

# **Appendix 2 – Transaction Configuration**

## Table of Contents

<b>TRANSACTION .....</b>	<b>3</b>
DESCRIPTION .....	3
TRANSACTION BUSINESS RULE PACKET AND COSMETICS .....	4
<i>TransactionBusinessRulePacket</i> .....	4
<i>TransactionCosmetics</i> .....	4
<i>EligibleTransactionsByPolicyStatus</i> .....	4
<b>POLICY-FINANCIAL TRANSACTION ELEMENTS .....</b>	<b>5</b>
REQUIRED ELEMENTS .....	5
<i>Transaction Element</i> .....	5
<i>EffectiveDate Element</i> .....	6
COPYBOOK ELEMENT .....	7
ALLOCATION ELEMENTS .....	7
<i>FundAllocation Element</i> .....	8
<i>DefaultAllocation Element</i> .....	9
<i>AllocationFrom Element</i> .....	10
<i>Allocation Element</i> .....	11
SUSPENSE ELEMENTS.....	13
<i>Suspense Element</i> .....	13
<i>MultiSuspense Element</i> .....	14
WITHHOLDING ELEMENTS.....	14
<i>Withholding Element</i> .....	15
VALUATION ELEMENTS .....	15
<i>Valuation Element</i> .....	15
<i>ValuesBlock Element</i> .....	17
<b>FIELDS ELEMENTS.....</b>	<b>18</b>
<i>Field Elements</i> .....	18
ONLOAD/ONCHANGE ELEMENTS .....	22
<i>OnLoad Element</i> .....	23
<i>OnChange Element</i> .....	23
<b>VALIDATION ELEMENTS.....</b>	<b>27</b>
<i>Validation Element</i> .....	27
MATH ELEMENT .....	31
<i>MathVariables Element</i> .....	31
<i>Assignments Element</i> .....	40
<i>ActivityAmounts Element</i> .....	51
<i>Disbursement Element</i> .....	51
SPAWNS ELEMENT .....	55

# Transaction

## Description

Transactions are the blueprint for activities. Transactions are configured to meet business needs including entry, accounting, validations, and valuation. Like Business Rules, configuration consists of Elements, Sub-Elements, Attributes and Values.

There are different types of Transactions within OIPA:

Transaction Type	Definition
Client-Batch	Client level batch transactions
Client-Document	Client level documents
Client-Document-Nonreversible	Client level documents nonreversible
Client-Extract	Client level extracts
Client-Financial	Client level financial transactions
Client-Financial-Nonreversible	Client level financial transactions nonreversible
Client-Import	Client level imports
Cycle-Document	Plan level documents during nightly cycle
Cycle-Extract	Plan level extracts during nightly cycle
Cycle-Import	Plan level imports during nightly cycle
Plan-Batch	Batch entry of activities
Plan-Document	Plan level documents
Plan-Extract	Plan level extracts
Plan-Financial	Plan level financial transactions
Plan-Import	Plan level imports
Policy-Document	Policy level documents
Policy-Extract	Policy level extracts
Policy-Financial	Policy level financial transactions
Policy-Illustration	Illustration transactions
Policy-Financial-Nonreversible	Policy level financial transactions nonreversible
Plan-Financial-Nonreversible	Plan level financial transactions nonreversible
Policy-Document-Nonreversible	Policy level documents nonreversible
Plan-Document-Nonreversible	Plan level documents nonreversible

**Note:** 'Financial' transaction does not necessarily imply that there is a movement of funds. It's simply a designation within OIPA to identify activities that impact data or initiate processing at the Policy or Plan level.

## Transaction Business Rule Packet and Cosmetics

In addition to the configuration done at the Transaction level every transaction requires two additional rules in order to process successfully.

### TransactionBusinessRulePacket

The TransactionBusinessRulePacket controls the order in which the rules attached to a transaction are processed. All rules attached to a Transaction must be listed in the Packet. Exceptions to this are rules that do not require transaction processing for the rule to be invoked (ex. TransactionCosmetics).

### TransactionCosmetics

The XML data of a Transaction Cosmetics rule controls the icon, button and reverse icon for the transaction to which it is associated. Also, a value from the transaction's XML may be displayed on the Activity Screen.

**Note:** Policy Level transactions also require updates to a third business rule.

### EligibleTransactionsByPolicyStatus

The EligibleTransactionsByPolicyStatus is a Plan level override business rule that determines which status(es) a transaction can process under and whether the transaction can be user entered and/or spawned by another transaction. See the Business Rule Configuration section for additional details.

## Policy-Financial Transaction Elements

The following list categorizes the types of elements that can be included in a Policy-Financial transaction. Each possible element is detailed further below.

General Sequence for Transactions:

EffectiveDate  
Allocation  
Suspense  
Withholding  
Valuation  
Fields  
Validation  
OnLoad  
OnChange  
Math  
Spawns.

## Required Elements

Required Elements must be present in the Transaction. Most elements are optional and configured if necessary to meet the Transaction Processing requirements.

## Transaction Element

The Transaction element is the starting tag for each Transaction.

Element\Tag	Attribute\Definition\Value\Data Type	
Yes or No. Yes enables Quote functionality for the activity. The 'Quote' button will be visible on the Activity Detail Screen.	This is the Start and End Element for the Transaction.	
	ALLOWQUOTE	
	ALLOWCONFIRMATION	Yes or No. Yes will invoke the Confirmation Screen Business Rule (which must be overridden at the Transaction level).

## EffectiveDate Element

Every transaction activity has an effective date controlled by this element. It will be the first field displayed when an activity is added and is the one displayed on the Activity Screen. The element can be calculated, defaulted or left for entry without restriction.

Element\Tag	Attribute\Definition\Value\Data Type
<EffectiveDate> </EffectiveDate>	Controls the display and values of the Effective Date field in the transaction. This is a required element for all transactions.
	STATUS
	Enabled. Allows entry of the EffectiveDate field
	Disabled. Restricts entry of the EffectiveDate field
	TYPE
	SYSTEM. Defaults the EffectiveDate to the current system date
	RULE. Defaults the EffectiveDate based on Rule. The tag element defines the rule name. PLAN attribute can also be set to the PlanGUID.
	SQL. Defaults the Effective Date with the result of a Query
	USER.
TITLE	String. Display name of the EffectiveDate field. Optional, if no TITLE is defined the field will display as Effective Date
	PLAN
	See description in TYPE=RULE
METHOD	

Example of Effective Date XML:

```
<EffectiveDate STATUS="Enabled" TYPE="SYSTEM"></EffectiveDate>
<EffectiveDate STATUS="Disabled" TITLE="Effective Date" TYPE="SYSTEM"></EffectiveDate>
  <EffectiveDate STATUS="Enabled" TITLE="Composite/Effective Date" TYPE="SQL">
    SELECT DateValue FROM AsPolicyField WHERE
      FieldName='PolicyEffectiveDate' AND
      PolicyGUID=[PolicyGUID].ActivityGMT DESC</EffectiveDate>
```

## CopyBook Element

Transactions can point to 'CopyBook' business rules. This allows for plans/products to share common functionality. Maintenance is less complicated since the transaction's XML is held in only one place. The CopyBook named will be resolved and the configuration contained it in will be replaced within the Transaction.

Element\Tag	Attribute\Definition\Value\DataType
<CopyBook>	This is the starting and end tag for this element.
</CopyBook>	The Value is the Name of the CopyBook being called. The exact name of the CopyBook must exist in the Business Rule configuration.

Example of a CopyBook XML:

```
<Transaction>
<CopyBook>CopyBook-CopyBookName</CopyBook>
</Transaction>
```

CopyBooks can be used under all elements of transactions. If the entire configuration of a Transaction has been placed in a CopyBook, the Transaction XML will be configured as shown above.

## Allocation Elements

The following elements allow for configuration of Allocation functionality. Configuration of them enables the Allocation Section to appear on the Activity Detail Screen and they control defaults and allowable entries. If this element is present a record will be written to AsAllocation when the activity is saved.

## FundAllocation Element

This element dictates if the Allocation frame will be attached to the Activity Detail Screen. If this element is present and set to **Yes**, the Allocation section will appear on the Activity Detail Screen. Further configuration will determine the Funds that can be selected.

**FundAllocation Element Attribute Table**

Element\Tag	Attribute\Definition\Value\Data Type	
<FundAllocation> </FundAllocation>	This is the Start and End Element. This is a required element for all transactions. Possible Values are Yes & No. If Yes FundAllocation functionality is invoked.	
	ALLOCATIONTYPES	Percent, Amount, Units, Mixed, PercentInFund. Comma separated list.
	MODEL	String or comma-separated list of strings. Name of Model (or AsCodeAllocationType)
	SHOWVALUATION	Yes or No. If Yes Fund Values will be displayed.
	PERCENTPRECISION	Integer. Sets the number of decimal places to be used for allocations with percent method code. Default values are "2" for Amount, "4" for Percent, and "6" for Unit.
	AMOUNTPRECISION	
	UNITPRECISION	
	REPEATFUND	Yes or No. Determines if the same fund can be selected in the From and To sections of a transfer allocation
	ALWAYSEQUALPERCENT	Yes or No. Used in conjunction with Equal Percents. If Yes, system will populate percent (when Equal Percent is checked) only if the result is even percentages across funds, if not they will be blank. If No is configured the percentages will be calculated with any remainder going to the last fund
	EXCLUDEFUNDSTATUS	Code, as defined in AsCodeFundStatus table. Comma separated list. Excludes funds from being displayed based on fund status
	INCLUDEFUNDFIELD	String. Field value defines the fund name.
	INCLUDEFUNDFIELDVALUE	A comma-separated list of fund names which are allowed to be included.
	EXCLUDEFUNDFIELD	String. Field value defines the fund name.
	EXCLUDEFUNDFIELDVALUE	A comma-separated list of fund names which are allowed to be excluded.



	CLASS	String. Defines the class of the fund.
	POLICYFIELD	String. Defines the field name.
	ASSIGNMENT	String. Defines the type of assignment.
	USEFUNDCLASSALLOCATION	Indicates the fund class type code. Also, if defined, indicates fund class allocation to be used.

Example of FundAllocation XML:

```
<FundAllocation ALLOCATIONTYPES="Percent, Amount" MODEL="String" >Yes</FundAllocation>
<FundAllocation PERCENTPRECISION="0" EXCLUDEFUNDSTATUS="01,02,03">Yes</FundAllocation>
```

## DefaultAllocation Element

This element defines the Default Allocation that will be displayed in the Allocation section and whether or not it can be modified. This Element is only applicable if the FundAllocation tag has been configured.

Element\Tag	Attribute\Definition\Value\DataType		
<DefaultAllocation> </DefaultAllocation>	This is the Start and End Element. This is a required element for all transactions. Possible Values are Locked (restricts entry) and Blank (allows default to be changed).		
	TYPE	Default	
		Policy	
		Plan	
		Segment	
		Loan	
		BenefitSplit	
	CODE	CodeValue of the Allocation Type (as defined in AsCode AllocationType)	
DISABLED	Yes or No. Optional. Value only used if element value is blank. Yes is the same as “Locked” element value.		
<Typecodes>	New root tag for typecodes.		
<Typecode>	IF	String – Defines a condition that will execute when true	CodeValue of the Allocation Type (as defined in AsCode AllocationType)

Examples of DefaultAllocation XML:

```
<DefaultAllocation TYPE="Default">Locked</DefaultAllocation>
<DefaultAllocation TYPE="Policy">Locked</DefaultAllocation>
<DefaultAllocation TYPE="Plan" CODE="01">Locked</DefaultAllocation>
```

OR

```
<Transaction>
  <...>
  ...
  <DefaultAllocation TYPE="Plan" CODE="01" DISABLED="Yes">
    <TypeCodes>
      <TypeCode IF="Policy:IssueState &lt;&gt; 37">01</TypeCode>
      <TypeCode IF="Policy:IssueState = 09 And Policy:JointOrSingle =
01">06</TypeCode>
    </TypeCodes>
  </DefaultAllocation>
</Transaction>
```

## AllocationFrom Element

This element allows Allocation configuration development for money movement out of funds. The AllocationFrom element specifies the sources and targets for the AllocationFrom and Allocation 'To' elements. This is an Optional element.

Element\Tag	Attribute\Definition\Value\DataType	
<AllocationFrom> </AllocationFrom>	This is the Start and End tag. This element is used if configuration is different for the 'From' and 'To' funds.	
	See FundAllocation section for details on configurable Attributes.	
	INCLUDEFUNDFIELD	String. Defines the value of the field of fund name to be included.
	INCLUDEFUNDFIELDVALUE	A comma-separated list of fund names which are allowed to be included.
	EXCLUDEFUNDFIELD	String. Defines the value of the field of fund name to be excluded.
	EXCLUDEFUNDFIELDVALUE	A comma-separated list of fund names which are allowed to be excluded.
	EXCLUDEFUNDNAME	Comma-separated list. Defines the funds to exclude from allocations moving money out, based on the name of the fund
	AMOUNTMINIMUM	Validation to determine if the amount has exceeded the minimum amount, based on the required specifications of the transaction.
	FUNDLIMIT	Specify the limit on the number of allocations that can take out money.

	ALLOCATIONDATE	Activity:EffectiveDate	
	ALLOCATIONTYPES	Percent	Define the available distribution format options for Allocations. Comma-separated list.
		Amount	
		Units	
		PercentInFund	
	ALLOWMIXEDMETHODS	Yes	Allows different method codes to be specified for each of the allocations moving money out.
		No	
	MIXEDALLOCATIONTYPES	Percent	Define the available distribution format options when “Mixed” is chosen as an Allocation Type. Comma-separated list.
		Amount	
		Units	
	SHOWVALUATION	Yes	Displays the value of each fund on a policy based on the unit values of previous day.
		No	
	EXCLUDETYPE	Code as defined in AsCodeFundType. Excludes types of funds from being displayed based on fund status.	
	EXCLUDEFUNDSTATUS	Code as defined in AsCodeFundStatus table. Comma-separated list. Excludes funds from being displayed based on fund status.	

Examples of AllocationFrom XML:

```
<AllocationFrom EXCLUDEFUNDNAME="Suspense" AMOUNTMINIMUM="100"></AllocationFrom>
```

```
<AllocationFrom ALLOWMIXEDMETHODS="No" FUNDLIMIT="01"
```

```
ALLOCATIONTYPES="Amount"></AllocationFrom>
```

```
<AllocationFrom . . . . . ALLOWMIXEDMETHODS="Yes" ALLOCATIONTYPES="Percent,Amount,Units"
```

```
MIXEDALLOCATIONTYPES="Percent,Amount,Units" . . .></AllocationFrom>
```

## Allocation Element

The Allocation element allows configuration development to specify settings for allocations moving money into funds. This is an Optional element.

Element\Tag	Attribute\Definition\Value\DataType	
<Allocation> </Allocation>	This is the Start and End tag. This element is used if configuration is different for the 'From' and 'To' funds.  See FundAllocation section for details on configurable Attributes.	
	FUNDLIMIT	Specify the limit on the number of allocations that can take out money.
	EXCLUDETYPE	Allows funds to be excluded from the allocations that are moving money into a fund, based on the type of the fund.
	INCLUDEFUNDFIELD	String. Defines the value of the field of fund name to be included.
	INCLUDEFUNDFIELDVALUE	A comma-separated list of fund names which are allowed to be included.
	EXCLUDEFUNDFIELD	String. Defines the value of the field of fund name to be excluded.
	EXCLUDEFUNDFIELDVALUE	A comma-separated list of fund names which are allowed to be included.
	EXCLUDEFUNDNAME	Allows listing fund names not to have money moved into. These Fund names are separated by a comma.
	ALLOCATIONTYPES	Percent
		Amount
		Units
		PercentInFund
		Mixed
	ALLOWMIXEDMETHODS	Yes
		No
	MIXEDALLOCATIONTYPES	Percent
		Amount
		Units
	EXCLUDEFUNDSTATUSERROR	Yes
		No
		Process

			section
			No - will not show the closed funds in the list nor will it show an error (as it cannot happen in this case)
			Process - will show the funds in the list but no error will be displayed in this case.
	EXCLUDEFUNDSTATUS	Yes	Excludes funds from being displayed based on fund status.
		No	

Examples of Allocation XML:

```
<Allocation EXCLUDETYPE = "01,02,05" EXCLUDEFUNDNAME = "Suspense,AIMFund"></Allocation>
<Allocation ALLOWMIXEDMETHODS = "YES" FUNDLIMIT="18"> </Allocation>
<Allocation EXCLUDEFUNDSTATUS = "01,02,03"></Allocation>
<Allocation . . . ALLOWMIXEDMETHODS="Yes" ALLOCATIONTYPES="Percent,Units"
MIXEDALLOCATIONTYPES="Percent,Units" . .></Allocation>
```

## Suspense Elements

The following Elements allow for configuration of Suspense functionality. Configuration of them enables the Suspense functionality. Depending on the element configured either a Suspense field or Suspense Section will appear on the Activity Detail Screen.

If a Suspense element is present in the Transaction, a suspense record will be written in AsSuspense when the activity processes.

## Suspense Element

Displays and controls the suspense field inside the activity. With this element present, a suspense field will be in the Activity entry screen.

Element\Tag	Attribute\Definition\Value\DataType	
<Suspense> </Suspense>	This is the Start and End tag. This element is used multiple suspense items can be attached to the Activity. A String is used to identify the ActivityField the Suspense items must total.	
	OVERRIDABLE	If set to "Ignore", errors are ignored.
	VALUE	Equals. The Suspense record must total the Activity Field exactly
		Sufficient. The Suspense record(s) must be at least the value of the Activity Field (can be greater)

	AUTOENTRY	Yes or No. If Yes, a Suspense record will be created if the entry field is left blank. If No, the field must be entered or an edit validation will prohibit activity processing.
	ALTERNATE	String. Allows for a link to other suspense systems if no valid suspense number is given.

Example of Suspense XML:

```
<Suspense OVERRIDABLE="Yes" VALUE="Sufficient" >GrossAmount</Suspense>
```

```
<Suspense OVERRIDABLE="No" AUTOENTRY="Yes" ALTERNATE="RequestID">Amount</Suspense>
```

## MultiSuspense Element

This element is placed in a transaction to include multiple suspense items in an activity. This element overrides the <suspense> element. When included, this element adds a sub-window to the bottom of the Activity Screen.

Element\Tag	Attribute\Definition\Value\Data Type	
<MultiSuspense> </MultiSuspense>	This is the Start and End tag. This element is used multiple suspense items can be attached to the Activity. Possible values are Yes/No. If Yes MultiSuspense is enabled.	
	START	Integer or a SQL query that returns an integer. Minimum number of suspense items. Default
	STOP	Integer or a SQL query that returns an integer. Maximum number of suspense items
	FIELD	ActivityField that Suspense Records must total.

Example of MultiSuspense XML Data

```
<MultiSuspense START="0" STOP="3" FIELD="BonusAmount">Yes</MultiSuspense>
```

## Withholding Elements

The following Element allows for configuration of the Withholding functionality. If present the withholding field (link to the Withholding table) will be present in the Activity Detail Screen and a record will be written to AsWithholding when the activity is saved.

## Withholding Element

Displays and controls the withholding detail inside the activity.

Element\Tag	Attribute\Definition\Value\DataType
<Withholding> </Withholding>	This is the Start and End tag. If this element is used, a Withholding field will appear on the Activity Detail Screen.  Yes No Activity

Example of Withholding XML:

```
<Withholding>No</Withholding>
<Withholding>Yes</Withholding>
<Withholding>Activity</Withholding>
```

## Valuation Elements

The Valuation elements initiate the Valuation functionality. A valuation will be run for a policy prior to the activity being processed. The valuation information can be used during activity processing.

### Valuation Element

If included, the Valuation engine will be called and a record is written to AsValuation XML. The values will be PRIOR to the activity processing. This element should be included if the Assignment Element is included.

Element\Tag	Attribute\Definition\Value\DataType
<Valuation> </Valuation>	This is the Start and End tag. If this element is used a Withholding field will appear on the Activity Detail Screen.
	EFFECTIVEDATE String. As of Date for the Valuation. Overrides the date set in the EffectiveDate tag.
	POLICYVALUES Yes or No. If yes, the PolicyValues Business Rule will be run when Valuation runs. The default is Yes.
<EffectiveDateNUVMustExist>	Yes or No. If Yes, NUVs as of the activity effective date must be present in order for the transaction to process. If product only entails Fixed Fund this value is set to No.
<SystemDateNUVMustExist>	Yes or No. If Yes, NUVs as of the system effective date must be present in order for the transaction to process. If product only entails Fixed Fund this value is set to No.

Example of Valuation XML:

```
<Valuation EFFECTIVEDATE=" ValuationDate">
  <EffectiveDateNUVMustExist>No</EffectiveDateNUVMustExist>
    <SystemDateNUVMustExist>No</SystemDateNUVMustExist>
</Valuation>
<Valuation POLICYVALUES="Yes">
  <EffectiveDateNUVMustExist>Yes</EffectiveDateNUVMustExist>
    <SystemDateNUVMustExist>No</SystemDateNUVMustExist>
</Valuation>
```



## ValuesBlock Element

The ValuesBlock element creates a section on the Activity Detail Screen that displays the current policy fund values. The section is display only.

Element\Tag		Attribute\Definition\Value\Data Type	
<ValuesBlock> </ValuesBlock>		This is the Start and End tag. This tag will cause the Values Block section display on the Activity Detail screen.	
<FundDetails>	This node can have multiple occurrences. The first node to which the Expression results in true or the Expression attribute does not exist will be used to render the values block.	EXPRESSION	Text. If the Expression results in true then use this funds details. This expression has access to the policy fields/columns. Any valid math expression.
		TYPE	Text. Defines the type of values shown in the values block. Accumulation Payout.

Example of ValueBlock XML:

```
<ValuesBlock>
  <FundDetails TYPE="Accumulation" EXPRESSION="Policy:StatusCode" != "39"></FundDetails>
  <FundDetails TYPE="Payout" EXPRESSION="Policy:StatusCode" = "39"></FundDetails>
</ValuesBlock>
```

## Fields Elements

The following elements allows for configuration of the Fields. The Fields will dictate the entry information required for activity processing. The same Field configuration is also used in Screen Business Rules.

### Field Elements

Displays and controls the entry fields inside the activity.

Element\Tag	Attribute\Definition\Value\DataType
<Fields>	Represents on input control that will accept input from the user or the system.
<Field>	Indicates the start of the values that define and apply to one element.
<Name>	String. This name will be recognized by the business rule. ID of field that will be used to access its data in other parts of the system. No spaces allowed
<Group>	String. This is the name of the group that should be used to obtain the value.
	ROLECODE      The role code for the group.
<Display>	String. Text value that will be used as display of the field data when on screen. Spaces are allowed.
	MASK      This defines the mask of the field.
<DataType>	Defines the format of the data and functionality.
	Check box field
	Combo – Combination/drop down field
	Date - Date field with calendar icon
	Decimal – Displays decimal point and calculator icon
	Integer – Formats to whole number. Displays calculator icon
	Line - Displays line across frame for aesthetics
	Money – Displays dollar sign, commas, decimals and Calculator icon
	Percent – Displays percent sign
	Radio – Two mutually exclusive options. Selecting one automatically unselects the other
	Text – Free form entry
	Blank
	Address
	Identifier
	Comment – displays a comment text box which enables a comment to be entered for the transaction. Data entered will be stored as a text field in the fields database for the page that it was added (i.e. PolicyField for Policy Screen, ActivityField if entered on a

	Transaction), Data is also stored to AsComment.	
<Encrypt>	Yes or No. If Yes, field name is added to encryption list.	
	ERASE	Yes or No. If Yes, field name is added to erase encryption list.
<ByStateApproval>	For some screens, the field will only be displayed if the state approval value here matches the state approval setup for the screen.	
<Calculated>	Allows comparison functionality for transaction and evaluating policy data.	
	TYPE	SQL
		FUNCTION
	FROM	Defines the field to format data.
<Query>	WHERE	Defines database source and policy data for comparison.
	Used with the Combo DataType to retrieve options for the combo box. The Query must return a Code Value and Text description.	
<MultiFields>	TYPE	SQL
		Uses SQL to fill the options.
		Uses Options tag to fill the options.
<Options>	String. Element value defines the multiple fields to be used. RULE attribute sets the multi-field rule.	
<OptionValue>	Allows you to specify "Hard Coded" values to be selected for Combo or Radio DataTypes. Used only with FIXED or RADIO types.	
<OptionText>	The Value of the option.	
<Disabled>	The label of the option.	
<Hidden>	Allows the field to be display only/grayed out. If set to "Yes", entry of the field is prohibited. Values are Yes, No, or Readonly (if Readonly field is disabled but font is not grayed out)	
<Length>	Allows values to be carried in ActivityXML but not displayed. Values are Yes or No, if yes, hides the field from the user if set to "Yes".	
<Tag>	Integer. Applicable for Integer and Text type fields. Limits character entry	
<Calculated>	String.	
	Applicable for Integer and Text type fields. Limits character entry. Calculated elements overrides Default activities.	
	TYPE	EVAL- Allows you to evaluate the Calculated Element by determining if the value is a number, operator, or string, then evaluating the expression. It allows you to use the numbers, operators, and fields to evaluate a value.

		FUNCTION - Function allows you to execute a Function	
		REPLACE -Replaces one substring with another substring the specified number of times	
		RULE - Rule will look up the Rule listed	
		SQL -SQL allows for a SQL Query to be executed with the result being the date displayed in the field.	
		SYSTEM – Results in the system date being displayed in the field.	
	METHOD	IFEMPTY - Used to check if a roles address is empty, it is used in conjunction with a sql statement to retrieve the correct address to fill in the field	
		FORCE - For Fields the FORCE attribute is used to force the field to have the calculated value	
PLAN	Used to set the PlanGUID.		
<ToolTip>	String. Provides the user with information regarding the field. Information is viewed when they mouseover Field on Activity Detail screen.		
<DefaultValue>	String, Code or Integer (depending on DataType). Default Value of Field. If set to SYSTEMDATE, the default value will be set to the system date.		
<Value>	The Value to display in the field.		
<Parts>	String. Provides information to create a next identifier.		
	TYPE	VALUE	Element value of tag set to be the value.
		SYSTEMDATE	System date, in format defined by FORMAT, is set to be the value.
		SEQUENCE	Element value is the name of sequence to use. SEQUENCEDATE is used to get sequence date value.
		FIELD	Element value is the name of the field.
	FORMAT		Format of the system date or sequence value.
	LEFT		Positive integer. Used with TYPE=FIELD. Defines the left attribute. Only relevant if MID and RIGHT are not set.
	MID		Positive integer. Used with TYPE=FIELD. Defines the middle attribute. Only relevant if RIGHT is not set.

	RIGHT	Positive integer. Used with TYPE=FIELD. Defines the right attribute.
	SEQUENCEDATE	Date used in SEQUENCE type.
<RoleAssignments>	MINIMUM	Integer value. Minimum role value.
	MAXIMUM	Integer value. Maximum role value.
	CLONEROLE	Yes or No. If Yes, the roles are to be cloned and not set.
<RoleAssignment>	Sets the segment type code and role code.	
<ShowSegmentTypeCode>	The element value is set to segment type code.	
<ShowRoleCode>	The element value us set to role code.	
<InputFocus>	Yes or No. If Yes, the focus is put on the input.	
<FixedFields>	Used exactly as <Fields> tag only FixedFields is not dynamic and does not create a new fielddcl.	

Example of Field XML:

#### General Example

```
<Field>
  <Name>Adjustment</Name>
  <Display>Adjustment</Display>
  <DataType>Money</DataType>
  <DefaultValue>0</DefaultValue>
  <Disabled>No</Disabled>
  <Hidden>No</Hidden>
  <Length>15</Length>
</Field>
```

#### Line DataType Example

```
<Field>
  <Name>Line1</Name>
  <Display/>
  <DataType>Line</DataType>
</Field>
```

#### Radio DataType/OptionText population Example

```
<Field>
  <Name>AccountType</Name>
  <Display>Account Type</Display>
  <DataType>Radio</DataType>
  <Query TYPE="RADIO">
    <Options>
```

```

    <Option>
      <OptionValue>C</OptionValue>
      <OptionText>Checking</OptionText>
    </Option>
    <Option>
      <OptionValue>S</OptionValue>
      <OptionText>Savings</OptionText>
    </Option>
  </Options>
</Query>
</Field>

```

#### Combo DataType/SQL population Example

```

<Field>
  <Name>BankStateLocation</Name>
  <Display>Bank State</Display>
  <DataType>Combo</DataType>
  <Query TYPE="SQL">SELECT CodeValue, ShortDescription FROM AsCodeState
    UNION SELECT '$$$Blank$$$', '' ORDER BY ShortDescription</Query>
</Field>

```

#### Element/Mask example

```

<Field>
  <Name>MemberPhone</Name>
  <Display>Phone</Display>
  <DataType>Text</DataType>
  <Element>Mask:Phone</Element>
</Field>

```

## OnLoad/OnChange Elements

The OnChange element helps with processing when a field has changed and will affect the contents of another field. OnChange configuration allows you to change one or more fields based on the value of another field, aka, the trigger field. An OnChange can not be triggered until an entry has been made into a field. This means that when a screen is loaded no entries have been made. If a field is present in the 'OnLoad' section the system treats it as an entered value and therefore invokes the OnChange functionality prior to a physical entry.

**Note:** All field names referenced in the OnLoad and OnChange must be prefixed with that field's data type. The following table contains the list of prefixes recognized by the system:

Data Type	Prefix
-----------	--------

Check	Chk
Radio	Rb
Combo	cmb
All other	txt

## OnLoad Element

This element is applicable when you have OnChange processing. It contains a list of field name(s) that execute (i.e., trigger) the OnChange when the page is loaded.

Element\Tag	Attribute\Definition\Value\Data Type
<OnLoad>	The starting and tag used to enable this functionality.
<Field>	String. Name of field preceded by it's Datatype (see table below)

Example of OnLoad XML:

```
<OnLoad>
  <Field>cmbAutoInvestmentFeature</Field>
  <Field>rbAssetAllocation</Field>
</OnLoad>
```

## OnChange Element

The OnChange element helps with processing when a field has changed and will affect the contents of another field. OnChange configuration allows you to change one or more fields based on the value of another field, aka, the trigger field. OnChange processing can be used in Business Rules that contain entry fields (ex. PolicyScreen) or Segment as well as Transactions.

The OnChange Element helps with processing when a field has changed and will affect the contents of another field. OnChange processing can be used in Business Rules that contain entry fields (ex. PolicyScreen) or Segment as well as transactions.

```
<OnChange>
  <Change>
    <Name>Field Name</Name> (Note: Field Name should be the exact HTML Name.)
    <Type>Check|Combo|Percent</Type>
    (if Type=Check)
    <Checked>
      <Calculated TYPE="EVAL">JavaScript</Calculated>
    </Checked>
    <Unchecked>
      <Calculated TYPE="EVAL">JavaScript</Calculated>
    </Unchecked>
  </Change>
</OnChange>
```

```

        (if Type=Combo)
        <Items>
            <Item INDEX="0-~">
                <Calculated TYPE="EVAL">JavaScript</Calculated>
            </Item>
        </Items>
        (if Type=Percent)
        <IfBlank>JavaScript</IfBlank>
        <Calculated TYPE="EVAL">JavaScript</Calculated>
        (if Type=Compare)
        <Expression>JavaScript</Expression>
        <IfTrue></IfTrue>
        <IfFalse></IfFalse>
    </Change>
</OnChange>

```

#### Name

The Name section of the Change definition indicates the Field/Value that will trigger the OnChange event. The name of the field will be prefaced with the data type abbreviation (see DataType/Field Value below). The applicable data types are Combo, Check and Radio (other data type fields currently cannot be 'trigger' fields).

#### Type

The Type field indicates the data type of the 'trigger' field.

#### Value Option Elements

The elements used to indicate the value of the 'trigger' field are dependent on the data type of the 'trigger' field. See sample above for available elements for each data type.

#### Calculated Element

The Calculated statement is made up of the HTML form name, the data type/name of the field, and the action that should occur based on the value selected in the trigger field.

#### HTML Form Name

```
<Calculated TYPE="EVAL">document.frmS3Policy.cmbMinorAnnuitant.disabled = true;</Calculated>
```



Screen Name	HTML Form Name
Transaction Detail	document.frmAsActivityDetail
Policy Screen	document.frmS3Policy
Segment Screen	document.frmS3Segment
Plan Detail	document.frmAsPlanDetail
Role Detail	document.frmS3RoleDetail
AddressScreen	document.frmS3Client
PolicyOverviewScreen	document.frmS3PolicyOverview
Client Detail	document.frmS3ClientDetail

## Data Type/Field Name

Whenever the field name is referenced in the OnChange section the data type prefix must precede it.

```

                                <Change>
                                <Name>cmbField1</Name>
<Type>Combo</Type>
<Items>
  <Item INDEX="0">
    <CalculatedTYPE="EVAL">document.frmAsActivityDetail.txtField2.disabled=true;</Calculated>...

```

In the above example when the first option is selected from the combo box in the Field1 field, the Field2 field should become disabled. The Field1 field is a combo box and the Field2 field is a text data type. All data types can be referenced in the Calculated element.

If the DataType is:

**Check** - prefaced with 'chk' above example would be chkField1

**Radio** - prefaced with 'rb' above example would be rbField1

**Combo** - prefaced with 'cmb' above example would be cmbField1

**All other datatypes** are prefaced with 'txt' (fields with icons next to them Calendar and Calculator also have 'dis' for display as well as a 'txt' name). If the OnChange requirements include setting the value of the field both a 'txt' and a 'dis' Calculated element are needed.

## Value Options

The data type of the field determines the applicable values.

```
<CalculatedTYPE="EVAL">document.frmS3Policy.cmbMinorAnnuitant.disabled = true;</Calculated>
```

Command	Values	ApplicableData Types
disabled	True (field becomes disabled) False (field becomes enabled)	All
style.backgroundColor	#E4E4E4 (gray) #FFFFFF (no background color)	All
selectedIndex	0 - ~	Combo Box
Value	Any - both a txt and a dis value should be indicated	Text, Money, Integer
checked	True (Checked) False (Unchecked)	Check

**Note:** OnChange for Checkboxes only happens when the checkbox loses "focus" (when you go to another field)

## Hidden Fields

Fields can be hidden using the OnChange feature by Declaring:

```
Document.all['tblS3SegmentFields'].rows[#].style.display = 'none'
```

Conversly if the field is hidden then there must be an option to show the field which will be declared:

```
Document.all['tblS3SegmentFields'].rows[#].style.display = 'block'
```

## Validation Elements

These elements are used to validate activity entry fields. Validations can only occur on Field and Allocation entry values.

### Validation Element

Provides online page error messages regarding Field and Allocation entries via JavaScript.

Element\Tag	Attribute\Definition\Value\DataType			
<Validation>	The starting and tag used to enable this functionality. Provides online page error message via Java Script. Limited to the variables available on the page/in the Transaction.			
<Required>	List the Fields that require entry			
<Field>	Element value is the name of field that must be entered. Note: name can be wrapped in <Name> tag.			
<OneOf>	List the Fields that there should be only one of.			
<Field>	Element value is the name of field that there should only be one of.			
<Allocation>	Describes requirements for allocations (receiving/positive)			
<Percent>	Minimum Percent that must be allocated	ERRORMESSAGE	String	Error Message that will be displayed if Validation fails
		SCOPE	Total	Total of all Funds in allocation
			Individual	Validation is done against each Fund  If SCOPE Attribute is not present then validation is against Allocation Total
		OPERATION	Equal GreaterThan LessThan GreaterThanOrEqual LessThanOrEqual RANGE (Integer, Integer)  If OPERATION Attribute is not present then validation is for the	

			exact value present in the <Percent> tag (Example 1 below)
<Amount>	Minimum Amount that must be allocated. Resolves to a Field Value	ERRORMESSAGE	Same as above
		SCOPE	Same as above
		OPERATION	Same as above
<Units>	Minimum Units that must be allocated in Transfer.	ERRORMESSAGE	Same as above
		SCOPE	
		OPERATION	
<Prorata>	Minimum Prorata that must be allocated in Transfer.	ERRORMESSAGE	Same as above
		SCOPE	
		OPERATION	
<PercentInFund>	Minimum Percent in Fund that must be allocated in Transfer.	ERRORMESSAGE	Same as above
		SCOPE	
		OPERATION	
<Transfer>	Used only for 'Transfer' transactions. Allows you to specify different validations for To and From Allocations	FROMMETHODCODE	Amount Percent Mixed
		TOMETHODCODE	Amount Percent Mixed
<Percent>	Minimum Percent that must be allocated in Transfer	ERRORMESSAGE	Same as above
		SCOPE	
		OPERATION	
<Amount>	Minimum Amount that must be allocated in Transfer	ERRORMESSAGE	Same as above
		SCOPE	
		OPERATION	

<Units>	Minimum Units that must be allocated in Transfer	ERRORMESSAGE	Same as above	
		SCOPE		
		OPERATION		
<Prorata>	Minimum Prorata that must be allocated in Transfer	ERRORMESSAGE	Same as above	
		SCOPE		
		OPERATION		
<PercentInFund>	Minimum Percent In Fund that must be allocated in Transfer.	ERRORMESSAGE	Same as above	
		SCOPE		
		OPERATION		
<AllocationFrom>	Describes requirements for allocations from. Uses same elements as Allocation tag. See above.			
<Expressions>	Groups the math and expressions tags	ROLECODE	String. The expressions section will only be executed for these role codes.	Any role code from 01 to 99
		ACTION	String. The expressions section will only be executed for these actions.	Save New Delete
<Math>	Groups MathVariable tags			
<MathVariable>	String. Creates a math variable that will be interpreted by the math engine. Value is directly impacted by the variable type.			
	VariableName	String. Name of the math variable.		
	TYPE	String. Tells the math engine how to evaluate the value of the maths variable value.	Value Expression Field Sql Function XML NumericArray Array	
<Expression>	MESSAGE	String. The value of this attribute will be displayed to the user if the criterion is met	Any String	

	TYPE	String. Tells the math engine to error if the expression is evaluated to be true or false.	ErrorOnTrue ErrorOnFalse
<PolicyRoles>	List the Policy Roles.		
<Role>	Contains role code for policy role.		

#### Example 1 of Validation XML

```
<Validation>
  <Allocation>
    <Percent> 100</Percent>
    <Amount>5000</Amount>
  </Allocation>
  <Expressions>.
    <Expression MESSAGE="Routing number incorrect, check digit does not
      match">ValidCheckDigit(document.frmS3Client.txtRoutingNumber.value) ==
      true</Expression>
  </Expressions>
</Validation>
```

#### Example 2 of Validation XML:

```
<Validation>
  <Amount SCOPE="Total" OPERATION="GreaterThan"></Amount>
  <Percent SCOPE="Individual" OPERATION="Equal"></Percent>
  <Transfer FROMMETHODCODE="Amount" TOMETHODCODE="Percent">
    <Amount SCOPE="Total" OPERATION="GreaterThan"></Amount>
    <Percent SCOPE="Individual" OPERATION="Equal"></Percent>
  </Transfer>
</Validation>
```

## Math Element

Within the Math Element, calculations can be evaluated using variables. Variable names are assigned as the value of the attribute "VARIABLENAME". Additionally, the <Math> element may optionally contain the elements <Disbursement>, <Assignment>, <ActivityAmounts> and <PostAssignmentMath>.

## MathVariables Element

The OIPA Math Engine is an XML stream that defines variables and expressions that drive business functionality. Math variables can be initialized with constant strings and numbers, values from another business rule, or a function that is located in a special business rule called JavaScriptingCode. Once these variables are initialized, they can be used in expressions that drive other variables in the functionality. Several types can define these expressions.

TYPE		AGGREGATEFUNCTION		
		Method Name	Element Value	Result
Parameters	METHOD	MIN	The name of a previously defined array	The smallest number in the given array
		MAX	The name of a previously defined array	The largest number in the given array
		SUM	The name of a previously defined array	The sum of all numbers in the given array
		COUNT	The name of a previously defined array	The size of the array
		INDEX	The name of a previously defined array	The value at the specified index. The INDEX attribute must contain an array index or variable whose value is an array index or FIRST or LAST

TYPE		COMMUTATION
Parameters	INDEX	Index is the lower bound of the array that will populate the math variable.
	ENDINDEX	EndIndex is the upper bound of the array that will populate the math variable. Using the same value for INDEX and ENDINDEX will mean that only one value populates the array but it will still be an array.
	SOURCEARRAY	This attribute tells the commutation variable where to retrieve the factors it needs. The QARRAY is provided as a math variable. It is a list of Cost Of Insurance rates derived from a rate lookup
Element Value		RawQ, RawQ12, Q, Dx12, Nx12, Cx12, Mx12, Dx, Nx
Function(s)	BuildCommutationCarrier	BuildCommutationCarrier is called in the case when the commutation values for this transaction have not already been created. The commutation values are dependent on policy and plan related values and must be created at runtime based on instructions in the transaction definition. When the first TYPE="COMMUTATION" math variable is encountered by the math processor it will check to see if these values exist. If not, it will call BuildCommutationCarrier and populate a variant array called vCommutationCarrier. This variant carries the factors (Element Values) referred to above. <b>BuildRawQ, BuildRawQ12, BuildQ, BuildDx12, BuildNx12, BuildCx12, BuildMx12, BuildDx, BuildNx</b> are all called by <b>BuildCommutationCarrier</b> in order. There values can only be calculated in order. That is, the calculations in <b>BuildRawQ12</b> are dependent on the results of <b>BuildRawQ</b> , the first called. <b>BuildRawQ</b> is dependent on an array of "Cost Of Insurance (Q)" values specified in the transaction definition. This Cost Of Insurance array is itself created through the massaging of rates provided by a lookup inside the transaction definition: ExecuteCommutation.
Result		Returns the Factor (Element Value) from the SOURCEARRAY defined.
Note		



TYPE		COLLECTION + KEY
Element Value		SQL Statement used to return a collection of values. Used in conjunction with the COLLECTIONVALUE Element.
Result		The first column returned from the SQL statement will contain the "KEY" value and the second column the associated value
Note		Must be used in conjunction with COLLECTIONVALUE Element

TYPE		COLLECTIONVALUE
Element Value		The value of the element is the KEY value of the COLLECTION Element
Result		Looks up the associated COLLECTION Value
Note		Must be used in conjunction with COLLECTIONVALUE Element
KEY		
KEYFIELD		String. Math variable to define key use and to access the key in the collection

TYPE		EXPRESSION
Element Value		Expression
Result		Evaluation of the Expression
Note		

TYPE		FIELD
Element Value		<b>1.1.1.1.1.1.1.1 VariableName</b>
Result		Value of the <i>VariableName</i>
Note		The Field type allows you to pull in a value from the Fields section of the transaction as well as available XML and table values from Valuation, Activity and Policy. See Appendix 2 – Field Type Values for complete list.

TYPE		FUNCTIONCALL
Element Value		Function and parameters; parameters can be numeric or a <i>VariableName</i>
Result		The return value from the function
Note		The functions are defined in the business rule named JavaScriptingCode

TYPE		IIF
Parameters	EXPRESSION	Expression
Element Value		Expression
Result		Expression1, Expression2
Note		If the <i>Expression</i> in EXPRESSION attribute evaluates to true, the <i>Expression1</i> (evaluated) is the result; if false, <i>Expression2</i> (evaluated) is the result.

TYPE		LOOKUP-IMPORT-RATE
Parameters	NAME	String constant corresponding to the RateDescription column in AsRate
	NAMEFIELD	<i>VariableName</i> that provides the string constant corresponding to the RateDescription column in AsRate
	ROW	<i>VariableName</i> that provides the numeric value corresponding to the 'Row' Column in AsRate.
	IF	Expression – If Expression evaluates to false, the value is set empty.
	METHOD	Previous-Date, Next-Date, see Element Value
Element Value		DateCriteria, Criteria1, Criteria2, Criteria3, Criteria4, Criteria5, Criteria6, Criteria7, Criteria8, Criteria9, Criteria10. Contains <i>VariableNames</i> that provide string values corresponding to the columns in AsRate. If a match is not desired for a certain Criteria, Null is used. The DateCriteria does have to match the record. When there is no record in AsRate with a direct match to DateCriteria, the METHOD attribute can specify whether to pick the nearest previous date or the next date.
Result		The value from the rate column of AsRate that was picked using the various Criteria
Note		NAMEFIELD has precedence over NAME

TYPE		LOOKUP-RATE
Parameters	RULE	A business rule name. The business rule must be of type Table File and have Excel files uploaded. This upload would populate AsUploadRate.
	RULEFIELD	<i>VariableName</i> that provides the name of the intended business rule
	ROW	<i>VariableName</i> that provides the numeric value corresponding to the Row column in AsUploadRate
Element Value		Criteria1, Criteria2, Criteria3, Criteria4, Criteria5. Contains <i>VariableNames</i> that provide the string values corresponding to the columns in AsUploadRate. If a match is not desired for a certain Criteria, Null is used.
Result		The value from the Rate column of AsRate that was picked using the various criteria

Note		RULEFIELD has precedence over RULE
------	--	------------------------------------

TYPE		LOOKUP-XML (Backward compatibility only – Do not Use)
Element Value		LookupPremiumTax, LookupMortalityRate
Result		Rate from Rate XML
Note		Loads either PremiumTaxTable or MortalityTable rules, which provide an UploadGUID. Using the Upload GUID, the rate XML is loaded. The Rate XML, Policy's statecode, and Qualifier is then used to load the rate.

TYPE		NUMERICARRAY		
		Operation Name	Element Value	Result
Parameters	OPERATION	FILLBY_LIST	Comma separated list of numeric constants and/or variable names	Fills the array with the values entered
		FILLBY-SQL	SQL select statement that selects numeric values from a single column from the database	Fills the array with the values returned from SQL
		INSERTITEMS	Name of a variable containing another NUMERICARRAY. Use the INDEX attribute to specify the location at which to insert the new elements. INDEX can contain a number or variable whose value is an array index or FIRST or LAST. To append the elements to the end of the array, leave out the INDEX attribute. INSERTITEM	Inserts the elements of another array into the current array
		INSERTITEM	Value for the OPERATION attribute on MathVariable elements where the TYPE='NUMERICARRAY'.  <MathVariable VARIABLENAME="array" TYPE="NUMERICARRAY" OPERATION="INSERTITEM" INDEX="5">389.45</MathVariable>	Allows you to insert a single element into any position of an array

			<p>This line will insert the value 389.45 into the array named "array" at index 5.</p> <p>You can also use INDEX="FIRST" to insert at the beginning.</p>	
		REMOVEITEMS	Semi-colon separated list of ranges indicating which elements to remove. Each range is either a single value (fixed or variable) or two values separated by a comma.	Removes the specified elements from the array
		TRANSFORM	Any valid expression. If the array elements are to be updated by using their current value in the calculation, use the name of the array in place of that value. If the value is just the name of another array variable, then that array is copied.	Performs the indicated calculation where one or more of the variables used in the expression are arrays
		ARRAYFUNCTION	Any valid function call. Valid functions are defined in the JavaScriptingCode business rule.	Executes a function in a manner similar to a FUNCTIONCALL variable except that it operates on each element of any arrays in the expression.
		EXPAND	Use the MULTIPLIER attribute to indicate the number of times each element of the original array should be duplicated. The first and last elements may have a different number of duplicates than the other elements, so this	Creates a new array by duplicating each element of the original array a given number of times.

			attribute's value can be a comma delimited list of, at most, 3 values. If only one value is supplied, the others are defaulted to that value.	
--	--	--	---	--

TYPE		PLANFIELD
Element Value		Name of a PlanField
Result		Value of PlanField
Note		

TYPE		POLICYFIELD
Element Value		Name of a PolicyField
Result		Value of PolicyField
Note		

TYPE		PROCESS
Parameters	OBJECT	ProgID of the object that's method is invoked
Element Value		Method name that is invoked
Result		The return value of the method
Note		

TYPE		RANGE
Element Value		Required attribute: FIELD – evaluates to a number
Result		The value of this element is a comma separated list of floating point numbers
Note		Used to pick a number from a list based on the value of another number. The result of the calculation is the first number from the list that is greater than or equal to the value of the Field attribute.

TYPE		RECORD
Element Value		The value of this element is an arbitrary SQL select statement
Result		The first record of the result set is kept and referred to by a variable of type RECORDINDEX
Note		Must be used in conjunction with RECORDINDEX Element

TYPE		RECORDINDEX
Element Value		refers to the index of the RECORD array. Each array element is the value of one column from the select statement
Result		The value of the element is the name of the RECORD type variable
Note		Must be used in conjunction with RECORD Element

TYPE		RULE
Element Value		A business rule name
Result		The content of the element with the business rule name
Note		This rule can have only one element. <i>&lt;RuleName&gt;value&lt;/RuleName&gt;</i>

TYPE		SEGMENTFIELD
Element Value		Name of a SegmentField
Result		Value of Policy XML Field
Note		

TYPE		SOLVEFOR
Element Value		Benefit/Cost
Result		The result of the XPath /Segment/SolveFor from the SegmentXML
Note		The SolveFor element in the Segment XML stores the result of the SolveFor field on the segment screen.

TYPE		SQL
Element Value		A SQL query. Must be a SELECT clause.
Result		The result of the SQL query. The first column of the first row that was fetched.
Note		The SQL query can have dynamic items within its construct. These items take the form of [VariableName]. In this case, VariableName can also include PolicyGUID and SegmentGUID, apart from elements that have appeared before this element.

TYPE		SUBSTITUTE
Element Value		A string with arguments enclosed with [ and ]
Result		The same string with the arguments resolved
Note		

TYPE		SUMXMLFIELD
Element Value		XPATH defines the XPath location of the desired number value in each XML Stream
Result		Sums the values of items read from a set of XML streams as

		determined by an SQL select statement
Note		The value of the element is the select statement that reads the XML streams from the database

<b>TYPE</b>		<b>SYSTEMDATE</b>
Element Value		Date / Time
Result		The AdminServer system date
Note		This is not the same as the machine date.

<b>TYPE</b>		<b>VALUATION</b>
Element Value		
Result		Valuation XML
Note		The valuation array converted to an XML

<b>TYPE</b>		<b>VALUE</b>
Element Value		A string or numeric constant
Result		The string or numeric within the element
Note		

<b>TYPE</b>		<b>VB</b>
Element Value		Function Name
Result		Value of the function called
Note		Executes a javascript script function

TYPE		XML
Parameters	XPATH	A valid XPATH
Element Value		The <i>VariableName</i> whose value is in an XML stream
Result		The result of applying the XPATH on the XML stream

## Assignments Element

The Assignment Element, located in the Transaction XML, describes how values should be assigned to the policy and written to AsValuation.

Type	AdjustedFullWithdrawal	
Allocations needed?	No	
Parameters	ADJUSTEDMONEYTYPE	Note: This value is set but never used.
	SURRENDERCHARGE	Yes, No, or a field that holds 'Yes' or 'No', whether or not a surrender charge is applied.
	SURRENDERMONEYTYPE	The type of money for the surrender charge.
Element Value	One or more "MoneyType" tags that define the amount and type of money to be removed.	
Note	Assignment fields must be negative (removals). Allocation records are created by the system with 'PercentInAllocation' set to the ratio of the removal value to the fund cash value. If there are no cash values in all funds, the removal is split equally on all funds. Add adjustments if the fund goes negative. An attribute named ADJUSTMENTMONEYTYPE is required. Surrender and MVA charges can be applied.	
Used when	When fund/account values could go negative	
Examples	FreeLook w/ Gain	

Type	Apply	
Allocations needed?	Yes	
Parameters	BUCKET	The current bucket. Must be an integer value.
	RATELOCKDATE	Date of rate lock.
Element Value	One or more "MoneyType" tags that define the amount and type of money to be applied.	
Note	Assignment fields can be both negative (removals) and positive (additions). Removals are not validated against fund values. Hence, additions should be more than removals. The allocation records should be positive and must match total money-in amount/units or 100%.	
Used when	To add Premium	



Examples	Premium
	Premium Bonus

Type	ApplyByFund	
Allocations needed?	No	
Parameters	BUCKET	The current bucket. Must be a integer value.
	RATELOCKDATE	Date of rate lock
Element Value	One or more "MoneyType" tags.that define the amount and type of money to be applied. Each moneytype must have a FUND attribute set to a FundGUID.	
Note	Assignment fields can be both negative (removals) and positive (additions). Removals are not validated against fund values. Hence, additions should be more than removals. The allocation records should be positive and must match total money-in amount/units or 100%. This allows the apply to be fund specific	
Used when	To direct deposits to specific funds	
Examples	LoanRepayment	

Type	ApplyFixed	
Allocations Needed?	Yes	
Parameters	RATELOCKDATE	Date of rate lock
Element Value	One or more "MoneyType" tags.that define the amount and type of money to be applied. Each moneytype must have a FUND attribute set to a FundGUID.	
Note	Assignment fields can be both negative (removals) and positive (additions). Removals are not validated against fund values. The allocation records should be positive and must match total money-in amount/units or 100%. ApplyFixed is similar to Apply, except that the Policy is not valued when this activity takes place (<Valuation>...</Valuation> is not provided in the transaction definition).	
Used when	Valuation is used purely for accounting/billing purposes	
Examples	DuplicateCertificate	

Type	ApplyUnits	
Allocations needed?	Yes	
Parameters	MONEYTYPE	Required. Date of rate lock.
Element Value	None.	
Note	Assignment fields are not allowed. An attribute named MONEYTYPE is required at the <Assignment> element level. The Allocation records should hold AllocationUnits. The Valuation record gets the ValuationUnits from the AllocationUnits column. The ValuationAmount column is left to be NULL.	
Used when	Policies are converted	

Examples	ConvertPremium
----------	----------------

Type	AutomaticTransfer	
Allocations needed?	Yes	
Parameters	SURRENDERCHARGE	
	SURRENDERMONEYTYPE	
	MVACHARGE	Yes, No, or a field that holds 'Yes' or 'No'. Indicates whether there is an MVA Charge.
	MVAMONEYTYPE	The MVA money type.
	REDEMPTIONFEE	Yes or No. Indicates whether there is a Redemption Fee.
	REDEMPTIONMONEYTYPE	The Redemption Fee money type.
	BUCKET	The bucket for the EIFund. Must be an integer value.
	PRORATEREMOVALS	Prorate the amount to remove.
Element Value	Zero or more "MoneyType" tags that define the amount and type of money to be applied.	
Note	Since the cash value/units of a policy is not definite, only 'Percent' allocation type should be used. Allocations should be positive and total to 100%. Negative Allocation records for the existing funds are created by the system. No assignment fields are required.	
Used when	All existing money is moved to other funds.	
Examples	AutomaticTransfer	

Type	Billing	
Allocations needed?	No	
Parameters	None	
Element Value	One or more "MoneyType" tags that define the amount and type of money to be applied. Each moneytype must have a FUND attribute set to a FundGUID.	
Note	Billing is only valid for plan level transactions. Assignment fields should have a FUND attribute providing the GUID of the fund affected. The valuation amount is summed for each policy in the plan and a valuation record with the FundGUID provided and summed amount multiplied by -1 is inserted. The total sum for the plan is placed in the Math Variable.	
Used when	Billing type transactions are needed	
Examples	Group Billing	

Type	DepositRollover
------	-----------------

Allocations needed?	Yes
Parameters	None
Element Value	One “Rollover” tag or “MoneyType” tag named “Rollover”. All other tags are ignored. A “DEPOSIT” attribute in the tag gives the deposit GUID for the deposit that is to be rolled over.
Note	An attribute called DEPOSIT provides the information on the deposit that has to be rolled over. There can be only one element and its name should be Rollover. All the money from the deposit will be transferred to other funds based on the allocation records.
Used when	Deposit is rolled over
Examples	DCA Rollover

Type	FullWithdrawal	
Allocations needed?	No	
Parameters	SURRENDERCHARGE	Yes, No, or a field that holds ‘Yes’ or ‘No’. Whether or not a surrender charge is applied.
	SURRENDERMONEYTYPE	The type of money for the surrender charge.
	MVACHARGE	Yes, No, or a field that holds ‘Yes’ or ‘No’. Indicates whether there is an MVA charge.
	MVAMONEYTYPE	The MVA money type.
	REDEMPTIONFEE	Yes or No. Indicates whether there is a Redemption Fee.
	REDEMPTIONMONEYTYPE	The Redemption Fee money type.
Parameters	BUCKET	The bucket for the EIFund. Must be an integer value.
Element Value	One or more “MoneyType” tags that define the amount and type of money to be applied.	
Note	Assignment fields must be negative (removals). Allocation records are created by the system with ‘PercentInAllocation’ set to the ratio of the fund value to the policy cash value.	
Used when	Total Money removal	
Examples	Surrender	

Type	FundAssignment
Allocations needed?	Yes

Parameters	RATELOCKDATE	Date of rate lock.
	MONEYTYPE	The type of money to be applied in the form of the field name.
	MONEYTYPECODE	Only used if MONEYTYPE is missing or numeric. Contains the code of the type of money to be applied.
	SURRENDERCHARGE	Yes, No, or a field that holds 'Yes' or 'No'. Whether or not a surrender charge is applied.
	SURRENDERMONEYTYPE	The type of money for the surrender charge.
	MVACHARGE	Yes, No, or a field that holds 'Yes' or 'No'. Indicates whether there is an MVA Charge.
	MVAMONEYTYPE	The MVA money type.
	REDEMPTIONFEE	Yes or No. Indicates whether there is a Redemption Fee.
	REDEMPTIONMONEYTYPE	The Redemption Fee money type.
	BUCKET	The bucket for the EIFund. Must be an integer value.
Element Value	None	
Note	Amounts applied to the funds are provided by the allocation records. Assignment elements are not allowed for this type.	
Used when	Conversion sync up	
Examples	ConvertPolicy	

Type	GuaranteedFundAdjustment
Allocations needed?	No
Parameters	None.
Element Value	One or more "MoneyType" tags that define the amount and type of money to be applied. Each moneytype must have a FUND attribute set to a FundGUID.
Note	Assignment fields can be both negative (removals) and positive (additions). Each assignment field is specific to a fund. The FundGUID must be place in an attribute - FUND. Since each assignment is applied on a single fund, the allocation records are created by the system with 'PercentInAllocation' set to one. Although the name suggests a fund adjustment, there is no adjustment done in assignment processing. A business rule, WriteBenefitSplitAssignment can be attached to the Transaction to calculate the amount/type of assignments
Used when	Money-out/in items specific to a fund
Examples	AnnuityDisbursement

Type	GrossWithdrawal	
Allocations needed?	Yes	
Parameters	SURRENDERCHARGE	Yes, No, or a field that holds 'Yes' or 'No'. Whether or not a surrender charge is applied.
	SURRENDERMONEYTYPE	The type of money for the surrender charge.
	MVACHARGE	Yes, No, or a field that holds 'Yes' or 'No'. Indicates whether there is an MVA Charge.
	MVAMONEYTYPE	The MVA money type.
	REDEMPTIONFEE	Yes or No. Indicates whether there is a Redemption Fee.
	REDEMPTIONMONEYTYPE	The Redemption Fee money type.
	BUCKET	The bucket for the EIFund. Must be an integer value.
Element Value	One or more "MoneyType" tags that define the amount and type of money to be applied. The tags can have an attribute named NETFIELD.	
Note	Assignment fields must be negative (removal). The allocation records should be positive and must match total money-out amount/units or 100%. This is similar to PartialWithdrawal, but the charges are applied within the removal amount.	
Used when	Unscheduled money-out	
Examples	Withdrawal	
	Systematic Withdrawal	

Type	GrossFullWithdrawal	
Allocations needed?	No	
Parameters	EXCLUDEFUNDTYPES	Comma-separated list of fund types. Allows funds to be excluded from the allocations that are moving money into a fund, based on the type of fund.
	SURRENDERCHARGE	Yes, No, or a field that holds 'Yes' or 'No'. Whether or not a surrender charge is applied.
	SURRENDERMONEYTYPE	The type of money for the surrender charge.
	MVACHARGE	Yes, No, or a field that holds 'Yes' or 'No'. Indicates whether there is an MVA Charge.
	MVAMONEYTYPE	The MVA money type.
	REDEMPTIONFEE	Yes or No. Indicates whether there is a Redemption Fee.
	REDEMPTIONMONEYTYPE	The Redemption Fee money type.
	BUCKET	The bucket for the EIFund. Must be an integer value.
Element Value	One or more "MoneyType" tags that define the amount and type of money to be applied. The tags can have an attribute named NETFIELD.	
Note	Assignment fields must be negative (removals). Allocation records are created by the system with PercentInAllocation set to the ratio of the fund value to the policy cash value %. This is similar to PartialWithdrawal, but the charges are applied within the removal amount.	
Used when	Total Money-out	
Examples	FreeLook	

Type	GrossFullWithdrawalWithAllocations	
Allocations needed?	Yes	
Parameters	SURRENDERCHARGE	Yes, No, or a field that holds 'Yes' or 'No'. Whether or not a surrender charge is applied.
	SURRENDERMONEYTYPE	The type of money for the surrender charge.
	MVACHARGE	Yes, No, or a field that holds 'Yes' or 'No'. Indicates whether there is an MVA

		Charge.
	MVAMONEYTYPE	The MVA money type.
	REDEMPTIONFEE	Yes or No. Indicate whether there is a Redemption Fee.
	REDEMPTIONMONEYTYPE	The Redemption Fee money type.
	BUCKET	The bucket for the EIFund. Must be an integer value.
Element Value	Assignment fields must be negative (removal). The allocation records should be positive and must match total money-out amount/units or 100%. This is similar to PartialWithdrawal, but the charges are applied within the removal amount.	
Note	Assignment fields must be negative (removals). Allocations on the activity at the time of assignment are used to reduce the contract value first. Any remaining amount that must be assigned is distributed among the remaining funds on the contract at a ratio of the fund value to the total remaining fund value %. This is similar to GrossFullWithdrawal.	
Used when	Total Money-out	
Examples	FreeLook	

Type	InterestTransfer	
Allocations needed?	Yes	
Parameters	FUNDTYPE	Comma-separated list of fund type codes whose interest will be moved to the finds listed in the allocation records.
Element Value	None	
Note	Allocations entered are positive. Negative allocations are generated by the system. The attribute FUNDTYPE will list comma delimited fund types who's interest will be moved to the funds listed in the allocation records.	
Used when	Interest Transferred	
Examples		

Type	Loan	
Allocations needed?		
Parameters	RATELOCKDATE	Date of rate lock.
	MONEYTYPE	The type of money to be applied in the form of the field name.
	MVACHARGE	Yes, No, or a field that holds 'Yes' or 'No'. Indicates whether there is an MVA Charge.

	MVAMONEYTYPE	The MVA money type.
	COLLATERALFUND	The GUID for the Collateral Fund.
	LOANAMOUNT	The loan amount.
Element Value	One or more "MoneyType" tags that define the amount and type of money to be applied.	
Note		
Used when		
Examples		

Type	PartialWithdrawal	
Allocations needed?	Yes	
Parameters	SURRENDERCHARGE	Yes, No, or a field that holds 'Yes' or 'No'. Whether or not a surrender charge is applied.
	SURRENDERMONEYTYPE	The type of money for the surrender charge.
	MVACHARGE	Yes, No, or a field that holds 'Yes' or 'No'. Indicates whether there is an MVA Charge.
	MVAMONEYTYPE	The MVA money type.
	REDEMPTIONFEE	Yes or No. Indicates whether there is a Redemption Fee.
	REDEMPTIONMONEYTYPE	The Redemption Fee money type.
	BUCKET	The bucket for the EIFund. Must be an integer value.
Element Value	One or more "MoneyType" tags that define the amount and type of money to be applied. The tags can have an attribute named NETFIELD.	
Note	Assignment fields must be negative (removal). The allocation records should be positive and must match total money-out amount/units or 100%.	
Used when	Unscheduled money-out	
Examples	Withdrawal	
	Systematic Withdrawal	

Type	SplitDepositApply	
Allocations needed?	Yes	
Parameters	SPLITDEPOSITREMOVALGUID	Activity GUID for split deposit removal.
	SPLITDEPOSITAPPLYMONEYTYPE	Money type for split deposit.
	SURRENDERCHARGE	Yes, No, or a field that holds 'Yes' or 'No'.



		Whether or not a surrender charge is applied.
	SURRENDERMONEYTYPE	The type of money for the surrender charge.
	MVACHARGE	Yes, No, or a field that holds 'Yes' or 'No'. Indicates whether there is an MVA Charge.
	MVAMONEYTYPE	The MVA money type.
	REDEMPTIONFEE	Yes or No. Indicates whether there is a Redemption Fee.
	REDEMPTIONMONEYTYPE	The Redemption Fee money type.
	BUCKET	The bucket for the EIFund. Must be an integer value.
Element Value	None	
Note	Yes For each removal associated with an activity, create a corresponding deposit on the policy.	
Used when	Applying deposits that have been removed to the new policy	
Examples	NewContract	

Type	SplitDepositRemoval	
Allocations needed?	Yes	
Parameters	TOTALPRIORPERCENTAGE	Total percentage removed from policy to date under SplitDepositRemoval.
	SURRENDERCHARGE	Yes, No, or a field that holds 'Yes' or 'No'. Whether or not a surrender charge is applied.
	SURRENDERMONEYTYPE	The type of money for the surrender charge.
	MVACHARGE	Yes, No, or a field that holds 'Yes' or 'No'. Indicates whether there is an MVA Charge.
	MVAMONEYTYPE	The MVA money type.
	REDEMPTIONFEE	Yes or No. Indicates whether there is a Redemption Fee.
	REDEMPTIONMONEYTYPE	The Redemption Fee money type.
	BUCKET	The bucket for the EIFund. Must be an integer value.
Element Value	One or more "MoneyType" tags that define the amount and type of money to be applied. The tag can have an attribute named NETFIELD.	
Note	This assignment allows users to remove a percentage or all of the deposits made in	

	each fund.
Used when	Removing a portion of the deposits so that they can later be applied to a new policy
Examples	NewContract

Type	Transfer, Rebalance	
Allocations needed?	Yes	
Parameters	PRORATEREMOVALS	Prorate the amount to remove.
	SURRENDERCHARGE	Yes, No, or a field that holds 'Yes' or 'No'. Whether or not a surrender charge is applied.
	SURRENDERMONEYTYPE	The type of money for the surrender charge.
	MVACHARGE	Yes, No, or a field that holds 'Yes' or 'No'. Indicates whether there is an MVA Charge.
	MVAMONEYTYPE	The MVA money type.
	REDEMPTIONFEE	Yes or No. Indicates whether there is a Redemption Fee.
	REDEMPTIONMONEYTYPE	The Redemption Fee money type.
	BUCKET	The bucket for the EIFund. Must be an integer value.
Element Value	Zero or more "MoneyType" tags that define the amount and type of money to be applied.	
Note	Assignment fields are not needed. Allocation records are needed in both negative and positive.	
Used when	Money is moved from a set of one or more funds to another.	
Examples	Transfer	
	RebalanceAssets	

Type	TransferDeposit	
Allocations needed?	Yes	
Parameters	DEPOSIT	The deposit GUID or a field that hold the deposit GUID.
Element Value	None	
Note	Assignment fields not needed. Allocation records are needed in both negative and positive. Only money from the deposit identified by the DEPOSIT attribute is transferred. Surrender and MVA charges apply	
Used when	Money is moved from a set of one or more funds to another.	
Examples	TransferDeposit Used for DCA, one fund/deposit going to multiple	

	TransferDeposit
--	-----------------

Type	WithdrawalByUnits	
Allocations needed?	Yes	
Parameters	MONEYTYPE	The type of money to be applied in the form of the field name.
Element Value	None	
Note		
Used when		
Examples		

### Example of Assignment XML

```
<Assignment TYPE="GrossFullWithdrawal">
    <NegativeRemovalAmount>32</NegativeRemovalAmount>
</Assignment>
```

## ActivityAmounts Element

The ActivityAmount element generates accounting based on values passed in. There must be a corresponding accounting record set up in Chart of Accounts for this Transaction/ActivityAmount.

Element\Tag	Attribute\Definition\Value\DataType
<ActivityAmounts> </ ActivityAmounts >	The starting and tag used to enable this functionality.
<ActivityAmount>	String. Variable defining field value that will generate accounting records when activity processes.

### Example of ActivityAmounts XML

```
<ActivityAmounts>
    <ActivityAmount>RemovalAmount</ActivityAmount>
    <ActivityAmount>InterestRemoved</ActivityAmount>
    <ActivityAmount>ExceptionAmount</ActivityAmount>
</ActivityAmounts>
```

## Disbursement Element

The Disbursement Element triggers the creation of the AsDisbursementRecord.

Element\Tag	Attribute\Definition\Value\DataType		
<Disbursement> </Disbursement>	The starting and tag used to enable this functionality.		
	OVERRIDABLE		Yes or No
	AUTOENTRY		Yes or No
	METHOD	Determines the method used to process the disbursements	Unbalanced, Balanced  Unbalanced – outstanding disbursement adjustment exists as a result of reversal/recycle processing.  Balanced – reversal/recycle processing did not result in any disbursement adjustment.
	APPROVAL		Yes, No, or Variance  Yes – indicates that all pending disbursements will be displayed in the Disbursement Approval screen  No – indicates that Disbursement Approval will not be used  Variance – indicates that only disbursements generated from a redo or reversal of an activity will be displayed in the Disbursement Approval screen
	CheckEFTInformation		Yes or No.  Yes, processing is determined by CheckEFTInformation Business Rule. No, If Attribute is not present, No is assumed. Default is No.
	DisbursementType		String. (as defined in AsCodeDisbursementMethod). Variable defining the Disbursement type
	APPROVALSTATUS		String. Approval status code. Approved, Disapproved, or

		Unapproved. From a math variable.
	USEROLEPERCENT	Yes or No. Defines whether to use role percent.
	RECOVERABLE	Yes or No. Defines whether this is recoverable. Default is No.
	TAXABLEAMOUNT	Amount that is taxable. Default is the disbursement amount.
<AddressDisplay>	Element value	Defines the field containing the address of the disbursement
<WithholdingFields>	String. Variable defining Federal or State Withholding amounts. Comma separated list.	
<DisbursementRole>	Identifies the disbursement payee. Variable defining RoleGUID	
	Element Value	The RoleGUID or RoleCode to use for disbursements. In addition, can contain the address GUID field name, separated from RoleGUID by a comma.
	TYPE	
	DISBURSEMENTAMOUNT	FinalDisbursement. Variable defining the disbursement amount
<DisbursementFields> </DisbursementFields>		
<DisbursementField> </DisbursementField>		
<From>	Sets the From value of the Disbursement Field. Required.	
<To>	Sets the To value of the Disbursement Field. Required.	
<DataType>	Sets the Data type value of the Disbursement Field. Required.	

### Examples of Disbursement XML

```

<Transaction>
  <EffectiveDate STATUS="Enabled"
    TITLE="Effective Date"
    TYPE="SYSTEM"></EffectiveDate>
  <Math>
    <Disbursement OVERRIDABLE="No"
      AUTOENTRY="Yes"
      METHOD="[Balanced|Unbalanced]"
      APPROVAL="[Yes|No|Variance]">
    </Disbursement>
  </Math>
</Transaction>

```

or

```
<Disbursement CheckEFTInformation="Yes" DisbursementType="DisbursementMethod">  
  <WithholdingFields>FederalWithholding,StateWithholding</WithholdingFields>  
  <DisbursementRole TYPE="GUID"  
DISBURSEMENTAMOUNT="FinalDisbursement">RoleGUID</DisbursementRole>  
</Disbursement>
```

## Spawns Element

The Spawns element generates subsequent transactions based on defined criteria. Includes the values of the fields in the activity it is spawning.

Element\Tag	Attribute\Definition\Value\Data Type		
<Spawns> </Spawns>	Indicates the opening and closing tags to define elements that enable spawn transaction functionality.		
<Spawn>	IF	String. Condition when true initiates the spawn. If not present Spawn will generate under all conditions.	
<Transaction>	String. Name of the transaction to be spawned.		
	SPAWNCODE	Code (as defined in AsCode SpawnCode). Required.	Text
	SPAWNONREVERSAL	Yes or No. If Yes – supports the spawning of activities during the “Undo” process. If No – does not allow spawning of activities during the “Undo” process.	
	FIELD	String. Variable defining the EffectiveDate of the Spawn. Only applicable for SpawnCodes 03, 08 and 10.	
	CLIENT	A Client GUID. Not applicable for SpawnCodes 03, 08 and 10.	
	CLIENTLISTQUERY	A query with results being a single Client GUID. Not applicable for SpawnCodes 06, 08, 09 or 10.	
	SQL	A query with the result being a policy GUID. Applicable when the SpawnCode is 09.	
	SCHEDULEGUID	String. This attribute will populate the ScheduleGuid column in the AsActivity Table when spawning a new activity.	Math Variable name
<Suspense>	A single suspect ticket to be spawned.		
	TYPE	GUID or Number. Indicates whether the suspense is based on Suspense Guld or Suspense Number.	
<MultiSuspense>	Indicates this tag spawns multiple Suspense tickets to a Transaction with MultiSuspense and initializes multiple suspense items in the activity. Not applicable when SpawnCode is set to 06 or 08.		
	COLLECTION	Defines the collection of	Math

			SuspenseGUIDs and the SpecifiedAmounts to be spawned.	variable name
<SpawnFields>	MULTIFIELD	Yes	Spawn the current activity's Multifields to the spawned activity.	Text
		No	Starting tag which lists the SpawnFields	
<SpawnField>	Indicates the opening of Field information passed from the spawned by and spawned activities.			
<From>	Defines the value of the Activity field or MathVariable passed to Spawn.			
<To>	Defines value of the Spawned Activity field to be populated.			
<DataType>	Defines the value of the DataType for the spawned Activity Field.			
<Allocation>	TYPE		Plan or Policy	

### Example of Spawns XML

```

<Spawns>
  <Spawn>
    <Transaction SPAWNCODE="03" FIELD="NextCycleDate">
      DetailAccountingEntryReportByAccount</Transaction>
    <SpawnFields>
      <SpawnField>
        <From>GroupByField</From>
        <To>GroupBy</To>
        <DataType>Text</DataType>
      </SpawnField>
      <SpawnField>
        <From>NextCycleDate</From>
        <To>FromDate</To>
        <DataType>Date</DataType>
      </SpawnField>
    <Allocation TYPE="Policy">02</Allocation>
  </Spawn>
</Spawns>

```