1. **To remove the prefix in the outbound file.**

Select source and target in a separate transformation and add the below code in xsl file.

<xsl:output method="xml"/>

<xsl:template match="/">

<xsl:apply-templates select="\*" mode="remprefix"/>

</xsl:template>

<xsl:template match="\*" mode="remprefix">

<xsl:variable name="newname" select="local-name(.)"/>

<xsl:element name="{$newname}" namespace="{namespace-uri()}">

<xsl:if test="$newname ='Document'">

<xsl:attribute name="xsi">http://www.w3.org/2001/XMLSchema-instance</xsl:attribute>

</xsl:if>

<xsl:apply-templates mode="copyall"

select="@\*|comment()|processing-instruction()|text()"/>

<xsl:apply-templates select="\*" mode="remprefix"/>

</xsl:element>

</xsl:template>

<xsl:template mode="copyall"

match="@\*|comment()|processing-instruction()|text()">

<xsl:copy>

<xsl:apply-templates mode="copyall"

select="@\*|comment()|processing-instruction()|text()"/>

</xsl:copy>

</xsl:template>

</xsl:stylesheet>

1. **To get the recently joined employee in each department.**

<imp1:ename>

<xsl:for-each select="$Invoke2\_GetEmpSelect\_OutputVariable.EmpCollection/ns1:EmpCollection/ns1:Emp[(ns1:deptno = $dept)]">

<xsl:sort select="ns1:hiredate" order="descending"/>

<xsl:if test="position() = 1">

<xsl:value-of select="ns1:ename"/>

</xsl:if>

</xsl:for-each>

</imp1:ename>

1. **To get the count of employees department wise.**

count($Invoke2\_GetEmpSelect\_OutputVariable.EmpCollection/ns1:EmpCollection/ns1:Emp[(ns1:deptno = $dept)])

1. **Write custom function in xslt template for square.**

<xsl:template match="/">

<xsl:variable name="var" select="/ns0:Customer/ns0:CustomerInfo"/>

<tns:processResponse>

<tns:Square>

<xsl:call-template name="square">

<xsl:with-param name="n" select="$var"/>

</xsl:call-template>

</tns:Square >

</tns:processResponse>

</xsl:template>

<xsl:template name="square">

<xsl:param name="n"/>

<xsl:variable name="res" select="$n \* $n"/>

<xsl:value-of select = "$res"/>

1. **Write custom function in xslt template to get month value based on month name.**

<xsl:variable name="var" select="/ns0:Customer/ns0:CustomerInfo"/>

<tns:processResponse>

<tns:Month>

<xsl:call-template name="RetMonth">

<xsl:with-param name="n" select="$var"/>

</xsl:call-template>

</tns:Month>

</tns:processResponse>

</xsl:template>

<xsl:template name="RetMonth">

<xsl:param name="n"/>

<xsl:if test="$n = 'JAN' or $n = 'JANUARY'">

<xsl:value-of select="01"/>

</xsl:if>

<xsl:if test="$n = 'FEB' or $n = 'FEBUARY'">

<xsl:value-of select="02"/>

</xsl:if>

<xsl:if test="$n = 'MAR' or $n = 'MARCH'">

<xsl:value-of select="03"/>

</xsl:if>

<xsl:if test="$n = 'APR' or $n = 'APRIL'">

<xsl:value-of select="04"/>

</xsl:if>

<xsl:if test="$n = 'MAY'">

<xsl:value-of select="05"/>

</xsl:if>

<xsl:if test="$n = 'JUN' or $n = 'JUNE'">

<xsl:value-of select="06"/>

</xsl:if>

<xsl:if test="$n = 'JUL' or $n = 'JULY'">

<xsl:value-of select="07"/>

</xsl:if>

<xsl:if test="$n = 'AUG' or $n = 'AUGUST'">

<xsl:value-of select="08"/>

</xsl:if>

<xsl:if test="$n = 'SEP' or $n = 'SEPTEMBER'">

<xsl:value-of select="09"/>

</xsl:if>

<xsl:if test="$n = 'OCT' or $n = 'OCTOBER'">

<xsl:value-of select="10"/>

</xsl:if>

<xsl:if test="$n = 'NOV' or $n = 'NOVEMBER'">

<xsl:value-of select="11"/>

</xsl:if>

<xsl:if test="$n = 'DEC' or $n = 'DECEMBER'">

<xsl:value-of select="12"/>

</xsl:if>

</xsl:template>

1. **Execute database query in xslt.**

<xsl:value-of select='oraext:query-database("SELECT SYSDATE FROM DUAL", true(), true(), "jdbc:oracle:thin:soademo/soademo@localhost:1521:XE")'/>

1. **To get the day of the week.**

xp20:format-dateTime (xp20:current-date(), &quot;[FNn]&quot; )

xp20:format-dateTime (xp20:current-date(), &quot;[F01]&quot; )

1. **To get distinct class values from the data.**

<xsl:key name="class" match="/ns0:Root-Element/ns0:Test/ns0:ITEM\_CLASS" use="." />

<xsl:template match="/">

<ns0:Root-Element>

<xsl:for-each select="/ns0:Root-Element/ns0:Test/ns0:ITEM\_CLASS[generate-id()= generate-id(key('class',.)[1])]">

<ns0:Test>

<ns0:ITEM\_CLASS>

<xsl:value-of select="."/>

</ns0:ITEM\_CLASS>

</ns0:Test>

</xsl:for-each>

</ns0:Root-Element>

</xsl:template>