

step.xsl

```

<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
  <xsl:output method="text"/>

  <!-- Xstep2step
        Author: Zhisheng Huang 2002
        Version: 0.20
  -->

  <xsl:template match="/step">
    <xsl:apply-templates/>
  </xsl:template>

  <xsl:template match="head">
    <xsl:apply-templates/>
  </xsl:template>

  <xsl:template match="library">
    <xsl:apply-templates/>
  </xsl:template>

  <xsl:template match="action">
    script(<xsl:value-of select="@name"/>,Action):-
      Action =<xsl:apply-templates/>
      !.
  </xsl:template>

  <xsl:template match="start">
    start_action(<xsl:value-of select="@action"/>).
  </xsl:template>

  <xsl:template match="world">
    worldURL(<xsl:value-of select="@url"/>).
  </xsl:template>

  <xsl:template match="meta_language">
    meta_language(<xsl:value-of select="@value"/>).
  </xsl:template>

  <xsl:template match="seq">
    seq([<xsl:apply-templates/>
      ]),</xsl:template>

  <xsl:template match="par">
    par([<xsl:apply-templates/>
      ]),</xsl:template>

```

```

<xsl:template match="choice">
  choice([<xsl:apply-templates/>
    ]),</xsl:template>

<xsl:template match="dir" i:xsl:value-of select="@value"/></xsl:template>

<xsl:template match="speed" i:xsl:value-of select="@value"/></xsl:template>

<xsl:template match="increment" i:increment(i:xsl:value-of select="@x" /i, i:xsl:value-
  of select="@y" /i, i:xsl:value-of select="@z" />)</xsl:template>

<xsl:template match="position" i:position(i:xsl:value-of select="@x" /i, i:xsl:value-
  of select="@y" /i, i:xsl:value-of select="@z" />)</xsl:template>

<xsl:template match="rotation" i:rotation(i:xsl:value-of select="@x" /i, i:xsl:value-
  of select="@y" /i, i:xsl:value-of select="@z" /i, i:xsl:value-of select="@r" />)</xsl:template>

<xsl:template match="time" i:time(i:xsl:value-of select="@value" /i, i:xsl:value-
  of select="@unit" />)</xsl:template>

<xsl:template match="turn">
  turn(<xsl:value-of select="@actor" /i, i:xsl:value-of select="@part" /i, i:xsl:apply-
    templates select="(dir—rotation)" /i, i:xsl:apply-templates se-
    lect="(speed—time)" /i);xsl:if test="not(position()=last())">,</xsl:if></xsl:template>

<xsl:template match="move">
  move(<xsl:value-of select="@actor" /i, i:xsl:value-of select="@part" /i,
    i:xsl:apply-templates select="(dir—position—increment)" /i, i:xsl:apply-
    templates select="(speed—time)" /i);xsl:if test="not(position()=last())">,</xsl:if></xsl:template>

<xsl:template match="move_body">
  move_body(<xsl:value-of select="@actor" /i, i:xsl:apply-templates
    select="(dir—position—increment)" /i, i:xsl:apply-templates se-
    lect="(speed—time)" /i);xsl:if test="not(position()=last())">,</xsl:if></xsl:template>

<xsl:template match="script_action">
  script_action(<xsl:value-of select="@name" /i);xsl:if test="not(position()=last())">,</xsl:if></xsl:template>

<xsl:template match="if_then_else">
  if_then_else(<xsl:value-of select="@condition" /i, i:xsl:value-of
    select="@then" /i, i:xsl:value-of select="@else" /i);xsl:if test="not(position()=last())">,</xsl:if>

<xsl:template match="do">
  do(<xsl:value-of select="@state" /i);xsl:if test="not(position()=last())">,</xsl:if></xsl:template>

<xsl:template match="repeat">
  repeat(<xsl:value-of select="@action" /i, i:xsl:value-of se-
    lect="@times" /i);xsl:if test="not(position()=last())">,</xsl:if></xsl:template>

<!-- ignore all not matched -->

```

```
<!-- xsl:template match="*" priority="-1" / -->  
</xsl:stylesheet>
```