

Color-Contexting in the SAP XMII IGrid Using Multiple Conditional Logic

Applies to:

SAP XMII 11.5.x

Summary

The XMII iGrid control provides a powerful but easy-to-use grid component for viewing data from virtually any of the SAP XMII connectors. It also provides color-context sensitive highlighting to accentuate data elements based on certain values or value ranges. The purpose of this document is to instruct and show examples of the color-contexting feature of the SAP XMII iGrid component when multiple conditional logic is required to do the highlighting.

Author(s): Diana Hoppe

Company: SAP Labs, LLC

Created on: 7 December 2006

Author Bio

Diana Hoppe has worked for Lighthammer Software since November 2004 supporting the Lighthammer CMS product. Since the SAP acquisition of Lighthammer in July 2005 Diana has become a Technical Support Consultant for SAP xMII. She has a B.S. Computer Science and is currently working toward the completion of a Masters degree in Computer Science at West Chester University.

Table of Contents

Introduction	2
Procedure	3
Setting Up the Transaction Output Property	3
Setting up the Tag Query Action Block	4
Setting up the Calculated Columns Action Block.....	5
Setting up the Assignment Action Block	6
Save the Transaction	7
Viewing the Result	7
Setting Up the Display Template	9
Conclusion	14
Copyright.....	14

Introduction

The color-contexting feature of the xMII iGrid component is a great way to highlight elements based on certain values or value ranges. However, when color-contexting is processed in Number Match Mode, a less-than-or-equal-to (<=) row-by-row comparison based upon the number is done. If the comparison is not true, the operation continues until a match is found. If color-contexting is based upon a single value (one column in the query output), the color-contexting works as you would expect. However, trying to use more than one numeric column to do the color-contexting value can be problematic because the conditions of multiple columns are “anded” together. For example, say you wanted to color-context a row under this condition: if column1_value <= 0 AND column2_value <=0, then highlight the row in red - this works just fine. But what if you want to highlight the row if column1_value <= 0 OR column2_value <= 0? Or if column1_value > 50 AND column2_value < 100? This document will show you how to convert these conditions into a single String representation in order to overcome this limitation.

Additionally, color-contexting in the Display Template Editor is limited to matching either String or Number values; it doesn't handle matching both at once; this problem too can be solved by conversion to a single String representation.

One way to get over these hurdles is to add a match column in your SQL query such as:

```
(CASE WHEN Category = 'Engineering Stops' AND Time >= 6 THEN 'R'  
  
      ELSE 'G' END) As ColorIndicator
```

Thus, a single String value represents multiple conditional logic statements. Then, use the String Match Mode to do the color-contexting where R = red and G = green (for example), and setting the ColorIndicator column width=0 to hide it from view.

Another way to accomplish this is to use Business Logic Services to provide a **Calculated Column** called something like “ColorIndicator”. In the grid, use the ColorIndicator column as your context-sensitive coloring mechanism. In Business Logic Services, you could use almost any complex rule/expression you wanted to

set the ColorIndicator value. It is always best to do this type of work on the server side if possible, as this makes it much easier to manage and maintain.

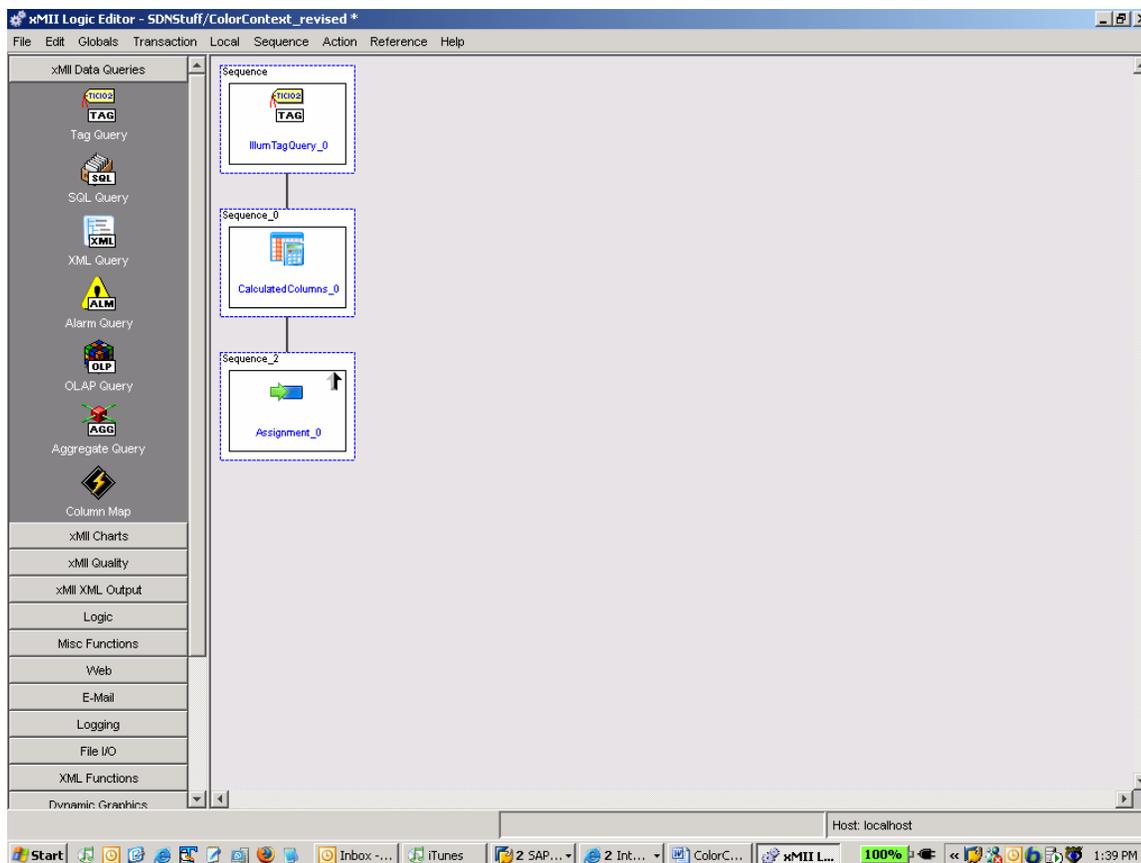
Some things to remember:

1. The new column name must be mapped in on the Data Mapping tab of the Display Template; on the Layout tab, you can adjust its width to zero so the column is not displayed.
2. Make sure the Match Mode is set to String.

Procedure

Open up the Logic Editor found under Business Logic Services in the xMII main menu.

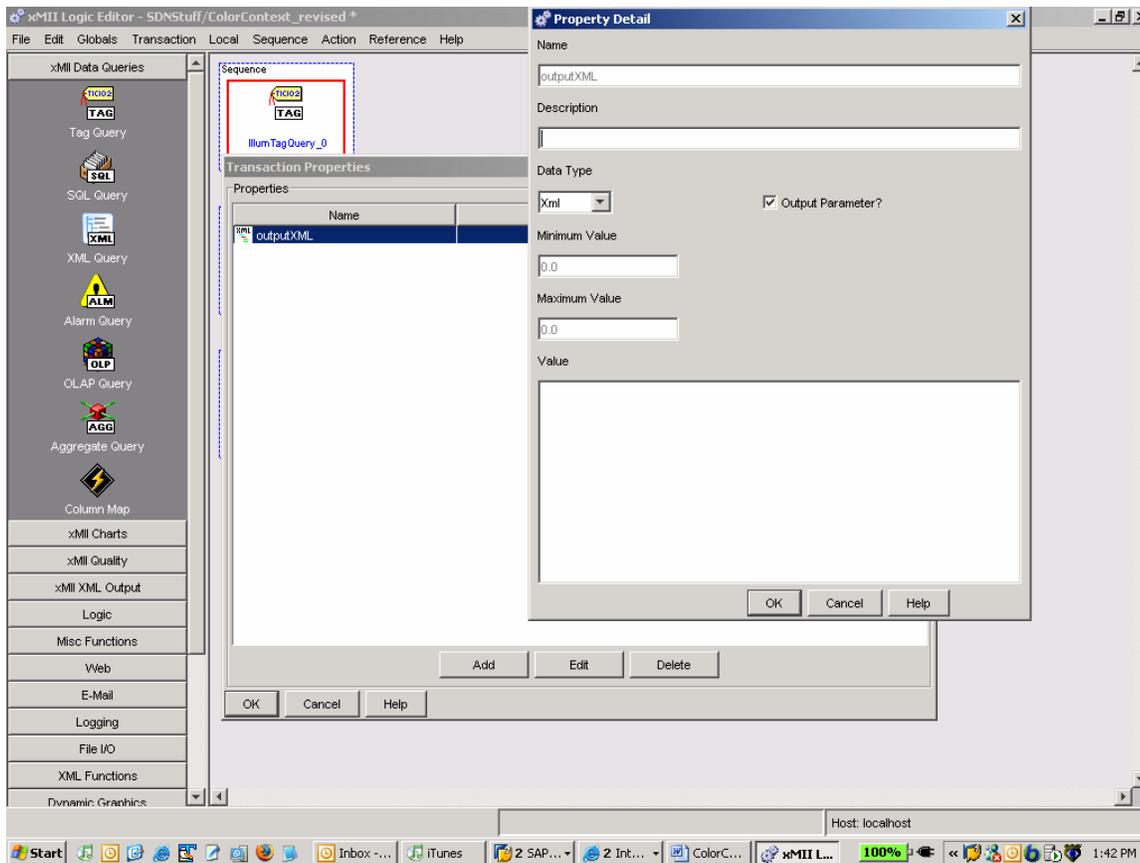
This is the overall transaction that we will set up step by step:



Setting Up the Transaction Output Property

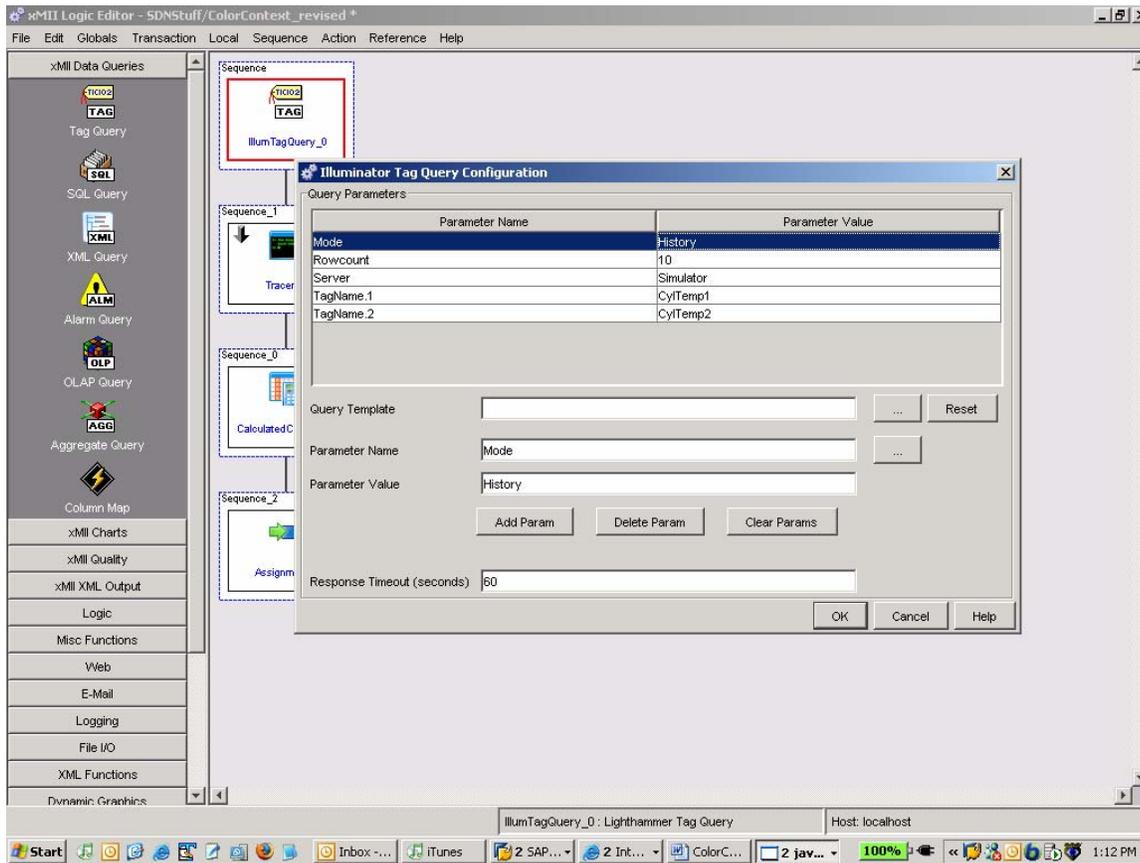
1. Click on the Transaction menu item at the top of the Logic Editor.
2. Click Add and create a property called "outputXML" of Data Type Xml; make sure the Output Parameter? checkbox is checked.
3. Click OK.

4. Click OK again.



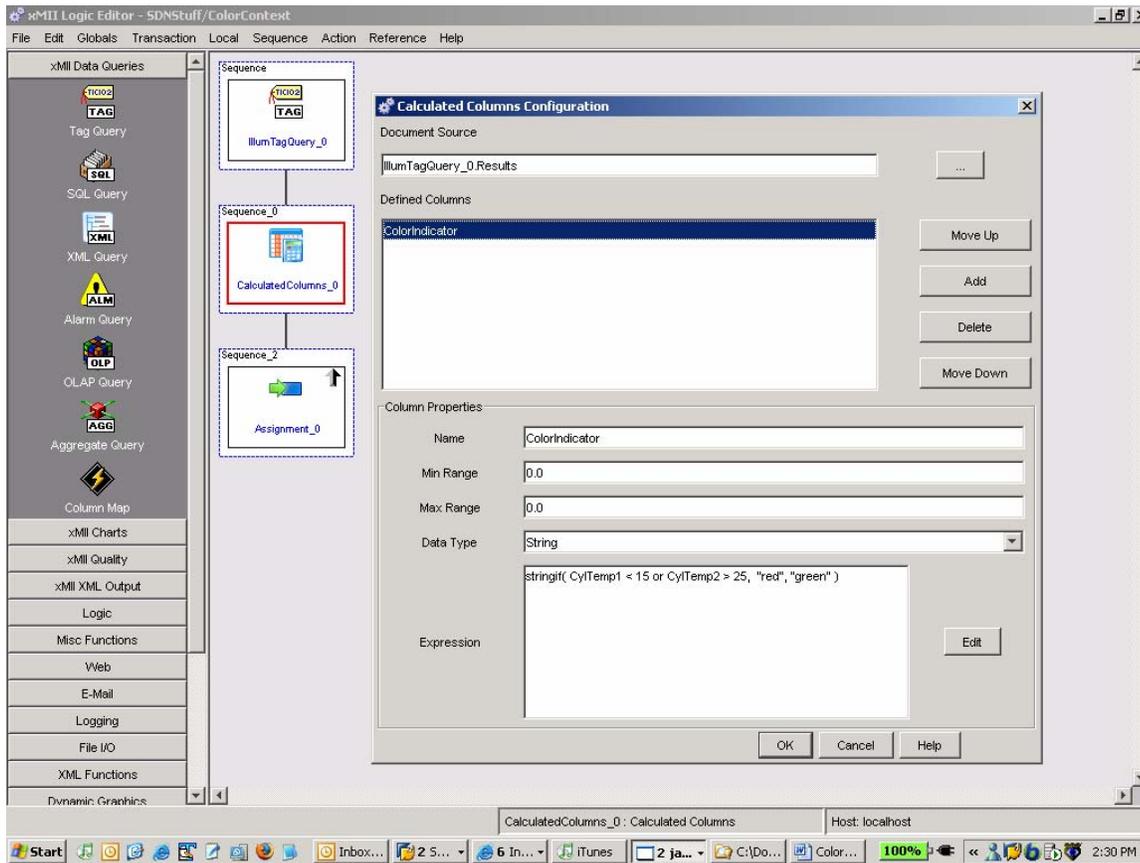
Setting up the Tag Query Action Block

1. Add a Tag Query action block (found under xMII Data Queries) to the initial sequence.
2. Right click on the action block and select 'Configure' to open the Configuration Editor.
3. Add the following parameters by filling in the parameter name and parameter value and clicking the "Add Param" button:
 - Mode History
 - Server Simulator
 - Tagname.1 CylTemp1
 - Tagname.2 CylTemp2
4. Click OK to close the Editor.



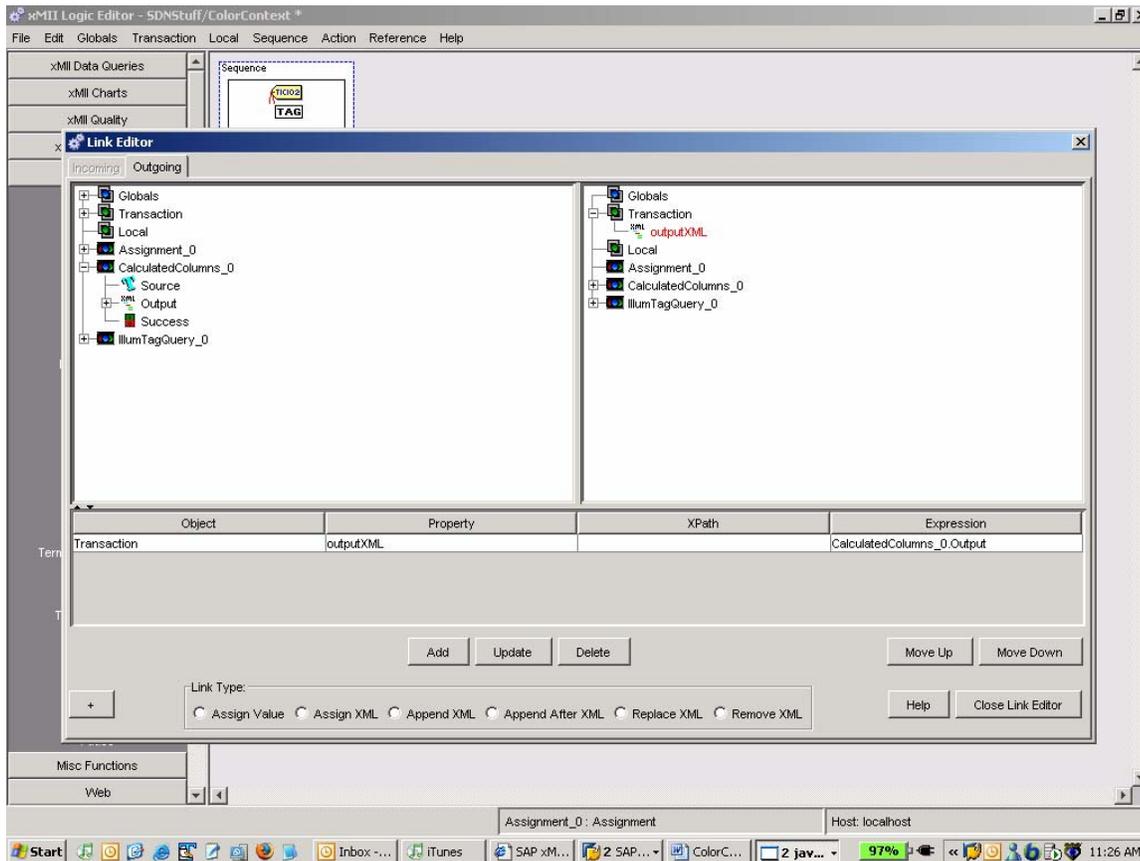
Setting up the Calculated Columns Action Block

1. Add a sequence by right-clicking on the IllumTagQuery action block and clicking 'Add Sequence'.
2. From the XML Functions menu item, add a Calculated Columns action.
3. Right click on this action block and select 'Configure' to open the Configuration Editor. Use the Browse button to configure the Document Source to be the result of the Illuminator Tag Query.
4. Next, define a column called 'ColorIndicator' of Data Type String.
5. In the Expression box, type `stringif(CylTemp1 < 15 or CylTemp2 > 20, "red", "green")`, then click the 'Add' button.
6. Click OK. (Note: you may need to experiment with the comparison values to produce the two different colors).



Setting up the Assignment Action Block

1. Add a sequence by right-clicking on the Calculated Columns action block and clicking 'Add Sequence'.
2. From the Logic menu item, add an Assignment action.
3. Right click on this action block and click 'Links' to open up the Link Editor.
4. Expand the Calculated Columns action in the left-hand pane; expand the Transaction option in the right-hand pane. Link the Output of the Calculated Columns action to the outputXML Transaction property we created earlier.
5. Close the Link Editor.

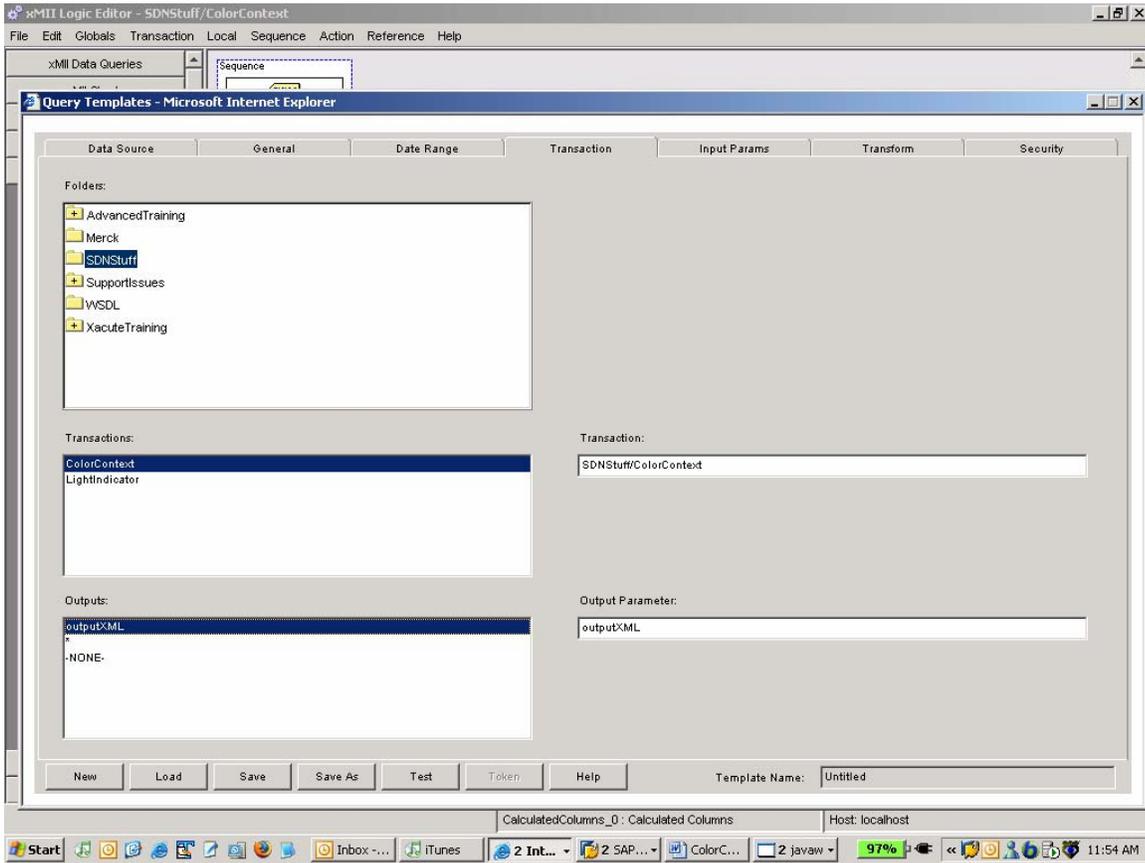


Save the Transaction

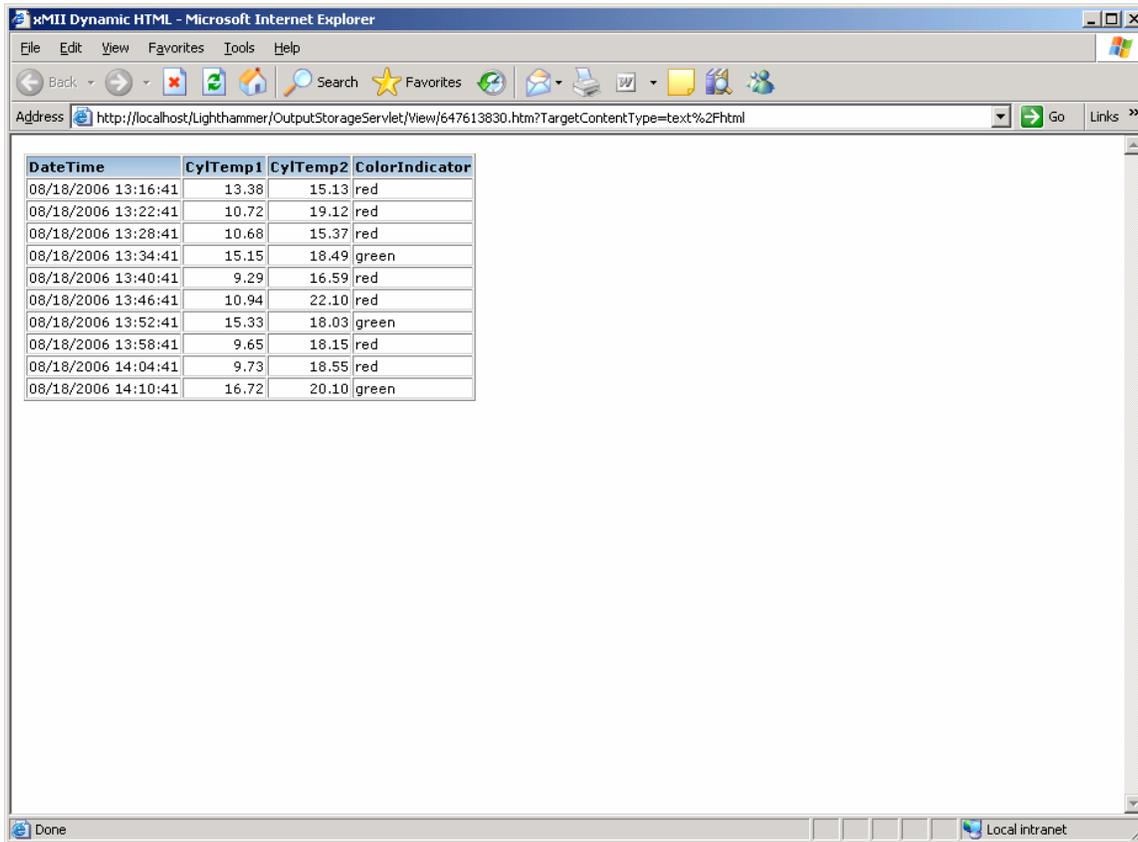
Name the transaction and save it. If any errors occur, recheck the configuration. Run the transaction by pressing F5 (Execute) or F6 (Execute Without Step Trace) to make sure there are no errors.

Viewing the Result

1. To view the result of this transaction, open up the Query Template Editor and click the 'New' button, then select 'XacuteQuery' as the template type and click OK.
2. On the Transaction tab, from the Folders box, click on the folder containing the transaction.
3. In the Transactions box, click on the transaction name.
4. In the Outputs box, click on the 'outputXML' property we defined in the transaction..

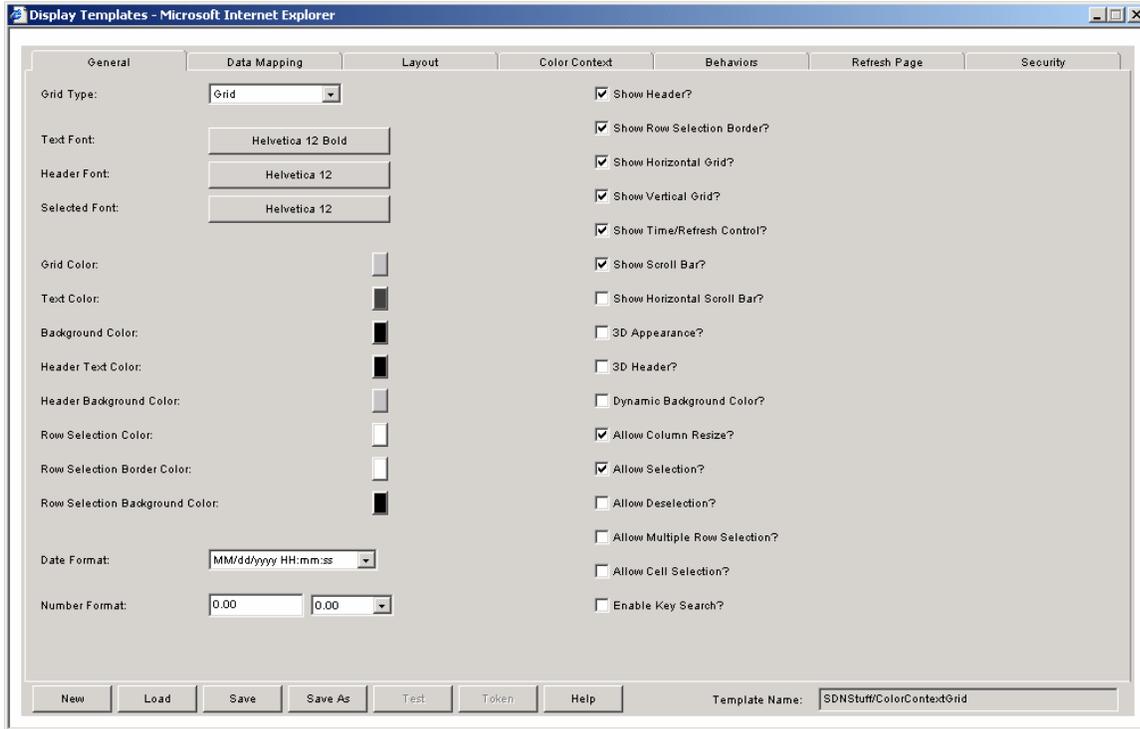


5. Click on the 'Test' button and then click on OK to view the results of this query. The result set should look something like this:

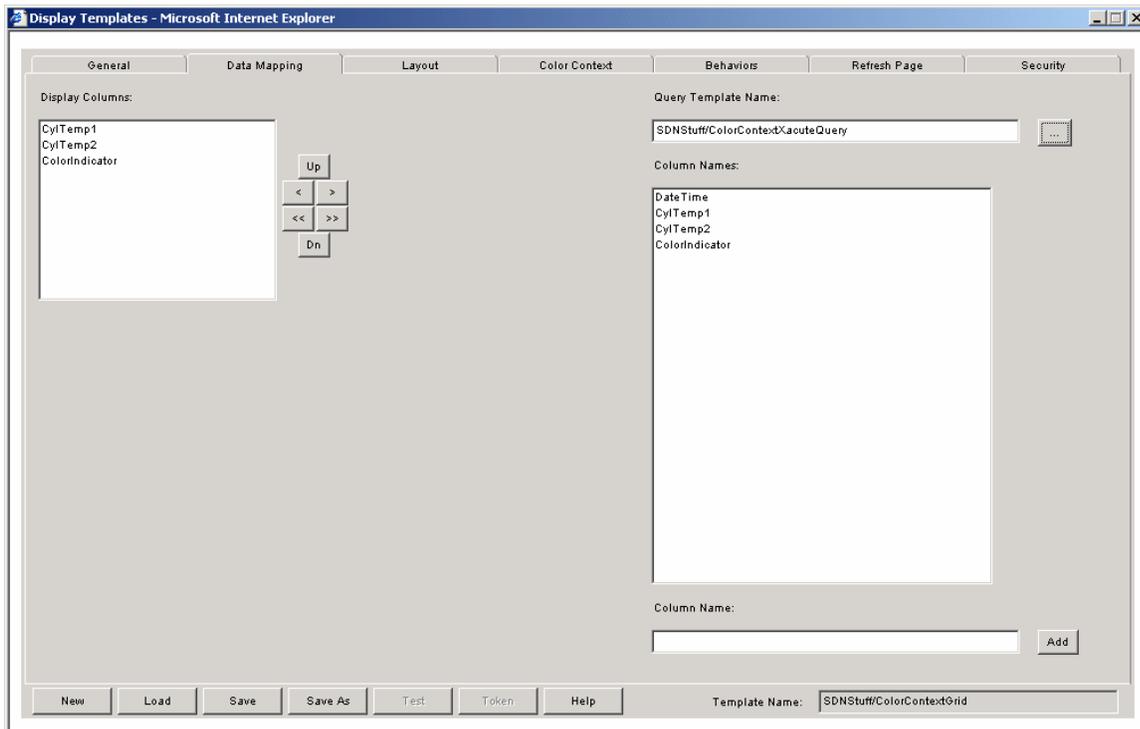


Setting Up the Display Template

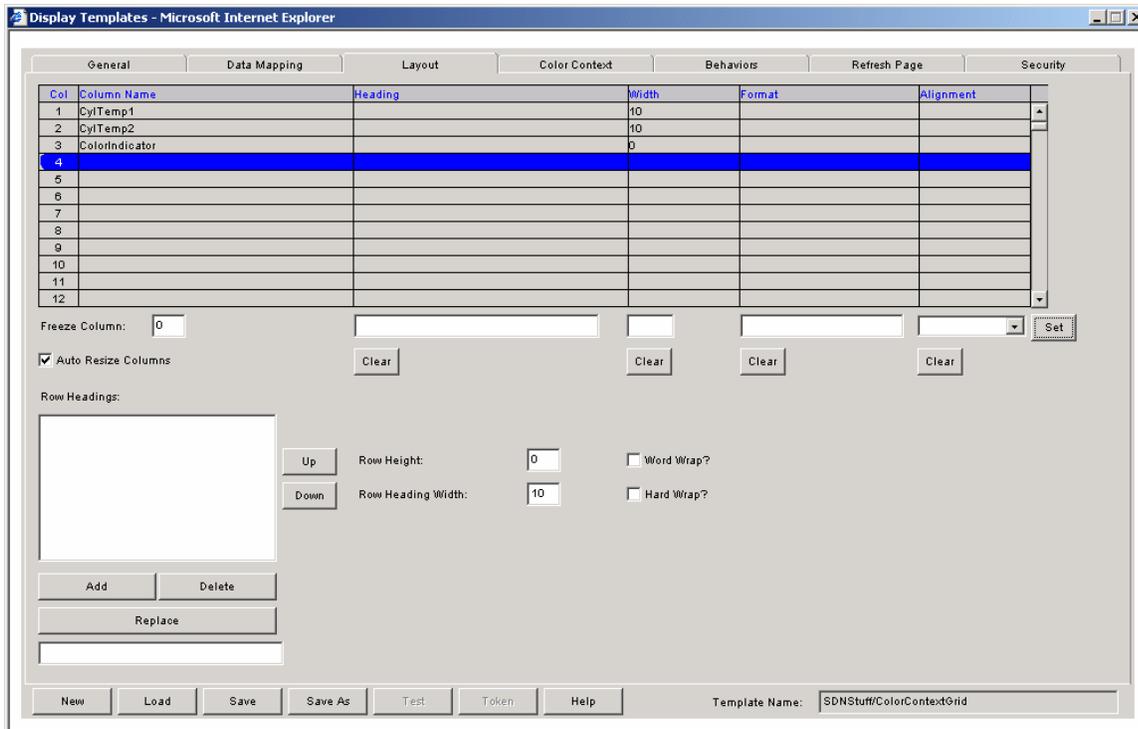
1. Create a new iGrid template. Make sure the Grid Type is "Grid".



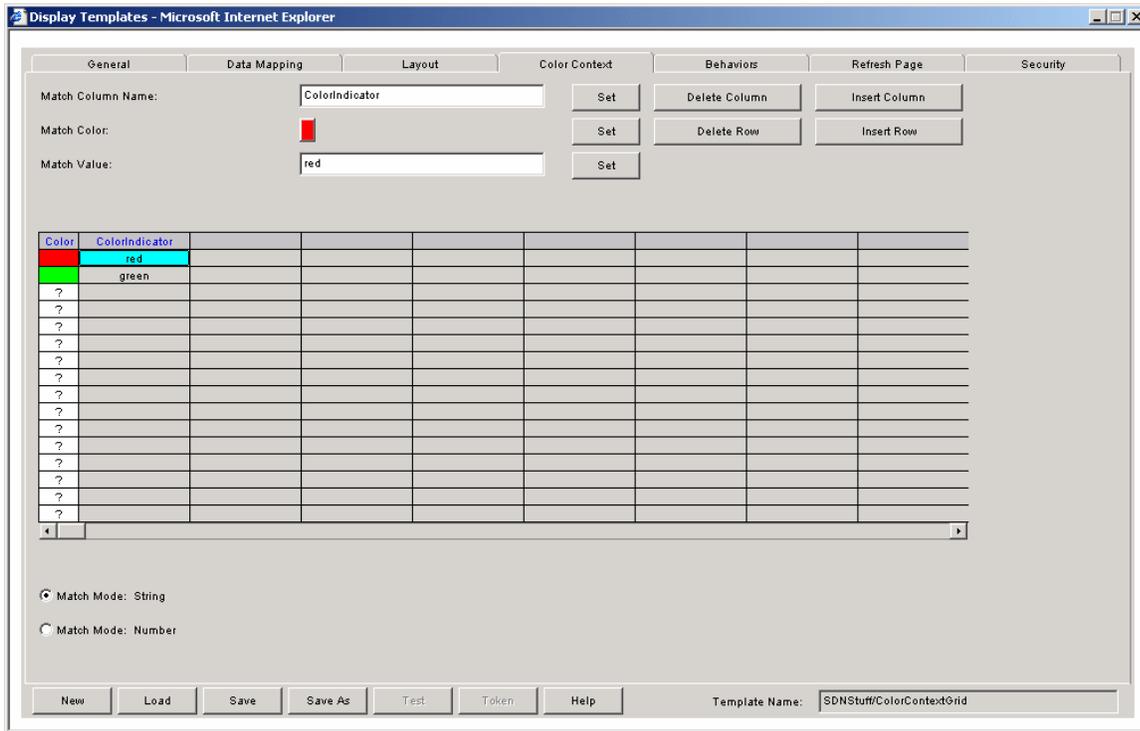
2. Click on the Data Mapping Tab. Browse for and select the XacuteQuery you created. Map the CylTemp1, CylTemp2, and ColorIndicator columns into the Display Columns area on the left.



3. Click on the Layout tab. Note that the ColorIndicator column can have a Width of “0” if you don’t want to display it in the grid.



4. Click on the Color Context tab. Type in “ColorIndicator” as the Match Column Name, choose red as the Match Color and type in “red” (one of the String values our query is returning) as the first Match Value – make sure to click the set button after each. Repeat the procedure with the Match Color as green and the Match Value as “green” (the other String value our query is returning). Make sure the Match Mode is “String”



5. Save the Display Template. Create an applet using the Dynamic Page Generator to view the result. Your applet should look something like this:

Copyright

© Copyright 2006 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, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

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.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, 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 in several other countries all over the world. 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.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

Any software coding and/or code lines/strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.