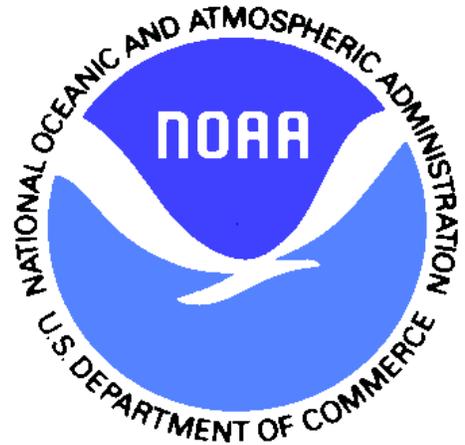


CAMS Project



NOAA/CAMS CFS Implementation
Prepare FIMA ACTDOC for Open
Document Conversion Program Module
(FINAL)

Version 2.0

Prepared by Accenture for

U.S. Department of Commerce
National Oceanic and Atmospheric Administration
Implementation Center
201 Perry Parkway
Gaithersburg, MD 20877

February 01, 2002

 Document Control
Change Record

Date	Author	Document Version	Change Reference
12/19/2002	K. Stockmeier	1.0	Draft
01/14/2002	K. Stockmeier	1.2	Draft
01/23/2002	K. Stockmeier	1.3	Draft
01/24/2002	K. Stockmeier	1.4	Draft
02/01/2002	K. Stockmeier	2.0	Final

Reviewer

Name	Position	Date
K. Kirwan	Manager	01/08/2002
M. Barron	Manager	01/10/2002

Approver

Name	Position	Date
W. Holdsworth	Team Lead	01/11/2002

Table of Contents

1	<u>Introduction Prepare FIMA ACTDOC For Open Document Conversion</u>	1
1.1	Naming Standards	1
1.2	Purpose of Prepare FIMA ACTDOC For Open Document Conversion . .	2
1.3	Process Flow	2
1.4	Business Rules	3
1.5	Input to Module	3
2	<u>Conversion Table Setup (Function 1)</u>	3
2.1	Purpose of Conversion Table Setup	3
2.2	Process Flow	4
2.3	Input to Conversion Table Setup	5
2.4	Conversion Table Setup Processing	5
	2.4.1 Create CONV_ACTDOC and CONV_ACTDOC_ACCOUNT Tables (Sub-function 1)	5
	2.4.1.1 Table Layouts	5
	2.4.1.2 Logic	15
	2.4.1.3 Output	15
	2.4.2 Populate CONV_ACTDOC and CONV_ACTDOC_ACCOUNT from <i>FIMA ACTDOC</i> (Sub-function 2)	15
	2.4.2.1 Input	15
	2.4.2.2 Logic	16
	2.4.2.3 Output	17
	2.4.3 Record Earliest Fiscal Year on each Document (Sub-Function 3) .	17
	2.4.3.1 Input	17
	2.4.3.2 Logic	17
	2.4.3.3 Output	17
	2.4.4 Record Minimum Open Fiscal Year on Each Document (Sub- Function 4)	18
	2.4.4.1 Input	18
	2.4.4.2 Logic	18
	2.4.4.3 Output	18
	2.5.5 Create CONV_ERRORS table (Sub-Function 5)	18
	2.5.5.1 Table Layout	19
	2.5.5.2 Logic	20
	2.5.5.3 Output	20
3	<u>Convert and Validate Accounting Codes (Function 2)</u>	20
3.1	Purpose of Convert and Validate Accounting Codes	20
3.2	Process Flow	20

3.3	Input	22
3.4	Convert and Validate Accounting Codes Processing	22
3.4.1	Convert FIMA Accounting Codes (Sub-Function 1)	22
3.4.1.1	Process Flow for Convert FIMA Accounting Codes	23
3.4.1.2	Input	23
3.4.1.3	Logic	24
3.4.1.4	Output	24
3.4.2	Validate Converted CFS Accounting Codes	24
3.4.2.1	Validate Converted CFS Accounting Codes Sub- Process Flow	24
3.4.2.2	Input	25
3.4.2.3	Logic	25
3.4.2.4	Output	26
3.5	Validate Document Bureau Codes	26
3.5.1	Validate Bureau Codes	26
3.5.2	Input	26
3.5.3	Logic	26
3.5.4	Output	27
4	<u>Overall Issues</u>	27
5	<u>References</u>	27

1 Introduction Prepare FIMA ACTDOC For Open Document Conversion

This program copies data from the FIMA **ACTDOC** table and inserts into conversion tables for the open document conversion process. The purpose of this document is to provide the detail-level design for the development of this process. This module:

- ▶ Creates the Convert ACTDOC table (**CONV_ACTDOC**).
- ▶ Creates the Convert ACTDOC Account table (**CONV_ACTDOC_ACCOUNT**).
- ▶ Creates the Convert Runtime table (**CONV_RUNTIME**) to keep track of the amount of time that each module runs in the conversion process.
- ▶ Populates the conversion tables from the FIMA **ACTDOC** table.
- ▶ Records the minimum fiscal year on each document.
- ▶ Records the earliest open fiscal year on each document.
- ▶ Converts the FIMA accounting detail.
- ▶ Validates converted values against CFS Accounting Code Classification Structure (ACCS) set-up.
- ▶ Creates the Convert Errors table (**CONV_ERRORS**) to contain a record and description of all the errors found during the conversion and validation processes.

1.1 Naming Standards

The Prepare FIMA ACTDOC For Open Document Conversion Program Module adheres to the latest version of the CAMS Support Center (CSC) Programming Standards and Guidelines.

Throughout this document suggested names of objects are consistently formatted in order to make it easier to distinguish what type of object is being referred to in the design. The following table is a list of each object in this document and its corresponding format.

<u>Object</u>	<u>Description</u>
SCREENS	Named in all caps and italics
<i>Field Names</i>	Named in italics with the first letter of each word capitalized
TABLES	Named in all caps and bold
Column Names	Named in bold with the first letter of each word capitalized
<u>FILES</u>	Named in all caps and underlined

variables

Named in all lowercase and underlined

1.2 Purpose of Prepare FIMA ACTDOC For Open Document Conversion

The purpose of this document is to provide the detail-level design to create and populate the Convert ACTDOC (**CONV_ACTDOC**), Convert ACTDOC Account (**CONV_ACTDOC_ACCOUNT**), Convert Runtime (**CONV_RUNTIME**), and Convert Errors (**CONV_ERRORS**) tables.

1.3 Process Flow

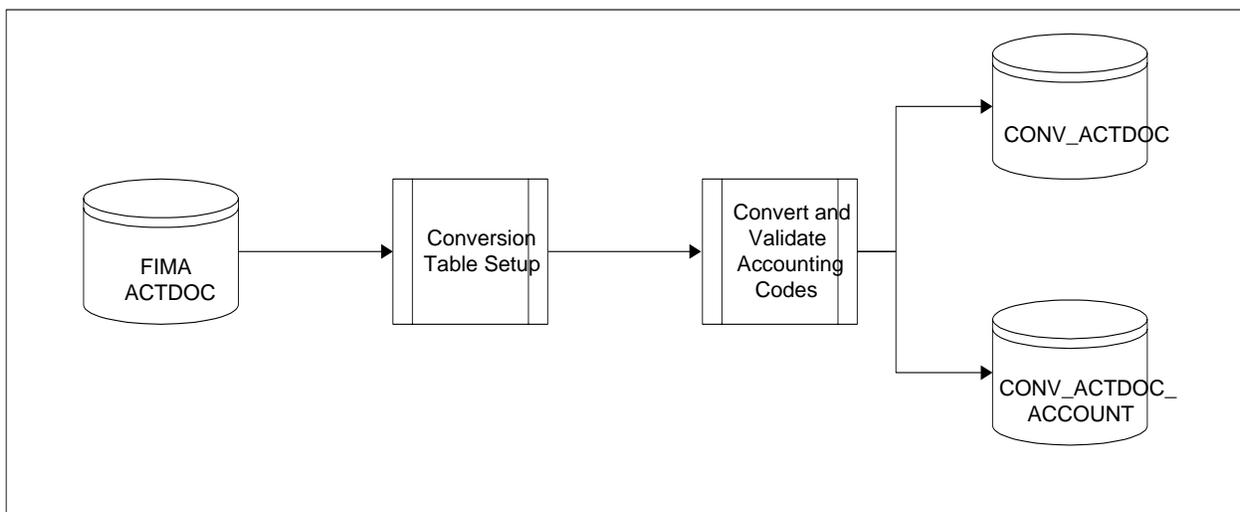


Figure 1. Process Flow for Prepare FIMA ACTDOC for Open Document Conversion

Figure 1 represents the overall process flow for the module and contains the following components:

FIMA ACTDOC. Data from this table is copied into two conversion tables that are created during the *Conversion Table Setup* Function.

Conversion Table Setup. Creates the Convert ACTDOC and Convert ACTDOC Account tables.

Convert and Validate Accounting Codes. After the conversion tables are created and data from the FIMA **ACTDOC** table has been copied to them, the accounting codes are converted and validated.

Convert ACTDOC. The **Convert ACTDOC Table** contains the summary information for each document.

Convert ACTDOC ACCOUNT. The **Convert ACTDOC Account Table** contains the detail information for each accounting line in a document.

1.4 Business Rules

- ▶ Only open documents from FIMA **ACTDOC** will be converted from FIMA to CFS.
- ▶ The minimum fiscal year of a document is the earliest fiscal year that appears on the document. (Note: The fiscal year could be closed.)
- ▶ The minimum open fiscal year of a document is the earliest fiscal year on the document (for a specific FIMA document number) where the document is "Open" (i.e. the **Completed Document** field has a NULL value).

1.5 Input to Module

The inputs for this module are:

- ▶ All open documents on the FIMA **ACTDOC** table for FIMA **Document Types** '01' to '49'.
- ▶ Data from CFS Production will be used to validate accounting line information attached to the FIMA open documents.

2 Conversion Table Setup (Function 1)

2.1 Purpose of Conversion Table Setup

The Conversion Table Setup function creates and populates the **Convert ACTDOC** and **Convert ACTDOC Account** tables. This process also includes populating four columns on the Convert ACTDOC table: the Minimum Fiscal Year (**Min_Fiscal_Year**), the Minimum Open Fiscal Year (**Min_Open_Fiscal_Year**), the FIMA Undelivered Orders Document Total (**FIMA_UDO_Document_Total**), and the FIMA Unpaid Accrued Expenditures Document Total (**FIMA_UAE_Document_Total**).

2.2 Process Flow

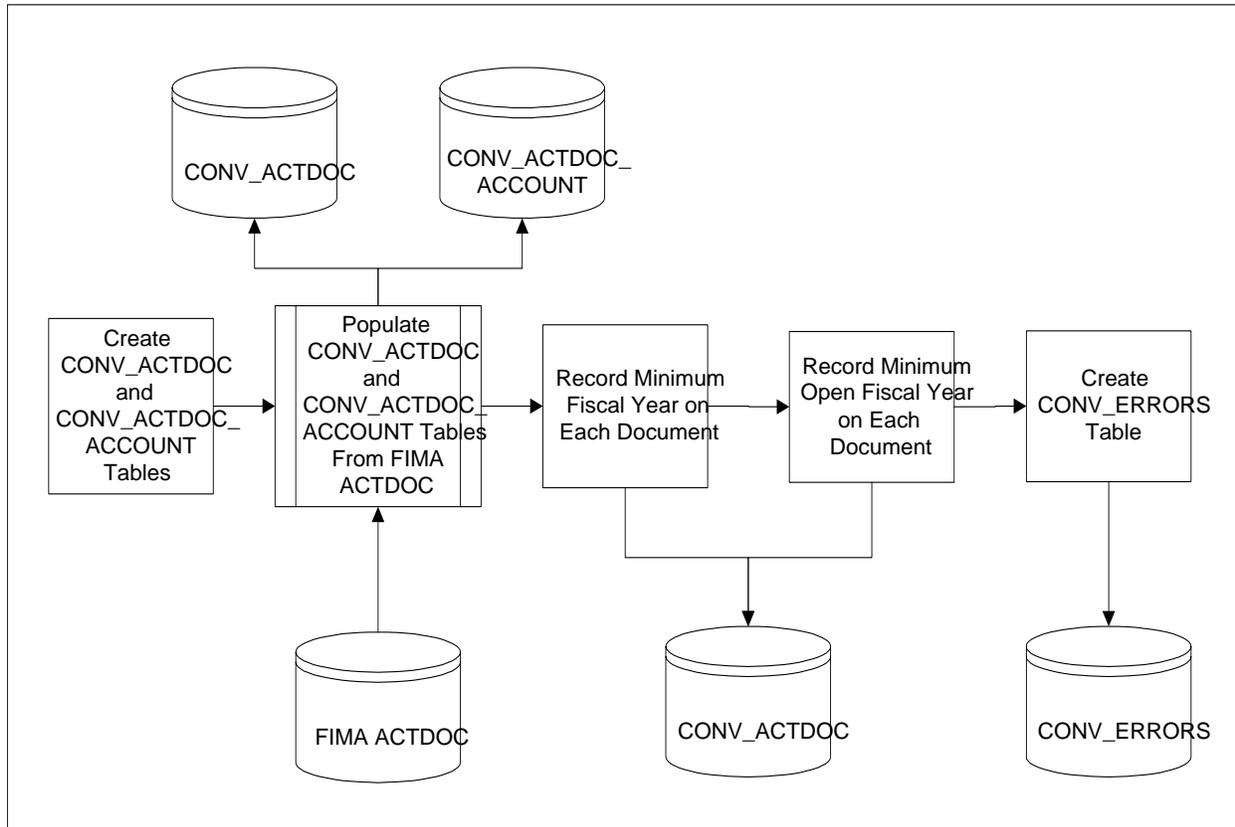


Figure 2. Process Flow for Conversion Table Setup (Function 1)

Figure 2 represents the process flow for the Conversion Table Setup Function and contains the following sub-functions:

Create CONV_ACTDOC and CONV_ACTDOC_ACCOUNT Tables. Conversion tables are created at this point in the process.

Populate CONV_ACTDOC and CONV_ACTDOC_ACCOUNT Tables from FIMA ACTDOC. The conversion tables will be populated with open document information from the FIMA **ACTDOC** table.

Record Minimum Fiscal Year on Each Document. The Minimum Fiscal Year is selected from FIMA **ACTDOC** and recorded in the **Convert ACTDOC Table** for each document.

Record Minimum Open Fiscal Year on Each Document. The Minimum Open Fiscal Year is selected from FIMA **ACTDOC**, and recorded in the **Convert ACTDOC Table** for each document.

Create CONV_ERRORS Table. This table stores errors that are found during the conversion process.

*Note: The creation of the Convert Runtime Table (**CONV_RUNTIME**), and the population of the FIMA Undelivered Orders Document Total (**FIMA_UDO_Document_Total**) and the FIMA Unpaid Accrued Expenditures Document Total (**FIMA_UAE_Document_Total**) are not represented in this diagram, but are still included in the processing of the Program Module.

2.3 Input to Conversion Table Setup

- ▶ All open documents on the FIMA **ACTDOC** table for FIMA **Document Types** '01' to '49'.

2.4 Conversion Table Setup Processing

2.4.1 Create **CONV_ACTDOC** and **CONV_ACTDOC_ACCOUNT** Tables (Sub-function 1)

- This function creates the Convert ACTDOC (**CONV_ACTDOC**), Convert ACTDOC Account (**CONV_ACTDOC_ACCOUNT**), and Convert Runtime (**CONV_RUNTIME**) tables in the CFS environment. These tables hold information relevant to the conversion process, including fields copied from the FIMA **ACTDOC** table.

2.4.1.1 Table Layouts

The tables created by this function are listed below in Table 1. Tables 2, 3, and 4 provide the table descriptions.

Table 1. Conversion Tables to be Created	
Table Name	Table Description
CONV_ACTDOC	A conversion table which stores a summary of all the accounts for FIMA ACTDOC open documents.
CONV_ACTDOC_ACCOUNT	A conversion table which stores detailed accounting code information for each open document to be converted.
CONV_RUNTIME	A conversion table which records the start and stop time of each module in the conversion process.

* Indicates field pulled from FIMA **ACTDOC**

Table 2. CONV_ACTDOC Table Description		
Column Name	Data Type	Description
* Document_Type	VARCHAR2(2)	Stores the FIMA document type.
* Document_Number	VARCHAR2(9)	Stores the FIMA document number.
Vendor_No	NUMBER(10)	CFS Vendor Number associated with this document.
Purchase_Vendor_Id	NUMBER(6)	CFS Purchase Vendor Address ID associated with this document.
Payment_Vendor_Id	NUMBER(6)	CFS Payment Vendor Address ID associated with this document.
Vendor_TIN	VARCHAR(9)	NPS 2000 vendor identification data for Non-Federal documents.
Min_Fiscal_Year	NUMBER(4)	The earliest fiscal year that appears on a document.
Min_Open_Fiscal_Year	NUMBER(4)	The earliest open fiscal year on this document.

Table 2. CONV_ACTDOC Table Description		
Column Name	Data Type	Description
Cfs_Document_No	NUMBER(8)	Updated in the Undelivered Orders process. Refer to the Convert Non-CFS Undelivered Orders and Accruals Program Module and the Compare FIMA ACTDOC for Open Document Conversion Program Module.
Cfs_Unrecon_Document_No	VARCHAR(30)	Updated in the reconciliation process. Refer to the Compare FIMA ACTDOC and CFS Production Program Module.
Conv_Date	DATE	Date that the FIMA document was successfully converted into CFS.
Cfs_Ea_Control_Id	NUMBER(8)	Updated in the Unpaid Accrued Expenditures process. Refer to Convert Non-CFS Undelivered Orders and Accruals Program Module.
Conv_Status_Flag	NUMBER(1)	Indicates whether or not the document has been successfully converted. Default value is 'N'.
Cfs_Reconcile_Flag	VARCHAR2(1)	Indicates whether or not the document has been reconciled. Default value is 'N'.
Cfs_Match_Flag	VARCHAR2(1)	Indicates whether the document is CFS or Non-CFS; set in the Compare FIMA ACTDOC and CFS Production Program Module. Default value is 'N'.
Error_Flag	VARCHAR2(1)	Indicates whether or not an error has been found in the document or any of its records. Default value is set to 'N'.
Vendor_Error_Flag	VARCHAR2(1)	Indicates whether or not there is a document error related to the vendor. Default value is 'N'.
Cfs_Document_Type_Error	VARCHAR2(1)	Indicates whether or not there is a document error because of an invalid document type.

Table 2. CONV_ACTDOC Table Description		
Column Name	Data Type	Description
FIMA_UDO_Document_Total	NUMBER(13,2)	Contains the combined sum of the Undelivered Orders column for all the open lines in a document copied from FIMA.
FIMA_UAE_Document_Total	NUMBER(13,2)	Contains the combined sum of the Unpaid Accrued Expenditures column for all the open lines in a document copied from FIMA.
CFS_UDO_Document_Total	NUMBER(13,2)	Contains the combined sum of the Undelivered Orders column for all the open lines in a document. Populated by the Compare FIMA ACTDOC and CFS Production Program Module.
CFS_UAE_Document_Total	NUMBER(13,2)	Contains the combined sum of the Unpaid Accrued Expenditures column for all the open lines in a document. Populated by the Compare FIMA ACTDOC and CFS Production Program Module.
CFS_UDO_Document_TRIAL_Total	NUMBER(13,2)	Contains the combined sum of the Undelivered Orders column for all the open lines in a document for TRIAL entries. Populated by the Compare FIMA ACTDOC and CFS Production Program Module.
CFS_UAE_Document_TRIAL_Total	NUMBER(13,2)	Contains the combined sum of the Unpaid Accrued Expenditures column for all the open lines in a document for TRIAL entries. Populated by the Compare FIMA ACTDOC to CFS Production Program Module.
User_Name	VARCHAR2(30)	DB_User_Name of the last person that updated the table.
Modification_Date	DATE	Date and time that the conversion table was last modified. This field will contain the format: 'MM/DD/YYYY HH:MM:SS'.

Table 2. CONV_ACTDOC Table Description		
Column Name	Data Type	Description
Device_Name	VARCHAR2(30)	Name of the program running the insert/updates to the table.

Table 3. CONV_ACTDOC_ACCOUNT Table Description		
Column Name	Data Type	Description
*Document_Type	VARCHAR2(2)	Stores the FIMA document type.
*Document_Number	VARCHAR2(9)	Stores the FIMA document number.
*Document_Sequence	NUMBER(4)	Sequence number of this record. Indicates chronological sequence of processing transaction.
*Secondary_Type	VARCHAR2(2)	FIMA Secondary document type.
*Secondary_Number	VARCHAR2(9)	FIMA Secondary document number, used as a reference.
*Fiscal_Year	NUMBER(4)	Identifies the fiscal year in FIMA.
*Terminal_Code	NUMBER(2)	FIMA Terminal code identifying origin of entry by specific ASC.
*Financial_Management_Ctr	NUMBER(3)	FIMA Financial Management Center Code. See FMC table for corresponding FMC titles.
*Organization_Code	VARCHAR2(6)	FIMA Code representing NOAA organization elements. Refer to NOAORG table for corresponding Organization titles.
*Appropriation	VARCHAR2(1)	NOAA appropriation symbols in FIMA. Refer to APN table for corresponding titles.
*Task_Number	VARCHAR2(6)	Predefined six character code in FIMA identifying individual NOAA tasks. Refer to TASK table for information on individual tasks.

Table 3. CONV_ACTDOC_ACCOUNT Table Description		
Column Name	Data Type	Description
*Phase_Number	VARCHAR2(2)	User defined sub-element of TASK in FIMA that identifies individual stages.
*Major_Object_Class	NUMBER(2)	First two digits of object class, identifying major object classification in FIMA.
*Minor_Object_Class	NUMBER(2)	Second two digits of object class, identifying minor object classification in FIMA.
*Govt_Code	VARCHAR2(1)	FIMA Government code. Used to indicate status of disbursee.
*Country_Code	VARCHAR2(3)	FIMA three character country of or identifying country of origin, or in the case of prompt payment or lost discount penalties, used to indicate reason for late payment penalty.
*Undelivered_Orders	NUMBER(13,3)	FIMA amount in dollars and cents of goods and/or services ordered but not yet received or paid for.
*Unpaid_Accrued_Exp	NUMBER(13,3)	FIMA amount in dollars and cents of goods and/or services received and/or accrued on estimated basis for which payment has not been made or recorded.
*Accounts_Receivable	NUMBER(13,3)	Dollars and cents amount owed to NOAA for which payment has not been received or recorded in FIMA.
*Paid_Accrued_Exp	NUMBER(13,3)	Amount of payments made and recorded for goods and/or services received in FIMA.
*Receipts	NUMBER(13,3)	Value of an amount in dollars and cents of credit or payment of account payable in FIMA.
*Completed_Document	VARCHAR2(1)	Indicates if the document is considered completed/closed (C) or active/open, if blank in FIMA.

Table 3. CONV_ACTDOC_ACCOUNT Table Description		
Column Name	Data Type	Description
*Creation_Date	DATE	Date this record was created in the FIMA system.
*Active_Date	DATE	Date last activity, or transaction, was recorded against this record in FIMA.
Bureau_Code	NUMBER(2)	The treasury identification code for each bureau.
Fund_Code	NUMBER(2)	Populated from the Convert and Validate Accounting Codes function in this module.
Org1_Code	VARCHAR2(2)	Populated from the Convert and Validate Accounting Codes function in this module.
Org2_Code	VARCHAR2(2)	Populated from the Convert and Validate Accounting Codes function in this module.
Org3_Code	VARCHAR2(4)	Populated from the Convert and Validate Accounting Codes function in this module.
Org4_Code	NUMBER(2)	Populated from the Convert and Validate Accounting Codes function in this module.
Org5_Code	NUMBER(2)	Populated from the Convert and Validate Accounting Codes function in this module.
Org6_Code	NUMBER(2)	Populated from the Convert and Validate Accounting Codes function in this module.
Org7_Code	NUMBER(2)	Populated from the Convert and Validate Accounting Codes function in this module.
Project_Code	VARCHAR2(7)	Unique within a bureau. A project can only relate to one fund and program. Populated from CFS-FIMA Crosswalk Routine.

Table 3. CONV_ACTDOC_ACCOUNT Table Description		
Column Name	Data Type	Description
Task_Code	VARCHAR2(3)	Three characters that relate to the project. Populated from CFS-FIMA Crosswalk Routine.
Program1_Code	NUMBER(2)	Relates to the Activity, Sub-activity, and Line Item Budget Structure as presented in NOAA Budget Submissions. Populated from the Convert and Validate Accounting Codes function in this module.
Program2_Code	NUMBER(2)	Relates to the Activity, Sub-activity, and Line Item Budget Structure as presented in NOAA Budget Submissions. Populated from the Convert and Validate Accounting Codes function in this module.
Program3_Code	NUMBER(2)	Relates to the Activity, Sub-activity, and Line Item Budget Structure as presented in NOAA Budget Submissions. Populated from the Convert and Validate Accounting Codes function in this module.
Program4_Code	NUMBER(3)	Relates to the Activity, Sub-activity, and Line Item Budget Structure as presented in NOAA Budget Submissions. Populated from the Convert and Validate Accounting Codes function in this module.
Object1_Code	NUMBER(2)	Populated by the major object class code from the Convert and Validate Accounting Codes function in this module.
Object2_Code	NUMBER(2)	Populated by the major object class code from the Convert and Validate Accounting Codes function in this module.

Table 3. CONV_ACTDOC_ACCOUNT Table Description		
Column Name	Data Type	Description
Object3_Code	NUMBER(2)	Populated by the minor object class code from the Convert and Validate Accounting Codes function in this module.
Object4_Code	NUMBER(2)	Populated by the minor object class code from the Convert and Validate Accounting Codes function in this module.
Cfs_Item_Number	NUMBER(3)	Updated by the Undelivered Orders and Unpaid Accrued Expenditures process. Refer to Convert Non-CFS Undelivered Orders and Accruals Program Module.
Cfs_Ea_Item_Id	NUMBER(8)	Updated by the Unpaid Accrued Expenditures process. Refer to Convert Non-CFS Undelivered Orders and Accruals Program Module.
Cfs_Ea_Account_Id	NUMBER(8)	Updated by the Unpaid Accrued Expenditures process. Refer to Convert Non-CFS Undelivered Orders and Accruals Program Module.
Cfs_Recon_Udo_Total	NUMBER(13,2)	Displays the total CFS Undelivered Orders amount for the transactions of the document. This value comes from the Compare FIMA ACTDOC and CFS Production Program Module.
Cfs_Recon_Uae_Total	NUMBER(13,2)	Displays the total CFS Unpaid Accrued Expenditures amount for the transactions of the document. This value comes from the Compare FIMA ACTDOC and CFS Production Program Module.
Cfs_Recon_Udo_Trial_Total	NUMBER(13,2)	Displays the total CFS Undelivered Orders amount for the trial transactions of the document. This value comes from the Compare FIMA ACTDOC and CFS Production Program Module.

Table 3. CONV_ACTDOC_ACCOUNT Table Description		
Column Name	Data Type	Description
Cfs_Recon_Uae_Trial_Total	NUMBER(13,2)	Displays the total CFS Unpaid Accrued Expenditures amount for the trial transactions of the document. This value comes from the Compare FIMA ACTDOC and CFS Production
ACCS_Status_Flag	VARCHAR2(1)	Indicates if the document row has been converted (C), converted and validated (V), or if it has failed conversion (F), or if the row has failed validation (E).
User_Name	VARCHAR2(30)	DB_User_Name of the last person that updated the table.
Modification_Date	DATE	Date the conversion table was last modified.
Device_Name	VARCHAR2(30)	Name of the program running the insert/updates to the table.

The following table is created to record the start and end times of the conversion run for each module. Each program module in the conversion process will be responsible for updating this table at the start and end of its processing.

Table 4. CONV_RUNTIME Table Description		
Column Name	Data Type	Description
Runtime_Seq	NUMBER(7)	System generated number.
Conv_Module	VARCHAR2(30)	Stores the name of the module.
Start_Time	DATE	Stores the date and time stamp of when the module started processing.
End_Time	DATE	Stores the date and time stamp of when the module finished processing.

2.4.1.2 Logic

The new tables, **CONV_ACTDOC** and **CONV_ACTDOC_ACCOUNT**, are created using the formats listed in Tables 2 and 3. After the tables are created, indexes need to be added:

```
create index conv_actdoc_pk on conv_actdoc (document_type,  
document_number);
```

```
create index conv_actdoc_account_pk on conv_actdoc_account  
(document_type, document_number, document_sequence);
```

2.4.1.3 *Output*

The Convert ACTDOC and Convert ACTDOC Account tables are created.

2.4.2 *Populate **CONV_ACTDOC** and **CONV_ACTDOC_ACCOUNT** from FIMA ACTDOC (Sub-function 2)*

This function copies the relevant rows from the FIMA **ACTDOC** table to the **Convert ACTDOC** and **Convert ACTDOC Account** tables. The relevant rows are those with a **Document_Type** between '01' and '49', and a **Completed_Document** field equal to NULL. This selection process will gather all of the open documents in FIMA that need to be converted.

2.4.2.1 *Input*

The input for this sub-function will be the FIMA **ACTDOC** table.

2.4.2.2 *Logic*

To populate the **CONV_ACTDOC** and **CONV_ACTDOC_ACCOUNT** tables:

- ▶ Count the number of rows that currently exist on the FIMA **ACTDOC** table where the **Completed_Document** column has a null value, and the **Document_Type** is between '01' and '49'. This count identifies the number of active rows which currently exist in FIMA. It will be used to verify that the correct number of documents have been copied to the new conversion tables.
- ▶ For each active document, one row will be entered into the **CONV_ACTDOC** table.

-
- ▶ This insert into **CONV_ACTDOC** is done by selecting all of the fields from Table 2 marked with an asterisk (from FIMA **ACTDOC**) where the **Completed_Document** column has a null value and the **Document_Type** is between '01' and '49'.
 - ▶ If there are multiple rows for one document, the detail for each row, will be entered into the **CONV_ACTDOC_ACCOUNT** table.
 - ▶ This insert into **CONV_ACTDOC_ACCOUNT** is done by selecting all of the fields that are to be copied from Table 3 marked with an asterisk (from FIMA **ACTDOC**) where the **Completed_Document** field is equal to NULL, and the **Document_Type** field is between '01' and '49'.
 - ▶ Calculate a running sum of the **Undelivered_Orders** for the line items of each document and enter it into the **FIMA_UDO_Document_Total**. A running sum also needs to be calculated for **Unpaid_Accrued_Exp** for the line items of each document and entered into the **FIMA_UAE_Document_Total**. These values are put into the **CONV_ACTDOC** table as each document is converted.
 - ▶ Once all of the rows have been inserted, count the total number of rows on the **CONV_ACTDOC_ACCOUNT** table. This number should be the same as the number of records counted on the FIMA **ACTDOC** table at the beginning of the process. This information should be written to a log file with the other count to verify that all the data has been converted successfully from FIMA **ACTDOC**.

2.4.2.3 Output

All open FIMA accounts payable documents appear in the **Convert ACTDOC** and **Convert ACTDOC Account** tables.

2.4.3 Record Earliest Fiscal Year on each Document (Sub-Function 3)

The purpose of this function is to determine the minimum fiscal year on each document. This function can be executed once the correct documents have been copied to the **Convert ACTDOC Table** from FIMA **ACTDOC**. The function entails going back to the FIMA **ACTDOC** table and executing a query to find the minimum fiscal year that is associated with each document and then updating the **Min_Fiscal_Year** field on the **Convert ACTDOC Table** with the results.

2.4.3.1 *Input*

The inputs to this sub-function will include a selected set of data from FIMA **ACTDOC** and the **Convert ACTDOC Table**.

2.4.3.2 *Logic*

To populate the **Min_Fiscal_Year** column on **CONV_ACTDOC**:

Select the minimum fiscal year for each FIMA document where the **Document_Number** and **Document_Type** on FIMA **ACTDOC** are equal to the **Document_Number** and **Document_Type** on **CONV_ACTDOC**. Update the **Min_Fiscal_Year** column on **CONV_ACTDOC**.

If the **Min_Fiscal_Year** is not found for a document, then an error should be written to the **Convert Errors Table**, the value of the **Min_Fiscal_Year** field should be set to '9999', and the **Error_Flag** on the **Convert ACTDOC** table should be set to 'Y'.

2.4.3.3 *Output*

When this sub-function is complete, each document found on **Convert ACTDOC** should have its minimum fiscal year stored on the table.

2.4.4 *Record Minimum Open Fiscal Year on Each Document (Sub-Function 4)*

The purpose of this sub-function is to determine the earliest open fiscal year on each document. This sub-function can be executed once the correct documents have been copied to the **Convert ACTDOC Table** from FIMA **ACTDOC**. The process entails going back to FIMA **ACTDOC** and executing a query to find the minimum fiscal year for all records associated with each document where the **Completed_Document** status is NULL (or the document is "open"). The **Min_Open_Fiscal_Year** field on the **Convert ACTDOC Table** is then updated with the results.

2.4.4.1 *Input*

The input to this sub-function will be the data stored on the **Convert ACTDOC Table**.

2.4.4.2 *Logic*

To populate the **Min_Open_Fiscal_Year** column on the **CONV_ACTDOC** table:

Select the minimum fiscal year for each document where the **Completed_Document** field is equal to NULL and **Document_Type** is between '01' and '49'. Update the **Min_Open_Fiscal_Year** column where the **Document_Number** and **Document_Type** on FIMA ACTDOC are equal to the **Document_Number** and **Document_Type** on **CONV_ACTDOC**.

If the **Min_Open_Fiscal_Year** is not found for a document, then an error should be written to the **Convert Errors Table**, the value of the **Min_Open_Fiscal_Year** field should be set to '9999', and the **Error_Flag** on the **Convert ACTDOC** table should be set to 'Y'.

2.4.4.3 *Output*

The output of this sub-function is the population of the **Minimum Open Fiscal Year** field for each document on the **Convert ACTDOC Table**.

2.5.5 *Create **CONV_ERRORS** table (Sub-Function 5)*

This sub-function creates the Convert Errors (**CONV_ERRORS**) table. This table holds specific document information and details associated with errors that occur throughout the entire conversion process. This table can be updated by any of the conversion modules and provides an accurate and detailed description of the error so that appropriate action can be taken.

A record will be placed on the **Convert Errors Table** when the Error Flag (**Error_Flag**), Vendor Error Flag (**Vendor_Error_Flag**), or CFS Document Type Error Flag (**Cfs_Document_Type_Error**) field on the **Convert ACTDOC Table** are set to 'Y' by one of the modules in the conversion process. An error can also be written to the table if the ACCS Status Flag (**ACCS_Status_Flag**) on the **Convert ACTDOC Account Table** has a value of 'E' (for validation errors), or 'F' (for conversion failure).

2.5.5.1 *Table Layout*

The table to be created by this sub-function is listed below in Table 5. Table 6 provides a detailed list of the columns and a data description for each field.

Table 5. Error Table to be Created	
Table Name	Table Description
CONV_ERRORS	This table stores the detail of all errors encountered during the conversion process.

Table 6. CONV_ERRORS Table Description		
Column Name	Data Type	Description
Error_Sequence	NUMBER(5)	System generated sequence number to track the errors entered into the table.
Document_Type	VARCHAR2(2)	Stores the FIMA document type.
Document_Number	VARCHAR2(9)	Stores the FIMA Document Number.
Error_Text	VARCHAR2(100)	A description of the error that occurred in the conversion process.
Conv_Module	VARCHAR2(30)	The name of the module that added the error to the table.
Add_Date	DATE	The date that the error record was added.

2.5.5.2 Logic

In Table 4, the Convert Errors (**CONV_ERRORS**) table is created using the formats listed in Table 6. After the table is created, add the following index:

```
create index conv_errors_pk on conv_errors (document_type,
document_number, error_sequence);
```

2.5.5.3 Output

The **Convert Errors Table** is created.

3 Convert and Validate Accounting Codes (Function 2)

3.1 Purpose of Convert and Validate Accounting Codes

The purpose of this function is to:

- ▶ Convert FIMA accounting codes data to the CFS ACCS.
- ▶ Validate the converted CFS ACCS.
- ▶ Create a re-executable program. (Note: Conversion only occurs once. CFS ACCS codes are manually corrected afterwards).

3.2 Process Flow

Figure 3 illustrates the process flow to convert and validate the FIMA accounting codes into the CFS environment.

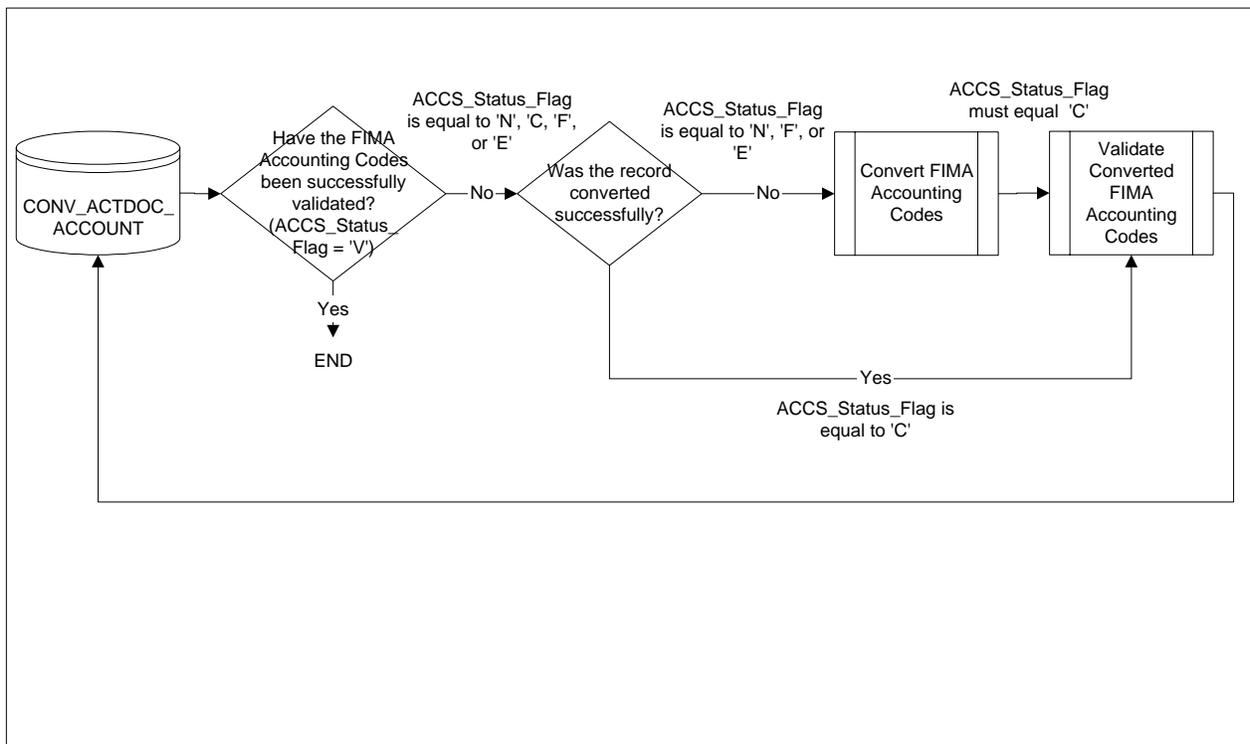


Figure 3. Process Flow to Convert and Validate Accounting Codes

CONV_ACTDOC_ACCOUNT. The Convert ACTDOC Account Table contains the detail information for each accounting line in a document. Data will be selected from this table to perform tasks in this function.

Have the FIMA Accounting Codes been successfully validated? The Program Module determines if the FIMA accounting codes have been successfully validated in a previous attempt of this process (the **ACCS_Status_Flag** on **Convert ACTDOC Account** is 'V'). If the value of the **ACCS_Status_Flag** is not 'V', then the **ACCS_Status_Flag** for the accounting line has a value of 'N' (it has never been converted before), 'C' (the accounting line has only been successfully converted, not validated), 'F' (the accounting line has failed conversion in a previous attempt), or 'E' (the accounting line has previously failed validation). In these four cases, the accounting line must move to the question: *Was the record converted successfully?* If the value of the **ACCS_Status_Flag** is 'V', then the Program Module ends its processing of that accounting line.

Was the record converted successfully? If the accounting codes have been successfully validated, the Program Module determines if the FIMA accounting codes have been successfully converted. This situation occurs when the **ACCS_Status_Flag** on the **Convert ACTDOC Account Table** is equal to 'N' (the line has never been converted), 'F' (the accounting line has previously failed conversion), or 'E' (the accounting line has previously failed validation). If the value of the **ACCS_Status_Flag** matches any of the values above, then the accounting line moves to the *Convert FIMA Accounting Codes* sub-function. If the value of the **ACCS_Status_Flag** is equal to 'C' instead, then the accounting line must only go through the *Validate FIMA Converted Accounting Codes* sub-function.

Convert FIMA Accounting Codes. This sub-function converts the FIMA accounting codes into the CFS environment. If this function fails, the **ACCS_Status_Flag** on the **Convert ACTDOC Account Table** is updated to 'F', indicating that the accounting line has failed conversion. If the function is successful, the **ACCS_Status_Flag** on the **Convert ACTDOC Account Table** is updated to 'C', indicating that the accounting line has been successfully converted.

Validate Converted Accounting Codes. This sub-function validates the new CFS ACCS against existing table data in CFS Production. If this function fails, the **ACCS_Status_Flag** on the **Convert ACTDOC Account Table** is updated to 'E', indicating that the accounting line has failed validation. If this function is successful, then the **ACCS_Status_Flag** on the **Convert ACTDOC Account Table** is updated to 'V', indicating that the accounting line been validated successfully.

3.3 Input

The input to this function will be rows found on the Convert ACTDOC (**CONV_ACTDOC**) and Convert ACTDOC Account (**CONV_ACTDOC_ACCOUNT**) tables where:

- ▶ Documents are not yet converted (**Conv_Status_Flag = 'N'**).
- ▶ The ACCS of documents have not been successfully converted and validated (**ACCS_Status_Flag** does not equal 'V').

3.4 Convert and Validate Accounting Codes Processing

3.4.1 Convert FIMA Accounting Codes (Sub-Function 1)

This function will convert the FIMA accounting codes into a format that is accepted by CFS. The Conversion Status Flag (**Conv_Status_Flag**) present on the Convert ACTDOC (**CONV_ACTDOC**) table is used to determine whether or not a document is considered for conversion. Each time the program is executed, only documents with a Conversion Status Flag of 'N' should be considered.

If the conversion of the ACCS is completed successfully, then the ACCS Status Flag (**ACCS_Status_Flag**) on the Convert ACTDOC Account (**CONV_ACTDOC_ACCOUNT**) table should be updated to 'C' to indicate that the document line has been successfully converted. Otherwise, the flag should be updated to 'F' to indicate that the conversion has failed.

3.4.1.1 Process Flow for Convert FIMA Accounting Codes

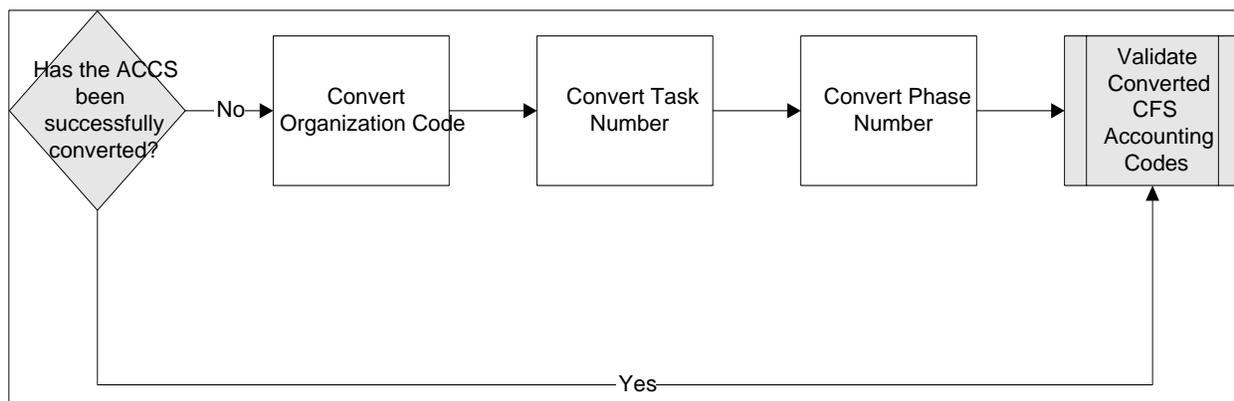


Figure 4. Process Flow for Convert FIMA Accounting Codes

The functions shown above are a breakdown of the *Convert FIMA Accounting Codes* sub-function.

Has the ACCS been successfully converted? If the document has not been successfully converted in a previous run of the Program Module (the **Conv_Status_Flag** equal to 'N'), then the accounting codes are converted.

Convert Organization Code. This sub-function converts the FIMA Organization Code into a format usable by CFS.

Convert Task Number. This sub-function converts the FIMA Task Number into a format usable by CFS.

Convert Phase Number. This sub-function converts the FIMA Phase Number into a format usable by CFS.

Once these values have been converted, they will move to the *Validate Converted CFS Accounting Codes* sub-function for processing.

3.4.1.2 Input

The input for this sub-function will be documents on the Convert ACTDOC (**CONV_ACTDOC**) and Convert ACTDOC Account (**CONV_ACTDOC_ACCOUNT**) tables that:

- ▶ Are not yet converted (**Conv_Status_Flag** = 'N').
- ▶ Have not been successfully converted (**ACCS_Status_Flag** is equal to 'N', 'E', or 'F').

3.4.1.3 Logic

To convert the FIMA accounting data into a format usable by CFS, the module should mirror the approach and code found in the *FIMA - CAMS ACCS Conversion Lookup Screen (FIM302)* in CFS.

If an error is found in the conversion process, a record should be added to the **Convert Errors** table, and the **Error_Flag** on the **Convert ACTDOC Table** should be set to 'Y'.

3.4.1.4 Output

The output of this sub-function is a group of successfully converted accounting codes for each document line on the **Convert ACTDOC Table**.

3.4.2 Validate Converted CFS Accounting Codes

This process will validate the FIMA Accounting code values once they have been converted into CFS. The **ACCS_Status_Flag** on the **Convert ACTDOC Account Table** and the **Conv_Status_Flag** on the **Convert ACTDOC Table** will make this distinction. Each time the program is executed, only documents with a **ACCS_Status_Flag** equal to 'C' and **Conv_Status_Flag** of 'N' should be considered.

If the validation of ACCS codes is completed successfully, then the **ACCS_Status_Flag** on the **Convert ACTDOC Account Table** should be updated to 'V' to indicate that the document line has been successfully validated. Otherwise, the flag should be updated to 'E' to indicate that the validation has failed.

3.4.2.1 Validate Converted CFS Accounting Codes Sub-Process Flow

Figure 5 illustrates the process that a document follows when the FIMA

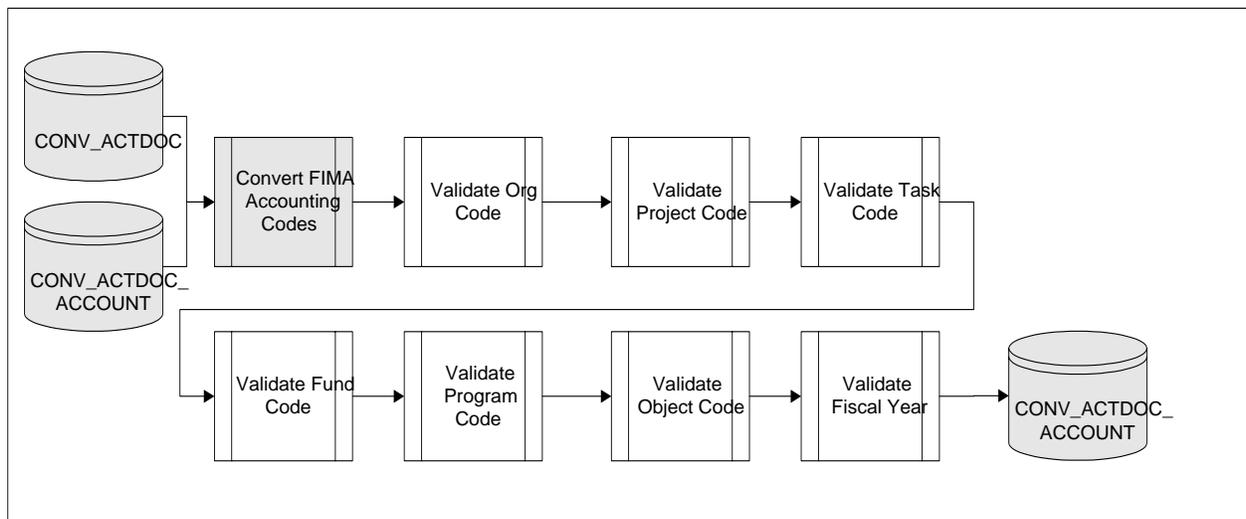


Figure 5. Process Flow for Validate Converted CFS Accounting Codes

accounting codes are validated.

The Validate Org Code function will verify that the CFS Organization Code of the current document is present and active in the CFS Organization code tables.

The Validate Project Code function will verify that the CFS Project Code of the current document is present and active in the CFS Project code tables.

The Validate Task Code function will verify that the CFS Task Code of the current document is present and active in the CFS Task code tables.

The Validate Fund Codes function will verify that the CFS Fund Code of the current document is present and active in the CFS Fund code tables.

The Validate Program Code function will verify that the CFS Program Code of the current document is present and active in the CFS Program code tables.

The Validate Object Codes function will verify that the CFS Object Code of the current document is present and active in the CFS Object code tables.

The Validate Fiscal Year function will verify that the CFS Fiscal Year of the current document is present and active in the CFS Fiscal Year tables.

3.4.2.2 Input

The input for this sub-function will be all documents on the **Convert ACTDOC** and **Convert ACTDOC Account** tables that:

- ▶ Have a value of 'N' in the **Conv_Status_Flag** column.
- ▶ Have an **ACCS_Status_Flag** value equal to 'C'.

3.4.2.3 Logic

The accounting codes validation process should mirror the approach and code found in the *Purchase Order Transaction Screen (FM040)* and the *Estimated Accrual Transaction Screen (PM050)* in CFS.

If any steps of the validation process should fail, the **ACCS_Status_Flag** should be updated to 'E' on the **Convert ACTDOC Account Table**, indicating that an error occurred, and a record should be written to the **Convert Errors Table** describing what part of the validation failed.

At the end of the validation process, the **ACCS_Status_Flag** is checked for the following:

-
- ▶ If the value is 'C' (set from the ACCS conversion process), then all of the validations have been passed and the **ACCS_Status_Flag** is updated with a 'V' to indicate that the document row has been successfully validated and converted.
 - ▶ If the **ACCS_Status_Flag** is 'E,' indicating that the row did not pass part of the validation, the **Error_Flag** on the **CONV_ACTDOC** table is updated to 'Y'.

3.4.2.4 Output

The output for this section is an updated value in the **ACCS_Status_Flag** column in the document's row on the **Convert ACTDOC Account Table**.

3.5 Validate Document Bureau Codes

3.5.1 Validate Bureau Codes

This function verifies that a document on the **Convert ACTDOC Account Table** does not have more than one CFS Bureau Code (**Bureau_Code**) associated with it.

3.5.2 Input

The input to this function are the documents on the **Convert ACTDOC Account Table** where all accounting lines on the document have been successfully converted and validated.

3.5.3 Logic

In order to correctly validate the **Bureau Code** for each document, the program must look for all accounting lines with the same **Document Number** and check that they all have the same **Bureau Code**.

If any document is found to have more than one **Bureau Code**, then an error should be written to the **Convert Errors Table** and the **Error Flag** on the **Convert ACTDOC Table** should be updated to 'Y'.

3.5.4 Output

The output of this function is a count of the **Bureau Codes** on each document in the **Convert ACTDOC Account Table**.

4 Overall Issues

- ▶ The CD&C Team must determine how users will access the **Convert ACTDOC Table** information to correct errors that prevent documents from being converted.
Issue Owner(s): CAMS Design and Conversion Team
- ▶ The CD&C Team must determine how to effectively clean up data that is not successfully converted.
Issue Owner(s): CAMS Design and Conversion Team

5 References

The references for this document can be found in the Overview of the Automated Accounts Payable Open Document Conversion Program Module.