```
<?xml version="1.0" encoding="iso-8859-1"?>

<!DOCTYPE xsl:stylesheet

[<!ENTITY nbsp " ">]

>

<!--

File:           DischargeMonitoringReport.xsl

Description:    This XSLT stylesheet outputs XML DMRs conforming to
NetDMRInternal_DischargeMonitoringReport_v1.0.xsd

as formatted HTML.

Author:         Easter Research Group, Inc.


Last Changed:   06/16/2008

-->

<xsl:stylesheet id="root_sst" version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:output method="xml" version="1.0" encoding="iso-8859-1" omit-xml-declaration="yes"
standalone="no"

doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN"

doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd" indent="yes"
/>


<!-- This template matches the XML document's root node -->

<xsl:template match="/">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<meta http-equiv="Content-Style-Type" content="text/css" />

<title>DMR Copy of Record</title>


<!-- Note that all needed styles must be declared inline in the HTML result -->

<style type="text/css">

@page{ size: 17in 11in;

margin: 0.2in;

}

.dmr_header_footer_tbl {

width:100%;

border-collapse:collapse;

}

.dmr_header_footer_tbl tr:hover {

background: #D1E9E9;

}

.dmr_header_footer_tbl td {
```

```
padding:3px;
}
.dmr_header_footer_tbl td.border_right {
border-right:1px solid #000000;
}
.dmr_header_footer_tbl td.padding_left {
padding-left:25px;
}
.dmr_header_footer_tbl td.section_row {
background-color: #e0e0e0;
height:16px;
vertical-align:bottom;
}
.datatable td.section_row {
background-color: #e0e0e0;
height:12px;
vertical-align:bottom;
padding-bottom:3px;
}
#header_text {
color: #2D4E6A;
font-weight: bold;
}
.header_text {
color: #2D4E6A;
font-weight: bold;
}
.datatable {
border-collapse: collapse;
border: 1px solid #d1e9e9;
color: #000;
background: #fff;
cell-spacing: 0px;
font-size: 10px;
}
.datatable th {
background-color:#B3D0D9;
color: #2D4E6A;
}
```

```
.datatable td {
border: 1px solid #e0e0e0;
}
.datatable #tfoot {
border: 1px solid #e0e0e0;
text-align: center;
}
td.centeredtext {
text-align:center;
font-size: 9px;
padding-top: 1px;
padding-right: 0px;
padding-bottom: 1px;
padding-left: 0px;
}
th.centeredtext {
text-align:center;
font-size: 9px;
padding-top: 1px;
padding-right: 0px;
padding-bottom: 1px;
padding-left: 0px;
}
th.centeredtext_and_vert, td.centeredtext_and_vert {
text-align:center;
vertical-align:middle;
}
td.valignmiddle {
vertical-align:middle;
font-size: 9px;
padding-top: 1px;
padding-right: 0px;
padding-bottom: 1px;
padding-left: 0px;
}
th.valignmiddle {
vertical-align:middle;
font-size: 9px;
padding-top: 1px;
```

```
padding-right: 0px;
padding-bottom: 1px;
padding-left: 0px;
}

#container {
background-color: #FFFFFF;
text-align: left;
width:100%;
min-width: 800px;
}
#programming_content {
padding-top: 0px;
padding-right: 20px;
padding-bottom: 0px;
padding-left: 20px;
}
#content_wide {
background-color: #FFFFFF;
text-align: left;
margin: 0px 20px 0px 20px;
min-width: 100%;
}
#sub_content_no_top_padding {
padding-top: 0px;
padding-bottom: 0px;
padding-left: 8px;
padding-right: 8px;
}
#standard_hover tr:hover {
background: #ffffd9;
}
body {
background-color: #0E6794;
margin: 0px;
padding: 0px;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 12px;
height: 100%;
}
```

```
table {
font-size: 11px;
}
h1 {
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 16px;
font-weight: 600;
color: #0E6794;
}
h2 {
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 15px;
font-weight: 600;
color: #0E6794;
}
h3 {
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 13px;
font-weight: 550;
color: #333333;
}
a:link {
color: #0E6794;
}
a:hover {
color: #AF7347;
text-decoration: none;
}
hr {
border: none 0;
border-top: 1px dotted #b3d0d9;
border-bottom: 1px solid #b3d0d9;
width: 100%;
height: 2px;
margin: 10px auto 0 0;
text-align: left;
}
label {
display: block;
```

```
width: 300px;
float: left;
margin-bottom: 10px;
}
br {
clear: left;
}
caption {
padding: 0 0.6em 0.8em;
font-size: 1.3em;
font-weight: bold;
text-align: left;
color: #000;
background: transparent;
}
td, th{
padding: 0.5em;
}
thead th {
border: 1px solid #e0e0e0;
text-align: left;
font-size: 1em;
font-weight: bold;
background: #B3D0D9;
}
tfoot th, tfoot td {
border: 1px solid #e0e0e0;
text-align: left;
font-size: 1em;
background: #e8e8cf;
}
tfoot th {
font-weight: bold;
}
tbody td a {
background: transparent;
color: #00c;
text-decoration: underline;
}
```

```
a:hover {
background: transparent;
color: #00c;
text-decoration: underline;
}
tbody th a {
background: transparent;
color: #3f7c5f;
text-decoration: underline;
font-weight: bold;
}
tbody th a:visited {
color: #b98b00;
}
tbody th, tbody td {
vertical-align: top;
text-align: left;
}
tfoot td {
border: 1px solid #996;
}
.style1 {
font-size: 16px;
font-weight: bold;
color: #CE2633;
}
</style>
</head>

<body style="border: 0pt none; margin: 0pt;">
<table id="container" style="padding: 0pt;" align="center" border="0" cellpadding="0"
cellspacing="0">
<tbody>
<tr>
<td bgcolor="#0e6794" width="5"></td>
<td style="padding: 0pt;" bgcolor="#ffffff">
<div id="programming_content">
<table border="0" cellpadding="0" cellspacing="0">
<tbody>
<tr>
```

```
<td>
<div id="content_wide">
<div id="sub_content_no_top_padding">
<h1>DMR Copy of Record</h1>
<div id="dmr_form">
<div id="dmr_header" style="background-color: rgb(209, 233, 233); width: 100%;">
<div id="dmr_header_expanded" style="display: block;">
<table class="dmr_header_footer_tbl">
<tbody>
<tr>
<td colspan="6" class="section_row" style="height: 14px;">
<span class="header_text">
<em>Permit</em>
</span>
</td>
</tr>
<tr>
<td class="header_text" nowrap="nowrap">Permit #:</td>
<td class="border_right">
<strong>
<xsl:value-of select="/DischargeMonitoringReport/PermitIdentifier"/>
</strong>
</td>
<td class="header_text" nowrap="nowrap">Permittee:</td>
<td class="border_right">
<xsl:value-of select="/DischargeMonitoringReport/PermitteeName"/>
</td>
<td class="header_text" nowrap="nowrap">Facility:</td>
<td>
<xsl:choose>
<xsl:when test="string-length(/DischargeMonitoringReport/FacilityName) > 0">
<xsl:value-of select="/DischargeMonitoringReport/FacilityName"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
</tr>
```

```
<tr>
<td class="header_text">Major:</td>
<td class="border_right">
<xsl:choose>
<xsl:when test="/DischargeMonitoringReport/MajorStatusIndicator='Y'">
Yes
</xsl:when>
<xsl:when test="/DischargeMonitoringReport/MajorStatusIndicator='N'">
No
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="header_text" nowrap="nowrap">Permittee Address:</td>
<td class="border_right">
<xsl:if test="string-length(/DischargeMonitoringReport/PermitteeAddress/AttentionTo) >
0">
Attn: <xsl:value-of select="/DischargeMonitoringReport/PermitteeAddress/AttentionTo"/>
<xsl:if
test="count(/DischargeMonitoringReport/PermitteeAddress/AttentionTo[last()]/following-
sibling::*) > 0">
<br/>
</xsl:if>
</xsl:if>

<xsl:if test="string-length(/DischargeMonitoringReport/PermitteeAddress/Address) > 0">
<xsl:value-of select="/DischargeMonitoringReport/PermitteeAddress/Address"/>
<xsl:if
test="count(/DischargeMonitoringReport/PermitteeAddress/Address[last()]/following-
sibling::*) > 0">
<br/>
</xsl:if>
</xsl:if>

<xsl:if test="string-
length(/DischargeMonitoringReport/PermitteeAddress/SupplementalAddress) > 0">
<xsl:value-of
select="/DischargeMonitoringReport/PermitteeAddress/SupplementalAddress"/>
<xsl:if
test="count(/DischargeMonitoringReport/PermitteeAddress/SupplementalAddress[last()]/fol
lowing-sibling::*) > 0">
<br/>
```

```
</xsl:if>

</xsl:if>


<xsl:choose>

<xsl:when test="string-length(/DischargeMonitoringReport/PermitteeAddress/City) > 0 and
string-length(/DischargeMonitoringReport/PermitteeAddress/State) > 0">

<xsl:value-of select="/DischargeMonitoringReport/PermitteeAddress/City"/>, <xsl:value-
of select="/DischargeMonitoringReport/PermitteeAddress/State"/>

<xsl:if
test="count(/DischargeMonitoringReport/PermitteeAddress/State[last()]/following-
sibling::*) > 0">

<xsl:text> </xsl:text>

</xsl:if>

</xsl:when>

<xsl:when test="string-length(/DischargeMonitoringReport/PermitteeAddress/City) > 0">

<xsl:value-of select="/DischargeMonitoringReport/PermitteeAddress/City"/>

<xsl:if test="count(/DischargeMonitoringReport/PermitteeAddress/City[last()]/following-
sibling::*) > 0">

<xsl:text> </xsl:text>

</xsl:if>

</xsl:when>

<xsl:when test="string-length(/DischargeMonitoringReport/PermitteeAddress/State) > 0">

<xsl:value-of select="/DischargeMonitoringReport/PermitteeAddress/State"/>

<xsl:if
test="count(/DischargeMonitoringReport/PermitteeAddress/State[last()]/following-
sibling::*) > 0">

<xsl:text> </xsl:text>

</xsl:if>

</xsl:when>

</xsl:choose>


<xsl:if test="string-length(/DischargeMonitoringReport/PermitteeAddress/ZipCode) > 0">

<xsl:value-of select="/DischargeMonitoringReport/PermitteeAddress/ZipCode"/>

</xsl:if>


<!-- Fill a non-breaking space if no address is available -->

<xsl:if test="string-length(/DischargeMonitoringReport/PermitteeAddress/AttentionTo) +
string-length(/DischargeMonitoringReport/PermitteeAddress/Address) + string-
length(/DischargeMonitoringReport/PermitteeAddress/SupplementalAddress) + string-
length(/DischargeMonitoringReport/PermitteeAddress/City) + string-
length(/DischargeMonitoringReport/PermitteeAddress/State) + string-
length(/DischargeMonitoringReport/PermitteeAddress/ZipCode) = 0">

 

</xsl:if>

</td>

<td class="header_text" nowrap="nowrap">Facility Location:</td>
```

```
<td>

<xsl:if test="string-length(/DischargeMonitoringReport/FacilityAddress/AttentionTo) >
0">

Attn: <xsl:value-of select="/DischargeMonitoringReport/FacilityAddress/AttentionTo"/>

<xsl:if
test="count(/DischargeMonitoringReport/FacilityAddress/AttentionTo[last()]/following-
sibling::*) > 0">

<br/>

</xsl:if>

</xsl:if>


<xsl:if test="string-length(/DischargeMonitoringReport/FacilityAddress/Address) > 0">

<xsl:value-of select="/DischargeMonitoringReport/FacilityAddress/Address"/>

<xsl:if
test="count(/DischargeMonitoringReport/FacilityAddress/Address[last()]/following-
sibling::*) > 0">

<br/>

</xsl:if>

</xsl:if>


<xsl:if test="string-
length(/DischargeMonitoringReport/FacilityAddress/SupplementalAddress) > 0">

<xsl:value-of select="/DischargeMonitoringReport/FacilityAddress/SupplementalAddress"/>

<xsl:if
test="count(/DischargeMonitoringReport/FacilityAddress/SupplementalAddress[last()]/foll
owing-sibling::*) > 0">

<br/>

</xsl:if>

</xsl:if>


<xsl:choose>

<xsl:when test="string-length(/DischargeMonitoringReport/FacilityAddress/City) > 0 and
string-length(/DischargeMonitoringReport/FacilityAddress/State) > 0">

<xsl:value-of select="/DischargeMonitoringReport/FacilityAddress/City"/>, <xsl:value-of
select="/DischargeMonitoringReport/FacilityAddress/State"/>

<xsl:if test="count(/DischargeMonitoringReport/FacilityAddress/State[last()]/following-
sibling::*) > 0">

<xsl:text> </xsl:text>

</xsl:if>

</xsl:when>

<xsl:when test="string-length(/DischargeMonitoringReport/FacilityAddress/City) > 0">

<xsl:value-of select="/DischargeMonitoringReport/FacilityAddress/City"/>

<xsl:if test="count(/DischargeMonitoringReport/FacilityAddress/City[last()]/following-
sibling::*) > 0">

<xsl:text> </xsl:text>

</xsl:if>
```

```
</xsl:when>

<xsl:when test="string-length(/DischargeMonitoringReport/FacilityAddress/State) > 0">

<xsl:value-of select="/DischargeMonitoringReport/FacilityAddress/State"/>

<xsl:if test="count(/DischargeMonitoringReport/FacilityAddress/State[last()]/following-
sibling::*) > 0">

<xsl:text> </xsl:text>

</xsl:if>

</xsl:when>

</xsl:choose>


<xsl:if test="string-length(/DischargeMonitoringReport/FacilityAddress/ZipCode) > 0">

<xsl:value-of select="/DischargeMonitoringReport/FacilityAddress/ZipCode"/>

</xsl:if>


<!-- Fill a non-breaking space if no address is available -->

<xsl:if test="string-length(/DischargeMonitoringReport/FacilityAddress/AttentionTo) +
string-length(/DischargeMonitoringReport/FacilityAddress/Address) + string-
length(/DischargeMonitoringReport/FacilityAddress/SupplementalAddress) + string-
length(/DischargeMonitoringReport/FacilityAddress/City) + string-
length(/DischargeMonitoringReport/FacilityAddress/State) + string-
length(/DischargeMonitoringReport/FacilityAddress/ZipCode) = 0">

 

</xsl:if>

</td>

</tr>

<tr>

<td colspan="6" style="border: 0px none; height: 5px;"></td>

</tr>

<tr>

<td class="header_text" nowrap="nowrap">Permitted Feature:</td>

<td class="border_right">

<xsl:value-of select="/DischargeMonitoringReport/PermittedFeatureIdentifier"/>


<xsl:if test="string-length(/DischargeMonitoringReport/PermittedFeatureTypeDescription)
> 0">

<br/>

<xsl:value-of select="/DischargeMonitoringReport/PermittedFeatureTypeDescription"/>

</xsl:if>

</td>

<td class="header_text">Discharge:</td>

<td>

<strong>

<xsl:value-of select="/DischargeMonitoringReport/PermittedFeatureIdentifier"/>-
<xsl:value-of select="/DischargeMonitoringReport/LimitSetDesignator"/>
```

```
</strong>

<xsl:if test="string-length(/DischargeMonitoringReport/LimitSetDesignatorName) > 0">

<br/>

<xsl:value-of select="/DischargeMonitoringReport/LimitSetDesignatorName"/>

</xsl:if>

</td>

<td> </td>

<td> </td>

</tr>

<tr>

<td colspan="6" class="section_row" bgcolor="#e0e0e0">

<span class="header_text">

<em>Report Dates &amp; Status</em>

</span>

</td>

</tr>

<tr>

<td class="header_text" nowrap="nowrap">Monitoring Period:</td>

<td class="border_right" nowrap="nowrap">

<strong>

<!-- Note: As xsd:dates, monitoring period dates will be in this form:

'-'? yyyy '-' mm '-' dd zzzzzz?

The first hyphen is an optional negative sign, yyyy is a four+ digit year,

mm is a two digit month, dd is a two digit day, and zzzzzz is an optional time

zone.  The logic below translates monitoring period dates into this form:

mm/dd/-?yy

The hyphen is an optional negative sign, and yy is the final two digits of

the year. -->

<!-- Output start date if present -->

<xsl:if test="string-length(/DischargeMonitoringReport/MonitoringPeriodStartDate) > 0">

<xsl:choose>

<xsl:when test="starts-with(/DischargeMonitoringReport/MonitoringPeriodStartDate, '-
')">

<xsl:variable name="mp_start_yr_str" select="substring-before(substring-
after(/DischargeMonitoringReport/MonitoringPeriodStartDate, '-'), '-')"/>

<xsl:variable name="mp_start_mon_str" select="substring(substring-after(substring-
after(/DischargeMonitoringReport/MonitoringPeriodStartDate, '-'), '-'), 1, 2)"/>

<xsl:variable name="mp_start_day_str" select="substring(substring-after(substring-
after(/DischargeMonitoringReport/MonitoringPeriodStartDate, '-'), '-'), 4, 2)"/>

From <xsl:value-of select="$mp_start_mon_str"/>/<xsl:value-of
select="$mp_start_day_str"/>/-<xsl:value-of select="substring($mp_start_yr_str, string-
```

```
length($mp_start_yr_str) - 1)"/>

</xsl:when>

<xsl:otherwise>

<xsl:variable name="mp_start_yr_str" select="substring-
before(/DischargeMonitoringReport/MonitoringPeriodStartDate, '-')"/>

<xsl:variable name="mp_start_mon_str" select="substring(substring-
after(/DischargeMonitoringReport/MonitoringPeriodStartDate, '-'), 1, 2)"/>

<xsl:variable name="mp_start_day_str" select="substring(substring-
after(/DischargeMonitoringReport/MonitoringPeriodStartDate, '-'), 4, 2)"/>

From <xsl:value-of select="$mp_start_mon_str"/>/<xsl:value-of
select="$mp_start_day_str"/>/<xsl:value-of select="substring($mp_start_yr_str, string-
length($mp_start_yr_str) - 1)"/>

</xsl:otherwise>

</xsl:choose>

</xsl:if>


<!-- Output "To " or " to " -->

<xsl:choose>

<xsl:when test="string-length(/DischargeMonitoringReport/MonitoringPeriodStartDate) >
0">

<xsl:text> to </xsl:text>

</xsl:when>

<xsl:otherwise>

<xsl:text>To </xsl:text>

</xsl:otherwise>

</xsl:choose>


<!-- Output end date -->

<xsl:choose>

<xsl:when test="starts-with(/DischargeMonitoringReport/MonitoringPeriodEndDate, '-')">

<xsl:variable name="mp_end_yr_str" select="substring-before(substring-
after(/DischargeMonitoringReport/MonitoringPeriodEndDate, '-'), '-')"/>

<xsl:variable name="mp_end_mon_str" select="substring(substring-after(substring-
after(/DischargeMonitoringReport/MonitoringPeriodEndDate, '-'), '-'), 1, 2)"/>

<xsl:variable name="mp_end_day_str" select="substring(substring-after(substring-
after(/DischargeMonitoringReport/MonitoringPeriodEndDate, '-'), '-'), 4, 2)"/>

<xsl:value-of select="$mp_end_mon_str"/>/<xsl:value-of select="$mp_end_day_str"/>/-
<xsl:value-of select="substring($mp_end_yr_str, string-length($mp_end_yr_str) - 1)"/>

</xsl:when>

<xsl:otherwise>

<xsl:variable name="mp_end_yr_str" select="substring-
before(/DischargeMonitoringReport/MonitoringPeriodEndDate, '-')"/>

<xsl:variable name="mp_end_mon_str" select="substring(substring-
after(/DischargeMonitoringReport/MonitoringPeriodEndDate, '-'), 1, 2)"/>

<xsl:variable name="mp_end_day_str" select="substring(substring-
after(/DischargeMonitoringReport/MonitoringPeriodEndDate, '-'), 4, 2)"/>
```

```
<xsl:value-of select="$mp_end_mon_str"/>/<xsl:value-of
select="$mp_end_day_str"/>/<xsl:value-of select="substring($mp_end_yr_str, string-
length($mp_end_yr_str) - 1)"/>

</xsl:otherwise>

</xsl:choose>

</strong>

</td>

<td class="header_text" nowrap="nowrap">DMR Due Date:</td>

<td class="border_right">

<strong>

<!-- See the note on monitoring period date formats above -->

<!-- Output report due date -->

<xsl:if test="string-length(/DischargeMonitoringReport/ReportDueDate) > 0">

<xsl:choose>

<xsl:when test="starts-with(/DischargeMonitoringReport/ReportDueDate, '-')">

<xsl:variable name="due_yr_str" select="substring-before(substring-
after(/DischargeMonitoringReport/ReportDueDate, '-'), '-')"/>

<xsl:variable name="due_mon_str" select="substring(substring-after(substring-
after(/DischargeMonitoringReport/ReportDueDate, '-'), '-'), 1, 2)"/>

<xsl:variable name="due_day_str" select="substring(substring-after(substring-
after(/DischargeMonitoringReport/ReportDueDate, '-'), '-'), 4, 2)"/>

<xsl:value-of select="$due_mon_str"/>/<xsl:value-of select="$due_day_str"/>/-
<xsl:value-of select="substring($due_yr_str, string-length($due_yr_str) - 1)"/>

</xsl:when>

<xsl:otherwise>

<xsl:variable name="due_yr_str" select="substring-
before(/DischargeMonitoringReport/ReportDueDate, '-')"/>

<xsl:variable name="due_mon_str" select="substring(substring-
after(/DischargeMonitoringReport/ReportDueDate, '-'), 1, 2)"/>

<xsl:variable name="due_day_str" select="substring(substring-
after(/DischargeMonitoringReport/ReportDueDate, '-'), 4, 2)"/>

<xsl:value-of select="$due_mon_str"/>/<xsl:value-of select="$due_day_str"/>/<xsl:value-
of select="substring($due_yr_str, string-length($due_yr_str) - 1)"/>

</xsl:otherwise>

</xsl:choose>

</xsl:if>

</strong>

</td>

<td class="header_text">Status:</td>

<td>

<strong>

<xsl:value-of select="/DischargeMonitoringReport/NetDMRStatusCode"/>

</strong>
```

```
</td>
</tr>
<tr>
<td colspan="6" class="section_row">
<span class="header_text">
<em>Considerations for Form Completion</em>
</span>
</td>
</tr>
<tr>
<td colspan="6">
<xsl:choose>
<xsl:when test="string-length(/DischargeMonitoringReport/LimitSetCommentText) > 0">
<xsl:value-of select="/DischargeMonitoringReport/LimitSetCommentText"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
</tr>
<tr>
<td colspan="6" class="section_row">
<span class="header_text">
<em>Principal Executive Officer</em>
</span>
</td>
</tr>
<tr>
<td class="header_text" nowrap="nowrap">First Name:</td>
<td class="border_right">
<xsl:choose>
<xsl:when test="string-
length(/DischargeMonitoringReport/ReportedPrincipalExecutiveOfficerFirstName) > 0">
<xsl:value-of
select="/DischargeMonitoringReport/ReportedPrincipalExecutiveOfficerFirstName"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
```

```
</xsl:choose>

</td>

<td class="header_text">Title:</td>

<td class="border_right">

<xsl:choose>

<xsl:when test="string-
length(/DischargeMonitoringReport/ReportedPrincipalExecutiveOfficerTitle) > 0">

<xsl:value-of
select="/DischargeMonitoringReport/ReportedPrincipalExecutiveOfficerTitle"/>

</xsl:when>

<xsl:otherwise>

 

</xsl:otherwise>

</xsl:choose>

</td>

<td class="header_text">Telephone:</td>

<td>

<xsl:choose>

<xsl:when test="string-
length(/DischargeMonitoringReport/ReportedPrincipalExecutiveOfficerTelephone) > 0">

<xsl:value-of
select="/DischargeMonitoringReport/ReportedPrincipalExecutiveOfficerTelephone"/>

</xsl:when>

<xsl:otherwise>

 

</xsl:otherwise>

</xsl:choose>

</td>

</tr>

<tr>

<td class="header_text" nowrap="nowrap">Last Name:</td>

<td class="border_right">

<xsl:choose>

<xsl:when test="string-
length(/DischargeMonitoringReport/ReportedPrincipalExecutiveOfficerLastName) > 0">

<xsl:value-of
select="/DischargeMonitoringReport/ReportedPrincipalExecutiveOfficerLastName"/>

</xsl:when>

<xsl:otherwise>

 

</xsl:otherwise>

</xsl:choose>
```

```
</td>
</tr>
<tr>
<td colspan="6" class="section_row">
<span class="header_text">
<em>No Data Indicator (NODI)</em>
</span>
</td>
</tr>
<tr>
<td class="header_text">Form NODI:</td>
<td colspan="5">--</td>
</tr>
</tbody>
</table>
</div>
<!-- End <div id="dmr_header_expanded"> -->
</div>
<!-- End <div id="dmr_header"> -->
<div id="dmr_parameters">
<table class="datatable" border="1" cellpadding="0" cellspacing="0" width="100%">
<thead>
<tr>
<th scope="col" class="centeredtext" colspan="2">Parameter</th>
<th scope="col" class="centeredtext" rowspan="2">Monitoring Location</th>
<th scope="col" class="centeredtext" rowspan="2">Season #</th>
<th scope="col" class="centeredtext" rowspan="2">Param. NODI</th>
<th scope="col" class="centeredtext diagonal_lines" rowspan="2"> </th>
<th scope="col" class="centeredtext" colspan="5">Quantity or Loading</th>
<th scope="col" class="centeredtext" colspan="7">Quality or Concentration</th>
<th scope="col" class="centeredtext" rowspan="2">
<span title="# of Excursions"># of Ex.</span>
</th>
<th scope="col" class="centeredtext" rowspan="2">Frequency of Analysis</th>
<th scope="col" class="centeredtext" rowspan="2">Sample Type</th>
</tr>
<tr>
<th scope="col" class="centeredtext" nowrap="nowrap">Code</th>
<th class="centeredtext" scope="col">Name</th>
```

```
<th scope="col" class="centeredtext">Qualifier 1</th>
<th scope="col" class="centeredtext">Value 1</th>
<th scope="col" class="centeredtext">Qualifier 2</th>
<th scope="col" class="centeredtext">Value 2</th>
<th scope="col" class="centeredtext">Units</th>
<th scope="col" class="centeredtext">Qualifier 1</th>
<th scope="col" class="centeredtext">Value 1</th>
<th scope="col" class="centeredtext">Qualifier 2</th>
<th scope="col" class="centeredtext">Value 2</th>
<th scope="col" class="centeredtext">Qualifier 3</th>
<th scope="col" class="centeredtext">Value 3</th>
<th scope="col" class="centeredtext">Units</th>
</tr>
</thead>
<tbody>
<xsl:for-each select="/DischargeMonitoringReport/Parameters/Parameter">
<!-- Apply Sorting -->
<xsl:sort select="ParameterCode"/>
<xsl:sort select="MonitoringLocationCode" />
<xsl:sort select="MonitoringLocationDescription" />
<xsl:sort select="PermitSeasonNumber" />

<!-- Determine # of rows needed for parameter (3 or 4) -->
<xsl:variable name="param_rows_num" select="3 +
number(PermitEffluentTradingPartnerFlag='Y')"/>

<!-- Set a variable used in error row coloring -->
<xsl:variable name="param_code_str" select="ParameterCode"/>
<xsl:variable name="mon_loc_str" select="MonitoringLocationCode"/>
<xsl:variable name="season_number_str" select="PermitSeasonNumber"/>

<!-- BEGIN: First parameter row (Sample). -->
<tr>
<!-- Add alternate row coloring -->
<xsl:if test="position() mod 2 = 0">
<xsl:attribute name="style">background-color: rgb(241, 248, 248);</xsl:attribute>
</xsl:if>

<!-- Add error row coloring -->
<xsl:if
test="count(/DischargeMonitoringReport/AcknowledgedSoftErrors/SoftError[ParameterCode=$
param_code_str and MonitoringLocationCode=$mon_loc_str and
PermitSeasonNumber=$season_number_str and MonitoringLocationCode=$mon_loc_str and
PermitSeasonNumber=$season_number_str]) > 0">
```

```
<xsl:attribute name="style">background-color:#FCEFF0;</xsl:attribute>

</xsl:if>


<td rowspan="{$param_rows_num}" class="valignmiddle" scope="row">

<xsl:choose>

<xsl:when
test="count(/DischargeMonitoringReport/AcknowledgedSoftErrors/SoftError[ParameterCode=$
param_code_str and MonitoringLocationCode=$mon_loc_str and
PermitSeasonNumber=$season_number_str and MonitoringLocationCode=$mon_loc_str and
PermitSeasonNumber=$season_number_str]) > 0">

<span class="style1">X</span> <xsl:value-of select="ParameterCode"/>

</xsl:when>

<xsl:otherwise>

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

</xsl:otherwise>

</xsl:choose>

</td>

<td rowspan="{$param_rows_num}" class="valignmiddle" scope="row">

<xsl:choose>

<xsl:when test="string-length(ParameterName) > 0">

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

</xsl:when>

<xsl:otherwise>

 

</xsl:otherwise>

</xsl:choose>

</td>

<td rowspan="{$param_rows_num}" class="valignmiddle" scope="row">

<xsl:choose>

<xsl:when test="string-length(MonitoringLocationCode) > 0 and string-
length(MonitoringLocationDescription) > 0">

<xsl:value-of select="MonitoringLocationCode"/> - <xsl:value-of
select="MonitoringLocationDescription"/>

</xsl:when>

<xsl:when test="string-length(MonitoringLocationCode) > 0">

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

</xsl:when>

<xsl:when test="string-length(MonitoringLocationDescription) > 0">

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

</xsl:when>

<xsl:otherwise>

 
```

```
</xsl:otherwise>
</xsl:choose>
</td>
<td rowspan="{$param_rows_num}" class="valignmiddle" scope="row">
<xsl:choose>
<xsl:when test="string-length(PermitSeasonNumber) > 0">
<xsl:value-of select="PermitSeasonNumber"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td rowspan="{$param_rows_num}" class="valignmiddle" scope="row">--</td>
<th class="centeredtext valignmiddle" scope="row">Sample</th>
<td class="valignmiddle" rowspan="{$param_rows_num - 2}">
<xsl:choose>
<xsl:when test="string-length(Quantity1/ReportedQualifier) > 0">
<xsl:value-of select="Quantity1/ReportedQualifier"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle">
<xsl:choose>
<xsl:when test="string-length(Quantity1/ReportedValue) > 0">
<xsl:value-of select="Quantity1/ReportedValue"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle" rowspan="{$param_rows_num - 2}">
<xsl:choose>
<xsl:when test="string-length(Quantity2/ReportedQualifier) > 0">
<xsl:value-of select="Quantity2/ReportedQualifier"/>
```

```
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle">
<xsl:choose>
<xsl:when test="string-length(Quantity2/ReportedValue) > 0">
<xsl:value-of select="Quantity2/ReportedValue"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle">
<xsl:choose>
<xsl:when test="string-length(ReportedQuantityUnitCode) > 0 and string-
length(ReportedQuantityUnitDescription) > 0">
<xsl:value-of select="ReportedQuantityUnitCode"/> - <xsl:value-of
select="ReportedQuantityUnitDescription"/>
</xsl:when>
<xsl:when test="string-length(ReportedQuantityUnitCode) > 0">
<xsl:value-of select="ReportedQuantityUnitCode"/>
</xsl:when>
<xsl:when test="string-length(ReportedQuantityUnitDescription) > 0">
<xsl:value-of select="ReportedQuantityUnitDescription"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle" rowspan="{$param_rows_num - 2}">
<xsl:choose>
<xsl:when test="string-length(Concentration1/ReportedQualifier) > 0">
<xsl:value-of select="Concentration1/ReportedQualifier"/>
</xsl:when>
<xsl:otherwise>
```

```
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle">
<xsl:choose>
<xsl:when test="string-length(Concentration1/ReportedValue) > 0">
<xsl:value-of select="Concentration1/ReportedValue"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle" rowspan="{$param_rows_num - 2}">
<xsl:choose>
<xsl:when test="string-length(Concentration2/ReportedQualifier) > 0">
<xsl:value-of select="Concentration2/ReportedQualifier"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle">
<xsl:choose>
<xsl:when test="string-length(Concentration2/ReportedValue) > 0">
<xsl:value-of select="Concentration2/ReportedValue"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle" rowspan="{$param_rows_num - 2}">
<xsl:choose>
<xsl:when test="string-length(Concentration3/ReportedQualifier) > 0">
<xsl:value-of select="Concentration3/ReportedQualifier"/>
</xsl:when>
```

```
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle">
<xsl:choose>
<xsl:when test="string-length(Concentration3/ReportedValue) > 0">
<xsl:value-of select="Concentration3/ReportedValue"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle">
<xsl:choose>
<xsl:when test="string-length(ReportedConcentrationUnitCode) > 0 and string-
length(ReportedConcentrationUnitDescription) > 0">
<xsl:value-of select="ReportedConcentrationUnitCode"/> - <xsl:value-of
select="ReportedConcentrationUnitDescription"/>
</xsl:when>
<xsl:when test="string-length(ReportedConcentrationUnitCode) > 0">
<xsl:value-of select="ReportedConcentrationUnitCode"/>
</xsl:when>
<xsl:when test="string-length(ReportedConcentrationUnitDescription) > 0">
<xsl:value-of select="ReportedConcentrationUnitDescription"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td rowspan="{$param_rows_num}" class="valignmiddle" scope="row">
<xsl:choose>
<xsl:when test="string-length(ReportedNumberOfExcursions) > 0">
<xsl:value-of select="ReportedNumberOfExcursions"/>
</xsl:when>
<xsl:otherwise>
 
```

```
</xsl:otherwise>

</xsl:choose>

</td>

<td class="valignmiddle">

<xsl:choose>

<xsl:when test="string-length(ReportedFrequencyOfAnalysisCode) > 0 and string-
length(ReportedFrequencyOfAnalysisDescription) > 0">

<xsl:value-of select="ReportedFrequencyOfAnalysisCode"/> - <xsl:value-of
select="ReportedFrequencyOfAnalysisDescription"/>

</xsl:when>

<xsl:when test="string-length(ReportedFrequencyOfAnalysisCode) > 0">

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

</xsl:when>

<xsl:when test="string-length(ReportedFrequencyOfAnalysisDescription) > 0">

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

</xsl:when>

<xsl:otherwise>

 

</xsl:otherwise>

</xsl:choose>

</td>

<td class="valignmiddle">

<xsl:choose>

<xsl:when test="string-length(ReportedSampleTypeCode) > 0 and string-
length(ReportedSampleTypeDescription) > 0">

<xsl:value-of select="ReportedSampleTypeCode"/> - <xsl:value-of
select="ReportedSampleTypeDescription"/>

</xsl:when>

<xsl:when test="string-length(ReportedSampleTypeCode) > 0">

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

</xsl:when>

<xsl:when test="string-length(ReportedSampleTypeDescription) > 0">

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

</xsl:when>

<xsl:otherwise>

 

</xsl:otherwise>

</xsl:choose>

</td>

</tr>

<!-- END: First parameter row (Sample). -->
```

```
<!-- BEGIN: Second parameter row (Effluent Trading Sample), if needed. -->
<xsl:if test="$param_rows_num = 4">
<tr>
<!-- Add alternate row coloring -->
<xsl:if test="position() mod 2 = 0">
<xsl:attribute name="style">background-color: rgb(241, 248, 248);</xsl:attribute>
</xsl:if>

<!-- Add error row coloring -->
<xsl:if
test="count(/DischargeMonitoringReport/AcknowledgedSoftErrors/SoftError[ParameterCode=$
param_code_str and MonitoringLocationCode=$mon_loc_str and
PermitSeasonNumber=$season_number_str and MonitoringLocationCode=$mon_loc_str and
PermitSeasonNumber=$season_number_str]) > 0">
<xsl:attribute name="style">background-color:#FCEFF0;</xsl:attribute>
</xsl:if>

<th class="centeredtext valignmiddle" scope="row">Effluent Trading Sample</th>
<td class="valignmiddle">
<xsl:choose>
<xsl:when test="string-length(Quantity1/ReportedEffluentValue) > 0">
<xsl:value-of select="Quantity1/ReportedEffluentValue"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle">
<xsl:choose>
<xsl:when test="string-length(Quantity2/ReportedEffluentValue) > 0">
<xsl:value-of select="Quantity2/ReportedEffluentValue"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle"> </td>
<td class="valignmiddle">
<xsl:choose>
<xsl:when test="string-length(Concentration1/ReportedEffluentValue) > 0">
```

```
<xsl:value-of select="Concentration1/ReportedEffluentValue"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle">
<xsl:choose>
<xsl:when test="string-length(Concentration2/ReportedEffluentValue) > 0">
<xsl:value-of select="Concentration2/ReportedEffluentValue"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle">
<xsl:choose>
<xsl:when test="string-length(Concentration3/ReportedEffluentValue) > 0">
<xsl:value-of select="Concentration3/ReportedEffluentValue"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle"> </td>
<td class="valignmiddle"> </td>
<td class="valignmiddle"> </td>
</tr>
</xsl:if>
<!-- END: Second parameter row (Effluent Trading Sample), if needed. -->
<!-- BEGIN: Third parameter row (Permit Req.). -->
<tr>
<!-- Add alternate row coloring -->
<xsl:if test="position() mod 2 = 0">
<xsl:attribute name="style">background-color: rgb(241, 248, 248);</xsl:attribute>
</xsl:if>
```

```
<!-- Add error row coloring -->

<xsl:if
test="count(/DischargeMonitoringReport/AcknowledgedSoftErrors/SoftError[ParameterCode=$
param_code_str and MonitoringLocationCode=$mon_loc_str and
PermitSeasonNumber=$season_number_str and MonitoringLocationCode=$mon_loc_str and
PermitSeasonNumber=$season_number_str]) > 0">

<xsl:attribute name="style">background-color:#FCEFF0;</xsl:attribute>

</xsl:if>


<th class="centeredtext valignmiddle" scope="row">Permit Req.</th>

<td class="valignmiddle">

<xsl:choose>

<xsl:when test="string-length(Quantity1/PermitLimitQualifier) > 0">

<xsl:value-of select="Quantity1/PermitLimitQualifier"/>

</xsl:when>

<xsl:otherwise>

 

</xsl:otherwise>

</xsl:choose>

</td>

<td class="valignmiddle">

<xsl:choose>

<xsl:when test="string-length(Quantity1/PermitLimitValue) > 0 and string-
length(Quantity1/PermitStatisticalBaseDescription) > 0">

<xsl:value-of select="Quantity1/PermitLimitValue"/><xsl:text> </xsl:text><xsl:value-of
select="Quantity1/PermitStatisticalBaseDescription"/>

</xsl:when>

<xsl:when test="string-length(Quantity1/PermitLimitValue) > 0">

<xsl:value-of select="Quantity1/PermitLimitValue"/>

</xsl:when>

<xsl:when test="string-length(Quantity1/PermitStatisticalBaseDescription) > 0">

<xsl:value-of select="Quantity1/PermitStatisticalBaseDescription"/>

</xsl:when>

<xsl:otherwise>

 

</xsl:otherwise>

</xsl:choose>

</td>

<td class="valignmiddle">

<xsl:choose>

<xsl:when test="string-length(Quantity2/PermitLimitQualifier) > 0">

<xsl:value-of select="Quantity2/PermitLimitQualifier"/>
```

```
</xsl:when>

<xsl:otherwise>

 

</xsl:otherwise>

</xsl:choose>

</td>

<td class="valignmiddle">

<xsl:choose>

<xsl:when test="string-length(Quantity2/PermitLimitValue) > 0 and string-
length(Quantity2/PermitStatisticalBaseDescription) > 0">

<xsl:value-of select="Quantity2/PermitLimitValue"/><xsl:text> </xsl:text><xsl:value-of
select="Quantity2/PermitStatisticalBaseDescription"/>

</xsl:when>

<xsl:when test="string-length(Quantity2/PermitLimitValue) > 0">

<xsl:value-of select="Quantity2/PermitLimitValue"/>

</xsl:when>

<xsl:when test="string-length(Quantity2/PermitStatisticalBaseDescription) > 0">

<xsl:value-of select="Quantity2/PermitStatisticalBaseDescription"/>

</xsl:when>

<xsl:otherwise>

 

</xsl:otherwise>

</xsl:choose>

</td>

<td class="valignmiddle">

<xsl:choose>

<xsl:when test="string-length(PermitQuantityUnitCode) > 0 and string-
length(PermitQuantityUnitDescription) > 0">

<xsl:value-of select="PermitQuantityUnitCode"/> - <xsl:value-of
select="PermitQuantityUnitDescription"/>

</xsl:when>

<xsl:when test="string-length(PermitQuantityUnitCode) > 0">

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

</xsl:when>

<xsl:when test="string-length(PermitQuantityUnitDescription) > 0">

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

</xsl:when>

<xsl:otherwise>

 

</xsl:otherwise>

</xsl:choose>
```

```
</td>
<td class="valignmiddle">
<xsl:choose>
<xsl:when test="string-length(Concentration1/PermitLimitQualifier) > 0">
<xsl:value-of select="Concentration1/PermitLimitQualifier"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle">
<xsl:choose>
<xsl:when test="string-length(Concentration1/PermitLimitValue) > 0 and string-
length(Concentration1/PermitStatisticalBaseDescription) > 0">
<xsl:value-of select="Concentration1/PermitLimitValue"/><xsl:text>
</xsl:text><xsl:value-of select="Concentration1/PermitStatisticalBaseDescription"/>
</xsl:when>
<xsl:when test="string-length(Concentration1/PermitLimitValue) > 0">
<xsl:value-of select="Concentration1/PermitLimitValue"/>
</xsl:when>
<xsl:when test="string-length(Concentration1/PermitStatisticalBaseDescription) > 0">
<xsl:value-of select="Concentration1/PermitStatisticalBaseDescription"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle">
<xsl:choose>
<xsl:when test="string-length(Concentration2/PermitLimitQualifier) > 0">
<xsl:value-of select="Concentration2/PermitLimitQualifier"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle">
```

```
<xsl:choose>
<xsl:when test="string-length(Concentration2/PermitLimitValue) > 0 and string-length(Concentration2/PermitStatisticalBaseDescription) > 0">
<xsl:value-of select="Concentration2/PermitLimitValue"/><xsl:text></xsl:text><xsl:value-of select="Concentration2/PermitStatisticalBaseDescription"/>
</xsl:when>
<xsl:when test="string-length(Concentration2/PermitLimitValue) > 0">
<xsl:value-of select="Concentration2/PermitLimitValue"/>
</xsl:when>
<xsl:when test="string-length(Concentration2/PermitStatisticalBaseDescription) > 0">
<xsl:value-of select="Concentration2/PermitStatisticalBaseDescription"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle">
<xsl:choose>
<xsl:when test="string-length(Concentration3/PermitLimitQualifier) > 0">
<xsl:value-of select="Concentration3/PermitLimitQualifier"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle">
<xsl:choose>
<xsl:when test="string-length(Concentration3/PermitLimitValue) > 0 and string-length(Concentration3/PermitStatisticalBaseDescription) > 0">
<xsl:value-of select="Concentration3/PermitLimitValue"/><xsl:text></xsl:text><xsl:value-of select="Concentration3/PermitStatisticalBaseDescription"/>
</xsl:when>
<xsl:when test="string-length(Concentration3/PermitLimitValue) > 0">
<xsl:value-of select="Concentration3/PermitLimitValue"/>
</xsl:when>
<xsl:when test="string-length(Concentration3/PermitStatisticalBaseDescription) > 0">
<xsl:value-of select="Concentration3/PermitStatisticalBaseDescription"/>
</xsl:when>
<xsl:otherwise>
```

```
 

</xsl:otherwise>

</xsl:choose>

</td>

<td class="valignmiddle">

<xsl:choose>

<xsl:when test="string-length(PermitConcentrationUnitCode) > 0 and string-
length(PermitConcentrationUnitDescription) > 0">

<xsl:value-of select="PermitConcentrationUnitCode"/> - <xsl:value-of
select="PermitConcentrationUnitDescription"/>

</xsl:when>

<xsl:when test="string-length(PermitConcentrationUnitCode) > 0">

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

</xsl:when>

<xsl:when test="string-length(PermitConcentrationUnitDescription) > 0">

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

</xsl:when>

<xsl:otherwise>

 

</xsl:otherwise>

</xsl:choose>

</td>

<td class="valignmiddle">

<xsl:choose>

<xsl:when test="string-length(PermitFrequencyOfAnalysisCode) > 0 and string-
length(PermitFrequencyOfAnalysisDescription) > 0">

<xsl:value-of select="PermitFrequencyOfAnalysisCode"/> - <xsl:value-of
select="PermitFrequencyOfAnalysisDescription"/>

</xsl:when>

<xsl:when test="string-length(PermitFrequencyOfAnalysisCode) > 0">

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

</xsl:when>

<xsl:when test="string-length(PermitFrequencyOfAnalysisDescription) > 0">

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

</xsl:when>

<xsl:otherwise>

 

</xsl:otherwise>

</xsl:choose>

</td>

<td class="valignmiddle">
```

```
<xsl:choose>

<xsl:when test="string-length(PermitSampleTypeCode) > 0 and string-
length(PermitSampleTypeDescription) > 0">

<xsl:value-of select="PermitSampleTypeCode"/> - <xsl:value-of
select="PermitSampleTypeDescription"/>

</xsl:when>

<xsl:when test="string-length(PermitSampleTypeCode) > 0">

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

</xsl:when>

<xsl:when test="string-length(PermitSampleTypeDescription) > 0">

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

</xsl:when>

<xsl:otherwise>

 

</xsl:otherwise>

</xsl:choose>

</td>

</tr>

<!-- END: Third parameter row (Permit Req.). -->

<!-- BEGIN: Fourth parameter row (Value NODI). -->

<tr>

<!-- Add alternate row coloring -->

<xsl:if test="position() mod 2 = 0">

<xsl:attribute name="style">background-color: rgb(241, 248, 248);</xsl:attribute>

</xsl:if>


<!-- Add error row coloring -->

<xsl:if
test="count(/DischargeMonitoringReport/AcknowledgedSoftErrors/SoftError[ParameterCode=$
param_code_str and MonitoringLocationCode=$mon_loc_str and
PermitSeasonNumber=$season_number_str and MonitoringLocationCode=$mon_loc_str and
PermitSeasonNumber=$season_number_str]) > 0">

<xsl:attribute name="style">background-color:#FCEFF0;</xsl:attribute>

</xsl:if>


<th class="centeredtext valignmiddle" scope="row">Value NODI</th>

<td class="valignmiddle"> </td>

<td class="valignmiddle">

<xsl:choose>

<xsl:when test="string-length(Quantity1/ReportedNoDischargeCode) > 0 and string-
length(Quantity1/ReportedNoDischargeDescription) > 0">

<xsl:value-of select="Quantity1/ReportedNoDischargeCode"/> - <xsl:value-of
select="Quantity1/ReportedNoDischargeDescription"/>

</xsl:when>
```

```
<xsl:when test="string-length(Quantity1/ReportedNoDischargeCode) > 0">
<xsl:value-of select="Quantity1/ReportedNoDischargeCode"/>
</xsl:when>
<xsl:when test="string-length(Quantity1/ReportedNoDischargeDescription) > 0">
<xsl:value-of select="Quantity1/ReportedNoDischargeDescription"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle"> </td>
<td class="valignmiddle">
<xsl:choose>
<xsl:when test="string-length(Quantity2/ReportedNoDischargeCode) > 0 and string-
length(Quantity2/ReportedNoDischargeDescription) > 0">
<xsl:value-of select="Quantity2/ReportedNoDischargeCode"/> - <xsl:value-of
select="Quantity2/ReportedNoDischargeDescription"/>
</xsl:when>
<xsl:when test="string-length(Quantity2/ReportedNoDischargeCode) > 0">
<xsl:value-of select="Quantity2/ReportedNoDischargeCode"/>
</xsl:when>
<xsl:when test="string-length(Quantity2/ReportedNoDischargeDescription) > 0">
<xsl:value-of select="Quantity2/ReportedNoDischargeDescription"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle"> </td>
<td class="valignmiddle"> </td>
<td class="valignmiddle">
<xsl:choose>
<xsl:when test="string-length(Concentration1/ReportedNoDischargeCode) > 0 and string-
length(Concentration1/ReportedNoDischargeDescription) > 0">
<xsl:value-of select="Concentration1/ReportedNoDischargeCode"/> - <xsl:value-of
select="Concentration1/ReportedNoDischargeDescription"/>
</xsl:when>
<xsl:when test="string-length(Concentration1/ReportedNoDischargeCode) > 0">
<xsl:value-of select="Concentration1/ReportedNoDischargeCode"/>
```

```
</xsl:when>
<xsl:when test="string-length(Concentration1/ReportedNoDischargeDescription) > 0">
<xsl:value-of select="Concentration1/ReportedNoDischargeDescription"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle"> </td>
<td class="valignmiddle">
<xsl:choose>
<xsl:when test="string-length(Concentration2/ReportedNoDischargeCode) > 0 and string-
length(Concentration2/ReportedNoDischargeDescription) > 0">
<xsl:value-of select="Concentration2/ReportedNoDischargeCode"/> - <xsl:value-of
select="Concentration2/ReportedNoDischargeDescription"/>
</xsl:when>
<xsl:when test="string-length(Concentration2/ReportedNoDischargeCode) > 0">
<xsl:value-of select="Concentration2/ReportedNoDischargeCode"/>
</xsl:when>
<xsl:when test="string-length(Concentration2/ReportedNoDischargeDescription) > 0">
<xsl:value-of select="Concentration2/ReportedNoDischargeDescription"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle"> </td>
<td class="valignmiddle">
<xsl:choose>
<xsl:when test="string-length(Concentration3/ReportedNoDischargeCode) > 0 and string-
length(Concentration3/ReportedNoDischargeDescription) > 0">
<xsl:value-of select="Concentration3/ReportedNoDischargeCode"/> - <xsl:value-of
select="Concentration3/ReportedNoDischargeDescription"/>
</xsl:when>
<xsl:when test="string-length(Concentration3/ReportedNoDischargeCode) > 0">
<xsl:value-of select="Concentration3/ReportedNoDischargeCode"/>
</xsl:when>
<xsl:when test="string-length(Concentration3/ReportedNoDischargeDescription) > 0">
<xsl:value-of select="Concentration3/ReportedNoDischargeDescription"/>
```

```
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td class="valignmiddle"> </td>
<td class="valignmiddle"> </td>
<td class="valignmiddle"> </td>
</tr>
<!-- END: Fourth parameter row (Value NODI). -->
</xsl:for-each>
</tbody>
</table>
</div>  <!-- End <div id="dmr_parameters"> -->
<div id="dmr_footer" style="background-color: rgb(209, 233, 233); width: 100%;">
<table class="dmr_header_footer_tbl">
<tbody>
<tr>
<td class="section_row" style="height: 14px;" colspan="4" bgcolor="#e0e0e0">
<span class="header_text">
<em>Submission Note</em>
</span>
</td>
</tr>
<tr>
<td colspan="4">
If a parameter row does not contain any values for the Sample nor Effluent Trading,
then none of the following fields will be
submitted for that row:  Units, Number of Excursions, Frequency of Analysis, and Sample
Type.
</td>
</tr>
<tr>
<td class="section_row" style="height: 14px;" colspan="4" bgcolor="#e0e0e0">
<span class="header_text">
<em>Edit Check Errors</em>
</span>
</td>
</tr>
```

```
<tr>
<td colspan="4">
<!-- Only display a table of errors if there's at least one -->
<xsl:choose>
<xsl:when test="count(/DischargeMonitoringReport/AcknowledgedSoftErrors/SoftError) >
0">
<table class="datatable" cellpadding="0" cellspacing="0" width="100%">
<thead>
<tr>
<th class="centeredtext_and_vert" colspan="2">Parameter</th>
<th class="centeredtext_and_vert" rowspan="2">Monitoring Location</th>
<th class="centeredtext_and_vert" rowspan="2">Field</th>
<th class="centeredtext_and_vert" rowspan="2">Type</th>
<th class="centeredtext_and_vert" rowspan="2">Description</th>
<th class="centeredtext_and_vert" rowspan="2">Acknowledge</th>
</tr>
<tr>
<th class="centeredtext_and_vert" nowrap="nowrap">Code</th>
<th class="centeredtext_and_vert">Name</th>
</tr>
</thead>
<tbody id="standard_hover">
<xsl:for-each select="/DischargeMonitoringReport/AcknowledgedSoftErrors/SoftError">
<tr>
<td>
<xsl:choose>
<xsl:when test="string-length(ParameterCode) > 0">
<xsl:value-of select="ParameterCode"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
<td>
<xsl:choose>
<xsl:when test="string-length(ParameterName) > 0">
<xsl:value-of select="ParameterName"/>
</xsl:when>
<xsl:otherwise>
```

```
 

</xsl:otherwise>

</xsl:choose>

</td>

<td>

<xsl:choose>

<xsl:when test="string-length(MonitoringLocationCode) > 0 and string-
length(MonitoringLocationDescription) > 0">

<xsl:value-of select="MonitoringLocationCode"/> - <xsl:value-of
select="MonitoringLocationDescription"/>

</xsl:when>

<xsl:when test="string-length(MonitoringLocationCode) > 0">

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

</xsl:when>

<xsl:when test="string-length(MonitoringLocationDescription) > 0">

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

</xsl:when>

<xsl:otherwise>

 

</xsl:otherwise>

</xsl:choose>

</td>

<td>

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

</td>

<td>Soft</td>

<td>

<xsl:value-of select="ErrorDescription"/> (Error Code: <xsl:value-of
select="ErrorCode"/>)

</td>

<td class="centeredtext_and_vert">

Yes

</td>

</tr>

</xsl:for-each>

</tbody>

</table>

</xsl:when>

<xsl:otherwise>

<em>No errors.</em>

</xsl:otherwise>
```

```
</xsl:choose>
</td>
</tr>
<tr>
<td class="section_row" colspan="4" bgcolor="#e0e0e0">
<span class="header_text">
<em>Comments</em>
</span>
</td>
</tr>
<tr>
<td colspan="4">
<xsl:choose>
<xsl:when test="string-length(/DischargeMonitoringReport/ReportedCommentText) > 0">
<xsl:value-of select="/DischargeMonitoringReport/ReportedCommentText"/>
</xsl:when>
<xsl:otherwise>
 
</xsl:otherwise>
</xsl:choose>
</td>
</tr>
<tr>
<td class="section_row" colspan="4">
<span class="header_text">
<em>Attachments</em>
</span>
</td>
</tr>
<tr>
<td class="valignmiddle" colspan="4">
<!-- Only display a table of attachments if there's at least one -->
<xsl:choose>
<xsl:when test="count(/DischargeMonitoringReport/Attachments/Attachment) > 0">
<table class="datatable" border="1" cellpadding="0" cellspacing="0" width="100%">
<thead>
<tr>
<th class="centeredtext_and_vert">Name</th>
<th class="centeredtext_and_vert">Type</th>
```

```
<th class="centeredtext_and_vert">Size</th>
</tr>
</thead>
<tbody id="standard_hover">
<xsl:for-each select="/DischargeMonitoringReport/Attachments/Attachment">
<tr>
<td>
<xsl:value-of select="Name"/>
</td>
<td>
<xsl:value-of select="FileType"/>
</td>
<td>
<xsl:value-of select="Size"/>
</td>
</tr>
</xsl:for-each>
</tbody>
</table>
</xsl:when>
<xsl:otherwise>
<em>No attachments.</em>
</xsl:otherwise>
</xsl:choose>
</td>
</tr>
<tr>
<td class="section_row" colspan="4" bgcolor="#e0e0e0">
<span class="header_text">
<em>Report Last Saved By</em>
</span>
</td>
</tr>
<tr>
<td class="section_row" colspan="4" bgcolor="#e0e0e0">
<span class="header_text">
<em> <xsl:value-of select="/DischargeMonitoringReport/PermitteeName"/></em>
</span>
</td>
```

```
</tr>

<tr>

<td>User:</td>

<td class="border_right" style="padding-right: 90px;"> <xsl:value-of
select="/DischargeMonitoringReport/DmrLastEditedByUsername"/></td>

<td style="padding-left: 90px;">Date/Time:</td>

<td>

<xsl:value-of select="substring(/DischargeMonitoringReport/DmrLastEditedDate,
1,10)"/> 

<xsl:value-of select="substring(/DischargeMonitoringReport/DmrLastEditedDate, 12,5)"/>
 

(Time Zone: <xsl:value-of
select="substring(/DischargeMonitoringReport/DmrLastEditedDate, 20,6)"/>)</td>

</tr>

<tr>

<td>Name:</td>

<td>

<xsl:value-of select="/DischargeMonitoringReport/DmrLastEditedByUserFirstName"/>
  

<xsl:value-of select="/DischargeMonitoringReport/DmrLastEditedByUserLastName"/>

</td>

<td colspan="2"> </td>

</tr>

<tr>

<td>E-Mail:</td>

<td> <xsl:value-of select="/DischargeMonitoringReport/DmrLastEditedByUserEmail"/></td>

<td colspan="2"> </td>

</tr>

</tbody>

</table>

</div>  <!-- End <div id="dmr_footer"> -->

</div>

<!-- End <div id="dmr_form"> -->

</div>

<!-- End <div id="sub_content_no_top_padding"> -->

</div>

<!-- End <div id="content_wide"> -->

</td>

</tr>

</tbody>

</table>
```

```
</div>
<!-- End <div id="programming_content"> -->
</td>
<td bgcolor="#0e6794" width="5"></td>
</tr>
</tbody>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```