

Technical Approaches to MPage Development

The CCLS Approach to Building MPages

Monday October 5, 2009

10:30 AM – 12:00 PM



Presenter

Joshua Faulkenberry

Senior Web Applications Engineer

- Project Lead, MPages Development
- Head Client Architect, Web Services



Session Objectives

Background

- Packard's MPage Implementation
- Definitions
- The CCL as a Service Architecture

The CCLS Approach to Building an MPage

- Recommended Expertise
- The CCLS Build Process
- Best Practices

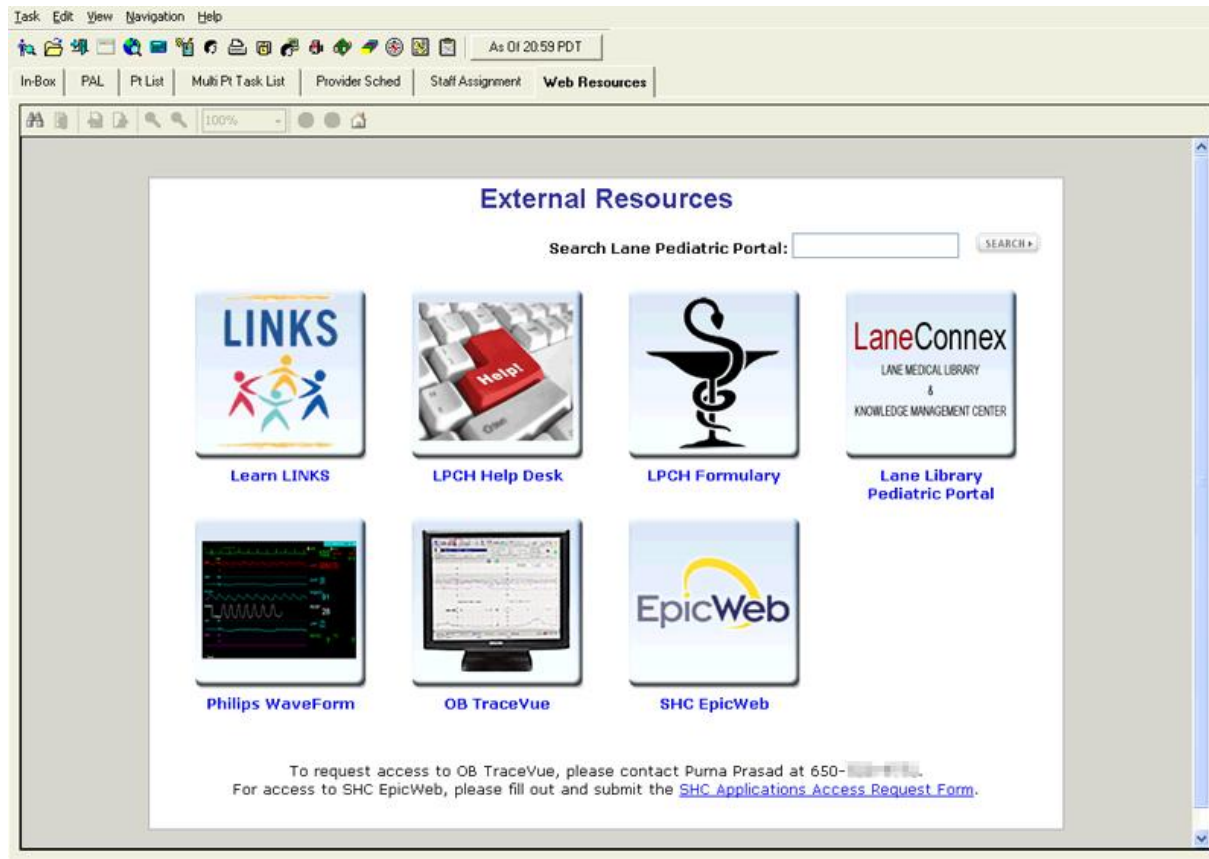


BACKGROUND



Packard's MPage Implementation

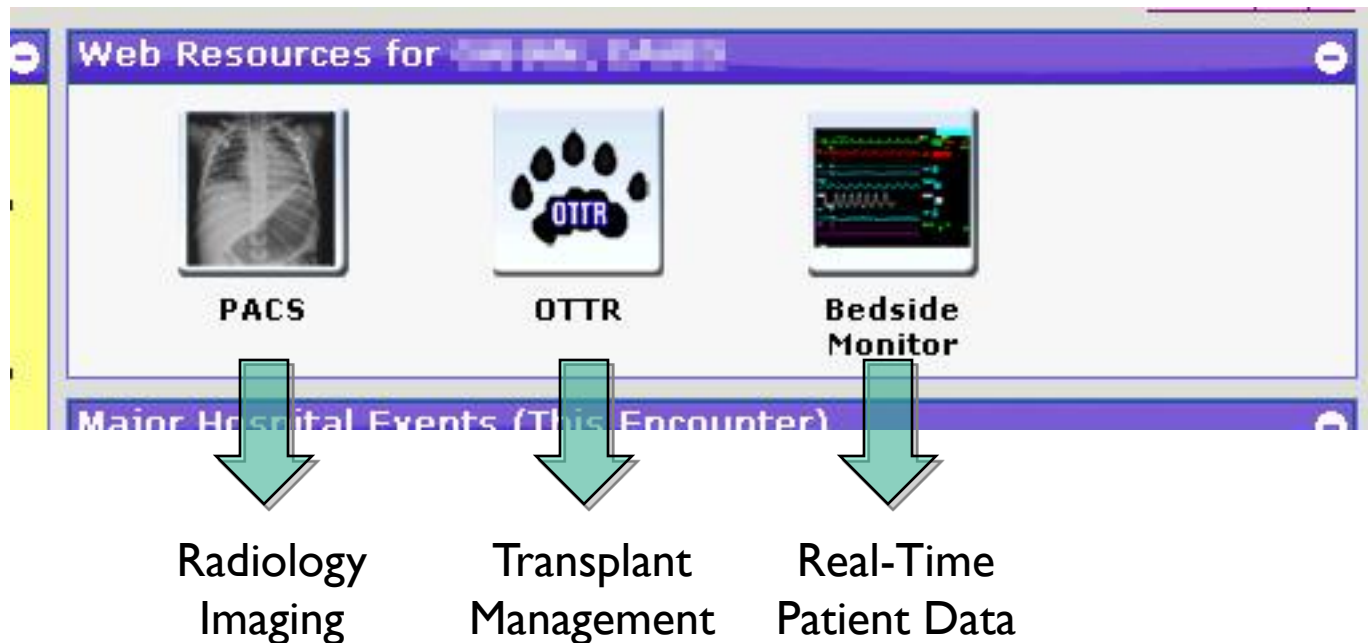
External Resource Integration (At the Organizer level)



Packard's MPage Implementation

External Resource Integration (At the Patient level)

Links capitalize on patient data to provide virtual **single sign-on** to external patient info



Packard's MPage Implementation

Packard is Remote Hosted

Unique challenges when implementing MPages as remote hosted client

- Bandwidth Limitations
 - Data from external sources use excess bandwidth, as it is routed through Citrix to the client network
- Resource Placement
 - Remote hosts are encouraged to place MPage resource files into separate network shares accessible to all domains



Definitions

AJAX

(Asynchronous JavaScript and XML)

- Allows web applications to send/retrieve data from server in background

JSON

(JavaScript Object Notation)

- Text-based, human-readable format for representing data structures and associative arrays, called objects

SOA

(Service Oriented Architecture)

- Aims at *loose coupling* of programming languages and other technologies that underlie applications



Definitions

CCLS

(CCL as a Service)

- Adaptation of SOA where CCL programs function like a Web Service
 - Establishes clear, well defined separation between client-side and CCL development
 - Facilitates modularization and reusability of code
 - Implementable in combination with other means of MPage generation



Definitions

XMLCclRequest

- Facilitates asynchronous communication between MPages and the Cerner Backend (providing AJAX functionality)

```
1 <script type="text/javascript" src="mpages.js"></script>
2 <script type="text/javascript">
3     function reportStatus() {
4         if (myReq.readyState == 4) {
5             alert(myReq.responseText);
6         }
7     }
8     var myReq = new XMLCclRequest();
9     myReq.onreadystatechange = reportStatus;
10    myReq.open("GET", "MP_TEST"); //CUST_WH:MP_TEST.PRG
11    myReq.send("^MINE^", value($USR_Personid$), value($PAT_Personid$), value($VIS_Encntrid$));
12 </script>
```



Definitions

XMLCclRequest (MPages 2.0 Implementation)

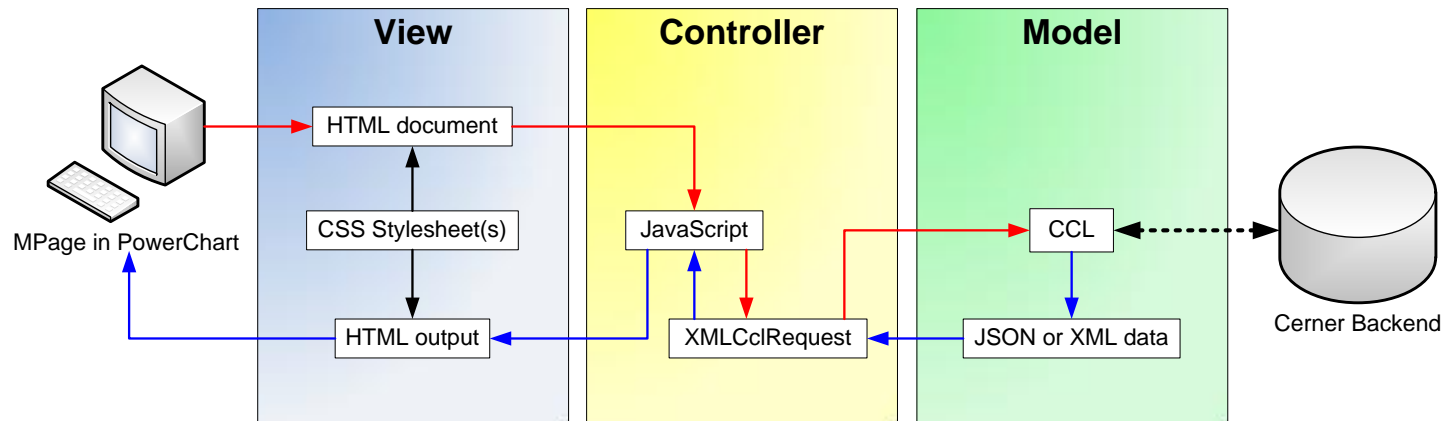
- Cerner integrated the code
 - Natively implemented within Cerner – no external script required
 - Supports synchronous as well as asynchronous processing
 - Allows passing of string parameters greater than 2000 characters



The CCLS Architecture

The Service Oriented Model in an MPage

- HTML document loaded into PowerChart
- Rendering engine applies desired style from CSS referenced in document
- XMLCclRequest is called from JavaScript, executing a CCL program
- The CCL queries the database, returning data as XML or JSON
- JavaScript processes returned data to create new or update content
- HTML output rendered to screen in PowerChart via JavaScript



THE CCLS APPROACH TO BUILDING AN MPAGE



Recommended Expertise

Cerner Architects

- Clinical knowledge
- Mastery of the various Cerner applications (e.g. PrefMaint)

CCL Engineers

- Advanced knowledge of Cerner's DB structure
- Real world CCL experience
- Real world database experience
 - Query optimization
 - Expertise in indexing
 - Understanding of complex of JOINS



Recommended Expertise

Web Engineers

- HTML
 - Mastery of HTML/XHTML
- CSS
 - Proficiency in advanced use of CSS Selectors
- JavaScript
 - Real world AJAX experience
 - Advanced Object Oriented JavaScript
 - Familiarity with JavaScript libraries such as jQuery and DWR
- Server-Side Scripting Languages
 - Knowledge of languages such as PHP, ASP, or Ruby and experience with SQL a plus



The CCLS Build Process

1. Create the “Service” (CCL)
2. Build the Page (HTML)
3. Create the Client (JavaScript)
4. Display the Information (JavaScript)
5. Make it Pretty (CSS)



Creating the “Service”

Write a Basic CCL Report

Structured like any standard CCL Report

- Outputs JSON, XML, CSV, etc. as needed
 - The standard in use at Packard is JSON
 - Packard has developed a CCL Include script known as **MP_JSON.INC** that simplifies the creation of JSON in CCL



Creating the “Service”

The screenshot displays the Discern Visual Developer interface with a COBOL program named `cust_wh.LP_PS_WEIGHT.prg`. The program code is shown in the main editor, and the left pane shows the project structure. Five green callout boxes with arrows point to specific parts of the code, illustrating the steps to create a service:

- Include MP_JSON.INC**: Points to the `*INCLUDE CUST_WH.MP_JSON.INC` line at the top of the program.
- Start the JSON**: Points to the `CALL addToJson('weight':{})` line, which initiates the JSON output.
- Query the database**: Points to the `SELECT INTO` block that retrieves data from the `CLINICAL_EVENT` table.
- Fill the JSON**: Points to the `CALL addToJson` calls within the `DETAIL` loop, which populate the JSON object with data from the database query.
- End the JSON**: Points to the `CALL addToJson('')` line at the end of the program, which closes the JSON output.

```
*INCLUDE CUST_WH.MP_JSON.INC

SET cvINERROR = GET_CODE_VALUE(8,"INERROR","")
SET cvDELETED = GET_CODE_VALUE(48,"DELETED","")
SET cvDOSECALCULATIONWEIGHTKG = GET_CODE_VALUE(72,"","DOSECALCULATIONWEIGHTKG")
SET cvWEIGHTKG = GET_CODE_VALUE(72,"","WEIGHTKG")
CALL addToJson('weight':{})

SELECT INTO "NL:"
    EVENT_END_DT      = FORMAT(CE.EVENT_END_DT_TM,"MM/DD/YYYY HH:MM:SS"),
    RESULT            = CNVTREAL(CE.RESULT_VAL)
FROM CLINICAL_EVENT CE
PLAN CE
WHERE CE.PERSON_ID = nPERSON_ID
AND CE.EVENT_CD IN (cvWEIGHTKG, cvDOSECALCULATIONWEIGHTKG)
AND CE.RESULT_STATUS_CD != cvINERROR
AND CE.RECORD_STATUS_CD != cvDELETED
AND CE.VALID_UNTIL_DT_TM > CNVTDATETIME(CURDATE, CURTIME3)
ORDER CE.EVENT_END_DT_TM
HEAD REPORT
nWEIGHT = 0.0
cWEIGHT_DATE = FILLSTRING(16," ")
nPREVWEIGHT = 0.0
cPREVWEIGHT_DATE = FILLSTRING(16," ")
nDOSECALCWEIGHT = 0.0
cDOSECALC_DATE = FILLSTRING(16," ")
nFIRST = 1
SUBROUTINE FORMAT_WEIGHT(nVALUE, nCOMPARE, cDATE, cLABEL)
cRET = FILLSTRING(20," ")
IF (nCOMPARE > 5.0)
    cRET = CONCAT(TRIM(FORMAT(nVALUE,"#####.#####:T(1)",3)), " kg")
ELSE
    cRET = CONCAT(TRIM(FORMAT(nVALUE * 1000.0, "#####.#####:T(1)",3)), " g")
ENDIF
IF (TRIM(cDATE) != "")
    CALL addToJson(CONCAT(' ', TRIM(cLABEL), ' ':{}))
    CALL addStrVar("value", TRIM(cRET))
    CALL addDateVar("date", TRIM(cDATE))
    CALL addToJson(" ")
ENDIF
END
DETAIL
IF (CE.EVENT_CD = cvWEIGHTKG AND RESULT > 0.0)
    nPREVWEIGHT = nWEIGHT
    cPREVWEIGHT_DATE = cWEIGHT_DATE
    nWEIGHT = RESULT
    cWEIGHT_DATE = EVENT_END_DT
ELSEIF (CE.EVENT_CD = cvDOSECALCULATIONWEIGHTKG AND RESULT > 0.0)
    nDOSECALCWEIGHT = RESULT
    cDOSECALC_DATE = EVENT_END_DT
ENDIF
FOOT REPORT
CALL FORMAT_WEIGHT(nWEIGHT, nWEIGHT, cWEIGHT_DATE, "current")
CALL FORMAT_WEIGHT(nWEIGHT - nPREVWEIGHT, nDOSECALCWEIGHT, cPREVWEIGHT_DATE, "last")
CALL FORMAT_WEIGHT(nDOSECALCWEIGHT, nDOSECALCWEIGHT, cDOSECALC_DATE, "doseCalc")
WITH NOCOUNTER
CALL addToJson('')
END GO
```



Building the Page

Create an HTML Document

This document
will be loaded
into the
desired tab
within
PowerChart

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
4   <head>
5     <title>Care Summary</title>
6     <link rel="stylesheet" type="text/css" href="//\cernerasp\dfs\lucica\MPages\c2857\css\jquery-ui2.css" />
7     <link rel="stylesheet" type="text/css" href="//\cernerasp\dfs\lucica\MPages\c2857\css\cs.css" />
8   </head>
9   <body>
10    <a href="javascript:" id="forget" class="topLink">Reset My Layout</a>
11    <div id="contentShell">
12      <div id="container1" class="portalBox contentThird"></div>
13      <div id="container2" class="portalBox content2Third"></div>
14      <div id="container3" class="portalBox contentFull"></div>
15    </div>
16    <div class='pageLoader'></div>
17  </body>
18 </html>
19 <!--<script type="text/javascript" src="//\cernerasp\dfs\lucica\MPages\c2857\js\firebug.js"></script-->
20 <script type="text/javascript" src="//\cernerasp\dfs\lucica\MPages\c2857\js\mpages\mpages.js"></script>
21 <script type="text/javascript" src="//\cernerasp\dfs\lucica\MPages\c2857\js\mpages\mpages.iconlist.js"></script>
22 <script type="text/javascript" src="//\cernerasp\dfs\lucica\MPages\c2857\js\jquery\jquery.js"></script>
23 <script type="text/javascript" src="//\cernerasp\dfs\lucica\MPages\c2857\js\jquery\jquery.jcal.js"></script>
24 <script type="text/javascript" src="//\cernerasp\dfs\lucica\MPages\c2857\js\jquery\jquery.json.js"></script>
25 <script type="text/javascript" src="//\cernerasp\dfs\lucica\MPages\c2857\js\jquery\jquery.ui.js"></script>
26 <script type="text/javascript" src="//\cernerasp\dfs\lucica\MPages\c2857\js\jquery\jquery.misc.js"></script>
27 <script type="text/javascript" src="//\cernerasp\dfs\lucica\MPages\c2857\js\jquery\jquery.portal.js"></script>
28 <script type="text/javascript" src="//\cernerasp\dfs\lucica\MPages\c2857\js\jquery\jquery.tablesorter.js"></script>
29 <script type="text/javascript" src="//\cernerasp\dfs\lucica\MPages\c2857\js\jquery\jquery.popmenu.js"></script>
30 <script type="text/javascript" src="//\cernerasp\dfs\lucica\MPages\c2857\js\date.f.js"></script>
31 <script type="text/javascript" src="//\cernerasp\dfs\lucica\MPages\c2857\pat\js\webresources-pat-ps.js"></script>
32 <script type="text/javascript" src="//\cernerasp\dfs\lucica\MPages\c2857\pat\js\ps.js"></script>
33 <script type="text/javascript">
34   $MP.webRoot = "\\\\cernerasp\dfs\lucica\MPages\c2857\";
35 </script>
```



Building the Page

Save the HTML Document

Remote hosted clients

- Use Discern Visual Developer to save the document to the Cerner Backend

Self hosted clients

- Also have the option to place document on the server and replicate it to all other servers



Building the Page

Load the Document into PowerChart

Remote hosted clients

- In PrefMaint, enter `CCL_READFILE` into the Report Name field and `"MINE", \\path\to\file.html` into the Report Param field

Self hosted clients

- In PrefMaint, enter the path to the document prefixed with "`<url>file:`" into the Report Name field:
`<url>file:c:\path\to\file.html`



Creating the Client

The JavaScript Interface

Create JavaScript framework to connect HTML document to CCL Service

- To simplify this process, Packard has developed the **mPages JavaScript Framework**, otherwise known as **mPages.js**
- This library provides a toolkit for the client side of MPage development
 - Advanced client-side management of user, patient, and other data
 - Wrapper methods to simplify use of **XMLCclRequest** object
 - Run-time JSON debugging



Creating the Client

Implement the JavaScript Interface

```
$MP.getJSON("LP_PS", function(data) {  
    $MP.cache(data);  
}, {  
    seeJSON: 0  
});
```

- The mPages object referenced by the name “\$MP”
- The *getJSON* method executes the CCL script LP_PS.PRG in the background using XMLCclRequest; the resulting JSON object loaded into the “data” variable of the callback method



Creating the Client

Communicating Both Ways

Data parameters can be passed to the CCL in two ways

1. As a comma separated string using carets in place of quotation marks

```
$MP.getJSON("LP_PS", "^some data^,^more data^", function(data) {  
    $MP.cache(data);  
}), {  
    seeJSON: 0  
});
```

2. As an array in which the carets will be automatically inserted before sending

```
$MP.getJSON("LP_PS", ["some data","more data"], function(data) {  
    $MP.cache(data);  
}), {  
    seeJSON: 0  
});
```



Creating the Client

Processing the Results

```
$(document).ready(function() {  
  
    $MP.getJSON("MP_PAT_SUMMARY", [LPCH.loadList, "true"], function(res) {  
        $MP.cache(res);  
        $.when('$MP.data.usr.lo', function() {  
            $(".pageLoader").remove();  
            $MP.data.pat.encounter_type = ($MP.data.pat.encounter_type == "INPATIENT") ? "Inpatient" : "Outpatient";  
            $(document.body).prepend(  
                $("<h3>").text($MP.data.pat.encounter_type + " Encounter")  
            );  
            $MP.data.pat.eventCal = [1];  
            $MP.data.pat.webResources = [1];  
  
            if ($MP.data.usr.lo["container1"] || $MP.data.usr.lo["container2"] || $MP.data.usr.lo["container3"] || $MP.data.usr.lo["container4"]) {  
                $.each($MP.data.usr.lo, function(cont, cnt) {  
                    $.each(cnt, function(x, id) {  
                        if (!$.isEmpty($MP.data.pat[id])) LPCH.loadModule(id, cont);  
                    });  
                });  
            }  
            $.each(LPCH.layout, function(id, cfg) {  
                if (!$.isEmpty($MP.data.pat[id]) && !$("#" + id).length) LPCH.loadModule(id, cfg.shell);  
            });  
            var tim2 = new Date();  
            $(document.body).prepend($("<div>").text("CCL loaded in: " + ((tim2.getTime() - tim.getTime())*.001) + " seconds"));  
        });  
    }, {  
        seeJSON:0  
    });  
});
```



Display the Information

Translate Data

Develop JavaScript that will interpret the raw JSON data into understandable information

```
build_weight: function(hw) {
  var out = $("

").addClass("content"),
      cont = $("

").appendTo(out),
      found = false;
  if (hw.current) {
    if (hw.current.date == "NaN") hw.current.date = "";
    cont.append($("<div />").append($("<dt><div>Current Weight</div>:</dt>")).append($("<dd />").append(hw.current.value
      + ", last measured (" + hw.current.date.f("E, MM/dd/yyyy HH:mm") + ")"));
    found = true;
  }
  if (hw.last) {
    cont.append($("<div />").append($("<dt><div>Difference between Last Charted Weight and Today's Weight</div>:</dt>")).append($("<dd />").append(hw.last.value
      + ", last weighed (" + hw.last.date.f("E, MM/dd/yyyy HH:mm") + ")"));
    found = true;
  }
  if (hw.doseCalc) {
    cont.append($("<div />").append($("<dt><div>Dose Calc Weight</div>:</dt>")).append($("<dd />").append(hw.doseCalc.value
      + ", last documented (" + hw.doseCalc.date.f("E, MM/dd/yyyy HH:mm") + ")"));
    found = true;
  }
  return out;
},


```



Creating the Client

Display the Raw HTML

Here we see the processed information, but it is hardly presentable

Pending Labs				
General Labs				
Test Name	Details		Collected	
Chem 10	Routine, once		Thu, 08/27/2009 17:11:35	
Hematocrit	Routine, once		Thu, 08/27/2009 17:11:59	
Hemoglobin	Routine, once		Thu, 08/27/2009 17:12:02	
Intake & Output (Calculations based on current dose calc weight)				
Since 0600				
Intake	0 mL		= 0 mL/kg/24hrs	
Output	0 mL		= 0 mL/kg/hr UOP	
Bal	0 mL			
Fluid Management Orders				
Start Date/Time				
: 09/16/09 15:54:00 PDT				
Medication Volumes Included				
: Yes				
PO + IV Fluid Minimum				
: 150				
PO/Enteral Fluid Minimum				
: 100				
Total IV Fluid				
: 300				
Fluid Management Frequency				
: q1hr				
PO + IV Fluid Maximum				
: 250				
PO/Enteral Fluid Maximum				



Make It Pretty

Apply CSS

Use Cascading Style Sheets to style the HTML output to achieve the desired display format

- This is where the mastery of CSS selectors is a must

```
div.portlet {
    margin: 0 3px 7px;
    text-align:left;
    border:1px solid #6666FF;
}

div.portlet h3 {
    font-size:12px;
    margin:4px 0 0;
}

div.portlet a {
    color:blue;
    text-decoration:none;
    cursor:pointer;
}

div.portlet a:hover {
    text-decoration:underline;
}

div.portlet .portlet-header {
    padding-bottom: 4px;
    padding-left: 0.2em;
    font-weight:600;
    font-size:12px;
    background: #6633FF url(images/ui-bg_gloss-wave_75_3204be_500x100.png) 50% 50% repeat-x;
}

div.portlet .portlet-header .ui-icon {
    float: right;
    margin: 1px 0 2px;
    padding: 0;
}

div.portlet .portlet-content {
    padding: 0;
}

div.portlet .portlet-content .content {
    padding: 1px 8px 6px;
    overflow:auto;
}
```



The End Result

150%

Discharge Information

MD/AHP Discharge Goals: abcdefghijklmnopqrstuvwxyz abcdefghijklmnopqrstuvwxyz
 abcdefghijklmnopqrstuvwxyz abcdefghijklmnopqrstuvwxyz
 abcdefghijklmnopqrstuvwxyz abcdefghijklmnopqrstuvwxyz
 abcdefghijklmnopqrstuvwxyz abcdefghijklmnopqrstuvwxyz
 12345678 1234345987 abcdefghijklmon

Target Discharge Date: **Tuesday, September 22, 2009 (Please Update TDD)**
 Target Discharge Time: **8:00-10:00pm**

Web Resources for RED-MG, RUBY

PACS

Bedside Monitor

Demographics

Code Status
 Approving Attending MD: XXXXConversion, Provider
 Resuscitate Orders: Do not resuscitate
 Resuscitate Orders: Do not intubate
 Resuscitate Orders: Do not perform chest compression
 Date order signed: 07/17/09 16:06:00 PDT

Care Level: Critical Care as of Mon, 07/13/2009 16:19
 Service: Adolescent/Pediatrics
 Primary Resident/NP: XXXXConversion, Provider
 Primary Social Worker: XXXTestMD, Don't Use
 Interpreter Required: Yes
 Primary Language: Akan
 Social Work Alert: Yes
 Refer to Social Service Note

Parent Partnership Type
 FCC Amount / Level of: High information
 Information
 FCC Involvement in: Involved in medical procedures/ongoing care when
 Child's Care possible
 FCC Amount of Input: High input

Isolation Status: Airborne (Immunity required) From Date: 08/14/09 08:47

Isolation Order Type
 Airborne N95, Airborne: Tue, 07/14/2009 11:52
 (Immunity Required),
 Solid Organ Transplant
 Date

Isolation Required: Yes- Mask Only Beg. Date: 07/16/09 End Date:

Chicken Pox Exposure
 Had Chicken Pox Disease: Unknown
 Exposed in Past 3 Weeks: Yes
 Exposure Date: Sun, 07/12/2009 00:00

Varicella Vaccine Given: Yes: 1 Dose

Also

Pending Labs

Test Name	Details	Collected
Chem 10	Routine, once	Thu, 08/27/2009 17:11:35
Hematocrit	Routine, once	Thu, 08/27/2009 17:11:59
Hemoglobin	Routine, once	Thu, 08/27/2009 17:12:02

Intake & Output (Calculations based on current dose calc weight)

	Since 0600	
Intake	0 mL	= 0 mL/kg/24hrs
Output	0 mL	= 0 mL/kg/hr UOP
Bal	0 mL	

Fluid Management Orders
 Start Date/Time: 09/16/09 15:54:00 PDT
 Medication Volumes: Yes
 Included
 PO + IV Fluid Minimum: 150
 PO/Enteral Fluid: 100
 Minimum
 Total IV Fluid: 300
 Fluid Management: q1hr
 Frequency
 PO + IV Fluid Maximum: 250
 PO/Enteral Fluid: 200
 Maximum
 Total IV + PO: 300

Major Hospital Events (This Encounter)

Wed, 09/23/2009 Code Blue
 Wed, 09/23/2009 Rapid Response
 Thu, 09/17/2009 Code Blue
 Thu, 09/17/2009 Rapid Response
 Thu, 09/17/2009 ET Tube Unexpected Event - Unplanned extubation



Best Practices

Encapsulation

Data (variables) and operations on that data (methods) should be grouped together in a single entity – an Object

Namespace Control

Create unique variable names and utilize encapsulation so as to prevent variable name overlapping, potentially causing data corruption

- Create variable names that mean something so that the next programmer will understand its purpose
- This applies in JavaScript, CSS, and CCL



Best Practices

Modularization

A technique that increases the extent to which software is composed from separate parts, called modules, each representing a separation of concerns, aiming to improve maintainability by enforcing logical boundaries between components

The ***build_weight*** method seen earlier and the ***build_majorHospEvents***, seen here, are each real production modules used to generate sections of the LPCH Patient Summary

```
build_majorHospEvents: function(mhe) {  
    var content = $("

").addClass("content"),  
        cont = $("

").appendTo(content);  
    $.each(mhe, function(x, his) {  
        cont.append(  
            $("

").text(his.date.f("E, MM/dd/yyyy")), $(": ").text(his.label) ); }); return content; },


```



Best Practices

Debugging

The mPages Framework provides a method, *prettyJSON*, intended to simplify debugging of CCL JSON output

- To use it, simply set the *seeJSON* parameter to a non-empty value:

```
$MP.getJSON("mp_gvrounds", $MP.Rounds.init, {  
  seeJSON: 1  
});
```

- The resulting JSON will be formatted and displayed onscreen as seen to the right

```
{  
  "usr": {  
    "id": "8489060.000000",  
    "user": "JFAULK",  
    "name": "Faulkenberry, Joshua"  
  },  
  "pat": {  
    "id": 8755122.000000,  
    "encounter": 15924595.000000,  
    "encounter_type": "INPATIENT",  
    "arn": 49040579,  
    "fullName": "RED-MG, RUBY",  
    "bday": new Date("07/04/2009 00:00:00"),  
    "gestationalAge": "30",  
    "transplant": false,  
    "loc": {  
      "facility": "LPCH",  
      "location": "A",  
      "nurseUnit": "2E PICU",  
      "room": "2521",  
      "bed": "A"  
    },  
    "labs": [  
      "weight": {  
        "current": {  
          "value": "7.5 kg",  
          "date": new Date("08/28/2009 14:00")  
        },  
        "last": {  
          "value": "-0.1 kg",  
          "date": new Date("07/13/2009 14:50")  
        },  
        "doseCalc": {  
          "value": "10 kg",  
          "date": new Date("09/17/2009 17:02")  
        }  
      }  
    ]  
  }  
}
```



Contact

- jfaulkenberry@lpch.org

Resources

- <http://mpagescommons.org/>
- <http://mpagesjs.org/>
- <http://jquery.com/>
- <http://w3schools.com/>
- <http://lpch.org/>

