

Josh Anderson

(228) 424.0380
janders4@gmail.com

OBJECTIVE: To develop and maintain intuitive and beautiful websites and web applications

EDUCATION: **Tulane University, Tulane College**, New Orleans, LA.
Bachelor of Arts in Computer Science and Philosophy, Architecture minor, May 2002

EXPERIENCE: **TurboSquid** New Orleans, LA
Software Engineer May 2011-December 2012
<http://www.turbosquid.com>

Developed a proprietary MVC framework utilizing ColdFusion and Mustache templates. Expanded REST API. Provisioned site to tie into various 3rd party web-services. Engineered a multi-server synchronized WordPress backend as a consumer friendly CMS.

Creative Virtual London, England*
Web Developer May 2008-May 2011

Designed, configured, and installed the AJAX based html/javascript front-end for Creative Virtual's Virtual Agent knowledge engine for a variety of high profile clients including Tesco, Intuit, Verizon, Virgin Media, and British Gas among others. Utilized subversion to manage the codebase and development of multiple projects, and refactored that codebase to leverage the jQuery framework.

(*while Creative Virtual is based in London, the majority of my work for them was done while telecommuting from New Orleans, LA)

WildBlue Communications Denver, CO
Web Developer January 2007-April 2008
<http://help.wildblue.net>

Developed, expanded, and maintained the corporate knowledgebase. Created a custom knowledgebase application based on a pre-existing framework provided by Talisma, using ASP and MS SQL. Supervised and coordinated the transition of this high volume site into a new hosting environment. Increased traffic by 28%, and ROI by 42% (measuring the cost of development against the decreased cost of customer service).

Precision Camera & Video Austin, TX
Webmaster June 2004-December 2006
<http://www.precision-camera.com>

Maintained, updated, and expanded company website using PHP and MySQL. Developed e-mail marketing campaign and monthly newsletter. Increased revenue by 34% and site traffic by 18% within a year. Developed an interactive cross-platform promotional cd in flash. Coordinated site branding with company branding, and supervised the advertising and internet departments within the company (staff of 4). Negotiated and developed interactive media campaigns for the company with the Austin American Statesman, Google Adwords, & others.

Realizations
Web/Graphic Designer
<http://www.walterandersonart.com>

Ocean Springs, MS
April 2003-April 2004

Developed and maintained Miva-Merchant based company ecommerce site with PHP and MySQL. Developed e-mail marketing campaign and promotional materials for print (with Dreamweaver, Photoshop and Quark).

**COMPUTER
SKILLS:**

Languages and Frameworks:

Highly proficient in css-based-html, JavaScript, the jQuery javascript framework, PHP, the CakePHP framework, Wordpress, MySQL, ASP, and MS SQL. Proficient in ColdFusion, C++, ASP classic, and MSSQL.

Software and Operating Systems:

Comfortable with Windows, Macintosh, and Unix platforms. Highly proficient in Adobe Photoshop, Illustrator, InDesign, Macromedia Flash, Dreamweaver and Microsoft Excel.

REFERENCES:

Will Perkins – Application developer
Creative Virtual
wilperkins@gmail.com
504-453-8483

Joel Carranza – Software Engineer
joel.carranza@gmail.com
504-654-9494

Stacy Wood - Knowledge base development manager
WildBlue Communications
stacy_wood@yahoo.com
303-618-8589

CODING SAMPLES

CSS-based HTML

Below is a bit of css-based html that I used to generate a 'breadcrumb-tree' for a client.

```
<ul id = 'breadcrumbTree'>
  <li>
    <ul>
      <li class = 'activeSibling' style = 'background:none;'>
        <a href = '/kb/category/3003'>Installation</a>
      </li>
    </ul>
  </li>
  <li>
    <ul>
      <li class = 'activeSibling'><a href = '/kb/category/3010'>Getting Installed</a></li>
      <li><a href = '/kb/category/3011'>Equipment</a></li>
      <li class = 'last'><a href = '/kb/category/3012'>Anti-virus / Anti-spam</a></li>
    </ul>
  </li>
  <li>
    <ul>
      <li class = 'currentSibling'><a href = '/kb/category/3030'>Process</a></li>
      <li><a href = '/kb/category/3031'>Site</a></li>
      <li><a href = '/kb/category/3032'>PC</a></li>
      <li><a href = '/kb/category/3033'>Agreement</a></li>
      <li class = 'last'><a href = '/kb/category/3034'>Scheduling</a></li>
    </ul>
  </li>
</ul>
```



CODING SAMPLES

Javascript

Below is an excerpt from a 'survey' object I created that populated survey questions (via AJAX) into a jQuery-UI accordion object (<http://jqueryui.com/demos/accordion/>).

```
this.fetch = function(type){

    //optional
    var parameters = (arguments[1]) ? arguments[1] : {};
    var url = self.url;

    var postData = "";
    if (url == "generic_proxy.php") postData = "url=" +
    encodeURIComponent("https://dvp.ca.com/casurvey/bot.htm?jsEnabled=1") + "&mimeType=text/xml&";
    if ((type != 'select') && (self.ident) && (self.userlogid))        postData += "ident=" +
    self.ident + "&userlogid=" + self.userlogid + "&";

    var callback = false;

    //SAML TOKEN
    var DVPST = _get('DVPST');

    switch (type) {
        case "select":
            postData += "ACTION=INIT";
            if (DVPST) postData += "&DVPST=" + DVPST;
            callback = renderSelect;
            break;

        case "survey":
            postData += "ACTION=SELECTSURVEY";
            if (parameters.surveyId){
                postData += "&SURVEYID=" + parameters.surveyId;

            } else {
                dialog(localize("surveyIdRequired"));
                return false;
            }
            if (parameters.siteId) postData += "&SITEID=" + parameters.siteId;

            callback = renderSurvey;
            break;

        case "next":
            postData += "ACTION=NEXT";
            callback = renderSurvey;
            break;

        case "submit":
            postData += "ACTION=SUBMIT";
            if (parameters.data){
                postData += "&DATA=" + parameters.data;
            }
        }
    }
}
```

```
        } else {
            dialog(localize("dataRequired"));
            return false;
        }
        callback = renderResults;
        break;

    default:
        type = "select";
        break;

} //end switch

loading(false, localize('loading'));
$.ajax({
    type: "POST",
    url: url,
    data: postData,
    dataType: 'XML',
    cache: false,
    success: callback,
    error: callback
});

} //end fetch
```

Step 1: Choose a Questionnaire Step 2: Complete the Questionnaire

CA OPS/MVS Event Management and Automation (EMA)

▶ 1 - Environment: How long have you been using CA OPS/MVS EMA? 6-10 Years

▼ 2 of 29: What release of CA OPS/MVS EMA are you running in production?

- 11.8
- 11.7
- 11.6
- Other

CODING SAMPLES

PHP / MySQL

Below is an excerpt from a CakePHP model class I wrote to play word association games.

```
/**
 * Return an array of playable pongs (and writes that array to the cache)
 *
 * @param int $limit - Max # of pongs to return (should have been previously sanitized)
 * @return array of pongs
 * @access public
 */

function getPlayablePongs($limit = 10){

    Cache::set(array('duration' => '+1 hour'));

    //$pongs = Cache::read('playablePongs');
    //if (!empty($pongs)) return $pongs;

    $expirationDate = date('Y-m-d H:i:s', strtotime("-1 hour"));

    $pongs = $this->query("SELECT DISTINCT Pong.id, Pong.name, Pong.num_players,
Pong.length, Pong.created, User.display_name
    FROM pongs AS Pong
    JOIN users AS User ON User.id = Pong.user_id
    WHERE Pong.type = 'open'
    AND Pong.active = 'y'
    AND Pong.num_players < Pong.max_players
    AND Pong.length > 1
    AND Pong.modified >= " . $expirationDate . "
    ORDER BY Pong.num_players ASC, Pong.modified DESC
    LIMIT " . $limit);

    //close expired games
    $this->query("UPDATE pongs SET active = 'n' WHERE active = 'y' AND modified < " .
$expirationDate . "");

    Cache::set(array('duration' => '+1 hour'));
    Cache::write('playablePongs', $pongs);

    return $pongs;

} //end function
```

```

/**
 * Returns an array of players for the given $pongId, writes that array to the cache, and purges players
 that have left
 *
 * @param int $pongId - valid game id
 * @param string $type - Defaults to 'current', accepts 'all' or 'current'. If 'all' is selected, only 'current'
 players are cached.
 * @return array of players
 * @access public
 */

function getPlayers($pongId, $type = 'current', $removeOldPlayers = true){

    if (empty($pongId)) return false;
    $pong = Cache::read('pong_' . $pongId);
    $players = Cache::read('pongPlayers_' . $pongId);

    if (empty($players)){
        $players = $this->query("SELECT DISTINCT User.id, User.display_name, User.type,
User.avatar, PongsUser.id, PongsUser.order, PongsUser.score, PongsUser.type, PongsUser.created,
PongsUser.modified
                                FROM pongs AS Pong
                                JOIN pongs_users AS PongsUser ON PongsUser.pong_id = Pong.id
                                JOIN users AS User ON User.id = PongsUser.user_id
                                WHERE Pong.id = " . $pongId . "
                                ORDER BY PongsUser.created ASC, PongsUser.id ASC");
    }

    $currentPlayers = array();

    //kick out any players that haven't answered in a while
    $expirationInterval = 480; //in seconds
    $playerCount = count($players);

    foreach($players as $key => $player){

        if ($players[$key]['PongsUser']['type'] != 'former'){

            if ($removeOldPlayers){//if we haven't removed the old players yet, look for
them (and remove the first)
                $seconds = time() - strtotime($player['PongsUser']['modified']);

                if (($seconds > ($expirationInterval * ($playerCount -
1)))&&($player['User']['id'] != 1)){
                    if (isset($pong['Pong']['paused'])){
                        if ($pong['Pong']['paused'] != 'y'){
                            $players[$key]['PongsUser']['type'] = 'former';
                            $this->leaveGame($pongId,
                                $player['User']['id']);
                            // $removeOldPlayers = false;
                        }
                    }
                }
            }
        }
    }
}

```

```
        if ($players[$key]['PongsUser']['type'] != 'former'){
            $currentPlayers[] = $players[$key];
        }//end if ! former
    }//end foreach

    Cache::write('pongPlayers_' . $pongId, $currentPlayers);

    if ($type == 'all')
        return $players;
    else
        return $currentPlayers;
} //end function
```


CODING SAMPLES

ASP / MS SQL

Below is a recursive function that I wrote to traverse a category tree.

'returns array of subcategories within category given by 'id', returns top level categories if none
'will recursively return the subcategories of the subcategories, going 'levels' deep

```
function get_category_descendants(cat_id, levels, arrCategories)
```

```
dim sqlstr, portalId
```

```
portalId = Cint(session("defaultPortal"))
```

```
if len(session("CPid")) > 0 then portalId = session("CPid")
```

```
if len(cat_id) > 0 then 'CATEGORIES that are children of given category in a given portal
```

```
    sqlstr = "SELECT DISTINCT c.cat_id, c.cat_name, c.cat_order, cs.csu_parentCatID FROM  
tb_knbCategory AS c, tb_knbCategoryGroup AS cg, tb_knbCPortalArticleAccess AS paa,  
tb_knbCategorySub AS cs WHERE c.cat_status = 1 AND c.Cat_kbID = 6 AND c.cat_id = cs.csu_childCatID  
AND cs.csu_parentCatID = '' & cat_id & '' AND c.cat_id = cg.CGr_categoryID AND paa.CPC_portalID = ''  
& portalId & '' AND paa.CPC_groupID = cg.CGr_groupID AND c.cat_name != 'Archive' ORDER BY  
c.cat_order, c.cat_name"
```

```
    if not isArray(arrCategories) then 'if first execution
```

```
        if not category_access(cat_id, true) then 'if portal does not have access to category and  
every category parent then exit
```

```
            get_category_descendants = ""
```

```
            exit function
```

```
        end if
```

```
    end if
```

```
else 'GET TOP LEVEL CATEGORIES
```

```
    sqlstr = "SELECT DISTINCT c.cat_id, c.cat_name, c.cat_order FROM tb_knbCategory AS c,  
tb_knbCategoryGroup AS cg, tb_knbCPortalArticleAccess AS paa WHERE c.cat_status = 1 AND  
c.Cat_kbID = 6 AND c.cat_id NOT IN (select csu_childCatID from tb_knbCategorySub) AND c.cat_id =  
cg.CGr_categoryID AND paa.CPC_portalID = '' & portalId & '' AND paa.CPC_groupID = cg.CGr_groupID  
AND c.cat_name != 'Archive' ORDER BY c.cat_order, c.cat_name"
```

```
end if
```

```
dim dbconn, rs, rs_rows, rowMax, rowCounter, colCounter, colMax, catCounter, catMax, levelsCopy,  
categories
```

```
const keyCatId = 0
```

```
const keyCatName = 1
```

```
Set dbconn = Server.CreateObject("ADODB.Connection")
```

```
dbconn.Open ecrmKB
```

```
set rs = dbconn.Execute(sqlstr)
```

```
if not rs.eof then
```

```
    rs_rows = rs.GetRows()
```

```
else
```

```
    rs_rows = ""
```

```
end if
```

```
rs.close
```

```
dbconn.close
```

```
set rs = nothing
set dbconn = nothing
```

```
if isArray(rs_rows) and Cint(levels) > 0 then
```

```
    levels = levels - 1
    rowMax = UBound(rs_rows, 2)
    colMax = UBound(rs_rows, 1)
```

```
    if IsArray(arrCategories) then
        catMax = ubound(arrCategories, 2)
        catCounter = catMax + 1
        redim preserve arrCategories(3, rowMax + catCounter)
```

```
    else
        catCounter = 0
        redim arrCategories(3, rowMax)
```

```
    end if
```

```
        For rowCounter = 0 To rowMax
            for colCounter = 0 to colMax
                'if inStr(rs_rows(keyCatName, rowCounter), ">") < 1 then
arrCategories(colCounter, rowCounter + catCounter) = rs_rows(colCounter, rowCounter)
arrCategories(colCounter, rowCounter + catCounter) =
rs_rows(colCounter, rowCounter)
                next
            Next
```

```
        For rowCounter = 0 To rowMax
            levelsCopy = levels
            arrCategories = get_category_descendants(rs_rows(0, rowCounter),
levels, arrCategories)
            levels = levelsCopy
        Next
```

```
    else
        get_category_descendants = ""
    end if 'has kids and levels > 0
```

```
get_category_descendants = arrCategories
end function
```