



Smarten Working with Reports

Document Information	
Document ID	Smarten-Working-with-Reports
Document Version	1.0
Product Version	5.2
Date	14-June-2021
Recipient	NA
Author	EMTPL

© Copyright Elegant MicroWeb Technologies Pvt. Ltd. 2021. All Rights Reserved.

Statement of Confidentiality, Disclaimer and Copyright

This document contains information that is proprietary and confidential to EMTPL, which shall not be disclosed, transmitted, or duplicated, used in whole or in part for any purpose other than its intended purpose. Any use or disclosure in whole or in part of this information without the express written permission of EMTPL is prohibited.

Any other company and product names mentioned are used for identification purpose only, may be trademarks of their respective owners and are duly acknowledged.

Disclaimer

This document is intended to support administrators, technology managers or developers using and implementing Smarten. The business needs of each organization will vary and this document is expected to provide guidelines and not rules for making any decisions related to Smarten. The overall performance of Smarten depends on many factors, including but not limited to hardware configuration and network throughput.

Contents

1	Introduction	4
2	Report Creation Workflow	5
3	Creating Report Template.....	6
3.1	FreeMarker Template Engine Overview	6
3.1.1	FreeMarker Basic Directives.....	6
3.1.2	Defining and using Variables	7
3.1.3	Number Formatting.....	8
3.1.4	Date Formatting	8
3.2	Integrating Smarten Object Data into Template.....	9
3.2.1	Get Smarten Object Data—Full Data	9
3.2.2	Get Smarten Object Data—Based on Row Number.....	10
3.2.3	Get Smarten object Data—Based on Row Number and Column Number	11
3.3	Rendering Different Controls	11
3.3.1	Showing Data in Table.....	11
3.3.2	Showing Data in Graph.....	12
3.4	Defining Report Page Formatting.....	14
3.4.1	Defining Page Break	14
3.4.2	Defining Page Header/Footer	17
3.4.3	Defining Page Numbers.....	21
4	Creating Report	25
4.1	Upload Report Template	25
4.2	Associate BI Objects	26
4.3	Preview.....	27
4.4	Define Page Filters and Retrieval Parameters.....	28
4.5	Assign Permissions	29
4.6	Define Export Parameters	29
5	Product and Support Information.....	31

1 Introduction

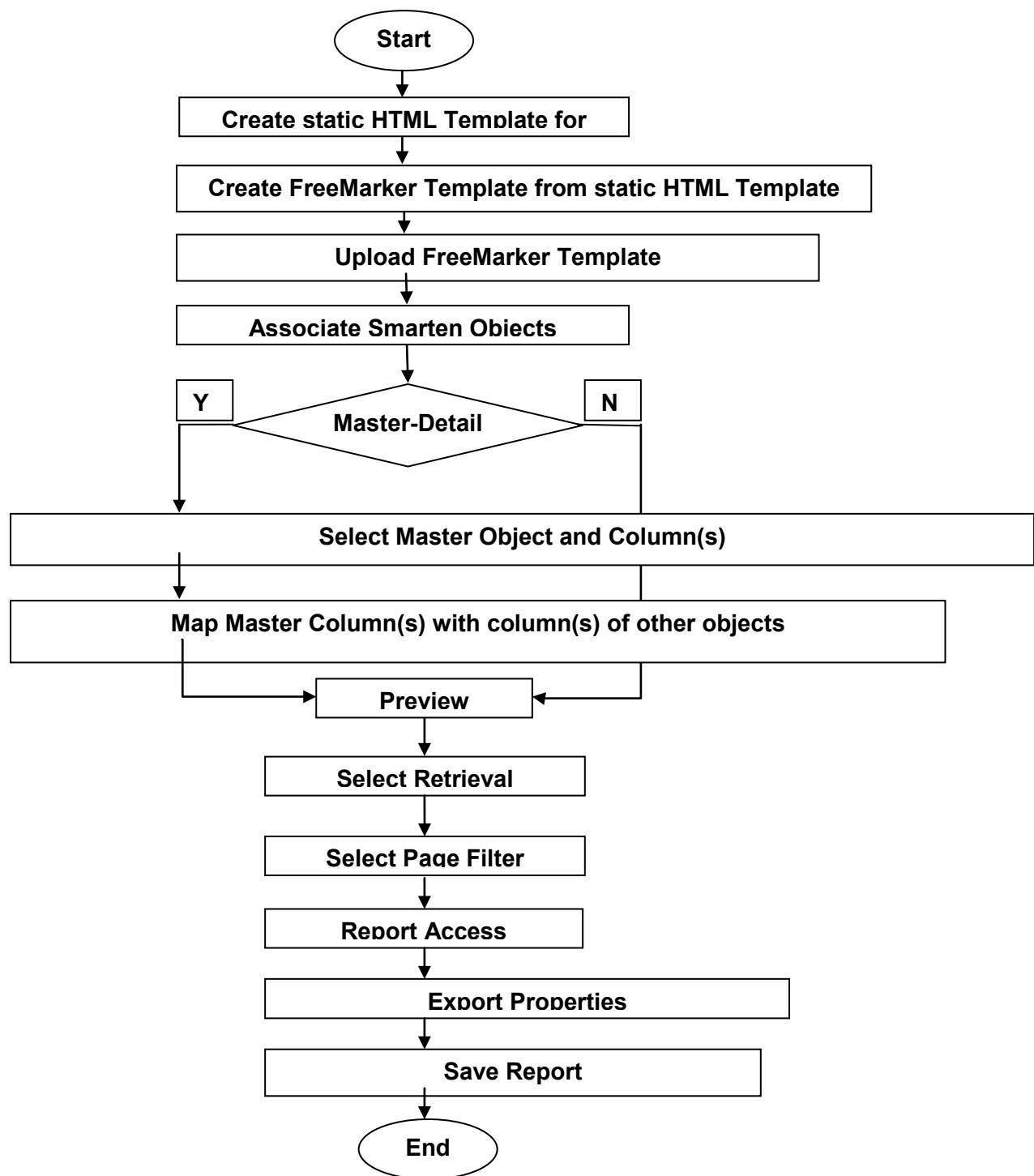
Smarten provides a reports module for creating custom reports with a complex layout or pixel-perfect reports. Users can create such reports as a credit card statement, invoice, purchase orders, or an annual report. Reports can be exported or published in PDF format through delivery and publishing agent built in Smarten.

The reports framework uses the FreeMarker template engine to define a report template.

Below are the basic steps to create reports:

- Create FreeMarker template file
- Upload template file
- Associate Smarten objects and data to the report template
- Preview report
- Define export parameters

2 Report Creation Workflow



3 Creating Report Template

3.1 FreeMarker Template Engine Overview

Apache FreeMarker is a *template engine*, which is a Java library to generate text output, e.g., HTML web pages, e-mails, configuration files, and source code, based on templates and changing data. Templates are written in the FreeMarker Template Language (FTL), which is a simple specialized language. Usually, a general-purpose programming language, such as Java, .NET, and PHP, is used to prepare the data. Then, Apache FreeMarker displays that prepared data using templates. In the template, you are focusing on how to present the data, and outside the template, you are focusing on what data to present.

3.1.1 FreeMarker Basic Directives

The table below describes FreeMarker basic directives with examples.

Directive	Description	Example
if condition	You can use <code>if</code> , <code>elseif</code> , and <code>else</code> directives to conditionally skip a section of the template. The <i>condition(s)</i> must evaluate to a boolean value or else an error will abort template processing. The <code>elseif</code> and <code>else</code> must occur inside <code>if</code> (that is, between the <code>if</code> start-tag and end-tag). The <code>if</code> can contain any number of <code>elseif</code> (including 0) and at the end optionally one <code>else</code> .	<code><#if condition></code> <code><#else></code> <code></#if></code>
counter	Returns the 1-based index where the iteration (which is identified by the loop variable name) currently stands.	<code><#list ["a," "b," "c"] as i></code> <code> \${i?counter}: \${i}</code> <code></#list></code>
index	Returns the 0-based index where the iteration (which is identified by the loop variable name) currently stands.	<code><#list ["a," "b," "c"] as i></code> <code> \${i?index}: \${i}</code> <code></#list></code>
is_first	Tells if the item where the iteration (which is identified by the loop variable name) currently stands is the first item.	<code><#list ["a," "b," "c"] as i></code> <code> \${i?is_first?c}</code> <code></#list></code>
is_last	Tells if the item where the iteration (which is identified by the loop variable name) currently stands is the last item.	<code><#list ["a," "b," "c"] as i></code> <code> \${i?is_last?c}</code> <code></#list></code>
has_next	Tells if the item where the iteration (which is identified by the loop variable name) currently stands is not the last item.	<code><#list ["a," "b," "c"] as i></code> <code> \${i?has_next?c}</code> <code></#list></code>
is_even_item	Tells if the item where the iteration (which is identified by the loop variable name) currently stands is an even item.	<code><#list ["a," "b," "c"] as i></code> <code> \${i?is_even_item?c}</code> <code></#list></code>

	is identified by the loop variable name) currently stands has an even 1-based index.	<code> \${i?is_even_item?c} </#list></code>
is_odd_item	Tells if the item where the iteration (which is identified by the loop variable name) currently stands has an odd 1-based index.	<code><#list ["a," "b," "c"] as i> \${i?is_odd_item?c} </#list></code>
c	This built-in must be used to print the number.	<code>?c</code>
true	boolean value “true”	<code><if condition == true> ... </if></code>
false	boolean value “false”	<code><if condition == false> ... </if></code>
gt	comparison operator “greater than”	<code><if obj?number gt 35> ... </if></code>
gte	comparison operator “greater than or equivalent”	<code><if obj?number gte 35> ... </if></code>
lt	comparison operator “less than”	<code><if obj?number lt 35> ... </if></code>
lte	comparison operator “less than or equivalent”	<code><if obj?number lte 35> ... </if></code>

3.1.2 Defining and using Variables

Two types of variables are defined in a template:

- **Plain variables:** They are accessible from everywhere in the template. You can create and replace these variables with the `assign` directive.

Name	Description	Example
assign	You can create a new variable or replace an existing variable.	<code><#assign FLAG="TRUE"> <#assign PI="3.14"></code>

- **Loop variables:** Loop variables are created automatically by directives, such as `list` (e.g., `x` in `<#list xs as x>...</#list>`), and they only exist between the start-tag and end-tag of the directive.

Name	Description	Example
list	The simplest form for listing a collection is	<code><#list users as user> \${user} </#list></code>

3.1.3 Number Formatting

These are the built-in number formatting variables that can be used to format numbers:

Name	Description	Example
abs	Gives the absolute value of a number.	x?abs, if x is -5, will evaluate to 5
round	Rounds to the nearest whole number. If the number ends with .5, then it rounds upward, i.e., toward positive infinity.	x?round, if x is 1.5, will evaluate to 2
floor	Rounds the number downward, i.e., toward negative infinity.	x?floor, if x is 1.5, will evaluate to 1
ceiling	Rounds the number upward, i.e., toward positive infinity.	x?ceiling, if x is 1.5, will evaluate to 2
string	Converts a number to a string.	<#assign x = 1.234> \${x?string}
?string(",##0.00")	Displaying two digits after decimal point.	<#assign total = 3343434> <#assign number = -3343434> \${total?string(",##0.00") } \${number?string(",##0.00") } Output: 3,343,434.00 -3,343,434.00
?string(",#0")	Comma separator	<#assign total = 123456789> \${total?string(",#0")} Output: 1,23,45,67,89
?string(",##0")	Comma separator	<#assign total = 123456789> \${total?string(",##0")} Output: 123,456,789

3.1.4 Date Formatting

These are the built-in date formatting variables that can be used to format date values:

Name	Description	Example
.now	Returns the current date-time	\$.now}

?date	Extracts date from .now	<code> \${.now?date}</code>
?time	Extracts time from .now	<code> \${.now?time}</code>
?string.dd	Returns string day of month from .now	<code> \${.now?string.dd}</code>
?string.MM	Returns string month of year from .now	<code> \${.now?string.MM}</code>
?string.MMM	Returns string month of year from .now	<code> \${.now?string.MMM}</code>
?string.yyyy	Returns string year from .now	<code> \${.now?string.yyyy}</code>
?string["dd.M M.yyyy, HH:mm"]	Returns string date with time	<code> \${.now?string["dd.MM.yyyy, HH:mm"]}</code> Output: 08.04.2003 21:24
?string["EEEE, MMMM dd, yyyy, hh:mm a"]	Returns string date with day of week and full month name	<code> \${.now?string["EEEE, MMMM dd, yyyy, hh:mm a"]}</code> Output: Tuesday, April 08, 2003, 09:24 PM
?string["EEE, MMM d, 'yy"]	Returns string date with day of week	<code> \${.now?string["EEE, MMM d, 'yy"]}</code> Output: Tue, Apr 8, '03

For more details, you can refer to FreeMarker template framework documentation from:

<https://freemarker.apache.org/docs/index.html>

3.2 Integrating Smarten Object Data into Template

This section explains how to integrate Smarten object data into the FreeMarker template.

3.2.1 Get Smarten Object Data—Full Data

This section shows how to use Smarten object data in the FreeMarker template. We have used “Yearly Sales” tabular report shown below in this example.

Year	GrossSales
2015	39497594.10
2016	56141853.47
2017	59748479.98
2018	63890412.68

TABULAR REPORT

The snippet code below iterates object data above and displays it.

```
<#list r170611e29cf as innerList>
<#list innerList as obj>
${obj}
</#list>
<br>
</#list>
```

In the example above,

r170611e29cf—This is the object id of the Smarten object that we want to integrate into the template. The user can get this object ID from the Smarten administration panel >> Repository section. They must use object ID without extension in template script. In the example above, the actual object ID of the tabular report is “r170611e29cf.rpt.” In FreeMarker script, the user should use “r170611e29cf” (without extension).

innerList—This is the user-defined list variable that is used to iterate loop in the template. This variable contains each row of Smarten object data passed to the template.

Output

Year	GrossSales
2015	39497594.10
2016	56141853.47
2017	59748479.98
2018	63890412.68

OUTPUT—FULL DATA

3.2.2 Get Smarten Object Data—Based on Row Number

This section shows how to get data of a specific row from Smarten object data. We have used the same object as in the example above.

The example below shows how to read second row (Year 2015 row) data from object data.

```
<#list r170611e29cf as innerList>
<if innerList?counter == 2><!-- Displaying content of second row only -->
<#list innerList as obj>
${obj}
</#list>
</if>
<br>
</#list>
```

Output

2015 39497594.10

OUTPUT—BASED ON ROW NUMBER

3.2.3 Get Smarten object Data—Based on Row Number and Column Number

This section shows how to get data of a specific row and column cell from Smarten object data. We have used the same object as in the example above.

The example below shows how to read fifth row and second column (Year 2018 row and Grosssales) data from object data.

```
<#list r170611e29cf as innerList>
<#list innerList as obj>
<if innerList?counter == 5><!-- Displaying content of fifth row -->
    <if obj?counter == 2><!-- Displaying content of second column -->
        ${obj}
    </if>
</if>
<br>
</list>
```

Output

63890412.68

OUTPUT—BASED ON ROW COLUMN NUMBER

3.3 Rendering Different Controls

This section shows how to render different controls in the FreeMarker template.

3.3.1 Showing Data in Table

This section shows how to render HTML table control using Smarten object data. We have used the same object as in the example above.

Below is the example script for showing data in table:

```
<html>
<body>
<table>
<tbody>
<#list r170611e29cf as innerList>
```

```
<tr>
<#list innerList as obj>
    <td>${obj}</td>
</#list>
</tr>
</#list>
</tbody>
</table>
</body>
</html>
```

Output

Year	GrossSales
2015	39497594.10
2016	56141853.47
2017	59748479.98
2018	63890412.68

OUTPUT—TABLE

3.3.2 Showing Data in Graph

This section shows how to render graph control using Smarten object data. We have used the same object as in the example above.

Below is the example script for showing data in a bar graph:

```
<html>
<head>
<script type="text/javascript" src="../js/am/amcharts.js"></script>
<script type="text/javascript" src="../js/am/serial.js"></script>
<script>
    var chart = AmCharts.makeChart("chartdiv", {
        "type": "serial",
        "theme": "none",
        "rotate": true,
        "dataProvider": [
            <#list a175ffac5d97 as outerList>
                <if outerList?is_last>
                {
                    <list outerList as obj>
                        <if obj?is_first>
                            "Team": "${obj}",
                        </if>
                        <if obj?is_last>

```

```
        "value": ${obj}
        </#if>
        </#list>
    }
    <#else>
    {
        <#list outerList as obj>

            <#if obj?is_first>

                "Team": "${obj}",

            </#if>

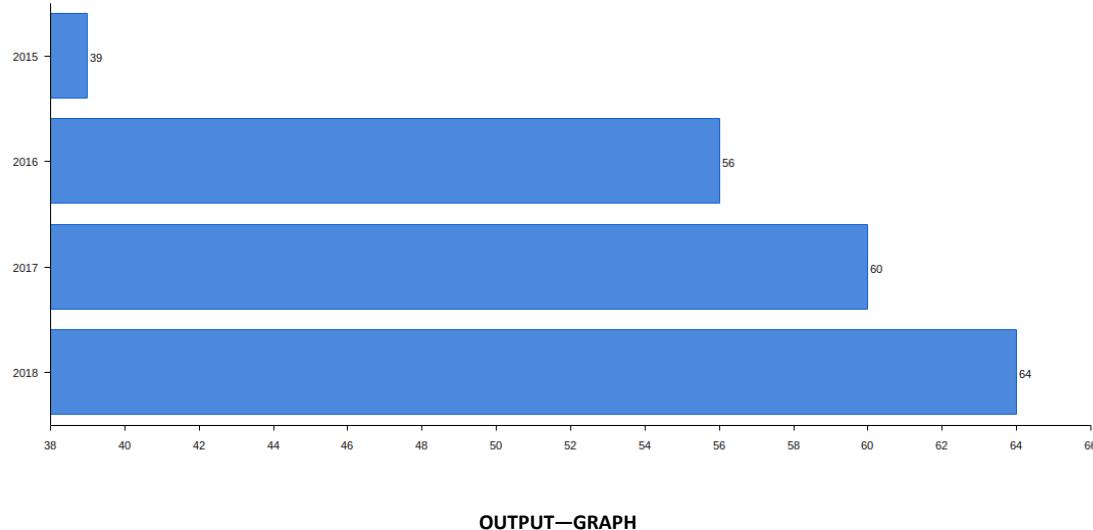
            <#if obj?is_last>

                "value": ${obj}

            </#if>
            </#list>
        },
        </#if>
        </#list>
    ],
    "startDuration": 0,
    "graphs": [
        {
            "labelText": "[[value]]",
            "balloonText": "<b>[[category]]: [[value]]</b>",
            "fillColorsField": "color",
            "fillAlphas": 0.7,
            "lineAlpha": 0.9,
            "lineColor": "#0D52D1",
            "type": "column",
            "valueField": "value"
        },
        {
            "categoryField": "Team",
            "borderAlpha": 0,
            "categoryAxis": {
                "gridPosition": "start",
                "gridAlpha": 0,
                "labelRotation": 45
            },
            "valueAxes": [
                {
                    "gridAlpha": 0,
                    "title": ""
                }
            ]
        });
    
```

```
</script>
</head>
<body>
<div id="chartdiv"></div>
</body>
</html>
```

Output



OUTPUT—GRAPH

3.4 Defining Report Page Formatting

This section explains how to define page formatting, such as page break, page header/footer, and page numbers.

3.4.1 Defining Page Break

This section shows an example for how to define page break in a report template.

The “page-break-before” property in CSS is used to achieve page break in PDF. This property helps to define page break in an HTML document when it is printed or outputted in PDF format. In the example below, we present a scenario of showing 48 records per page. After every 48 records, the next record will be printed on a new page. A page break code snippet is highlighted in bold in the template code below.

Example:

```
<html>
<head>
    <title>Ledger Account Demo</title>
<style>
    table { page-break-inside:auto }
    tr { page-break-inside:avoid; page-break-after:auto }
    thead { display:table-header-group }
```

```
</style>

</head>
<body>
<div style="max-width: 1000px; width: 100%; margin:0px auto; padding: 10px !important;">

<#assign indexCounter = 0>
<#assign counter = 48>
<div style="">
<table width="100%" border="0" cellspacing="0" cellpadding="2" style="font-family:Arial, 'sans-serif'; font-size: 8px;">

<#list r17472c10651 as innerList>
<if innerList?is_first>
    <thead style="display: table-row-group;">
        <tr>
            <th colspan="2" align="center" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid; ">${innerList[0]}</th>
            <th align="left" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid; ">${innerList[2]}</th>
            <th align="left" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid; ">${innerList[3]}</th>
            <th align="left" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid; ">${innerList[5]}</th>
            <th align="center" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid; padding-left:42px;">${innerList[7]}</th>
            <th align="center" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid; ">${innerList[8]}</th>
        </tr>
    </thead>
<else>
    <tr>
        <td align="center" >${innerList[0]}</td>
        <td align="left">Cr</td>
        <td><strong>${innerList[2]}</strong></td>
        <td align="left"><strong>${innerList[3]}</strong></td>
        <td align="center">${innerList[5]}&nbsp;</td>
        <td align="right" style="padding-right:55px">${innerList[7]}&nbsp;</td>
        <td align="left"></td>
    </tr>
<#assign indexCounter = indexCounter + 1>
<if indexCounter == counter>
<#assign counter = counter + 48>
<tr><!-- Display Carried Over at end -->
```

```
<td colspan="2" align="center">&nbsp;</td>
<td align="left">Carried Over</td>
<td colspan="2"></td>
<td align="center" style="border-top: 1px solid; border-bottom: 1px
solid">&nbsp;${innerList[9]}</td>
    <td align="center" style="border-top: 1px solid; border-bottom: 1px
solid;">&nbsp;${innerList[10]}</td>
</tr>

<!-- Page Break code start -->

</table>
</div>

<div style="page-break-before: always;">

    <table width="100%" border="0" cellspacing="0" cellpadding="2" style="font-
family:Arial, 'sans-serif'; font-size: 8px;">
        <thead style="display: table-row-group;">
            <tr>
                <th colspan="2" align="center" valign="middle" bgcolor="#FFFFFF"
style="border-top: 1px solid; border-bottom: 1px solid;">TimDate</th>
                <th align="left" valign="middle" bgcolor="#FFFFFF" style="border-
top: 1px solid; border-bottom: 1px solid;">ProductName</th>
                <th align="left" valign="middle" bgcolor="#FFFFFF" style="border-
top: 1px solid; border-bottom: 1px solid;">Type_Sale</th>
                <th align="left" valign="middle" bgcolor="#FFFFFF" style="border-
top: 1px solid; border-bottom: 1px solid;">Qty_Defect </th>
                <th align="center" valign="middle" bgcolor="#FFFFFF"
style="border-top: 1px solid; border-bottom: 1px solid; padding-
left:42px;">Amount_NetBeforeDisc</th>
                <th align="center" valign="middle" bgcolor="#FFFFFF"
style="border-top: 1px solid; border-bottom: 1px solid;">Amount_Net</th>
            </tr>
        </thead>
        <!-- Page Break code end -->

        <tr>
            <td colspan=7>&nbsp;</td>
        </tr>
        </#if>
        </#if>
    </#list>
</table>
</div>
</div>
```

```
</body>
</html>
```

Output

Sahil Desai	221.00	187.00	34.00												
Soham Desai	1446.00	187.00	178.50	186.50	186.00	154.50	169.00	152.50	111.50	118.50	8.50				
Summary	18402.75	2294.60	2326.00	2305.75	2549.75	2095.55	2194.80	2149.50	1381.90	1002.90	43.50				
Abhijit Ghosh	1552.75	188.00	169.00	177.50	192.25	176.00	184.00	185.00	141.00	134.00	8.50				
Disha Kavai	316.25							146.75	131.00	38.50					
Poorva Pandya	438.50	204.50	186.00	68.00											
Sanjay Bharad	1339.40	226.00	166.00	135.00	186.00	144.50	142.40	176.50	128.00	35.00					
Summary	3666.90	618.50	521.00	380.50	378.25	320.50	326.40	508.25	406.00	207.50	8.50				
Aastha Patel	812.90				187.25	129.25	168.75	133.75	111.65	82.25					
Aayushi Pawar	1786.50	173.00	189.00	237.50	216.00	225.00	218.50	221.50	189.00	117.00					

Amruta Chalke	1670.50	188.00	199.70	208.60	223.00	199.60	192.60	207.40	166.40	85.20					
Apama Verma	1149.20	74.00	169.50	139.50	187.20	131.75	92.25	130.50	124.00	80.50					
Bajju Trivedi	1852.50	220.00	211.50	219.00	230.00	215.00	200.00	211.00	193.00	153.00					
Bhumi Mehta	175.00								95.00	80.00					
Dhruv Suthar	1539.70	190.50	182.50	182.00	187.50	161.50	190.50	175.70	141.75	127.75					
Jignesh Patel	1527.25	172.50	163.50	174.50	210.25	166.00	173.00	178.50	151.00	138.00					
Mehulkumar Kalara	1344.80	195.05	197.75	183.50	184.25	154.00	157.75	179.00	93.50						
Mitalee Khanva	495.20	154.00	137.70	183.50											
Parth Dineshbhai Patel	1062.90	186.00	180.65	171.75	170.50	79.50	1.50	43.75	112.75	116.50					
Pranay Bhalerao	399.10	190.30	179.00	29.80											
Prasad Jadhav	1813.30	171.00	171.00	233.30	244.00	209.00	235.00	234.00	153.00	163.00					
Ritu Gupta	36.00									36.00					

REPORT—PAGE BREAK

3.4.2 Defining Page Header/Footer

This section explains how to include a page header and a footer in a report. The example below shows a template code for defining a page header and a footer. The code snippet is highlighted in bold.

Example:

```
<html>
  <head>
    <title>Ledger Account Demo</title>
    <style>
      table { page-break-inside:auto }
      tr { page-break-inside:avoid; page-break-after:auto }
      thead { display:table-header-group }
    </style>
  </head>
  <body>
    <div style="max-width: 1000px; width: 100%; margin:0px auto; padding: 10px !important;">
      <div style="text-align: center; width: 100%;">
        <table width="100%" border="0" cellspacing="0" cellpadding="0">
          <tbody>
            <tr>
              <td align="center" width="30%" style="font-family:Arial, 'sans-serif'; font-size:12px; line-height: 20px;">
                <b>ABC Industries Ltd.</b><br>
                <u>Ahmedabad</u>
              </td>
            </tr>
          </tbody>
        </table>
      </div>
    </div>
  </body>
</html>
```

```
<br>
<b> XYZ Enterprises</b><br>
Ledger Account <br>
Maninagar, Ahmedabad <br>
<p style="font-size:10px">1-Apr-2019 to 12-Feb-2020</p>
</td>
</tr>
<tr>
    <td width="92%" align="left" style="font-family:Arial, 'sans-serif'; font-size:8px; line-height: 20px; float: left">
        </td>
        <td width="8%" style="float: right; font-family:Arial, 'sans-serif'; font-size:8px;"> Page
1/<span class="spld"></span></td>
    </tr>
</tbody>
</table>
</div>
<#assign indexCounter = 0>
<#assign counter = 48>
<div style="">
    <table width="100%" border="0" cellspacing="0" cellpadding="2" style="font-family:Arial, 'sans-serif'; font-size: 8px;">
        <#list r17472c10651 as innerList>
        <if innerList?is_first>
            <thead style="display: table-row-group;">
                <tr>
                    <th colspan="2" align="center" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid; ">${innerList[0]}</th>
                    <th align="left" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid;">${innerList[2]}</th>
                    <th align="left" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid;">${innerList[3]}</th>
                    <th align="left" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid;">${innerList[5]}</th>
                    <th align="center" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid; padding-left:42px;">${innerList[7]}</th>
                    <th align="center" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid;">${innerList[8]}</th>
                </tr>
            </thead>
            <#else>
                <tr>
                    <td align="center" >${innerList[0]}</td>
                    <td align="left">Cr</td>
                    <td><strong>${innerList[2]}</strong></td>
                    <td align="left"><strong>${innerList[3]}</strong></td>
                    <td align="center">${innerList[5]}&nbsp;</td>
                    <td align="right" style="padding-right:55px">${innerList[7]}&nbsp;</td>
                </tr>
            </#else>
        </#list>
    </table>
</div>
```

```
<td align="left"></td>
</tr>
<#assign indexCounter = indexCounter + 1>
<if indexCounter == counter>
<#assign counter = counter + 48>
<tr>
    <!-- Display Carried Over at end -->
    <td colspan="2" align="center">&nbsp;</td>
    <td align="left">Carried Over</td>
    <td colspan="2"></td>
    <td align="center" style="border-top: 1px solid; border-bottom: 1px solid">&nbsp;${innerList[9]}</td>
    <td align="center" style="border-top: 1px solid; border-bottom: 1px solid;">&nbsp;${innerList[10]}</td>
</tr>
</table>
</div>
<!-- Footer code start -->
<div style="border-top: #bfbfbf 1px solid; font-size:10px;">
    <table width="100%" border="0" cellspacing="0" cellpadding="0" style="font-size:8px;">
        <tbody>
            <tr>
                <td style="text-align:left;">&nbsp;</td>
                <td style="text-align:right; font-family:Arial; font-size:8px;">&nbsp; continued ...</td>
            </tr>
        </tbody>
    </table>
</div>
<!-- Footer code end -->
<br>
<div style="page-break-before: always;">
    <!-- Header code start -->
    <div style="border-bottom: #bfbfbf 1px solid;">
        <table width="100%" border="0" cellspacing="0" cellpadding="0" style="font-size:10px;">
            <tbody>
                <tr>
                    <td style="display: inline-block; width=170" style="text-align:left; font-family:Arial; font-size:10px;"><b>ABC Industries Ltd.</b><br></td>
                </tr>
                <tr>
                    <td style="text-align:left; font-family:Arial; font-size:10px;">XYZ Enterprises Ledger</td>
                </tr>
            <tbody>
                <tr>
                    <td style="text-align:left; font-family:Arial; font-size:10px;">Account : 1-Apr-2019 to 12-Feb-2020</td>
                </tr>
            </tbody>
        </table>
    </div>
    <!-- Header code end -->
```

```

<table width="100%" border="0" cellspacing="0" cellpadding="2" style="font-family:Arial,
'sans-serif'; font-size: 8px;">
    <thead style="display: table-row-group;">
        <tr>
            <th colspan="2" align="center" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid;">TimDate</th>
            <th align="left" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid;">ProductName</th>
            <th align="left" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid;">Type_Sale</th>
            <th align="left" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid;">Qty_Defect</th>
            <th align="center" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid; padding-left:42px;">Amount_NetBeforeDisc</th>
            <th align="center" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid;">Amount_Net</th>
        </tr>
    </thead>
    <!-- Page Break code end -->
    <tr>
        <td colspan=7>&nbsp;</td>
    </tr>
    </#if>
    </#if> </#list>
    </table>
</div> </div>
</body> </html>

```

Output

TimDate	ProductName	Type_Sale	Qty_Defect	Amount_NetBeforeDisc	Amount_Net
10-May-2008	Cr	15 inch Laptop	back log	128	2594.00
13-May-2008	Cr	15 inch Laptop	back log	146	4148.00
15-May-2008	Cr	15 inch Laptop	back log	199	4203.00
20-May-2008	Cr	15 inch Laptop	back log	142	2595.00
23-May-2008	Cr	15 inch Laptop	back log	196	3464.00
26-May-2008	Cr	15 inch Laptop	back log	152	4693.00
26-May-2008	Cr	15 inch Laptop	back log	141	4925.00
17-Jun-2008	Cr	15 inch Laptop	back log	150	3686.00
20-Jun-2008	Cr	15 inch Laptop	back log	135	4021.00
23-Jun-2008	Cr	15 inch Laptop	back log	171	4406.00
24-Jun-2008	Cr	15 inch Laptop	back log	143	4591.00
10-Jul-2008	Cr	15 inch Laptop	back log	158	4118.00
12-Jul-2008	Cr	15 inch Laptop	back log	101	3303.00
13-Jul-2008	Cr	15 inch Laptop	back log	133	4460.00
18-Jul-2008	Cr	15 inch Laptop	back log	161	4109.00
18-Jul-2008	Cr	15 inch Laptop	back log	156	4388.00
02-Aug-2008	Cr	15 inch Laptop	back log	105	3517.00
05-Aug-2008	Cr	15 inch Laptop	back log	135	4581.00
05-Aug-2008	Cr	15 inch Laptop	back log	113	3278.00
07-Aug-2008	Cr	15 inch Laptop	back log	121	3989.00
14-Aug-2008	Cr	15 inch Laptop	back log	105	3525.00
15-Aug-2008	Cr	15 inch Laptop	back log	191	4772.00
15-Aug-2008	Cr	15 inch Laptop	back log	110	3628.00
15-Aug-2008	Cr	15 inch Laptop	back log	110	2820.00
27-Aug-2008	Cr	15 inch Laptop	back log	116	3677.00
07-Sep-2008	Cr	15 inch Laptop	back log	183	3840.00
08-Sep-2008	Cr	15 inch Laptop	back log	189	3677.00
10-Sep-2008	Cr	15 inch Laptop	back log	132	3835.00
18-Sep-2008	Cr	15 inch Laptop	back log	206	7606.00
20-Sep-2008	Cr	15 inch Laptop	back log	109	2590.00
20-Sep-2008	Cr	15 inch Laptop	back log	144	2859.00
20-Sep-2008	Cr	15 inch Laptop	back log	191	3246.00
24-Sep-2008	Cr	15 inch Laptop	back log	176	3112.00
25-Sep-2008	Cr	15 inch Laptop	back log	179	4126.00
25-Sep-2008	Cr	15 inch Laptop	back log	108	2761.00
26-Sep-2008	Cr	15 inch Laptop	back log	111	3854.00
26-Sep-2008	Cr	15 inch Laptop	back log	186	3854.00
29-Sep-2008	Cr	15 inch Laptop	back log	138	4075.00
07-Oct-2008	Cr	15 inch Laptop	back log	109	3739.00
11-Oct-2008	Cr	15 inch Laptop	back log	104	3318.00
11-Oct-2008	Cr	15 inch Laptop	back log	174	4372.00
15-Oct-2008	Cr	15 inch Laptop	back log	101	2654.00
19-Oct-2008	Cr	15 inch Laptop	back log	190	2784.00
21-Oct-2008	Cr	15 inch Laptop	back log	138	2526.00
25-Oct-2008	Cr	15 inch Laptop	back log	173	3496.00
27-Oct-2008	Cr	15 inch Laptop	back log	101	3850.00
28-Oct-2008	Cr	15 inch Laptop	back log	157	4932.00
07-Nov-2008	Cr	15 inch Laptop	back log	122	3621.00

continued ...

3.4.3 Defining Page Numbers

This section shows how to define page numbers in a page header or a footer of a report.

In the example below, we show the page number in the page header of a report. The page number is shown in “Page *n* / Total pages” format. We need to follow the steps mentioned below to include the page number.

1. Define “pageNum” variable in the FreeMarker template as shown below.

```
<#assign pageNum = 1>
```

2. Then in the template code where you want to display the total number of pages, you need to insert a span tag as shown below.

```
<span class="spId"></span>
```

3. After that please add the JavaScript snippet below in the template code. This script assigns the total number of pages value dynamically to the span tag.

```
<script type="text/javascript">
    var cols = document.getElementsByClassName('spId');

    for(i = 0; i < cols.length; i++) {
        cols[i].innerText = ${pageNum};
    }
</script>
```

In the template example below, we have highlighted the code snippet explained in the steps above.

Example:

```
<html> <head>
    <title>Ledger Account Demo</title>
<style>
    table { page-break-inside:auto }
    tr { page-break-inside:avoid; page-break-after:auto }
    thead { display:table-header-group }
</style>
</head>
<body>
    <div style="max-width: 1000px; width: 100%; margin:0px auto; padding: 10px !important;">
        <div style="text-align: center; width: 100%;">
            <table width="100%" border="0" cellspacing="0" cellpadding="0">
                <tbody>
                    <tr>
                        <td align="center" width="30%" style="font-family:Arial, 'sans-serif'; font-size:12px; line-height: 20px;">
                            <b>ABC Industries Ltd.</b><br>
                            <u>Ahmedabad</u>
                            <br>
                            <b>XYZ Enterprises</b><br>
                        </td>
                </tbody>
            </table>
        </div>
    </div>
</body>
</html>
```

```

        Ledger Account <br>
        Maninagar, Ahmedabad <br>
        <p style="font-size:10px">1-Apr-2019 to 12-Feb-2020</p>
        </td>
    </tr>
    <tr>
        <td width="92%" align="left" style="font-family:Arial, 'sans-serif';
font-size:8px; line-height: 20px; float: left">
        </td>
        <td width="8%" style="float: right; font-family:Arial, 'sans-serif'; font-
size:8px;"> Page 1/<span class="spId"></span></td>
        </tr>
    </tbody>
</table>
</div>

<#assign indexCounter = 0>
<#assign fourCounter = 40>
<#assign pageNum = 1>

<div style="">

    <table width="100%" border="0" cellspacing="0" cellpadding="2"
style="font-family:Arial, 'sans-serif'; font-size: 8px;">
        <#list r17472c10651 as innerList>
            <if innerList?is_first>
                <thead style="display: table-row-group;">
                    <tr>
                        <th colspan="2" align="center" valign="middle"
bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid;">${innerList[0]}</th>
                        <th align="left" valign="middle" bgcolor="#FFFFFF"
style="border-top: 1px solid; border-bottom: 1px solid;">${innerList[2]}</th>
                        <th align="left" valign="middle" bgcolor="#FFFFFF"
style="border-top: 1px solid; border-bottom: 1px solid;">${innerList[3]}</th>
                        <th align="left" valign="middle" bgcolor="#FFFFFF"
style="border-top: 1px solid; border-bottom: 1px solid;">${innerList[5]}</th>
                        <th align="center" valign="middle"
bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid; padding-
left:42px;">${innerList[7]}</th>
                        <th align="center" valign="middle"
bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid;">${innerList[8]}</th>
                    </tr>
                </thead>
            <#else>
                <tr>
                    <td align="center" >${innerList[0]}</td>
                    <td align="left">Cr</td>
                    <td><strong>${innerList[2]}</strong></td>

```

```
<td align="left"><strong>${innerList[3]}</strong></td>
<td align="center">${innerList[5]} &nbsp;</td>
<td align="right" style="padding-
right:55px">${innerList[7]}&nbsp;</td>
<td align="left"></td>
</tr>
<#assign indexCounter = indexCounter + 1>
<if indexCounter == fourCounter>
<#assign fourCounter = fourCounter + 48>

</table>
</div>
<!-- Footer code start -->
<div style="border-top: #bfbfbf 1px solid;font-size:10px;">
<table width="100%" border="0" cellspacing="0" cellpadding="0" style="font-
size:8px;">
<tbody>
<tr>
<td style="text-align:left;">&nbsp;</td>
<td style="text-align:right; font-family:Arial; font-
size:8px;">&nbsp; continued ...</td>
</tr>
</tbody>

</table>

</div>

<!-- Footer code end -->

<br>

<div style="page-break-before: always;">
<#assign pageNum = pageNum + 1>    <!-- Manually incrementing page number
(When new page starts) -->

<!-- Header code start -->
<div style="border-bottom: #bfbfbf 1px solid;">
<table width="100%" border="0" cellspacing="0" cellpadding="0" style="font-
size:10px;">
<tbody>
<tr>
<td style="display: inline-block;" width="170" style="text-
align:left; font-family:Arial; font-size:10px;"><b>ABC Industries Ltd.</b><br></td>
</tr>
<tr>
<td style="text-align:left; font-family:Arial; font-
size:10px;">XYZ Enterprises Ledger Account : 1-Apr-2019 to 12-Feb-2020</td>
```

```
<td style="text-align: center; font-family: Arial; font-size: 8px;">Page  
${pageNum} / <span class="spId"></span></td>  
  
</tr>  
</tbody>  
  
</table>  
  
</div>  
  
<!-- Header code end -->  
  
<table width="100%" border="0" cellspacing="0" cellpadding="2" style="font-family: Arial, 'sans-serif'; font-size: 8px;">  
    <thead style="display: table-row-group;">  
        <tr>  
            <th colspan="2" align="center" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid;">TimDate</th>  
            <th align="left" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid;">ProductName</th>  
            <th align="left" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid;">Type_Sale</th>  
            <th align="left" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid;">Qty_Defect</th>  
            <th align="center" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid; padding-left: 42px;">Amount_NetBeforeDisc</th>  
            <th align="center" valign="middle" bgcolor="#FFFFFF" style="border-top: 1px solid; border-bottom: 1px solid;">Amount_Net</th>  
        </tr>  
        </thead>  
        <tr>  
            <td colspan="7">&nbsp;</td>  
        </tr>  
        <#if> </#if> </#list>  
    </table> </div> </div> </body> </html>  
  
<script type="text/javascript">  
    var cols = document.getElementsByClassName('spId');  
    for(i = 0; i < cols.length; i++) {  
        cols[i].innerText = ${pageNum};  
    }  
</script>
```

Output

ABC Industries Ltd. XYZ Enterprises Ledger Account : 1-Apr-2019 to 12-Feb-2020						Page 2/23
TimeDate	ProductName	Type_Sale	Qty_Defect	Amount_NetBeforeDisc	Amount_Net	
10-May-2008	Cr	15 Inch Laptop	back log	128	2594.00	
13-May-2008	Cr	15 Inch Laptop	back log	146	4148.00	
19-May-2008	Cr	15 Inch Laptop	back log	199	4203.00	
20-May-2008	Cr	15 Inch Laptop	back log	142	2595.00	
23-May-2008	Cr	15 Inch Laptop	back log	198	3496.00	
28-May-2008	Cr	15 Inch Laptop	back log	152	4093.00	
29-May-2008	Cr	15 Inch Laptop	back log	141	4925.00	
17-Jun-2008	Cr	15 Inch Laptop	back log	150	3686.00	
20-Jun-2008	Cr	15 Inch Laptop	back log	135	4021.00	
23-Jun-2008	Cr	15 Inch Laptop	back log	171	4406.00	
24-Jun-2008	Cr	15 Inch Laptop	back log	143	4591.00	
10-Jul-2008	Cr	15 Inch Laptop	back log	158	4118.00	
12-Jul-2008	Cr	15 Inch Laptop	back log	101	3303.00	
13-Jul-2008	Cr	15 Inch Laptop	back log	133	4460.00	
18-Jul-2008	Cr	15 Inch Laptop	back log	161	4109.00	
18-Jul-2008	Cr	15 Inch Laptop	back log	156	4388.00	
03-Aug-2008	Cr	15 Inch Laptop	back log	165	3517.00	
05-Aug-2008	Cr	15 Inch Laptop	back log	125	4561.00	
05-Aug-2008	Cr	15 Inch Laptop	back log	113	3278.00	
07-Aug-2008	Cr	15 Inch Laptop	back log	121	3969.00	
14-Aug-2008	Cr	15 Inch Laptop	back log	165	3562.00	
15-Aug-2008	Cr	15 Inch Laptop	back log	191	4772.00	
15-Aug-2008	Cr	15 Inch Laptop	back log	110	3628.00	
19-Aug-2008	Cr	15 Inch Laptop	back log	110	2820.00	
27-Aug-2008	Cr	15 Inch Laptop	back log	116	3677.00	
07-Sep-2008	Cr	15 Inch Laptop	back log	183	3840.00	
08-Sep-2008	Cr	15 Inch Laptop	back log	189	3677.00	
10-Sep-2008	Cr	15 Inch Laptop	back log	132	3835.00	
18-Sep-2008	Cr	15 Inch Laptop	back log	206	7606.00	
20-Sep-2008	Cr	15 Inch Laptop	back log	109	2590.00	
20-Sep-2008	Cr	15 Inch Laptop	back log	144	2859.00	
20-Sep-2008	Cr	15 Inch Laptop	back log	191	3246.00	
24-Sep-2008	Cr	15 Inch Laptop	back log	176	3112.00	
25-Sep-2008	Cr	15 Inch Laptop	back log	179	4126.00	
25-Sep-2008	Cr	15 Inch Laptop	back log	168	2761.00	
26-Sep-2008	Cr	15 Inch Laptop	back log	111	3854.00	
28-Sep-2008	Cr	15 Inch Laptop	back log	198	3832.00	
29-Sep-2008	Cr	15 Inch Laptop	back log	138	4075.00	
07-Oct-2008	Cr	15 Inch Laptop	back log	108	3739.00	
11-Oct-2008	Cr	15 Inch Laptop	back log	104	3318.00	
11-Oct-2008	Cr	15 Inch Laptop	back log	174	4372.00	
15-Oct-2008	Cr	15 Inch Laptop	back log	101	2654.00	
19-Oct-2008	Cr	15 Inch Laptop	back log	190	2784.00	
21-Oct-2008	Cr	15 Inch Laptop	back log	138	2526.00	
25-Oct-2008	Cr	15 Inch Laptop	back log	173	3496.00	
27-Oct-2008	Cr	15 Inch Laptop	back log	101	3850.00	
28-Oct-2008	Cr	15 Inch Laptop	back log	157	4932.00	
02-Nov-2008	Cr	15 Inch Laptop	back log	122	3621.00	

continued ...

REPORT PAGE NUMBER

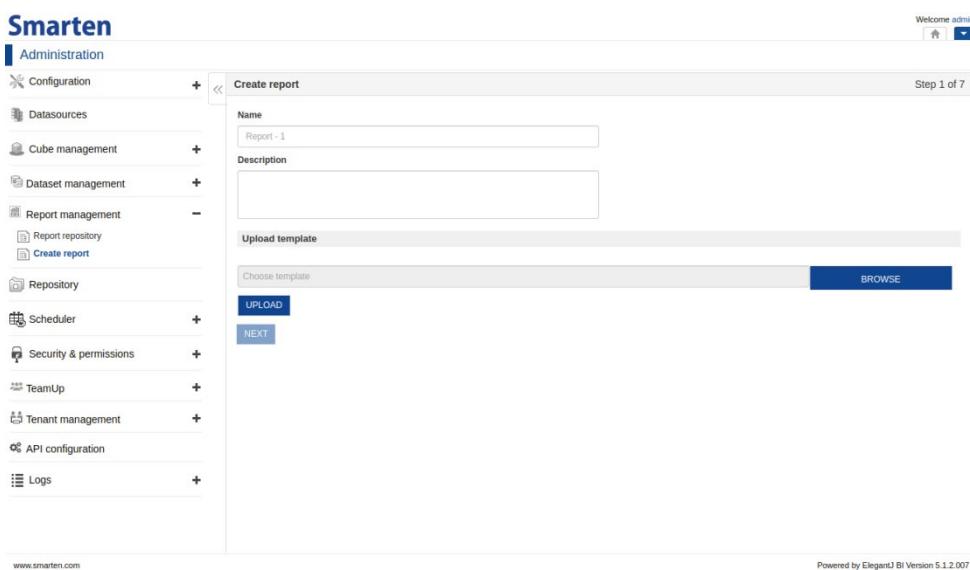
4 Creating Report

Smarten provides a step-by-step wizard for creating a print report. The following steps are involved to create a print report:

1. Upload the FreeMarker report template, which is created as explained in the sections above.
2. Associate BI objects
3. Preview report output
4. Define page filters and retrieval parameters
5. Assign permissions
6. Define export parameters

4.1 Upload Report Template

The user can create a FreeMarker template as explained in the sections above. Once the template is ready, the user can create a new report from the Smarten administrator panel. In the first step, the user needs to upload a created report template.



REPORT—UPLOAD REPORT TEMPLATE

4.2 Associate BI Objects

Once the report template is uploaded, the user needs to select Smarten BI objects, which are used in the FreeMarker template script to fetch data. Based on the report template, the user can select and associate one or more BI objects to the report. The system fetches data from these objects and generates the report based on the report template.

Note:

Currently, Smarten allows the user to associate Crosstab and Tabular BI objects only.

It is also recommended that associated BI object's metadata or column sequence should not be changed after creating a print report. This may lead to the wrong reference of object data and column sequence and may generate a wrong print report. It is also recommended to give view only permission for associated BI objects to users to avoid any user modifying the report.

For more than one object, the system provides a mechanism to set a relationship between two objects for a master-detail data kind of scenario. For example, let us take a credit card statement report scenario. For this report, two objects will be required. One is CustomerDetails, and the other is TransactionDetails. The CustomerDetails object will have customer-specific master details, and TransactionDetails will have transactions done by customers. Now, a credit card statement should be generated for each customer, and there should be a separate statement for each customer. For such a scenario, we need to specify a master-detail relationship between the two objects, and the system can fetch data accordingly to generate a report. So here, you should select the CustomerDetails object as the master object and CustID as the master ID column. TransactionDetail will be considered as the detail object, and you need to map its CustID column with the master ID column. So, based on this definition, the system will fetch data of each customer from the master object, and the detail object and will generate the report.

Note:

Currently, Smarten allows mapping of a single key column. If you have a multiple keys column scenario, i.e., combined keys, you should combine key columns into one column in your dataset using SSDP.

For more than one object, if you don't want to define a master-detail relationship between objects, you just need to select NONE as the master object. So, there will not be any relationship between objects, and the system fetches data accordingly and will generate a report. For example, let's take a company annual report in which you want to show, for example, a P/L statement, balance sheet, and performance of a key product segment. You can associate required objects to the report. Here, there is no relationship between objects, so you need not specify the master object and mapping column. The system will take individual object data accordingly and generate a report as per template logic.

The screenshot shows the 'Create report' wizard in the Smarten application. The left sidebar lists various administration modules like Configuration, Datasources, Cube management, etc. The main panel is titled 'Create report' and is at 'Step 1 of 7'. It has fields for 'Name' (Credit Card Statement) and 'Description'. Below these, an 'Uploaded template' section shows 'creditCardDemo.fl'. The 'Select objects' section contains a 'Select folder' dropdown and a list of objects under 'Objects': Customer Master, T Details w/ CardNo w/o T type, Transaction Details, and Transaction Details with Card No. An 'ADD' button is available to add more objects. The 'Selected objects' section shows the same four objects listed. At the bottom, 'Select master object' is set to 'Customer Master' and 'Select master column' is set to 'Cust_ID'. A 'Column mapping' table shows 'Transaction Details' mapped to 'Cust_ID'. A 'NEXT' button is at the bottom right.

REPORT—ASSOCIATE BI OBJECTS

4.3 Preview

Once object association with template is done, the system will generate an actual report preview using limited data. Users can verify their report output and confirm whether it is proper or not.

Smarten

Administration

Create report - report preview Step 2 of 7



BANK N.A.,
Acropolis,
New Door No.148(Old No.68),
Dr. Radhakrishnan Salai, Mylapore,
Chennai-600 004.
Tamil Nadu GSTIN - 33AAACC0462F2ZA

BANK REWARDS PLATINUM CARD
Statement for Bank Card Number *****1234
Statement Period: 21 February 2019 to 20 March 2019

At a glance		Account Summary	Update Contact Details
Statement Date:	20/03/19	Previous balance: Rs.857917.6	Customer Name Sachin Shah
Total Amount Due:	Rs.42691.4	Current Purchases & Other Charges Rs.0	To update your contact details, login to www.bank.com/india with your Internet Password (IPIN) and click on the links under "Manage your account".
Minimum Amount Due:	Rs.13500000	Current Cash Advance: Rs.775137.2	To self-select IPIN (Internet Password) instantly Click here
Due Date:	05/03/27.6	Last Payment Received: Rs.1651540	Pay Online: - Bank Online - E-Pay/NEFT
Credit Limit:	Rs.12646180	Points Earned: 0	Highlights
Available Credit Limit:	Rs.1080000		
Available Cash Limit	Rs.771047.2		

Did you know that this statement also contains details on

Powered by ElegantJ BI Version 5.1.2.007

www.smarten.com

Powered by ElegantJ BI Version 5.1.2.007

REPORT—PREVIEW

4.4 Define Page Filters and Retrieval Parameters

Users can define appropriate page filters or retrieval parameters for filtering the data.

Smarten

Administration

Create report - page filters Step 4 of 7

Select page filter

Cube/Dataset

Common columns - all cubes/datasets

Available columns

	Selected columns
Cust_ID	+ \$ CustomerName [Common columns - all cubes/datasets]
CustomerName	+ \$

SELECT DEFAULT VALUES

CustomerName [Common columns - all cubes/datasets] CustomerName(0)

NEXT CANCEL BACK

Powered by ElegantJ BI Version 5.1.2.007

www.smarten.com

Powered by ElegantJ BI Version 5.1.2.007

REPORT—CONFIGURE PAGE FILTER

4.5 Assign Permissions

The user can assign permissions to appropriate users. Permitted users will be able to see the report in the object repository along with other BI objects. They can open, filter, and export a report.

Welcome admin

Step 5 of 7

Configuration

Datasources

Cube management

Dataset management

Report management

Report repository

Create report

Repository

Scheduler

Security & permissions

TeamUp

Tenant management

API configuration

Logs

Create report - report permissions

Select users

All groups

Available users

Selected users

test

\$ admin

NEXT CANCEL BACK

Powered by ElegantJ BI Version 5.1.2.007

REPORT—PERMISSIONS

4.6 Define Export Parameters

The user can define export parameters, such as PDF page size, orientation, export file naming pattern, and export in a single file or separate files.

For a master object scenario, Smarten provides two options to generate an export file:

Generate a separate file for each record—The system will generate a separate file for each master object record. For a credit card statement example, it will generate a separate file for each customer.

Generate single file for all records—The system will generate a single file for all records. In a master object scenario, each master record will start from a new page. For a credit card example, statements for all customers will be exported in a single PDF file, and each customer statement will start from a new page.

For a non-master object scenario, only this option will be applicable. For a company annual report example, it will be exported in a single file only.

In case of a large single file, the system will break it into small chunks. For a ledger report example, if there are a large number of transactions and it creates thousands of pages, the system will generate multiple export files by breaking it based on the number of records per file.

File name pattern predefined tags

The following tags can be used to create export file name pattern:

Tag	Description
\$ReportName\$	Returns the name of the report

\$DD\$	Returns the day part of current system date. If system date is 01-Mar-2021, returns 01
\$MM\$	Returns the month part of current system date. If system date is 01-Mar-2021, returns 03
\$MMM\$	Returns the short form of month name of current system date. If system date is 01-Mar-2021, returns Mar
\$YY\$	Returns the 2-digit year part of current system date. If system date is 03-Mar-2021, returns 21
\$YYYY\$	Returns the 4-digit year part of current system date. If system date is 03-Mar-2021, returns 2021
\$HH\$	Returns the hour part of system date. If system date is 01-Mar-2021 10:30:50, returns 10
\$mm\$	Returns the minutes part of system date. If system date is 01-Mar-2021 10:30:50, returns 30
\$ss\$	Returns the seconds part of system date. If system date is 01-Mar-2021 10:30:50, returns 50
\$master-object-column-name\$	These tags are available only when a Master object is selected in the report definition. All master objects' dimensions are available as tags. For example, if it is a CustomerDetails object and custID, custName as dimensions, \$custID\$ and \$custName\$ will be the tags.

For example,

1. For a credit card statement example, if you want to generate an export file name as shown in the pattern below:

ABC Bank-Card Statement-0001001-20210304

Where 0001001 is customer id and 20210304 is statement generation date

The pattern below should be specified in the report:

ABC Bank-\$ReportName\$-\$CustomerID\$-\$YYYY\$\$MM\$\$DD\$

2. For an annual report example, if you want to generate an export file name as shown below:

ABC Pharma-Annual Report-2021

The pattern will be:

ABC Pharma-\$ReportName\$-\$YYYY\$

3. For an outstanding customer report, if you want to generate an export file pattern as shown below:

XYZ LTD-Outstanding Report-Feb2020

The pattern will be:

\$CustomerName\$-\$ReportName\$-\$MMM\$\$YYYY\$

Create report - export properties Step 6 of 7

Generate separate file(s) for each record
 Generate single file for all records
Large files will be automatically broken in chunks

PDF setup
Page size: A4
Orientation:
 Portrait
 Landscape

Export file name pattern
\$ReportName\$
Example:
Credit Card Statement

NEXT **CANCEL** **BACK**

REPORT—EXPORT PARAMETERS

5 Product and Support Information

Find more information about Smarten and its features at www.smarten.com

Support: support@smarten.com

Sales: sales@smarten.com

Feedback & Suggestions: support@smarten.com

Support & Knowledgebase Portal: support.smarten.com