

6. SOAP/XML FILES (EOD).....2

6.1 FOOD INSTRUMENT FILE IMPORT.....2

6.2 FOOD DISTRIBUTION FILE EXPORT18

6.3 FOOD INSTRUMENT TYPES FILE EXPORT22

6.4 CSFP STOP PAYMENT FILE IMPORT26

6.5 CSFP NEW ISSUANCE FILE IMPORT31

6.6 VENDOR FILE EXPORT44

6.7 SOAP INTERNAL DESIGN52

6. SOAP/XML Files (EOD)

With the need for different computer systems to share information with one another, a non-intrusive way to do this is needed. In order for either system to be able to share information, some type of universal format will have to be created. This universal format will be in a Simple Object Access Protocol (SOAP) format. The following sections will describe which methods and data that will be needed to complete the transactions.

6.1 Food Instrument File Import

This SOAP method will have Agency Information and all the food instruments from that agency. This will be an update of the food instrument database tables with the information that is received. The sending application will be responsible for building the SOAP file with all the food instrument information. If EOD_FOODINSTRUMENTFILEIMPORT = 'Y' and a Food Instrument file is located in the \\<STATECODE>EOD\FOODINSTRUMENT\REQUEST\ directory, the End of Day Process will Add/Update Issued, Paid or Rejected Food Instrument data in the associated Food Instrument tables. See Chapter 03 - End of Day Processes (Product).doc for more information on processing.

6.1.1 Food Instrument File Layout

```
<FoodInstruments>
  <FromMachine>string</FromMachine>
  <ToMachine>string</ToMachine>
  <Date>date</Date>
  <Time>time</Time>
  <Version>string</Version>
  <Site_Information>
    <Agency_ID>1234</Agency_ID>
    <Local_Agency>4321</Local_Agency>
    <ServiceSite>1111</ServiceSite>
  <Participant_Info>
    <WICID>1222</WICID>
    <TransactionCode>string</TransactionCode>
    <CurrentWICStatus>string</CurrentWICStatus>
    <CreateUserID>string</CreateUserID>
    <CreateDate>date</CreateDate>
  <FoodInstrumentSet>
    <DateIssued>string</DateIssued>
    <WICStatus>string</WICStatus>
    <ParticipationMonth>string</ParticipationMonth>
    <ExpirationDate>date</ExpirationDate>
    <Priority>string</Priority>
    <SubPriority>string</SubPriority>
    <First_Date_To_Use>date</First_Date_To_Use>
    <Last_Date_To_Use>date</Last_Date_To_Use>
    <ComplianceBuy>boolean</ComplianceBuy>
    <PC_Create_User>string</PC_Create_User>
    <PC_Create_Date>date</PC_Create_Date>
    <PC_Modified_User>string</PC_Modified_User>
    <PC_Modified_Date>date</PC_Modified_Date>
```

```
<FoodInstrument>
  <SerialNumber>string</SerialNumber>
  <StateWICID>string</StateWICID>
  <IssuedCode>string</IssuedCode>
  <Rebate>string</Rebate>
  <PharmacyAllowed>string</PharmacyAllowed>
  <VoidedDate>date</VoidedDate>
  <VoidedReason>string</VoidedReason>
  <ReprintReason>string</ReprintReason>
  <ProgramCode>string</ProgramCode>
  <ReportedStolenDate>date</ReportedStolenDate>
  <StaffIDVoidStolen>string</StaffIDVoidStolen>
  <ObligationAmt>number</ObligationAmt>
  <PresentmentAmt>number</PresentmentAmt>
  <PaidAmt>number</PaidAmt>
  <PaidDate>date</PaidDate>
  <ExpirationDate>date</ExpirationDate>
  <VendorID>string</VendorID>
  <PeerGroupID>string</PeerGroupID>
  <Stampnumber>string</Stampnumber>
  <FoodInstrumentType>string</FoodInstrumentType>
  <CreateUserID>string</CreateUserID>
  <CreateUserDate>date</CreateUserDate>
  <ModifyUserID>string</ModifyUserID>
  <ModifyUserDate>date</ModifyUserDate>
  <UpdateRecord>string</UpdateRecord>
  <Rejections>
    <SerialNo>string</SerialNo>
    <RejectionCode>string</RejectionCode>
    <ModifyDate>date</ModifyDate>
    <ModifyUserID>string</ModifyUserID>
    <RejectionDate>date</RejectionDate>
    <UpdateRecord>string</UpdateRecord>
    <CreateUserID>string</CreateUserID>
    <CreateDate>date</CreateDate>
  </Rejections>
  <FoodItem>
    <FoodItemCode>string</FoodItemCode>
    <FoodItemQty>number</FoodItemQty>
    <Food_RX_Item_Code>string</Food_RX_Item_Code>
  </FoodItem>
</FoodInstrument>
</FoodInstrumentSet>
</Participant_Info>
</Site_Information>
</FoodInstruments>
```

6.1.2 File Elements

6.1.2.1 Food Instruments

6.1.2.1.1 FromMachine

This is the name of the machine that sent the file. The machine that receives the file will check this element and make sure that the file is from an authorized to send a file.

6.1.2.1.1.1 Characteristics

This element will be an alphanumeric field.

6.1.2.1.1.2 Edits

This is a required field.

6.1.2.1.2 ToMachine

This is the name of the machine to which the file was sent. The machine that receives the file will check this element and make sure that the file is to be sent to the machine.

6.1.2.1.2.1 Characteristics

This element will be an alphanumeric field.

6.1.2.1.2.2 Edits

This is a required field.

6.1.2.1.3 Date

This is the date that the file was build.

6.1.2.1.3.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.1.2.1.3.2 Edits

This is a required field.

6.1.2.1.4 Time

This is the time that the file was build.

6.1.2.1.4.1 Characteristics

This element will be a numeric field. It will have a format of HHMMSS. It will be in a 24 hour format.

6.1.2.1.4.2 Edits

This is a required field.

6.1.2.1.5 Version

This is the file version that is coming in.

6.1.2.1.5.1 Characteristics

This element will be a numeric field. It will have a format of XXX.XX.XX.

6.1.2.1.5.2 Edits

This is a required field.

6.1.2.2 Site_Information

This section is the information on the site that the food instruments came from.

6.1.2.2.1 Agency_ID

This is the Agency ID where the food instruments were created.

6.1.2.2.1.1 Characteristics

This element will be a numeric field. It will have a format of XXX.

6.1.2.2.1.2 Edits

This is not a required field.

6.1.2.2.2 Local_Agency

This is the local agency where the food instruments were created

6.1.2.2.1 Characteristics

This element will be a numeric field. It will have a format of XXX.

6.1.2.2.2 Edits

This is not a required field. If there is a value in Agency_ID, then this field is required.

6.1.2.2.3 ServiceSite

This is the Service Site where the food instruments were created.

6.1.2.2.3.1 Characteristics

This element will be a numeric field. It will have a format of XXX

6.1.2.2.3.2 Edits

This is not a required field. If there is a value in Agency_ID, then this field is required.

6.1.2.3 Participant_Info

This section is the information on the participant.

6.1.2.3.1 WICID

This is the participant's unique identifier to the system.

6.1.2.3.1.1 Characteristics

This element will be an alphanumeric field. It will have a format of XXXXXXXX where the identifier is zero padded to the left.

6.1.2.3.1.2 Edits

This is not a required field.

6.1.2.3.2 CurrentWICStatus

This is the participant's current WIC Category (Infant, Child, Breastfeeding, etc).

6.1.2.3.2.1 Characteristics

This element will be an alphanumeric field. It will be one of the following codes (F – Formula Fed Infant, I – Breastfed Infant, C – Child, B – Breastfeeding, N – Non-Breastfeeding, P – Pregnant, S - Senior).

6.1.2.3.2.2 Edits

This is not a required field. If WICID is there, then this is a required field.

6.1.2.4 FoodInstrumentSet

This section has information on the food instrument set that was issued to the participant.

6.1.2.4.1 DateIssued

This is the date that the participant's checks were issued.

6.1.2.4.1.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.1.2.4.1.2 Edits

This is not a required field.

6.1.2.4.2 WICStatus

This is the participant's WIC Category when the food instruments were issued.

6.1.2.4.2.1 Characteristics

This element will be an alphanumeric field. It will be one of the following codes (F – Formula Fed Infant, I – Breastfed Infant, C – Child, B – Breastfeeding, N – Non-Breastfeeding, P – Pregnant, S - Senior).

6.1.2.4.2.2 Edits

This is a required field if the WICID is populated.

6.1.2.4.3 ParticipationDate

This is the date that represents the month the food instrument is reported.

6.1.2.4.3.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.1.2.4.3.2 Edits

This is a required field if the WICID is populated.

6.1.2.4.4 Priority

This is the participant's risk priority at the time the food instrument set was issued.

6.1.2.4.4.1 Characteristics

This element will be an alphanumeric field.

6.1.2.4.4.2 Edits

This is a required field if the WICID is populated.

6.1.2.4.5 SubPriority

This is the participant's risk sub priority at the time the food instrument set was issued.

6.1.2.4.5.1 Characteristics

This element will be an alphanumeric field.

6.1.2.4.5.2 Edits

This is not a required field.

6.1.2.4.6 First_Date_To_Use

This is the first date the food instrument set can be used.

6.1.2.4.6.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.1.2.4.6.2 Edits

This is a required field if the WICID is populated.

6.1.2.4.7 Last_Date_To_Use

This is the last date the food instrument set can be used.

6.1.2.4.7.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.1.2.4.7.2 Edits

This is a required field if the WICID is populated.

6.1.2.4.8 ComplianceBuy

This is a flag to tell if the food instrument set is a compliance buy or not.

6.1.2.4.8.1 Characteristics

This element will be a numeric field. It will have a value of 'N' if the food instrument set is not a compliance buy and 'Y' if it is a compliance buy.

6.1.2.4.8.2 Edits

This is not a required field.

6.1.2.4.9 PC_Create_User

This is the name of the user who created the food instrument set.

6.1.2.4.9.1 Characteristics

This element will be an alphanumeric field.

6.1.2.4.9.2 Edits

This is not a required field.

6.1.2.4.10 PC_Create_Date

This is the date in which the food instrument set was created.

6.1.2.4.10.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.1.2.4.10.2 Edits

This is not a required field.

6.1.2.4.11 PC_MODIFIED_USER

This is the name of the user who modified the food instrument set.

6.1.2.4.11.1 Characteristics

This element will be an alphanumeric field.

6.1.2.4.11.2 Edits

This is not a required field.

6.1.2.4.12 PC_MODIFIED_DATE

This is the date in which the food instrument set was created.

6.1.2.4.12.1

This element will be a numeric field. It will have a format of MMDDYYYY.

6.1.2.4.12.2 Edits

This is not a required field.

6.1.2.5 FoodInstrument

This is the information on the actual food instrument.

6.1.2.5.1 Check (Serial) Number

This is the check (serial) number of the food instrument that is being imported.

6.1.2.5.1.1 Characteristics

This element will be a numeric field and could have leading zeros.

6.1.2.5.1.2 Edits

This is a required field.

6.1.2.5.2 FoodInstrumentType

This is the code for the type of food instrument that is being imported.

6.1.2.5.2.1 Characteristics

This element will be an alphanumeric field and could have leading zeros.

6.1.2.5.2.2 Edits

This is not a required field.

6.1.2.5.3 IssuedCode

This is the code indicating what is happening with this food instrument.

6.1.2.5.3.1 Characteristics

This element will be an alphanumeric field. It will be one of the following codes: I – Issuance, V - Void

6.1.2.5.3.2 Edits

This is not a required field.

6.1.2.5.4 ProgramCode

This is the code indicating what program this food instrument was produced for.

6.1.2.5.4.1 Characteristics

This element will be a numeric field. It will be one of the following codes: 0 – WIC, 1 – Farmers Market, 2 – Senior’s Farmers Market

6.1.2.5.4.2 Edits

This is not a required field.

6.1.2.5.5 PharmacyAllowed

This is a code indicating if the food instrument can only be redeemed at a pharmacy.

6.1.2.5.5.1 Characteristics

This element will be a numeric field. It will be 'Y' if it can only be redeemed at a pharmacy and 'N' if it can be redeemed at any vendor.

6.1.2.5.5.2 Edits

This is not a required field.

6.1.2.5.6 PaidAmt

This is the amount that the food instrument was redeemed for by the vendor and accepted by the bank.

6.1.2.5.6.1 Characteristics

This element will be a numeric field. It will have a format of XXXX.XX where the first 4 digits are dollars and the last two digits are cents.

6.1.2.5.6.2 Edits

This is not a required field.

6.1.2.5.7 VendorID

This is the code of the vendor who accepted the food instrument.

6.1.2.5.7.1 Characteristics

This element will be an alphanumeric field. It will have a format of WWWWWW where the identifier is zero padded to the left.

6.1.2.5.7.2 Edits

This is not a required field.

6.1.2.5.8 VoidedDate

This is the date the food instrument was voided

6.1.2.5.8.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.1.2.5.8.2 Edits

This is not a required field.

6.1.2.5.9 VoidedReason

This is a code for the reason that the food instrument was voided.

6.1.2.5.9.1 Characteristics

This element will be an alphanumeric field. It will have a format of WW where the identifier is zero padded to the left.

6.1.2.5.9.2 Edits

This is not a required field.

6.1.2.5.10 StateWICID

This is the StateWICID of the participant that was issued the food instrument.

6.1.2.5.10.1 Characteristics

This element will be an alphanumeric field. It will have a format of WWWWWWWW and will be zero padded.

6.1.2.5.10.2 Edits

This is not a required field.

6.1.2.5.11 Rebate

This is a flag that indicates whether or not the food instrument includes a rebate item.

6.1.2.5.11.1 Characteristics

If the food instrument does include a rebate item the value is the contractor's code.

6.1.2.5.11.2 Edits

This is not a required field.

6.1.2.5.12 ReportedStolenDate

This is the date the food instrument was reported stolen.

6.1.2.5.12.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.1.2.5.12.2 Edits

This is not a required field. If the Void date is non-zero then this field should be zero, If this field is non-zero the Void date should be zero.

6.1.2.5.13 StaffIDVoidStolen

This is the ID of the person that voided the food instrument or marked it as stolen.

6.1.2.5.13.1 Characteristics

This element will be a character field.

6.1.2.5.13.2 Edits

This is not a required field. This field should be populated if the Void or Reported Stolen date is not zero.

6.1.2.5.14 ObligationAmt

This is the price the State can expect to pay for this food instrument.

6.1.2.5.14.1 Characteristics

This element will be a numeric field. It will have a format of XXXX.XX where the first 4 digits are dollars and the last two digits are cents.

6.1.2.5.14.2 Edits

This is not a required field.

6.1.2.5.15 PresentmentAmt

Presentment Amount is the dollar amount of the food instrument presented for payment to the bank.

6.1.2.5.15.1 Characteristics

This element will be a numeric field. It will have a format of XXXX.XX where the first 4 digits are dollars and the last two digits are cents.

6.1.2.5.15.2 Edits

This is not a required field

6.1.2.5.16 ExpirationDate

This is the last date that the bank should make a payment on this food instrument.

6.1.2.5.16.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.1.2.5.16.2 Edits

This is not a required field.

6.1.2.5.17 PeerGroupID

This is the peer group value assigned to the redeeming vendor at the time the food instrument is paid.

6.1.2.5.17.1 Characteristics

This element will be an alphanumeric field.

6.1.2.5.17.2 Edits

This is not a required field.

6.1.2.5.18 StampNumber

This is the stamp number the vendor used on the food instrument.

6.1.2.5.18.1 Characteristics

This element will be a numeric field

6.1.2.5.18.2 Edits

This is not a required field, unless the food instrument has passed the bank.

6.1.2.5.19 CreateUserID

This is the name of the user who created the food instrument.

6.1.2.5.19.1 Characteristics

This element will be an alphanumeric field.

6.1.2.5.19.2 Edits

This is not a required field.

6.1.2.5.20 CreateUserDate

This is the date in which the food instrument was created.

6.1.2.5.20.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.1.2.5.20.2 Edits

This is not a required field.

6.1.2.5.21 ModifyUserID

This is the name of the user who modified the food instrument.

6.1.2.5.21.1 Characteristics

This element will be an alphanumeric field.

6.1.2.5.21.2 Edits

This is not a required field.

6.1.2.5.22 ModifyUserData

This is the date on which the food instrument was modified.

6.1.2.5.22.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.1.2.5.22.2 Edits

This is not a required field.

6.1.2.6 Rejections

6.1.2.6.1 RejectionCode

This is the reason food instrument was rejected from the bank.

6.1.2.6.1.1 Characteristics

This element will be an alphanumeric field. It will have a format of WW where the identifier is zero padded to the left

6.1.2.6.1.2 Edits

This is a required field

6.1.2.6.2 RejectionDate

This is the date the food instrument was rejected from the bank.

6.1.2.6.2.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.1.2.6.2.2 Edits

This is a required field.

6.1.2.6.3 CreateUserID

This is the name of the user who created the food instrument.

6.1.2.6.3.1 Characteristics

This element will be an alphanumeric field.

6.1.2.6.3.2 Edits

This is not a required field.

6.1.2.6.4 CreateUserDate

This is the date in which the food instrument was created.

6.1.2.6.4.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.1.2.6.4.2 Edits

This is not a required field.

6.1.2.6.5 ModifyUserID

This is the name of the user who modified the food instrument.

6.1.2.6.5.1 Characteristics

This element will be an alphanumeric field.

6.1.2.6.5.2 Edits

This is not a required field.

6.1.2.6.6 ModifyUserDate

This is the date on which the food instrument was modified.

6.1.2.6.6.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.1.2.6.6.2 Edits

This is not a required field

6.2 Food Distribution File Export

This SOAP method will have all the Food Items that are in the COVANSYS system that will be used in the STATE system. This is done so that both systems will have the same information on the food items. The sending application will be responsible for building the SOAP file with all the food items information. This process is applicable for your state if the State Business Rule EOD_FOODITEMSFILEEXPORT = 'Y'. See Chapter 03 - End of Day Processes (Product).doc for more information on processing.

6.2.1 Food Distribution Items File Layout

```
<FOODITEMS>  
  <FROMMACHINE />  
  <TOMACHINE />  
  <DATE />  
  <TIME />  
  <VERSION />  
  <FOODITEM>  
    <DistributionitemID />  
    <Description />  
    <Description2 />  
    <BaseFoodCategoryID />  
  </FOODITEM>  
</FOODITEMS>
```

6.2.2 File Elements

6.2.2.1 Food Items

6.2.2.1.1 FromMachine

This is the name of the machine that sent the file. The machine that receives the file will check this element and make sure that the file is from an authorized to send a file.

6.2.2.1.1.1 Characteristics

This element will be an alphanumeric field.

6.2.2.1.1.2 Edits

This is a required field.

6.2.2.1.2 ToMachine

This is the name of the machine to which the file was sent. The machine that receives the file will check this element and make sure that the file is to be sent to the machine.

6.2.2.1.2.1 Characteristics

This element will be an alphanumeric field.

6.2.2.1.2.2 Edits

This is a required field.

6.2.2.1.3 Date

This is the date that the file was build.

6.2.2.1.3.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.2.2.1.3.2 Edits

This is a required field.

6.2.2.1.4 Time

This is the time that the file was build.

6.2.2.1.4.1 Characteristics

This element will be a numeric field. It will have a format of HHMMSS. It will be in a 24 hour format.

6.2.2.1.4.2 Edits

This is a required field.

6.2.2.1.5 Version

This is the file version that is coming in.

6.2.2.1.5.1 Characteristics

This element will be a numeric field. It will have a format of XXX.XX.XX.

6.2.2.1.5.2 Edits

This is a required field.

6.2.2.2 Food Item

This will have the food item information that will be received from the STATE System. It may have many different records.

6.2.2.2.1 DistributionItemID

This is the unique identifier for the food item.

6.2.2.2.1.1 Characteristics

This element will be a numeric field, zero padded to the left.

6.2.2.2.1.2 Edits

This is a required field.

6.2.2.2.2 Description

This is the description of the food item.

6.2.2.2.2.1 Characteristics

This element will be an alphanumeric field.

6.2.2.2.2.2 Edits

This is a required field.

6.2.2.2.3 Description2

This is the description of the food item.

6.2.2.2.3.1 Characteristics

This element will be an alphanumeric field.

6.2.2.2.3.2 Edits

This is a not a required field.

6.2.2.2.4 BaseFoodCategoryID

This is the Base Food Category identifier for the food item.

6.2.2.2.4.1 Characteristics

This element will be an alpha field.

6.2.2.2.4.2 Edits

This is a required field.

6.3 Food Instrument Types File Export

This SOAP method will have all the Food Instrument Types in the COVANSYS system that will be used in the STATE system. This is done so that both systems will have the same information on the food instrument types. The sending application will be responsible for building the SOAP file with all the food instrument type information. This process will be applicable for your state if the State Business Rule EOD_FOODINSTTYPEFILEEXPORT = 'Y'. See Chapter 03 - End of Day Processes (Product).doc for more information on processing.

6.3.1 Food Instrument Types File Layout

```
<FILEELEMENTS>
  <FROMMACHINE />
  <TOMACHINE />
  <DATE />
  <TIME />
  <VERSION />
  <FOODINSTRUMENTS>
    <FoodInstrument />
    <WICSTATUSES>
      <WICStatus />
    </WICSTATUSES>
    <ITEM>
      <DistributionItemID />
      <Quantity />
    </ITEM>
  </FOODINSTRUMENTS>
</FILEELEMENTS>
```

6.3.2 File Elements

6.3.2.1 FileElements

6.3.2.1.1 FromMachine

This is the name of the machine that sent the file. The machine that receives the file will check this element and make sure that the file is from an authorized to send a file.

6.3.2.1.1.1 Characteristics

This element will be an alphanumeric field.

6.3.2.1.1.2 Edits

This is a required field.

6.3.2.1.2 ToMachine

This is the name of the machine to which the file was sent. The machine that receives the file will check this element and make sure that the file is to be sent to the machine.

6.3.2.1.2.1 Characteristics

This element will be an alphanumeric field.

6.3.2.1.2.2 Edits

This is a required field.

6.3.2.1.3 Date

This is the date that the file was build.

6.3.2.1.3.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.3.2.1.3.2 Edits

This is a required field.

6.3.2.1.4 Time

This is the time that the file was build.

6.3.2.1.4.1 Characteristics

This element will be a numeric field. It will have a format of HHMMSS. It will be in a 24 hour format.

6.3.2.1.4.2 Edits

This is a required field.

6.3.2.1.5 Version

This is the file version that is coming in.

6.3.2.1.5.1 Characteristics

This element will be a numeric field. It will have a format of XXX.XX.XX.

6.3.2.1.5.2 Edits

This is a required field.

6.3.2.2 FoodInstruments

This will have the food instrument information that will be sent to the STATE System. It may have many different records.

6.3.2.2.1 FoodInstrument

This is the unique identifier for the food item.

6.3.2.2.1.1 Characteristics

This element will be an alphanumeric field.

6.3.2.2.1.2 Edits

This is a required field.

6.3.2.3 WICStatuses

This will have the WIC Categories for this food instrument type. These are the WIC Categories that will be able to use these food instrument types. It will be sent to the COVANSYS System to be used for the food instrument production. It may have many different records. It is not required for all States.

6.3.2.3.1 WICStatus

This is the WIC Category for the food instrument.

6.3.2.3.1.1 Characteristics

This element will be an alpha field.

6.3.2.3.1.2 Edits

This is a required field.

6.3.2.4 Item

This will have the food item information that will be on the food instrument type. Each item record is a line on the food instrument type. It will be sent to the COVANSYS System to be used for the food instrument production. It may have many different records.

6.3.2.4.1 DistributionItemID

This is the unique identifier for the food item.

6.3.2.4.1.1 Characteristics

This element will be a numeric field, zero padded to the left.

6.3.2.4.1.2 Edits

This is a required field.

6.3.2.4.2 Quantity

This is the quantity of the food item that is on the food instrument.

6.3.2.4.2.1 Characteristics

This element will be a numeric field.

6.3.2.4.2.2 Edits

This is a required field.

6.4 CSFP Stop Payment File Import

The End of Day process will import CSFP Stop Payment data from an XML file format using SOAP methods. This process is applicable for your state if the State Business Rule EOD_CSFPSTOPPAYFILEIMPORT = 'Y'. See Chapter 03 - End of Day Processes (Product).doc for more information on processing.

6.4.1 CSFP Stop Payment File Layout

```
<STOPPAYMENT>
  <FromMachine />
  <ToMachine />
  <Date />
  <Time />
  <Version />
  <CHECK>
    <SerialNo />
    <IssuedCode />
    <PrgmCode />
    <VoidDate />
    <VoidCode />
    <StolenDate />
    <StaffStopPay />
    <ExprtnDate />
    <FIModifyUser />
    <FIModifyDate />
    <FIUpdateRec />
  </CHECK>
</STOPPAYMENT>
```

6.4.2 File Elements

6.4.2.1 Stop Payment

6.4.2.1.1 FromMachine

This is the name of the machine that sent the file. The machine that receives the file will check this element and make sure that the file is from an authorized to send a file.

6.4.2.1.1.1 Characteristics

This element will be an alphanumeric field.

6.4.2.1.1.2 Edits

This is a required field.

6.4.2.1.2 ToMachine

This is the name of the machine to which the file was sent. The machine that receives the file will check this element and make sure that the file is to be sent to the machine.

6.4.2.1.2.1 Characteristics

This element will be an alphanumeric field.

6.4.2.1.2.2 Edits

This is a required field.

6.4.2.1.3 Date

This is the date that the file was build.

6.4.2.1.3.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.4.2.1.3.2 Edits

This is a required field.

6.4.2.1.4 Time

This is the time that the file was build.

6.4.2.1.4.1 Characteristics

This element will be a numeric field. It will have a format of HHMMSS. It will be in a 24 hour format.

6.4.2.1.4.2 Edits

This is a required field.

6.4.2.1.5 Version

This is the file version that is coming in.

6.4.2.1.5.1 Characteristics

This element will be a numeric field. It will have a format of XXX.XX.XX.

6.4.2.1.5.2 Edits

This is a required field.

6.4.2.2 *Check*

6.4.2.2.1 SerialNo (Check Number)

This is the check (serial) number of the food instrument.

6.4.2.2.1.1 *Characteristics*

This element will be a numeric field and could have leading zeros.

6.4.2.2.1.2 *Edits*

This is a required field.

6.4.2.2.2 IssuedCode

This is the code indicating what is happening with this food instrument

6.4.2.2.2.1 *Characteristics*

This element will be an alphanumeric field. It will be one of the following codes: I – Issuance, V - Void.

6.4.2.2.2.2 *Edits*

This is a not required field.

6.4.2.2.3 PrgrmCode

This is the code indicating the program for which the food instrument was issued.

6.4.2.2.3.1 *Characteristics*

This element will be a numeric field. It will be one of the following codes: 0 – WIC, 1 – Farmers Market, 2 – Senior’s Farmers Market

6.4.2.2.3.2 *Edits*

This is a required field.

6.4.2.2.4 VoidDate

This is the date the food instrument was voided.

6.4.2.2.4.1 *Characteristics*

This element will be a numeric field. It will have a format of MMDDYYYY.

6.4.2.2.4.2 Edits

This is a not required field.

6.4.2.2.5 VoidCode

This is the code indicating why the food instrument was voided.

6.4.2.2.5.1 Characteristics

This element will be an alphanumeric field.

6.4.2.2.5.2 Edits

This is a not required field.

6.4.2.2.6 StolenDate

This is the date the food instrument was stolen.

6.4.2.2.6.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.4.2.2.6.2 Edits

This is a not required field.

6.4.2.2.7 StaffStopPay

This is the staff member who requested the stop payment for the food instrument.

6.4.2.2.7.1 Characteristics

This element will be an alphanumeric field.

6.4.2.2.7.2 Edits

This is not a required field.

6.4.2.2.8 ExprtnDate

This is the date the food instrument expires.

6.4.2.2.8.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.4.2.2.8.2 Edits

This is a required field.

6.4.2.2.9 FIModifyUser

This is the staff member who modified the stop payment activity.

6.4.2.2.9.1 Characteristics

This element will be an alphanumeric field.

6.4.2.2.9.2 Edits

This is not a required field.

6.4.2.2.10 FIModifyDate

This is the date the stop payment activity was modified.

6.4.2.2.10.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.4.2.2.10.2 Edits

This is a not required field.

6.4.2.2.11 FIUpdateRec

This is the code indicating to update the Food Instrument Record.

6.4.2.2.11.1 Characteristics

This element will be an alphanumeric field. N - new issuance U- Updated information on existing food instrument.

6.4.2.2.11.2 Edits

This is a required field.

6.5 CSFP New Issuance File Import

The End of Day process will import or export CSFP New Issuance data from an XML file format using SOAP methods. This process is applicable for your state if the State Business Rule EOD_CSFPISUANCEFILEIMPORT.

6.5.1 CSFP New Issuance File Layout

```
<ISSUANCE>
  <FromMachine />
  <ToMachine />
  <Date />
  <Time />
  <Version />
  <SITEINFO>
    <Agency />
    <SrcvSite />
    <CHECKSET>
      <DateIssued />
      <Priority />
      <SubPriority />
      <PrtcptnDate />
      <FirstUseDate />
      <LastUseDate />
      <CompBuy />
      <OvrReason />
      <PkgSize />
      <StampNbr />
      <FrmrsMrktFI />
      <ManualFI />
      <ManualFICode />
      <CHECK>
        <SerialNo />
        <FIType />
        <IssuedCode />
        <PrgrmCode />
        <PharmacyOk />
        <VoidDate />
        <VoidCode />
        <Rebate />
        <ReprintCode />
        <StolenDate />
        <StaffStopPay />
        <OblgtnAmt />
        <PrsntmntAmt />
        <PaidAmt />
        <PaidDate />
        <Reconciled />
        <ExprtnDate />
        <VendorID />
        <StampNbr />
        <PeerGroup />
        <District />
        <ForcePayInd />
      </CHECK>
    </CHECKSET>
  </SITEINFO>
</ISSUANCE>
```

6.5.2 File Elements

6.5.2.1 Issuance

6.5.2.1.1 FromMachine

This is the name of the machine that sent the file. The machine that receives the file will check this element and make sure that the file is from an authorized to send a file.

6.5.2.1.1.1 Characteristics

This element will be an alphanumeric field.

6.5.2.1.1.2 Edits

This is a required field.

6.5.2.1.2 ToMachine

This is the name of the machine to which the file was sent. The machine that receives the file will check this element and make sure that the file is to be sent to the machine.

6.5.2.1.2.1 Characteristics

This element will be an alphanumeric field.

6.5.2.1.2.2 Edits

This is a required field.

6.5.2.1.3 Date

This is the date that the file was build.

6.5.2.1.3.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.5.2.1.3.2 Edits

This is a required field.

6.5.2.1.4 Time

This is the time that the file was build.

6.5.2.1.4.1 Characteristics

This element will be a numeric field. It will have a format of HHMMSS. It will be in a 24 hour format.

6.5.2.1.4.2 Edits

This is a required field.

6.5.2.1.5 Version

This is the file version that is coming in.

6.5.2.1.5.1 Characteristics

This element will be a numeric field. It will have a format of XXX.XX.XX.

6.5.2.1.5.2 Edits

This is a required field.

6.5.2.2 Site Info

6.5.2.2.1 Agency

This is the Agency ID where the food instruments were created.

6.5.2.2.1.1 Characteristics

This element will be a numeric field. It will have a format of XXX

6.5.2.2.1.2 Edits

This is a required field.

6.5.2.2.2 SvcSite

This is the Service Site where the food instruments were created.

6.5.2.2.2.1 Characteristics

This element will be a numeric field. It will have a format of XXX

6.5.2.2.2.2 Edits

This is not a required field. If there is a value in Agency, then this field is required.

6.5.2.3 CheckSet

This is the information printed on the check

6.5.2.3.1 DateIssued

This is the date that the participant's checks were issued

6.5.2.3.1.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.5.2.3.1.2 Edits

This is not a required field.

6.5.2.3.2 Priority

This is the participant's risk priority at the time the food instrument was issued.

6.5.2.3.2.1 Characteristics

This element will be an alphanumeric field.

6.5.2.3.2.2 Edits

This is not a required field.

6.5.2.3.3 SubPriority

This is the participant's risk sub-priority at the time the food instrument was issued.

6.5.2.3.3.1 Characteristics

This element will be an alphanumeric field.

6.5.2.3.3.2 Edits

This is not a required field.

6.5.2.3.4 PrcptnDate

This is the participation date for the participant receiving the food instrument.

6.5.2.3.4.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.5.2.3.4.2 Edits

This is not a required field.

6.5.2.3.5 FirstUseDate

This is the first date the food instrument can be redeemed.

6.5.2.3.5.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.5.2.3.5.2 Edits

This is not a required field.

6.5.2.3.6 LastUseDate

This is the last date the food instrument can be redeemed.

6.5.2.3.6.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.5.2.3.6.2 Edits

This is not a required field.

6.5.2.3.7 CompBuy

This is the code to indicate if this food instrument was issued as a compliance buy.

6.5.2.3.7.1 Characteristics

This element will be an alphanumeric value.

6.5.2.3.7.2 Edits

This is not a required field.

6.5.2.3.8 OvrReason

???

6.5.2.3.8.1 Characteristics

???

6.5.2.3.8.2 Edits

This is not a required field.

6.5.2.3.9 PkgSize

???

6.5.2.3.9.1 Characteristics

???

6.5.2.3.9.2 Edits

This is not a required field.

6.5.2.3.10 StampNbr

This is the stamp number of the Vendor who redeemed the food instrument

6.5.2.3.10.1 Characteristics

This element will be a numeric value.

6.5.2.3.10.2 Edits

This is not a required field.

6.5.2.3.11 FrmrsMrktFI

This is the code indicating if this is a Farmers Market food instrument.

6.5.2.3.11.1 Characteristics

This element will be an alphanumeric value.

6.5.2.3.11.2 Edits

This is not a required field.

6.5.2.3.12 ManualFI

This is a code indicating if the food instrument was paid manually

6.5.2.3.12.1 Characteristics

This element will be an alphanumeric value. It will be one of the following codes: Y – yes or N – No.

6.5.2.3.12.2 Edits

This is not a required field.

6.5.2.3.13 ManualFICode

???

6.5.2.3.13.1 Characteristics

???

6.5.2.3.13.2 Edits

This is not a required field.

6.5.2.4 Check

6.5.2.4.1 SerialNo (Check Number)

This is the check (serial) number of the food instrument.

6.5.2.4.1.1 Characteristics

This element will be a numeric field and could have leading zeros.

6.5.2.4.1.2 Edits

This is a required field.

6.5.2.4.2 FIType

This is the food instrument type code associated with the food instrument.

6.5.2.4.2.1 Characteristics

This element will be an alphanumeric field.

6.5.2.4.2.2 Edits

This is a required field.

6.5.2.4.3 IssuedCode

This is the code indicating what is happening with this food instrument

6.5.2.4.3.1 Characteristics

This element will be an alphanumeric field. It will be one of the following codes: I – Issuance, V - Void.

6.5.2.4.3.2 Edits

This is a not required field.

6.5.2.4.4 PrgmCode

This is the program code indicating the program for which the food instrument was issued.

6.5.2.4.4.1 Characteristics

This element will be a numeric field. It will be one of the following codes: 0 – WIC, 1 – Farmers Market, 2 – Senior’s Farmers Market

6.5.2.4.4.2 Edits

This is a required field.

6.5.2.4.5 PharmacyOk

This is the code indicating if the food instrument can be used at a pharmacy.

6.5.2.4.5.1 Characteristics

This element will be an alphanumeric field. It will be one of the following codes: Y – Yes, N – No.

6.5.2.4.5.2 Edits

This is a required field.

6.5.2.4.6 VoidDate

This is the date the food instrument was voided.

6.5.2.4.6.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.5.2.4.6.2 Edits

This is a not required field.

6.5.2.4.7 VoidCode

This is the code indicating why the food instrument was voided.

6.5.2.4.7.1 Characteristics

This element will be an alphanumeric field.

6.5.2.4.7.2 Edits

This is a not required field.

6.5.2.4.8 Rebate

This will indicate if the food instrument is associated with a rebate item.

6.5.2.4.8.1 Characteristics

This element will be an alphanumeric field.

6.5.2.4.8.2 Edits

This is not a required field.

6.5.2.4.9 ReprintCode

This is the code to indicate if the food instrument was reprinted

6.5.2.4.9.1 Characteristics

This element will be an alphanumeric field.

6.5.2.4.9.2 Edits

This is not a required field.

6.5.2.4.10 StolenDate

This is the date the food instrument was stolen.

6.5.2.4.10.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.5.2.4.10.2 Edits

This is a not required field.

6.5.2.4.11 StaffStopPay

This is the staff member who requested the stop payment for the food instrument.

6.5.2.4.11.1 Characteristics

This element will be an alphanumeric numeric field.

6.5.2.4.11.2 Edits

This is not a required field.

6.5.2.4.12 OblgtnAmt

This is the obligation amount for the food instrument.

6.5.2.4.12.1 Characteristics

This element will be a numeric field.

6.5.2.4.12.2 Edits

This is a required field.

6.5.2.4.13 PrsntmntAmt

This is the presentment amount of the food instrument.

6.5.2.4.13.1 Characteristics

This element will be a numeric field.

6.5.2.4.13.2 Edits

This is a required field.

6.5.2.4.14 PaidAmt

This is the paid amount for the food instrument.

6.5.2.4.14.1 Characteristics

This element will be a numeric field.

6.5.2.4.14.2 Edits

This is not a required field.

6.5.2.4.15 PaidDate

This is the date the food instrument was paid.

6.5.2.4.15.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.5.2.4.15.2 Edits

This is not a required field.

6.5.2.4.16 Reconciled

This is the indicator if the food instrument was reconciled.

6.5.2.4.16.1 Characteristics

This element will be an alphanumeric field

6.5.2.4.16.2 Edits

This is not a required field.

6.5.2.4.17 ExprtnDate

This is the date the food instrument expires.

6.5.2.4.17.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.5.2.4.17.2 Edits

This is a required field.

6.5.2.4.18 VendorID

This is the Vendor who redeemed the food instrument.

6.5.2.4.18.1 Characteristics

This element will be an alphanumeric field.

6.5.2.4.18.2 Edits

This is not a required field.

6.5.2.4.19 StampNbr

This is the stamp number of the Vendor who redeemed the food instrument.

6.5.2.4.19.1 Characteristics

This element will be a numeric field. .

6.5.2.4.19.2 Edits

This is not a required field.

6.5.2.4.20 PeerGroup

This is the peer group code of the vendor who redeemed the food instrument.

6.5.2.4.20.1 Characteristics

This element will be a numeric field. It will have a format of XX.

6.5.2.4.20.2 Edits

This is not a required field.

6.5.2.4.21 District

This is the district of the Vendor who redeemed the food instrument.

6.5.2.4.21.1 Characteristics

This element will be a numeric field. It will have a format of XX.

6.5.2.4.21.2 Edits

This is not a required field.

6.5.2.4.22 ForcePayInd

This is the code indicating if the food instrument was forced paid.

6.5.2.4.22.1 Characteristics

This element will be an alphanumeric field.

6.5.2.4.22.2 Edits

This is not a required field.

6.5.2.5 CheckItem

6.5.2.5.1 CheckItemCode

This is the distribution item code associated with the food item printed on the food instrument.

6.5.2.5.1.1 Characteristics

This element will be a numeric field. It will have a format of XXXXX.

6.5.2.5.1.2 Edits

This is a required field.

6.5.2.5.2 CheckItemQty

This is the quantity of food items printed on the food instrument.

6.5.2.5.2.1 Characteristics

This element will be a numeric field. It will have a format of XX.

6.5.2.5.2.2 Edits

This is a required field.

6.5.2.5.3 CheckItemRX

6.5.2.5.3.1 Characteristics

This element will be a numeric field.

6.5.2.5.3.2 Edits

This is a required field.

6.6 Vendor File Export

This SOAP method will have all the Vendors that are on the State (COVANSYS) system. The sending application will be responsible for building the SOAP file with all the Vendor Information. This processes will be applicable for your state if the State Business Rule EOD_VENDORFILEEXPORT = 'Y'.

6.6.1 Vendor File Layout

```
<VENDORS>
  <FROMMACHINE />
  <TOMACHINE />
  <DATE />
  <TIME />
  <VERSION />
  <VENDORDEMOGRAPHIC>
    <VendorID />
    <TradeName />
    <PhysicalAddress1 />
    <PhysicalAddress2 />
    <PhysicalCity />
    <PhysicalZip />
    <County />
    <Municipality />
    <Telephone />
    <MailingAddress1 />
    <MailingAddress2 />
    <MailingCity />
    <MailingState />
    <MailingZIP />
    <CurrentStatus />
    <LastStatusChangeDate />
  <STAMPINFORMATION>
    <StampIssuanceID />
    <IssuedDate />
    <BankEffectiveDate />
    <DeactivateDate />
  </STAMPINFORMATION>
</VENDORDEMOGRAPHIC>
</VENDORS>
```

6.6.2 File Elements

6.6.2.1 Vendors

6.6.2.1.1 FromMachine

This is the name of the machine that sent the file. The machine that receives the file will check this element and make sure that the file is from an authorized to send a file.

6.6.2.1.1.1 Characteristics

This element will be an alphanumeric field.

6.6.2.1.1.2 Edits

This is a required field.

6.6.2.1.2 ToMachine

This is the name of the machine to which the file was sent. The machine that receives the file will check this element and make sure that the file is to be sent to the machine.

6.6.2.1.2.1 Characteristics

This element will be an alphanumeric field.

6.6.2.1.2.2 Edits

This is a required field.

6.6.2.1.3 Date

This is the date that the file was build.

6.6.2.1.3.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.6.2.1.3.2 Edits

This is a required field.

6.6.2.1.4 Time

This is the time that the file was build.

6.6.2.1.4.1 Characteristics

This element will be a numeric field. It will have a format of HHMMSS. It will be in a 24 hour format.

6.6.2.1.4.2 Edits

This is a required field.

6.6.2.1.5 Version

This is the file version that is coming in.

6.6.2.1.5.1 Characteristics

This element will be a numeric field. It will have a format of XXX.XX.XX.

6.6.2.1.5.2 Edits

This is a required field.

6.6.2.2 VendorDemographic

This will contain the demographic information for all State Vendors. It may have many different records.

6.6.2.2.1 VendorID

This is the code assigned to the vendor.

6.6.2.2.1.1 Characteristics

This element will be an alphanumeric field. It will have a format of WWWWWW where the identifier is zero padded to the left.

6.6.2.2.1.2 Edits

This is a required field.

6.6.2.2.2 TradeName

This is the name of the Vendor's store.

6.6.2.2.2.1 Characteristics

This element will be an alphanumeric field.

6.6.2.2.2.2 Edits

This is a required field.

6.6.2.2.3 PhysicalAddress1

This is the first line of the address of the Vendor's physical address.

6.6.2.2.3.1 Characteristics

This element will be an alphanumeric field.

6.6.2.2.3.2 Edits

This is a required field.

6.6.2.2.4 PhysicalAddress2

This is the second line of the address of the Vendor's physical address.

6.6.2.2.4.1 Characteristics

This element will be an alphanumeric field.

6.6.2.2.4.2 Edits

This is not a required field.

6.6.2.2.5 PhysicalCity

This is the city of the Vendor's physical address.

6.6.2.2.5.1 Characteristics

This element will be an alphanumeric field.

6.6.2.2.5.2 Edits

This is a required field.

6.6.2.2.6 PhysicalZip

This is the zip code of the Vendor's physical address.

6.6.2.2.6.1 Characteristics

This element will be an alphanumeric field. It will have a format of
XXXXXXXXXX

6.6.2.2.6.2 Edits

This is a required field.

6.6.2.2.7 County

This is the county of the Vendor's physical address.

6.6.2.2.7.1 Characteristics

This element will be an alphanumeric field. It will have a format of XX.

6.6.2.2.7.2 Edits

This is a required field.

6.6.2.2.8 Municipality

This is the Legal Municipality of the Vendor's physical address.

6.6.2.2.8.1 Characteristics

This element will be an alphanumeric field. It will have a format of XX.

6.6.2.2.8.2 Edits

This is a required field.

6.6.2.2.9 Telephone

This is the Vendor's Telephone number.

6.6.2.2.9.1 Characteristics

This element will be an alphanumeric field. It will have a format of XXXXXXXXXXXX.

6.6.2.2.9.2 Edits

This is a required field.

6.6.2.2.10 MailingAddress1

This is the first line of the address of the Vendor's mailing address.

6.6.2.2.10.1 Characteristics

This element will be an alphanumeric field.

6.6.2.2.10.2 Edits

This is a required field.

6.6.2.2.11 MailingAddress2

This is the second line of the address of the Vendor's mailing address.

6.6.2.2.11.1 Characteristics

This element will be an alphanumeric field.

6.6.2.2.11.2 Edits

This is not a required field.

6.6.2.2.12 MailingCity

This is the city of the Vendor's mailing address.

6.6.2.2.12.1 Characteristics

This element will be an alphanumeric field.

6.6.2.2.12.2 Edits

This is a required field.

6.6.2.2.13 MailingState

This is the State of the Vendor's mailing address.

6.6.2.2.13.1 Characteristics

This element will be an alphanumeric field. It will have a format of XX.

6.6.2.2.13.2 Edits

This is a required field.

6.6.2.2.14 MailingZip

This is the zip code of the Vendor's mailing address.

6.6.2.2.14.1 Characteristics

This element will be an alphanumeric field. It will have a format of XXXXXXXXX

6.6.2.2.14.2 Edits

This is a required field.

6.6.2.2.15 CurrentStatus

This is the Vendor's current status.

6.6.2.2.15.1 Characteristics

This element will be an alphanumeric field. It will have a format of X. (1 – Applicant, 2 – Waiting List, 3 – Enrolled, 4 –Terminated, 5 – Disqualified, 6 – Denied)

6.6.2.2.15.2 Edits

This is a required field.

6.6.2.2.16 LastStatusChangeDate

This is the date the Vendor's current status was set.

6.6.2.2.16.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.6.2.2.16.2 Edits

This is a required field.

6.6.2.3 Stamp Information

This section will have the Vendor Stamp Information for the Vendor. There can be more than one record for this.

6.6.2.3.1 StampIssuanceID

This is the Vendor's stamp id.

6.6.2.3.1.1 Characteristics

This element will be a numeric field. It will have a format of XXXXXXXX.

6.6.2.3.1.2 Edits

This is a required field.

6.6.2.3.2 IssuedDate

This is the date the Vendor's stamp was issued.

6.6.2.3.2.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.6.2.3.2.2 Edits

This is a required field.

6.6.2.3.3 BankEffectiveDate

This is the date Vendor's stamp id becomes active at the bank.

6.6.2.3.3.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.6.2.3.3.2 Edits

This is a required field.

6.6.2.3.4 DeactivateDate

This is the date Vendor's stamp id becomes deactivated at the State.

6.6.2.3.4.1 Characteristics

This element will be a numeric field. It will have a format of MMDDYYYY.

6.6.2.3.4.2 Edits

This is a required field.

6.7 SOAP Internal Design

This section will explain the System to System integration process and the methods used to accomplish the integration. With the need for integration between the old and new system a non-intrusive method of data transfer is needed. This transfer method will be SOAP – Simple Object Access Protocol.

The following sections will describe how the SOAP protocol will be implemented.

6.7.1 General

The SOAP protocol defines the use of XML (Extensible Markup Language) and HTTP (Hyper Text Transport Protocol) to access services, objects, and servers in a platform-independent manner. The SOAP Protocol is based on a Request /Response pattern. A request is made from the client application and a response is returned by the called method.

The request will contain the method name that will be called to process the data sent with the request. The called method runs and returns a response that will indicate the success or failure of the called method. The client application and called method do not need to reside on the same server but in our implementation they will.

The SOAP request is made up of three components the HTTP header information the SOAP envelope and the SOAP payload. The SOAP envelope and payload are encoded using XML. The SOAP envelope will contain the method name that is being called to process the SOAP payload (there is no special XML identifier for the method name. The SOAP specification indicates that the method name is a positional parameter and that it should immediately follow the SOAP:Body tag). The SOAP payload will contain the data element(s) that will be processed by the called method (see Sections 40.1.1 and 40.2.1 for a list of the data elements).

Example – SOAP Request.

```
POST //HTTP/1.0

Host: 127.0.0.1

SoapMethodName: #soapStringReverse

Content-Length: 292

Content-Type: text/xml

<SOAP:Envelope xmlns:SOAP="urn:schemas-xmlsoap-org:soap.v1"
    SOAP:encodingStyle="urn:schemas-xmlsoap-org:soap.v1">
```

```
<SOAP:Body>
  <p:soapStringReverse xmlns:p='www.pdainc.com/VbXml'>
    <text>Hello World!</text>
  </p:soapStringReverse>
</SOAP:Body>
</SOAP:Envelope>
```

The SOAP response is also made up of three components they are the HTTP response header the response SOAP envelope and the response SOAP payload. The SOAP envelope will contain the method name from the SOAP request (with the word “response” concatenated to the end of the method name) and the SOAP payload will contain the response from the method called from the SOAP request. The SOAP payload response may contain data element(s) or just a simple response regarding the outcome of the called method from the SOAP request. The response should be enclosed by the <Return> and end </Return> tags.

Example – SOAP Response

```
HTTP/1.0 200 OK
Content-Type: text/xml
Content-Length: 275
<SOAP:Envelope xmlns:SOAP='urn:schemas-xmlsoap-org:soap.v1'
  SOAP:encodingStyle = 'urn:schemas-xmlsoap-org:soap.v1'>
  SOAP:Body>
    <p:soapStringReverseResponse xmlns:p='www.pdainc.com/VbXml'>
      <Return>!dlroW olleH</Return>
    </p:soapStringReverseResponse>
  </SOAP:Body>
</SOAP:Envelope>
```

6.7.1.1 Soap File Methods

This section will describe the SOAP file import (coming in to COVANSYS) and file export (going out of COVANSYS) file transfer methods. The SOAP file Import/Export process will be run once a day as part of the End of Day process.

The SOAP file export (Vendor file) will be created and written out as part of the End Of Day process. The Vendor file will take the form of a SOAP Request, once written out it will need to be retrieved and processed by the State to load the States vendor data. Once processed by the State it will need to write a SOAP Response indicating the success or failure of their Vendor data load.

The SOAP file import (Food Instrument file) will be created and written out by the State as a SOAP Request. The End Of Day process will read and process the SOAP Request. The Food Instrument records will be inserted into the COVANSYS database. A SOAP Response indicating the success or failure of the load will be written and will need to be checked by the State before creating the next Food Instrument import file. The Food Instrument import file will contain all the Food Instrument data from the State and will be an update to the data in the COVANSYS database.

The following directory structure on the State Office application server will support these processes.

```
| --SOAP-  
  
  |--FOODINST  
  
    |--ARCHIVE  -archive of previous runs request files  
  
    |--LOG      -log files for Food Instrument imports  
  
    |--REQUEST  -SOAP Food Instrument Request file(s)  
  
    |--RESPONSE -SOAP Food Instrument Response file(s)  
  
  |--VENDOR  
  
    |--ARCHIVE  -archive of previous runs request files  
  
    |--LOG      -log files for Vendor file imports  
  
    |--REQUEST  -SOAP Vendor Request file  
  
    |--RESPONSE -SOAP Vendor Response file
```

6.7.1.2 Soap File Import

This SOAP Method will read and process the SOAP file containing all the Food Instrument records from the State System. This SOAP Food Instrument import file will be created by the State End of Day process and formatted as a SOAP Request. Once processed a SOAP Response will be created and written indicating the status returned from the method called from the SOAP Request. This Process will help guarantee that the State and COVANSYS Food Instrument data remains integrated. The following methods are called to read and process the Food Instrument SOAP import file.

6.7.1.2.1 RetrieveFISoapRequest

This method will retrieve/read the Food Instrument SOAP import file from the SOAP/FOODINST/REQUEST directory (see Section 40.4.2).

The SOAP/FOODINST/REQUEST directory will be read to pick up the latest file for processing (based on the files date and time.) The Food Instrument Soap import file is saved as a string.

If no file is present then processing will not take place and no errors will be generated.

6.7.1.2.2 VerifySoapMethodHeader

This is a SOAP Utility method that is used to verify that the SOAP method name in the HTTP Header portion of the SOAP request matches the method name in the SOAP Envelope. This helps verify that the information in the SOAP HTTP Header and SOAP Envelope belong together.

If the method names do not match then an error will be raised indicating that the "Http Header Method Name was not found in SOAP envelope." And processing is terminated.

6.7.1.2.3 RetrieveSoapBody

This is a SOAP Utility method that is used to strip off the HTTP Header data out of the SOAP Food Instrument import file leaving the SOAP XML envelope and SOAP XML payload.

With just the XML left the file can be loaded into the Document Object Model. The document object model will allow the XML file to be split apart (parsed) programmatically.

6.7.1.2.4 ProcessSoapRequest

- This is a SOAP Utility method that is used to call the other methods to process the Food Instrument SOAP import file. This routine is the main function for processing the SOAP request. The input to this method is the SOAP Envelope from the previous step (See Section 40.4.3.1.3).

6.7.1.2.4.1 VerifySoapVersion

This is a SOAP Utility method that is used to verify that the SOAP request is using a version of SOAP that we can parse. The SOAP Envelope tag is checked to make sure that SOAP.V1 is specified as the SOAP version.

If the SOAP version is not V1 then an error message – “Invalid version not SOAP:V1” is sent and processing is terminated.

6.7.1.2.4.2 RetrieveSoapMethodName

This is a SOAP Utility method that is used to retrieve the method name that is being requested from the SOAP:Envelope. The placement of the method name is positional and based on the Soap specification. The method name is located immediately following the <SOAP:Body> tag. The method name contains the method that will be called to process the SOAP payload data (in this case to insert the Food Instrument records). If the SOAP method name is not found then an error message – “Method name not found in SOAP Envelope” is sent and processing is terminated.

6.7.1.2.4.3 RetrieveSoapObjectName

This is a SOAP Utility method that is used to retrieve the Object name associated with the method name from the SOAP envelope. The SOAP method name is passed into this routine as a parameter and is used to access the SOAP\SOAPXREF.XML file to retrieve the object named associated with the method. The SOAP\SOAPXREF.XML file is parsed to retrieve/read the object name.

The use of the SOAPXREF.XML file allows us to associate an object name to a method name which gives us the ability to have our methods in many different objects. The SOAP cross reference file will also allow us to assign an alias to the method name which will allow us some flexibility if a method name ever changes or we need to associate the alias to a different method. Following is an example of the SOAPXREF.XML file. The ServerName parameter is currently not used. The SOAPXREF.XML file is formatted as XML.

```
<SOAPXREF>  
<SoapMethods>
```

```
<SoapMethod>
  <SoapMethodName>UpdateFIRRecords</SoapMethodName>
  <ObjectName>SoapWrapper.clsSoapWrapper</ObjectName>
  <MethodName>Update_FoodInst_Records_From_Agency</MethodName>
  <ServerName>" "</ServerName>
</SoapMethod>
<SoapMethod>
  <SoapMethodName>UpdateFIRRecords</SoapMethodName>
  <ObjectName>SoapWrapper.clsSoapWrapper</ObjectName>
  <MethodName>Update_FoodInst_Records</MethodName>
  <ServerName>" "</ServerName>
</SoapMethod>
<SoapMethod>
  <SoapMethodName>UpdateVendorRecords</SoapMethodName>
  <ObjectName>????????????????</ObjectName>
  <MethodName>Update_State_Vendors_List</MethodName>
  <ServerName>" "</ServerName>
</SoapMethod>
</SoapMethods>
</SOAPXREF>
```

If the SOAP method name is not found in the SOAPXREF.XML file then an error message – “Object name not found in SOAPXREF file.” is sent and processing is terminated.

6.7.1.2.4.4 *CallMethodByName*

This is the actual method call that will execute the method that was specified in the SOAP Request (see Section 40.4.3.2.1 for a description of the Update_FoodInst_Records_From_Agency method). The results of this call will be the SOAP Response.

By using the SOAPXREF.XML file to associate the method to an object we can use the Visual Basic CallByName function which will allow us to instantiate the object and then call the method without having to know its name in advance. To instantiate the object we use the CreateObject Visual Basic function. If an error is detected while creating the object an error message will be sent "Failed to create method object" and processing will terminate.

The results of the CallByName function is the SOAP Response string which is then wrapped in the outer SOAP:Body Response element which is the completed SOAP Response (see Section 40.4.1). The SOAP Response is returned to the ProcessSoapRequest method.

6.7.1.2.4.5 *ProcessSoapResponse*

This is a SOAP Utility method that is used to process the SOAP Response. This method will parse the SOAP Response to see if a SOAP Fault was thrown. Check what type of result was returned from the backend method that was called from the SOAP Request. And call the necessary functions to process the SOAP Response.

This method will notify the client of the result of the method called by sending messages with the response returned from the method (notifies the client if the run was successful, failed or had errors).

The Response is written out to the SOAP\FOODINST\RESPONSE directory where the State will read the file to verify the status of the Food Instrument load before creating the next Food Instrument SOAP Request file.

6.7.1.2.4.6 Update_FoodInst_Records_From_Agency

This function is used to control the process of replacing all the Food Instrument records. The input to this function is a Food Instrument XML file. The file will contain the SOAP envelope as well as the Food Instrument XML payload.

The XML payload contains the Food Instrument records that will be inserted into the database. The Food Instrument tables are first truncated (deleted) and then the records are inserted. If a record is rejected for any reason it will be written to a log file and a message at the end of the process will be sent stating that errors were detected. It will be up to the State to review the log file and correct any errors.

Once the records have been inserted a backup (archive) of the input to this function is written to the SOAP/FOODINST/ARCHIVE directory for backup purposes.

This function can return one of three possible results.

- Failed – Process was not completed and updates were rolled back
- Errors – Process was completed but errors were detected.
- Success – Process was completed without errors.

The following functions are called to process the file.

- Load the XML file so the data can be parsed.
- Validate_FoodInstrument_XML_File
- Validate_FoodInstrument_Agency
- Truncate_FoodInstrument_Files
- Insert_FoodInstrument_Records
- Archive_FoodInstrument_XML_File

Following is the layout for the Food Instrument XML file minus the

Soap Envelope and HTTP header.

```
<FoodInstruments>
  <FromMachine>string</FromMachine>
  <ToMachine>string</ToMachine>
  <Date>date</Date>
  <Time>time</Time>
  <Version>string</Version>
```

```
<Site_Information>
  <Agency_ID>1234</Agency_ID>
  <Local_Agency>4321</Local_Agency>
  <ServiceSite>1111</ServiceSite>
  <Participant_Info>
    <WICID>1222</WICID>
    <TransactionCode>string</TransactionCode>
    <CurrentWICStatus>string</CurrentWICStatus>
    <CreateUserID>string</CreateUserID>
    <CreateDate>date</CreateDate>
  <FoodInstrumentSet>
    <DateIssued>string</DateIssued>
    <WICStatus>string</WICStatus>
    <ParticipationMonth>string</ParticipationMonth>
    <ExpirationDate>date</ExpirationDate>
    <Priority>string</Priority>
    <SubPriority>string</SubPriority>
    <First_Date_To_Use>date</First_Date_To_Use>
    <Last_Date_To_Use>date</Last_Date_To_Use>
    <ComplianceBuy>boolean</ComplianceBuy>
    <PC_Create_User>string</PC_Create_User>
    <PC_Create_Date>date</PC_Create_Date>
    <PC_Modified_User>string</PC_Modified_User>
    <PC_Modified_Date>date</PC_Modified_Date>
  <FoodInstrument>
    <SerialNumber>string</SerialNumber>
    <StateWICID>string</StateWICID>
    <IssuedCode>string</IssuedCode>
    <Rebate>string</Rebate>
    <PharmacyAllowed>string</PharmacyAllowed>
    <VoidedDate>date</VoidedDate>
    <VoidedReason>string</VoidedReason>
    <ReprintReason>string</ReprintReason>
    <ProgramCode>string</ProgramCode>
    <ReportedStolenDate>date</ReportedStolenDate>
    <StaffIDVoidStolen>string</StaffIDVoidStolen>
    <ObligationAmt>number</ObligationAmt>
    <PresentmentAmt>number</PresentmentAmt>
    <PaidAmt>number</PaidAmt>
    <PaidDate>date</PaidDate>
    <ExpirationDate>date</ExpirationDate>
    <VendorID>string</VendorID>
    <PeerGroupID>string</PeerGroupID>
    <Stampnumber>string</Stampnumber>
    <FoodInstrumentType>string</FoodInstrumentType>
    <CreateUserID>string</CreateUserID>
    <CreateUserDate>date</CreateUserDate>
    <ModifyUserID>string</ModifyUserID>
    <ModifyUserDate>date</ModifyUserDate>
    <UpdateRecord>string</UpdateRecord>
  <Rejections>
    <SerialNo>string</SerialNo>
    <RejectionCode>string</RejectionCode>
    <ModifyDate>date</ModifyDate>
    <ModifyUserID>string</ModifyUserID>
    <RejectionDate>date</RejectionDate>
    <UpdateRecord>string</UpdateRecord>
    <CreateUserID>string</CreateUserID>
    <CreateDate>date</CreateDate>
  </Rejections>
</Site_Information>
```

```
<FoodItem>
  <FoodItemCode>string</FoodItemCode>
  <FoodItemQty>number</FoodItemQty>
  <Food_RX_Item_Code>string</Food_RX_Item_Code>
</FoodItem>
</FoodInstrument>
</FoodInstrumentSet>
</Participant_Info>
</Site_Information>
</FoodInstruments>
```

The following list of edits is performed on the Food Instrument XML data before it is inserted into the Food Instrument tables in the database. In addition to the following edits database referential integrity constraints will also be enforced.

- FoodInstrumentSet – Priority value must be 1 thru 7.
- FoodInstrumentSet – WICSTATUS must be I, C, B, N, P, F.
- FoodInstrument – PaidAmt, PaidDate and VendorID must all be present if any one has a value.
- FoodInstrument – VoidedDate and VoidedReason must both be present if either has a value.
- FoodInstrument – VoidedDate and ReportedStolenDate are mutually exclusive.

6.7.1.3 SOAP File Export

This SOAP method will create and write out a SOAP file that contains all the Vendors that are on the COVANSYS system. This file once created will be read and processed by the State to load the State Vendor data. This process will help guarantee that the State and COVANSYS Vendor data remain integrated. The following methods are called to create the Vendor SOAP export file.

6.7.1.3.1 ValidateVendorSoapResponse

This method will read the Vendor SOAP response file from the States previous run of the load process to validate that the Vendor file was loaded successfully. The Vendor SOAP response file will be read from the SOAP/VENDOR/RESPONSE directory (see Section 40.4.2 above).

If the Vendor SOAP response file indicates that the load of the Vendor data by the State was not successful then the SOAP Vendor File Export process will terminate and an error message will be sent indicating that an error was detected and that the previous runs data did not get processed.

If the Vendor SOAP response file indicates that the load was successful then the following methods will be called to create the new Vendor SOAP file.

6.7.1.3.2 RetrieveVendorData

This method will retrieve the Vendor data from the COVANSYS Oracle table and save it as an XML string. The XML string contains all the Vendor data. All Vendor data in the table is retrieved, no Vendor records are dropped.

The XML element tag names will match the Oracle column names in the database. Following is an example of the XML generated minus the XML schema.

```
<SOAP:Envelope xmlns:SOAP='urn:schemas-xmlsoap-org:soap.v1'  
<SOAP:encodingStyle = 'urn:schemas-xmlsoap-org:soap.v1'>  
<SOAP:Body>  
<p:Update_Vendor_Records xmlns:p='www.pdainc.com'>  
  <VENDORS>  
    <FROMMACHINE>ralph</FROMMACHINE>  
    <TOMACHINE>?????</TOMACHINE>  
    <DATE>08062001</DATE>  
    <TIME>081640</TIME>  
    <VERSION>1.0.0</VERSION>  
    <VENDORDEMOGRAPHIC>  
      <VendorID>009915</VendorID>  
      <TradeName>LIL GENERAL #10</TradeName>  
      <PhysicalAddress1>14-16 BOUND ROAD </PhysicalAddress1>  
      <PhysicalAddress2>LIL GENERAL #10 </PhysicalAddress2>  
      <PhysicalCity>WOONSOCKET</PhysicalCity>  
      <PhysicalZip>028950000</PhysicalZip>  
      <County>004</County>  
      <Municipality>0027</Municipality>  
      <Telephone>4016584490</Telephone>
```

```
<MailingAddress1>14-16 BOUND ROAD </MailingAddress1>
<MailingAddress2>LIL GENERAL #10 </MailingAddress2>
<MailingCity>WOONSOCKET</MailingCity>
<MailingState>RI</MailingState>
<MailingZIP>028950000</MailingZIP>
<CurrentStatus>3</CurrentStatus>
<LastStatusChangeDate>04/16/2002</LastStatusChangeDate>
<STAMPINFORMATION>
  <StampIssuanceID>9915</StampIssuanceID>
  <IssuedDate>06/22/1999</IssuedDate>
  <BankEffectiveDate>06/22/1999</BankEffectiveDate>
</STAMPINFORMATION>
</VENDORDEMOGRAPHIC>
<VENDORDEMOGRAPHIC>
</VENDORDEMOGRAPHIC>
</VENDORS>
</p:Update_Vendor_Records>
</SOAP Body>
</SOAP Envelope>
```