TEAMCENTER

Retail Footwear and Apparel — Techpacks Customization

Teamcenter 2412

SIEMENS

About Siemens Digital Industries Software

Siemens Digital Industries Software is a global leader in the growing field of product lifecycle management (PLM), manufacturing operations management (MOM), and electronic design automation (EDA) software, hardware, and services. Siemens works with more than 100,000 customers, leading the digitalization of their planning and manufacturing processes. At Siemens Digital Industries Software, we blur the boundaries between industry domains by integrating the virtual and physical, hardware and software, design and manufacturing worlds. With the rapid pace of innovation, digitalization is no longer tomorrow’s idea. We take what the future promises tomorrow and make it real for our customers today. Where today meets tomorrow. Our culture encourages creativity, welcomes fresh thinking and focuses on growth, so our people, our business, and our customers can achieve their full potential.

Support Center: **support.sw.siemens.com**

Send Feedback on Documentation: **support.sw.siemens.com/doc_feedback_form**

Contents

1. An overview of techpacks in Retail Footwear and Apparel

In Retail Footwear and Apparel, techpacks are generated using eXtensible Stylesheet Language (XSL). You can therefore customize these using XSL. The techpack customization follows the coding standards laid down by World Wide Web Consortium (W3C). For more information, see **http://www.w3.org/1999/XSL/Transform**.

2. Prerequisites to customizing techpacks

You must have the following installed to customize techpacks:

- Retail Footwear and Apparel
- An XSL Editor. Siemens Digital Industries Software recommends Altova XMLSpy.

3. Understanding Extensible Stylesheet Language

In Retail Footwear and Apparel, techpacks are generated using Extensible Stylesheet Language (XSL). XSL consists of the following parts:

Extensible Stylesheet Language Transformers (XSLT)

A language used to transform XML documents into XHTML or other XML documents. For more information on XSLT, see **https://www.w3schools.com/xml/xsl_intro.asp**.

XPath

A syntax for defining the parts of an XML document. XPath uses expressions to select nodes or node sets in an XML document. For more information on XPath, see **https://www.w3schools.com/xml/xpath_intro.asp**.

XSL-FO/ CSS3

XSL-FO is a language used to specify the output format for XML documents. You can also use CSS3 as a replacement for XSL-FO. For more information on CSS3, see **https://www.w3schools.com/css/**.

4. XSL development and its customization

Introduction

For every type of techpack you generate, a master XSL is developed. This analyzes TCXML for generating the FO file in intermediate format (IF), a proprietary XML format that represents the area tree generated by the layout engine. This IF file is converted to PDF through PDF-conversion processors.

There are different sections printed in the techpack. To print the each section, a child XSL is created and imported in the master XSL.

Printing an image associated with a business object

Images that are associated with business objects are fetched by combining the name of the image file with the associated **fmsticket** from TcXML, along with the **FMSBootStrapURL** value.

The image can be printed using the **fo:external-graphic** tag.

Product techpack

Master XSL

XSL Name

ProductTechpackData_Master.xsl

XSL Description

This is the master XSL file and is invoked by the **generate Techpack** call. It references different child XSL files to print different sections in the Techpack report.

XSL Code Flow

Based on the print group sections selected by the user, the master XSL file invokes different templates in the child XSL files, using the `call-template` function.

The master XSL accesses the child XSL files by importing them using the **import** call.

```
<!--This XSL contains reusable variables needed for Techpack report-->
<xsl:import href="StyleHelperXSL_Child.xsl"/>

<!--This XSL builds the header for all the pages-->
<xsl:import href="StylePageHeader_Child.xsl"/>
<!--This XSL builds the footer for all the pages-->
<xsl:import href="PageFooter_Child.xsl"/>

<!--This XSL builds the Cover Page-->
<xsl:import href="StyleCoverPage_Child.xsl"/>

<!--This XSL builds the Measurements Table-->
<xsl:import href="StyleMeasurementsDetailsPage_Child.xsl"/>
<!--This XSL builds the HTM Guide-->
<xsl:import href="StyleHTMImagesPage_Child.xsl"/>

<!--This XSL builds the Construction Details page-->
<xsl:import href="StyleConstructionDetailsPage_Child.xsl"/>
<!--This XSL build the Technical Summary Page-->
<xsl:import href="StyleTechnicalSummaryPage_Child.xsl"/>
<!--This XSL build the Construction Images Page-->
<xsl:import href="StyleConsDetailsImagesPage_Child.xsl"/>

<!--This XSL builds the BOM Data-->
<xsl:import href="StyleBOMPage_Child.xsl"/>
<!--This XSL builds the Artwork Images Sheet-->
<xsl:import href="StyleArtworkImagesPage_Child.xsl"/>

<!--This XSL builds the Samples Worksheet page-->
<xsl:import href="StyleSampleWorkSheetPage_Child.xsl"/>
<!--This XSL builds the Samples Evaluation Comments page-->
<xsl:import href="StyleSampleEvalComments_Child.xsl"/>
```

The print group sections selected by the user and other information relevant to generating the Techpack report are sent from the client to XSL as parameters.

The different parameters that are sent from the client are:

- **FMSBootStrapURL**

  Description:

  Specifies the FMS BootStrap URL defined in the **Fms_BootStrap_Urls** preference in Teamcenter.

  Usage:

This URL value is used to fetch the images associated with a business object by using the file name and the FMS ticket value.

- **OutDir**

  Description:

  Specifies the directory where all the XSL files and static images are stored during Techpack generation.

  Usage:

  This parameter value is used to print static images such as the header logo, home icon, and footer logo in PDF.

- **PrintGroup**

  Description:

  Specifies the print group sections selected by the user while generating a Techpack. The different print group sections selected are stored as a comma-separated list.

  Usage:

  This parameter is used to print the sections requested by the user.

  For example, while generating a Techpack, if the user chooses **Cover Page** and **BOM**, the PrintGroup parameter value is **Cover Page,BOM**.

  The print group sections for a product techpack are as follows:

Use this section	To print
Cover Page	The contents of the cover page.
BOM Consolidated	The contents of the **Bill of Materials** page and the **Artwork Images** page.
Colorway Consolidated	The **Colorway Grid** page.
ColorsAndQuality Consolidated	The **Colors And Quality** page.
Technical Summary	The contents of the **Technical Summary** page.
Measurement Details	The details of the **Measurement Details** page.
HTM Images	The contents of the **How to Measure Images** page.
Construction Details	The contents of the **Construction Images** page.

Use this section	To print
Sample	The contents of the **Sample Worksheets** page and the **Sample Evaluation Comments** page.
Product Summary	The contents of the **Product Summary** page.
PackageAndLabel	The contents of the **PackageAndLabel** page.
PackageAndLabel Constr Details	The packaging and labeling construction details.
PackageAndLabel Constr Images	The packaging and labeling construction images.

- **PrintGroupName**

  Description

  Specifies the print group selected by the user while generating the techpack.

  Usage:

  This value is used to print the print group name on the techpack report.

- **NumberOfDecimals**

  Description

  Specifies the number of decimals as defined in the **Retail_NumberOfDecimals** preference in Teamcenter.

  Usage:

  This value is used to round the decimal value to the number specified in this parameter.

- **ClassificationId**

  Description

  This value represents the **Retail PackageAndLabel** preference in Teamcenter.

  Usage

  This value is used to get the packaging and labeling details that are classified at the node as mentioned in this parameter value.

```
<!--Declaring the parameters-->
<!--Parameter to store the FMS BootStrap URL Value to fetch the images-->
<xsl:param name="FMSBootStrapURL"/>
<xsl:param name="OutDir"/>
<!--Parameter to store the Print Group string sent from UI-->
<xsl:param name="PrintGroup"/>
<xsl:param name="PrintGroupName"/>

<xsl:param name="NumberOfDecimals"/>
<xsl:param  name="ClassificationId"/>
```

- **Classification Tree**

  Specifies the classification tree structure of Teamcenter.

  > Note:
  >
  > The name of the classification tree must be **retail_classification.xml**. If the name of the classification tree is different, the name of the parameter must be changed accordingly.

  For printing the techpack, set the page layout first and then add the contents.

  To set the page layout, specify different attributes, such as those related to the page height, page width, and margin. You can use the following attribute sets defined in master XSL:

  - **PageMasterLayoutSet**
  - **BodyLayoutSet**
  - **HeaderLayoutSet**
  - **FooterLayoutSet**

  An `<fo:root>` object is written, which represents the root (top) node for XSL-FO documents.

  This `<fo:root>` object is divided into two main parts:

  - One layout master set (`layout-master-set`), which specifies one or more page master templates (`simple-page-master`) to be applied to the document.
  - One or more page sequences containing the contents of the Techpack.

  The `layout-master-set` is defined in master XSL, while the `simple-page-master` templates are defined in child XSLs so that each section or page can have its own layout.

For example, the user can choose to print the **Measurements** or **Sample Measurements** page in the landscape layout and the **Cover Page** in the portrait layout.

```
<xsl:template match="/">
  <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <!--Setting the master layout-->
    <fo:layout-master-set>
      <!--Loop through the print group values and print sections defined in them.
        These Print Group section values are the Print Group descriptions defined in the BMIDE.-->

      <xsl:for-each select="$PrintGroupValues">
        <!--Get the section value-->
        <xsl:variable name="SectionValue">
          <xsl:value-of select="."></xsl:value-of>
        </xsl:variable>

        <!--Calls the cover page pagination details-->
        <xsl:if test="$SectionValue = fn:string('Cover Page')">
          <xsl:call-template name="Cover_page_set">
          </xsl:call-template>
        </xsl:if>

        <!--Calls the technical summary pagination details-->
        <xsl:if test="$SectionValue = fn:string('Technical Summary')">
          <xsl:call-template name="technicalSummary_page_set">
          </xsl:call-template>
        </xsl:if>
```

The XSL loops through all the print group sections and calls the templates that set the page layout for that particular section.

The print group sections selected by the user must match the sections defined in XSL.

For example, if the print group description contains **Technical Summary**, it calls the **technicalSummary_page_set** template to define the page layouts. The **technicalSummary_page_set** template defines the unique master name and the **region-body**, **region-before**, **region-after** layouts.

```
<xsl:template name="technicalSummary_page_set">
  <!--Sets the page layout.-->
  <!--These attribute sets are present in Master XSL.-->
  <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
    master-name="Style_TechnicalSummary">
    <fo:region-body xsl:use-attribute-sets="BodyLayoutSet"></fo:region-body>
    <fo:region-before xsl:use-attribute-sets="HeaderLayoutSet"></fo:region-before>
    <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
  </fo:simple-page-master>
</xsl:template>
```

Once the layout is set, the page sequence is called to print the contents of the page.

```
<xsl:template match="/">
  <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <!--Setting the master layout-->
    <fo:layout-master-set>
      <!--Loop through the print group values and print sections defined in them.
        These Print Group section values are the Print Group descriptions defined in the BMIDE.-->
      <xsl:for-each select="$PrintGroupValues">
        <!--Get the section value-->
        <xsl:variable name="SectionValue">
          <xsl:value-of select="."></xsl:value-of>
        </xsl:variable>

        <!--Calls the cover page pagination details-->
        <xsl:if test="$SectionValue = fn:string('Cover Page')">
          <xsl:call-template name="Cover_page_set">
          </xsl:call-template>
        </xsl:if>
      ...
    </fo:layout-master-set>
    <!--Create the Page Contents-->
    <xsl:call-template name="PrintStyleTechpackPages">
    </xsl:call-template>
  </fo:root>
</xsl:template>
```

The **PrintStyleTechpackPages** template loops through all the print group sections and calls the relevant templates to print the contents of the page.

For example, if the print group section contains **Technical Summary**, it calls the **PrintStyleTechSummaryPages** template to print the technical summary contents.

```
<xsl:template name="PrintStyleTechpackPages">

    <xsl:for-each select="fn:tokenize($PrintGroup,',')">

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

        <!--Prints the technical summary page contents-->
        <xsl:if test="$SectionValue = fn:string('Technical Summary')">
            <xsl:call-template name="PrintStyleTechSummaryPages">
            </xsl:call-template>
        </xsl:if>

        <!--Prints the BOM and Artwork page contents-->
        <xsl:if test="$SectionValue = fn:string('BOM')">
            <xsl:call-template name="PrintStyleBOMPages">
            </xsl:call-template>
            <xsl:call-template name="PrintArtworkImagePages">
            </xsl:call-template>
        </xsl:if>

        ........

    </xsl:for-each>
</xsl:template>
```

The **Technical Summary** page is as follows:

```
<xsl:template name="PrintStyleTechSummaryPages">
    <fo:page-sequence master-reference="Style_TechnicalSummary">
        <!--Prints the header content for Technical Summary Images page.-->
        <fo:static-content flow-name="xsl-region-before">
            <xsl:call-template name="PrintStylePageHeader">
                <xsl:with-param name="PageName" select="fn:string('Technical Summary Page')"></xsl:with-param>
            </xsl:call-template>
            <!--Calls the templates which prints the common header-->
            <xsl:call-template name="PrintCommonHeader"/>
        </fo:static-content>

        <!--Prints the footer content for Technical Summary Images page and the template is
            defined in PageFooter_Child.xsl.-->
        <fo:static-content flow-name="xsl-region-after">
            <xsl:call-template name="PrintPageFooter">
            </xsl:call-template>
        </fo:static-content>

        <fo:flow flow-name="xsl-region-body">
            <!--Block Id is given as "Technical Summary" and is utilised in Table of contents page
                for navigating to Technical Summary page-->
            <fo:block id="fn:string('Technical Summary')">
                    Page Content flows
            </fo:block>
        </fo:flow>
    </fo:page-sequence>
</xsl:template>
```

xsl:region-before represents the header content.

xsl:region-after represents the footer content.

xsl:region-body represents the body content.

These are present in the **fo:page-sequence** to build the contents of the entire page.

Child XSLs

Cover page

XSL Name

ProductCoverPage_Child.xsl

XSL Description

This XSL prints the cover page when the user selects the **Cover Page** print group while generating the techpack.

This cover page contains the table of contents along with style details.

XSL Code Flow

The page layout for the cover page is defined in the **Cover_page_set** template. Some of the applicable attribute sets such as **PageMasterLayoutSet** and **FooterLayoutSet** are referenced from the **StyleTechpackData_Master.xsl** master XSL file.

The cover page contains only some parts of the header. New attribute sets are defined in the XSL files for the header and body contents. The new attribute sets are added in simple-page-master to change the layout.

```
<!--Sets the attributes for region body-->
<xsl:attribute-set name="CoverBodyLayoutSet">
   <xsl:attribute name="margin-top">0.5in</xsl:attribute>
   <xsl:attribute name="margin-bottom">0.5in</xsl:attribute>
</xsl:attribute-set>

<!--Sets the attributes for region before-->
<xsl:attribute-set name="CoverHeaderLayoutSet">
   <xsl:attribute name="extent">0.5in</xsl:attribute>
   <xsl:attribute name="overflow">hidden</xsl:attribute>
</xsl:attribute-set>

<xsl:template name="Cover_page_set">
   <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
      master-name="Techpack_CoverPage">
      <fo:region-body xsl:use-attribute-sets="CoverBodyLayoutSet"></fo:region-body>
      <fo:region-before xsl:use-attribute-sets="CoverHeaderLayoutSet"></fo:region-before>
      <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
   </fo:simple-page-master>
</xsl:template>
```

The **PrintCoverPage** template prints the cover page contents. It contains the **fo:page-sequence** tag that references the **simple-page-master** template.

The master name defined in the **master_name** tag in the **Cover_page_set** template must match the master-reference in the **fo:page-sequence** tag.

The **fo:page-sequence** tag contains the following:

xsl-region-before: This adds the header for the cover page. It calls the **PrintStylePageHeader** template and prints the following:

- The name of the page as specified in the **PageName** tag.
- The **Home** icon specified in the **IsCoverPage** tag.

  The **Home** icon appears on all pages in the techpack except the cover page.

xsl-region-after: This adds the footer for the cover page.

xsl-region-body: This adds the body contents for the cover page.

Each region tag contains a formatting block with a cover page ID. These formatting blocks generate the table of contents and the IDs define navigation.

Footer page

XSL Name

PageFooter_Child.xsl

XSL Description

This XSL prints the footer for all the pages.

XSL Code Flow

The **PrintPageFooter** template prints the footer contents, such as the bottom logo and the date.

Technical summary page

XSL Name

ProductTechnicalSummaryPage_Child.xsl

XSL Description

This XSL generates the **Technical Summary** page when the user selects the **Technical Summary** print group while generating the Techpack.

This page includes construction sketches. If there are multiple sketches, one sketch is shown per page.

XSL Code Flow

The page layout for the technical summary is defined in the **technicalSummary_page_set** template. The attribute sets in this template are referenced from the *ProductTechpackData_Master.xsl* master XSL file.

```
<xsl:template name="technicalSummary_page_set">
  <!--Sets the page layout.-->
  <!--These attribute sets are present in Master XSL.-->
  <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
      master-name="Style_TechnicalSummary">
      <fo:region-body xsl:use-attribute-sets="BodyLayoutSet"></fo:region-body>
      <fo:region-before xsl:use-attribute-sets="HeaderLayoutSet"></fo:region-before>
      <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
   </fo:simple-page-master>
</xsl:template>
```

The header height must be adjusted to include additional header attributes specific to the **Technical Summary** page. New attributes must be defined in the **Technical Summary** XSL file for the body content to move down and for the header content to extend. The new attribute sets are added in simple-page-master to change the layout.

```
<!--Sets the attributes for Technical Summary region body-->
<xsl:attribute-set name="TechSummaryBodyLayoutSet">
   <xsl:attribute name="margin-top">3.5in</xsl:attribute>
   <xsl:attribute name="margin-bottom">0.5in</xsl:attribute>
</xsl:attribute-set>
```

```
<!--Sets the attributes for Technical Summary region before-->
<xsl:attribute-set name="TechSummaryHeaderLayoutSet">
   <xsl:attribute name="extent">3.5in</xsl:attribute>
   <xsl:attribute name="overflow">hidden</xsl:attribute>
</xsl:attribute-set>
```

```
<xsl:template name="technicalSummary_page_set">
  <!--Sets the page layout.-->
  <!--These attribute sets are present in Master XSL.-->
  <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
      master-name="Style_TechnicalSummary">
      <fo:region-body xsl:use-attribute-sets="TechSummaryBodyLayoutSet"></fo:region-body>
      <fo:region-before xsl:use-attribute-sets="TechSummaryHeaderLayoutSet"></fo:region-before>
      <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
   </fo:simple-page-master>
</xsl:template>
```

The **PrintStyleTechSummaryPages** template prints the technical summary contents. It contains the **fo:page-sequence** tag, which references the simple-page-master template.

The master name defined in the **master_name** tag in the **Cover_page_set** template must match the master-reference in the **fo:page-sequence** tag.

The **fo:page-sequence** tag contains the following:

xsl-region-before: This adds the header for the **Technical Summary** page. It calls the **PrintStylePageHeader** template, and prints the name of the page as specified in the **PageName** tag.

xsl-region-after: This adds the footer for the page.

xsl-region-body: This adds the body contents for the page.

Each **region** tag contains a formatting block with a technical summary ID. These formatting blocks contain technical sketches and also define navigation from the cover page to the technical summary page.

The sketch details are stored in the *ConsSketchDetails* XSL variable. The *ConsSketchDetails* variable uses the **PrintConstructionsketchDetails** template to fetch the details in a result-tree format.

The sketch is printed using the **fo:external-graphic** tag.

Product summary page

XSL Name

ProductSummaryPage_Child.xsl

XSL Description

This XSL prints the **Product Summary** page when the user selects the **Product Summary** print group. The **Product Summary** page displays the front and back images of the style as well as the primary fabric details.

XSL Code Flow

The page layout for the product summary is defined in the **ProductTechpackData_Master.xsl** template. The attribute sets in this template are referenced from the *ProductTechpackData_Master.xsl* master XSL file.

The new attribute sets are added in **simple-page-master** to change the layout.

The **PrintStyleProductSummaryPages** template prints the **Product Summary** page contents. It contains the **fo:page-sequence** tag, which references the simple-page-master template.

The master name defined in the **master_name** tag in the **ProductSummary_page_set** template must match the master-reference in the **fo:page-sequence** tag.

The **fo:page-sequence** tag contains the following:

xsl-region-before: This adds the header for the **ProductSummary** page. It calls the **PrintStylePageHeader** template and prints the name of the page as specified in the **PageName** tag.

xsl-region-after: This adds the footer for the page.

xsl-region-body: This adds the body contents for the page.

Each **region** tag contains a formatting block with a product summary ID. These formatting blocks contain the front and back images of the style and also define the navigation from the cover page to the **Product Summary** page.

The style image details are stored in the *PrintImage* XSL variable. The *PrintImage* variable uses the **ImageDetails** template to fetch the details in a result-tree format.

The front and back images are printed using the **fo:external-graphic** tag.

The primary fabric details are stored in the *FabricContent* XSL variable. The *FabricContent* variable uses the **PrintPrimaryFabricData** template to fetch the details in a result-tree format.

Bill of Materials consolidated page

XSL Name

ProductBOMPage_Child.xsl

XSL Description

This XSL prints the **Bill of Materials Consolidated** page when the user selects the **BOM Consolidated** print group. The **Bill of Materials Consolidated** page displays fabrics, trims, and other details in separate tables.

XSL Code Flow

The page layout for the BOM is defined in the **BOM_page_set** template. The attribute sets in this template are referenced from the *StyleTechpackData_Master.xsl* file.

```
<xsl:template name="BOM_page_set">
  <!--Sets the page layout.-->
  <!--These attribute sets are present in Master XSL.-->
  <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
    master-name="Style_BOM">
    <fo:region-body xsl:use-attribute-sets="BodyLayoutSet"></fo:region-body>
    <fo:region-before xsl:use-attribute-sets="HeaderLayoutSet"></fo:region-before>
    <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
  </fo:simple-page-master>
</xsl:template>
```

The header height must be adjusted to include any additional header attributes specific to the **Bill of Materials Consolidated** page. New attributes must be defined in the **Bill of Materials Consolidated** XSL file for the body content to move down and for the header content to extend. The new attribute sets are added in simple-page-master to change the layout.

The **PrintStylePageHeader** template prints the BOM-page contents. It contains the **fo:page-sequence** tag, which references the simple-page-master template.

The master name defined in the **master_name** tag in the **BOM_page_set** template must match the master-reference in the **fo:page-sequence** tag.

The **fo:page-sequence** tag contains:

xsl-region-before: This adds the header for the **Bill of Materials Consolidated** page. It calls the **PrintStylePageHeader** template, and prints the name of the page as specified in the **PageName** tag.

xsl-region-after: This adds the footer for the page.

xsl-region-body: This adds the body contents for the page.

Each **region** tag contains a formatting block with a BOM ID. These formatting blocks contain tables with the details of all the fabric objects, trim objects, and artwork objects. The fabric types and subtypes are stored as comma-separated values in the *FabricSubTypes* variable. By default, the following values are added in the variable:

- Fabric
- Leather
- Knit
- Woven
- NonWoven

You can add custom types and subtypes and print them on the **Bill of Materials Consolidated** page.

Colorway Grid consolidated page

XSL Name

ProductColorwayGrid_Child.xsl

XSL Description

This XSL prints the **Colorway Grid Consolidated** page when the user selects the **Colorway Consolidated** print group.

This page includes the colorway details for fabrics, trims, and other details in a single table.

XSL Code Flow

The page layout for colorway grids is defined in the **ColorwayGrid_page_set** template. The attribute sets in this template are referenced from the *ProductTechpackData_Master.xsl* file.

```
<xsl:template name="ColorwayGrid_page_set">
  <!--Sets the page layout.-->
  <!--These attribute sets are present in Master XSL.-->
  <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
    master-name="Style_ColorwayGrid">
    <fo:region-body xsl:use-attribute-sets="BodyLayoutSet"></fo:region-body>
    <fo:region-before xsl:use-attribute-sets="HeaderLayoutSet"></fo:region-before>
    <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
  </fo:simple-page-master>
</xsl:template>
```

The header height must be adjusted to include any additional header attributes specific to the **Colorway Grid Consolidated** page. New attributes must be defined in the **Colorway Grid** XSL file for the body content to move down and for the header content to extend. The new attribute sets are added in simple-page-master to change the layout.

The **PrintStyleColorwayGridPages** template prints the **Colorway Grid Consolidated Page** contents. It contains the **fo:page-sequence** tag, which references the simple-page-master template.

```
master-name="Style_ColorwayGrid">
```

The **fo:page-sequence** tag contains:

xsl-region-before: This adds the header for the **Colorway Grid Consolidated** page. It calls the **PrintStylePageHeader** template, and prints the name of the page as specified in the **PageName** tag.

xsl-region-after: This adds the footer for the page.

xsl-region-body: This adds the body contents for the page.

Each **region** tag contains a formatting block with a *Colorway Grid ID*. This formatting block contains the colorway grid details. Fabrics, trims, and other objects are stored in tables. The fabric types and subtypes are stored as comma-separated values in the *FabricSubTypes* variable defined in the *StyleBOMPage_Child* XSL file. By default, the following values are added in the variable:

- Fabric
- Knit
- Woven
- NonWoven

You can add custom types and subtypes and print them on the **Colorway Grid** page.

The trim types and subtypes are stored as comma-separated values in the *TrimSubTypes* variable defined in the *StyleBOMPage_Child* XSL file. The only value *TrimSubTypes* contains is Trim.

You can add custom types and subtypes and print them on the **Colorway Grid** page.

The other types and subtypes are stored as comma-separated values in the *OtherSubTypes* variable defined in the *StyleBOMPage_Child* XSL file. The only value *OtherSubTypes* contains is Artwork.

You can add custom types and subtypes and print them on the **Colorway Grid** page.

The *BOMData* variable tags are stored in the *OrderedBOMData* variable. The *OrderedBOMData* variable contents are displayed in the result tree by the *PrintBOMDetails* template.

The preview image for BOM details is referenced from the *GetPreviewImageDetails* template, while classification details are referenced from the *GetClassificationDetails*, *FetchClassificationDetails*, and *FetchClassificationAttributes* templates.

Packaging and labeling page

XSL Name

ProductPackageAndLabelPage_Child.xsl

XSL Description

This XSL prints the **Packaging and Label** page in a table when the user selects the **PackageAndLabel** print group.

XSL Code Flow

The page layout for packaging and labeling is defined in the **PackageAndLabel_page_set** template. The attribute sets in this template are referenced from the *ProductTechpackData_Master.xsl* file.

```
<xsl:template name="PackageAndLabel_page_set">
  <!--Sets the page layout.-->
  <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
    master-name="Style_PackageAndLabel">
     <fo:region-body xsl:use-attribute-sets="BodyLayoutSet"></fo:region-body>
    <fo:region-before xsl:use-attribute-sets="HeaderLayoutSet"></fo:region-before>
    <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
  </fo:simple-page-master>
</xsl:template>
```

The header height must be adjusted to include any additional header attributes specific to the **Packaging and Labeling** page. New attributes must be defined in the **Package and Label** XSL file for the body content to move down and for the header content to extend. The new attribute sets are added in s**imple-page-master** to change the layout.

The **PrintStylePackageAndLabelPages** template prints the **Packaging and Labeling** page contents. It contains the **fo:page-sequence** tag, which references the simple-page-master template.

```
master-name="Style_PackageAndLabel">
```

The **fo:page-sequence** tag contains:

xsl-region-before: This adds the header for the **Packaging and Labeling Page** page. It calls the **PrintStylePageHeader** template and prints the name of the page as specified in the **PageName** tag.

xsl-region-after: This adds the footer for the page.

xsl-region-body: This adds the body contents for the page.

Each **region** tag contains a formatting block with a **PackagingAndLabel** ID. These formatting blocks contain packaging and labeling details classified in the **Retail_PackageAndLabel** preference. You can modify this preference to update the packaging and labeling details. The number of images to be printed per page is configured using the **PackageAndLabelImagesPerPage** parameter. The default value is **3**.

The classified images are printed using the **fo:external-graphic** tag. The dimensions for the classified images are specified in the XSL.

Measurements details page

XSL Name

ProductMeasurementsDetailsPage_Child.xsl

XSL Description

This XSL prints the measurement details of a style when the user selects the **Measurement Details** print group.

XSL Code Flow

The page layout for measurement details is defined in the **measurement_page_set** template. The **PageMasterLayoutSet** and **FooterLayoutSet** attribute sets in this template are referenced from the *StyleTechpackData_Master.xsl* file.

The header height must be adjusted to include any additional header attributes specific to **Measurement Details** page. New attributes must be defined in the **Measurement Details** XSL file for the body content to move down and for the header content to extend. The new attribute sets are added in simple-page-master to change the layout.

```
<!--Sets the attributes for Measurement Details region body-->
<xsl:attribute-set name="MeasurementBodyLayoutSet">
    <xsl:attribute name="margin-top">3in</xsl:attribute>
    <xsl:attribute name="margin-bottom">0.5in</xsl:attribute>
</xsl:attribute-set>

<!--Sets the attributes for Measurement Details region before-->
<xsl:attribute-set name="MeasurementHeaderLayoutSet">
    <xsl:attribute name="extent">3in</xsl:attribute>
    <xsl:attribute name="overflow">hidden</xsl:attribute>
</xsl:attribute-set>

<xsl:template name="measurement_page_set">
    <!--Sets the page layout.-->
    <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
        master-name="Style_Measurements">
        <fo:region-body xsl:use-attribute-sets="MeasurementBodyLayoutSet"></fo:region-body>
        <fo:region-before xsl:use-attribute-sets="MeasurementHeaderLayoutSet"></fo:region-before>
        <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
    </fo:simple-page-master>
</xsl:template>
```

The **PrintStyleMeasurementsPages** template prints the **Measurement Details** page contents. It contains the **fo:page-sequence** tag, which references the **simple-page-master** template.

```
master-name="Style_Measurements">
```

The master name defined in **master_name** must match the **master-reference** in the **fo:page-sequence** tag.

The **fo:page-sequence** tag contains:

xsl-region-before: This adds the header for the **Measurement Details** page. It calls the **PrintStylePageHeader** template and prints the name of the page as specified in the **PageName** tag.

xsl-region-after: This adds the footer for the page.

xsl-region-body: This adds the body contents for the page.

Each **region** tag contains a formatting block with a **Measurement Details** ID.

The measurement details are stored in the *POMRows* variable defined in the *ProductHelperXSL_Child* XSL file.

The number of size details to be printed per page is configured through the **SizePerPage** parameter. The default value is **3**.

HTM images page

XSL Name

ProductHTMImagesPage_Child.xsl

XSL Description

This XSL prints the HTM images of a Style when the user selects the **HTM Images** print group.

XSL Code Flow

The page layout is defined in the **HTM_page_set** template. The attribute sets in this template are referenced from the *ProductTechpackData_Master.xsl* file.

```
<xsl:template name="HTM_page_set">
 <!--Sets the page layout.-->
  <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
      master-name="Style_HTM">
      <fo:region-body xsl:use-attribute-sets="BodyLayoutSet"></fo:region-body>
      <fo:region-before xsl:use-attribute-sets="HeaderLayoutSet"></fo:region-before>
      <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
  </fo:simple-page-master>
</xsl:template>
```

The header height must be adjusted to include any additional header attributes specific to the **HTM Images Page** page. New attributes must be defined in the **HTM Images** XSL file for the body content to move down and for the header content to extend. The new attribute sets are added in **simple-page-master** to change the layout.

The **PrintStyleHTMPages** template prints the **Artwork Images Page** contents. It contains the **fo:page-sequence** tag, which references the **simple-page-master** template.

```
master-name="Style_HTM">
```

The master name defined in **master_name** must match the **master-reference** in the **fo:page-sequence** tag.

The **fo:page-sequence** tag contains:

xsl-region-before: This adds the header for the **Colorway Grid Consolidated** page. It calls the **PrintStylePageHeader** template and prints the name of the page as specified in the **PageName** tag.

xsl-region-after: This adds the footer for the page.

xsl-region-body: This adds the body contents for the page.

Each **region** tag contains a formatting block with a HTM Images ID.

The HTM Image data is stored in the *POMRows* variable defined in the *ProductHelperXSL_Child* XSL file.

The number of HTM images to be printed per page is configured through the **HTMImagePerPage** parameter. The default value is **3**.

The HTM images are printed using the **fo:external-graphic** tag. The **HTMImageDimensions** attribute set is defined in the XSL.

Construction details page

XSL Name

ProductConstructionDetailsPage_Child.xsl

XSL Description

This XSL prints the construction details of a style when the user selects the **Construction Details** print group.

XSL Code Flow

The page layout is defined in the **constructiondetails_page_set** template. The **FooterLayoutSet** and **PageMasterLayoutSet** attribute sets are referenced from the *ProductTechpackData_Master.xsl* file.

The header height must be adjusted to include any additional header attributes specific to the **Construction Details** page. New attributes must be defined in the **Construction Details** XSL file for the body content to move down and for the header content to extend. The new attribute sets are added in **simple-page-master** to change the layout.

```
<!--Sets the attributes for region body-->
<xsl:attribute-set name="ConstructionBodyLayoutSet">
   <xsl:attribute name="margin-top">3in</xsl:attribute>
   <xsl:attribute name="margin-bottom">0.5in</xsl:attribute>
</xsl:attribute-set>

<!--Sets the attributes for region before-->
<xsl:attribute-set name="ConstructionHeaderLayoutSet">
   <xsl:attribute name="extent">3in</xsl:attribute>
   <xsl:attribute name="overflow">hidden</xsl:attribute>
</xsl:attribute-set>

<xsl:template name="constructiondetails_page_set">
    <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
        master-name="Style_ConstructionDetails">
        <fo:region-body xsl:use-attribute-sets="ConstructionBodyLayoutSet"></fo:region-body>
        <fo:region-before xsl:use-attribute-sets="ConstructionHeaderLayoutSet"></fo:region-before>
        <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
    </fo:simple-page-master>
</xsl:template>
```

The **PrintStyleConstructionDetailsPages** template prints the **Construction Details Page** contents. It contains the **fo:page-sequence** tag, which references the **simple-page-master** template.

```
master-name="Style_ConstructionDetails">
```

The master name defined in **master_name** must match the master-reference in the **fo:page-sequence** tag.

The **fo:page-sequence** tag contains:

xsl-region-before: This adds the header for the **Construction Details** page. It calls the **PrintStylePageHeader** template and prints the name of the page as specified in the **PageName** tag.

xsl-region-after: This adds the footer for the page.

xsl-region-body: This adds the body contents for the page.

Each **region** tag contains a formatting block with a construction details ID.

The measurement details are stored in the *ConstructionDetailsRows* variable defined in the *ProductHelperXSL_Child* XSL file.

Construction images page

XSL Name

ProductConsDetailsImagesPage_Child.xsl

XSL Description

This XSL prints the construction images associated with a product when the user selects the **Construction Images** print group.

XSL Code flow

The page layout is defined in the **ConsDetailImage_page_set** template. The attribute sets used in this template are referenced from the *StyleTechpackData_Master.xsl* file.

```
<xsl:template name="ConsDetailImage_page_set">
   <!--Sets the page layout.-->
   <!--These attribute sets are present in Master XSL.-->
   <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
      master-name="Style_ConsDetailImage">
      <fo:region-body xsl:use-attribute-sets="BodyLayoutSet"></fo:region-body>
      <fo:region-before xsl:use-attribute-sets="HeaderLayoutSet"></fo:region-before>
      <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
   </fo:simple-page-master>
</xsl:template>
```

The header height must be adjusted to include some more header attributes specific to **Construction Images** page. New attributes must be defined in the **Construction Images** XSL file for the body content to move down and for the header content to extend. The new attribute sets are added in **simple-page-master** to change the layout.

The **PrintStyleHTMPages** template prints the **Artwork Images** page contents. It contains the **fo:page-sequence** tag which references the **simple-page-master** template.

```
master-name="Style_ConsDetailImage">
```

The master name defined in **master_name** must match the **master-reference** in the **fo:page-sequence** tag.

The **fo:page-sequence** tag contains:

xsl-region-before: This adds the header for the **Construction Images** page. It calls the **PrintStylePageHeader** template and prints the name of the page as specified in the **PageName** tag.

xsl-region-after: This adds the footer for the page.

xsl-region-body: This adds the body contents for the page.

Each **region** tag contains a formatting block with a construction images ID.

The construction image details are stored in the *ConstructionDetailsRows* variable defined in the *ProductHelperXSL_Child* XSL file.

The number of construction images to be printed per page is configured through the **SizePerPage** parameter. The default value is **3**.

The construction images are printed using the **fo:external-graphic** tag. The **ConstructionImageDimensions** attribute set is defined in XSL.

Sample worksheet page

XSL Name

ProductSampleWorkSheetPage_Child.xsl

XSL Description

This XSL prints the last three rounds of sample measurement details of a style when the user selects the **Sample** print group.

XSL Code Flow

The page layout is defined in the **Sample_page_set** template. Some attribute sets such as **PageMasterLayoutSet** and **FooterLayoutSet** are referenced from the **StyleTechpackData_Master** master XSL file.

The header height must be adjusted to include any additional header attributes specific to the **Sample Worksheet** page. New attributes must be defined in the **Sample Worksheet** XSL file for the body content to move down and for the header content to extend. The new attribute sets are added in **simple-page-master** to change the layout.

```
<!--Sets the attributes for Sample sheet region body-->
<xsl:attribute-set name="SampleBodyLayoutSet">
   <xsl:attribute name="margin-top">3.8in</xsl:attribute>
   <xsl:attribute name="margin-bottom">0.5in</xsl:attribute>
</xsl:attribute-set>

<!--Sets the attributes for Sample sheet region before-->
<xsl:attribute-set name="SampleHeaderLayoutSet">
   <xsl:attribute name="extent">3.8in</xsl:attribute>
   <xsl:attribute name="overflow">hidden</xsl:attribute>
</xsl:attribute-set>

<!--Sets the Sample Worksheet page layout set-->
<xsl:template name="Sample_page_set">
   <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
      master-name="Style_Sample">
      <fo:region-body xsl:use-attribute-sets="SampleBodyLayoutSet"></fo:region-body>
      <fo:region-before xsl:use-attribute-sets="SampleHeaderLayoutSet"></fo:region-before>
      <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
   </fo:simple-page-master>
</xsl:template>
```

The **PrintStyleSamplePages** template prints the **Sample Measurement Details Page** contents. It contains the **fo:page-sequence** tag, which references the **simple-page-master** template.

```
master-name="Style_Sample">
```

The master name defined in **master_name** must match the master-reference in the **fo:page-sequence** tag.

The **fo:page-sequence** tag contains the following:

xsl-region-before: This adds the header for the **Sample Worksheet** page. It calls the **PrintStylePageHeader** template and prints the name of the page as specified in the **PageName** tag.

xsl-region-after: This adds the footer for the page.

xsl-region-body: This adds the body contents for the page.

Each **region** tag contains a formatting block with a Sample ID.

Various sample details are stored in the following variables:

- *Sample Details*: stores sample worksheet details.

- *StyleSampleLatestRevTag*: stores latest revision or latest sequence details.
- *SampleMeasuredBy*: stores the name of the user from the last sample round.
- *SampleRequestedBy*: stores the name of the user who requested the sample. The value is stored in the *ret0RequestedBy* attribute.
- *AttentionValue*: stores attention value. The value is stored in the *ret0Attention* attribute.

These variables are referenced by the *StyleSampleHeader* template to print the style sample header.

Sample evaluation comments page

XSL Name

ProductSampleEvalComments_Child.xsl

XSL Description

This XSL prints the comments on the style sample. Comments include review comments, approval or rejection comments, and inputs by the designer and the technical designer when the user selects the **Sample** print group. This page is printed along with **Sample Worksheet** page.

XSL Code Flow

The page layout is defined in the **SampleComments_page_set** template. Some attribute sets such as **PageMasterLayoutSet** and **FooterLayoutSet** are referenced from the *StyleTechpackData_Master.xsl* file.

The header height must be adjusted to include some more header attributes specific to the **Sample Evaluation Comments** page. New attributes must be defined in the **Sample Evaluation Comments** XSL file for the body content to move down and for the header content to extend. The new attribute sets are added in **simple-page-master** to change the layout.

The **SampleBodyLayoutSet** and **SampleHeaderLayoutSet** attribute sets are referenced from *StyleSampleWorkSheetPage_Child.xsl*.

```
<!--Defines the Sample comments page set-->
<xsl:template name="SampleComments_page_set">
  <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
    master-name="Style_SampleEvaluation">
    <!--These attribute sets are defined in StyleSampleWorksheetPage_child.xsl-->
    <fo:region-body xsl:use-attribute-sets="SampleBodyLayoutSet"></fo:region-body>
    <fo:region-before xsl:use-attribute-sets="SampleHeaderLayoutSet"></fo:region-before>
     <!--This attribute sets is defined in Master XSL-->
    <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
  </fo:simple-page-master>
</xsl:template>
```

The **PrintStyleSampleCommentsPage** template prints the **Sample Evaluation Comments** page contents. It contains the **fo:page-sequence** tag which references the **simple-page-master** template.

```
master-name="Style_SampleEvaluation">
```

The master name defined in **master_name** must match the **master-reference** in the **fo:page-sequence** tag.

The **fo:page-sequence** tag contains the following:

xsl-region-before: This adds the header for the **Sample Evaluation Comments** page. It calls the **PrintStylePageHeader** template and prints the name of the page as specified in the **PageName** tag.

xsl-region-after: This adds the footer for the page.

xsl-region-body: This adds the body contents for the page.

Each **region** tag contains a formatting block with a Sample Comments ID.

The **Sample Evaluation Comments** page details are obtained directly from the attributes.

Sample additional comments page

XSL Name

StyleSampleAddComments_Child.xsl

XSL Description

This XSL prints the external comments about the style sample when the user selects the **Sample** print group. This page is printed along with **Sample Worksheet Page**.

XSL Code Flow

The page layout is defined in the **Sample_addComments_page_set** template. Some attribute sets such as **PageMasterLayoutSet**, **FooterLayoutSet** are referenced from the *StyleTechpackData_Master.xsl* file.

The header height must be adjusted to include any additional header attributes specific to the **Sample Additional Comments** page. New attributes must be defined in the *Sample Additional Comments* XSL file for the body content to move down and for the header content to extend. The new attribute sets are added in **simple-page-master** to change the layout.

```
<!--Defines the Sample comments page set-->
<xsl:template name="Sample_addComments_page_set">
  <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
    master-name="Style_SampleAddComments">
    <!--These attribute sets are defined in StyleSampleWorksheetPage_child.xsl-->
    <fo:region-body xsl:use-attribute-sets="SampleBodyLayoutSet"></fo:region-body>
    <fo:region-before xsl:use-attribute-sets="SampleHeaderLayoutSet"></fo:region-before>
     <!--This attribute sets is defined in Master XSL-->
    <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
  </fo:simple-page-master>
</xsl:template>
```

The **PrintStyleSampleAddCommentsPage** template prints the **Sample Additional Comments Page** contents. It contains the **fo:page-sequence** tag, which references the **simple-page-master** template.

```
master-name="Style_SampleAddComments">
```

The master name defined in **master_name** must match the **master-reference** in the **fo:page-sequence** tag.

The **fo:page-sequence** tag contains the following:

xsl-region-before: This adds the header for the **Sample Additional Comments** page. It calls the **PrintStylePageHeader** template and prints the name of the page as specified in the **PageName** tag.

xsl-region-after: This adds the footer for the page.

xsl-region-body: This adds the body contents for the page.

Each **region** tag contains a formatting block with a Sample Add Comments ID.

Data for **Sample Additional Comments** page is obtained directly from the external comments attribute.

Sample fit round images page

XSL Name

StyleFitRoundImagesPage_Child.xsl

XSL Description

This XSL prints the sample fit images along with the comments when the user selects the **Sample** print group. This page is printed along with **Sample Worksheet Page**.

XSL Code Flow

The page layout is defined in the **FitRound_page_set** template. Some attribute sets such as **PageMasterLayoutSet**, **FooterLayoutSet** are referenced from the *StyleTechpackData_Master.xsl* file.

The header height must be adjusted to include some more header attributes specific to the **Sample Fit Round Images** page. New attributes must be defined in the **Sample Fit Round Images** XSL file for the body content to move down and for the header content to extend. The new attribute sets are added in **simple-page-master** to change the layout.

Attribute sets **SampleBodyLayoutSet**, **SampleHeaderLayoutSet** are defined in the *StyleSampleWorkSheetPage_Child* XSL file.

```
<xsl:template name="FitRound_page_set">
<xsl:param name="MasterName"/>
  <!--Sets the page layout.-->
  <!--These attribute sets are present in Master XSL.-->
  <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet">
      <xsl:attribute name="master-name"><xsl:value-of select="$MasterName"/></xsl:attribute>
      <fo:region-body xsl:use-attribute-sets="BodyLayoutSet"></fo:region-body>
      <fo:region-before xsl:use-attribute-sets="HeaderLayoutSet"></fo:region-before>
      <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
  </fo:simple-page-master>
</xsl:template>
```

The **PrintStyleFitRoundImagesPage** template prints the **Sample Fit Round Images with Comments Page** contents. It contains the **fo:page-sequence** tag, which references the **simple-page-master** template.

The master name defined in **master_name** must match the **master-reference** in the **fo:page-sequence** tag.

The **fo:page-sequence** tag contains:

xsl-region-before: This adds the header for the **Sample Fit Round Images** page. It calls the **PrintStylePageHeader** template and prints the name of the page as specified in the **PageName** tag.

xsl-region-after: This adds the footer for the page.

xsl-region-body: This adds the body contents for the page.

Data for the **Sample Fit Images with Comments** page is obtained from the *SampleDetails* variable.

Helper XSL

XSL Name

StyleHelperXSL_Child.xsl

XSL Description

This XSL file defines the reusable variables and templates described in the previous sections.

Header page

XSL Name

ProductPageHeader_Child.xsl

XSL Description

This XSL generates the header content for all the pages in the final Techpack report. The header content may include page name, top logo, and the **Home** icon. The user can click the **Home** icon to go to the cover page.

XSL Code Flow

The **PrintStylePageHeader** template prints the page name and the top logo.

The **PageName** parameter value is printed for each section. Based on the value sent to this parameter, the page name is printed for each section.

The **PrintCommonHeader** named template prints the following header contents which are common across the pages.

Some of the header values are fetched from variable values. These variables are present in **StyleHelperXSL_Child.xsl**.

Softline set techpack

Master XSL

XSL Name

SoftlineSetTechpackData_Master.xsl

XSL Description

This is the master XSL file and will be invoked by the **generate Techpack** call. It references different child XSL files to print various sections in the techpack report.

XSL Code Flow

Based on the print group sections selected by the user, the master XSL file invokes different templates in the child XSLs, using the `call-template` function.

The master XSL accesses the child XSL files by importing them using the **import** call.

```
<!--Importing the helper XSLs-->
<!--This XSL builds the header for all the pages-->
<xsl:import href="StyleSetPageHeader_Child.xsl"/>
<!--This XSL builds the footer for all the pages-->
<xsl:import href="PageFooter_Child.xsl"/>
<!--This XSL builds the Cover Page-->
<xsl:import href="StyleSetCoverPage_Child.xsl"/>
<xsl:import href="StyleSetHelper_Child.xsl"/>
<!--This XSL builds the BOM Data-->
<xsl:import href="StyleSetBOMPage_Child.xsl"/>
<!--This XSL builds the  Information Data-->
<xsl:import href="StyleSetInformationPage_Child.xsl"/>
<!--This XSL builds the Summary Data-->
<xsl:import href="StyleSetSummaryPage_Child.xsl"/>
<!--This XSL builds the Packageand Label Data-->
<xsl:import href="StyleSetPackageAndLabelPage_Child.xsl"/>
```

The print group sections selected by the user, and other information relevant to generating the techpack report are sent from the client to XSL as parameters.

The different parameters that are sent from the client are:

- **FMSBootStrapURL**

Description:

Specifies the FMS BootStrap URL defined in the **Fms_BootStrap_Urls** preference in Teamcenter.

Usage:

This URL value is used to fetch the images associated with a business object by using the file name and the FMS ticket value.

- **OutDir**

  Description:

  Specifies the directory where all the XSL files and static images are stored during Techpack generation.

  Usage:

  This parameter value is used to print static images such as the header logo, home icon, and footer logo in PDF.

- **PrintGroup**

  Description:

  Specifies Print Group sections selected by the user while generating the techpack. The different Print Group sections selected are stored as a comma-separated list.

  Usage:

  This parameter is used to print the sections requested by the user.

  For example, while generating a techpack, if the user chooses the section Cover Page and BOM, the PrintGroup parameter value is **Cover Page,BOM**.

  Print Group sections for the Softline Set Techpack are as follows:

Use this section	To print
Cover Page	The cover page contents
BOM Consolidated	The contents of the **Bill of Materials** page and the **Artwork Images** page
Information Page	The **Information** page
ProductColorMatrix	The **ProductColorMatrix** page

Use this section	To print
Product Summary	The **StyleSet Summary** page
Package and Label	The **PackageAndLabel** page

- **PrintGroupName**

  Description:

  Specifies the print group selected by the user while generating the techpack.

  Usage:

  This value is used to print the Print Group name on the techpack report.

- **ClassificationId**

  Description:

  Specifies the **Retail PackageAndLabel** preference in Teamcenter.

  Usage:

  This value is used to get the packaging and labeling details that are classified at the node as mentioned in this parameter value.

```
<!--Parameter to store Packaging and Labeling Classification Id-->
<xsl:param name="ClassificationId"/>
```

Some important variables are discussed in this section.

1. **PrintGroupValues**

```
<xsl:variable name="PrintGroupValues" select="fn:tokenize($PrintGroup,',')"/>
```

2. **ClassificationTree**

   This variable stores the classification tree structure of Teamcenter.

Note:

The name of the classification tree must be **retail_classification.xml**. If the name of the classification tree is different, the name of the parameter must be changed accordingly.

For printing the techpack, set the page layout first and then add the contents.

To set the page layout, specify different attributes such as those related to the page height, page width, and margin. You can use the following attribute sets defined in the master XSL:

- **PageMasterLayoutSet**
- **BodyLayoutSet**
- **HeaderLayoutSet**
- **FooterLayoutSet**

An `<fo:root>` object is written which represents the root (top) node for XSL-FO documents.

This `<fo:root>` is divided into two main parts:

- One layout master set (`layout-master-set`), which specifies one or more page master templates (`simple-page-master`) to be applied to the document.
- One or more page sequences that contain the contents of the techpack.

The `layout-master-set` is defined in the master XSL, while the `simple-page-master` templates are defined in child XSLs so that each section or page can have its own layout.

```
<xsl:template match="/">
  <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <!--Setting the master layout-->
    <fo:layout-master-set>
        <xsl:for-each select="$PrintGroupValues">
          <xsl:variable name="SectionValue">
            <xsl:value-of select="."></xsl:value-of>
          </xsl:variable>
        <!--Calls the cover page pagination details-->
        <xsl:if test="$SectionValue = fn:string('Cover Page')">
          <xsl:call-template name="Cover_page_set">
          </xsl:call-template>
        </xsl:if>
        <!--Calls the BOM  page pagination details-->
        <xsl:if test="$SectionValue = fn:string('BOM')">
          <xsl:call-template name="BOM_page_set">
          </xsl:call-template>
        </xsl:if>
        <!--Calls the Information Page pagination details-->
        <xsl:if test="$SectionValue = fn:string('Information Page')">
          <xsl:call-template name="Information_page_set">
          </xsl:call-template>
        </xsl:if>
```

The XSL loops through all the print group sections and calls the templates that set the page layout for that particular section.

> Note:
>
> Print Group sections selected by the user should match the sections defined in XSL.

For example, if the print group description contains **BOM**, it calls the template called **BOM_page_set** to define the page layouts. The **BOM_page_set** template defines the unique master name and the **region-body**, **region-before**, and **region-after** layouts.

```
<xsl:template name="BOM_page_set">
  <!--Sets the page layout.-->
  <!--These attribute sets are present in Master XSL.-->
  <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
    master-name="StyleSet_BOM">
    <fo:region-body xsl:use-attribute-sets="BodyLayoutSet"></fo:region-body>
    <fo:region-before xsl:use-attribute-sets="HeaderLayoutSet"></fo:region-before>
    <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
  </fo:simple-page-master>
</xsl:template>
```

Once the layout is set, the page sequence is called to print the contents of the page.

```xml
<xsl:template match="/">
  <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <!--Setting the master layout-->
    <fo:layout-master-set>
        <xsl:for-each select="$PrintGroupValues">
          <xsl:variable name="SectionValue">
        <!--Calls the cover page pagination details-->
        <xsl:if test="$SectionValue = fn:string('Cover Page')">
        <!--Calls the BOM  page pagination details-->
        <xsl:if test="$SectionValue = fn:string('BOM')">
          <xsl:call-template name="BOM_page_set">
          </xsl:call-template>
        </xsl:if>
        <!--Calls the Information Page pagination details-->
        <xsl:if test="$SectionValue = fn:string('Information Page')">
        <!--Calls the ProductSummary worksheet pagination details-->
        <xsl:if test="$SectionValue = fn:string('Product Summary')">
        <!--<xsl:if test="$SectionValue = fn:string('ProductColorMatrix')">
        <!--Calls the Package and Labeling pagination details-->
        <xsl:if test="contains($SectionValue,string('PackageAndLabel'))">
      </xsl:for-each>
    </fo:layout-master-set>
    <!--Create the Page Contents-->
    <xsl:call-template name="PrintStyleSetTechpackPages">
    </xsl:call-template>
  </fo:root>
</xsl:template>
```

The **PrintStyleSetTechpackPages** template loops through all the print group sections and calls the templates that print the contents of the page.

For example, if the print group description contains **BOM**, it calls the **PrintStyleSetBOMPages** template to print the BOM contents.

```
<xsl:template name="PrintStyleSetTechpackPages">
  <xsl:for-each select="fn:tokenize($PrintGroup,',')">
    <xsl:variable name="SectionValue">
      <xsl:value-of select="."></xsl:value-of>
    </xsl:variable>
    <!--Prints the Style Set Cover page contents-->
    <xsl:if test="$SectionValue = fn:string('Cover Page')">
      <xsl:call-template name="PrintCoverPage">
      </xsl:call-template>
    </xsl:if>
    <!--Prints the Style Set BOM page contents-->
    <xsl:if test="$SectionValue = fn:string('BOM')">
      <xsl:call-template name="PrintStyleSetBOMPages">
      </xsl:call-template>
    </xsl:if>
    <!--Prints the Style Set Information page contents-->
    <xsl:if test="$SectionValue = fn:string('Information Page')">
      <xsl:call-template name="PrintStyleSetInformationPages">
      </xsl:call-template>
    </xsl:if>
    <!--Prints the Style Set ProductSummary page contents-->
    <xsl:if test="$SectionValue = fn:string('Product Summary')">
      <xsl:call-template name="PrintStyleSetSummaryPages">
      </xsl:call-template>
    </xsl:if>
```

The **BOM** page is as follows:

```xml
<xsl:template name="PrintStyleSetBOMPages">
  <fo:page-sequence master-reference="StyleSet_BOM">
    <fo:static-content flow-name="xsl-region-before">
      <!--Displays the BOM page header.-->
      <xsl:call-template name="PrintStyleSetPageHeader">
        <xsl:with-param name="PageName" select="fn:string('Set TechPack BOM')"></xsl:with-param>
      </xsl:call-template>
      <!--Calls the templates which prints the common header-->
      <xsl:call-template name="PrintCommonHeader"/>
      <xsl:call-template name="PrintSizeDetailsHeader"/>
    </fo:static-content>
    <!--Displays the BOM page footer.-->
    <fo:static-content flow-name="xsl-region-after">
      <xsl:call-template name="PrintPageFooter">
      </xsl:call-template>
    </fo:static-content>

        <!--Displays the BOM page -->
        <fo:flow flow-name="xsl-region-body">
          <!--Block Id is given as "BOM " and is utilised in Table of contents page for navigating to StyleSetBOM page-->
          <fo:block id="fn:string('BOM')">
      </fo:flow>
    </fo:page-sequence>
</xsl:template>
```

xsl:region-before represents the header content.

xsl:region-after represents the footer content.

xsl:region-body represents the body contents.

These are present in the **fo: page-sequence** to build the entire page contents.

Softline techpack child XSLs

Header page

XSL Name

SoftlineSetPageHeader_Child.xsl

XSL Description

This XSL generates header content for all the pages in the final techpack report. The header content may include page name, top logo, and the **Home** icon. The user can click the **Home** icon to go to the cover page.

XSL Code Flow

The **PrintStylePageHeader** template prints the page name and the top logo.

The **PageName** parameter value is printed for each section. Based on the value sent to this parameter, the page name is printed for each section.

The **PrintCommonHeader** template prints the following header contents, which are common across the pages.

Some of the header values are fetched from variable values. These variables are present in the *StyleSetHelperXSL_Child.xsl* file.

Cover page

XSL Name

SoftlineSetCoverPage_Child.xsl

XSL Description

This XSL generates the cover page when the user selects the **Cover Page** print group while generating the techpack.

The cover page contains the table of contents along with StyleSet details.

XSL Code Flow

The page layout for cover page is defined in the **Cover_page_set** template. Some of the applicable attribute sets such as **PageMasterLayoutSet** and **FooterLayoutSet** are referenced from the *StyleTechpackData_Master.xsl* file.

The cover page contains only some parts of the header. New attribute sets are defined in the XSL files for the header and body contents. The new attribute sets are added in **simple-page-master** to change the layout.

```
<!--Sets the attributes for region body-->
<xsl:attribute-set name="CoverBodyLayoutSet">
   <xsl:attribute name="margin-top">0.5in</xsl:attribute>
   <xsl:attribute name="margin-bottom">0.5in</xsl:attribute>
</xsl:attribute-set>

<!--Sets the attributes for region before-->
<xsl:attribute-set name="CoverHeaderLayoutSet">
   <xsl:attribute name="extent">0.5in</xsl:attribute>
   <xsl:attribute name="overflow">hidden</xsl:attribute>
</xsl:attribute-set>

<xsl:template name="Cover_page_set">
   <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
      master-name="Techpack_CoverPage">
      <fo:region-body xsl:use-attribute-sets="CoverBodyLayoutSet"></fo:region-body>
      <fo:region-before xsl:use-attribute-sets="CoverHeaderLayoutSet"></fo:region-before>
      <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
   </fo:simple-page-master>
</xsl:template>
```

The **PrintCoverPage** template prints the cover page contents. It contains the **fo:page-sequence** tag, which references the **simple-page-master** template.

The master name defined in the **master_name** tag in the **Cover_page_set** template must match the **master-reference** in the **fo:page-sequence** tag.

The **fo:page-sequence** tag contains the following:

xsl-region-before: This adds the header for the **Cover Page**. It calls the **PrintStyleSetPageHeader** template and prints:

- The name of the page as specified in the **PageName** tag.
- The **Home** icon specified in the **IsCoverPage** tag.

  The Home icon appears on all pages in the techpack except the cover page.

xsl-region-after: this adds the footer for the cover page.

xsl-region-body: this adds the body contents for the cover page.

Each **region** tag contains a formatting block with a cover page ID. These formatting blocks generate the table of contents and the IDs define navigation.

Footer page

XSL Name

PageFooter_Child.xsl

XSL Description

This XSL prints the footer for all the pages.

XSL Code Flow

The **PrintPageFooter** template prints the footer contents, such as the bottom logo and the date.

Bill of materials page

XSL Name

SoftlineSetBOMPage_Child.xsl

XSL Description

This XSL prints the **Bill of Materials Page** when the user selects the **BOM** print group.

This page prints style-related data referenced in the style set.

XSL Code Flow

The page layout for BOM is defined in the **BOM_page_set** template. The attribute sets in this template are referenced from the *SoftlineSetTechpackData_Master.xsl* file.

```
<xsl:template name="BOM_page_set">
  <!--Sets the page layout.-->
  <!--These attribute sets are present in Master XSL.-->
  <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
    master-name="StyleSet_BOM">
    <fo:region-body xsl:use-attribute-sets="BodyLayoutSet"></fo:region-body>
    <fo:region-before xsl:use-attribute-sets="HeaderLayoutSet"></fo:region-before>
    <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
  </fo:simple-page-master>
</xsl:template>
```

The header height must be adjusted to include any additional header attributes specific to the **Bill of Materials** page. New attributes must be defined in the **Bill of Materials** XSL file for the body content to move down and for the header content to extend. The new attribute sets are added in **simple-page-master** to change the layout.

The **PrintStyleSetBOMPages** template prints the **Bill of Materials Page** contents. It contains the **fo:page-sequence** tag, which references the **simple-page-master** template.

The master name defined in the **master_name** tag in the **BOM_page_set** template must match the **master-reference** in the **fo:page-sequence** tag.

The **fo:page-sequence** tag contains the following:

xsl-region-before: This adds the header for the **Bill of Materials Consolidated** page. It calls the **PrintStylePageHeader** template and prints the name of the page as specified in the **PageName** tag.

xsl-region-after: This adds the footer for the page.

xsl-region-body: This adds the body contents for the page.

Each **region** tag contains a formatting block with a BOM ID. These formatting blocks contain BOM details. All styles referenced in a style set are stored in a table.

BOM data is retrieved using the *BOMData* variable and the *PRINTBOMDetails* template. The *BOMData* variable is stored in the *SoftlineSetHelper_Child* XSL file.

Preview images are retrieved from the *GetPreviewImageDetails* template in the *SoftlineSetHelper_Child* XSL.

Softline set information page

XSL Name

SoftlineSetInformationPage_Child.xsl

XSL Description

This XSL prints the **Softline Set Information** page when the user selects the **Information Page** print group.

This page prints the data related to styles referenced in the style set and primary fabric details of each style in a table.

XSL Code Flow

The page layout for the Information page is defined in the **Information_page_set** template. The attribute sets used in this template are referenced from the **StyleSetTechpackData_Master** master XSL file.

```
<xsl:template name="Information_page_set">
    <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
      master-name="Style_Image" >
      <fo:region-body xsl:use-attribute-sets="BodyLayoutSet"></fo:region-body>
      <fo:region-before xsl:use-attribute-sets="HeaderLayoutSet"></fo:region-before>
      <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
    </fo:simple-page-master>
</xsl:template>
```

The header height needs to be adjusted to include some more header attributes specific to the **Information** page. The new attributes must be defined in the **Information** XSL file for the body content to move down and for the header content to extend. The new attribute sets are added in **simple-page-master** to change the layout.

The **PrintStyleSetInformationPages** template prints the **Information** page contents. It contains the **fo:page-sequence** tag, which references the **simple-page-master** template.

The **fo:page-sequence** contains:

xsl-region-before: Adds the header for the **Information** page. It calls the **PrintStyleSetPageHeader** template and prints the name of the page as specified in the **PageName** tag.

xsl-region-after: Adds the footer for the page.

xsl-region-body: Adds the body contents for the page.

Each **region** tag contains a formatting block with an Information Page ID. These formatting blocks contain images with Style ID and the Style Name and Primary Fabric for each style.

Preview images for each style are retrieved from the **GetPreviewImageDetails** template in the **StyleSetHelperXSL_Child** XSL. These images are printed using the **fo:external-graphic** tag. The number of style images to be printed per page is configured through the **SizePerPage** parameter. The default value is **3**.

The BOM details referenced in the style set are stored in the **BOMDetailsForStyleTypes** variable. These BOM details are retrieved using the **BOMData** variable in the **PrintBOMDetails** template.

The **PrintChildBOMData** template loops through children of each style and calls the **PrintFabricBOMDetails** template to collect Primary Fabric Details.

Softline set summary page

XSL Name

SoftLineSetSummaryPage_Child.xsl

XSL Description

This XSL prints the Set Summary Page when the user selects the **Product Summary** print group.

This page prints the Set Summary details in a tabular format.

XSL Code Flow

The page layout is defined in the **Summary_page_set** template. The attribute sets used in this template are referenced from the **StyleSetTechpackData_Master.xsl** master XSL file.

```
<!--Setting the pagination details of the StyleSetSummaryPage-->
<xsl:template name="Summary_page_set">
    <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
        master-name="StyleSet_Summary" >
        <fo:region-body xsl:use-attribute-sets="BodyLayoutSet"></fo:region-body>
        <fo:region-before xsl:use-attribute-sets="HeaderLayoutSet"></fo:region-before>
        <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
    </fo:simple-page-master>
</xsl:template>
```

The header height needs to be adjusted to include some more header attributes specific to the **Set Summary** page. New attribute must be defined in the **Set Summary** XSL file for the body content to move down and for the header content to extend. The new attribute sets are added in **simple-page-master** to change the layout.

The **PrintStyleSetSummaryPages** template prints the Set Summary Page contents. It contains an **fo:page-sequence** tag, which has a reference to **simple-page-master** templates name.

```
master-name="StyleSet_Summary" >
```

> Note:
>
> The master name defined in **master_name** must match the master-reference in the **fo:page-sequence** tag.

The **fo:page-sequence** contains:

xsl-region-before: adds the header for the **Construction Images** page. It calls the **PrintStylePageHeader** template and prints the name of the page as specified in the **PageName** tag.

xsl-region-after: Adds the footer for the page.

xsl-region-body: Adds the body contents for the page.

Each **region** tag contains a formatting block with a Product Summary ID. Comments and fit round details are displayed according to the date of creation.

The **PrintStyleSampleDetails** template retrieves the date and time of creation of a style sample from the **StyleSampleDetails** variable format and stores this information in a fragment tree.

The latest style samples are stored in the **LatestStyleSampleDetails** variable and the maximum sequence ID for the style samples, are stored in the variable **StyleSampleLatestTimeDetails** in the form of a resultant fragment tree.

Packaging and labeling page

XSL Name

StyleSetPackageAndLabelPage_Child.xsl

XSL Description

This XSL prints the **Packaging and Label** page in a table when the user selects the **PackageAndLabel** print group.

XSL Code Flow

The page layout is defined in the **PackageAndLabel_page_set** template. The attribute sets used in this template are referenced from the *SoftLineSetTechpackData_Master.xsl* master XSL file.

```
<xsl:template name="PackageAndLabel_page_set">
  <!--Sets the page layout.-->
  <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
    master-name="StyleSet_PackageAndLabel">
     <fo:region-body xsl:use-attribute-sets="BodyLayoutSet"></fo:region-body>
    <fo:region-before xsl:use-attribute-sets="HeaderLayoutSet"></fo:region-before>
    <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
  </fo:simple-page-master>
</xsl:template>
```

The header height must be adjusted to include some more header attributes specific to the **Packaging and Labeling** page. The new attributes must be defined in the **Package and Label** XSL file for the body content to move down and for the header content to extend. The new attribute sets are added in **simple-page-master** to change the layout.

The **PrintSoftLineSetPackageAndLabelPages** template prints the **Packaging and Label** page contents. It contains the **fo:page-sequence** tag which references the **simple-page-master** template.

The **fo:page-sequence** contains:

xsl-region-before: Adds the header for the **Packaging and Labeling** page. It calls the **PrintSoftlineSetPageHeader** template and prints the name of the page as specified in the **PageName** tag.

xsl-region-after: Adds the footer for the page.

xsl-region-body: Adds the body contents for the page.

Each **region** tag contains a formatting block with a **PackagingAndLabel** ID. These formatting blocks contain packaging and labeling details classified in the **Retail_PackageAndLabel** preference. You can modify this preference to update packaging and labeling details. The number of images to be printed per page is configured through the **PackageAndLabelImagesPerPage** parameter. The default value is **3**.

The classified images are printed using the **fo:external-graphic** tag.

The product color matrix page

XSL Name

SoftLineSetColorwayGridPage_Child.xsl

XSL Description

This XSL prints the **Product Color Matrix** page when the user selects the **ProductColorMatrix** print group.

This page prints the **Configured Colorway Details** for styles referred in a style set.

XSL Code Flow

The page layout is defined in the **Colorway_page_set** template. The attribute sets used in this template are referenced from the **StyleSetTechpackData_Master.xsl** master XSL file.

```
<xsl:template name="Colorway_page_set">
  <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
    master-name="StyleSet_ProductColorMatrix" >
    <fo:region-body xsl:use-attribute-sets="BodyLayoutSet"></fo:region-body>
    <fo:region-before xsl:use-attribute-sets="HeaderLayoutSet"></fo:region-before>
    <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
  </fo:simple-page-master>
</xsl:template>
```

The header height needs to be adjusted to include some more header attributes specific to **Product Color Matrix Page** page. New attributes must be defined in the **SoftlineSetColorwayGridPage** XSL file for the body content to move down and for the header content to extend. The new attribute sets are added in **simple-page-master** to change the layout.

The **PrintProductColorMatrix** template prints the **Product Color Matrix Page** contents. It contains the **fo:page-sequence** tag, which references the **simple-page-master** template.

The **fo:page-sequence** contains:

xsl-region-before: Adds the header for the **Product Color Matrix** page. The number of SKU images to be printed per page is configured through the **ColorwaysPerPage** parameter. It calls the **PrintStyleSetPageHeader** template and prints the name of the page as specified in the **PageName** tag.

xsl-region-after: Adds the footer for the page.

xsl-region-body: Adds the body contents for the page.

Each **region** tag contains a formatting block with a **ProductColorMatrix** ID. These formatting blocks contain color matrix details.

Bill of Materials SKU page

XSL Name

StyleSetSKUGridPage_Child.xsl

XSL Description

This XSL prints the **Bill of Materials SKU** page when the user selects the **BOM SKU** print group.

This page prints the **Configured Colorway Details** for a style set.

XSL Code Flow

The page layout is defined in the **SKU_page_set** template. The attribute sets used in this template are referenced from the **StyleSetTechpackData_Master.xsl** master XSL file.

The header height needs to be adjusted to include some more header attributes specific to **Bill of Materials SKU** page. New attributes must be defined in the **StyleSetSKUGridPage XSL** XSL file for the body content to move down and for the header content to extend. The new attribute sets are added in **simple-page-master** to change the layout.

The **PrintSKUGridData** template prints the **Bill of Materials SKU Page** contents. It contains the **fo:page-sequence** tag which references the **simple-page-master** template.

The **fo:page-sequence** contains:

xsl-region-before: Adds the header for the **Bill of Materials SKU** page. It calls the **PrintStyleSetPageHeader** template and prints the name of the page as specified in the **PageName** tag. The number of SKUs to be printed per page is configured through the **SKUsPerPage** parameter. The default value is **2**.

xsl-region-after: Adds the footer for the page.

xsl-region-body: Adds the body contents for the page.

Each **region** tag contains a formatting block with a **BOM SKU** ID. These formatting blocks contain color matrix details.

The styles are collected and shown in a tabular format along with the configured SKU details.

Helper XSL

XSL Name

SoftLineSetHelper_Child.xsl

XSL Description

This XSL file defines reusable variables and templates described in the previous sections.

Hardline techpack

Master XSL

XSL Name

HardlineTechpackData_Master.xsl

XSL Description

This XSL is the master XSL and will be invoked from the **generate Techpack** call. It references different child XSLs to print different sections in the techpack report.

XSL Code Flow

Based on the print group sections selected by the user, the master XSL invokes differently named templates in child XSLs using the `call-template` function.

To access the child XSL files, the master XSL imports them using the **import** call.

```
<!--This XSL contains reusable variables needed for Techpack report-->
<xsl:import href="HardLineHelperXSL_Child.xsl"/>

<!--This XSL builds the header for all the pages-->
<xsl:import href="HardLinePageHeader_Child.xsl"/>
<!--This XSL builds the footer for all the pages-->
<xsl:import href="PageFooter_Child.xsl"/>

<!--This XSL builds the Cover Page-->
<xsl:import href="HardLineCoverPage_Child.xsl"/>


<!--This XSL builds the HardLineProductSummary Data-->
<xsl:import href="HardLineProductSummaryPage_Child.xsl"/>
<!--This XSL builds the HardLinePackagingSummary Data-->
<xsl:import href="HardLinePackagingSummaryPage_Child.xsl"/>
```

The **Print Group** sections selected by user and other information relevant to generating the techpack report are sent from the client to the XSL as parameters.

The different parameters that are sent from the client are:

- **FMSBootStrapURL**

  Description:

  Specifies the FMS BootStrap URL defined in the **Fms_BootStrap_Urls** preference in Teamcenter.

  Usage:

  This URL value is used to fetch the images associated with a business object by using the file name and the FMS ticket value.

- **OutDir**

  Description:

  Specifies the directory where all XSL files and the static images are stored during techpack generation.

  Usage:

  This parameter value is used to print static images such as the header logo, home icon and footer logo in PDF.

- **PrintGroup**

Description:

Specifies the **Print Group** sections selected by the user while generating a techpack. The different **Print Group** sections selected are stored as a comma-separated list.

Usage:

This parameter is used to print the sections requested by the user.

For example, while generating a techpack, if the user selects **Cover Page** and **BOM**, then the **PrintGroup** parameter value would be **Cover Page,BOM**.

The **PrintGroup** sections specific to the hardLine techpack are as follows:

Section Name	Purpose
Cover Page	To print the cover page contents
Product Summary	To print the contents of the **HardLine Summary** page.
Packaging Summary	To print the contents of the **Packaging Summary** page

- **PrintGroupName**

  Description:

  Specifies the **Print Group** selected by the user while generating the techpack.

  Usage:

  This value is used to print the **Print Group Name** on the techpack report.

```
<!--Declaring the parameters-->
<!--Parameter to store the FMS BootStrap URL Value to fetch the images-->
<xsl:param name="FMSBootStrapURL"/>
<xsl:param name="OutDir"/>
<!--Parameter to store the Print Group string sent from UI-->
<xsl:param name="PrintGroup"/>
<xsl:param name="PrintGroupName"/>
```

For printing the techpack, set the page layout first and then add the contents.

To set the page layout, specify different attributes such as page-height, page-width, and margin-related attributes. You can use the following attribute sets defined in the master XSL:

- **PageMasterLayoutSet**

- **BodyLayoutSet**
- **HeaderLayoutSet**
- **FooterLayoutSet**

An `<fo:root>` object is written, which represents the root (top) node for XSL-FO documents.

This `<fo:root>` is divided into two main parts:

- One layout master set (layout-master-set), which specifies one or more page master templates (simple-page-master) to be applied to the document.
- One or more page sequences that contain the contents of the techpack.

The **layout-master-set** is defined in the master XSL and the **simple-page-masters** are defined in child XSLs so that each section or page can have its own layout.

For example, landscape layout can be set for the **Product Summary** and **Packaging Summary** page, where as portrait layout can be set for **Cover Page**.

```
<xsl:template match="/">
  <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <!--Setting the master layout-->
    <fo:layout-master-set>
      <!--Loop through the print group values and print sections defined in them.These Print Group
          section values are the Print Group descriptions defined in the BMIDE.-->
      <xsl:for-each select="$PrintGroupValues">
        <!--Get the section value-->
        <xsl:variable name="SectionValue">
          <xsl:value-of select="."></xsl:value-of>
        </xsl:variable>

        <!--Calls the cover page pagination details-->
        <xsl:if test="$SectionValue = fn:string('Cover Page')">
          <xsl:call-template name="Cover_page_set">
          </xsl:call-template>
        </xsl:if>

        <!--Calls the Product Summary pagination details-->
        <xsl:if test="$SectionValue = fn:string('Product Summary')">
          <xsl:call-template name="Product_Summary_page_set">
          </xsl:call-template>
        </xsl:if>

        <!--Calls the Packaging Summary pagination details-->
        <xsl:if test="$SectionValue = fn:string('Packaging Summary')">
          <xsl:call-template name=" Packaging_Summary_page_set">
          </xsl:call-template>
        </xsl:if>
```

The XSL loops through all the **Print Group** sections and calls the templates that set the page layout for that particular section.

> Note:
>
> The **Print Group** sections selected by the user must match the sections defined in the XSL.

For example, if the Print Group Description contains **Product Summary**, it calls the template called **Product_Summary_page_set** to define the page layouts. The **Product_Summary_page_set** template defines the unique master name and the **region-body**, **region-before**, and **region-after** layouts.

```xml
<!--Setting the pagination details of the HardlineProductSummaryPage-->
<xsl:template name="Product_Summary_page_set">
    <!--Sets the page layout.-->
    <!--These attribute sets are present in Master XSL.-->
    <fo:simple-page-master xsl:use-attribute-sets="PageMasterLayoutSet"
        master-name="HardLine_ProductSummary">
        <fo:region-body xsl:use-attribute-sets="BodyLayoutSet"></fo:region-body>
        <fo:region-before xsl:use-attribute-sets="HeaderLayoutSet"></fo:region-before>
        <fo:region-after xsl:use-attribute-sets="FooterLayoutSet"></fo:region-after>
    </fo:simple-page-master>
</xsl:template>
```

Once the layout is set, the page sequence calls to print the contents of the page.

```xml
<xsl:template match="/">
    <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
    <!--Setting the master layout-->
        <fo:layout-master-set>
            <!--Loop through the print group values and print sections defined in them.These Print Group section values are the Print Group description
in the BMIDE.-->
            <xsl:for-each select="$PrintGroupValues">
                <!--Get the section value-->
                <xsl:variable name="SectionValue">
                    <xsl:value-of select="."></xsl:value-of>
                </xsl:variable>

                <!--Calls the cover page pagination details-->
                <xsl:if test="$SectionValue = fn:string('Cover Page')">
                    <xsl:call-template name="Cover_page_set">
                    </xsl:call-template>
                </xsl:if>

                <!--Calls the Product Summary pagination details-->
                <xsl:if test="$SectionValue = fn:string('Product Summary')">
                    <xsl:call-template name="Product_Summary_page_set">
                    </xsl:call-template>
                </xsl:if>

                </xsl:for-each>
        </fo:layout-master-set>
        <!--Create the Page Contents-->
        <xsl:call-template name="PrintHardLineTechpackPages">
        </xsl:call-template>
    </fo:root>
</xsl:template>
```

The **PrintHardLineTechpackPages** template loops out through all the Print Group descriptions and calls the templates that print the contents of the page.

For example, if the print group description contains **Product Summary**, it calls the **PrintHardLineProductSummaryPages** template to print the technical summary contents.

```
<xsl:template name="PrintHardLineTechpackPages">

    <xsl:for-each select="fn:tokenize($PrintGroup,',')">

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

        <!--Prints the cover page contents-->
        <xsl:if test="$SectionValue = fn:string('Cover Page')">
            <xsl:call-template name="PrintCoverPage">
            </xsl:call-template>
        </xsl:if>

        <!--Prints the ProductSummary page contents-->
        <xsl:if test="$SectionValue = fn:string('Product Summary')">
            <xsl:call-template name="PrintHardLineProductSummaryPages">
            </xsl:call-template>
        </xsl:if>

        <!--Prints the PackagingSummary page contents-->
        <xsl:if test="$SectionValue = fn:string('Packaging Summary')">
            <xsl:call-template name="PrintHardLinePackagingSummaryPages">
            </xsl:call-template>
        </xsl:if>

    </xsl:for-each>
</xsl:template>
```

```
<!--Template which prints the HardLineProductSummaryPages.This page will fetch the construction sketches attached to product and will print them-->
<xsl:template name="PrintHardLineProductSummaryPages">
  <fo:page-sequence master-reference="HardLine_ProductSummary">
    <!--Prints the header content for Product Summary Images page -->
    <fo:static-content flow-name="xsl-region-before">
      <xsl:call-template name="PrintHardLinePageHeader">
        <xsl:with-param name="PageName" select="fn:string('Product Summary')"/>
      </xsl:call-template>
      <!--Calls the templates which prints the common header-->
      <xsl:call-template name="PrintCommonHeader"/>
    </fo:static-content>

    <!--Prints the footer content for Product Summary Images page and the template is defined in PageFooter_Child.xsl-->
    <fo:static-content flow-name="xsl-region-after">
      <!--Calls the template which prints the footer-->
      <xsl:call-template name="PrintPageFooter"></xsl:call-template>
    </fo:static-content>

    <!--the flow of the xsl-region body starts here-->
    <fo:flow flow-name="xsl-region-body">
      <!--Block Id is given as "Product Summary" and is utilised in Table of contents page for navigating to Product Summary page-->
      <fo:block id="fn:string('Product Summary')">
      Page Content Flows
        <fo:table width="100%" border-spacing="2pt" border-left="solid 1pt black" border-bottom="solid 1pt black" border-right="solid 1pt black">
      </fo:block>
    </fo:flow>
  </fo:page-sequence>
</xsl:template>
```

The **xsl:region-before** represents the header content.

The **xsl:region-after** represents the footer content.

The **xsl:region-body** represents the body contents.

All these are present in the **fo:page-sequence** to build the entire page contents.

Generating the techpack for a set

A set may contain a style or a footwear or hardline as its child item. When the techpack for set is being generated, the system should also generate the techpack for its children. To do this, you must set the **Retail_PrintGroups** preference. This preference mentions the default report template and the default print group that needs to be used for generating the child techpack based on its type.

Example:

```
StyleRevision_DefaultPrintGroup:PrintGroup001
StyleRevision_DefaultTechpackTemplate:029665
```

In the example, a **StyleRevision** if present as one of the child items of the set will use `PrintGroup001` as its print group, and **Report template** with ID `029665` as the report template for generating the techpack.

Similarly, if there is a **HardLineRevision** within the set, then the system will use **Pre Production-Style** as its print group, and report template with ID `200` as the report template for generating the techpack.

```
HardLineRevision_DefaultPrintGroup:Pre Production-Style
HardLineRevision_DefaultTechpackTemplate:200
```

Configuring footwear, footwear component and softline component techpacks

You can configure techpack templates and preferences for the following:

- Footwear
- Footwear component objects, such as soles, or trims.
- Softline component objects such as fabric, or lace.

For details, see the corresponding sections in the **Product techpack** section.

5. Add new print group

Create a new print group

1. Log on to Retail Footwear and Apparel.

2. Navigate to and open the folder where you want to create the line plan, for example, your **Newstuff** folder.

3. Click **Add** ⊕.

4. In the **Add** panel, do one of the following:

   - Select **Retail Print Group** from the **Recent** list.

     OR

   - In **Other**, type **Print** and select **Retail Print Group** from the list.

5. In the **Add** panel, enter values for the following:

   - **Print Group Name**

   - **Description**

   - **Print Group Sections**

6. Click **Add**.

6. Add new techpack template

Create a new techpack template

1. Log on to Retail Footwear and Apparel.

2. Navigate to and open the folder where you want to create the line plan, for example, your **Newstuff** folder.

3. Click **Add** ⊕.

4. In the **Add** panel, do one of the following:

   - Select **Retail Report Template** from the **Recent** list.

     OR

   - In **Other**, type **Retail Report** and select **Retail Report Template** from the list.

5. In the **Add** panel, enter values for the following:

   - **Report Template ID**
   - **Name**
   - **Description**
   - **Print Group**
   - **Option Set**
   - **Report Style Sheet**

6. Click **Add**.

To use this report template on a style, classify it at the required root node.

7. Manage print group sections

Update a print group

To update a print group:

1. Log on to Retail Footwear and Apparel.

2. Select **Search** → **Retail Print Group**.

3. In the Search view that appears, specify the criteria to search for a print group.

4. Click **Search**.

5. Double-click the required **Print Group** from the displayed results.

6. In the **Summary** page of the **Print Group**, click **Start Edits** .

7. Modify the required details and click **Save Edits** .

XSL changes to support new section

If you add new sections to the XSLs, make sure that you follow these guidelines:

- A Child XSL must be created for each new section.

  For example, To support a **Product Summary**, you must create a new child XSL called **StyleProductSummaryPage_Child.xsl**.

- For creating the basic content, such as the page layout, header (region-before), footer (region-after), and body (region-body) refer to the OOTB child XSLs. All the OOTB Child XSLs must follow the **XXX_Child.xsl** naming convention.

- Master XSL has to be changed to support this new section. In the master XSL the newly developed child XSL has to be imported.

```
<!--This XSL builds the ProductSummary  page-->
<xsl:import href="StyleProductSummaryPage_Child.xsl"/>
```

- Once the child XSL is imported, you must call the page set template and the contents template.

For example, use the following code to call the pagination details template of product summary child XSL from the master XSL.

```
<!--Calls the ProductSummary worksheet pagination details-->
<xsl:if test="contains($SectionValue,string('Product Summary'))">
   <xsl:call-template name="ProductSummary_page_set">
   </xsl:call-template>
</xsl:if>
```

> Note:
>
> The print group section name must be the same as the section name in Master XSL.

Fetching data for a new section

Closure rules and property sets must be modified to fetch the data required for new sections. An example is fetching the data for the **Product Summary page**. To do this, you must:

- Understand the data flow for fetching the front back images.
- Modify the closure rule accordingly for processing and traversal of the objects.
- Modify the property set accordingly to fetch the properties required.

Update the child XSL and the master XSL for new sections

1. Browse to and select the required Report Dataset, right-click, and choose **Named References**.

2. Click **Import**.

3. Select **Ret0MasterXSL_Reference** and based on the report template type browse to the stylesheet location.

   For example: For Style Techpack generation, attach the Style Techpack related XSLs as named references to the newly created report dataset. Browse to the Style Techpack XSLs location at:

   $TC_DATA\RetailSamples\windows\retail_samples\samples\reports\StyleTechpackXSLs and select the **XXX_Master.xsl** file. Attach the file it with the **Ret0MasterXSL_Reference**.

4. Click **Import**. The selected master XSL is added to the list.

5. Now attach the remaining XSLs, that is, the helper files, with **Ret0HelperXSL_Reference**.

8. Add a new attribute to the techpack section

Fetching data in TCXML

Closure rules and property sets must be modified to fetch the data required for new sections. An example is adding new attributes in the **Header** page. To do this, you must:

- Understand the data flow for fetching the attribute value.
- Modify the closure rule accordingly for processing and traversal of the objects.
- Modify the property set accordingly to fetch the properties required.

XSL changes to support new attributes

To support new attributes, and display and fetch the relevant data, you must also change the child XSL related to the changed attribute.

For example, if you add a new attribute to a common header section, then you must also modify the **StylePageHeader_Child.xsl** file.

Update the child XSL and the master XSL for new attributes

1. Select the required **Report Dataset**, right-click, and select **Named References**.

2. Click **Import**.

3. Select **Ret0MasterXSL_Reference** and based on the report template type browse to the stylesheet location.

   For example: For Style Techpack generation, attach the Style Techpack related XSLs as named references to the newly created report dataset. Browse to the Style Techpack XSLs location at:

   $TC_DATA\RetailSamples\windows\retail_samples\samples\reports\StyleTechpackXSLs and select the **XXX_Master.xsl** file. Attach the file with the **Ret0MasterXSL_Reference**.

4. Click **Import**. The selected master XSL is added to the list.

5. Now attach the remaining XSLs, that is, the helper files, with **Ret0HelperXSL_Reference**.

9. Good practices for creating techpacks

- To ensure that the content fits on a single cover page, limit the table of contents to 13 sections and include a maximum of three lines for comments.

> Note:
>
> All the other style design attributes contain a single line of text.

- For the images, maintain an aspect ratio of 1:1. In order to achieve this, make sure that the **content-width** and the **content-height** have the same value, and set the scaling to **uniform**. A low resolution image helps to reduce the file size of the tech pack.

  - The preview image on the cover page has a content-width of 2.5 inches, a content-height of 2.5 inches, and the scaling as **uniform**.

  - The image on the **BOM** page has a content-width of 64 pixels, a content-height of 64 pixels, and the scaling as **uniform**.

- To enhance the appearance of the page, include a maximum of 10 attributes in the main header, and a maximum of 6 attributes in the sub header. Each attribute must have a single line of text. Such headers only occupy 25% of the page, with 75% for the body text.

  If any attribute in the header contains lengthy text, it can be moved to a new page.

  As an example, you can refer to the **Sample Additional Comments Page**.

- Assuming that all the attributes have a single line of text, on the **Bill of Materials** page, if the number of classification attributes in the **Details** column is up to 10, the page will have two BOM lines per page.

- On the **Artwork Summary** page, we recommend a maximum of three attributes per artwork image.

- On the **Measurements Details** Page:

  - A maximum of 15 point of measure (POM) values is recommended, where each POM field description has a single line text.

  - If a POM field description spans more than one line of text, the **Measurement Details** page can accommodate eight POMs.

  - There can be four sizes per page on the **Measurements Details** page.

- Support for lengthy text:

  - A maximum of 15 words can be accommodated on a single line in the techpack.
  - As the FOP recognizes a space in between characters as a line break, insert a space in the case of lengthy text for viewing the page better.
  - Comma-separated lengthy text supported with a space between characters provides a better page view.
- Width of the **Measurements** column: The **OOTB** width for the **Size** column supports eight digits. This can be configured using the XSL update based on the customer data.
- Width of the **POM ID** column: The **OOTB** width for the **POM ID** column supports six characters. This can be configured using the XSL update based on the customer data.