```
private List m_avts = null
```

```
public void execute(TransformerImpl transformer) throws TransformerException
  {

        super.execute(transformer);

        if (null != m_avts)
        {
           int nAttrs = m_avts.size();


         for (int i = (nAttrs -1 ); i >=0 ; i--)
           {
              AVT avt = (AVT) m_avts.get(i);
              XPathContext xctxt = transformer.getXPathContext();
              int sourceNode = xctxt.getCurrentNode();
              String stringedValue =  avt.evaluate(xctxt, sourceNode, this);

              if (null != stringedValue)
              {
                 rhandler.addAttribute( avt.getURI(), avt.getName(), avt.getRawName(),
"CDATA", stringedValue, false);
              }
           }
        }
  }
```

```
public void compose(StylesheetRoot sroot) throws TransformerException
   {
    super.compose(sroot);
    StylesheetRoot.ComposeState cstate = sroot.getComposeState();
    java.util.Vector vnames = cstate.getVariableNames();
    if (null != m_avts)
    {
      int nAttrs = m_avts.size();


    for (int i = (nAttrs - 1); i >= 0; i--)
     {
       AVT avt = (AVT) m_avts.get(i);
      avt.fixupVariables(vnames, cstate.getGlobalsSize());
     }
    }
   }
```

```
public void resolvePrefixTables() throws TransformerException
  {

    if (null != m_avts)
    {
      int n = m_avts.size();


    for (int i = 0; i < n; i++)
    {
      AVT avt = (AVT) m_avts.get(i); String ns = avt.getURI();

      if ((null != ns) && (ns.length() > 0))
     {
       NamespaceAlias nsa =
stylesheet.getNamespaceAliasComposed(m_namespace); // %REVIEW% ns?
        if (null != nsa)
       {
        String namespace = nsa.getResultNamespace(); String resultPrefix =
nsa.getStylesheetPrefix();  // As per XSL WG
        String rawName = avt.getName();

        if ((null != resultPrefix) && (resultPrefix.length() > 0))
          rawName = resultPrefix + ":" + rawName;

        avt.setURI(namespace);  avt.setRawName(rawName);
      }
     }
    }
   }
```

It is very common that loop controlling variable are array and collection. Here 3 loop controlling variable from three different method are alias. If one of loop counter is known then other will be same. Static pointer and alias analysis is helpful to determine loop counter.