check Reconciliation table in FIN based on the SCO Warrant Number and Provider (Vendor) ID. If a match is found, it will update the Check Reconciliation table with the SCO warrant number and SCO warrant date. If a match is not found, it will write a record to a final exception file.
- The HRM Batch job will also write to an output file those updates that processed successfully so that this file can be sent to CM. CM will receive this file on a nightly basis and process it so they will be in sync with what is in Payroll. The SCO Warrant Number and SCO Warrant Date fields will be passed to this file along with the employee ID and the Advantage Check Number.

Error Processing

Table – Error Codes

Error No.	Error Code	Description
1	WRNTASGNMT1001	Advantage Warrant Number Not Provided
2	WRNTASGNMT1002	Invalid Issue Date Provided
3	WRNTASGNMT1003	Invalid Warrant Amount Provided
4	WRNTASGNMT1004	SCO Warrant Number Not Provided
5	WRNTASGNMT1005	No Match Found. Writing to the HRM Out file for FIN lookup.
6	WRNTASGNMT1006	More Than One Record Returned
7	WRNTASGNMT1007	Record Exists With Same SCO Warrant Number and Same SCO Warrant Date
8	WRNTASGNMT1008	Status Of Warrant Is Not PAID
9	WRNTASGNMT1010	Match on Adv Warrant Number but Amount Does Not Match
10	WRNTASGNMT1011	Payee ID Not Provided
11	GEN1006	General Exception

Related Components

Business Process Functions

N/A

Business Flows

N/A

Business Rules

N/A

Tasks/Notifications

N/A

Screens

N/A

Reports

N/A

DSD 19/Payroll – Warrant Management/Business Process /External Interfaces/Warrant Paid – PRDR110A

CI	Document Name
CI-116410 - DSD EINTF Warrant Paid PRDR110A IMPLEMENTED	DSD_EINTF_Warrant_Paid_PRDR110A.doc

This is an interface between STO and CMIPS. STO will send to CMIPS a file with information about the warrants that were cleared. Based on this file, Warrant Paid interface will mark warrant records stored in the CMIPS system as Cleared.

This interface is part of the warrant processing cycle. If the interface is not executed, warrants will be created by SCO and SCO-specific warrant numbers will be displayed in CMIPS; however, information on whether the warrants have cleared will not be available.

Since warrants will be generated by both Advantage HRM (Providers) as well as Advantage Financial (vendors), it will be necessary to process the return file through both HRM and Financial.

Data Elements of Interface

Table – HRM Warrant Paid

Source:		Target: CHCK Document			
Table Name/ File Name/WSDL Name	Field Name	Field Name	Data Type (Format-length)	Description (Data Transformation rules)	R/O (Required or optional)
Warrant Paid File	SCO Warrant Number			Used in the lookup to Pay_Check. SCO_WARRANT_NO	R
Warrant Paid File	SCO Paid Date			N/A	R
Warrant Paid File	SCO Issue Date			Used in the lookup to PAY_CHECK. SCO_ISSUE_DT	R
Warrant Paid File	Amount			Used in the lookup to PAY_CHECK. PAY_CHK_AM	R
		CHK_STA		Set to "C" for Cleared	R

Table – FIN Warrant Paid

Source:		Target: R_AP_CHK_RECON			
Table Name/ File Name /WSDL Name	Field Name	Field Name	Data Type (Format-length)	Description (Data Transformation rules)	R/O (Required or optional)
Warrant Paid File	SCO Warrant Number	Advantage Warrant Number		Used in the lookup to R_AP_CHK_RECON. SCO_WARRANT_NO	R
Warrant Paid File	Paid Date	CLR_DT	Date	Set the CLR_DT on the R_AP_CHK_RECON table	R
Warrant Paid File	SCO Issue Date			N/A	R
Warrant Paid File	Amount			N/A	R
		CHK_STA	String	Set the CHK_STA to '3' for Cleared	R

Trigger Events

Warrant Paid jobs will be triggered by the script called by AutoSys. The jobs will be run daily.

Processing Criteria

Warrant Paid interface will process the file received from STO. This file will have SCO Warrant Number, Warrant Amount and SCO Issue Date information for each warrant cleared by STO. Interface will consist of two separate batch jobs:

- HRM Warrant Paid job will process the entire input file. It will do a lookup to Paycheck tables to see if there is a match based on the SCO Warrant Number, Warrant Amount and SCO Issue Date provided in the file. If there is a match, the job will retrieve the appropriate information to populate

a CHCK (Check Disposition) document and update it with the cleared status and cleared date. If there is not a match, the record will be written to a mismatch file which will be used by the FIN batch job.

- FIN Warrant Paid job will be run after HRM and will use the mismatch file created by the HRM batch job as its input. It will do a look up to the Check Reconciliation table in FIN to see if there is a match. The match is against the warrant number. If there is a match, the job will update the Check Reconciliation table with the cleared status as well as the cleared date from the file. If there is not a match, the job will write to a final exception file that will include the error or reason that it wasn't able to update.
- The HRM Batch job will also write to an output file those updates that processed successfully so that this file can be sent to CM. CM will receive this file on a nightly basis and process it so they will be in sync with what is in Payroll. The Warrant Status Code and Warrant Status Date fields will be passed to this file along with the employee ID and the Advantage Check Number.

Error Processing

Table – Error Codes

Error No.	Error Code	Description
1	WRNTPAID1001	SCO Warrant Number Not Provided
2	WRNTPAID1002	Warrant Amount Not Provided
3	WRNTPAID1003	Invalid Paid Date
4	WRNTPAID1004	No Match Found Writing to HRM Out
5	WRNTPAID1005	More Than One Record Returned in PAID Status
6	WRNTPAID1006	Record Returned Not in PAID Status
7	WRNTPAID1007	SQL Exception Retrieving FIN Results
8	WRNTPAID1008	CHCK Document Failed To Submit
9	WRNTPAID1009	Invalid Issue Date
10	WRNTPAID1010	Paid Date is before the Issue Date
11	GEN1006	General Exception

Related Components

Business Process Functions

N/A

Business Flows

N/A

Business Rules

N/A

Tasks/Notifications

N/A

Screens

N/A

Reports

N/A

DSD 19/Payroll – Warrant Management/Business Process /External Interfaces/Undeliverable Warrants – PRDR111A

CI	Document Name
CI-822207 - DSD EINTF Undeliverable Warrants PRDR111A IMPLEMENTED	DSD_EINTF_Undeliverable_Warrants_PRDR111A.doc

This is an interface between FI\$Cal and CMIPS. FI\$Cal will send to CMIPS warrant information for the warrants that were returned or cancelled and should be redeposited.

Upon receipt of the file, CMIPS will redeposit the warrants and update accounting entries appropriately.

Data Elements of Interface

Table – Warrant Information File to Payroll

Source: BPM		Target: File to Payroll			
Batch File	Field Name	Field Name	Data Type (Format-length)	Description (Data Transformation rules)	R/O (Required or optional)
Returned and Canceled Warrant File from BPM	WarrantNumber	WarrantNumber	String (20)	SCO Warrant number is 8 chars, however Payroll stores in 20 char column.	R
	PrintedWarrant Amount	PrintedWarrant Amount	Number (9,2)	Warrant Amt used for search criteria.	R
	ReturnCancelDate	ReturnCancelDate	Date (CCYY-MM-DD)	Used for logging errors in the exception file and passed to CM in the WarrantUpdate file	R
	CancelJulianDesc	CancelJulianDesc	String (6)	Values are: "CAN" = To be processed as Stale "RTN" = To be processed as Void "URD" = Undo Re-deposit – Records will be skipped "RPL" = Reissue/Replacement – Records will be skipped	R
	IssueDate	IssueDate	Date (CCYY-MM-DD)	Date warrant was issued by the SCO	R
	AgencyCode	AgencyCode	String (4)	Not used	O
	RANumber	RANumber	String (15)	Captured on the SCO_RA_NUMBER table	R
	ClaimSchedule Number	ClaimSchedule Number	String (6)	Not used	O
	PayeeID	PayeeID	String (9)	This field will be blank	O
	PayeeNameLine1	PayeeNameLine1	N/A	Not used	O
	PayeeAddress	PayeeAddress	N/A	Not used	O

Table - HRM CHCK Document

The following table describes the mapping that is used to form the CHCK document. Since it is indirect lookup into number of fields in order to get the CHCK document; it is best described in a separate table.

Source: BPM		Target: CHCK Document			
Batch File	Field Name	Field Name	Data Type (Format-length)	Description (Data Transformation rules)	R/O (Required or optional)
Returned and Canceled Warrant File from BPM	See the Description	CHK_NO	String (20)	Derive CHK_NO from PAY_CHECK table based on the following values from the BPM file: <ul style="list-style-type: none"> WarrantNumber (SCO Warrant Number in Advantage) IssueDate (SCO issue date in Advantage) PrintedWarrantAmount (Paycheck Amount in Advantage) 	R

		EMPLOYEE_ID	String (7)	Derive from PAY_CHECK table based on the following values from the BPM file: <ul style="list-style-type: none"> WarrantNumber (SCO Warrant Number in Advantage) IssueDate (SCO issue date in Advantage) PrintedWarrantAmount (Paycheck Amount in Advantage) 	R
	See the Description	APPOINTMENT_ID	String (1)	Derive APPOINTMENT_ID from PAY_CHECK table based on the following values from the BPM file: <ul style="list-style-type: none"> WarrantNumber (SCO Warrant Number in Advantage) IssueDate (SCO issue date in Advantage) PrintedWarrantAmount (Paycheck Amount in Advantage) 	R
		BANK_ACCOUNT_CD	String (4)	Default to "1000"	R
	CancelJulianDesc	CHK_STA	String (1)	Always set to "R" (Redeposit – Cancelled) (financial entries will not get updated correctly if the check status is set to "S")	R
		REPL_CHK	String (1)	Set to 'N' for Do Not Replace	R

Table – Financial Table Update

Source: Payroll (HRM)		Target: Payroll(FIN)			
Batch File	Field Name	Field Name	Data Type (Format-length)	Description (Data Transformation rules)	R/O (Required or optional)
Output File from HRM Undeliverable Warrant job	WarrantNumber	WarrantNumber	String (20)	Warrant Number used for search criteria	R
	PrintedWarrantAmount	PrintedWarrantAmount	Number(9,2)	Warrant Amt used for search criteria	R
	IssueDate	IssueDate	String (10)	IssueDate used for search criteria	R
	CancelJulianDesc	CancelJulianDesc	String (1)	If it's "R," set the warrant status to "7" (Cancel). If it's "T," set the warrant status to "5" (Stale).	R
	ReturnCancelDate	ReturnCancelDate	Date (CCYY-MM-DD)	Used for writing to the exception file in case of an error.	R

Table – SCO_RA_NUMBER

Source: Payroll (HRM)		Target: Payroll(FIN)			
Batch File	Field Name	Field Name	Data Type (Format-length)	Description (Data Transformation rules)	R/O (Required or optional)
PAY_CHECK	INT_CTRL_NO	INT_CTRL_NO	Number (15)		R
UndeliverableWarrantInput.xml	UndeliverableWarrantType.RANumber	RA_NUMBER	String (15)		R
Session	userId	LAST_UPDATE_USERID	String (15)		R
Database Trigger		LAST_UPDATE_DT	Timestamp		R

Trigger Events

Undeliverable Warrant jobs will be triggered by the script called by AutoSys. They will be run daily.

Processing Criteria

The Undeliverable Warrants interface will process the file received from FI\$Cal. The following items are received:

- RTN = Returned for Voided Checks

- CAN = Cancelled for Stale Dated Checks
- URD = Undo Re-deposit – Records will be skipped
- RPL = Reissue/Replacement – Records will be skipped

Each file will have SCO Warrant Number and Warrant Amount information for each warrant in the file. Undeliverable Warrant interface will consist of two separate batch jobs:

- HRM Undeliverable Warrants job will do a look up to Paycheck tables to see if there is a match based on the SCO Warrant Number, Warrant Amount and SCO Issue Date from the file. If there is a match, it will determine the Advantage Warrant Number based on the SCO Warrant Number and Warrant Amount. Then, it will retrieve the CHCK (Check Disposition) document based on Advantage Warrant Number, update this document with the new status (Void/Do Not Replace) as well as the Void or Stale Date, and process the document through SMU. If there is not a match, the record will be written to a mismatch file which will be used by the FIN batch job. If there are multiple matches the job will write to a final exception file that will include the error or a reason that it wasn't able to process
- FIN Undeliverable Warrants job will run after the HRM job and will use the mismatch file created by the HRM batch job as the input. It will do a look up on SCO Warrant Number, Warrant Amount, Issue Date and Payee ID to the Check Reconciliation table in FIN to see if there is a match. If there is a match, the job will find either a MD (Manual Disbursement) or AD (Auto Disbursement) document corresponding to the warrant. It will retrieve it and convert it to DC (Disbursement Cancellation) document. It will then retrieve the DC document, set the Void/Stale date information on it and submit it. If there is not a match, the job will write to a final exception file that will include the error or a reason that it wasn't able to process.
- This HRM batch job will also write to an output file those updates that were processed successfully so that this file can be sent to CM. CM will receive this file on a nightly basis and process it so they will be in sync with what is in Payroll. The Warrant Status Code and Warrant Status Date fields will be passed to this file along with the employee ID and the Advantage Check Number as specified in the interface mapping section below.

Updates that were not processed successfully will be written to an error log.

Error Processing

Table – Error Codes

Error No.	Error Code	Description
1	UNDELWAR1000	Warrant Number is not provided
2	UNDELWAR1001	Warrant Amount is not provided
3	UNDELWAR1002	Warrant Amount is not valid
4	UNDELWAR1003	Warrant Issue Date is not provided
5	UNDELWAR1004	Warrant Issue Date is not in the valid format
6	UNDELWAR1005	Warrant Issue Date is not valid
7	UNDELWAR1006	Warrant represents an EFT Payment
8	UNDELWAR1007	No longer used (CR 1175)
9	UNDELWAR1008	Warrant was not found using record parameters
10	UNDELWAR1009	Multiple warrants were found using record parameters
11	UNDELWAR1010	Invalid value in CancelJulianDesc field of the input record
12	UNDELWAR1011	Warrant Returned/Canceled Date is not provided
13	UNDELWAR1012	Warrant Returned/Canceled Date is not in the valid format
14	UNDELWAR1013	Warrant Returned/Canceled Date is not valid
15	UNDELWAR1014	Undeliverable Warrant SQL Exception
16	UNDELWAR1015	Error obtaining the Exception Output Writer
17	UNDELWAR1016	Error occurred while calling SMU to update PAY_CHECK.CHK_DISP_DT
18	UNDELWAR1017	Unable to find a DC document
19	UNDELWAR1018	DC Document Submission was unsuccessful
20	UNDELWAR1019	Unable to populate and submit a DC document
21	GEN1006	General Exception
22	UNDELWAR1020	Record being skipped due to URD status
23	UNDELWAR1021	Record being skipped due to RPL status

NOTE: Records with a CancelJulianDesc of "URD" or "RPL" will be skipped and will not be included in error counts.

Related Components

Business Process Functions

N/A

Business Flows

N/A

Business Rules

N/A

Tasks/Notifications

N/A

Screens

N/A

Reports

N/A

DSD 19/Payroll – Warrant Management/Business Process /External Interfaces/Vendor Listing to SCO – PRMS109I

CI	Document Name
CI-706093 - DSD EINTF Vendor Listing to SCO PRMS109I IMPLEMENTED	DSD_EINTF_Vendor_Listing_to_SCO_PRMS109I.docx

Each month CMIPS will deliver a file to SCO that will provide information on all current CMIPS Vendors.

Internal Layout/System Mapping

Table - FIN Vendor Listing to SCO

Source: Payroll			Target: File to BPM		
Table Name/ File Name/ WSDL Name	Field Name	Field Name	Data Type (Format-length)	Description(Data Transformation rules)	R/O (Required or optional)
		<?xml version="1.0" encoding="UTF-8"?><SCOVendorListing xmlns="http://www.cgi.com/Adv/HR/SCO_output">	N/A	XML Header	R
		<SCOVendorInfo>	N/A	Transaction Start	R
R_VEND_CUST	Lgl_Nm	vendorName	String (30)	Vendor Name	R
R_VEND_CUST	Vend_Cust_Cd	vendorId	String (12)	Vendor ID	R
R_VEND_CUST	Tin	vendorTaxId	String (9)	Vendor Tax ID (TIN)	O
R_VEND_CNTAC	Str_1_Nm	addressLine1	String (30)	Address Line 1	R
R_VEND_CNTAC	Str_2_Nm	addressLine2	String (30)	Address Line 2	O
R_VEND_CNTAC	City_Nm	city	String (28)	City	R
R_VEND_CNTAC	St	state	String (2)	State	R
R_VEND_CNTAC	Zip	zip	String (10)	Zip-Zip + 4	R Minimum 5 characters
		</SCOVendorInfo>	N/A	Transaction End	R
		</SCOVendorListing>	N/A	File End	R

Trigger Events

The vendor listing job will be triggered by the script called by AutoSys. It will be run monthly.

Processing Criteria

There will be one file created and sent to BPM containing current vendors.

Error Processing

Error No.	Error Code	Description
1	UTLS1002	Batch Processing Error
2	GEN1006	General Exception Occurred

Related Components

Business Process Functions

N/A

Business Flows

N/A

Business Rules

N/A

Tasks/Notifications

N/A

Screens

N/A



















Reports













N/A

DSD 19/Payroll – Warrant Management/Business Process /Batch Processing

This section will describe the identified CMIPS batch processes.



Table – Batch Processing

No.	Batch Process Name	CI	Description	Estimated Size (Records)	Frequency	SendReceiveMaintenance
1	County Contractor and Homemaker Journal Voucher (10010)	 CI-116249 - DSD BTCH PRO ADV p f int 10010 CountyContractor 	Loads the County Contractor and Homemaker Journal Voucher from CM to Financial. Each of the contractor counties will send billings once or twice per month. When a file is processed by Case Management, the data will be loaded to payroll.	100-5,000	Daily	Receive
2	Sick Leave Accrual from CM (1210066)	 CI-790582 - DSD BTCH PRO ADV p h int 1210066 SickLeaveAccrual 	Processes Sick Leave Accrual balances sent from CM.	N/A	Daily	Receive
3	Update Sick Leave Accrual (1210067)	 CI-790583 - DSD BTCH PRO ADV p h npc 1210067 UpdateSickLeaveAccrual 	Updates Sick Leave Accrual balances for payments that were redeposited as part of the nightly cycle.	N/A	Daily	N/A
4	HRM - Warrant Assignment (10036) - paper	 CI-291377 - DSD BTCH PRO ADV p h int 10036 hrmwarrantassignpaper 	Captures the SCO Warrant Number on the file for the matching warrant number in HRM and produces an output file to pass the information to Case Management.	5,000 - 160,000	Daily	Receive
5	FIN - Warrant Assignment (10014) - paper	 CI-291384 - DSD BTCH PRO ADV p f int 10014 finwarrantassignpaper 	Captures the SCO Warrant Number on the file for the matching warrant number in FIN.	1,000	Daily	Receive
6	HRM - Warrant Assignment (10036) - EFT	 CI-818544 - DSD BTCH PRO ADV p h int 10036 hrmwarrantassigneft 	Captures the SCO Warrant Number on the file for the matching warrant number in HRM and produces an output file to pass the information to Case Management.	5,000 - 160,000	Daily	Receive
7	FIN - Warrant Assignment (10014) - EFT	 CI-818545 - DSD BTCH PRO ADV p f int 10014 finwarrantassigneft 	Captures the SCO Warrant Number on the file for the matching warrant number in FIN.	1,000	Daily	Receive
8		 CI-116328 - CMIPS 2 BTCH pro adv p h int 10044 undelivwarrant 	Cancelled by CR1175 – PSR – INT – Undeliverable Warrant File from FISCal			
9	HRM - Undeliverable and Stale Warrants (10044)	 CI-822209 - DSD BTCH PRO ADV p h int 10044 undelivwarrant 	Sets the warrant status to Redeposited for records in HRM that match records in the Undeliverable Warrant file, produces an output file to pass the information to Case Management. It also produces an output file of records not found, which is consumed by the FIN - Undeliverable and Stale Warrants (10071) batch job.	1,000	Daily	Receive

10	FIN - Undeliverable and State Warrants (10071)	 CI-116315 - DSD BTCH PRO ADV p f int 10071 undelivwarrant IMPLEMENTED	Sets the warrant status to Redeposited for records in FIN that match records in the Undeliverable Warrant file.	1,000	Daily	Receive
11	Warrant Paid (HRM) (10045)	 CI-116332 - DSD BTCH PRO ADV p h int 10045 hrmwarrantpaid IMPLEMENTED	Sets the warrant status to Cleared for records in HRM that match records in the Warrant Paid file and produces an output file to pass the information to Case Management	5,000 - 160,000	Daily	Receive
12	Warrant Paid (FIN) (10015)	 CI-291383 - DSD BTCH PRO ADV p f int 10015 finwarrantpaid IMPLEMENTED	Sets warrant status to Cleared for records in FIN that match records in the Warrant Paid file.	1,000	Daily	Receive
13	FIN - Warrant Status Change to CM (10080)	 CI-291365 - DSD BTCH PRO ADV p f int 10080 warstatchgtocm IMPLEMENTED	Generates an output file containing warrant status changes initiated by Back Office staff. This job was implemented to process converted warrants during transition to CMIPS II.	0	Daily	N/A
14	HRM Warrant Status Change to CM (10067)	 CI-291367 - DSD BTCH PRO ADV p h out 10067 warrantstatuscm IMPLEMENTED	Generates an output file containing warrant status changes initiated by Back Office staff.	Less than 100	Daily	N/A
15	Ad Hoc Undeliverable Warrant (1210053)	 CI-465058 - DSD BTCH PRO ADV adhoc undeliv warrant IMPLEMENTED	Manually run to Consume an xml file manually by Back Office staff to void warrants based upon direction from SCO. This makes it possible to associate the RA Number with the warrant being voided.	Less than 20	Ad Hoc	Maintenance
16	Warrant Reissue - HRM (10037)	 CI-291376 - DSD BTCH PRO ADV p h int 10037 warrantreissue IMPLEMENTED	Consumes the prdr917a batch job from Case Management, and creates CHCK documents to redeposit and replace paper warrants.	less than 100	Daily	Receive
17	Warrant File Merge (10049)	 CI-116316 - DSD BTCH PRO ADV p h out 10049 warrantfilemerge IMPLEMENTED	Merges the output files from Warrant Assignment, Warrant Paid, Undeliverable Warrant, Warrant Reissue, Warrant Status Change Advance Pay Recon Status into an updateWarrant xml file for the Case Management cmdr921a batch job to consume.	20-160,000	Daily	Send
18	Warrant Information Table Create (10033)	 CI-116333 - DSD BTCH PRO ADV p h out 10033 cmwarrant IMPLEMENTED	Generates the CM Warrant file to be consumed by the Case Management cmdr920a job. The CM Warrant file will display the correct pay rate for single and multiple transactions based on the transactions corresponding rules.	5,000 - 160,000	Daily	Send
19	Daily Claim to BPS - HRM (10034)	 CI-116313 - DSD BTCH PRO ADV p h out 10034 scowarrant IMPLEMENTED	Identifies and loads all components related to the pay type and generates the Daily Payment file to be sent to SCO out of HRM	5,000 - 160,000	Daily	N/A
20	Create Lien Holder Gax Document (1210060)	 CI-717499 - DSD BTCH PRO ADV 1210060 f pvp CreateLienHolderGaxDocument IMPLEMENTED	Creates the General Accounting Transaction (GAX) documents to generate lien holder vendor warrants.	Less than 100	Daily	N/A
21	Journal Posting Initiator (7)	 CI-116273 - DSD BTCH PRO ADV p f pvp 7 journalpostinginitiator IMPLEMENTED	Sets up the postings for the journal from documents that have been entered into the system.	N/A	Twice in each cycle	N/A

22	Journalization Engine (5)	CI-116274 - DSD BTCH PRO ADV p f pvp 86 journalengine IMPLEMENTED	Posts the documents to the journal in the financial system.	N/A	Twice in each cycle	N/A
23	Ledgerization Engine (85)	CI-116278 - DSD BTCH PRO ADV p f fyc 85 ledgerengine IMPLEMENTED	Posts the documents to the ledger in the financial system.	N/A	Twice in each cycle	N/A
24	Check Reconciliation (101)	CI-116248 - DSD BTCH PRO ADV p f int 101 checkreconchain IMPLEMENTED	Used to reconcile the warrants with the paid and updated status.	N/A	Daily	N/A
25	Pre-note Batch Process (220)	CI-116302 - DSD BTCH PRO ADV p f fri 220 prenoteachtrans IMPLEMENTED	Used to process the vendor pre-note files.	20-200	Daily	Maintenance
26	Disbursement Parameters Update (1035)	CI-116256 - DSD BTCH PRO ADV 1035 disbursementparamupdates IMPLEMENTED	Used to establish the processing business rules for setting the disbursement parameters to be used for the vendor payments. This is set for both checks and EFT payments.	2	Daily	N/A
27	FIN - EFT Pre-note (10013)	CI-116314 - DSD BTCH PRO ADV p f fri 10013 eftprenote IMPLEMENTED	Used to create the SCO pre-note file for the SCO.	20-200	Weekly	Send
28	Automated Disbursement Chain (133)	CI-116239 - DSD BTCH PRO ADV p f pvp 133 adchain IMPLEMENTED	Produces the vendor warrants and EFT transactions.	20-5,000	Daily	N/A
29	Disbursement Printing (21)	CI-116257 - DSD BTCH PRO ADV p f pvp 21 disbursementprinting IMPLEMENTED	Creates the .dat file for all warrants processed in the financial system.	20-5,000	Daily	N/A
30	EF ACH Transaction (171)	CI-116260 - DSD BTCH PRO ADV p f pvp 171 efachtransaction IMPLEMENTED	Creates the EFT warrants.	20-5,000	Daily	N/A
31	FIN - CTX EDI Formatting (1458)	CI-816462 - DSD BTCH PRO ADV p f pvp 1458 ctxediformatting IMPLEMENTED	Generates the CTX Addenda Records based on the EDI ANSI ASC X12 820 Remittance Advice /Payment Order. This batch process is only applicable to EFT documents with the Disbursement Format of CTX.	4		N/A
32	Daily Claims to SCO FIN EFT (10011)	CI-291386 - DSD BTCH PRO ADV p f out 10011 scofineft IMPLEMENTED	Creates the daily payment file for SCO for all vendor EFT warrants.	4	Daily	Send
33	Daily Claims to SCO FIN Non-EFT (10012)	CI-291385 - DSD BTCH PRO ADV p f out 10012 scofinnoneft IMPLEMENTED	Creates the daily payment file for SCO for all vendor paper warrants.	1 - 5,000	Daily	Send
34	XML File Sort Batch Job (SCO) (1210051)	CI-465042 - DSD BTCH PRO ADV p h out 1210051 scowarrantsort IMPLEMENTED	Merges the HRM output files produced by the daily claim process and sorts the records by zip code.	20,000 - 140,000	Daily	Send

35	SCO Vendor Listing (1210059)	CI-706094 - DSD BTCH PRO ADV p f pvp 1210069 Vendor Listing to SCO IMPLEMENTED	Creates the list of current CMIPS vendors for SCO.	Less than 300	Daily	Send
36	Payroll Report ETL Load	CI-116308 - DSD BTCH PRO ADV p n out na etlloads IMPLEMENTED	Loads the financial data for reporting into the reporting database.	100,000 - 4,000,000	Daily	N/A
37	Systems Assurance 01 (1150) – Systems Assurance 1 Pre-Processor, Systems Assurance 1, SA01 Report	CI-116323 - DSD BTCH PRO ADV p f pvp 1150 systemassurance01 IMPLEMENTED	System assurance jobs that report any out of balance conditions that need research.	N/A	Daily	N/A
38	Systems Assurance 03 (256) - Budget Actuals Versus Ledgers, Journals and Posting Line Catalog (SA3, SA11)	CI-116241 - DSD BTCH PRO ADV p f pvp 256 systemassurance03 IMPLEMENTED	System assurance jobs that report any out of balance conditions that need research.	N/A	Daily	N/A
39	Catalog Indexing	CI-116244 - DSD BTCH Catalog Indexing IMPLEMENTED	Indexes the system tables from the daily processing.	N/A	Daily	N/A
40	NYTI Process (430)	CI-291334 - DSD BTCH PRO ADV p f xxx 430 nytiprocess IMPLEMENTED	Carries forward the existing Chart of Accounts configuration for the next two fiscal years.	N/A	Annual	Maintenance
41	Update Next Fiscal Year (10091)	CI-291361 - DSD BTCH PRO ADV p h fdi 10091 updatenextfy IMPLEMENTED	Sets up the format for automatic document numbering for each Document Code and Department combination for the next two fiscal years.	4,200	Annual	Maintenance
42		CI-116255 - CMIPS 2 BTCH pro adv Daily Payment File to SCO CANCELLED	Cancelled by SCF - 30: DSD - MAS Batch Process CI Traceability Updates			
43		CI-121482 - DSD BTCH Begin Day FIN CANCELLED	Cancelled by SCF - 30: DSD - MAS Batch Process CI Traceability Updates			
44	EFT Pre-Note – PRWS109B	CI-291587 - DSD BTCH PRO BAW prws109b IMPLEMENTED	This batch job extracts EFT data from CMIPS tables, format the data and send to SCO.	Variable	Weekly	Send
45	Warrant Assignment EFT – PRDR109H	CI-291613 - DSD BTCH PRO BAW prdr109h IMPLEMENTED	This batch job receives SCO-specific control numbers and issue dates for the EFT payments that were generated by SCO.	Variable	Daily	Receive
46	Warrant Assignment Non-EFT – PRDR109D	CI-291622 - DSD BTCH PRO BAW prdr109d IMPLEMENTED	This batch job receives SCO-specific control numbers and issue dates for the non-EFT payments that were generated by SCO.	Variable	Daily	Receive
47	Warrant Paid – PRDR110A	CI-291632 - DSD BTCH PRO BAW prdr110a IMPLEMENTED	Batch job receives a file from STO with the list of paper warrants that were cleared (EFT payments are not included). Based on this file, the Warrant Paid process will update the status of warrant records stored in the Advantage system to "Cleared."	Variable	Daily	Receive

48	Send CMIPS Vendor Info to SCO -PRMS109I	 CI-816710 - DSD BTCH PRO BAW p prms109i 	Delivers a file to SCO that provides information on all current CMIPS Vendors	N/A	Monthly	Send
----	---	---	--	-----	---------	------

DSD 19/Payroll – Warrant Management/Business Process /Applicable Security Roles

This section is not applicable for Warrant Management.

DSD 19/Payroll – Warrant Management/Business Process /Reporting

All payroll related reports are located in DSD Section 28C or 28D.

DSD 19/Payroll – Warrant Management/Business Process /Forms

See DSD Section 31A, 31B or 31C for all CMIPS Forms.