

Setting Custom NULL Values in an MII Illuminator Document



Applies to:

SAP MII v11.5 or newer. For more information, visit the [Manufacturing homepage](#).

Summary

This document will describe how to create and use an XLST that will allow you to set the custom values for the default values for NULL data in an MII Illuminator Document XML. This is a common issue where customers need the ability to return a customized value in their Illuminator Document responses.

Author: Salvatore Castro

Company: SAP Labs, LLC

Created on: 17 February 17, 2010

Author Bio

Salvatore Castro of SAP Labs and has a Bachelors Degree in Computer Engineering and a Masters Degree in Computer Science both through the Rochester Institute of Technology. He is a member of the MII Product Management group under John Schaefer.

Table of Contents

Overview	3
Creating the XSLT	3
Customizing the XSLT for your Environment	3
Using an XSLT in a Query	3
Appendix A.....	5
Related Content.....	8
Copyright.....	9

Overview

The need commonly arises where a custom default value for a null field needs to be set in the return XML data from an MII Query. The following document describes how to build an XSL Transform to allow control over these values. For example for a null date field MII will substitute the word 'TimeUnavailable' which may not be compatible with other computer software. The transform allows for the control of the substitute value to be whatever is necessary, for example: 00/00/0000 00:00:00 could replace it in the response.

Download the [related file](#) (ZIP 1 KB).

Creating the XSLT

The XLST should be created according to the content contained in **Appendix A** of this document.

Customizing the XLST for your Environment

The XLST has all of the necessary parameters for modification located at the top for ease of use in customization. You'll notice the following fields:

```
<xsl:param name="NULLDateValue">zz</xsl:param> <!-- Default value for Date entries -->
<xsl:param name="NULLDoubleValue">dd</xsl:param> <!-- Default value for Double entries -->
<xsl:param name="NULLIntegerValue">ii</xsl:param> <!-- Default value for Integer entries -->
<xsl:param name="NULLStringValue">ss</xsl:param> <!-- Default value for String entries -->
<xsl:param name="NULLBooleanValue">bb</xsl:param> <!-- Default value for boolean entries -->
```

In modifying the values set for these five parameters the default values for each of the null entries in your return Illuminator Document XML will be replaced based on the SQLDataType defined in the column definition.

Using an XSLT in a Query

In order to test out and use the XLST in your MII environment you will first need to define the XSLT in your MII environment. This XSLT (See Appendix A) should be saved in the WEB content and then referenced in your Query Template via the Template Category named Transformation. For Example the Inline Transformation field can be set as:

```
server://XMII/CM/SAP/Common/XSLT/FormatNullValuesInIllumDoc.xml
```

Then for the query in order to test the operation of each of the data types you can use the following NULL Query:

```
SELECT 'Test Label' as Label, cast(null as int) as NullInteger, cast(null as varchar) as NullString, cast(null as datetime) as NullDate, cast(null as real) as NullReal, cast(null as float) as NullFloat
```

Here is a sample response from the above example query that uses this XSLT:

Rowset 1					
Label	NullInteger	NullString	NullDate	NullReal	NullFloat
Test Label	ii	ss	zz	dd	dd

From here you can then configure the transformation parameters to map values in from runtime to the query template via the XParam.N parameter names.

Inline Transform

Trans. Param.	Transform Parameter Name	Transform Parameter Value
1	NULLDateValue	Date
2	NULLDoubleValue	Double
3	NULLIntegerValue	Integer
4	NULLStringValue	String
5	NULLBooleanValue	Boolean
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		

Query Results - SAP/BugTest/FormatNullValuesInIllumDocQuery

URL

Rowset 1

Label	NullInteger	NullString	NullDate	NullReal	NullFloat
TestLabel	Integer	String	Date	Double	Double

The ability to set the default handling for NULL values of any type provides greater control and flexibility over the handling of NULL values in your application.

Appendix A

The XSLT used to define custom null values in an Illuminator Document XML:

```
<!--
```

Use this transform to convert the default SAP MII XML null values into your own custom values based on the data type of the column

Author: Salvatore Castro

Date: December 3, 2009

Last Modified On: February 15, 2010

```
-->
```

```
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:java="http://xml.apache.org/xslt/java" exclude-result-prefixes="java">
<xsl:output method="xml" media-type="text/xml" encoding="UTF-8"/>
  <xsl:param name="NULLDateValue">zz</xsl:param> <!-- Default value for Date entries -->
  <xsl:param name="NULLDoubleValue">dd</xsl:param> <!-- Default value for Double entries -->
  <xsl:param name="NULLIntegerValue">ii</xsl:param> <!-- Default value for Integer entries -->
  <xsl:param name="NULLStringValue">ss</xsl:param> <!-- Default value for String entries -->
  <xsl:param name="NULLBooleanValue">bb</xsl:param> <!-- Default value for boolean entries -->
<xsl:template match="/">
  <Rowsets>
    <Rowset>
      <xsl:for-each select="Rowsets/Rowset">
        <xsl:variable name="CurrentColumns" select="Columns"/>
        <!-- From the MII XML simply copy the row information and it's child nodes to the result XML
document -->
        <xsl:copy-of select="/Rowsets/Rowset/Columns" />
          <xsl:for-each select="Row">
            <Row>
              <xsl:for-each select="*">
                <xsl:variable name="CurrentNodeName" select="name(.)"/>
                <xsl:call-template name="buildrowitem">
                  <xsl:with-param name="SQLDataType"><xsl:value-of
select="$CurrentColumns/Column[@Name=$CurrentNodeName]/@SQLDataType"/></xsl:with-param>
                  <xsl:with-param name="NodeName"><xsl:value-of
select="$CurrentNodeName"/></xsl:with-param>
                  <xsl:with-param name="NodeValue"><xsl:value-of
select="."/></xsl:with-param>
                  <xsl:with-param name="NULLDateValue"><xsl:value-of
select="$NULLDateValue"/></xsl:with-param>
                  <xsl:with-param name="NULLDoubleValue"><xsl:value-of
select="$NULLDoubleValue"/></xsl:with-param>
                  <xsl:with-param name="NULLIntegerValue"><xsl:value-of
select="$NULLIntegerValue"/></xsl:with-param>
                  <xsl:with-param name="NULLStringValue"><xsl:value-of
select="$NULLStringValue"/></xsl:with-param>
                  <xsl:with-param name="NULLBooleanValue"><xsl:value-of
select="$NULLBooleanValue"/></xsl:with-param>
                </xsl:call-template>
              </xsl:for-each> <!-- Each cell in the rows -->
            </Row>
          </xsl:for-each> <!-- Row -->
        </xsl:for-each> <!-- Rowsets/Rowset -->
      </Rowset>
    </Rowsets>
  </xsl:template>

<xsl:template name="buildrowitem" match="*">
```

```

<xsl:param name="SQLDataType" />
<xsl:param name="NodeName" />
<xsl:param name="NodeValue" />
<xsl:param name="NULLDateValue" />
<xsl:param name="NULLDoubleValue" />
<xsl:param name="NULLIntegerValue" />
<xsl:param name="NULLStringValue" />
<xsl:param name="NULLBooleanValue" />

<xsl:element name="{ $NodeName }">
  <xsl:choose>
    <!-- DateTypes:          91 = Date          92 = Time
93 = DateTime -->
    <xsl:when test=" $SQLDataType = '91' or $SQLDataType = '92' or $SQLDataType =
'93' ">
      <xsl:choose>
        <xsl:when test="string($NodeValue) = 'TimeUnavailable' ">
          <xsl:value-of select=" $NULLDateValue"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of select=" $NodeValue"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:when>
    <!-- DoubleTypes:      2 = Numeric      3 = Decimal
6 = Float      7 = Real      8 = Double -->
    <xsl:when test=" $SQLDataType = '2' or $SQLDataType = '3' or $SQLDataType = '6'
or $SQLDataType = '7' or $SQLDataType = '8' ">
      <xsl:choose>
        <xsl:when test="string($NodeValue) = 'NA' or string($NodeValue) =
'NaN' ">
          <xsl:value-of select=" $NULLDoubleValue"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of select=" $NodeValue"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:when>
    <!-- IntegerTypes:    4 = Long Integer    5 = Short Integer
-->
    <xsl:when test=" $SQLDataType = '4' or $SQLDataType = '5' ">
      <xsl:choose>
        <xsl:when test="string($NodeValue) = 'NA' or string($NodeValue) =
'NaN' ">
          <xsl:value-of select=" $NULLIntegerValue"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of select=" $NodeValue"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:when>
    <!-- Types:          -7 = Bit -->
    <xsl:when test=" $SQLDataType = '-7' ">
      <xsl:choose>
        <xsl:when test="string($NodeValue) = 'NA' or string($NodeValue) =
'NaN' ">
          <xsl:value-of select=" $NULLBooleanValue"/>
        </xsl:when>

```

```
        <xsl:otherwise>
            <xsl:value-of select="$NodeValue"/>
        </xsl:otherwise>
    </xsl:choose>
</xsl:when>
<!-- If the type is anything else then assume it's a string -->
<xsl:otherwise>
    <xsl:choose>
        <xsl:when test="string($NodeValue) = '---' ">
            <xsl:value-of select="$NULLStringValue"/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:value-of select="$NodeValue"/>
        </xsl:otherwise>
    </xsl:choose>
</xsl:otherwise>
</xsl:choose>
</xsl:element>
</xsl:template>
</xsl:stylesheet>
```

Related Content

SAP MII Wiki: <https://wiki.sdn.sap.com/wiki/display/xMII>

SAP MII Forum: <https://www.sdn.sap.com/irj/sdn/forum?forumID=237>

SAP Manufacturing Downloads: <http://www.sdn.sap.com/irj/scn/manufacturing-tools>

Copyright

© Copyright 2010 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.