

# **Appendix 4 – File Rule Configuration**

## Table of Contents

<b>File Rule Configuration Overview .....</b>	<b>3</b>
Description .....	3
File Element\Attributes .....	3
File Configuration Screen Image.....	4
XML Examples .....	5

# File Rule Configuration Overview

## Description

This File Rule (also referred to as AsFile) allows configuration to translate incoming data and insert it directly into the OIPA database. AsFile can also leverage its configurability to selectively insert records, data and activities based on the state of a particular request.

The configuration for each File has two sections: XMLData and XSLT.

The XMLData processes the incoming request by assigning values to variables that will be used for processing, when transformed by the XSLT.

## File Element\Attributes

When creating a new File, a FileID must be specified. This allows the caller to specify which File to use to process the request. A FileID is limited to three characters and should be an acronym or abbreviation of the calling or desired process.

The FileNameFormat field allows the user to give the new interface a name, similar to the naming of a business rule or transaction within the system.

The <File> element must be included as the first tag in the section.

ImportMapName and ClassName are two required attributes. The import name is the text of the root XML node of the AsXML. The ClassName attribute should be defined as one of the objects that AsXML mapping is going to create.

Pre and Post Insert operations are specified in the XmlData portion of a File's configuration, after the closing of the AssignAttributes element as <PreInsert>, <PostInsert>. These two operations are optional and are only used when there is a need to call other functionality in the application

## Validations

AsFile has the ability to perform validations in the XSLT portion of the configuration. This allows the developer to handle validation prior to business processing. It also provides flexibility to perform validation on requests that will NOT insert an activity record.

AsFile also has the ability to perform validation before it inserts any records into the database, using XSLT syntax.

## File Configuration Screen Image

### Steps to Access the File Rule

1. Navigate to **Rules** on the Main Menu.
2. Select **File** from the drop down list.
3. Select or Enter the **File Name Format**.

The screenshot shows a web-based configuration window titled "File(s)". At the top, there is a "File Name Format:" dropdown menu currently set to "(New File)". Below this, the window is split into two main sections. The top section, labeled "XSLT:", contains a large text area for editing XSLT rules, with "File Name Format:", "File ID:", and "Version:" labels above it. The bottom section, labeled "XMLData:", contains a similar large text area for editing XML data rules. At the bottom of the window, there are three buttons: "Check In", "New", and "Close".

## XML Examples

If a user needs to create an initial policy record, the record could only be the skeleton of a policy and the policy data can be loaded or input at another time.

The below request is sent to AsFile. This is accomplished by passing in the FileID and XML and invoking WebServiceBll.processFileReceived. AsFile is highly flexible and allows for any kind of incoming request format.

### Incoming Request

```
<Request>
  <Transactions>
    <Transaction>
      <Parameters>
        <Parameter NAME="ClientNumber">ACME</Parameter>
        <Parameter NAME="ProcessType">Policy</Parameter>
        <Parameter NAME="EffectiveDate">2006-12-11</Parameter>
        <Parameter NAME="PlanName">Acme Annual Term</Parameter>
        <Parameter NAME="CompanyName">Acme Life Insurance Company</Parameter>
      </Parameters>
      <Fields>
        <Field NAME="PolicyNumber" DATATYPE="Text">ACME09090</Field>
        <Field NAME="AdministrativeCode" DATATYPE="Text">R</Field>
      </Fields>
    </Transaction>
  </Transactions>
</Request>
```

### Gets passed to:

```
<File>
  <AssignAttributes>
    <Attribute NAME="PolicyGUID" TYPE="GUID"></Attribute>
    <Attribute NAME="ClientNumber"
TYPE="XPATH">/Request/Transactions/Transaction/Parameters/Parameter[@NAME='ClientNumber']</Attribute>
    <Attribute NAME="ImportMapName" TYPE="VALUE">AsXml</Attribute>
    <Attribute NAME="ClassName" TYPE="VALUE">com.adminserver.dcl.PolicyDcl</Attribute>
    <Attribute NAME="PolicyNumber"
TYPE="XPATH">/Request/Transactions/Transaction/Fields/Field[@NAME='PolicyNumber']</Attribute>
    <Attribute NAME="ExistingPolicyGuid" TYPE="SQL">Select policyGUID from AsPolicy where policyNumber =
'[PolicyNumber]'</Attribute>
  </AssignAttributes>
  <PostInsert>
    <Object
      CLASS="com.adminserver.utl.AsFilePostInsertActivityProcessorUtl">
    </Object>
  </PostInsert>
</File>
```

### Transformed by:

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0">
  <xsl:param name="PolicyGUID">none</xsl:param>
  <xsl:param name="PlanGUID">none</xsl:param>
  <xsl:param name="IULPlanGUID">none</xsl:param>
```

```

<xsl:param name="TermPlanGUID">none</xsl:param>
<xsl:param name="ClientGUID">none</xsl:param>
<xsl:param name="CompanyGUID"></xsl:param>
<xsl:param name="RoleGUID">none</xsl:param>
<xsl:param name="SystemDate">none</xsl:param>
<xsl:param name="ExistingPolicyGuid"></xsl:param>
<xsl:param name="ExistingPolicyError">PolicyNumberAlreadyExists</xsl:param>
<xsl:param name="MissingPolicyError">PolicyNumberIsMissingFromRequest</xsl:param>
<xsl:template match="/Request/Transactions/Transaction">
  <AsXml>
    <xsl:if test="Fields/Field[@NAME='PolicyNumber'] = "">
      <ValidationError ERRORSTATUSCODE="CRP001">
        <xsl:value-of select="$MissingPolicyError"></xsl:value-of>
      </ValidationError>
    </xsl:if>
    <xsl:if test="$ExistingPolicyGuid != "">
      <ValidationError ERRORSTATUSCODE="CRP002">
        <xsl:value-of select="$ExistingPolicyError"></xsl:value-of>
      </ValidationError>
    </xsl:if>
    <xsl:if test="$PlanGUID=$EclipsePlanGUID">
      <AsPolicy>
        <PolicyGuid>
          <xsl:value-of select="$PolicyGUID"></xsl:value-of>
        </PolicyGuid>
        <PolicyNumber>
          <xsl:value-of select="Fields/Field[@NAME='PolicyNumber']"></xsl:value-of>
        </PolicyNumber>
        <CreationDate>
          <xsl:value-of select="Parameters/Parameter[@NAME='EffectiveDate']"/>
        </CreationDate>
        <IssueStateCode></IssueStateCode>
        <PlanDate>
          <xsl:value-of select="Parameters/Parameter[@NAME='EffectiveDate']"/>
        </PlanDate>
        <StatusCode>08</StatusCode>
        <CompanyGuid>
          <xsl:value-of select="$CompanyGUID"></xsl:value-of>
        </CompanyGuid>
        <PlanGuid>
          <xsl:value-of select="$PlanGUID"></xsl:value-of>
        </PlanGuid>
        <XmlData></XmlData>
        <UpdatedGMT>
          <xsl:value-of select="Parameters/Parameter[@NAME='EffectiveDate']"/>
        </UpdatedGMT>
      </AsPolicy>
      <AsPolicyField>
        <PolicyGuid>
          <xsl:value-of select="$PolicyGUID"></xsl:value-of>
        </PolicyGuid>
        <FieldName>AdminUnit</FieldName>
        <FieldTypeCode>02</FieldTypeCode>
        <TextValue>
          <xsl:value-of select="Fields/Field[@NAME='AdministrativeCode']"/>
        </TextValue>
      </AsPolicyField>
      <AsPolicyField>
        <PolicyGuid>
          <xsl:value-of select="$PolicyGUID"></xsl:value-of>
        </PolicyGuid>
        <FieldName>PolicyFee</FieldName>
        <FieldTypeCode>02</FieldTypeCode>

```

```

                                <FloatValue>50</FloatValue>
                                </AsPolicyField>
                        </xsl:if>
                </AsXml>
        </xsl:template>
</xsl:stylesheet>

```

## Output/Response

```

<AsXml>
  <AsPolicy>
    <PolicyGuid>6C7259FD-9672-1D4D-55B1-66B294335E61</PolicyGuid>
    <PolicyNumber>980980988809</PolicyNumber>
    <CreationDate>2006-12-11</CreationDate>
    <IssueStateCode/>
    <PlanDate>2006-12-11</PlanDate>
    <StatusCode>08</StatusCode>
    <CompanyGuid>18B611A8-4429-4C67-94E6-3F4A882C9A8D</CompanyGuid>
    <PlanGuid>ABEB5E96-4F76-FDEF-E49A-EA5A42DD07F8</PlanGuid>
    <XmlData/>
    <UpdatedGMT>2006-12-11</UpdatedGMT>
  </AsPolicy>
  <AsPolicyField>
    <PolicyGuid>6C7259FD-9672-1D4D-55B1-66B294335E61</PolicyGuid>
    <FieldName>AdminUnit</FieldName>
    <FieldTypeCode>02</FieldTypeCode>
    <TextValue>R</TextValue>
  </AsPolicyField>
  <AsPolicyField>
    <PolicyGuid>6C7259FD-9672-1D4D-55B1-66B294335E61</PolicyGuid>
    <FieldName>PolicyFee</FieldName>
    <FieldTypeCode>04</FieldTypeCode>
    <FloatValue>50</FloatValue>
  </AsPolicyField>
</AsXml>

```