

Revision 15

Modifying Live Portals

Table of Contents

Live Portal Directory Structure:	3
Weekly/Monthly Updates	3
Using Webviews in REV15	4
Tree View Manager (NL0107)	4
Interactive Updates of key files	5
Creating a Portal	6
DHTML Grids & Pivot Tables	7
Google Charts/Graphs	8
RM Report	8
XM graph – example p_exec – DLYSALES graph	8
Manual Data – example p_sales – salesrep aging graph using AR04 aging totals.....	8
Apache Installation	9
Using Event Handler in Report Grids	11
Scenario 1:	11
Scenario 2:	12
Scenario 3:	15
Adding a link to call an object:	17
Setting Security:	17

Live Portal Directory Structure:

Html pages are automatically installed into:
swrun/nlhtml

When you log in for the first time for each company, we automatically create a NL control record. The directories defined in the control record are determined by the cblconfig used for the first login after the install.

NOTE:

- The conversion/installation creates a separate custom folder under nlhtml for you to use for modified portal pages.
- If you are not using thin client or terminal services, you will need to modify your NL control record "NL0101" and make sure the path to your nlhtml directory is the network UNC path.
- Weekly and Monthly updates contain html file changes and SouthWare file changes. These files are located in the swrun/swupd15 directory

Weekly/Monthly Updates

- Included with weekly and monthly updates, are changes to files in the nlhtml directory and usually RM format changes and NL file changes. To ensure these get installed for each company directory, a swupd15 folder is created with an extract id called UPDATE along with an nlhtml subfoder.
- The file UPDATE.X00 contains a Created Date. This date is used to determine if the updates should be installed for a company.

Extract Group: UPDATE REV15 NL/Live Portal Updates Created: 20090826 ----- ImportMate Formats - 1 Record Netlink Request Types - 2 Records NetLink RM Parameters - 1 Record
--

Each time you log into SouthWare, login will compare the created date in the UPDATE.X00 file against the "Last SouthWare Update" date in your company id record (field #15). If the date in UPDATE.X00 is more current than the company id date, login will install the extract id UPDATE into the company you are logging into and copy the contents of swrun/swupd15/nlhtml into your nlhtml folder defined in your NL control record.

NOTE:

If you do not want SouthWare to update your nlhtml folders or modify your NL files, modify each company id "Last SouthWare Update" to a date in the future such as 12/31/10.

These updates are cumulative and include all changes since the release of REV15.

Using Webviews in REV15

The REV15 installation will create or modify existing Webviews by using the “Web browser URL” option. To revert to the pre-15 version of the webviews, change display mode to “1” webpage for each zoom view in XX0901.

Tree View Manager (NL0107)

WIMAIN is the default portal that will display when logging into REV15.

WIMAIN

WIROLEMAIN

WIPEOPLEMAIN

WITASKMAIN

WIANALYSISMAIN

WILOGOFF

Interactive Updates of key files

New Data Source – NetLink WebEntry

How this works:

NetLink extracts the data sent from a web page form and writes it to an importable Fixed Position Text file. An optional format file tells NetLink which fixed position to use for each field.

The NetLink WebEntry option in ImportMate automatically creates the optional format file. ImportMate assigns positions to the fields you specify in the format, then writes the optional format file to tell NetLink what fields to write to which position. This happens behind the scenes so that all you need to do is make sure your field names in the format match your form field names and that you reference the option format file in the NetLink request that launches the import. And later if you change the format the optional format file will automatically be recreated to match the import definition.

To ensure that you are not updating a record that somebody else has updated:

```
<script>
function getTimeStamp() {
    var getdataurl =
"http://@var_form_action_url_@?session_no=@var_session_no_@&request_id=@getdata&@primaryfil
e=iv01&@stock_no=@var_dd_IV010001_@&@loc_no=@var_dd_IV010002_@&@template=getiv0101
02.txt";
    checkTimeStamp(getdataurl, '@var_dd_IV010102_@', 'statusarea', '@var_sw_webserver_mode_@',
'@var_ajax_file_name_@', 'http://@var_form_action_url_@', 'frmMain');
}
</script>
```

Example:

Add userfield #1 to the "simple stock edit"

- Modify Import format: WISTKEDT
- Add the import field UserField1 to wistkedit.htm in nlhtml folder

```
<td>Userl Field #1</td>
<td>
<input type="text" size=30 onchange="showChange(this)" onkeyup="limitText(this,this.size)"
name=UserField1 value="@var_dd_IV010058_@"></td>
```

Example:

Add option to change relationship type from Lead to Prospect

- Modify import format wireledt – add TYPE field
- Modify wireledit.htm

ADD:

```
<td width="50%" height="22">Type</td>
<td width="50%" height="22"><select size="1" name="Type">
    <option value="@var_dd_TW010003_@" selected="">@var_dd_TW040002_@
    </option>
    <option value="L">Lead</option>
    <option value="P">Sales Prospect</option>
</select></td>
```

Creating a Portal

The easiest way to create a new portal is to rename an existing portal and copying the html file associated with that portal to a new name.

We've included an example portal to be used for our class exercises.

- Copy flashdrive:\nlhtml.new\p_15class.htm into the nlhtml folder in your southware run directory
- Go to NL0102 – rename wip_slsmgr to WIP_15CLASS
- Change the html file in field #4 to p_15class.htm
- Clear out field #5 except for RM report NLTWOPDF
- Change Category Code to "C"lass

*1. Request ID	WIP_15CLASS	
2. Description	REV15 CLASS	
3. Requestor Type	I Internal	
4. HTML Template File	p_15class.htm	
5. +Use RM to Get Data?	<input checked="" type="checkbox"/> NLTWOPDF	
6. Security Group/Mask	Access Opr/Grp	
7. +Import Form Data?	<input type="checkbox"/> <input type="checkbox"/>	
8. Remove Temp Files?	<input checked="" type="checkbox"/>	
9. +Advanced Features	<input type="checkbox"/>	
10. +Comments		
11. Category Code	C CLASS	
12. +Use Suppl Process?		
13. Application ID	MASTER	

DHTML Grids & Pivot Tables

You can easily create a request with a grid on it by using the request wizard.

Let's modify the standard internal template request IRPTMP1.html file by deleting header and footer syntax.

Create a new request called CLASS1 using the request WIZARD.

The RM report should be called CLASS1 as well.

Create a RM report using AR01 as the primary file.

Fields on the RM report:

Customer#

Customer Name

State

SalesRep

YTD Sales

Now let's add this request to p_15class.htm – modify what menu1-1 calls by changing the request id to the one we just created CLASS1.

```
tree.setUserData("menu1-1", "saveurl", "http://@var_form_action_url_@?session_no=@var_session_no_@&request_id=class1");
```

Google Charts/Graphs

3 types of graphs:

RM Report

Create a RM report sorting by any field.

Now in NL0104 – create the NL parameter field #7 = “C” for google chart.

XM graph – example p_exec – DLYSALES graph

```
<p @var_xmgraph=DLYSALES_@>

<span class="@var_xmgraph=off_@"></span>
```

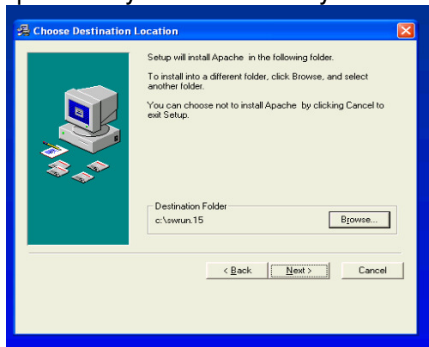
Manual Data – example p_sales – salesrep aging graph using AR04 aging totals.

```
@var_xmgraph=on_@
<p align="center">&nbsp;&nbsp; 
<span class="@var_xmgraph=off_@"></span>
```

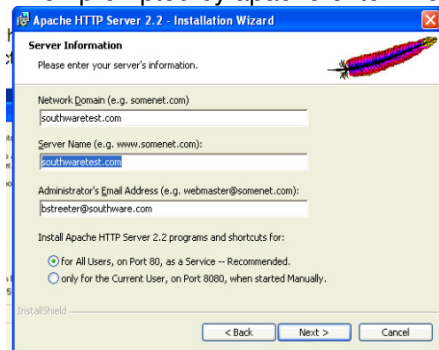

Apache Installation

- Download the self-extracting file apache.exe from the partnerzone.
- Unzip the file into any directory
- Execute installapache.exe

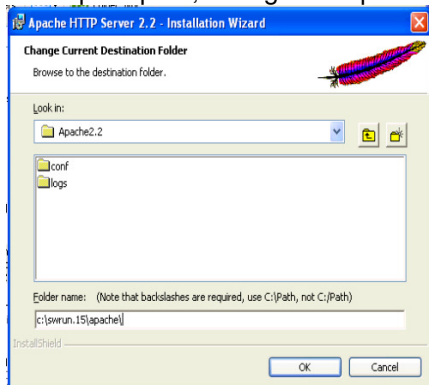
When prompted, enter the full path to your SouthWare run directory. This will create a directory named apache in your run directory.



When prompted by apache enter information about your server.



When prompted, change the apache installation directory to yoursouthwarerun\apache



When the installation is complete, double click on the apache icon and restart apache.

To test and make sure apache is installed correctly:

<http://localhost/nlhtml/index.htm>

Modify your NL control record. Field #2, base href must be changed to <http://servername/nlhtml>

02. Directories
Search Enter Help Exit

Local Standard Template Dir	C:\swrun.15\html\
Local Custom Template Dir	C:\swrun.15\html\custom\
Save Form Info Dir	C:\swrun.15\DM\IMPORT\
BASE HREF for Pages	http://localhost/nlhtml/
Remote AcuConnect directory	
Credit Card IF Data directory	
Logo Filename or URL	images/swlogo.jpg

Field #7, should be changed to servername/cgi-bin/NLNETUPD.CGI

5. E-Mail Contacts
6. +Date/Time of Data Live Data

7. URL for NLNETUPD.OBJ
localhost/cgi-bin/NLNETUPD.CGI

8. +Convenience Features

9. +Other Security

10. +Request Categories HAW

11. Template Editor Command

InterNet Explorer (Browser)

Field #9, change use SouthWare webprocessor to "N".

09. Other Security
Search Enter Help Exit

Does server support SSL?	N
Sys Admin Password	
Use SouthWare Web Processor?	N
Keep Internal Request History?	N

Using Event Handler in Report Grids

In the NL Report Parameter records (NL0104), if you are using the DHTMLX Grids (field 6 set to “D”), in field 7 Format Options you have the ability to set an Event Handler. This allows a user to click on a record in the report to have it fire an event.

There are three scenarios that we are going to look at using the event handler.

1. The records in the grid have a single key field (such as Customer #, Vendor #, etc.).
2. The records in the grid have multiple key fields (such as Stock #).
3. Depending on some value in the record, we want to take different actions.

Scenario 1:

For this example we are going to be looking at the Customer List report and when a record is selected it takes you to the details for that customer.

In NL0107, pull up the Request ID WICUSLST and in your HTML editor, pull up the file wicustst.htm. In NL0107 for the request WICUSLST, we want to zoom in to the RM Parameters and go to field 7 Format Options. The fields we’re going to look at are Record Key Definition, Show Selector Icon, and Has Event Handler.

*1. Report Name: WICUSLST Customer Search List

*2. A

3. F

4. F

5. C

6. F

7. F

8. G

9. E

10. Z

07. Format Options

Search Enter Help Exit

Grid Width	100%	
Grid Height	400	pixels
Show Toolbar?	Y	Excel? Y
Record Key Definition	1	First field in format
Show Selector Icon?	Y	
Show Totals?	Y	
Show Filters?	Y	
Has Preview Pane?	Y	Width: 33
Has Event Handler?	Y	
Show Pivot Table?	N	
- Action		
- Action Column	N/A	
- 'X axis' Column	N/A	
- 'Y axis' Column	N/A	

Exit

The Record Key Definition is used to tell the grid what is going to be used as the key to each record in the grid. For reports such as this one where the records in SouthWare only have one primary key field, this will usually be set to option 1 – First Field in the Format. You just need to make sure that the primary key field in SouthWare is your first field in your report format. For this customer listing, AR01 field 0001 Customer # is the first field in the RM format, which means that in the grid, the customer number is also going to be the key for each record.

The Show Selector Icon is what is used to show the green selector arrow in the first column of the grid. If you just want to show a report in the grid and are not planning on executing anything from the grid, set this field to “N” so you don’t have the arrow in the grid.

The Has Event Handler field tells the grid whether or not you are going to be firing events by clicking in the grid. If this is set to “N”, then clicking in the grid won’t do anything. If it is set to “Y”, then when we load the page it is going to be looking for the functions myOnRowSelected# (single click on the Selector Icon) and myOnRowDbClicked# (double click on a row in the grid). You then must set up these functions in the html page to tell the system what to do in each of these cases.

Now let’s look at the wicustlst.htm page and see how these functions are set up.

```

16 <script>
17 //function to execute when they click on or move to a row in the grid
18 function myOnRowSelected1(rowID,celInd){
19     var row_key=rowID;
20     var detailurl="http://@var_form_action_url_@?session_no=@var_session_no_@&request_id=wirelshpcc@cust_no="+row_key;
21     var previewurl="http://@var_form_action_url_@?session_no=@var_session_no_@&request_id=wiar01aj@cust_no="+row_key;
22     var activetab=tabbar.getActiveTab();
23     var curr_row_index=mygrid1.getRowIndex(mygrid1.getSelectedId());
24
25     if (celInd == 0 && activetab == "list") {
26         tabbar.setContentHref("details",detailurl);
27         tabbar.showTab("details");
28         tabbar.setTabActive("details");
29     }
30
31     if (activetab == "details") {
32         tabbar.setContentHref("details",detailurl);
33         tabbar.setTabActive("details");
34     }
35
36     var pvis = document.getElementById("previewpanel");
37     if(pvis.style.display == 'block') {
38         showGridPreviewPane(previewurl, 'previewpanel', '@var_sw_webserver_mode_@', '@var_ajax_file_name_@');
39     }
40 }
41
42 //function to execute when they double click on a row in the grid
43 function myOnRowDbClicked1(doublrowID,doublecelInd){
44     var doublerow_key=doublrowID;
45     var doubleurl="http://@var_form_action_url_@?session_no=@var_session_no_@&request_id=wirelshpcc@cust_no="+doublerow_key;
46     tabbar.showTab("details");
47     tabbar.setContentHref("details",doubleurl);
48     tabbar.setTabActive("details");
49 }
50 </script>

```

Notes:

Scenario 2:

For this example, we are going to be looking at the Stock List report and when you select a record it takes you to the details for the stock item. The Request ID we're going to look at is WISTKLST and the html file is wistklst.htm.

If we look at the RM Parameters for WISTKLST and go to field 7 Format Options, we see that in this report the Record Key Definition is set to option 2 – First field in format (don't include in grid). Since the stock record has two key fields, the stock number and the location, we can't use just one of those as the record key definition. For this report, we have a field in IV01 that is a combination of location plus stock number (IV01 field 104). We can use that as the first field in the report format so that each row in the grid has a unique key, but we don't want to show the field in the grid. We are showing the stock number and location in separate columns.

Because the stock record has two key fields, we need to pass both of those fields in our call to the stock details page. In order to do this, we need to capture the values from the grid for both the stock number and the location, so let's look at the wistklst.htm page to see how this is being done.

```

17 <script>
18 //function to execute when they click on or move to a row in the grid
19 function myOnRowSelected1(rowID,cellInd){
20     var row_key=rowID;
21     var stock_no=getCellValue(rowID,1);
22     var loc_no=getCellValue(rowID,2);
23     var detailurl="http://@var_form_action_url_@?session_no=@var_session_no_@&request_id=wiiw0lview&@stock_no="+stock_no+"&@loc_no="+loc_no;
24     var previewurl="http://@var_form_action_url_@?session_no=@var_session_no_@&request_id=wiiw0lpreview&@stock_no="+stock_no+"&@loc_no="+loc_no;
25
26     var activetab=tabbar.getActiveTab();
27     var curr_row_index=mygrid1.getRowIndex(mygrid1.getSelectedId());
28     if (cellInd == 0 && activetab == "list") {
29         tabbar.showTab("details");
30         tabbar.setContentHref("details",detailurl);
31         tabbar.setTabActive("details");
32     }
33
34     if (activetab == "details") {
35         tabbar.setContentHref("details",detailurl);
36         tabbar.setTabActive("details");
37     }
38
39     var pvis = document.getElementById("previewpanel");
40     if(pvis.style.display == 'block') {
41         showGridPreviewPane(previewurl, 'previewpanel', '@var_sw_webserver_mode_@', '@var_ajax_file_name_@');
42     }
43 }
44
45 function myOnRowDbClicked1(doublrowID,doublecelInd){
46     var doublerow_key=doublrowID;
47     var stock_no=getCellValue(doublrowID,1);
48     var loc_no=getCellValue(doublrowID,2);
49     var doubleurl="http://@var_form_action_url_@?session_no=@var_session_no_@&request_id=wiiw0lview&@stock_no="+stock_no+"&@loc_no="+loc_no;
50     tabbar.showTab("details");
51     tabbar.setContentHref("details",doubleurl);
52     tabbar.setTabActive("details");
53 }
54
55 function getCellValue(rowId,columnIndex) {
56     var value = mygrid1.cells(rowId,columnIndex).getValue();
57     return value;
58 }
59 </script>

```

For the most part, this is similar to the customer list except that we are setting the variables stock_no and loc_no with the value that is in those grid cells in the record that we selected. To do this we need to call the function getCellValue and pass it the rowID and column number of the appropriate cell. We get the rowID from the row that was selected and we have to tell the function what column to look at. The first column in the grid is always column zero. So looking at our stock listing in the grid:

Stock					
Stock Details					
	Stock Number	Location Number	Stock Description 1	Product Category	Quantity On Hand
	<input type="text"/>	<input type="text" value="001"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	*MISC	001	Miscellaneous Non Stock	PP	.000
	100	001	747 Swingline Desk Stapler	OS	1849.000
	100-BLUE	001	747 Swingline Desk Stapler	OS	1350.000
	100-TAN	001	747 Swingline Desk Stapler	OS	1175.000
	101	001	Swingline Half Size Stapler	OS	972.000
	101	002	Swingline Half Size Stapler	OS	838.000
	101	003	Swingline Half Size Stapler	OS	102.000
	105	001	Panasonic Phone/Answering Sys	OM	5.000
	105	002	Panasonic Phone/Answering Sys	OM	1.000
	105	003	Panasonic Phone/Answering Sys	OM	.000

The first column is taken up by the selector icon, so that would be column zero. The stock number is in column 1 and the location is in column 2. So to set our stock number we need to call `getCellValue(rowID,1)` and to get the location we call `getCellValue(rowID,2)`. The function `getCellValue` then goes out to the grid, grabs the value in that particular cell and returns it to our variable which we can then pass in our call to the `WIIV01VIEW` page.

Notes:

Scenario 3:

For this example, we are going to be looking at the Customer Activity History list. In this grid, when you select a row, we call different request ID's based on what type of record it is. The Request ID we're going to look at is WICUSACTHIST and the html file is wicusacthist.htm.

This example builds on the first two as we are going to pass a key value to the record we want to access, but we have to get that value from the grid along with the type of record that it is. Let's look at the wicusacthist.htm page to see how this is done:




```
20 <script>
21 //function to execute when they click on or move to a row in the grid
22 function myOnRowSelected1(rowID,celInd){
23     var row_key=rowID;
24     var file=getCellValue(rowID,6);
25     var activity_key=getCellValue(rowID,7);
26     var detailurl=null;
27     var previewurl=null;
28     var activetab=tabbar.getActiveTab();
29     var curr_row_index=mygrid1.getRowIndex(mygrid1.getSelectedId());
30
31     switch (file) {
32
33         case 'RS07':
34             detailurl="http://@var_form_action_url_@?session_no=@var_session_no_@request_id=wirs07view&ord_no="+activity_key+"&@sls_ord_no="+activity_key;
35             previewurl="http://@var_form_action_url_@?session_no=@var_session_no_@request_id=wirs07preview&ord_no="+activity_key+"&@sls_ord_no="+activity_key;
36             break;
37
38         case 'SV04':
39             detailurl="http://@var_form_action_url_@?session_no=@var_session_no_@request_id=wisv04view&cust_no=@var_cust_no_@&@svc_ord_no="+activity_key;
40             previewurl="http://@var_form_action_url_@?session_no=@var_session_no_@request_id=wisv04preview&cust_no=@var_cust_no_@&@svc_ord_no="+activity_key;
41             break;
42
43         case 'TW10':
44             detailurl="http://@var_form_action_url_@?session_no=@var_session_no_@request_id=witaskedit&task_id="+activity_key+"&@task_id="+activity_key;
45             previewurl=null;
46             break;
47
48         case 'RS10':
49             detailurl="http://@var_form_action_url_@?session_no=@var_session_no_@request_id=wirs10view&@inv_no="+activity_key;
50             previewurl="http://@var_form_action_url_@?session_no=@var_session_no_@request_id=wirs10preview&@inv_no="+activity_key;
51             break;
52
53         case 'JC02':
54             detailurl="http://@var_form_action_url_@?session_no=@var_session_no_@request_id=wijc02view&@job_no="+activity_key;
55             previewurl=null;
56             break;
57
58         case 'TW09':
59             detailurl="http://@var_form_action_url_@?session_no=@var_session_no_@request_id=witw09view&project_id="+activity_key;
60             previewurl=null;
61             break;
62
63         case 'SV02':
64             detailurl="http://@var_form_action_url_@?session_no=@var_session_no_@request_id=wisv02view&@cust_no=@var_cust_no_@&@SVC_CTR_NO="+activity_key;
65             previewurl="http://@var_form_action_url_@?session_no=@var_session_no_@request_id=wisv02preview&@cust_no=@var_cust_no_@&@SVC_CTR_NO="+activity_key;
66             break;
67
68         default:
69             if (celInd == 0 && activetab == "list") {
70                 alert ("Zoom to this Activity not Available");
71             }
72     }
73
74     if (celInd == 0 && activetab == "list" && detailurl != null) {
75         tabbar.showTab("details");
76         tabbar.setContentHref("details",detailurl);
77         tabbar.setTabActive("details");
78     }
79
80     if (activetab == "details") {
81         tabbar.setContentHref("details",detailurl);
82         tabbar.setTabActive("details");
83     }
84
85     var pvis = document.getElementById("previewpanel");
86     if(pvis.style.display == 'block') {
87         showGridPreviewPane(previewurl, 'previewpanel', '@var_sw_webserver_mode_@', '@var_ajax_file_name_@');
88     }
89 }
90
```

To determine where we want to go, we are capturing the primary file of each record and also capturing the key to that record using the getCellValue function that we saw in the previous example. Then using the switch statement, we test that primary file and based on what it is we call different request ID's and pass it the key value to that particular record.

So if we look at the Customer Activity History listing we see:

Activity List

Last 30 Days ▾ 09/03/09 thru 10/03/09 Submit



	Type	Date	Reference	Date Type	Time	File	Activity Key Value
➔	Credit Memo	09/21/09	Order #: 203	Order Date	0	RS07	203
➔	Customer Return	09/27/09	CRA # 0000000007 Sales Order: 181 0010	Date Initiated	0	RA03	00000000070010
➔	Credit Memo	09/27/09	Order #: 181	Order Date	0	RS07	181
➔	Service Order	09/27/09	Svc Order #: 00100000067 Scheduled: 09/27/09	Date Received	943	SV04	00100000067
➔	Customer Return	10/02/09	CRA # 0000000006 Sales Order: 191 0010	Date Initiated	0	RA03	00000000060010
➔	Customer Return	10/02/09	CRA # 0000000009 S/O Number: 00100000068 0010	Date Initiated	0	RA03	00000000090010
➔	Sales Order	10/02/09	Order #: 196	Order Date	0	RS07	196
➔	Invoice	10/02/09	Invoice #: 313 Origin: Sales Billing	Invoice Date	0	RS10	313
➔	Invoice	10/02/09	Invoice #: 316 Origin: Sales Billing	Invoice Date	0	RS10	316
➔	Invoice	10/02/09	Invoice #: 317 Origin: Sales Billing	Invoice Date	0	RS10	317
Total Records: 11							

We are capturing the file from column 6 and the key to that record in column 7. So if we were to select the first record in the list above, we get RS07 as the file, so we would call the request ID WIRS07VIEW and pass it the order number 203. Now if we were to select the second record, the file is RA03. RA03 does not match any of the cases that we currently have set up, so it goes on to the default where we just give a message that the zoom to this particular activity is not available.

Notes:

