```php
<?php require_once "app/includes/initialize.php"; ?>

<?php get_header(); ?>

<center>
    <img src="app/theme/img/logo.png" alt="iQualit" />
</center>

<div id="content">

<?php

//PAGINA PRINCIPAL SI EL USUARIO HA INICIADO SESIÓN

//Se muestra esta página, si ha iniciado sesión.
if(user_validates()) { ?>

<ul class="pageitem">
    <li class="menu">
        <a href="app/main.php">
            <img alt="Ir a tu cuenta" src="app/theme/icons/application_home.png" />
            <span class="name">El meu compte</span>
            <span class="arrowbig"></span>
        </a>
    </li>

    <li class="menu">
        <a href="app/logout.php">
            <img alt="Cerrar sesion" src="app/theme/icons/bigcancel.png" />
            <span class="name">Sortir</span>
            <span class="arrow"></span>
        </a>
    </li>
</ul>

<?php } else {

//PAGINA PRINCIPAL SI EL USUSARIO NO HA INICIADO SESIÓN ?>

<span class="graytitle">Què és iQualit?</span>

<p class="graytext">iQualit es una eïna dedicada al control i gestió de qualitat a peu d'obra,
actualitzada segons el CTE i la EHE-08.</p>

<ul class="pageitem">
    <li class="menu">
        <a href="app/login.php">
            <img alt="Iniciar sesion" src="app/theme/icons/user.png" />
            <span class="name">Inicia sessió</span>
            <span class="arrowbig"></span>
        </a>
    </li>

    <li class="menu">
        <a href="app/register.php">
            <img alt="Description" src="app/theme/icons/application_form_edit.png" />
            <span class="name">Registre d'usuari</span>
            <span class="arrow"></span>
        </a>
    </li>
</ul>

<?php }

// "Sobre iQualit" aparece siempre ?>

<ul class="pageitem">
    <li class="menu">
        <a href="app/about.php">
            <span class="name">Sobre iQualit</span>
            <span class="arrow"></span>
        </a>
```

```
    </li>
</ul>

</div>

<?php get_footer(); ?>
```

```php
<?php

include('app/includes/mobile_device_detect.php');
mobile_device_detect(true,true,true,true,true,true,'index_mobile.php',false);

//Si el usuario intenta entrar a la dirección index_mobile desde un dispositivo no
//permitido, le sale un mensaje de error
$detected = $_GET['r'];
if ($detected == "error") {
    $error_de_acceso = 1;
}

?>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<html>
<head>

    <title>iQualit - Home Page</title>

    <link href="/pfc/home/css/main.css" rel="stylesheet" type="text/css" />

    <script type="text/javascript" src="http://use.typekit.com/qse7wlt.js"></script>
    <script type="text/javascript">try{Typekit.load();}catch(e){}</script>

</head>
<body>

<div id="header">
    <div id="top_bar">
        <ul>
            <li><a href="index.php">Inici</a></li>
            <li><a href="index.php#sobre">Sobre iQualit</a></li>
            <li><a href="index.php#contacte">Contacte</a></li>
        </ul>
    </div>
</div>

<div id="main">

<?php if ($error_de_acceso) { ?>

    <div id="error">
        <p>Error. Ho sento, només pots accedir a aquesta direcció amb un dispositiu compatible</p>
    </div>

<?php } ?>

<h1>iQualit</h1>

<ul>
<li><a href="index.php#que_es">Què és?</a></li>
<li><a href="index.php#com_funciona">Còm funciona?</a></li>
<li><a href="index.php#compatibilitat">Compatibilitat</a></li>
<li><a href="index.php#participants">Participants</a></li>
</ul>

<div class="clear"></div>

<div id="featured">
<h3 class="tk-legendum">Control i gestio de la qualitat<br/>
segons CTE i EHE-08</h3>
</div>

<!--
<div id="featured">
    <div id="video">
        <div id="object">
            <object width="245" height="367"><param name="allowfullscreen" value="true" />
            <param name="allowscriptaccess" value="always" /><param name="movie" value="http://
vimeo.com/moogaloop.swf?
clip_id=8783738&amp;server=vimeo.com&amp;show_title=1&amp;show_byline=1&amp;show_portrait=0&amp;color=&amp;
```

```
>
            <embed src="http://vimeo.com/moogaloop.swf?
clip_id=8783738&amp;server=vimeo.com&amp;show_title=1&amp;show_byline=1&amp;show_portrait=0&amp;color=&amp;
type="application/x-shockwave-flash" allowfullscreen="true" allowscriptaccess="always" width="245"
height="367"></embed>
            </object>
        </div>
    </div>

<div style="clear:both"></div>

</div>
-->

<div id="info">
<div class="column-info">
<h2>Què és?</h2>
<p>Approaches content management with the underlying goals of simplicity and openness, so you can
build anything.</p>
</div>

<div class="column-info">
<h2>Què és?</h2>
<p>Approaches content management with the underlying goals of simplicity and openness, so you can
build anything.</p>
</div>

<div class="column-info">
<h2>Què és?</h2>
<p>Approaches content management with the underlying goals of simplicity and openness, so you can
build anything.</p>
</div>

<div class="column-info">
<h2>Què és?</h2>
<p>Approaches content management with the underlying goals of simplicity and openness, so you can
build anything.</p>
</div>
</div>

</div>

<div id="footer">
<div id="footer_content">

    <div id="footer_info">

    </div>

    <div id="contact">
        <h3>Contacte</h3>
        <span>nom</span>
        <input type="text" name="email" />
        <span>email</span>
        <input type="text" name="email" />
        <span>missatge</span>
        <textarea name="missatge" rows="4"></textarea>
        <input type="button" value="Enviar" />
    </div>

    <div class="clear"></div>

</div>
</div>

</body>
</html>
```

```php
<?php

require_once "includes/initialize.php";

//EL USUARIO ESTÁ DENTRO DE SU SESIÓN Y ACCEDE A ESTA PAGINA

$user_id = $_GET['id'];

if(user_validates($user_id)) { ?>

<?php get_header(); ?>

<?php top_bar('Editar Usuari', 'main.php'); ?>

<div id="content">

<?php

//ACTUALIZA LOS DATOS DE LA BASE DE DATOS SI SE HA PULSADO SUBMIT

if(isset($_POST['submit'])){

    $nombre = $db->escape($_POST['nombre']);
    $apellidos = $db->escape($_POST['apellidos']);
    $direccion = $db->escape($_POST['direccion']);
    $ciudad = $db->escape($_POST['ciudad']);
    $colegiado = $db->escape($_POST['colegiado']);
    $dni = $db->escape($_POST['dni']);

    $query = "UPDATE users SET nombre = '".$nombre."', apellidos = '".$apellidos."', direccion = '".
$direccion."', ciudad = '".$ciudad."', colegiado = '".$colegiado."', dni = '".$dni."' WHERE user_id =
'".$user_id."' ";

    $update = $db->query($query);

    if(mysql_affected_rows() == 1){ ?>

    <p class="graytext"><img src="theme/icons/accept.png" /> Els canvis han estat desats amb èxit. <a
href="main.php">Tornar</a></p>

    <?php } else { ?>

    <p class="graytext"><img src="theme/icons/cancel.png" /> Les teves dades no han pogut ser
registrades a la base de dades. <a href="main.php">Tornar</a></p>

    <?php } }


//MUESTRA LOS DATOS DE LA BASE DE DATOS

$user_query = $db->query("SELECT * FROM users WHERE user_id = '".$user_id."' ");

$user = $db->fetch($user_query); ?>

<span class="graytitle"><?php echo $user['email']; ?></span>

<ul class="pageitem">
    <li class="menu">
        <a href="user_change_pass.php?id=<?php echo urlencode($user_id); ?>">
        <img alt="Description" src="theme/icons/key.png" />
        <span class="name">Canviar password</span>
        <span class="arrow"></span>
        </a>
    </li>
</ul>

<span class="graytitle">Dades personals</span>

<form method="post" action="user_edit.php?id=<?php echo urlencode($user_id); ?>" name="submit"
id="submit">
    <ul class="pageitem">
        <li class="form">
```

```
            <span class="narrow">
            <span class="name">Nom</span>
            <input type="text" name="nombre" value="<?php echo $user['nombre']; ?>" />
            </span>
        </li>

        <li class="form">
            <span class="narrow">
            <span class="name">Cognom</span>
            <input type="text" name="apellidos" value="<?php echo $user['apellidos']; ?>" />
            </span>
        </li>

        <li class="form">
            <span class="narrow">
            <span class="name">Direcció</span>
            <input type="text" name="direccion" value="<?php echo $user['direccion']; ?>" />
            </span>
        </li>

        <li class="form">
            <span class="narrow">
            <span class="name">Ciutat</span>
            <input type="text" name="ciudad" value="<?php echo $user['ciudad']; ?>" />
            </span>
        </li>

        <li class="form">
            <span class="narrow">
            <span class="name">Nº de coleg.</span>
            <input type="text" name="colegiado" value="<?php echo $user['colegiado']; ?>" />
            </span>
        </li>

        <li class="form">
            <span class="narrow">
            <span class="name">DNI</span>
            <input type="text" name="dni" value="<?php echo $user['dni']; ?>" />
            </span>
        </li>
    </ul>

    <p class="graytext">Les teves dades seràn enregistrades a una base de dades privada i no seràn mai
publicades.</p>

    <ul class="pageitem">
        <li class="form">
            <input type="submit" name="submit" value="Actulitzar perfil" />
        </li>
    </ul>
</form>

</div>

<?php get_footer(); ?>

<?php
//ALGUIEN ACCEDE A ESTA DIRECCIÓN DIRECTAMENTE

} else { user_not_allowed(); } ?>
```

```php
<?php

require_once "includes/initialize.php";

//EL USUARIO ESTÁ DENTRO DE SU SESIÓN Y ACCEDE A ESTA PAGINA

$user_id = $_GET['id'];

if(user_validates($user_id)) { ?>

<?php get_header(); ?>

<?php top_bar('Canviar pass.', "user_edit.php?id={$user_id}"); ?>

<div id="content">

<?php

$user_query = $db->query("SELECT * FROM users WHERE user_id = '".$user_id."' ");
$user = $db->fetch($user_query);

$pass_antiguo = $user['password'];

$new_pass = md5(md5($_POST['new_pass']));

if(isset($_POST['submit'])){

//1 Mira que la contraseña antigua esté bien

if (md5(md5($_POST['last_pass'])) == $pass_antiguo) {

//2 Mira que la nueva contraseña y confirmar sean iguales

    if ($_POST['new_pass'] == $_POST['confirm_new_pass']) {

    //UPDATE //3 Cambia la antigua contraseña por la nueva
        $query = "UPDATE users SET password = '".$new_pass."' WHERE user_id = '".$user_id."' ";
        $update = $db->query($query);

        if(mysql_affected_rows() == 1){ ?>

    <p class="graytext"><img src="theme/thumbs/accept.png" /> Els canvis han estat gravats amb èxit.
<a href="index.php">Tornar</a></p>

    <?php } else {
        //Fail
    }

    } else { ?>

    <p class="graytext"><img src="theme/thumbs/cancel.png" /> El nou password no coincideix.
Assegura't d'escriure correcte 2 vegades.<a href="?id=<?php echo urlencode($user_id); ?>">Tornar</a></
p>

    <?php } } else { ?>

    <p class="graytext"><img src="theme/thumbs/cancel.png" /> L'antic password no és correcte.<a
href="?id=<?php echo urlencode($user_id); ?>">Volver</a></p>

<?php } } ?>

<form method="post" action="user_change_pass.php?id=<?php echo urlencode($user_id); ?>" name="submit"
id="submit">

    <ul class="pageitem">
        <li class="form">
            <span class="narrow">
            <span class="name">Antic pass.</span>
            <input type="password" name="last_pass" />
            </span>
        </li>
    </ul>
```

```
    <ul class="pageitem">
        <li class="form">
            <span class="narrow">
            <span class="name">Nou pass.</span>
            <input type="password" name="new_pass" />
            </span>
        </li>

        <li class="form">
            <span class="narrow">
            <span class="name">Confirmar</span>
            <input type="password" name="confirm_new_pass" />
            </span>
        </li>
    </ul>

    <ul class="pageitem">
        <li class="form">
            <input type="submit" name="submit" value="Canviar password" />
        </li>
    </ul>

</form>

</div>

<?php get_footer();

//ALGUIEN ACCEDE A ESTA DIRECCIÓN DIRECTAMENTE

} else { user_not_allowed(); } ?>
```

```
var iWebkit;

if (!iWebkit) {

    iWebkit = window.onload = function () {

        iWebkit.checkboxHeight = "25";
        iWebkit.radioHeight = "25";
        iWebkit.autolistNumVisible = 10;

        function url() {
            var a = document.getElementsByTagName("a");
            for (var i = 0; i < a.length;i++) {
                if (a[i].className.match("noeffect")) {
                }
                else {
                    a[i].onclick = function () {
                        window.location = this.getAttribute("href");
                        return false;
                    };
                }
            }
        }

        function hideURLbar() {
            window.scrollTo(0, 0.9);
        }

        iWebkit.popup = function () {
            window.scrollTo(0, 9999);
            var o_popup = document.getElementById(arguments[0]);
            var o_frame = o_popup.getElementsByClassName('confirm_screen')[0];
            o_frame.className = 'confirm_screenopen';
            var b = document.getElementById("cover");
            b.className = "cover";
            b.style.height = document.height + "px";
        };

        iWebkit.closepopup = function () {
            var o_parent = arguments[0].toElement;
            var b_found = false;
            do {
                o_parent = o_parent.parentNode;
                if (o_parent.tagName.toLowerCase() == 'div' && o_parent.className.toLowerCase() ==
'popup') {
                    b_found = true;
                    break;
                }
            } while (o_parent.parentNode);

            if (b_found === false) {
                return false;
            }

            var o_frameclose = o_parent.getElementsByClassName('confirm_screenopen')[0];
            o_frameclose.className = 'confirm_screenclose';
            var b = document.getElementById("cover");
            b.className = "nocover";
            b.style.height = 0;
        };

        function initAutoLists() {

            var ul = document.getElementsByTagName('ul');
            for (var i = 0; i < ul.length; i++) {
                var list = ul[i];
                if (list.getAttribute('class').search(/(autolist)/) === -1) {
                    continue;
                }
                var items = list.getElementsByTagName('li');
                if (items.length <= iWebkit.autolistNumVisible) {
                    continue;
```

```
            }
            list.numitems = items.length;
            list.visibleitems = 0;
            var button = list.getElementsByClassName('autolisttext')[0];
            button.onclick = function (event) {
                var list = this.parentNode;
                list.showItems(list.visibleitems + iWebkit.autolistNumVisible);
                return false;
            };
            list.showItems = function (numItems) {
                var items = this.getElementsByTagName('li');
                var count = 0;
                for (var i = 0; i < items.length; i++) {
                    items[i].className = items[i].className.replace(/hidden/g, '');
                    if (i >= numItems) {
                        items[i].className = items[i].className + ' hidden';
                    }
                    else {
                        count += 1;
                    }
                }
                this.visibleitems = count;
                button.className = button.className.replace(/hidden/g, '');
                if (count >= (items.length - 1)) {
                    button.className = button.className + ' hidden';
                }
            };
            list.showItems(iWebkit.autolistNumVisible);
        }
    }

    iWebkit.init = function () {
        url();
        hideURLbar();
        initAutoLists();
        var inputs = document.getElementsByTagName("input"), span = [], textnode, option, active;
        for (var a = 0;a < inputs.length;a++) {
            if (inputs[a].type === "checkbox" || inputs[a].type === "radio") {
                span[a] = document.createElement("span");
                span[a].className = inputs[a].type;
                if (inputs[a].checked) {
                    if (inputs[a].type === "checkbox") {
                        var position = "0 -" + (iWebkit.checkboxHeight * 2) + "px";
                        span[a].style.backgroundPosition = position;
                    } else {
                        position = "0 -" + (iWebkit.radioHeight * 2) + "px";
                        span[a].style.backgroundPosition = position;
                    }
                }
                inputs[a].parentNode.insertBefore(span[a], inputs[a]);
                inputs[a].onchange = iWebkit.clear;
                span[a].onmouseup = iWebkit.check;
                document.onmouseup = iWebkit.clear;
            }
        }
        inputs = document.getElementsByTagName("select");
        for (a = 0;a < inputs.length; a++) {
            if (inputs[a]) {
                option = inputs[a].getElementsByTagName("option");
                active = option[0].childNodes[0].nodeValue;
                textnode = document.createTextNode(active);
                for (var b = 0;b < option.length;b++) {
                    if (option[b].selected) {
                        textnode = document.createTextNode(option[b].childNodes[0].nodeValue);
                    }
                }
                span[a] = document.createElement("span");
                span[a].className = "select";
                span[a].id = "select" + inputs[a].name;
                span[a].appendChild(textnode);
                inputs[a].parentNode.insertBefore(span[a], inputs[a]);
                inputs[a].onchange = iWebkit.choose;
```

```
                }
            }
        };

        iWebkit.pushed = function () {
            var element = this.nextSibling;
            if (element.checked && element.type === "checkbox") {
                this.style.backgroundPosition = "0 -" + iWebkit.checkboxHeight * 3 + "px";
            } else {
                if (element.checked && element.type === "radio") {
                    this.style.backgroundPosition = "0 -" + iWebkit.radioHeight * 3 + "px";
                } else {
                    if (!element.checked && element.type === "checkbox") {
                        this.style.backgroundPosition = "0 -" + iWebkit.checkboxHeight + "px";
                    } else {
                        this.style.backgroundPosition = "0 -" + iWebkit.radioHeight + "px";
                    }
                }
            }
        };

        iWebkit.check = function () {
            var element = this.nextSibling;
            if (element.checked && element.type === "checkbox") {
                this.style.backgroundPosition = "0 0";
                element.checked = false;
            } else {
                if (element.type === "checkbox") {
                    this.style.backgroundPosition = "0 -" + iWebkit.checkboxHeight * 2 + "px";
                } else {
                    this.style.backgroundPosition = "0 -" + iWebkit.radioHeight * 2 + "px";
                    var group = this.nextSibling.name;
                    var inputs = document.getElementsByTagName("input");
                    for (var a = 0;a < inputs.length;a++) {
                        if (inputs[a].name === group && inputs[a] !== this.nextSibling) {
                            inputs[a].previousSibling.style.backgroundPosition = "0 0";
                        }
                    }
                }
                element.checked = true;
            }
        };

        iWebkit.clear = function () {
            var inputs = document.getElementsByTagName("input");
            for (var b = 0;b < inputs.length;b++) {
                if (inputs[b].type === "checkbox" && inputs[b].checked) {
                    inputs[b].previousSibling.style.backgroundPosition = "0 -" +
iWebkit.checkboxHeight * 2 + "px";
                } else {
                    if (inputs[b].type === "checkbox") {
                        inputs[b].previousSibling.style.backgroundPosition = "0 0";
                    }
                    else {
                        if (inputs[b].type === "radio" && inputs[b].checked) {
                            inputs[b].previousSibling.style.backgroundPosition = "0 -" +
iWebkit.radioHeight * 2 + "px";
                        } else {
                            if (inputs[b].type === "radio") {
                                inputs[b].previousSibling.style.backgroundPosition = "0 0";
                            }
                        }
                    }
                }
            }
        };

        iWebkit.choose = function () {
            var option = this.getElementsByTagName("option");
            for (var d = 0;d < option.length; d++) {
                if (option[d].selected) {
                    document.getElementById("select" + this.name).childNodes[0].nodeValue = option
```

```
[d].childNodes[0].nodeValue;
                }
            }
        };

        iWebkit.init();
    };
}

/* OCULTA Y MUESTRA UN DIV */

function togglecomments (postid) {
    var whichpost = document.getElementById(postid);
    if (whichpost.className=="shown") { whichpost.className="hidden"; } else
{ whichpost.className="shown"; }
}
```

```
<!DOCTYPE html>
<html>

<head>

<meta http-equiv="Content-Type" content="text/html" charset="iso-8859-1" />

<title>iQualit</title>

<meta content="yes" name="apple-mobile-web-app-capable" />

<meta content="index,follow" name="robots" />

<link rel="apple-touch-startup-image" href="http://ferranfg.com.es/pfc/theme_mobile/img/startup.png">
<link rel="apple-touch-icon" href="/pfc/theme/img/icon.png" />

<meta content="minimum-scale=1.0, width=device-width, maximum-scale=0.6667, user-scalable=no"
name="viewport" />

<link href="<?php echo HOME; ?>/app/theme/css.gz.php" rel="stylesheet" type="text/css" />

<script type="text/javascript" src="<?php echo HOME; ?>/app/theme/js/js.gz.php"></script>

</head>
```

```
</body>
</html>
```

```
body {
    position: relative;
    margin: 0;
    -webkit-text-size-adjust: none;
    min-height: 416px;
    font-family: helvetica,sans-serif;
    background: url("img/background.png") repeat;
    -webkit-touch-callout: none;
    -webkit-transition-duration: 300ms;
    -webkit-transition-property: -webkit-transform;
}
.center {
    margin: auto;
    display: block;
}
img {
    border: 0;
}
a:hover span.arrow, a:hover {
    background-position: 0 -13px!important;
}

#topbar {
    position: relative;
    left: 0;
    top: 0;
    height: 44px;
    width: auto;
    background: url("img/header.png") repeat;
}

#title {
    position: absolute;
    top: 0;
    left: 0;
    right: 0;
    padding: 0 10px;
    text-align: center;
    text-overflow: ellipsis;
    white-space: nowrap;
    overflow: hidden;
    height: 44px;
    line-height: 44px;
    font-weight: bold;
    color: #fff;
    text-shadow: rgba(0,0,0,0.6) 0 -1px 0;
    font-size: 16pt;
}
#content {
    width: 100%;
    position: relative;
    min-height: 250px;
    padding-top: 13px;
    height: auto;
    z-index: 0;
    overflow: hidden;
}

#scroll { overflow-y:scroll;height:390px; }

#footer {
    text-align: center;
    position: relative;
    margin: 20px 10px 0;
    height: auto;
    width: auto;
    bottom: 10px;
}
#footer a, #footer {
    text-decoration: none;
    font-size: 9pt;
    color: #4C4C4C;
```

```
    text-shadow: #FFF 0 1px 0;
}
.pageitem {
    -webkit-border-radius: 8px;
    background-color: #fff;
    border: #878787 solid 1px;
    font-size: 12pt;
    overflow: hidden;
    padding: 0;
    position: relative;
    display: block;
    height: auto;
    width: auto;
    margin: 3px 9px 17px;
    list-style: none;
}
.textbox {
    padding: 5px 9px;
    position: relative;
    overflow: hidden;
    border-top: 1px solid #878787;
}
.textbox p {
    margin-top: 2px;
    color: #000;
    margin-bottom: 2px;
    text-align: justify;
}
.textbox img {
    max-width: 100%;
}
.textbox ul {
    margin: 3px 0 3px 0;
    list-style: circle!important;
}
.textbox li {
    margin: 0!important;
}
.pageitem li:first-child {
    border-top: 0;
}
li.menu, li.form {
    position: relative;
    list-style-type: none;
    display: block;
    height: 43px;
    overflow: hidden;
    border-top: 1px solid #878787;
    width: auto;
}
ul.pageitem li:first-child:hover, .pageitem li:first-child a, li.form:first-child input[type=radio],
li.form:first-child select, li.form:first-child input[type=submit], li.form:first-child button,
li.form:first-child input[type=reset] {
    -webkit-border-top-left-radius: 8px 8px;
    -webkit-border-top-right-radius: 8px 8px;
}
ul.pageitem li:last-child:hover, .pageitem li:last-child a, li.form:last-child input[type=radio],
li.form:last-child select, li.form:last-child input[type=submit], li.form:last-child button,
li.form:last-child input[type=reset] {
    -webkit-border-bottom-left-radius: 8px 8px;
    -webkit-border-bottom-right-radius: 8px 8px;
}
li.menu:hover {
    background:#0274ee;
}
li.no-hover:hover {
    background:#fff !important;
}
li.menu a:hover span.name, li.menu a:hover span.comment, li.store:hover .starcomment,
li.store:hover .name, li.store:hover .comment, body.list li.withimage a:hover .comment {
    color: #fff;
}
```

```
li.menu a:hover span.comment {
    color: #CCF;
}
li.menu a {
    display: block;
    height: 43px;
    width: auto;
    text-decoration: none;
}
li.menu a img {
    width: auto;
    height: 32px;
    margin: 5px 0 0 5px;
    float: left;
}
li.menu span.name {
    margin: 11px 0 0 7px;
    width: auto;
    color: #000;
    font-weight: bold;
    font-size: 17px;
    text-overflow: ellipsis;
    overflow: hidden;
    max-width: 70%;
    white-space: nowrap;
    float: left;
}

li.menu span.comment {
    margin: 11px 30px 0 0;
    width: auto;
    color: #000;
    font-size: 17px;
    text-overflow: ellipsis;
    overflow: hidden;
    max-width: 75%;
    white-space: nowrap;
    float: right;
    color: #324f85;
}

li.menu span.arrow, li.store span.arrow, body.musiclist span.arrow, body.list span.arrow {
    position: absolute;
    width: 8px!important;
    height: 13px!important;
    right: 10px;
    top: 15px;
    margin: 0!important;
    background: url("img/arrow.png") 0 0 no-repeat;
}

.arrowbig {
    position: absolute;
    width: 28px!important;
    height: 28px!important;
    right: 10px;
    top: 8px;
    margin: 0!important;
    background: url("img/arrowbig.png") 0 0 no-repeat;
}

.arrowdown {
    position: absolute;
    width: 28px!important;
    height: 28px!important;
    right: 10px;
    top: 17px;
    margin: 0!important;
    background: url("img/arrowdown.png") 0 0 no-repeat;
}

.graytitle {
```

```
    position: relative;
    font-weight: bold;
    font-size: 17px;
    right: 20px;
    left: 9px;
    color: #4C4C4C;
    text-shadow: #FFF 0 1px 0;
    padding: 1px 0 3px 8px;
}

.graytext {
    position: relative;
    font-size: 13px;
    right: 20px;
    left: 9px;
    color: #4C4C4C;
    padding: 1px 20px 3px 8px;
}


.header {
    display: block;
    font-weight: bold;
    color: rgb(73,102,145);
    font-size: 12pt;
    margin-bottom: 6px;
    line-height: 14pt;
}
body.musiclist div#content {
    width: auto;
    margin: -29px auto auto -40px;
}
body.musiclist div#content ul {
    background: url("img/musiclistbg.png") repeat;
}
body.musiclist div#content ul li {
    list-style: none;
    height: 44px;
    width: auto;
    border-bottom: 1px solid #e6e6e6;
    position: relative;
}
body.musiclist div#content ul li a {
    text-decoration: none;
    color: #000;
    width: 100%!important;
    height: 100%;
    display: block;
}
body.musiclist ul li .number, body.musiclist .name, body.musiclist .time {
    display: inline-block;
    height: 44px;
    font-weight: bold;
    font-size: large;
    width: 44px;
    text-align: center;
    line-height: 46px;
}
body.musiclist ul li .name {
    margin-left: 0;
    width: auto!important;
    font-size: medium;
    padding-left: 5px;
    border-left: solid 1px #e6e6e6;
}
body.musiclist ul li .time {
    color: #848484;
    font-size: medium;
    margin-left: 4px;
    width: auto!important;
    font-weight: normal;
}
```

```
body.musiclist {
    background-image: none!important;
    background-color: #cbcccf;
}
body.musiclist ul li span.name {
    text-overflow: ellipsis;
    overflow: hidden;
    white-space: nowrap;
    max-width: 62%;
}
body.list ul li.title {
    background: url("img/listheader.gif") repeat-x;
    height: 22px!important;
    width: 100%;
    color: #fff;
    font-weight: bold;
    font-size: 16px;
    text-shadow: gray 0 1px 0;
    line-height: 22px;
    padding-left: 20px;
    border-bottom: none!important;
}
body.list ul {
    background-color: #fff;
    width: 100%;
    overflow: hidden;
    padding: 0;
    margin: 0;
}
body.list div#content li {
    height: 40px;
    border-bottom: 1px solid #e1e1e1;
    list-style: none;
}

body.list {
    background-color: #fff;
    background-image: none!important;
}
body.list div#footer {
    margin-top: 24px!important;
}
body.list div#content li a, body.list div#content li p {
    margin: 0;
    padding: 9px 0 0 20px;
    font-size: large;
    font-weight: bold;
    position: relative;
    display: block;
    color: #000;
    text-decoration: none;
    height: 32px;
}
body.list div#content li a span.name {
    text-overflow: ellipsis;
    overflow: hidden;
    max-width: 93%;
    white-space: nowrap;
    display: block;
}
body.list div#content li a:hover {
    color: #fff;
}
body.list div#content li a:hover {
    background-color:#0163e8;
}
body.list div#content {
    margin-top: -13px!important;
}

body.list li.withimage {
    height: 90px!important;
```

```
}
body.list li.withimage span.name {
    margin: 13px 0 0 90px;
    text-overflow: ellipsis;
    overflow: hidden;
    max-width: 63%!important;
    white-space: nowrap;
}
body.list li.withimage .comment {
    margin: 10px auto auto 90px !important;
    max-width: 63%!important;
}
body.list li.withimage a, body.list li.withimage:hover a {
    height: 81px!important;
    background: url("img/storebg.png") left top no-repeat!important;
}
body.list li.withimage:hover {
    background-image: url("img/storetouched.png");
}
.confirm_screen {
    position: absolute;
    bottom: 0;
    -webkit-transform: translate(0,100%);
    -webkit-transition-property: -webkit-transform;
    background-image: url("img/popup-bg.png");
    opacity: 0;
}
.confirm_screenopen {
    position: absolute;
    opacity: 0.8;
    overflow: hidden;
    bottom: -100%;
    width: 100%;
    background: url("img/popup-bg.png") repeat-x #4e545f;
    -webkit-transition-duration: 0.8s;
    -webkit-transition-property: -webkit-transform;
    -webkit-transform-style: preserve-3d;
    -webkit-transform: translate(0,0);
    text-align: center;
    z-index: 99999;
}
.confirm_screenopenfull {
    position: absolute;
    opacity: 0.8;
    overflow: hidden;
    bottom: -100%;
    width: 100%;
    background: url("img/popup-bg.png") repeat-x #4e545f;
    -webkit-transition-duration: 0.8s;
    -webkit-transition-property: -webkit-transform;
    -webkit-transform-style: preserve-3d;
    -webkit-transform: translate(0,45px);
    text-align: center;
    z-index: 99999;
}
.confirm_screenclose {
    -webkit-transition-duration: 1.2s;
    -webkit-transition-property: -webkit-transform;
    -webkit-transform-style: preserve-3d;
    position: absolute;
    opacity: 0.8;
    overflow: hidden;
    bottom: -100%;
    width: 100%;
    background: url("img/popup-bg.png") repeat-x #545A67;
    text-align: center;
    -webkit-transform: translate(0,150%);
}
.confirm_screenopen span, .confirm_screenclose span, .confirm_screenopenfull
span, .confirm_screenclosefull span {
    margin: 10px 0 20px;
    font-size: 17px;
```

```
    color: #fff;
    width: 100%;
    height: 10px;
    text-shadow: rgba(0,0,0,1) 0 -1px 0;
    display: block;
}
.popup {
    position: absolute;
    bottom: 0;
    width: 100%;
    left: 0;
    z-index: 9999;
}
.cover {
    width: 100%;
    position: absolute;
    top: 0;
    z-index: 9998;
    opacity: 0.4;
    left: 0;
    background-color: #000;
}
.nocover {
    opacity: 0;
    background-image: url('img/bluebutton.png'),url('img/darkredbutton.png');
}
#frame a {
    text-decoration: none;
    display: block;
    width: 90%;
    margin-left: auto;
    margin-right: auto;
    margin-bottom: -15px;
    margin-top: 0;
}
#frame span.black, #frame span.red, #frame span.gray {
    display: block;
    height: 46px;
    border-width: 0 14px;
    width: auto;
    background-repeat: no-repeat;
    line-height: 46px;
    font-size: large;
    opacity: 1;
    font-weight: bolder;
    white-space: nowrap;
    text-overflow: ellipsis;
    overflow: hidden;
    font-family: Arial,Helvetica,sans-serif;
}
#frame span.black {
    -webkit-border-image: url("img/graybutton.png") 0 14 0 14;
    color: #fff;
}
#frame span.red {
    -webkit-border-image: url("img/redbutton.png") 0 14 0 14;
    color: #fff;
}
#frame span.gray {
    -webkit-border-image: url("img/whitebutton.png") 0 14 0 14;
    color: #000;
    text-shadow: white 0 1px 0;
}
#frame a:last-child {
    margin-bottom: 20px!important;
}
#frame a:hover span.black, #frame a:hover span.gray {
    -webkit-border-image: url("img/bluebutton.png") 0 14 0 14;
    color: #fff;
    text-shadow: rgba(0,0,0,1) 0 -1px 0;
}
#frame a:hover span.red {
```

```
    -webkit-border-image: url("img/darkredbutton.png") 0 14 0 14;
}
#leftnav, #leftbutton {
    position: absolute;
    font-size: 12px;
    left: 9px;
    top: 7px;
    font-weight: bold;
}
#leftnav, #leftbutton, #rightnav, #rightbutton {
    z-index: 5000;
}
#leftnav a, #rightnav a, #leftbutton a, #rightbutton a {
    display: block;
    color: #fff;
    text-shadow: rgba(0,0,0,0.6) 0 -1px 0;
    line-height: 30px;
    height: 30px;
    text-decoration: none;
}
#leftnav img, #rightnav img {
    margin-top: 4px;
}
#leftnav a:first-child {
    z-index: 2;
    -webkit-border-image: url("img/navleft.png") 0 5 0 13;
    border-width: 0 5px 0 13px;
    -webkit-border-top-left-radius: 16px;
    -webkit-border-bottom-left-radius: 16px;
    -webkit-border-top-right-radius: 6px;
    -webkit-border-bottom-right-radius: 6px;
    width: auto;
}
#leftnav a {
    -webkit-border-image: url("img/navlinkleft.png") 0 5 0 13;
    z-index: 3;
    margin-left: -4px;
    border-width: 0 5px 0 13px;
    padding-right: 4px;
    -webkit-border-top-left-radius: 16px;
    -webkit-border-bottom-left-radius: 16px;
    -webkit-border-top-right-radius: 6px;
    -webkit-border-bottom-right-radius: 6px;
    float: left;
}
#rightnav, #rightbutton {
    position: absolute;
    font-size: 12px;
    right: 9px;
    top: 7px;
    font-weight: bold;
}
#rightnav a {
    -webkit-border-image: url("img/navlinkright.png") 0 13 0 5;
    z-index: 3;
    margin-right: -4px;
    border-width: 0 13px 0 5px;
    padding-left: 4px;
    -webkit-border-top-left-radius: 6px;
    -webkit-border-bottom-left-radius: 6px;
    float: right;
    -webkit-border-top-right-radius: 16px;
    -webkit-border-bottom-right-radius: 16px;
}
#rightnav a:first-child {
    z-index: 2;
    -webkit-border-top-left-radius: 6px;
    -webkit-border-bottom-left-radius: 6px;
    -webkit-border-image: url("img/navright.png") 0 13 0 5;
    border-width: 0 13px 0 5px;
    -webkit-border-top-right-radius: 16px;
    -webkit-border-bottom-right-radius: 16px;
```

```
}
#leftbutton a, #rightbutton a {
    -webkit-border-image: url("img/navbutton.png") 0 5 0 5;
    border-width: 0 5px;
    -webkit-border-radius: 6px;
}
li.form input[type=text], li.form input[type=password], li.form input[type=search], li.form input
[type=email] {
    border-width: 7px 7px;
    font-weight: normal;
    border-color: white;
    height: 10px;
    -webkit-border-image: url("img/textfield.png") 7 7 7 7;
    -webkit-appearance: none;
    line-height: 8px;
    font-size: 18px;
    margin: auto;
    display: block;
    position: relative;
    width: 90%;
}
li.form input[type=submit] {
    width: 100%;
    background: none;
    border: 0px;
    color: #000;
    margin-top: -5px;
    margin-bottom: -5px;
    font-weight: bold;
    font-size: 17px;
}
.form {
    padding: 5px 8px 0 5px;
    height: 37px!important;
    position: relative;
    overflow: hidden;
}
li.form .narrow textarea, li.form .narrow input[type=text], li.form .narrow input[type=checkbox],
li.form .narrow input[type=password], li.form .narrow input[type=search], li.form .narrow input
[type=email] {
    width: 60%!important;
    /*border-width: 7px 7px;
    border-color: white;*/
    border:none;
    padding:7px;
    height: 10px;
    -webkit-border-image: url("img/textfield.png") 7 7 7 7;
    -webkit-appearance: none;
    line-height: 8px;
    font-size: 17px;
    font-weight: normal;
    margin: 0!important;
    position: absolute;
    right: -17px;
}
li.form span.narrow, li.form span.check {
    width: 90%!important;
    display: block;
    position: relative;
    margin: auto;
}
li.form .name {
    width: 55%!important;
    white-space: nowrap;
    text-overflow: ellipsis;
    position: absolute;
    margin: 6px 0 0 7px;
    color: #000;
    font-weight: bold;
    font-size: 17px;
    overflow: hidden;
    left: -17px;
```

```
}
li.form .check .name {
    width: 70%!important;
}
li.form input[type=radio] {
    width: 100%;
    height: 42px;
    display: block;
    margin: -5px -8px 0 -5px;
    -webkit-appearance: none;
    border: 0;
    -webkit-border-radius: 0;
    position: relative;
    background: transparent;
    position: absolute;
}
span.radio {
    width: 16px;
    height: 30px;
    background: url(img/check.png) no-repeat;
    display: block;
    position: absolute;
    right: 5px;
    overflow: visible;
    z-index: 1;
}
span.checkbox {
    width: 94px;
    height: 27px;
    display: block;
    position: absolute;
    right: -14px;
    top: 2px;
}

select {
    height: 40px;
    opacity: 0;
    position: absolute;
    width: 100%;
    margin: -5px 0 0 -5px;
    -webkit-border-radius: 0;
}
.form .choice .name {
    left: 1.7%;
    width: 87%!important;
}
span.select {
    z-index: 1;
    position: absolute;
    white-space: nowrap;
    text-overflow: ellipsis;
    margin: 6px 0 0 7px;
    color: #000;
    font-weight: bold;
    font-size: 17px;
    overflow: hidden;
    max-width: 87%;
}
.form .arrow {
    background: url('img/more.png') no-repeat;
    width: 13px;
    height: 8px;
    position: absolute;
    right: 8px;
    top: 18px;
    margin: 0!important;
}
input[type=submit], button, input[type=button], input[type=reset] {
    background: transparent;
    width: 100%;
    height: 40px;
```

```
    left: 0px;
    position: absolute;
    top: 6px;
    display: block;
    -webkit-border-radius: 0;
    line-height: 40px;
}
button, input[type=button], input[type=reset] {
    top: 0px!important;
    border: none;
    color: black;
    font-weight: bold;
    font-size: 17px;
    -webkit-appearance: none;
}
.textbox textarea {
    min-height: 50px;
    margin: 3px auto 4px auto;
    position: relative;
    -webkit-border-image: url("img/textfield.png") 7 7 7 7;
    padding: 3px 0 0 0;
    line-height: 18px;
    left: -2px;
    font-size: 18px;
    font-weight: normal;


    width: 97%;
    display: block;
    border-width: 7px 7px;
}
ul li.hidden {
    display: none;
}
ul li.autolisttext {
    text-align: center;
}
body.musiclist ul li.autolisttext {
    line-height: 44px!important;
}
ul li.autolisttext a:hover {
    background-image: none!important;
    color: black!important;
}

.info {
    background-color:#eee;
    border-top:1px solid #ddd;
    padding-left:10px;
    font-size:13px;
}

/* OCULTAR / MOSTRAR EN EXECUCIO */

.hidden {display:none;}

.fet {background-color:#c8ffcd; }
.nofet {background-color:#ffc7c7; }

.p-hidden {
    font-weight:500 !important;
    color:#202020 !important;
    font-size:1em !important;
    padding:9px 20px !important;
    margin:0;
    border-bottom:1px solid #e1e1e1;
}

.p-hidden a { color:#202020 !important; text-decoration:none; }

/* RECEPCIO */
```

```
.obligatori {
    font-weight:500 !important;
    color:#353f5c !important;
    background:url('img/obligatori.png') repeat #dfe7ff;
    font-size:1em !important;
    padding:7px 20px 7px 45px !important;
    margin:0;
    border-bottom:1px solid #e1e1e1;
}

.voluntari {
    font-weight:500 !important;
    color:#5a552c !important;
    background:url('img/voluntari.png') repeat #fffbd9;
    font-size:1em !important;
    padding:7px 20px 7px 45px !important;
    margin:0;
    border-bottom:1px solid #e1e1e1;
}

.establert_ehe {
    background-color: #ececec;
    font-size:0.9em !important;
    text-align:justify;
}
```

```php
<?php

//include('includes/mobile_device_detect.php');
//mobile_device_detect(true,true,true,true,true,true,false,'http://localhost/pfc/index.php?red=0');

require_once "includes/initialize.php"; ?>

<?php get_header(); ?>

<?php top_bar('Nou Usuari', '../index_mobile.php', 'H'); ?>

<div id="content">

<?php

//EL USUARIO INTRODUCE VALORES DENTRO DEL FORMULARIO Y:

if(!empty($_POST['email']) && !empty($_POST['password'])) {

    $email = $db->escape($_POST['email']);
    $password = md5(md5($db->escape($_POST['password'])));
    $confirm_password = $_POST['confirm_password'];

    $checkusername = $db->query("SELECT email FROM users WHERE email = '".$email."'");

    //Comprueba que sea una dirección de email válida

    if(valid_email($email)) {

        error_message('La direcció de email no es válida. Introdueix una direcció amb el format: exemple@exemple.com', 'register.php');

    }

    //Comprueba antes que la dirección de email no esté ya registrada

    elseif(mysql_num_rows($checkusername) == 1) {

        error_message('La direcció de email ja està siguent utilitzada. Intenta-ho amb una altre diferent', 'register.php');

    }

    //Comprueba que la contraseña está escrita 2 veces iguales

    elseif($confirm_password != $_POST['password']) {

        error_message('Les contrasenyes no son iguals. Assegura\'t d\'escriure el mateix valor dues vegades', 'register.php');

    } else {

    //Si pasa todos los comprobantes anteriores,
    //LOS VALORES DEL FORMULARIO SON INTRODUCIDOS DENTRO DE LA TABLA 'users'

    $registerquery = $db->query("INSERT INTO users (email, password) VALUES('".$email."', '".$password."')");

    //Si se han introducido bien los datos

        if($registerquery) { ?>

            <span class="graytitle">Dades acceptades!</span>
            <p class="graytext">La informació introduida ha estat registrada correctamente a la base de dades. Ara, pots entrar en el teu compte fent clic <a href="login.php">aquí</a></p>

        <?php //Ha habido algun problema durante la introducción de los datos

        } else {

            error_message('Un error inesperat ha fet imposible la introducció de la teva informació a la base de dades. Inenta-ho un altre cop', 'register.php');
```

```
        }

    }

} else {

//FORMULARIO DE REGISTRO

?>

<span class="graytitle">Registre d'usuari</span>

<form method="post" action="register.php" name="registerform" id="registerform">
    <ul class="pageitem">
        <li class="form">
            <span class="narrow">
            <span class="name">Email</span>
                <input type="text" name="email" />
            </span>
        </li>

        <li class="form">
            <span class="narrow">
            <span class="name">Password</span>
                <input type="password" name="password" />
            </span>
        </li>

        <li class="form">
            <span class="narrow">
            <span class="name">Repetir pass.</span>
                <input type="password" name="confirm_password" />
            </span>
        </li>
    </ul>

    <ul class="pageitem">
        <li class="form">
            <input type="submit" name="register" id="register" value="Registrar" />
        </li>
    </ul>
</form>

<?php } ?>

</div>

<?php get_footer(); ?>
```

```php
<?php

require "includes/initialize.php";

$user_id = $_GET['id'];
$project_id = $_GET['project_id'];

//EL USUARIO ESTÁ DENTRO DE SU SESIÓN Y ACCEDE A ESTA PAGINA

if(user_validates($user_id, $project_id)) { ?>

<?php get_header(); ?>

<?php top_bar('Projecte', 'main.php'); ?>

<div id="content">

<?php //MUESTRA LOS DATOS DE LA BASE DE DATOS, SEGUN ID DEL PROJECTO

$project_query = $db->query("SELECT projecte FROM project WHERE id = '".$project_id."' ");
$project = $db->fetch($project_query); ?>

<span class="graytitle">
    <?php short_title($project['projecte']); ?>
</span>

<ul class="pageitem">
    <li class="menu">
        <a href="project_control.php?id=<?php enc($user_id); ?>&project_id=<?php enc($project_id); ?
>&control_de=recepcio">
            <span class="name">Control de Recepció</span>
            <span class="arrowbig"></span>
        </a>
    </li>

    <li class="menu">
        <a href="project_control.php?id=<?php enc($user_id); ?>&project_id=<?php enc($project_id); ?
>&control_de=execucio">
            <span class="name">Control d'Execució</span>
            <span class="arrowbig"></span>
        </a>
    </li>

    <li class="menu">
        <a href="project_control.php?id=<?php enc($user_id); ?>&project_id=<?php enc($project_id); ?
>&control_de=acabada">
            <span class="name">Control d'Unitat Acabada</span>
            <span class="arrowbig"></span>
        </a>
    </li>
</ul>

<span class="graytitle">Opcions</span>

<ul class="pageitem">
    <li class="menu">
        <a href="project_edit.php?id=<?php echo urlencode($user_id); ?>&project_id=<?php echo urlencode
($project_id); ?>">
            <img alt="Editar Projecte" src="theme/icons/table_edit.png" />
            <span class="name">Editar Projecte</span>
            <span class="arrow"></span>
        </a>
    </li>

    <li class="menu">
        <a href="project_print.php?id=<?php echo urlencode($user_id); ?>&project_id=<?php echo
urlencode($project_id); ?>">
            <img alt="Editar Projecte" src="theme/icons/printer.png" />
            <span class="name">Imprimir Documents</span>
            <span class="arrow"></span>
        </a>
    </li>
```

```
</ul>

<?php //ELIMINA EL PROYECTO Y TE ENVIA A LA PÁGINA PRINCIPAL
if($_POST['delete']) {

    $db->query("DELETE FROM project WHERE id = '".$project_id."' ");
    $db->query("DELETE FROM recepcio WHERE project_id = '".$project_id."' ");
    $db->query("DELETE FROM execucio WHERE project_id = '".$project_id."' ");
    $db->query("DELETE FROM acabada WHERE project_id = '".$project_id."' "); ?>
    <meta http-equiv="refresh" content="0;main.php">

<?php } ?>

<form method="post" action="project_view.php?id=<?php echo urlencode($user_id); ?>&project_id=<?php
echo urlencode($project_id); ?>" name="project_edit" id="project_edit">

    <ul class="pageitem">
        <li class="form">
            <img alt="Editar Projecte" src="theme/icons/delete.png" /><input type="submit"
name="delete" id="project_edit" value="Eliminar projecte" />
        </li>
    </ul>

</form>

</div>

<?php get_footer(); ?>

<?php } else { user_not_allowed(); } ?>
```

```php
<?php

require "includes/initialize.php";

$user_id = $_GET['id'];
$project_id = $_GET['project_id'];

if(user_validates($user_id, $project_id)) { ?>

<?php get_header(); ?>

<?php top_bar('Imprimir', "project_view.php?id={$user_id}&project_id={$project_id}"); ?>

<div id="content">

<!--<span class="graytitle">Programa del Control de Qualitat</span>

<ul class="pageitem">
    <li class="menu">
        <a href="project_print_pdf.php?id=<?php echo urlencode($user_id); ?>&project_id=<?php echo
urlencode($project_id); ?>">
            <span class="name">Català</span>
            <span class="arrow"></span>
        </a>
    </li>
</ul>-->

<span class="graytitle">Control en obra</span>

<ul class="pageitem">
    <li class="menu">
        <a href="processos/print/processos_view.php?id=<?php echo urlencode($user_id); ?>&project_id=<?
php echo urlencode($project_id); ?>&control_de=recepcio" class="noeffect" onclick="return alert('Avis:
Quan facis una impressió, una copia en format PDF serà enviada automaticament al teu email.');">
            <span class="name">Control de Recepció</span>
            <span class="arrow"></span>
        </a>
    </li>
    <li class="menu">
        <a href="processos/print/processos_view.php?id=<?php echo urlencode($user_id); ?>&project_id=<?
php echo urlencode($project_id); ?>&control_de=execucio" class="noeffect" onclick="return alert('Avis:
Quan facis una impressió, una copia en format PDF serà enviada automaticament al teu email.');">
            <span class="name">Control d'Execució</span>
            <span class="arrow"></span>
        </a>
    </li>
    <li class="menu">
        <a href="processos/print/processos_view.php?id=<?php echo urlencode($user_id); ?>&project_id=<?
php echo urlencode($project_id); ?>&control_de=acabada" class="noeffect" onclick="return alert('Avis:
Quan facis una impressió, una copia en format PDF serà enviada automaticament al teu email.');">
            <span class="name">Control d'Unitat Acabada</span>
            <span class="arrow"></span>
        </a>
    </li>

</ul>


</div>

<?php get_footer(); ?>

<?php } else { user_not_allowed(); } ?>
```

```php
<?php

require "includes/initialize.php";

$user_id = $_GET['id'];
$project_id = $_GET['project_id'];

if(user_validates($user_id, $project_id)) { ?>

<?php get_header(); ?>

<?php top_bar('Editar Projecte', "project_view.php?id={$user_id}&project_id={$project_id}"); ?>

<div id="content">

<?php

if(isset($_POST['submit'])) {

$projecte = $db->escape($_POST['projecte']);
$ref = $db->escape($_POST['ref']);
$localitzacio = $db->escape($_POST['localitzacio']);
$projectista = $db->escape($_POST['projectista']);
$promotor = $db->escape($_POST['promotor']);
$autor = $db->escape($_POST['autor']);
$data = $db->escape($_POST['data']);

$query = "UPDATE project SET projecte = '".$projecte."', ref = '".$ref."', localitzacio = '".
$localitzacio."', projectista = '".$projectista."', promotor = '".$promotor."', autor = '".$autor."',
data = '".$data."' WHERE id = '".$project_id."' ";

$update = $db->query($query);

if(mysql_affected_rows() == 1){ ?>
    <p class="graytext"><img src="theme/thumbs/accept.png" /> Els canvis han estat desats amb èxit. <a
href="project_view.php?id=<?php echo urlencode($user_id); ?>&project_id=<?php echo urlencode
($project_id); ?>">Tornar</a></p>
    <?php } else { ?>
    <p class="graytext"><img src="theme/thumbs/cancel.png" /> Les teves dades no han pogut ser
registrades a la base de dades. <a href="project_edit.php?id=<?php echo urlencode($user_id); ?
>&project_id=<?php echo urlencode($project_id); ?>">Torna a intentar-ho</a></p>
    <?php } }

//MUESTRA LOS DATOS DE LA BASE DE DATOS, SEGUN ID DEL PROJECTO */

$project_query = $db->query("SELECT * FROM project WHERE id = '".$project_id."' ") or die (mysql_error
() );

$project = $db->fetch($project_query); ?>

<form method="post" action="project_edit.php?id=<?php echo urlencode($user_id); ?>&project_id=<?php
echo urlencode($project['id']); ?>" name="project_edit" id="project_edit">

<ul class="pageitem">
    <li class="form">
        <span class="narrow">
        <span class="name">Projecte</span>
            <input type="text" name="projecte" value="<?php echo $project['projecte']; ?>" />
        </span>
    </li>

    <li class="form">
        <span class="narrow">
        <span class="name">Ref</span>
            <input type="text" name="ref" value="<?php echo $project['ref']; ?>" />
        </span>
    </li>

    <li class="form">
        <span class="narrow">
        <span class="name">Localització</span>
            <input type="text" name="localitzacio" value="<?php echo $project['localitzacio']; ?>" />
```

```
        </span>
    </li>

    <li class="form">
        <span class="narrow">
        <span class="name">Projectista</span>
            <input type="text" name="projectista" value="<?php echo $project['projectista']; ?>" />
        </span>
    </li>

    <li class="form">
        <span class="narrow">
        <span class="name">Promotor</span>
            <input type="text" name="promotor" value="<?php echo $project['promotor']; ?>" />
        </span>
    </li>

    <li class="form">
        <span class="narrow">
        <span class="name">Autor Prog.</span>
            <input type="text" name="autor" value="<?php echo $project['autor']; ?>" />
        </span>
    </li>

    <li class="form">
        <span class="narrow">
        <span class="name">Data</span>
            <input type="text" name="data" value="<?php echo $project['data']; ?>" />
        </span>
    </li>
</ul>

<ul class="pageitem">
    <li class="form">
        <input type="submit" name="submit" id="project_edit" value="Actualitzar Dades" />
    </li>
</ul>

</form>

</div>

<?php get_footer(); ?>

<?php } else { user_not_allowed(); } ?>
```

```php
<?php

require_once "includes/initialize.php";

//EL USUARIO ESTÁ DENTRO DE SU SESIÓN Y ACCEDE A ESTA PAGINA

$user_id = $_GET['id'];

if(user_validates($user_id)) { ?>

<?php get_header(); ?>

<?php top_bar('Nou Projecte', 'main.php'); ?>

<div id="content">

<?php

//EL USUARIO INTRODUCE VALORES DENTRO DEL FORMULARIO Y:

if(!empty($_POST['projecte'])) {

    $projecte = $db->escape($_POST['projecte']);
    $ref = $db->escape($_POST['ref']);
    $localitzacio = $db->escape($_POST['localitzacio']);
    $projectista = $db->escape($_POST['projectista']);
    $promotor = $db->escape($_POST['promotor']);
    $autor = $db->escape($_POST['autor']);
    $data = $db->escape($_POST['data']);

    $registerquery = $db->query("INSERT INTO project (user_id, projecte, ref, localitzacio,
projectista, promotor, autor, data) VALUES( '".$user_id."', '".$projecte."', '".$ref."', '".
$localitzacio."', '".$projectista."', '".$promotor."', '".$autor."', '".$data."')");

    //Si se han introducido bien los datos

    if($registerquery) { ?>

    <span class="graytitle">Dades acceptades!</span>
    <p class="graytext">La informació introduida ha estat registrada correctamente a la base de dades.
<a href="main.php">Tornar</a></p>


    <?php //Ha habido algun problema durante la introducción de los datos

    } else {

    error_message('Un error inesperat ha fet imposible la introducció de la teva informació a la base
de dades', 'project_create.php');

    }

} else {

//FORMULARIO DE REGISTRO

?>

<span class="graytitle">Crear un nou projecte</span>

<form method="post" action="project_create.php?id=<?php echo urlencode($user_id); ?>"
name="project_create" id="project_create">

    <ul class="pageitem">

        <li class="form">
            <span class="narrow">
            <span class="name">Projecte</span>
                <input type="text" name="projecte" />
            </span>
        </li>
```

```
        <li class="form">
            <span class="narrow">
            <span class="name">Ref</span>
                <input type="text" name="ref" />
            </span>
        </li>

        <li class="form">
            <span class="narrow">
            <span class="name">Localització</span>
                <input type="text" name="localitzacio" />
            </span>
        </li>

        <li class="form">
            <span class="narrow">
            <span class="name">Projectista</span>
                <input type="text" name="projectista" />
            </span>
        </li>

        <li class="form">
            <span class="narrow">
            <span class="name">Promotor</span>
                <input type="text" name="promotor" />
            </span>
        </li>

        <li class="form">
            <span class="narrow">
            <span class="name">Autor Prog.</span>
                <input type="text" name="autor" />
            </span>
        </li>

        <li class="form">
            <span class="narrow">
            <span class="name">Data</span>
                <input type="text" name="data" />
            </span>
        </li>

    </ul>

    <ul class="pageitem">
        <li class="form">
            <input type="submit" name="project_new" id="project_new" value="Crear un nou Projecte" />
        </li>
    </ul>

</form>

</div>

<?php } ?>

<?php get_footer(); ?>

<?php } else { user_not_allowed(); } ?>
```

```php
<?php require "includes/initialize.php";

$user_id = $_GET['id'];
$project_id = $_GET['project_id'];
$control_de = $_GET['control_de'];

if(user_validates($user_id, $project_id)) { ?>

<?php get_header('list'); ?>

<?php

switch ($control_de) {
    case 'recepcio';
        $title = 'Cont. Recepció';
    break;
    case 'execucio';
        $title = "Cont. Execució";
    break;
    case 'acabada';
        $title = "Obra finalitzada";
    break;
}

top_bar($title, "project_view.php?id={$user_id}&project_id={$project_id}"); ?>

<div id="content">

<?php

//CUANDO SE DA A GUARDAR DADES

if (isset($_POST['submit'])) {

    //01 Moviments de terres
    if (isset($_POST['PQ_0111'])) { $insert = '.PQ_0111'; }
    if (isset($_POST['PQ_0121'])) { $insert .= '.PQ_0121'; }
    if (isset($_POST['PQ_0122'])) { $insert .= '.PQ_0122'; }
    if (isset($_POST['PQ_0127'])) { $insert .= '.PQ_0127'; }
    if (isset($_POST['PQ_0131'])) { $insert .= '.PQ_0131'; }
    if (isset($_POST['PQ_0141'])) { $insert .= '.PQ_0141'; }
    if (isset($_POST['PQ_0161'])) { $insert .= '.PQ_0161'; }
    if (isset($_POST['PQ_0162'])) { $insert .= '.PQ_0162'; }
    if (isset($_POST['PQ_0171'])) { $insert .= '.PQ_0171'; }
    if (isset($_POST['PQ_0181'])) { $insert .= '.PQ_0181'; }
    if (isset($_POST['PQ_0191'])) { $insert .= '.PQ_0191'; }

    //02 Fonaments
    if (isset($_POST['PQ_021A'])) { $insert .= '.PQ_021A'; }
    if (isset($_POST['PQ_0212'])) { $insert .= '.PQ_0212'; }
    if (isset($_POST['PQ_0213'])) { $insert .= '.PQ_0213'; }
    if (isset($_POST['PQ_0214'])) { $insert .= '.PQ_0214'; }
    if (isset($_POST['PQ_0215'])) { $insert .= '.PQ_0215'; }
    if (isset($_POST['PQ_0216'])) { $insert .= '.PQ_0216'; }
    if (isset($_POST['PQ_0217'])) { $insert .= '.PQ_0217'; }
    if (isset($_POST['PQ_0218'])) { $insert .= '.PQ_0218'; }
    if (isset($_POST['PQ_0219'])) { $insert .= '.PQ_0219'; }

    //03 Fusta
    if (isset($_POST['PQ_0311'])) { $insert .= '.PQ_0311'; }

    //04 Formigo
    if (isset($_POST['PQ_0411'])) { $insert .= '.PQ_0411'; }
    if (isset($_POST['PQ_0413'])) { $insert .= '.PQ_0413'; }

    //05 Acer
    if (isset($_POST['PQ_0501'])) { $insert .= '.PQ_0501'; }
    if (isset($_POST['PQ_0511'])) { $insert .= '.PQ_0511'; }

    //06 Fabrica
    if (isset($_POST['PQ_06X1'])) { $insert .= '.PQ_06X1'; }
    if (isset($_POST['PQ_0612'])) { $insert .= '.PQ_0612'; }
```

```php
    if (isset($_POST['PQ_0613'])) { $insert .= '.PQ_0613'; }
    if (isset($_POST['PQ_0616'])) { $insert .= '.PQ_0616'; }

    //07 Prefabricada
    if (isset($_POST['PQ_0711'])) { $insert .= '.PQ_0711'; }

    //08 Cobertes
    if (isset($_POST['PQ_08A1'])) { $insert .= '.PQ_08A1'; }
    if (isset($_POST['PQ_08B1'])) { $insert .= '.PQ_08B1'; }
    if (isset($_POST['PQ_08C1'])) { $insert .= '.PQ_08C1'; }
    if (isset($_POST['PQ_0811'])) { $insert .= '.PQ_0811'; }
    if (isset($_POST['PQ_0831'])) { $insert .= '.PQ_0831'; }
    if (isset($_POST['PQ_0841'])) { $insert .= '.PQ_0841'; }
    if (isset($_POST['PQ_0851'])) { $insert .= '.PQ_0851'; }
    if (isset($_POST['PQ_0861'])) { $insert .= '.PQ_0861'; }
    if (isset($_POST['PQ_0871'])) { $insert .= '.PQ_0871'; }
    if (isset($_POST['PQ_0881'])) { $insert .= '.PQ_0881'; }
    if (isset($_POST['PQ_0891'])) { $insert .= '.PQ_0891'; }

    //09 Tancaments Exteriors
    if (isset($_POST['PQ_0911'])) { $insert .= '.PQ_0911'; }
    if (isset($_POST['PQ_0921'])) { $insert .= '.PQ_0921'; }
    if (isset($_POST['PQ_0931'])) { $insert .= '.PQ_0931'; }
    if (isset($_POST['PQ_0941'])) { $insert .= '.PQ_0941'; }
    if (isset($_POST['PQ_0961'])) { $insert .= '.PQ_0961'; }
    if (isset($_POST['PQ_0971'])) { $insert .= '.PQ_0971'; }
    if (isset($_POST['PQ_0981'])) { $insert .= '.PQ_0981'; }

    //10 Tancaments Interiors
    if (isset($_POST['PQ_1021'])) { $insert .= '.PQ_1021'; }
    if (isset($_POST['PQ_1031'])) { $insert .= '.PQ_1031'; }

    //13 Acabats
    if (isset($_POST['PQ_13A1'])) { $insert .= '.PQ_13A1'; }
    if (isset($_POST['PQ_13B1'])) { $insert .= '.PQ_13B1'; }
    if (isset($_POST['PQ_1311'])) { $insert .= '.PQ_1311'; }
    if (isset($_POST['PQ_1321'])) { $insert .= '.PQ_1321'; }
    if (isset($_POST['PQ_1331'])) { $insert .= '.PQ_1331'; }
    if (isset($_POST['PQ_1341'])) { $insert .= '.PQ_1341'; }
    if (isset($_POST['PQ_1351'])) { $insert .= '.PQ_1351'; }
    if (isset($_POST['PQ_1371'])) { $insert .= '.PQ_1371'; }
    if (isset($_POST['PQ_1381'])) { $insert .= '.PQ_1381'; }
    if (isset($_POST['PQ_1391'])) { $insert .= '.PQ_1391'; }

    //14 Paviments
    if (isset($_POST['PQ_14A1'])) { $insert .= '.PQ_14A1'; }
    if (isset($_POST['PQ_14B1'])) { $insert .= '.PQ_14B1'; }
    if (isset($_POST['PQ_14C1'])) { $insert .= '.PQ_14C1'; }
    if (isset($_POST['PQ_14D1'])) { $insert .= '.PQ_14D1'; }
    if (isset($_POST['PQ_14E1'])) { $insert .= '.PQ_14E1'; }
    if (isset($_POST['PQ_14F1'])) { $insert .= '.PQ_14F1'; }
    if (isset($_POST['PQ_1421'])) { $insert .= '.PQ_1421'; }
    if (isset($_POST['PQ_1431'])) { $insert .= '.PQ_1431'; }

    //15 Elements practicables
    if (isset($_POST['PQ_15P1'])) { $insert .= '.PQ_15P1'; }
    if (isset($_POST['PQ_1511'])) { $insert .= '.PQ_1511'; }
    if (isset($_POST['PQ_1531'])) { $insert .= '.PQ_1531'; }

    //16 Proteccio
    if (isset($_POST['PQ_1611'])) { $insert .= '.PQ_1611'; }
    if (isset($_POST['PQ_1641'])) { $insert .= '.PQ_1641'; }

    //17 Sanejament
    if (isset($_POST['PQ_1711'])) { $insert .= '.PQ_1711'; }
    if (isset($_POST['PQ_1741'])) { $insert .= '.PQ_1741'; }
    if (isset($_POST['PQ_1761'])) { $insert .= '.PQ_1761'; }
    if (isset($_POST['PQ_1771'])) { $insert .= '.PQ_1771'; }
    if (isset($_POST['PQ_1781'])) { $insert .= '.PQ_1781'; }

    //18 Calefaccio
    if (isset($_POST['PQ_1811'])) { $insert .= '.PQ_1811'; }
```

```
    if (isset($_POST['PQ_1821'])) { $insert .= '.PQ_1821'; }

    //19 Climatitzacio
    if (isset($_POST['PQ_1911'])) { $insert .= '.PQ_1911'; }

    //20 Electricitat
    if (isset($_POST['PQ_2011'])) { $insert .= '.PQ_2011'; }
    if (isset($_POST['PQ_2013'])) { $insert .= '.PQ_2013'; }
    if (isset($_POST['PQ_2021'])) { $insert .= '.PQ_2021'; }
    if (isset($_POST['PQ_2031'])) { $insert .= '.PQ_2031'; }

    //21 Gas
    if (isset($_POST['PQ_2111'])) { $insert .= '.PQ_2111'; }

    //22 Aigua
    if (isset($_POST['PQ_2211'])) { $insert .= '.PQ_2211'; }
    if (isset($_POST['PQ_2213'])) { $insert .= '.PQ_2213'; }

    //23 Foc
    if (isset($_POST['PQ_2311'])) { $insert .= '.PQ_2311'; }

    //24 Parallamp
    if (isset($_POST['PQ_2411'])) { $insert .= '.PQ_2411'; }

    //25 Telecomunicacio
    if (isset($_POST['PQ_2511'])) { $insert .= '.PQ_2511'; }

    //26 Elevadors
    if (isset($_POST['PQ_2621'])) { $insert .= '.PQ_2621'; }

    //28 Altres
    if (isset($_POST['PQ_2811'])) { $insert .= '.PQ_2811'; }

$query = "UPDATE project SET ".$control_de." = '".$insert."' WHERE id = '".$project_id."' ";

$update = mysql_query($query);

} ?>

<ul>

<?php

//LISTA CON TODOS LOS PROCESOS QUE SE HAN ESCOGIDO

$project_query = $db->query("SELECT * FROM project WHERE id = '".$project_id."' ");

$project = $db->fetch($project_query);

//SEPARA DEL TEXTO TODOS LOS PROCESOS

$processos = $project[$control_de];

$proces = explode(".", $processos);

?>

<?php //01 Moviments de terres
titulo_procesos ("PQ_01", $proces);
procesos_individuales ($proces, "PQ_01", $control_de, $project_id, $user_id); ?>

<?php //02 Fonaments
titulo_procesos ("PQ_02", $proces);
procesos_individuales ($proces, "PQ_02", $control_de, $project_id, $user_id); ?>

<?php //03 Fusta
titulo_procesos ("PQ_03", $proces);
procesos_individuales ($proces, "PQ_03", $control_de, $project_id, $user_id); ?>

<?php //04 Formigo
titulo_procesos ("PQ_04", $proces);
procesos_individuales ($proces, "PQ_04", $control_de, $project_id, $user_id); ?>
```

```php
<?php //05 Acer
titulo_procesos ("PQ_05", $proces);
procesos_individuales ($proces, "PQ_05", $control_de, $project_id, $user_id); ?>

<?php //06 Fabrica
titulo_procesos ("PQ_06", $proces);
procesos_individuales ($proces, "PQ_06", $control_de, $project_id, $user_id); ?>

<?php //07 Prefabricada
titulo_procesos ("PQ_07", $proces);
procesos_individuales ($proces, "PQ_07", $control_de, $project_id, $user_id); ?>

<?php //08 Cobertes
titulo_procesos ("PQ_08", $proces);
procesos_individuales ($proces, "PQ_08", $control_de, $project_id, $user_id); ?>

<?php //09 Tancaments exteriors
titulo_procesos ("PQ_09", $proces);
procesos_individuales ($proces, "PQ_09", $control_de, $project_id, $user_id); ?>

<?php //10 Tancaments interiors
titulo_procesos ("PQ_10", $proces);
procesos_individuales ($proces, "PQ_10", $control_de, $project_id, $user_id); ?>

<?php //13 Acabats
titulo_procesos ("PQ_13", $proces);
procesos_individuales ($proces, "PQ_13", $control_de, $project_id, $user_id); ?>

<?php //14 Paviments
titulo_procesos ("PQ_14", $proces);
procesos_individuales ($proces, "PQ_14", $control_de, $project_id, $user_id); ?>

<?php //15 Elements practicables
titulo_procesos ("PQ_15", $proces);
procesos_individuales ($proces, "PQ_15", $control_de, $project_id, $user_id); ?>

<?php //16 Protecció
titulo_procesos ("PQ_16", $proces);
procesos_individuales ($proces, "PQ_16", $control_de, $project_id, $user_id); ?>

<?php //17 Sanejament
titulo_procesos ("PQ_17", $proces);
procesos_individuales ($proces, "PQ_17", $control_de, $project_id, $user_id); ?>

<?php //18 Calefacció
titulo_procesos ("PQ_18", $proces);
procesos_individuales ($proces, "PQ_18", $control_de, $project_id, $user_id); ?>

<?php //19 Climatitzacio
titulo_procesos ("PQ_19", $proces);
procesos_individuales ($proces, "PQ_19", $control_de, $project_id, $user_id); ?>

<?php //20 Electricitat
titulo_procesos ("PQ_20", $proces);
procesos_individuales ($proces, "PQ_20", $control_de, $project_id, $user_id); ?>

<?php //21 Gas
titulo_procesos ("PQ_21", $proces);
procesos_individuales ($proces, "PQ_21", $control_de, $project_id, $user_id); ?>

<?php //22 Aigua
titulo_procesos ("PQ_22", $proces);
procesos_individuales ($proces, "PQ_22", $control_de, $project_id, $user_id); ?>

<?php //23 Foc
titulo_procesos ("PQ_23", $proces);
procesos_individuales ($proces, "PQ_23", $control_de, $project_id, $user_id); ?>

<?php //24 Parallamp
titulo_procesos ("PQ_24", $proces);
procesos_individuales ($proces, "PQ_24", $control_de, $project_id, $user_id); ?>
```

```
<?php //25 Telecomunicació
titulo_procesos ("PQ_25", $proces);
procesos_individuales ($proces, "PQ_25", $control_de, $project_id, $user_id); ?>

<?php //26 Elevadors
titulo_procesos ("PQ_26", $proces);
procesos_individuales ($proces, "PQ_26", $control_de, $project_id, $user_id); ?>

<?php //28 Altres
titulo_procesos ("PQ_28", $proces);
procesos_individuales ($proces, "PQ_28", $control_de, $project_id, $user_id); ?>

    <li class="title">Afegir procés</li>

    <li>
        <a href="project_add_proces.php?id=<?php echo urlencode($user_id); ?>&project_id=<?php echo
urlencode($project_id); ?>&control_de=<?php echo urlencode($control_de); ?>">
        <span class="name">Tots els procesos</span>
        <span class="arrowbig"></span>
        </a>
    </li>
</ul>

</div>

<?php get_footer();

} else { user_not_allowed(); } ?>
```

```php
<?php

require_once "../includes/initialize.php";

$user_id = $_GET['id'];
$project_id = $_GET['project_id'];
$control_de = 'recepcio';
$codi = $_GET['codi'];

if(user_validates($user_id, $project_id) && !empty($codi)) { ?>

<?php get_header('list'); ?>

<?php $data = $recepcio->data($codi); ?>

<?php
$back = "../project_control.php?id={$user_id}&project_id={$project_id}&control_de={$control_de}";
top_proces_bar($codi, $back); ?>

<div id="content">

<?php

if($_POST['guardar']){

        foreach ($_POST as $key => $value){
            if(is_numeric($key)){
                $post_array[] = $key;
            }
        }

        if($post_array[0] != 0){

            $to_save = implode(', ', $post_array);

            $call_to_recepcio = "SELECT * FROM recepcio WHERE codi = '".$codi."' AND project_id = '".
$project_id."'";
            $call_query = $db->query($call_to_recepcio);
            $call_array = $db->fetch($call_query);

            if ($call_array == NULL){
                //Si es la primera vegada que guarda algo, inserta a la taula
                $query = "INSERT INTO recepcio (user_id, project_id, codi, marcats) VALUES ('".
$user_id."', '".$project_id."', '".$codi."', '".$to_save."')";
            } else {
                $query = "UPDATE recepcio SET marcats = '".$to_save."' WHERE project_id = '".
$project_id."' AND codi = '".$codi."'";
            }

        } else {
            //Si no hi ha res marcat, es borra de la db
            $query = "DELETE FROM recepcio WHERE project_id = '".$project_id."' AND codi = '".
$codi."'";

        }
        $db->query($query);
    }

?>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'].'?'.$_SERVER['argv'][0]; ?>" name="submit">

<ul>

<li class="title">Element</li>

<?php

$elements = explode(' | ', $data[0]);

foreach($elements as $key => $element){
```

```
    $element_control = explode(' - ', $element);

    $titol_element = $element_control[0];

    if($element_control[1] == NULL){ ?>

    <li>
        <a href="javascript:togglecomments(<?php echo $key+100; ?>)">
            <?php echo $titol_element; ?>
        </a>
    </li>

    <?php } else { ?>

    <li>
        <a href="javascript:togglecomments(<?php echo $key+100; ?>)">
            <?php echo $titol_element; ?>
        <span class="arrow"></span>
        </a>
    </li>

    <div id="<?php echo $key+100; ?>" class="hidden" >

    <?php $per_controlar = explode(' & ', $element_control[1]);

    foreach($per_controlar as $id => $control_individual) {

        if(!empty($control_individual)) { ?>

        <?php   $var = $key.$id+1000; //Javascritp no deja empezar con 0

        $element = explode(' @ ', $control_individual);

        $titol_element = $element[0];
        $docs_element = $element[1]; ?>

        <p class="p-hidden"><input type="checkbox" name="<?php echo $var; ?>" <?php $recepcio->checked
($var, $codi, $project_id); ?>> <a href="javascript:togglecomments(<?php echo $var; ?>)">
            <?php echo $titol_element; ?>

<?php if($docs_element != NULL){
//Si no hi ha documentació a presentar, que no surti el (+) ?>
 <img src="<?php echo HOME; ?>/app/theme/icons/plus.png" alt="(+)" />
<?php } ?>

        </a></p>

            <div id="<?php echo $var; ?>" class="hidden">

                <?php $doc_element = explode(' ', $docs_element); ?>

                <?php foreach ($doc_element as $doc){ ?>

                    <?php echo $recepcio->tipus_doc($doc); ?>

                <?php } ?>

            </div>

    <?php  }

        } ?>

    </div>

    <?php } #if element_control[1] == NULL ?>

<?php } ?>

<li class="title">Opcions</li>

    <li class="form">
```

```
        <input type="submit" name="guardar" value="Guardar" />
    </li>

</ul>

</form>

</div>

<?php get_footer(); ?>

<?php } else { user_not_allowed(); } ?>
```

```php
<?php require "../../includes/initialize.php";

$user_id = $_GET['id'];
$project_id = $_GET['project_id'];
$control_de = $_GET['control_de'];

if(user_validates($user_id, $project_id)) { ?>

<?php get_header('list'); ?>

<?php

switch ($control_de) {
    case 'recepcio';
        $title = 'Impr. Rec.';
    break;
    case 'execucio';
        $title = "Impr. Exe.";
    break;
    case 'acabada';
        $title = "Impr. Fin.";
    break;
}

top_bar($title, "../../project_print.php?id={$user_id}&project_id={$project_id}"); ?>

<div id="content">

<ul>

<?php

//LISTA CON TODOS LOS PROCESOS QUE SE HAN ESCOGIDO

$project_query = $db->query("SELECT * FROM project WHERE id = '".$project_id."' ");

$project = $db->fetch($project_query);

//SEPARA DEL TEXTO TODOS LOS PROCESOS

$processos = $project[$control_de];

$proces = explode(".", $processos);

?>

<?php //01 Moviments de terres
titulo_procesos ("PQ_01", $proces);
procesos_individuales ($proces, "PQ_01", $control_de, $project_id, $user_id); ?>

<?php //02 Fonaments
titulo_procesos ("PQ_02", $proces);
procesos_individuales ($proces, "PQ_02", $control_de, $project_id, $user_id); ?>

<?php //03 Fusta
titulo_procesos ("PQ_03", $proces);
procesos_individuales ($proces, "PQ_03", $control_de, $project_id, $user_id); ?>

<?php //04 Formigo
titulo_procesos ("PQ_04", $proces);
procesos_individuales ($proces, "PQ_04", $control_de, $project_id, $user_id); ?>

<?php //05 Acer
titulo_procesos ("PQ_05", $proces);
procesos_individuales ($proces, "PQ_05", $control_de, $project_id, $user_id); ?>

<?php //06 Fabrica
titulo_procesos ("PQ_06", $proces);
procesos_individuales ($proces, "PQ_06", $control_de, $project_id, $user_id); ?>

<?php //07 Prefabricada
titulo_procesos ("PQ_07", $proces);
```

```
procesos_individuales ($proces, "PQ_07", $control_de, $project_id, $user_id); ?>

<?php //08 Cobertes
titulo_procesos ("PQ_08", $proces);
procesos_individuales ($proces, "PQ_08", $control_de, $project_id, $user_id); ?>

<?php //09 Tancaments exteriors
titulo_procesos ("PQ_09", $proces);
procesos_individuales ($proces, "PQ_09", $control_de, $project_id, $user_id); ?>

<?php //10 Tancaments interiors
titulo_procesos ("PQ_10", $proces);
procesos_individuales ($proces, "PQ_10", $control_de, $project_id, $user_id); ?>

<?php //13 Acabats
titulo_procesos ("PQ_13", $proces);
procesos_individuales ($proces, "PQ_13", $control_de, $project_id, $user_id); ?>

<?php //14 Paviments
titulo_procesos ("PQ_14", $proces);
procesos_individuales ($proces, "PQ_14", $control_de, $project_id, $user_id); ?>

<?php //15 Elements practicables
titulo_procesos ("PQ_15", $proces);
procesos_individuales ($proces, "PQ_15", $control_de, $project_id, $user_id); ?>

<?php //16 Protecció
titulo_procesos ("PQ_16", $proces);
procesos_individuales ($proces, "PQ_16", $control_de, $project_id, $user_id); ?>

<?php //17 Sanejament
titulo_procesos ("PQ_17", $proces);
procesos_individuales ($proces, "PQ_17", $control_de, $project_id, $user_id); ?>

<?php //18 Calefacció
titulo_procesos ("PQ_18", $proces);
procesos_individuales ($proces, "PQ_18", $control_de, $project_id, $user_id); ?>

<?php //19 Climatitzacio
titulo_procesos ("PQ_19", $proces);
procesos_individuales ($proces, "PQ_19", $control_de, $project_id, $user_id); ?>

<?php //20 Electricitat
titulo_procesos ("PQ_20", $proces);
procesos_individuales ($proces, "PQ_20", $control_de, $project_id, $user_id); ?>

<?php //21 Gas
titulo_procesos ("PQ_21", $proces);
procesos_individuales ($proces, "PQ_21", $control_de, $project_id, $user_id); ?>

<?php //22 Aigua
titulo_procesos ("PQ_22", $proces);
procesos_individuales ($proces, "PQ_22", $control_de, $project_id, $user_id); ?>

<?php //23 Foc
titulo_procesos ("PQ_23", $proces);
procesos_individuales ($proces, "PQ_23", $control_de, $project_id, $user_id); ?>

<?php //24 Parallamp
titulo_procesos ("PQ_24", $proces);
procesos_individuales ($proces, "PQ_24", $control_de, $project_id, $user_id); ?>

<?php //25 Telecomunicació
titulo_procesos ("PQ_25", $proces);
procesos_individuales ($proces, "PQ_25", $control_de, $project_id, $user_id); ?>

<?php //26 Elevadors
titulo_procesos ("PQ_26", $proces);
procesos_individuales ($proces, "PQ_26", $control_de, $project_id, $user_id); ?>

<?php //28 Altres
titulo_procesos ("PQ_28", $proces);
```

```
procesos_individuales ($proces, "PQ_28", $control_de, $project_id, $user_id); ?>

</ul>

</div>

<?php get_footer();

} else { user_not_allowed(); } ?>
```

```php
<?php

require "../../../includes/initialize.php";

$user_id = $_GET['id'];
$project_id = $_GET['project_id'];
$codi = $_GET['codi'];

if(user_validates($user_id, $project_id)) {

//OBTENIM LES DADES DEL PROJECTE
$dades_project_query = $db->query("SELECT ref, projecte FROM project WHERE user_id = '".$user_id."'
AND id = '".$project_id."'");
$dades_project = $db->fetch($dades_project_query);


//OBTENIM LES DADES DELS PROCESOS FETS
$controls_fets_query = $db->query("SELECT marcats FROM recepcio WHERE project_id = '".$project_id."'
AND codi = '".$codi."'");
$controls_fets_array = $db->fetch($controls_fets_query);
$controls_fets = explode(', ', $controls_fets_array[0]);

$pdf=new FPDF();
$pdf->AddPage();

$pdf->SetTitle($pdf_settitle);
$pdf->SetAuthor($pdf_setauthor);
$pdf->SetSubject($pdf_setsubject);
$pdf->SetKeywords($pdf_setkeywords);
$pdf->SetCreator($pdf_setcreator);

$pdf->SetLineWidth(0.15);
$pdf->SetLeftMargin(17);

//Imagen en la esquina superior izquierda
$pdf->Image("../../../includes/pdf/img/recepcio/top.jpg", 17, 8, 170, 45, "JPG");

$pdf->SetFont('Arial','',10);

$pdf->Cell(170, 24, '', 0, 1);
$pdf->Cell(170,5,"{$dades_project[projecte]}", 0, 1);

$pdf->Cell(170, 4, '', 0, 1);
$pdf->Cell(170,5,"{$dades_project[ref]}", 0, 1);
$pdf->Cell(170, 4.95, '', 0, 1);

$data = $recepcio->data($codi);

//TITULO EN LAS TABLAS CON EL CONTROL
$pdf->SetFillColor(220);
$pdf->Cell(169.96, 5, $data[1], 1, 1, "L", true);


//EMPIEZA LOS LOOPS PARA QUE CADA UNO TENGA UN NUMERO
$elements = explode(' | ', $data[0]);

//Busca que el lot tenga de el codigo del titulo
foreach ($controls_fets as $control_fet){
    $titol_per_controlar[] = substr($control_fet, -2, 1);
}

foreach($elements as $key => $element){

    $element_control = explode(' - ', $element);

    //Si el codigo del lote, está entre los que se han hecho
    if(in_array($key, $titol_per_controlar)){

        $pdf->SetFillColor(240);
        $pdf->SetFont('Arial','',7);
        $pdf->Cell(89.5, 5, $element_control[0], 1, 0, "L", true);
        $pdf->Cell(13.41, 5, '', 1, 0);
```

```
        $pdf->Cell(22.35, 5, '', 1, 0);
        $pdf->Cell(44.7, 5, '', 1, 1);

    $per_controlar = explode(' & ', $element_control[1]);

    foreach($per_controlar as $id => $control_individual) {

        $var = $key.$id+1000;

        //Si encuentra un proceso que esté hecho, lo empieza a escribir
        if (in_array($var, $controls_fets)){

            $element = explode(' @ ', $control_individual);

            $pdf->SetFont('Arial','',7);
            $casella_control = substr($element[0], 0, 75);
            $pdf->Cell(89.5, 5, $casella_control, 1, 0, "L");

            $docs_element = $element[1];

            $doc_element = explode(' ', $docs_element);

                //Empieza a pintar las celdas por colores
                for ($i=0; $i<17; $i++){

                    $oscuro = $i.'a';
                    $claro = $i.'b';

                    if(in_array($oscuro, $doc_element)){
                        $pdf->SetFillColor(70);
                        $pdf->Cell(4.47, 5, '', 1, 0, "L", true);
                    } elseif(in_array($claro, $doc_element)){
                        $pdf->SetFillColor(150);
                        $pdf->Cell(4.47, 5, '', 1, 0, "L", true);
                    } else {
                        $pdf->Cell(4.47, 5, '', 1, 0, "L");

                    }

                }

                $pdf->Cell(4.47, 5, '', 1, 1, "L"); #17
        }
    }

    }

}

$pdf->SetFont('Arial','',7);
$pdf->Cell(170, 5, '',0, 1);
$pdf->Cell(112, 20, '',"LTR", 0);

$pdf->Cell(30, 5, 'Data inici: ',0, 0, "R");
$pdf->Cell(28, 5, '',1, 1);

$pdf->Cell(58, 1, '',0, 1);

$pdf->Cell(142, 5, 'Data final: ',0, 0, "R");
$pdf->Cell(28, 5, '',1, 1);

$pdf->SetFont('Arial','',9);
$pdf->Cell(112, 9, '',0, 1);
$pdf->Cell(112, 5, "Firmat pel Director d'Execucio de l'Obra", "LRB", 1, "C", false);

//COMIENZA EMAIL

$datos_email = datos_enviar_email($user_id, $codi, 'Recepcio');

$from       = $datos_email['from'];
$from_name  = $datos_email['from_name'];
$subject    = $datos_email['subject'];
```

```php
$body       = $datos_email['body'];
$to         = $datos_email['to'];
$doc_name   = $datos_email['doc_name'];

if($to != 'nadie'){
    $doc = $pdf->Output('', 'S');

    $mail = new PHPMailer();
    $mail->IsSMTP();
    $mail->SMTPAuth = true;
    $mail->Host = SMTPSERVER;
    $mail->Username = SMTPUSER;
    $mail->Password = SMTPPWD;

    $mail->SetFrom($from, $from_name);
    $mail->Subject = $subject;
    $mail->IsHTML(true);
    $mail->Body = $body;
    $mail->AddAddress($to);
    $mail->AddStringAttachment($doc, $doc_name, 'base64', 'application/pdf');

    $mail->Send();
}

$pdf->Output();

} else { user_not_allowed(); } ?>
```

```php
<?php

require "../../../includes/initialize.php";

$user_id = $_GET['id'];
$project_id = $_GET['project_id'];
$codi = $_GET['codi'];

if(user_validates($user_id, $project_id)) {

//OBTENIM LES DADES DEL PROJECTE
$dades_project_query = $db->query("SELECT ref, projecte FROM project WHERE user_id = '".$user_id."'
AND id = '".$project_id."'");
$dades_project = $db->fetch($dades_project_query);


//OBTENIM LES DADES DELS PROCESOS FETS
$controls_fets_query = $db->query("SELECT lots, fets FROM execucio WHERE project_id = '".
$project_id."' AND codi = '".$codi."'");
$controls_fets_array = $db->fetch($controls_fets_query);

$controls_fets_amb_data = explode(', ', $controls_fets_array[1]);

foreach ($controls_fets_amb_data as $control_fet_amb_data){
    $controls_fets_explode = explode(' @ ', $control_fet_amb_data);
    $controls_fets[] = $controls_fets_explode[0];
    $lots_fets[] = substr($controls_fets_explode[0], 0, 1);
    $dates_controls[] = $controls_fets_explode[1];
}

$proces = $execucio->data($codi);

$parametres_execucio = $proces['parametres_execucio'];


//EMPIEZA EL PDF

$pdf=new FPDF();
$pdf->AddPage();

$pdf->SetTitle($pdf_settitle);
$pdf->SetAuthor($pdf_setauthor);
$pdf->SetSubject($pdf_setsubject);
$pdf->SetKeywords($pdf_setkeywords);
$pdf->SetCreator($pdf_setcreator);

$pdf->SetLineWidth(0.15);
$pdf->SetLeftMargin(17);

//Imagen en la esquina superior izquierda
$pdf->Image("../../../includes/pdf/img/execucio/top.jpg", 17, 8, 170, 45, "JPG");

$pdf->SetFont('Arial','',10);

$pdf->Cell(170, 17.2, '', 0, 1);
$pdf->Cell(8.5, 5, '', 0, 0);
$pdf->Cell(160,5,"{$dades_project[projecte]}", 0, 1);

$pdf->Cell(170, 4, '', 0, 1);
$pdf->Cell(17, 5, '', 0, 0);
$pdf->Cell(160,5,"{$dades_project[ref]}", 0, 1);

$pdf->Cell(170, 11.5, "", 0, 1);

$pdf->SetFont('Arial','',9);
$pdf->SetFillColor(220);
$pdf->Cell(169.96, 5, $execucio->name($codi), 1, 1, "L", true);


$num_lots = $execucio->num_lots($project_id, $codi);

while ($i < $num_lots) { $i++;
```

```php
if(in_array($i, $lots_fets)) {

        $pdf->SetFont('Arial','',7);
        $pdf->SetFillColor(240);
        $pdf->Cell(134.28, 5, "Lot {$i}", 1, 0, "L", true);

        $pdf->Cell(35.68, 5, "", 1, 1, "L", true);

        foreach($parametres_execucio as $key => $parametre) {

            # id será la suma de número de lote + orden
            $id = $i.$key;

            if (in_array($id, $controls_fets)) {

                $comprovacions_fetes_amb_data = $parametre.' ['.$dates_controls[$key].']';

                $pdf->SetFont('Arial','',7);
                $pdf->Cell(134.28, 5, "{$parametre}", 1, 0, "L");

                $pdf->SetFillColor(70);
                $pdf->Cell(4.46, 5, "", 1, 0, "L", true);
                $pdf->Cell(4.46, 5, "", 1, 0, "L");
                $pdf->Cell(4.46, 5, "", 1, 0, "L");
                $pdf->Cell(4.46, 5, "", 1, 0, "L", true);
                $pdf->Cell(4.46, 5, "", 1, 0, "L");
                $pdf->Cell(4.46, 5, "", 1, 0, "L");
                $pdf->Cell(4.46, 5, "", 1, 0, "L");
                $pdf->Cell(4.46, 5, "", 1, 1, "L");

            }
        }
    }
} #end while */

$pdf->SetFont('Arial','',7);
$pdf->Cell(170, 5, '',0, 1);
$pdf->Cell(112, 20, '',"LTR", 0);

$pdf->Cell(30, 5, 'Data inici: ',0, 0, "R");
$pdf->Cell(28, 5, '',1, 1);

$pdf->Cell(58, 1, '',0, 1);

$pdf->Cell(142, 5, 'Data final: ',0, 0, "R");
$pdf->Cell(28, 5, '',1, 1);

$pdf->SetFont('Arial','',9);
$pdf->Cell(112, 9, '',0, 1);
$pdf->Cell(112, 5, "Firmat pel Director d'Execucio de l'Obra", "LRB", 1, "C", false);

//COMIENZA EMAIL

$datos_email = datos_enviar_email($user_id, $codi, 'Execucio');

$from       = $datos_email['from'];
$from_name  = $datos_email['from_name'];
$subject    = $datos_email['subject'];
$body       = $datos_email['body'];
$to         = $datos_email['to'];
$doc_name   = $datos_email['doc_name'];

if($to != 'nadie'){
    $doc = $pdf->Output('', 'S');

    $mail = new PHPMailer();
    $mail->IsSMTP();
    $mail->SMTPAuth = true;
    $mail->Host = SMTPSERVER;
    $mail->Username = SMTPUSER;
    $mail->Password = SMTPPWD;
```

```
    $mail->SetFrom($from, $from_name);
    $mail->Subject = $subject;
    $mail->IsHTML(true);
    $mail->Body = $body;
    $mail->AddAddress($to);
    $mail->AddStringAttachment($doc, $doc_name, 'base64', 'application/pdf');

    $mail->Send();
}

$pdf->Output();

} else { user_not_allowed(); } ?>
```

```php
<?php

require "../../../includes/initialize.php";

$user_id = $_GET['id'];
$project_id = $_GET['project_id'];
$codi = $_GET['codi'];

if(user_validates($user_id, $project_id)) {

//OBTENIM LES DADES DEL PROJECTE
$dades_project_query = $db->query("SELECT ref, projecte FROM project WHERE user_id = '".$user_id."'
AND id = '".$project_id."'");
$dades_project = $db->fetch($dades_project_query);

$pdf=new FPDF();
$pdf->AddPage();

$pdf->SetTitle($pdf_settitle);
$pdf->SetAuthor($pdf_setauthor);
$pdf->SetSubject($pdf_setsubject);
$pdf->SetKeywords($pdf_setkeywords);
$pdf->SetCreator($pdf_setcreator);

$pdf->SetLineWidth(0.15);
$pdf->SetLeftMargin(17);

//Imagen en la esquina superior izquierda
$pdf->Image("../../../includes/pdf/img/acabada/top.jpg", 17, 8, 170, 45, "JPG");

$pdf->SetFont('Arial','',10);

$pdf->Cell(170, 17.2, '', 0, 1);
$pdf->Cell(8.5, 5, '', 0, 0);
$pdf->Cell(160,5,"{$dades_project[projecte]}", 0, 1);

$pdf->Cell(170, 4, '', 0, 1);
$pdf->Cell(17, 5, '', 0, 0);
$pdf->Cell(160,5,"{$dades_project[ref]}", 0, 1);

$pdf->Cell(170, 11.5, '', 0, 1);



//EMPIEZA LOS LOOPS PARA QUE CADA UNO TENGA UN NUMERO
$cifrat     = $acabada->cifrat($codi);
$verifica   = $acabada->data($codi, 'verifica');
$control    = $acabada->data($codi, 'control');
$lotazos = $acabada->lotazos($codi, $project_id);

$lots_fets_query = $db->query("SELECT lots_fets FROM acabada WHERE project_id = '".$project_id."' AND
codi = '".$codi."'");
$lots_fets_array = $db->fetch($lots_fets_query);

$lots_fets = explode (', ', $lots_fets_array[0]);

foreach ($lots_fets as $lot_fet){
    $lot[] = substr($lot_fet, -1);
}

foreach ($lots_fets as $verifica_lot){
    $verifica_codi[] = substr($verifica_lot, -2, 1);
}

foreach($cifrat as $o) {
    foreach($lotazos as $id => $lotes){
        if($o[1] == $id && in_array($id, $verifica_codi)){
        $var = $o[0];

            //TITULO EN LAS TABLAS CON EL CONTROL
            $pdf->SetFont('Arial','',9);
            $pdf->SetFillColor(220);
```

```
            $pdf->Cell(169.96, 5, $verifica[$var], 1, 1, "L", true);

            for($i=1;$i<=$lotes;$i++){

                if(in_array($i, $lot)){

                $pdf->SetFont('Arial','',7);
                $pdf->SetFillColor(240);
                $pdf->Cell(134.28, 5, "Lot {$i}", 1, 0, "L", true);

                $pdf->Cell(35.68, 5, "", 1, 1, "L", true);


                $controls = explode(', ', $control[$id]);
                    foreach($controls as $or => $cind){

                    $codi_lot = $or.$id.$i;

                        if(in_array($codi_lot, $lots_fets)){

                            $pdf->SetFont('Arial','',7);
                            $pdf->Cell(134.28, 5, "{$cind}", 1, 0, "L");

                            $pdf->SetFillColor(70);
                            $pdf->Cell(4.46, 5, "", 1, 0, "L", true);
                            $pdf->Cell(4.46, 5, "", 1, 0, "L");
                            $pdf->Cell(4.46, 5, "", 1, 0, "L");
                            $pdf->Cell(4.46, 5, "", 1, 0, "L", true);
                            $pdf->Cell(4.46, 5, "", 1, 0, "L");
                            $pdf->Cell(4.46, 5, "", 1, 0, "L");
                            $pdf->Cell(4.46, 5, "", 1, 0, "L");
                            $pdf->Cell(4.46, 5, "", 1, 1, "L");

                            }

                        }
                    }
                }
        }
    }
}

$pdf->SetFont('Arial','',7);
$pdf->Cell(170, 5, '',0, 1);
$pdf->Cell(112, 20, '',"LTR", 0);

$pdf->Cell(30, 5, 'Data inici: ',0, 0, "R");
$pdf->Cell(28, 5, '',1, 1);

$pdf->Cell(58, 1, '',0, 1);

$pdf->Cell(142, 5, 'Data final: ',0, 0, "R");
$pdf->Cell(28, 5, '',1, 1);

$pdf->SetFont('Arial','',9);
$pdf->Cell(112, 9, '',0, 1);
$pdf->Cell(112, 5, "Firmat pel Director d'Execucio de l'Obra", "LRB", 1, "C", false);

//COMIENZA EMAIL

$datos_email = datos_enviar_email($user_id, $codi, 'Obra Finalitzada');

$from       = $datos_email['from'];
$from_name  = $datos_email['from_name'];
$subject    = $datos_email['subject'];
$body       = $datos_email['body'];
$to         = $datos_email['to'];
$doc_name   = $datos_email['doc_name'];

if($to != 'nadie'){
    $doc = $pdf->Output('', 'S');
```

```
    $mail = new PHPMailer();
    $mail->IsSMTP();
    $mail->SMTPAuth = true;
    $mail->Host = SMTPSERVER;
    $mail->Username = SMTPUSER;
    $mail->Password = SMTPPWD;

    $mail->SetFrom($from, $from_name);
    $mail->Subject = $subject;
    $mail->IsHTML(true);
    $mail->Body = $body;
    $mail->AddAddress($to);
    $mail->AddStringAttachment($doc, $doc_name, 'base64', 'application/pdf');

    $mail->Send();
}

$pdf->Output();

} else { user_not_allowed(); } ?>
```

```php
<?php

require_once "../includes/initialize.php";

$user_id = $_GET['id'];
$project_id = $_GET['project_id'];
$control_de = 'execucio';
$codi = $_GET['codi'];

if(user_validates($user_id, $project_id) && !empty($codi)) { ?>

<?php get_header('list'); ?>

<?php
$back = "../project_control.php?id={$user_id}&project_id={$project_id}&control_de={$control_de}";
top_proces_bar($codi, $back); ?>

<div id="content">

<?php

if($_POST['lot_delete']){
    $db->query("DELETE FROM execucio WHERE codi = '".$codi."' AND project_id = '".$project_id."'");
}


$proces = $execucio->data($codi);

$parametres_execucio = $proces['parametres_execucio'];
$tipus_control       = $proces['tipus_control'];
$unitat_lot          = $proces['unitat_lot'];
$tamany_lot          = $proces['tamany_lot'];

if($_POST['submit']){

    $valor = $_POST['total'];
    $top = valor_maxim($unitat_lot);

    if(is_numeric($valor)){
        if($valor<$top){
            $lots = $valor/$tamany_lot;
            $lots_totals = ceil($lots);

            $db->query("INSERT INTO execucio (user_id, project_id, codi, valor, lots) VALUES ('".
$user_id."', '".$project_id."', '".$codi."', '".$_POST['total']."', '".$lots_totals."')");

        } else { echo 'El valor introduit es massa gran.'; }
    } else {
        echo "Introdueix un valor numeric al formulari.";
    }
} ?>

<?php $num_lots = $execucio->num_lots($project_id, $codi);

if($execucio->sense_lots($project_id, $codi)){ ?>

<ul>

    <form method="post" action="<?php echo $_SERVER['PHP_SELF'].'?'.$_SERVER['argv'][0]; ?>"
name="submit">

        <li class="title">Formació de lots</li>

        <li class="form">
            <span class="narrow">
            <span class="name"><?php echo $unitat_lot; ?> totals</span>
                <input type="text" pattern="[0-9]*"name="total" />
            </span>
        </li>

        <?php //Si falta alguna parte, se añade en otro archivo
        $execucio->sense_lot_arxiu($codi); ?>
```

```
        <li class="form">
            <input type="submit" name="submit" value="Calcular" />
        </li>

    </form>

</ul>

<?php
//SI YA SABEMOS EL NUMERO DE LOTES, MUESTRA LOS QUE HAYA
} else { ?>

<?php
//Si se actualizan los datos, incluye los numeros en una lista
if($_POST['fets']){

$execucio->guardar_controls_fets($_POST, $project_id, $codi);


} ?>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'].'?'.$_SERVER['argv'][0]; ?>" name="fets"
id="submit">

<ul>

<?php $execucio->amb_lot_arxiu($codi, 1); ?>

    <script language="javascript">
    function checkAllLots(master){
        var checked = master.checked;
        var col = document.getElementsByClassName("lot");
        for (var i=0;i<col.length;i++) {
            col[i].checked= checked;
        }
    }
    </script>


<li class="title"><input type="checkbox" onclick="checkAllLots(this)" /> <?php echo $execucio->name
($codi); ?></li>

<?php while ($i < $num_lots) { $i++; ?>



<script language="javascript">
function checkAll<?php echo $i; ?>(master){
    var checked = master.checked;
    var col = document.getElementsByClassName("<?php echo $i; ?>");
    for (var i=0;i<col.length;i++) {
        col[i].checked= checked;
    }
}
</script>

    <li class="title"><input type="checkbox" class="lot" onclick="checkAll<?php echo $i; ?>(this)"/ >
Lot <?php echo $i; ?></li>



    <?php foreach($parametres_execucio as $key => $parametre) {

        # id será la suma de número de lote + orden
        $id = $i.$key; ?>

        <li class="<?php $execucio->color_fet($id, $project_id, $codi); ?>">
            <a href="javascript:togglecomments(<?php echo $id; ?>)">
            <?php echo $execucio->titol_comprobacio($parametre); ?>
            <span class="arrow"></span>
            </a>
```

```
        </li>

    <div id="<?php echo $id; ?>" class="hidden">
        <p class="p-hidden">


<?php

$cont_exe = $tipus_control[$key];

$parts_cont_exe = explode(' -> ', $cont_exe);

$pcpal_cont_exe = $parts_cont_exe[0]; ?>

    <span class="name">

    <?php if ($parts_cont_exe[1] != NULL) { ?>
        <a href="javascript:togglecomments(<?php echo $id; ?>1)">
        <b>Control:</b> <?php echo $pcpal_cont_exe; ?>
        <img src="<?php echo HOME; ?>/app/theme/icons/plus.png" alt="(+)" />
        </a>
    <?php } else { ?>
        <b>Control:</b> <?php echo $pcpal_cont_exe; ?>
    <?php } ?>

    </span></p>

        <div id="<?php echo $id; ?>1" class="hidden">

        <?php

        $secundaris_cont_exe = explode(' & ', $parts_cont_exe[1]);

        foreach ($secundaris_cont_exe as $secundari_cont_exe) { ?>
            <p class="p-hidden establert_ehe">
                <?php if (substr($secundari_cont_exe, -1) == '*') { ?>
                    <b><?php echo substr($secundari_cont_exe, 0, -1); ?></b>
                <?php } else { ?>
                    &bull; <?php echo $secundari_cont_exe; ?>
                <?php } ?>
            </p>
        <?php } ?>

        </div>

        <p class="p-hidden"><span class="name">
            <?php $execucio->marcar_com_fet($i, $key, $project_id, $codi); ?>
        </span></p>

        <!--<div id="<?php echo $id; ?>2" class="hidden">
            <?php $execucio->edit_date($id, $project_id, $codi); ?>
        </div>-->

    </div>

    <?php } ?>

<?php
} #end while

?>

<?php $execucio->amb_lot_arxiu($codi, 2); ?>

<li class="title">Opcions</li>

<li class="form">
<input type="submit" name="fets" value="Guardar" />
</li>

<li class="form">
<input type="submit" name="lot_delete" value="Eliminar lots" />
```

```
</li>

</ul>

</form>

<?php
} #end formulario de lotes
?>

</div>

<?php get_footer(); ?>

<?php } else { user_not_allowed(); } ?>
```

```php
<?php

require_once "../includes/initialize.php";

$user_id = $_GET['id'];
$project_id = $_GET['project_id'];
$control_de = 'acabada';
$codi = $_GET['codi'];

if(user_validates($user_id, $project_id) && !empty($codi)) { ?>

<?php get_header('list'); ?>

<?php
$back = "../project_control.php?id={$user_id}&project_id={$project_id}&control_de={$control_de}";
top_proces_bar($codi, $back); ?>

<div id="content">

<ul>

<?php

if($_POST['lot_delete']){
    $db->query("DELETE FROM acabada WHERE codi = '".$codi."' AND project_id = '".$project_id."'");
}

$cifrat     = $acabada->cifrat($codi);
$verifica   = $acabada->data($codi, 'verifica');
$control    = $acabada->data($codi, 'control');
$parametres = $acabada->data($codi, 'parametres');
$medicio    = $acabada->data($codi, 'medicio');

if($_POST['submit']){
    foreach ($cifrat as $o => $c){
        foreach($_POST as $key => $valor){
            if(is_numeric($key)){
                if($c[3] == $key){
                    $top = valor_maxim($medicio[$key]);
                    if($valor<$top){
                        $lotazos[] = ceil($valor/$parametres[$o]);
                    } else { $lotazos[] = 0; }
                }
            }
        }
    }

$save = implode(', ', $lotazos);

$query = "INSERT INTO acabada (user_id, project_id, codi, lotazos) VALUES ('".$user_id."', '".
$project_id."', '".$codi."', '".$save."')";

$db->query($query);

}

//Si ya estan creados los lotes, se ponen

$lotazos = $acabada->lotazos($codi, $project_id);

if($acabada->amb_lotazos($codi, $project_id)){

    //Guarda los lotes que estén hechos

    if($_POST['fets']){
        foreach ($_POST as $key => $value){
            if(is_numeric($key)){
                $post_array[] = $key;
            }
        }

        if($post_array[0] != 0){
```

```
            $to_save = implode(', ', $post_array);

            $query = "UPDATE acabada SET lots_fets = '".$to_save."' WHERE project_id = '".
$project_id."' AND codi = '".$codi."'";

        } else {

            $query = "UPDATE acabada SET lots_fets = '' WHERE project_id = '".$project_id."' AND codi
= '".$codi."'";

        }
        $db->query($query);
    }

?>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'].'?'.$_SERVER['argv'][0]; ?>" name="submit">

<?php
foreach($cifrat as $o) {
        foreach($lotazos as $id => $lotes){
            if($o[1] == $id){
                $var = $o[0]; ?>

                <script language="javascript">
                function checkAll<?php echo $id; ?>(master){
                    var checked = master.checked;
                    var col = document.getElementsByClassName("<?php echo $id; ?>");
                    for (var i=0;i<col.length;i++) {
                        col[i].checked= checked;
                    }
                }
                </script>

                <li class='title'><input type="checkbox" onclick="checkAll<?php echo $id; ?>(this)" />
<?php echo $verifica[$var]; ?></li>

                <?php for($i=1;$i<=$lotes;$i++){ ?>

                    <script language="javascript">
                    function checkAll<?php echo $id.$i; ?>(master){
                        var checked = master.checked;
                        var col = document.getElementsByClassName("<?php echo $id.$i; ?>");
                        for (var i=0;i<col.length;i++) {
                            col[i].checked= checked;
                        }
                    }
                    </script>

                    <li class='title'><input type="checkbox" class="<?php echo $id; ?>"
onclick="checkAll<?php echo $id.$i; ?>(this)"> Lot <?php echo $i; ?></li>

                    <?php $controls = explode(', ', $control[$id]);
                    foreach($controls as $or => $cind){ $codi_lot = $or.$id.$i; ?>

<p class="p-hidden <?php $acabada->clase($codi_lot, $codi, $project_id); ?>">
<span class="name">
    <input name="<?=$codi_lot?>" type="checkbox" class="<?php echo $id; ?> <?php echo $id.$i; ?>"<?php
$acabada->checked($codi_lot, $codi, $project_id); ?> />
    <b>Control: </b><?=$cind?>
</span></p>

                <?php }
                }
            }
        }
    }
?>

<?php $acabada->amb_lot_arxiu($codi); ?>

<li class="title">Opcions</li>
```

```
<li class="form">
    <input type="submit" name="fets" value="Guardar" />
</li>

<li class="form">
    <input type="submit" name="lot_delete" value="Eliminar lots" />
</li>

</form>

<?php

} else {

//Si todavia no existen los lotazos, se pone el formulario

?>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'].'?'.$_SERVER['argv'][0]; ?>" name="submit">

<li class="title">Formaci&oacute; de lots</li>

<?php foreach ($medicio as $key => $med){ ?>

    <li class="form">
        <span class="narrow">
        <span class="name"><?=$med?> totals</span>
            <input type="text" pattern="[0-9]*" name="<?=$key?>" />
        </span>
    </li>

<?php } ?>

<?php $acabada->sense_lot_arxiu($codi); ?>

    <li class="form">
        <input type="submit" name="submit" value="Calcular" />
    </li>

</form>

<?php } ?>

</ul>

</div>

<?php get_footer(); ?>

<?php } else { user_not_allowed(); } ?>
```

```
<?php require_once "includes/initialize.php";

$user_id = $_SESSION['user_id']; ?>

<?php if(user_validates()) { ?>

<?php get_header(); ?>

<?php top_bar('iQualit 0.1', '../index_mobile.php', 'H'); ?>

<div id="content">

<span class="graytitle">Projectes</span>

<ul class="pageitem">

<?php

//CREA UN LOOP CON NUESTROS PROYECTOS

$project_query = $db->query("SELECT * FROM project WHERE user_id = '".$user_id."' ");

while ($project = $db->fetch($project_query)) { ?>

    <li class="menu">

        <a href="project_view.php?id=<?php echo urlencode($user_id); ?>&project_id=<?php echo urlencode
($project['id']); ?>">
            <img alt="Veure Projecte" src="theme/icons/building_go.png" />
            <span class="name"><?php echo $project['projecte']; ?></span>
            <span class="arrowbig"></span>
        </a>

    </li>

<?php } ?>

    <li class="menu">
        <a href="project_create.php?id=<?php echo urlencode($user_id); ?>">
        <span class="name">Afegir projecte</span>
        <span class="arrow"></span>
        </a>
    </li>
</ul>

<span class="graytitle">Configuraci&oacute; personal</span>

<ul class="pageitem">
    <li class="menu">
        <a href="user_edit.php?id=<?php echo urlencode($user_id); ?>">
            <img alt="Description" src="theme/icons/cog.png" />
            <span class="name">Editar perfil</span>
            <span class="arrow"></span>
        </a>
    </li>
</ul>

<!--<span class="graytitle">Extra</span>

<ul class="pageitem">
    <li class="menu">
            <a href="register_mobile.php">
            <img alt="Description" src="theme/icons/book_open.png" />
            <span class="name">Veure tots els documents</span>
            <span class="arrow"></span>
        </a>
    </li>
</ul>-->

</div>

<?php get_footer(); ?>
```

```
<?php } else { user_not_allowed(); } ?>
```

```php
<?php require_once "includes/initialize.php";

if($_COOKIE['iQualit_user_id'] && $_COOKIE['iQualit_password']){
    setcookie("iQualit_password", "", time() - 60*60*24*30, '/');
    setcookie("iQualit_user_id", "", time() - 60*60*24*30, '/');
}

session_unset();
session_destroy(); ?>

<meta http-equiv="refresh" content="0;../index_mobile.php">
```

```php
<?php require_once "includes/initialize.php"; ?>

<?php if(!user_validates()) { ?>

<?php get_header(); ?>

<?php top_bar('Inicia sessió', '../index_mobile.php'); ?>

<div id="content">

<span class="graytitle">Entrar al teu compte</span>

<form method="post" action="index.php" name="loginform" id="loginform">

    <ul class="pageitem">
        <li class="form">
            <span class="narrow">
            <span class="name">Email</span>
            <input type="email" name="email" />
            </span>
        </li>

        <li class="form">
            <span class="narrow">
            <span class="name">Password</span>
            <input type="password" name="password" />
            </span>
        </li>
    </ul>

    <ul class="pageitem">
        <li class="menu no-hover">
            <span class="name">
            <input type="checkbox" name="remember" checked /> No tancar sessió</span>
        </li>
    </ul>

    <ul class="pageitem">
        <li class="form">
            <input type="submit" name="login" id="login" value="Accedir" />
        </li>
    </ul>

</form>

</div>

<?php get_footer(); ?>

<?php } else { ?>
<meta http-equiv="refresh" content="0;main.php">
<?php } ?>
```

```php
<?php require_once "includes/initialize.php";

//LA SESION HA SIDO INICIADA PREVIAMENTE TE ENVIA A MAIN DIRECTAMENTE

if(user_validates()) { ?>

<meta http-equiv="refresh" content="0;main.php">

<?php }

//EL USUARIO ACABA DE INTRODUCIR SUS DATOS EN EL FORMULARIO, LOS ACEPTA Y ENVIA A MAIN

elseif(!empty($_POST['email']) && !empty($_POST['password'])) {

    $email = mysql_real_escape_string($_POST['email']);
    $password = md5(md5(mysql_real_escape_string($_POST['password'])));
    $remember = $_POST['remember'];

    $checklogin_query = "SELECT email, password, user_id FROM users WHERE email = '".$email."' AND
Password = '".$password."'";
    $checklogin = $db->query($checklogin_query);

    $dades_user = $db->fetch($checklogin);

    $user_id = $dades_user['user_id'];

    //Si cuando hace login, encuentra sus datos, crea una sesion
    if(mysql_num_rows($checklogin) == 1) {

        $_SESSION['user_id'] = $user_id;
        $_SESSION['logged_in'] = 1;

        //Si se ha marcado la casilla de recordar, crea las cookies
        if($remember) {
            $time = time();
            setcookie("iQualit_user_id", $user_id, $time + 60*60*24*30, '/');
            setcookie("iQualit_password", $password, $time + 60*60*24*30, '/');
        } ?>

        <meta http-equiv="refresh" content="0;main.php">

    <?php

    } else {

    // ERROR DE CONTRASEÑA O NOMBRE DE USUARIO

    get_header(); ?>

    <?php top_bar('Acces denegat', 'index.php'); ?>

    <div id="content">

    <?php error_message('Les teves dades no han estat localitzades. Comprova que el nom d\'usuari i la
contrasenya siguin correctes', 'login.php'); ?>

    </div>

    <?php get_footer();

//LA SESION NO HA SIDO INICIADA, APARECE EL LOGIN

}


} else { ?>

<meta http-equiv="refresh" content="0;login.php">

<?php } ?>
```

```php
<?php

session_start();

if($_SERVER['HTTP_HOST'] == 'localhost'){
    $dbhost = "localhost";
    $dbname = "pfc";
    $dbuser = "root";
    $dbpass = "30058614";
} else {
    $dbhost = "localhost";
    $dbname = "iqualit";
    $dbuser = "myiqualit";
    $dbpass = "sinpassapelo";
}

//ABRIR CONEXIÓN CON LA BASE DE DATOS
$connection = mysql_connect($dbhost, $dbuser, $dbpass) or die("MySQL Error: " . mysql_error());

//SELECCIONA LA TABLA 'pfc' DE LA BASE DE DATOS
$db_users = mysql_select_db($dbname) or die("MySQL Error: " . mysql_error());

//PONE LA HORA
date_default_timezone_set('UTC');
```

```php
<?

//DATOS PARA EL PDF

$pdf_settitle = 'iQualit.com';
$pdf_setauthor = 'fer@iqualit.com';
$pdf_setsubject = 'Generat automaticament amb iQualit.';
$pdf_setkeywords = 'iqualit control qualitat';
$pdf_setcreator = 'iQualit.com (Versió 0.1)';

//DATOS PARA EL EMAIL

function datos_enviar_email($user_id, $codi, $control = ''){

    $datos_email['from'] = 'fer@iqualit.com';
    $datos_email['from_name'] = 'iQualit.com';

    $codi_query = mysql_query("SELECT titol FROM procesos WHERE codi = '".$codi."' ");
    $codi_fetch = mysql_fetch_array($codi_query);
    $codi_titol = $codi_fetch['titol'];
    $datos_email['subject'] = "iQualit.com - Control de {$control} per a {$codi_titol}";

    $datos_email['doc_name'] = "iQualit.com_{$control}_{$codi_titol}.pdf";

    $email_query = mysql_query("SELECT email FROM users WHERE user_id = '".$user_id."' ");
    $email_fetch = mysql_fetch_array($email_query);
    $email_to = $email_fetch['email'];

    if(valid_email($email_to)){
        $datos_email['to'] = $email_to;
    } elseif($email_to == 'ferran') {
        $datos_email['to'] = 'ferranfg@gmail.com';
    } elseif($email_to == 'joan') {
        $datos_email['to'] = 'joan.olona@upc.edu';
    } else {
        $datos_email['to'] = 'nadie';
    }

    $datos_email['body'] = "<em>Missatge generat automaticament per <a href='iqualit.com'>iQualit.com</
a>.<br/> Si us plau, no contesteu a aquest missatge.</em>";

    return $datos_email;
}
```

```php
<?php

define('SMTPUSER', 'ferranfg'); // sec. smtp username
define('SMTPPWD', 'meec'); // sec. password
define('SMTPSERVER', '212.36.75.250'); // sec. smtp server

/*~ class.phpmailer.php
.----------------------------------------------------------------------.
|  Software: PHPMailer - PHP email class                               |
|   Version: 5.1                                                       |
|   Contact: via sourceforge.net support pages (also www.worxware.com) |
|      Info: http://phpmailer.sourceforge.net                          |
|   Support: http://sourceforge.net/projects/phpmailer/                |
| -------------------------------------------------------------------- |
|     Admin: Andy Prevost (project admininistrator)                    |
|   Authors: Andy Prevost (codeworxtech) codeworxtech@users.sourceforge.net |
|          : Marcus Bointon (coolbru) coolbru@users.sourceforge.net    |
|   Founder: Brent R. Matzelle (original founder)                      |
| Copyright (c) 2004-2009, Andy Prevost. All Rights Reserved.          |
| Copyright (c) 2001-2003, Brent R. Matzelle                           |
| -------------------------------------------------------------------- |
|   License: Distributed under the Lesser General Public License (LGPL) |
|            http://www.gnu.org/copyleft/lesser.html                   |
| This program is distributed in the hope that it will be useful - WITHOUT |
| ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or |
| FITNESS FOR A PARTICULAR PURPOSE.                                    |
| -------------------------------------------------------------------- |
| We offer a number of paid services (www.worxware.com):               |
| - Web Hosting on highly optimized fast and secure servers            |
| - Technology Consulting                                              |
| - Oursourcing (highly qualified programmers and graphic designers)   |
'----------------------------------------------------------------------'
*/

/**
 * PHPMailer - PHP email transport class
 * NOTE: Requires PHP version 5 or later
 * @package PHPMailer
 * @author Andy Prevost
 * @author Marcus Bointon
 * @copyright 2004 - 2009 Andy Prevost
 * @version $Id: class.phpmailer.php 447 2009-05-25 01:36:38Z codeworxtech $
 * @license http://www.gnu.org/copyleft/lesser.html GNU Lesser General Public License
 */

if (version_compare(PHP_VERSION, '5.0.0', '<') ) exit("Sorry, this version of PHPMailer will only run
on PHP version 5 or greater!\n");

class PHPMailer {

  ////////////////////////////////////////////////
  // PROPERTIES, PUBLIC
  ////////////////////////////////////////////////

  /**
   * Email priority (1 = High, 3 = Normal, 5 = low).
   * @var int
   */
  public $Priority          = 3;

  /**
   * Sets the CharSet of the message.
   * @var string
   */
  public $CharSet           = 'iso-8859-1';

  /**
   * Sets the Content-type of the message.
   * @var string
   */
  public $ContentType       = 'text/plain';
```

```
    /**
     * Sets the Encoding of the message. Options for this are
     *  "8bit", "7bit", "binary", "base64", and "quoted-printable".
     * @var string
     */
    public $Encoding          = '8bit';

    /**
     * Holds the most recent mailer error message.
     * @var string
     */
    public $ErrorInfo         = '';

    /**
     * Sets the From email address for the message.
     * @var string
     */
    public $From              = 'root@localhost';

    /**
     * Sets the From name of the message.
     * @var string
     */
    public $FromName          = 'Root User';

    /**
     * Sets the Sender email (Return-Path) of the message.  If not empty,
     * will be sent via -f to sendmail or as 'MAIL FROM' in smtp mode.
     * @var string
     */
    public $Sender            = '';

    /**
     * Sets the Subject of the message.
     * @var string
     */
    public $Subject           = '';

    /**
     * Sets the Body of the message.  This can be either an HTML or text body.
     * If HTML then run IsHTML(true).
     * @var string
     */
    public $Body              = '';

    /**
     * Sets the text-only body of the message.  This automatically sets the
     * email to multipart/alternative.  This body can be read by mail
     * clients that do not have HTML email capability such as mutt. Clients
     * that can read HTML will view the normal Body.
     * @var string
     */
    public $AltBody           = '';

    /**
     * Sets word wrapping on the body of the message to a given number of
     * characters.
     * @var int
     */
    public $WordWrap          = 0;

    /**
     * Method to send mail: ("mail", "sendmail", or "smtp").
     * @var string
     */
    public $Mailer            = 'mail';

    /**
     * Sets the path of the sendmail program.
     * @var string
     */
    public $Sendmail          = '/usr/sbin/sendmail';
```

```
  /**
   * Path to PHPMailer plugins.  Useful if the SMTP class
   * is in a different directory than the PHP include path.
   * @var string
   */
  public $PluginDir         = '';

  /**
   * Sets the email address that a reading confirmation will be sent.
   * @var string
   */
  public $ConfirmReadingTo  = '';

  /**
   * Sets the hostname to use in Message-Id and Received headers
   * and as default HELO string. If empty, the value returned
   * by SERVER_NAME is used or 'localhost.localdomain'.
   * @var string
   */
  public $Hostname          = '';

  /**
   * Sets the message ID to be used in the Message-Id header.
   * If empty, a unique id will be generated.
   * @var string
   */
  public $MessageID         = '';

  /////////////////////////////////////////////////
  // PROPERTIES FOR SMTP
  /////////////////////////////////////////////////

  /**
   * Sets the SMTP hosts.  All hosts must be separated by a
   * semicolon.  You can also specify a different port
   * for each host by using this format: [hostname:port]
   * (e.g. "smtp1.example.com:25;smtp2.example.com").
   * Hosts will be tried in order.
   * @var string
   */
  public $Host          = 'localhost';

  /**
   * Sets the default SMTP server port.
   * @var int
   */
  public $Port          = 25;

  /**
   * Sets the SMTP HELO of the message (Default is $Hostname).
   * @var string
   */
  public $Helo          = '';

  /**
   * Sets connection prefix.
   * Options are "", "ssl" or "tls"
   * @var string
   */
  public $SMTPSecure    = '';

  /**
   * Sets SMTP authentication. Utilizes the Username and Password variables.
   * @var bool
   */
  public $SMTPAuth      = false;

  /**
   * Sets SMTP username.
   * @var string
   */
```

```
  public $Username      = '';

  /**
   * Sets SMTP password.
   * @var string
   */
  public $Password      = '';

  /**
   * Sets the SMTP server timeout in seconds.
   * This function will not work with the win32 version.
   * @var int
   */
  public $Timeout       = 10;

  /**
   * Sets SMTP class debugging on or off.
   * @var bool
   */
  public $SMTPDebug     = false;

  /**
   * Prevents the SMTP connection from being closed after each mail
   * sending.  If this is set to true then to close the connection
   * requires an explicit call to SmtpClose().
   * @var bool
   */
  public $SMTPKeepAlive = false;

  /**
   * Provides the ability to have the TO field process individual
   * emails, instead of sending to entire TO addresses
   * @var bool
   */
  public $SingleTo      = false;

   /**
   * If SingleTo is true, this provides the array to hold the email addresses
   * @var bool
   */
  public $SingleToArray = array();

 /**
   * Provides the ability to change the line ending
   * @var string
   */
  public $LE              = "\n";

  /**
   * Used with DKIM DNS Resource Record
   * @var string
   */
  public $DKIM_selector   = 'phpmailer';

  /**
   * Used with DKIM DNS Resource Record
   * optional, in format of email address 'you@yourdomain.com'
   * @var string
   */
  public $DKIM_identity   = '';

  /**
   * Used with DKIM DNS Resource Record
   * optional, in format of email address 'you@yourdomain.com'
   * @var string
   */
  public $DKIM_domain     = '';

  /**
   * Used with DKIM DNS Resource Record
   * optional, in format of email address 'you@yourdomain.com'
   * @var string
```

```
 */
public $DKIM_private    = '';

/**
 * Callback Action function name
 * the function that handles the result of the send email action. Parameters:
 *   bool    $result        result of the send action
 *   string  $to            email address of the recipient
 *   string  $cc            cc email addresses
 *   string  $bcc           bcc email addresses
 *   string  $subject       the subject
 *   string  $body          the email body
 * @var string
 */
public $action_function = ''; //'callbackAction';

/**
 * Sets the PHPMailer Version number
 * @var string
 */
public $Version         = '5.1';

////////////////////////////////////////////////
// PROPERTIES, PRIVATE AND PROTECTED
////////////////////////////////////////////////

private   $smtp           = NULL;
private   $to             = array();
private   $cc             = array();
private   $bcc            = array();
private   $ReplyTo        = array();
private   $all_recipients = array();
private   $attachment     = array();
private   $CustomHeader   = array();
private   $message_type   = '';
private   $boundary       = array();
protected $language       = array();
private   $error_count    = 0;
private   $sign_cert_file = "";
private   $sign_key_file  = "";
private   $sign_key_pass  = "";
private   $exceptions     = false;

////////////////////////////////////////////////
// CONSTANTS
////////////////////////////////////////////////

const STOP_MESSAGE  = 0; // message only, continue processing
const STOP_CONTINUE = 1; // message?, likely ok to continue processing
const STOP_CRITICAL = 2; // message, plus full stop, critical error reached

////////////////////////////////////////////////
// METHODS, VARIABLES
////////////////////////////////////////////////

/**
 * Constructor
 * @param boolean $exceptions Should we throw external exceptions?
 */
public function __construct($exceptions = false) {
  $this->exceptions = ($exceptions == true);
}

/**
 * Sets message type to HTML.
 * @param bool $ishtml
 * @return void
 */
public function IsHTML($ishtml = true) {
  if ($ishtml) {
    $this->ContentType = 'text/html';
  } else {
```

```
    $this->ContentType = 'text/plain';
  }
}

/**
 * Sets Mailer to send message using SMTP.
 * @return void
 */
public function IsSMTP() {
  $this->Mailer = 'smtp';
}

/**
 * Sets Mailer to send message using PHP mail() function.
 * @return void
 */
public function IsMail() {
  $this->Mailer = 'mail';
}

/**
 * Sets Mailer to send message using the $Sendmail program.
 * @return void
 */
public function IsSendmail() {
  if (!stristr(ini_get('sendmail_path'), 'sendmail')) {
    $this->Sendmail = '/var/qmail/bin/sendmail';
  }
  $this->Mailer = 'sendmail';
}

/**
 * Sets Mailer to send message using the qmail MTA.
 * @return void
 */
public function IsQmail() {
  if (stristr(ini_get('sendmail_path'), 'qmail')) {
    $this->Sendmail = '/var/qmail/bin/sendmail';
  }
  $this->Mailer = 'sendmail';
}

/////////////////////////////////////////////////
// METHODS, RECIPIENTS
/////////////////////////////////////////////////

/**
 * Adds a "To" address.
 * @param string $address
 * @param string $name
 * @return boolean true on success, false if address already used
 */
public function AddAddress($address, $name = '') {
  return $this->AddAnAddress('to', $address, $name);
}

/**
 * Adds a "Cc" address.
 * Note: this function works with the SMTP mailer on win32, not with the "mail" mailer.
 * @param string $address
 * @param string $name
 * @return boolean true on success, false if address already used
 */
public function AddCC($address, $name = '') {
  return $this->AddAnAddress('cc', $address, $name);
}

/**
 * Adds a "Bcc" address.
 * Note: this function works with the SMTP mailer on win32, not with the "mail" mailer.
 * @param string $address
 * @param string $name
```

```
   * @return boolean true on success, false if address already used
   */
  public function AddBCC($address, $name = '') {
    return $this->AddAnAddress('bcc', $address, $name);
  }

  /**
   * Adds a "Reply-to" address.
   * @param string $address
   * @param string $name
   * @return boolean
   */
  public function AddReplyTo($address, $name = '') {
    return $this->AddAnAddress('ReplyTo', $address, $name);
  }

  /**
   * Adds an address to one of the recipient arrays
   * Addresses that have been added already return false, but do not throw exceptions
   * @param string $kind One of 'to', 'cc', 'bcc', 'ReplyTo'
   * @param string $address The email address to send to
   * @param string $name
   * @return boolean true on success, false if address already used or invalid in some way
   * @access private
   */
  private function AddAnAddress($kind, $address, $name = '') {
    if (!preg_match('/^(to|cc|bcc|ReplyTo)$/', $kind)) {
      echo 'Invalid recipient array: ' . kind;
      return false;
    }
    $address = trim($address);
    $name = trim(preg_replace('/[\r\n]+/', '', $name)); //Strip breaks and trim
    if (!self::ValidateAddress($address)) {
      $this->SetError($this->Lang('invalid_address').': '. $address);
      if ($this->exceptions) {
        throw new phpmailerException($this->Lang('invalid_address').': '.$address);
      }
      echo $this->Lang('invalid_address').': '.$address;
      return false;
    }
    if ($kind != 'ReplyTo') {
      if (!isset($this->all_recipients[strtolower($address)])) {
        array_push($this->$kind, array($address, $name));
        $this->all_recipients[strtolower($address)] = true;
        return true;
      }
    } else {
      if (!array_key_exists(strtolower($address), $this->ReplyTo)) {
        $this->ReplyTo[strtolower($address)] = array($address, $name);
      return true;
    }
  }
  return false;
}

/**
 * Set the From and FromName properties
 * @param string $address
 * @param string $name
 * @return boolean
 */
  public function SetFrom($address, $name = '',$auto=1) {
    $address = trim($address);
    $name = trim(preg_replace('/[\r\n]+/', '', $name)); //Strip breaks and trim
    if (!self::ValidateAddress($address)) {
      $this->SetError($this->Lang('invalid_address').': '. $address);
      if ($this->exceptions) {
        throw new phpmailerException($this->Lang('invalid_address').': '.$address);
      }
      echo $this->Lang('invalid_address').': '.$address;
      return false;
    }
```

```
    $this->From = $address;
    $this->FromName = $name;
    if ($auto) {
      if (empty($this->ReplyTo)) {
        $this->AddAnAddress('ReplyTo', $address, $name);
      }
      if (empty($this->Sender)) {
        $this->Sender = $address;
      }
    }
    return true;
  }

  /**
   * Check that a string looks roughly like an email address should
   * Static so it can be used without instantiation
   * Tries to use PHP built-in validator in the filter extension (from PHP 5.2), falls back to a
reasonably competent regex validator
   * Conforms approximately to RFC2822
   * @link http://www.hexillion.com/samples/#Regex Original pattern found here
   * @param string $address The email address to check
   * @return boolean
   * @static
   * @access public
   */
  public static function ValidateAddress($address) {
    if (function_exists('filter_var')) { //Introduced in PHP 5.2
      if(filter_var($address, FILTER_VALIDATE_EMAIL) === FALSE) {
        return false;
      } else {
        return true;
      }
    } else {
      return preg_match('/^(?:[\w\!\#\$\%\&\'\*\+\-\/\=\?\^\`\{\|\}\~]+\.)*[\w\!\#\$\%\&\'\*\+\-\/\=\?
\^\`\{\|\}\~]+@(?:(?:(?:[a-zA-Z0-9_](?:[a-zA-Z0-9_\-](?!\.)){0,61}[a-zA-Z0-9_-]?\.)+[a-zA-Z0-9_](?:[a-
zA-Z0-9_\-](?!$)){0,61}[a-zA-Z0-9_]?)|(?:\[(?:(?:[01]?\d{1,2}|2[0-4]\d|25[0-5])\.){3}(?:[01]?\d{1,2}|2
[0-4]\d|25[0-5])\]))$/', $address);
    }
  }

  /////////////////////////////////////////////////
  // METHODS, MAIL SENDING
  /////////////////////////////////////////////////

  /**
   * Creates message and assigns Mailer. If the message is
   * not sent successfully then it returns false.  Use the ErrorInfo
   * variable to view description of the error.
   * @return bool
   */
  public function Send() {
    try {
      if ((count($this->to) + count($this->cc) + count($this->bcc)) < 1) {
        throw new phpmailerException($this->Lang('provide_address'), self::STOP_CRITICAL);
      }

      // Set whether the message is multipart/alternative
      if(!empty($this->AltBody)) {
        $this->ContentType = 'multipart/alternative';
      }

      $this->error_count = 0; // reset errors
      $this->SetMessageType();
      $header = $this->CreateHeader();
      $body = $this->CreateBody();

      if (empty($this->Body)) {
        throw new phpmailerException($this->Lang('empty_message'), self::STOP_CRITICAL);
      }

      // digitally sign with DKIM if enabled
      if ($this->DKIM_domain && $this->DKIM_private) {
```

```
        $header_dkim = $this->DKIM_Add($header,$this->Subject,$body);
        $header = str_replace("\r\n","\n",$header_dkim) . $header;
      }

      // Choose the mailer and send through it
      switch($this->Mailer) {
        case 'sendmail':
          return $this->SendmailSend($header, $body);
        case 'smtp':
          return $this->SmtpSend($header, $body);
        default:
          return $this->MailSend($header, $body);
      }

    } catch (phpmailerException $e) {
      $this->SetError($e->getMessage());
      if ($this->exceptions) {
        throw $e;
      }
      echo $e->getMessage()."\n";
      return false;
    }
  }

  /**
   * Sends mail using the $Sendmail program.
   * @param string $header The message headers
   * @param string $body The message body
   * @access protected
   * @return bool
   */
  protected function SendmailSend($header, $body) {
    if ($this->Sender != '') {
      $sendmail = sprintf("%s -oi -f %s -t", escapeshellcmd($this->Sendmail), escapeshellarg($this-
>Sender));
    } else {
      $sendmail = sprintf("%s -oi -t", escapeshellcmd($this->Sendmail));
    }
    if ($this->SingleTo === true) {
      foreach ($this->SingleToArray as $key => $val) {
        if(!@$mail = popen($sendmail, 'w')) {
          throw new phpmailerException($this->Lang('execute') . $this->Sendmail, self::STOP_CRITICAL);
        }
        fputs($mail, "To: " . $val . "\n");
        fputs($mail, $header);
        fputs($mail, $body);
        $result = pclose($mail);
        // implement call back function if it exists
        $isSent = ($result == 0) ? 1 : 0;
        $this->doCallback($isSent,$val,$this->cc,$this->bcc,$this->Subject,$body);
        if($result != 0) {
          throw new phpmailerException($this->Lang('execute') . $this->Sendmail, self::STOP_CRITICAL);
        }
      }
    } else {
      if(!@$mail = popen($sendmail, 'w')) {
        throw new phpmailerException($this->Lang('execute') . $this->Sendmail, self::STOP_CRITICAL);
      }
      fputs($mail, $header);
      fputs($mail, $body);
      $result = pclose($mail);
      // implement call back function if it exists
      $isSent = ($result == 0) ? 1 : 0;
      $this->doCallback($isSent,$this->to,$this->cc,$this->bcc,$this->Subject,$body);
      if($result != 0) {
        throw new phpmailerException($this->Lang('execute') . $this->Sendmail, self::STOP_CRITICAL);
      }
    }
    return true;
  }

  /**
```

```
   * Sends mail using the PHP mail() function.
   * @param string $header The message headers
   * @param string $body The message body
   * @access protected
   * @return bool
   */
  protected function MailSend($header, $body) {
    $toArr = array();
    foreach($this->to as $t) {
      $toArr[] = $this->AddrFormat($t);
    }
    $to = implode(', ', $toArr);

    $params = sprintf("-oi -f %s", $this->Sender);
    if ($this->Sender != '' && strlen(ini_get('safe_mode'))< 1) {
      $old_from = ini_get('sendmail_from');
      ini_set('sendmail_from', $this->Sender);
      if ($this->SingleTo === true && count($toArr) > 1) {
        foreach ($toArr as $key => $val) {
          $rt = @mail($val, $this->EncodeHeader($this->SecureHeader($this->Subject)), $body, $header,
$params);
          // implement call back function if it exists
          $isSent = ($rt == 1) ? 1 : 0;
          $this->doCallback($isSent,$val,$this->cc,$this->bcc,$this->Subject,$body);
        }
      } else {
        $rt = @mail($to, $this->EncodeHeader($this->SecureHeader($this->Subject)), $body, $header,
$params);
        // implement call back function if it exists
        $isSent = ($rt == 1) ? 1 : 0;
        $this->doCallback($isSent,$to,$this->cc,$this->bcc,$this->Subject,$body);
      }
    } else {
      if ($this->SingleTo === true && count($toArr) > 1) {
        foreach ($toArr as $key => $val) {
          $rt = @mail($val, $this->EncodeHeader($this->SecureHeader($this->Subject)), $body, $header,
$params);
          // implement call back function if it exists
          $isSent = ($rt == 1) ? 1 : 0;
          $this->doCallback($isSent,$val,$this->cc,$this->bcc,$this->Subject,$body);
        }
      } else {
        $rt = @mail($to, $this->EncodeHeader($this->SecureHeader($this->Subject)), $body, $header);
        // implement call back function if it exists
        $isSent = ($rt == 1) ? 1 : 0;
        $this->doCallback($isSent,$to,$this->cc,$this->bcc,$this->Subject,$body);
      }
    }
    if (isset($old_from)) {
      ini_set('sendmail_from', $old_from);
    }
    if(!$rt) {
      throw new phpmailerException($this->Lang('instantiate'), self::STOP_CRITICAL);
    }
    return true;
  }

  /**
   * Sends mail via SMTP using PhpSMTP
   * Returns false if there is a bad MAIL FROM, RCPT, or DATA input.
   * @param string $header The message headers
   * @param string $body The message body
   * @uses SMTP
   * @access protected
   * @return bool
   */
  protected function SmtpSend($header, $body) {
    require_once $this->PluginDir . 'class.smtp.php';
    $bad_rcpt = array();

    if(!$this->SmtpConnect()) {
      throw new phpmailerException($this->Lang('smtp_connect_failed'), self::STOP_CRITICAL);
```

```
    }
    $smtp_from = ($this->Sender == '') ? $this->From : $this->Sender;
    if(!$this->smtp->Mail($smtp_from)) {
      throw new phpmailerException($this->Lang('from_failed') . $smtp_from, self::STOP_CRITICAL);
    }

    // Attempt to send attach all recipients
    foreach($this->to as $to) {
      if (!$this->smtp->Recipient($to[0])) {
        $bad_rcpt[] = $to[0];
        // implement call back function if it exists
        $isSent = 0;
        $this->doCallback($isSent,$to[0],'','',$this->Subject,$body);
      } else {
        // implement call back function if it exists
        $isSent = 1;
        $this->doCallback($isSent,$to[0],'','',$this->Subject,$body);
      }
    }
    foreach($this->cc as $cc) {
      if (!$this->smtp->Recipient($cc[0])) {
        $bad_rcpt[] = $cc[0];
        // implement call back function if it exists
        $isSent = 0;
        $this->doCallback($isSent,'',$cc[0],'',$this->Subject,$body);
      } else {
        // implement call back function if it exists
        $isSent = 1;
        $this->doCallback($isSent,'',$cc[0],'',$this->Subject,$body);
      }
    }
    foreach($this->bcc as $bcc) {
      if (!$this->smtp->Recipient($bcc[0])) {
        $bad_rcpt[] = $bcc[0];
        // implement call back function if it exists
        $isSent = 0;
        $this->doCallback($isSent,'','',$bcc[0],$this->Subject,$body);
      } else {
        // implement call back function if it exists
        $isSent = 1;
        $this->doCallback($isSent,'','',$bcc[0],$this->Subject,$body);
      }
    }


    if (count($bad_rcpt) > 0 ) { //Create error message for any bad addresses
      $badaddresses = implode(', ', $bad_rcpt);
      throw new phpmailerException($this->Lang('recipients_failed') . $badaddresses);
    }
    if(!$this->smtp->Data($header . $body)) {
      throw new phpmailerException($this->Lang('data_not_accepted'), self::STOP_CRITICAL);
    }
    if($this->SMTPKeepAlive == true) {
      $this->smtp->Reset();
    }
    return true;
  }

  /**
   * Initiates a connection to an SMTP server.
   * Returns false if the operation failed.
   * @uses SMTP
   * @access public
   * @return bool
   */
  public function SmtpConnect() {
    if(is_null($this->smtp)) {
      $this->smtp = new SMTP();
    }

    $this->smtp->do_debug = $this->SMTPDebug;
    $hosts = explode(';', $this->Host);
```

```
    $index = 0;
    $connection = $this->smtp->Connected();

    // Retry while there is no connection
    try {
      while($index < count($hosts) && !$connection) {
        $hostinfo = array();
        if (preg_match('/^(.+):([0-9]+)$/', $hosts[$index], $hostinfo)) {
          $host = $hostinfo[1];
          $port = $hostinfo[2];
        } else {
          $host = $hosts[$index];
          $port = $this->Port;
        }

        $tls = ($this->SMTPSecure == 'tls');
        $ssl = ($this->SMTPSecure == 'ssl');

        if ($this->smtp->Connect(($ssl ? 'ssl://':'').$host, $port, $this->Timeout)) {

          $hello = ($this->Helo != '' ? $this->Helo : $this->ServerHostname());
          $this->smtp->Hello($hello);

          if ($tls) {
            if (!$this->smtp->StartTLS()) {
              throw new phpmailerException($this->Lang('tls'));
            }

            //We must resend HELO after tls negotiation
            $this->smtp->Hello($hello);
          }

          $connection = true;
          if ($this->SMTPAuth) {
            if (!$this->smtp->Authenticate($this->Username, $this->Password)) {
              throw new phpmailerException($this->Lang('authenticate'));
            }
          }
        }
        $index++;
        if (!$connection) {
          throw new phpmailerException($this->Lang('connect_host'));
        }
      }
    } catch (phpmailerException $e) {
      $this->smtp->Reset();
      throw $e;
    }
    return true;
  }

  /**
   * Closes the active SMTP session if one exists.
   * @return void
   */
  public function SmtpClose() {
    if(!is_null($this->smtp)) {
      if($this->smtp->Connected()) {
        $this->smtp->Quit();
        $this->smtp->Close();
      }
    }
  }

  /**
  * Sets the language for all class error messages.
  * Returns false if it cannot load the language file.  The default language is English.
  * @param string $langcode ISO 639-1 2-character language code (e.g. Portuguese: "br")
  * @param string $lang_path Path to the language file directory
  * @access public
  */
  function SetLanguage($langcode = 'en', $lang_path = 'language/') {
```

```
    //Define full set of translatable strings
    $PHPMAILER_LANG = array(
      'provide_address' => 'You must provide at least one recipient email address.',
      'mailer_not_supported' => ' mailer is not supported.',
      'execute' => 'Could not execute: ',
      'instantiate' => 'Could not instantiate mail function.',
      'authenticate' => 'SMTP Error: Could not authenticate.',
      'from_failed' => 'The following From address failed: ',
      'recipients_failed' => 'SMTP Error: The following recipients failed: ',
      'data_not_accepted' => 'SMTP Error: Data not accepted.',
      'connect_host' => 'SMTP Error: Could not connect to SMTP host.',
      'file_access' => 'Could not access file: ',
      'file_open' => 'File Error: Could not open file: ',
      'encoding' => 'Unknown encoding: ',
      'signing' => 'Signing Error: ',
      'smtp_error' => 'SMTP server error: ',
      'empty_message' => 'Message body empty',
      'invalid_address' => 'Invalid address',
      'variable_set' => 'Cannot set or reset variable: '
    );
    //Overwrite language-specific strings. This way we'll never have missing translations - no more
"language string failed to load"!
    $l = true;
    if ($langcode != 'en') { //There is no English translation file
      $l = @include $lang_path.'phpmailer.lang-'.$langcode.'.php';
    }
    $this->language = $PHPMAILER_LANG;
    return ($l == true); //Returns false if language not found
  }

  /**
  * Return the current array of language strings
  * @return array
  */
  public function GetTranslations() {
    return $this->language;
  }

  ////////////////////////////////////////////////
  // METHODS, MESSAGE CREATION
  ////////////////////////////////////////////////

  /**
   * Creates recipient headers.
   * @access public
   * @return string
   */
  public function AddrAppend($type, $addr) {
    $addr_str = $type . ': ';
    $addresses = array();
    foreach ($addr as $a) {
      $addresses[] = $this->AddrFormat($a);
    }
    $addr_str .= implode(', ', $addresses);
    $addr_str .= $this->LE;

    return $addr_str;
  }

  /**
   * Formats an address correctly.
   * @access public
   * @return string
   */
  public function AddrFormat($addr) {
    if (empty($addr[1])) {
      return $this->SecureHeader($addr[0]);
    } else {
      return $this->EncodeHeader($this->SecureHeader($addr[1]), 'phrase') . " <" . $this->SecureHeader
($addr[0]) . ">";
    }
  }
```

```php
  /**
   * Wraps message for use with mailers that do not
   * automatically perform wrapping and for quoted-printable.
   * Original written by philippe.
   * @param string $message The message to wrap
   * @param integer $length The line length to wrap to
   * @param boolean $qp_mode Whether to run in Quoted-Printable mode
   * @access public
   * @return string
   */
  public function WrapText($message, $length, $qp_mode = false) {
    $soft_break = ($qp_mode) ? sprintf(" =%s", $this->LE) : $this->LE;
    // If utf-8 encoding is used, we will need to make sure we don't
    // split multibyte characters when we wrap
    $is_utf8 = (strtolower($this->CharSet) == "utf-8");

    $message = $this->FixEOL($message);
    if (substr($message, -1) == $this->LE) {
      $message = substr($message, 0, -1);
    }

    $line = explode($this->LE, $message);
    $message = '';
    for ($i=0 ;$i < count($line); $i++) {
      $line_part = explode(' ', $line[$i]);
      $buf = '';
      for ($e = 0; $e<count($line_part); $e++) {
        $word = $line_part[$e];
        if ($qp_mode and (strlen($word) > $length)) {
          $space_left = $length - strlen($buf) - 1;
          if ($e != 0) {
            if ($space_left > 20) {
              $len = $space_left;
              if ($is_utf8) {
                $len = $this->UTF8CharBoundary($word, $len);
              } elseif (substr($word, $len - 1, 1) == "=") {
                $len--;
              } elseif (substr($word, $len - 2, 1) == "=") {
                $len -= 2;
              }
              $part = substr($word, 0, $len);
              $word = substr($word, $len);
              $buf .= ' ' . $part;
              $message .= $buf . sprintf("=%s", $this->LE);
            } else {
              $message .= $buf . $soft_break;
            }
            $buf = '';
          }
          while (strlen($word) > 0) {
            $len = $length;
            if ($is_utf8) {
              $len = $this->UTF8CharBoundary($word, $len);
            } elseif (substr($word, $len - 1, 1) == "=") {
              $len--;
            } elseif (substr($word, $len - 2, 1) == "=") {
              $len -= 2;
            }
            $part = substr($word, 0, $len);
            $word = substr($word, $len);

            if (strlen($word) > 0) {
              $message .= $part . sprintf("=%s", $this->LE);
            } else {
              $buf = $part;
            }
          }
        } else {
          $buf_o = $buf;
          $buf .= ($e == 0) ? $word : (' ' . $word);
```

```
        if (strlen($buf) > $length and $buf_o != '') {
          $message .= $buf_o . $soft_break;
          $buf = $word;
        }
      }
    }
    $message .= $buf . $this->LE;
  }

  return $message;
}

/**
 * Finds last character boundary prior to maxLength in a utf-8
 * quoted (printable) encoded string.
 * Original written by Colin Brown.
 * @access public
 * @param string $encodedText utf-8 QP text
 * @param int    $maxLength   find last character boundary prior to this length
 * @return int
 */
public function UTF8CharBoundary($encodedText, $maxLength) {
  $foundSplitPos = false;
  $lookBack = 3;
  while (!$foundSplitPos) {
    $lastChunk = substr($encodedText, $maxLength - $lookBack, $lookBack);
    $encodedCharPos = strpos($lastChunk, "=");
    if ($encodedCharPos !== false) {
      // Found start of encoded character byte within $lookBack block.
      // Check the encoded byte value (the 2 chars after the '=')
      $hex = substr($encodedText, $maxLength - $lookBack + $encodedCharPos + 1, 2);
      $dec = hexdec($hex);
      if ($dec < 128) { // Single byte character.
        // If the encoded char was found at pos 0, it will fit
        // otherwise reduce maxLength to start of the encoded char
        $maxLength = ($encodedCharPos == 0) ? $maxLength :
        $maxLength - ($lookBack - $encodedCharPos);
        $foundSplitPos = true;
      } elseif ($dec >= 192) { // First byte of a multi byte character
        // Reduce maxLength to split at start of character
        $maxLength = $maxLength - ($lookBack - $encodedCharPos);
        $foundSplitPos = true;
      } elseif ($dec < 192) { // Middle byte of a multi byte character, look further back
        $lookBack += 3;
      }
    } else {
      // No encoded character found
      $foundSplitPos = true;
    }
  }
  return $maxLength;
}


/**
 * Set the body wrapping.
 * @access public
 * @return void
 */
public function SetWordWrap() {
  if($this->WordWrap < 1) {
    return;
  }

  switch($this->message_type) {
    case 'alt':
    case 'alt_attachments':
      $this->AltBody = $this->WrapText($this->AltBody, $this->WordWrap);
      break;
    default:
      $this->Body = $this->WrapText($this->Body, $this->WordWrap);
      break;
```

```
    }
  }

  /**
   * Assembles message header.
   * @access public
   * @return string The assembled header
   */
  public function CreateHeader() {
    $result = '';

    // Set the boundaries
    $uniq_id = md5(uniqid(time()));
    $this->boundary[1] = 'b1_' . $uniq_id;
    $this->boundary[2] = 'b2_' . $uniq_id;

    $result .= $this->HeaderLine('Date', self::RFCDate());
    if($this->Sender == '') {
      $result .= $this->HeaderLine('Return-Path', trim($this->From));
    } else {
      $result .= $this->HeaderLine('Return-Path', trim($this->Sender));
    }

    // To be created automatically by mail()
    if($this->Mailer != 'mail') {
      if ($this->SingleTo === true) {
        foreach($this->to as $t) {
          $this->SingleToArray[] = $this->AddrFormat($t);
        }
      } else {
        if(count($this->to) > 0) {
          $result .= $this->AddrAppend('To', $this->to);
        } elseif (count($this->cc) == 0) {
          $result .= $this->HeaderLine('To', 'undisclosed-recipients:;');
        }
      }
    }

    $from = array();
    $from[0][0] = trim($this->From);
    $from[0][1] = $this->FromName;
    $result .= $this->AddrAppend('From', $from);

    // sendmail and mail() extract Cc from the header before sending
    if(count($this->cc) > 0) {
      $result .= $this->AddrAppend('Cc', $this->cc);
    }

    // sendmail and mail() extract Bcc from the header before sending
    if((($this->Mailer == 'sendmail') || ($this->Mailer == 'mail')) && (count($this->bcc) > 0)) {
      $result .= $this->AddrAppend('Bcc', $this->bcc);
    }

    if(count($this->ReplyTo) > 0) {
      $result .= $this->AddrAppend('Reply-to', $this->ReplyTo);
    }

    // mail() sets the subject itself
    if($this->Mailer != 'mail') {
      $result .= $this->HeaderLine('Subject', $this->EncodeHeader($this->SecureHeader($this-
>Subject)));
    }

    if($this->MessageID != '') {
      $result .= $this->HeaderLine('Message-ID',$this->MessageID);
    } else {
      $result .= sprintf("Message-ID: <%s@%s>%s", $uniq_id, $this->ServerHostname(), $this->LE);
    }
    $result .= $this->HeaderLine('X-Priority', $this->Priority);
    $result .= $this->HeaderLine('X-Mailer', 'PHPMailer '.$this-
>Version.' (phpmailer.sourceforge.net)');
```

```
    if($this->ConfirmReadingTo != '') {
      $result .= $this->HeaderLine('Disposition-Notification-To', '<' . trim($this-
>ConfirmReadingTo) . '>');
    }

    // Add custom headers
    for($index = 0; $index < count($this->CustomHeader); $index++) {
      $result .= $this->HeaderLine(trim($this->CustomHeader[$index][0]), $this->EncodeHeader(trim
($this->CustomHeader[$index][1])));
    }
    if (!$this->sign_key_file) {
      $result .= $this->HeaderLine('MIME-Version', '1.0');
      $result .= $this->GetMailMIME();
    }

    return $result;
  }

  /**
   * Returns the message MIME.
   * @access public
   * @return string
   */
  public function GetMailMIME() {
    $result = '';
    switch($this->message_type) {
      case 'plain':
        $result .= $this->HeaderLine('Content-Transfer-Encoding', $this->Encoding);
        $result .= sprintf("Content-Type: %s; charset=\"%s\"", $this->ContentType, $this->CharSet);
        break;
      case 'attachments':
      case 'alt_attachments':
        if($this->InlineImageExists()){
          $result .= sprintf("Content-Type: %s;%s\ttype=\"text/html\";%s\tboundary=\"%s\"%s",
'multipart/related', $this->LE, $this->LE, $this->boundary[1], $this->LE);
        } else {
          $result .= $this->HeaderLine('Content-Type', 'multipart/mixed;');
          $result .= $this->TextLine("\tboundary=\"" . $this->boundary[1] . '"');
        }
        break;
      case 'alt':
        $result .= $this->HeaderLine('Content-Type', 'multipart/alternative;');
        $result .= $this->TextLine("\tboundary=\"" . $this->boundary[1] . '"');
        break;
    }

    if($this->Mailer != 'mail') {
      $result .= $this->LE.$this->LE;
    }

    return $result;
  }

  /**
   * Assembles the message body.  Returns an empty string on failure.
   * @access public
   * @return string The assembled message body
   */
  public function CreateBody() {
    $body = '';

    if ($this->sign_key_file) {
      $body .= $this->GetMailMIME();
    }

    $this->SetWordWrap();

    switch($this->message_type) {
      case 'alt':
        $body .= $this->GetBoundary($this->boundary[1], '', 'text/plain', '');
        $body .= $this->EncodeString($this->AltBody, $this->Encoding);
        $body .= $this->LE.$this->LE;
```

```
        $body .= $this->GetBoundary($this->boundary[1], '', 'text/html', '');
        $body .= $this->EncodeString($this->Body, $this->Encoding);
        $body .= $this->LE.$this->LE;
        $body .= $this->EndBoundary($this->boundary[1]);
        break;
      case 'plain':
        $body .= $this->EncodeString($this->Body, $this->Encoding);
        break;
      case 'attachments':
        $body .= $this->GetBoundary($this->boundary[1], '', '', '');
        $body .= $this->EncodeString($this->Body, $this->Encoding);
        $body .= $this->LE;
        $body .= $this->AttachAll();
        break;
      case 'alt_attachments':
        $body .= sprintf("--%s%s", $this->boundary[1], $this->LE);
        $body .= sprintf("Content-Type: %s;%s" . "\tboundary=\"%s\"%s", 'multipart/alternative', $this-
>LE, $this->boundary[2], $this->LE.$this->LE);
        $body .= $this->GetBoundary($this->boundary[2], '', 'text/plain', '') . $this->LE; // Create
text body
        $body .= $this->EncodeString($this->AltBody, $this->Encoding);
        $body .= $this->LE.$this->LE;
        $body .= $this->GetBoundary($this->boundary[2], '', 'text/html', '') . $this->LE; // Create
the HTML body
        $body .= $this->EncodeString($this->Body, $this->Encoding);
        $body .= $this->LE.$this->LE;
        $body .= $this->EndBoundary($this->boundary[2]);
        $body .= $this->AttachAll();
        break;
    }

    if ($this->IsError()) {
      $body = '';
    } elseif ($this->sign_key_file) {
      try {
        $file = tempnam('', 'mail');
        file_put_contents($file, $body); //TODO check this worked
        $signed = tempnam("", "signed");
        if (@openssl_pkcs7_sign($file, $signed, "file://".$this->sign_cert_file, array("file://".$this-
>sign_key_file, $this->sign_key_pass), NULL)) {
          @unlink($file);
          @unlink($signed);
          $body = file_get_contents($signed);
        } else {
          @unlink($file);
          @unlink($signed);
          throw new phpmailerException($this->Lang("signing").openssl_error_string());
        }
      } catch (phpmailerException $e) {
        $body = '';
        if ($this->exceptions) {
          throw $e;
        }
      }
    }

    return $body;
  }

  /**
   * Returns the start of a message boundary.
   * @access private
   */
  private function GetBoundary($boundary, $charSet, $contentType, $encoding) {
    $result = '';
    if($charSet == '') {
      $charSet = $this->CharSet;
    }
    if($contentType == '') {
      $contentType = $this->ContentType;
    }
    if($encoding == '') {
```

```
    $encoding = $this->Encoding;
  }
  $result .= $this->TextLine('--' . $boundary);
  $result .= sprintf("Content-Type: %s; charset = \"%s\"", $contentType, $charSet);
  $result .= $this->LE;
  $result .= $this->HeaderLine('Content-Transfer-Encoding', $encoding);
  $result .= $this->LE;

  return $result;
}

/**
 * Returns the end of a message boundary.
 * @access private
 */
private function EndBoundary($boundary) {
  return $this->LE . '--' . $boundary . '--' . $this->LE;
}

/**
 * Sets the message type.
 * @access private
 * @return void
 */
private function SetMessageType() {
  if(count($this->attachment) < 1 && strlen($this->AltBody) < 1) {
    $this->message_type = 'plain';
  } else {
    if(count($this->attachment) > 0) {
      $this->message_type = 'attachments';
    }
    if(strlen($this->AltBody) > 0 && count($this->attachment) < 1) {
      $this->message_type = 'alt';
    }
    if(strlen($this->AltBody) > 0 && count($this->attachment) > 0) {
      $this->message_type = 'alt_attachments';
    }
  }
}

/**
 *  Returns a formatted header line.
 * @access public
 * @return string
 */
public function HeaderLine($name, $value) {
  return $name . ': ' . $value . $this->LE;
}

/**
 * Returns a formatted mail line.
 * @access public
 * @return string
 */
public function TextLine($value) {
  return $value . $this->LE;
}

/////////////////////////////////////////////////
// CLASS METHODS, ATTACHMENTS
/////////////////////////////////////////////////

/**
 * Adds an attachment from a path on the filesystem.
 * Returns false if the file could not be found
 * or accessed.
 * @param string $path Path to the attachment.
 * @param string $name Overrides the attachment name.
 * @param string $encoding File encoding (see $Encoding).
 * @param string $type File extension (MIME) type.
 * @return bool
 */
```

```
  public function AddAttachment($path, $name = '', $encoding = 'base64', $type = 'application/octet-
stream') {
    try {
      if ( !@is_file($path) ) {
        throw new phpmailerException($this->Lang('file_access') . $path, self::STOP_CONTINUE);
      }
      $filename = basename($path);
      if ( $name == '' ) {
        $name = $filename;
      }

      $this->attachment[] = array(
        0 => $path,
        1 => $filename,
        2 => $name,
        3 => $encoding,
        4 => $type,
        5 => false,  // isStringAttachment
        6 => 'attachment',
        7 => 0
      );

    } catch (phpmailerException $e) {
      $this->SetError($e->getMessage());
      if ($this->exceptions) {
        throw $e;
      }
      echo $e->getMessage()."\n";
      if ( $e->getCode() == self::STOP_CRITICAL ) {
        return false;
      }
    }
    return true;
  }

  /**
  * Return the current array of attachments
  * @return array
  */
  public function GetAttachments() {
    return $this->attachment;
  }

  /**
   * Attaches all fs, string, and binary attachments to the message.
   * Returns an empty string on failure.
   * @access private
   * @return string
   */
  private function AttachAll() {
    // Return text of body
    $mime = array();
    $cidUniq = array();
    $incl = array();

    // Add all attachments
    foreach ($this->attachment as $attachment) {
      // Check for string attachment
      $bString = $attachment[5];
      if ($bString) {
        $string = $attachment[0];
      } else {
        $path = $attachment[0];
      }

      if (in_array($attachment[0], $incl)) { continue; }
      $filename    = $attachment[1];
      $name        = $attachment[2];
      $encoding    = $attachment[3];
      $type        = $attachment[4];
      $disposition = $attachment[6];
      $cid         = $attachment[7];
```

```
      $incl[]      = $attachment[0];
      if ( $disposition == 'inline' && isset($cidUniq[$cid]) ) { continue; }
      $cidUniq[$cid] = true;

      $mime[] = sprintf("--%s%s", $this->boundary[1], $this->LE);
      $mime[] = sprintf("Content-Type: %s; name=\"%s\"%s", $type, $this->EncodeHeader($this-
>SecureHeader($name)), $this->LE);
      $mime[] = sprintf("Content-Transfer-Encoding: %s%s", $encoding, $this->LE);

      if($disposition == 'inline') {
        $mime[] = sprintf("Content-ID: <%s>%s", $cid, $this->LE);
      }

      $mime[] = sprintf("Content-Disposition: %s; filename=\"%s\"%s", $disposition, $this->EncodeHeader
($this->SecureHeader($name)), $this->LE.$this->LE);

      // Encode as string attachment
      if($bString) {
        $mime[] = $this->EncodeString($string, $encoding);
        if($this->IsError()) {
          return '';
        }
        $mime[] = $this->LE.$this->LE;
      } else {
        $mime[] = $this->EncodeFile($path, $encoding);
        if($this->IsError()) {
          return '';
        }
        $mime[] = $this->LE.$this->LE;
      }
    }

    $mime[] = sprintf("--%s--%s", $this->boundary[1], $this->LE);

    return join('', $mime);
  }

  /**
   * Encodes attachment in requested format.
   * Returns an empty string on failure.
   * @param string $path The full path to the file
   * @param string $encoding The encoding to use; one of 'base64', '7bit', '8bit', 'binary', 'quoted-
printable'
   * @see EncodeFile()
   * @access private
   * @return string
   */
  private function EncodeFile($path, $encoding = 'base64') {
    try {
      if (!is_readable($path)) {
        throw new phpmailerException($this->Lang('file_open') . $path, self::STOP_CONTINUE);
      }
      if (function_exists('get_magic_quotes')) {
        function get_magic_quotes() {
          return false;
        }
      }
      if (PHP_VERSION < 6) {
        $magic_quotes = get_magic_quotes_runtime();
        set_magic_quotes_runtime(0);
      }
      $file_buffer  = file_get_contents($path);
      $file_buffer  = $this->EncodeString($file_buffer, $encoding);
      if (PHP_VERSION < 6) { set_magic_quotes_runtime($magic_quotes); }
      return $file_buffer;
    } catch (Exception $e) {
      $this->SetError($e->getMessage());
      return '';
    }
  }

  /**
```

```
   * Encodes string to requested format.
   * Returns an empty string on failure.
   * @param string $str The text to encode
   * @param string $encoding The encoding to use; one of 'base64', '7bit', '8bit', 'binary', 'quoted-
printable'
   * @access public
   * @return string
   */
  public function EncodeString ($str, $encoding = 'base64') {
    $encoded = '';
    switch(strtolower($encoding)) {
      case 'base64':
        $encoded = chunk_split(base64_encode($str), 76, $this->LE);
        break;
      case '7bit':
      case '8bit':
        $encoded = $this->FixEOL($str);
        //Make sure it ends with a line break
        if (substr($encoded, -(strlen($this->LE))) != $this->LE)
          $encoded .= $this->LE;
        break;
      case 'binary':
        $encoded = $str;
        break;
      case 'quoted-printable':
        $encoded = $this->EncodeQP($str);
        break;
      default:
        $this->SetError($this->Lang('encoding') . $encoding);
        break;
    }
    return $encoded;
  }

  /**
   * Encode a header string to best (shortest) of Q, B, quoted or none.
   * @access public
   * @return string
   */
  public function EncodeHeader($str, $position = 'text') {
    $x = 0;

    switch (strtolower($position)) {
      case 'phrase':
        if (!preg_match('/[\200-\377]/', $str)) {
          // Can't use addslashes as we don't know what value has magic_quotes_sybase
          $encoded = addcslashes($str, "\0..\37\177\\\"");
          if (($str == $encoded) && !preg_match('/[^A-Za-z0-9!#$%&\'*+\/=?^_`{|}~ -]/', $str)) {
            return ($encoded);
          } else {
            return ("\"$encoded\"");
          }
        }
        $x = preg_match_all('/[^\040\041\043-\133\135-\176]/', $str, $matches);
        break;
      case 'comment':
        $x = preg_match_all('/[()"]/', $str, $matches);
        // Fall-through
      case 'text':
      default:
        $x += preg_match_all('/[\000-\010\013\014\016-\037\177-\377]/', $str, $matches);
        break;
    }

    if ($x == 0) {
      return ($str);
    }

    $maxlen = 75 - 7 - strlen($this->CharSet);
    // Try to select the encoding which should produce the shortest output
    if (strlen($str)/3 < $x) {
      $encoding = 'B';
```

```
      if (function_exists('mb_strlen') && $this->HasMultiBytes($str)) {
        // Use a custom function which correctly encodes and wraps long
        // multibyte strings without breaking lines within a character
        $encoded = $this->Base64EncodeWrapMB($str);
      } else {
        $encoded = base64_encode($str);
        $maxlen -= $maxlen % 4;
        $encoded = trim(chunk_split($encoded, $maxlen, "\n"));
      }
    } else {
      $encoding = 'Q';
      $encoded = $this->EncodeQ($str, $position);
      $encoded = $this->WrapText($encoded, $maxlen, true);
      $encoded = str_replace('='.$this->LE, "\n", trim($encoded));
    }

    $encoded = preg_replace('/^(.*)$/m', " =?".$this->CharSet."?$encoding?\\1?=", $encoded);
    $encoded = trim(str_replace("\n", $this->LE, $encoded));

    return $encoded;
  }

  /**
   * Checks if a string contains multibyte characters.
   * @access public
   * @param string $str multi-byte text to wrap encode
   * @return bool
   */
  public function HasMultiBytes($str) {
    if (function_exists('mb_strlen')) {
      return (strlen($str) > mb_strlen($str, $this->CharSet));
    } else { // Assume no multibytes (we can't handle without mbstring functions anyway)
      return false;
    }
  }

  /**
   * Correctly encodes and wraps long multibyte strings for mail headers
   * without breaking lines within a character.
   * Adapted from a function by paravoid at http://uk.php.net/manual/en/function.mb-encode-
mimeheader.php
   * @access public
   * @param string $str multi-byte text to wrap encode
   * @return string
   */
  public function Base64EncodeWrapMB($str) {
    $start = "=?".$this->CharSet."?B?";
    $end = "?=";
    $encoded = "";

    $mb_length = mb_strlen($str, $this->CharSet);
    // Each line must have length <= 75, including $start and $end
    $length = 75 - strlen($start) - strlen($end);
    // Average multi-byte ratio
    $ratio = $mb_length / strlen($str);
    // Base64 has a 4:3 ratio
    $offset = $avgLength = floor($length * $ratio * .75);

    for ($i = 0; $i < $mb_length; $i += $offset) {
      $lookBack = 0;

      do {
        $offset = $avgLength - $lookBack;
        $chunk = mb_substr($str, $i, $offset, $this->CharSet);
        $chunk = base64_encode($chunk);
        $lookBack++;
      }
      while (strlen($chunk) > $length);

      $encoded .= $chunk . $this->LE;
    }
```

```
    // Chomp the last linefeed
    $encoded = substr($encoded, 0, -strlen($this->LE));
    return $encoded;
  }

  /**
  * Encode string to quoted-printable.
  * Only uses standard PHP, slow, but will always work
  * @access public
  * @param string $string the text to encode
  * @param integer $line_max Number of chars allowed on a line before wrapping
  * @return string
  */
  public function EncodeQPphp( $input = '', $line_max = 76, $space_conv = false) {
    $hex = array('0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F');
    $lines = preg_split('/(?:\r\n|\r|\n)/', $input);
    $eol = "\r\n";
    $escape = '=';
    $output = '';
    while( list(, $line) = each($lines) ) {
      $linlen = strlen($line);
      $newline = '';
      for($i = 0; $i < $linlen; $i++) {
        $c = substr( $line, $i, 1 );
        $dec = ord( $c );
        if ( ( $i == 0 ) && ( $dec == 46 ) ) { // convert first point in the line into =2E
          $c = '=2E';
        }
        if ( $dec == 32 ) {
          if ( $i == ( $linlen - 1 ) ) { // convert space at eol only
            $c = '=20';
          } else if ( $space_conv ) {
            $c = '=20';
          }
        } elseif ( ($dec == 61) || ($dec < 32 ) || ($dec > 126) ) { // always encode "\t", which is
*not* required
          $h2 = floor($dec/16);
          $h1 = floor($dec%16);
          $c = $escape.$hex[$h2].$hex[$h1];
        }
        if ( (strlen($newline) + strlen($c)) >= $line_max ) { // CRLF is not counted
          $output .= $newline.$escape.$eol; //  soft line break; " =\r\n" is okay
          $newline = '';
          // check if newline first character will be point or not
          if ( $dec == 46 ) {
            $c = '=2E';
          }
        }
        $newline .= $c;
      } // end of for
      $output .= $newline.$eol;
    } // end of while
    return $output;
  }

  /**
  * Encode string to RFC2045 (6.7) quoted-printable format
  * Uses a PHP5 stream filter to do the encoding about 64x faster than the old version
  * Also results in same content as you started with after decoding
  * @see EncodeQPphp()
  * @access public
  * @param string $string the text to encode
  * @param integer $line_max Number of chars allowed on a line before wrapping
  * @param boolean $space_conv Dummy param for compatibility with existing EncodeQP function
  * @return string
  * @author Marcus Bointon
  */
  public function EncodeQP($string, $line_max = 76, $space_conv = false) {
    if (function_exists('quoted_printable_encode')) { //Use native function if it's available (>=
PHP5.3)
      return quoted_printable_encode($string);
    }
```

```
    $filters = stream_get_filters();
    if (!in_array('convert.*', $filters)) { //Got convert stream filter?
      return $this->EncodeQPphp($string, $line_max, $space_conv); //Fall back to old implementation
    }
    $fp = fopen('php://temp/', 'r+');
    $string = preg_replace('/\r\n?/', $this->LE, $string); //Normalise line breaks
    $params = array('line-length' => $line_max, 'line-break-chars' => $this->LE);
    $s = stream_filter_append($fp, 'convert.quoted-printable-encode', STREAM_FILTER_READ, $params);
    fputs($fp, $string);
    rewind($fp);
    $out = stream_get_contents($fp);
    stream_filter_remove($s);
    $out = preg_replace('/^\./m', '=2E', $out); //Encode . if it is first char on a line, workaround
for bug in Exchange
    fclose($fp);
    return $out;
  }

  /**
   * Encode string to q encoding.
   * @link http://tools.ietf.org/html/rfc2047
   * @param string $str the text to encode
   * @param string $position Where the text is going to be used, see the RFC for what that means
   * @access public
   * @return string
   */
  public function EncodeQ ($str, $position = 'text') {
    // There should not be any EOL in the string
    $encoded = preg_replace('/[\r\n]*/', '', $str);

    switch (strtolower($position)) {
      case 'phrase':
        $encoded = preg_replace("/([^A-Za-z0-9!*+\/ -])/e", "'='.sprintf('%02X', ord('\\1'))",
$encoded);
        break;
      case 'comment':
        $encoded = preg_replace("/([\(\)\"])/e", "'='.sprintf('%02X', ord('\\1'))", $encoded);
      case 'text':
      default:
        // Replace every high ascii, control =, ? and _ characters
        //TODO using /e (equivalent to eval()) is probably not a good idea
        $encoded = preg_replace('/([\000-\011\013\014\016-\037\075\077\137\177-\377])/e',
              "'='.sprintf('%02X', ord('\\1'))", $encoded);
        break;
    }

    // Replace every spaces to _ (more readable than =20)
    $encoded = str_replace(' ', '_', $encoded);

    return $encoded;
  }

  /**
   * Adds a string or binary attachment (non-filesystem) to the list.
   * This method can be used to attach ascii or binary data,
   * such as a BLOB record from a database.
   * @param string $string String attachment data.
   * @param string $filename Name of the attachment.
   * @param string $encoding File encoding (see $Encoding).
   * @param string $type File extension (MIME) type.
   * @return void
   */
  public function AddStringAttachment($string, $filename, $encoding = 'base64', $type = 'application/
octet-stream') {
    // Append to $attachment array
    $this->attachment[] = array(
      0 => $string,
      1 => $filename,
      2 => basename($filename),
      3 => $encoding,
      4 => $type,
      5 => true,  // isStringAttachment
```

```
      6 => 'attachment',
      7 => 0
    );
  }

  /**
   * Adds an embedded attachment.  This can include images, sounds, and
   * just about any other document.  Make sure to set the $type to an
   * image type.  For JPEG images use "image/jpeg" and for GIF images
   * use "image/gif".
   * @param string $path Path to the attachment.
   * @param string $cid Content ID of the attachment.  Use this to identify
   *        the Id for accessing the image in an HTML form.
   * @param string $name Overrides the attachment name.
   * @param string $encoding File encoding (see $Encoding).
   * @param string $type File extension (MIME) type.
   * @return bool
   */
  public function AddEmbeddedImage($path, $cid, $name = '', $encoding = 'base64', $type = 'application/
octet-stream') {

    if ( !@is_file($path) ) {
      $this->SetError($this->Lang('file_access') . $path);
      return false;
    }

    $filename = basename($path);
    if ( $name == '' ) {
      $name = $filename;
    }

    // Append to $attachment array
    $this->attachment[] = array(
      0 => $path,
      1 => $filename,
      2 => $name,
      3 => $encoding,
      4 => $type,
      5 => false,  // isStringAttachment
      6 => 'inline',
      7 => $cid
    );

    return true;
  }

  /**
   * Returns true if an inline attachment is present.
   * @access public
   * @return bool
   */
  public function InlineImageExists() {
    foreach($this->attachment as $attachment) {
      if ($attachment[6] == 'inline') {
        return true;
      }
    }
    return false;
  }

  ////////////////////////////////////////////////
  // CLASS METHODS, MESSAGE RESET
  ////////////////////////////////////////////////

  /**
   * Clears all recipients assigned in the TO array.  Returns void.
   * @return void
   */
  public function ClearAddresses() {
    foreach($this->to as $to) {
      unset($this->all_recipients[strtolower($to[0])]);
    }
```

```
    $this->to = array();
  }

  /**
   * Clears all recipients assigned in the CC array.  Returns void.
   * @return void
   */
  public function ClearCCs() {
    foreach($this->cc as $cc) {
      unset($this->all_recipients[strtolower($cc[0])]);
    }
    $this->cc = array();
  }

  /**
   * Clears all recipients assigned in the BCC array.  Returns void.
   * @return void
   */
  public function ClearBCCs() {
    foreach($this->bcc as $bcc) {
      unset($this->all_recipients[strtolower($bcc[0])]);
    }
    $this->bcc = array();
  }

  /**
   * Clears all recipients assigned in the ReplyTo array.  Returns void.
   * @return void
   */
  public function ClearReplyTos() {
    $this->ReplyTo = array();
  }

  /**
   * Clears all recipients assigned in the TO, CC and BCC
   * array.  Returns void.
   * @return void
   */
  public function ClearAllRecipients() {
    $this->to = array();
    $this->cc = array();
    $this->bcc = array();
    $this->all_recipients = array();
  }

  /**
   * Clears all previously set filesystem, string, and binary
   * attachments.  Returns void.
   * @return void
   */
  public function ClearAttachments() {
    $this->attachment = array();
  }

  /**
   * Clears all custom headers.  Returns void.
   * @return void
   */
  public function ClearCustomHeaders() {
    $this->CustomHeader = array();
  }

  /////////////////////////////////////////////////
  // CLASS METHODS, MISCELLANEOUS
  /////////////////////////////////////////////////

  /**
   * Adds the error message to the error container.
   * @access protected
   * @return void
   */
  protected function SetError($msg) {
```

```
    $this->error_count++;
    if ($this->Mailer == 'smtp' and !is_null($this->smtp)) {
      $lasterror = $this->smtp->getError();
      if (!empty($lasterror) and array_key_exists('smtp_msg', $lasterror)) {
        $msg .= '<p>' . $this->Lang('smtp_error') . $lasterror['smtp_msg'] . "</p>\n";
      }
    }
    $this->ErrorInfo = $msg;
  }

  /**
   * Returns the proper RFC 822 formatted date.
   * @access public
   * @return string
   * @static
   */
  public static function RFCDate() {
    $tz = date('Z');
    $tzs = ($tz < 0) ? '-' : '+';
    $tz = abs($tz);
    $tz = (int)($tz/3600)*100 + ($tz%3600)/60;
    $result = sprintf("%s %s%04d", date('D, j M Y H:i:s'), $tzs, $tz);

    return $result;
  }

  /**
   * Returns the server hostname or 'localhost.localdomain' if unknown.
   * @access private
   * @return string
   */
  private function ServerHostname() {
    if (!empty($this->Hostname)) {
      $result = $this->Hostname;
    } elseif (isset($_SERVER['SERVER_NAME'])) {
      $result = $_SERVER['SERVER_NAME'];
    } else {
      $result = 'localhost.localdomain';
    }

    return $result;
  }

  /**
   * Returns a message in the appropriate language.
   * @access private
   * @return string
   */
  private function Lang($key) {
    if(count($this->language) < 1) {
      $this->SetLanguage('en'); // set the default language
    }

    if(isset($this->language[$key])) {
      return $this->language[$key];
    } else {
      return 'Language string failed to load: ' . $key;
    }
  }

  /**
   * Returns true if an error occurred.
   * @access public
   * @return bool
   */
  public function IsError() {
    return ($this->error_count > 0);
  }

  /**
   * Changes every end of line from CR or LF to CRLF.
   * @access private
```

```
   * @return string
   */
  private function FixEOL($str) {
    $str = str_replace("\r\n", "\n", $str);
    $str = str_replace("\r", "\n", $str);
    $str = str_replace("\n", $this->LE, $str);
    return $str;
  }

  /**
   * Adds a custom header.
   * @access public
   * @return void
   */
  public function AddCustomHeader($custom_header) {
    $this->CustomHeader[] = explode(':', $custom_header, 2);
  }

  /**
   * Evaluates the message and returns modifications for inline images and backgrounds
   * @access public
   * @return $message
   */
  public function MsgHTML($message, $basedir = '') {
    preg_match_all("/(src|background)=\"(.*)\"/Ui", $message, $images);
    if(isset($images[2])) {
      foreach($images[2] as $i => $url) {
        // do not change urls for absolute images (thanks to corvuscorax)
        if (!preg_match('#^[A-z]+://#',$url)) {
          $filename = basename($url);
          $directory = dirname($url);
          ($directory == '.')?$directory='':'';
          $cid = 'cid:' . md5($filename);
          $ext = pathinfo($filename, PATHINFO_EXTENSION);
          $mimeType  = self::_mime_types($ext);
          if ( strlen($basedir) > 1 && substr($basedir,-1) != '/') { $basedir .= '/'; }
          if ( strlen($directory) > 1 && substr($directory,-1) != '/') { $directory .= '/'; }
          if ( $this->AddEmbeddedImage($basedir.$directory.$filename, md5($filename), $filename,
'base64',$mimeType) ) {
            $message = preg_replace("/".$images[1][$i]."=\"".preg_quote($url, '/')."\"/Ui", $images[1]
[$i]."=\"".$cid."\"", $message);
          }
        }
      }
    }
    $this->IsHTML(true);
    $this->Body = $message;
    $textMsg = trim(strip_tags(preg_replace('/<(head|title|style|script)[^>]*>.*?<\/\\1>/s','',
$message)));
    if (!empty($textMsg) && empty($this->AltBody)) {
      $this->AltBody = html_entity_decode($textMsg);
    }
    if (empty($this->AltBody)) {
      $this->AltBody = 'To view this email message, open it in a program that understands HTML!' . "\n
\n";
    }
  }

  /**
   * Gets the MIME type of the embedded or inline image
   * @param string File extension
   * @access public
   * @return string MIME type of ext
   * @static
   */
  public static function _mime_types($ext = '') {
    $mimes = array(
      'hqx'   =>  'application/mac-binhex40',
      'cpt'   =>  'application/mac-compactpro',
      'doc'   =>  'application/msword',
      'bin'   =>  'application/macbinary',
      'dms'   =>  'application/octet-stream',
```

```
    'lha'   =>  'application/octet-stream',
    'lzh'   =>  'application/octet-stream',
    'exe'   =>  'application/octet-stream',
    'class' =>  'application/octet-stream',
    'psd'   =>  'application/octet-stream',
    'so'    =>  'application/octet-stream',
    'sea'   =>  'application/octet-stream',
    'dll'   =>  'application/octet-stream',
    'oda'   =>  'application/oda',
    'pdf'   =>  'application/pdf',
    'ai'    =>  'application/postscript',
    'eps'   =>  'application/postscript',
    'ps'    =>  'application/postscript',
    'smi'   =>  'application/smil',
    'smil'  =>  'application/smil',
    'mif'   =>  'application/vnd.mif',
    'xls'   =>  'application/vnd.ms-excel',
    'ppt'   =>  'application/vnd.ms-powerpoint',
    'wbxml' =>  'application/vnd.wap.wbxml',
    'wmlc'  =>  'application/vnd.wap.wmlc',
    'dcr'   =>  'application/x-director',
    'dir'   =>  'application/x-director',
    'dxr'   =>  'application/x-director',
    'dvi'   =>  'application/x-dvi',
    'gtar'  =>  'application/x-gtar',
    'php'   =>  'application/x-httpd-php',
    'php4'  =>  'application/x-httpd-php',
    'php3'  =>  'application/x-httpd-php',
    'phtml' =>  'application/x-httpd-php',
    'phps'  =>  'application/x-httpd-php-source',
    'js'    =>  'application/x-javascript',
    'swf'   =>  'application/x-shockwave-flash',
    'sit'   =>  'application/x-stuffit',
    'tar'   =>  'application/x-tar',
    'tgz'   =>  'application/x-tar',
    'xhtml' =>  'application/xhtml+xml',
    'xht'   =>  'application/xhtml+xml',
    'zip'   =>  'application/zip',
    'mid'   =>  'audio/midi',
    'midi'  =>  'audio/midi',
    'mpga'  =>  'audio/mpeg',
    'mp2'   =>  'audio/mpeg',
    'mp3'   =>  'audio/mpeg',
    'aif'   =>  'audio/x-aiff',
    'aiff'  =>  'audio/x-aiff',
    'aifc'  =>  'audio/x-aiff',
    'ram'   =>  'audio/x-pn-realaudio',
    'rm'    =>  'audio/x-pn-realaudio',
    'rpm'   =>  'audio/x-pn-realaudio-plugin',
    'ra'    =>  'audio/x-realaudio',
    'rv'    =>  'video/vnd.rn-realvideo',
    'wav'   =>  'audio/x-wav',
    'bmp'   =>  'image/bmp',
    'gif'   =>  'image/gif',
    'jpeg'  =>  'image/jpeg',
    'jpg'   =>  'image/jpeg',
    'jpe'   =>  'image/jpeg',
    'png'   =>  'image/png',
    'tiff'  =>  'image/tiff',
    'tif'   =>  'image/tiff',
    'css'   =>  'text/css',
    'html'  =>  'text/html',
    'htm'   =>  'text/html',
    'shtml' =>  'text/html',
    'txt'   =>  'text/plain',
    'text'  =>  'text/plain',
    'log'   =>  'text/plain',
    'rtx'   =>  'text/richtext',
    'rtf'   =>  'text/rtf',
    'xml'   =>  'text/xml',
    'xsl'   =>  'text/xml',
    'mpeg'  =>  'video/mpeg',
```

```
    'mpg'   =>  'video/mpeg',
    'mpe'   =>  'video/mpeg',
    'qt'    =>  'video/quicktime',
    'mov'   =>  'video/quicktime',
    'avi'   =>  'video/x-msvideo',
    'movie' =>  'video/x-sgi-movie',
    'doc'   =>  'application/msword',
    'word'  =>  'application/msword',
    'xl'    =>  'application/excel',
    'eml'   =>  'message/rfc822'
  );
  return (!isset($mimes[strtolower($ext)])) ? 'application/octet-stream' : $mimes[strtolower($ext)];
}

/**
* Set (or reset) Class Objects (variables)
*
* Usage Example:
* $page->set('X-Priority', '3');
*
* @access public
* @param string $name Parameter Name
* @param mixed $value Parameter Value
* NOTE: will not work with arrays, there are no arrays to set/reset
* @todo Should this not be using __set() magic function?
*/
public function set($name, $value = '') {
  try {
    if (isset($this->$name) ) {
      $this->$name = $value;
    } else {
      throw new phpmailerException($this->Lang('variable_set') . $name, self::STOP_CRITICAL);
    }
  } catch (Exception $e) {
    $this->SetError($e->getMessage());
    if ($e->getCode() == self::STOP_CRITICAL) {
      return false;
    }
  }
  return true;
}

/**
 * Strips newlines to prevent header injection.
 * @access public
 * @param string $str String
 * @return string
 */
public function SecureHeader($str) {
  $str = str_replace("\r", '', $str);
  $str = str_replace("\n", '', $str);
  return trim($str);
}

/**
 * Set the private key file and password to sign the message.
 *
 * @access public
 * @param string $key_filename Parameter File Name
 * @param string $key_pass Password for private key
 */
public function Sign($cert_filename, $key_filename, $key_pass) {
  $this->sign_cert_file = $cert_filename;
  $this->sign_key_file = $key_filename;
  $this->sign_key_pass = $key_pass;
}

/**
 * Set the private key file and password to sign the message.
 *
 * @access public
 * @param string $key_filename Parameter File Name
```

```
   * @param string $key_pass Password for private key
   */
  public function DKIM_QP($txt) {
    $tmp="";
    $line="";
    for ($i=0;$i<strlen($txt);$i++) {
      $ord=ord($txt[$i]);
      if ( ((0x21 <= $ord) && ($ord <= 0x3A)) || $ord == 0x3C || ((0x3E <= $ord) && ($ord <= 0x7E)) ) {
        $line.=$txt[$i];
      } else {
        $line.="=".sprintf("%02X",$ord);
      }
    }
    return $line;
  }

  /**
   * Generate DKIM signature
   *
   * @access public
   * @param string $s Header
   */
  public function DKIM_Sign($s) {
    $privKeyStr = file_get_contents($this->DKIM_private);
    if ($this->DKIM_passphrase!='') {
      $privKey = openssl_pkey_get_private($privKeyStr,$this->DKIM_passphrase);
    } else {
      $privKey = $privKeyStr;
    }
    if (openssl_sign($s, $signature, $privKey)) {
      return base64_encode($signature);
    }
  }

  /**
   * Generate DKIM Canonicalization Header
   *
   * @access public
   * @param string $s Header
   */
  public function DKIM_HeaderC($s) {
    $s=preg_replace("/\r\n\s+/"," ",$s);
    $lines=explode("\r\n",$s);
    foreach ($lines as $key=>$line) {
      list($heading,$value)=explode(":",$line,2);
      $heading=strtolower($heading);
      $value=preg_replace("/\s+/"," ",$value) ; // Compress useless spaces
      $lines[$key]=$heading.":".trim($value) ; // Don't forget to remove WSP around the value
    }
    $s=implode("\r\n",$lines);
    return $s;
  }

  /**
   * Generate DKIM Canonicalization Body
   *
   * @access public
   * @param string $body Message Body
   */
  public function DKIM_BodyC($body) {
    if ($body == '') return "\r\n";
    // stabilize line endings
    $body=str_replace("\r\n","\n",$body);
    $body=str_replace("\n","\r\n",$body);
    // END stabilize line endings
    while (substr($body,strlen($body)-4,4) == "\r\n\r\n") {
      $body=substr($body,0,strlen($body)-2);
    }
    return $body;
  }

  /**
```

```
   * Create the DKIM header, body, as new header
   *
   * @access public
   * @param string $headers_line Header lines
   * @param string $subject Subject
   * @param string $body Body
   */
  public function DKIM_Add($headers_line,$subject,$body) {
    $DKIMsignatureType    = 'rsa-sha1'; // Signature & hash algorithms
    $DKIMcanonicalization = 'relaxed/simple'; // Canonicalization of header/body
    $DKIMquery            = 'dns/txt'; // Query method
    $DKIMtime             = time() ; // Signature Timestamp = seconds since 00:00:00 - Jan 1, 1970
(UTC time zone)
    $subject_header       = "Subject: $subject";
    $headers              = explode("\r\n",$headers_line);
    foreach($headers as $header) {
      if (strpos($header,'From:') === 0) {
        $from_header=$header;
      } elseif (strpos($header,'To:') === 0) {
        $to_header=$header;
      }
    }
    $from     = str_replace('|','=7C',$this->DKIM_QP($from_header));
    $to       = str_replace('|','=7C',$this->DKIM_QP($to_header));
    $subject  = str_replace('|','=7C',$this->DKIM_QP($subject_header)) ; // Copied header fields (dkim-
quoted-printable
    $body     = $this->DKIM_BodyC($body);
    $DKIMlen  = strlen($body) ; // Length of body
    $DKIMb64  = base64_encode(pack("H*", sha1($body))) ; // Base64 of packed binary SHA-1 hash of body
    $ident    = ($this->DKIM_identity == '')? '' : " i=" . $this->DKIM_identity . ";";
    $dkimhdrs = "DKIM-Signature: v=1; a=" . $DKIMsignatureType . "; q=" . $DKIMquery . "; l=" .
$DKIMlen . "; s=" . $this->DKIM_selector . ";\r\n".
                "\tt=" . $DKIMtime . "; c=" . $DKIMcanonicalization . ";\r\n".
                "\th=From:To:Subject;\r\n".
                "\td=" . $this->DKIM_domain . ";" . $ident . "\r\n".
                "\tz=$from\r\n".
                "\t|$to\r\n".
                "\t|$subject;\r\n".
                "\tbh=" . $DKIMb64 . ";\r\n".
                "\tb=";
    $toSign   = $this->DKIM_HeaderC($from_header . "\r\n" . $to_header . "\r\n" . $subject_header . "\r
\n" . $dkimhdrs);
    $signed   = $this->DKIM_Sign($toSign);
    return "X-PHPMAILER-DKIM: phpmailer.worxware.com\r\n".$dkimhdrs.$signed."\r\n";
  }

  protected function doCallback($isSent,$to,$cc,$bcc,$subject,$body) {
    if (!empty($this->action_function) && function_exists($this->action_function)) {
      $params = array($isSent,$to,$cc,$bcc,$subject,$body);
      call_user_func_array($this->action_function,$params);
    }
  }
}

class phpmailerException extends Exception {
  public function errorMessage() {
    $errorMsg = '<strong>' . $this->getMessage() . "</strong><br />\n";
    return $errorMsg;
  }
}
?>
```

```
<?php

/*

  This code is from http://detectmobilebrowsers.mobi/ - please do not republish it without due credit
and hyperlink to http://detectmobilebrowsers.mobi

  For help generating the function call visit http://detectmobilebrowsers.mobi/ and use the function
generator.

  Published by Andy Moore - .mobi certified mobile web developer - http://andymoore.info/

  This code is free to download and use on non-profit websites, if your website makes a profit or you
require support using this code please upgrade.

  Please pgrade for use on commercial websites http://detectmobilebrowsers.mobi/#upgrade1064282

  To submit a support request please forward your PayPal receipt with your questions to the email
address you sent the money to and I will endeavour to get back to you. It might take me a few days but
I reply to all support issues with as much helpful info as I can provide. Though really everything is
published on the site.

  The function has eight parameters that can be passed to it which define the way it handles different
scenarios. These paramaters are:

    * iPhone - Set to true to treat iPhones as mobiles, false to treat them like full browsers or set
a URL (including http://) to redirect iPhones and iPods to.
    * Android - Set to true to treat Android handsets as mobiles, false to treat them like full
browsers or set a URL (including http://) to redirect Android and Google mobile users to.
    * Opera Mini - Set to true to treat Opera Mini like a mobile, false to treat it like full browser
or set a URL (including http://) to redirect Opera Mini users to.
    * Blackberry - Set to true to treat Blackberry like a mobile, false to treat it like full browser
or set a URL (including http://) to redirect Blackberry users to.
    * Palm - Set to true to treat Palm OS like a mobile, false to treat it like full browser or set a
URL (including http://) to redirect Palm OS users to.
    * Windows - Set to true to treat Windows Mobiles like a mobile, false to treat it like full
browser or set a URL (including http://) to redirect Windows Mobile users to.
    * Mobile Redirect URL - This should be full web address (including http://) of the site (or page)
you want to send mobile visitors to. Leaving this blank will make the script return true when it
detects a mobile.
    * Desktop Redirect URL - This should be full web address (including http://) of the site (or page)
you want to send non-mobile visitors to. Leaving this blank will make the script return false when it
fails to detect a mobile.

  Change Log

    * 25.11.08 - Added Amazon's Kindle to the pipe seperated array
    * 27.11.08 - Added support for Blackberry options
    * 27.01.09 - Added usage samples & help with PHP in HTML
    * 09.03.09 - Added support for Windows Mobile options
    * 09.03.09 - Removed 'ppc;'=>'ppc;', from array to reduce false positives
    * 09.03.09 - Added support for Palm OS options
    * 09.03.09 - Added sample .htaccess html.html and help.html files to download
    * 16.03.09 - Edited sample .htaccess file - now work with GoDaddy

1. Mac OS X Safari Version 4.0 (5530.17) matches on "tosh" due to:
Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_5_7; en-us) AppleWebKit/530.17 (KHTML, like Gecko)
Version/4.0 Safari/530.17

1b. Must add "hiba" to be inclusive of "Toshiba"

2. IE 8.0 Matches on "comp" due to:
Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; Trident/4.0)

2b. Must add "mpal" to support "Compal"

2. Palm Pre is not supported. You need to add "pre/" to your list:
Mozilla/5.0 (webOS/1.0; U; en-US) AppleWebKit/525.27.1 (KHTML, like Gecko) Version/1.0 Safari/525.27.1
Pre/1.0

FYI: "Pre" matches to "Presto", so make sure you use "Pre/":
Opera/9.64 (Macintosh; Intel Mac OS X; U; en) Presto/2.1.1
```

```
3. LG Voyager is not supoorted. You need to add "lge " and/or "vx10" to your list:
Mozilla/4.1 (compatible; MSIE 6.0; ) 400x240 LGE VX10000

^^^^^^^^^^^^^^^^^^^^ MUCH credit to http://www.punchkickinteractive.com/ - thanks Ryan!

EI 8 bug fixed - removed pda from pipe array

*/

function mobile_device_detect($iphone=true,$android=true,$opera=true,$blackberry=true,$palm=true,
$windows=true,$mobileredirect=false,$desktopredirect=false){

  $mobile_browser   = false; // set mobile browser as false till we can prove otherwise
  $user_agent       = $_SERVER['HTTP_USER_AGENT']; // get the user agent value - this should be
cleaned to ensure no nefarious input gets executed
  $accept           = $_SERVER['HTTP_ACCEPT']; // get the content accept value - this should be
cleaned to ensure no nefarious input gets executed

  switch(true){ // using a switch against the following statements which could return true is more
efficient than the previous method of using if statements

    case (eregi('ipod',$user_agent)||eregi('iphone',$user_agent)); // we find the words iphone or ipod
in the user agent
      $mobile_browser = $iphone; // mobile browser is either true or false depending on the setting of
iphone when calling the function
      $status = 'Apple';
      if(substr($iphone,0,4)=='http'){ // does the value of iphone resemble a url
        $mobileredirect = $iphone; // set the mobile redirect url to the url value stored in the
iphone value
      } // ends the if for iphone being a url
    break; // break out and skip the rest if we've had a match on the iphone or ipod

    case (eregi('android',$user_agent));  // we find android in the user agent
      $mobile_browser = $android; // mobile browser is either true or false depending on the setting
of android when calling the function
      $status = 'Android';
      if(substr($android,0,4)=='http'){ // does the value of android resemble a url
        $mobileredirect = $android; // set the mobile redirect url to the url value stored in the
android value
      } // ends the if for android being a url
    break; // break out and skip the rest if we've had a match on android

    case (eregi('opera mini',$user_agent)); // we find opera mini in the user agent
      $mobile_browser = $opera; // mobile browser is either true or false depending on the setting of
opera when calling the function
      $status = 'Opera';
      if(substr($opera,0,4)=='http'){ // does the value of opera resemble a rul
        $mobileredirect = $opera; // set the mobile redirect url to the url value stored in the opera
value
      } // ends the if for opera being a url
    break; // break out and skip the rest if we've had a match on opera

    case (eregi('blackberry',$user_agent)); // we find blackberry in the user agent
      $mobile_browser = $blackberry; // mobile browser is either true or false depending on the
setting of blackberry when calling the function
      $status = 'Blackberry';
      if(substr($blackberry,0,4)=='http'){ // does the value of blackberry resemble a rul
        $mobileredirect = $blackberry; // set the mobile redirect url to the url value stored in the
blackberry value
      } // ends the if for blackberry being a url
    break; // break out and skip the rest if we've had a match on blackberry

    case (preg_match('/(pre\/|palm os|palm|hiptop|avantgo|fennec|plucker|xiino|blazer|elaine)/i',
$user_agent)); // we find palm os in the user agent - the i at the end makes it case insensitive
      $mobile_browser = $palm; // mobile browser is either true or false depending on the setting of
palm when calling the function
      $status = 'Palm';
      if(substr($palm,0,4)=='http'){ // does the value of palm resemble a rul
        $mobileredirect = $palm; // set the mobile redirect url to the url value stored in the palm
value
      } // ends the if for palm being a url
```

```
    break; // break out and skip the rest if we've had a match on palm os

    case (preg_match('/(iris|3g_t|windows ce|opera mobi|windows ce; smartphone;|windows ce; iemobile)/
i',$user_agent)); // we find windows mobile in the user agent - the i at the end makes it case
insensitive
      $mobile_browser = $windows; // mobile browser is either true or false depending on the setting
of windows when calling the function
      $status = 'Windows Smartphone';
      if(substr($windows,0,4)=='http'){ // does the value of windows resemble a rul
        $mobileredirect = $windows; // set the mobile redirect url to the url value stored in the
windows value
      } // ends the if for windows being a url
    break; // break out and skip the rest if we've had a match on windows

    case (preg_match('/(mini 9.5|vx1000|lge |m800|e860|u940|ux840|compal|wireless| mobi|ahong|lg380|
lgku|lgu900|lg210|lg47|lg920|lg840|lg370|sam-r|mg50|s55|g83|t66|vx400|mk99|d615|d763|el370|sl900|mp500|
samu3|samu4|vx10|xda_|samu5|samu6|samu7|samu9|a615|b832|m881|s920|n210|s700|c-810|_h797|mob-x|sk16d|
848b|mowser|s580|r800|471x|v120|rim8|c500foma:|160x|x160|480x|x640|t503|w839|i250|sprint|w398samr810|
m5252|c7100|mt126|x225|s5330|s820|htil-g1|fly v71|s302|-x113|novarra|k610i|-three|8325rc|8352rc|sanyo|
vx54|c888|nx250|n120|mtk |c5588|s710|t880|c5005|i;458x|p404i|s210|c5100|teleca|s940|c500|s590|foma|
samsu|vx8|vx9|a1000|_mms|myx|a700|gu1100|bc831|e300|ems100|me701|me702m-three|sd588|s800|8325rc|ac831|
mw200|brew |d88|htc\/|htc_touch|355x|m50|km100|d736|p-9521|telco|sl74|ktouch|m4u\/|me702|8325rc|kddi|
phone|lg |sonyericsson|samsung|240x|x320vx10|nokia|sony cmd|motorola|up.browser|up.link|mmp|symbian|
smartphone|midp|wap|vodafone|o2|pocket|kindle|mobile|psp|treo)/i',$user_agent)); // check if any of
the values listed create a match on the user agent - these are some of the most common terms used in
agents to identify them as being mobile devices - the i at the end makes it case insensitive
      $mobile_browser = true; // set mobile browser to true
      $status = 'Mobile matched on piped preg_match';
    break; // break out and skip the rest if we've preg_match on the user agent returned true

    case ((strpos($accept,'text/vnd.wap.wml')>0)||(strpos($accept,'application/vnd.wap.xhtml+xml')
>0)); // is the device showing signs of support for text/vnd.wap.wml or application/vnd.wap.xhtml+xml
      $mobile_browser = true; // set mobile browser to true
      $status = 'Mobile matched on content accept header';
    break; // break out and skip the rest if we've had a match on the content accept headers

    case (isset($_SERVER['HTTP_X_WAP_PROFILE'])||isset($_SERVER['HTTP_PROFILE'])); // is the device
giving us a HTTP_X_WAP_PROFILE or HTTP_PROFILE header - only mobile devices would do this
      $mobile_browser = true; // set mobile browser to true
      $status = 'Mobile matched on profile headers being set';
    break; // break out and skip the final step if we've had a return true on the mobile specfic
headers

    case (in_array(strtolower(substr($user_agent,0,4)),array
('1207'=>'1207','3gso'=>'3gso','4thp'=>'4thp','501i'=>'501i','502i'=>'502i','503i'=>'503i','504i'=>'504i','
wa'=>'a
wa','acer'=>'acer','acs-'=>'acs-','airn'=>'airn','alav'=>'alav','asus'=>'asus','attw'=>'attw','au-
m'=>'au-m','aur '=>'aur ','aus '=>'aus
','abac'=>'abac','acoo'=>'acoo','aiko'=>'aiko','alco'=>'alco','alca'=>'alca','amoi'=>'amoi','anex'=>'anex',
n'=>'bw-n','bw-u'=>'bw-
u','beck'=>'beck','benq'=>'benq','bilb'=>'bilb','blac'=>'blac','c55/'=>'c55/','cdm-'=>'cdm-','chtm'=>'chtm'
s'=>'dc-s','dica'=>'dica','ds-d'=>'ds-
d','ds12'=>'ds12','dait'=>'dait','devi'=>'devi','dmob'=>'dmob','doco'=>'doco','dopo'=>'dopo','el49'=>'el49'
mo'=>'g-mo','g1 u'=>'g1
u','g560'=>'g560','gf-5'=>'gf-5','grun'=>'grun','gene'=>'gene','go.w'=>'go.w','good'=>'good','grad'=>'grad'
m'=>'hd-m','hd-p'=>'hd-p','hd-t'=>'hd-t','hei-'=>'hei-','hp i'=>'hp i','hpip'=>'hpip','hs-c'=>'hs-
c','htc '=>'htc
','htc-'=>'htc-','htca'=>'htca','htcg'=>'htcg','htcp'=>'htcp','htcs'=>'htcs','htct'=>'htct','htc_'=>'htc_',
go'=>'i-go','i-ma'=>'i-
ma','i230'=>'i230','iac'=>'iac','iac-'=>'iac-','iac/'=>'iac/','ig01'=>'ig01','im1k'=>'im1k','inno'=>'inno',
'=>'kpt ','kwc-'=>'kwc-','klon'=>'klon','lexi'=>'lexi','lg g'=>'lg g','lg-a'=>'lg-a','lg-b'=>'lg-
b','lg-c'=>'lg-c','lg-d'=>'lg-d','lg-f'=>'lg-f','lg-g'=>'lg-g','lg-k'=>'lg-k','lg-l'=>'lg-l','lg-
m'=>'lg-m','lg-o'=>'lg-o','lg-p'=>'lg-p','lg-s'=>'lg-s','lg-t'=>'lg-t','lg-u'=>'lg-u','lg-w'=>'lg-
w','lg/k'=>'lg/k','lg/l'=>'lg/l','lg/u'=>'lg/
u','lg50'=>'lg50','lg54'=>'lg54','lge-'=>'lge-','lge/'=>'lge/','lynx'=>'lynx','leno'=>'leno','m1-
w'=>'m1-
w','m3ga'=>'m3ga','m50/'=>'m50/','maui'=>'maui','mc01'=>'mc01','mc21'=>'mc21','mcca'=>'mcca','medi'=>'medi'
'=>'mot ','mot-'=>'mot-','mt50'=>'mt50','mtp1'=>'mtp1','mtv '=>'mtv
','mate'=>'mate','maxo'=>'maxo','merc'=>'merc','mits'=>'mits','mobi'=>'mobi','motv'=>'motv','mozz'=>'mozz',
x'=>'o2 x','o2-x'=>'o2-
x','opwv'=>'opwv','owg1'=>'owg1','opti'=>'opti','oran'=>'oran','p800'=>'p800','pand'=>'pand','pg-1'=>'pg-1'
c'=>'pg-c','pg13'=>'pg13','phil'=>'phil','pn-2'=>'pn-2','pt-g'=>'pt-
g','palm'=>'palm','pana'=>'pana','pire'=>'pire','pock'=>'pock','pose'=>'pose','psio'=>'psio','qa-
```

```
a'=>'qa-
a','qc-2'=>'qc-2','qc-3'=>'qc-3','qc-5'=>'qc-5','qc-7'=>'qc-7','qc07'=>'qc07','qc12'=>'qc12','qc21'=>'qc21'
'=>'spv
','spv-'=>'spv-','sy01'=>'sy01','samm'=>'samm','sany'=>'sany','sava'=>'sava','scoo'=>'scoo','send'=>'send',
mo'=>'t-
mo','t218'=>'t218','t250'=>'t250','t600'=>'t600','t610'=>'t610','t618'=>'t618','tcl-'=>'tcl-','tdg-'=>'tdg-
v'=>'vk-
v','vk40'=>'vk40','vk50'=>'vk50','vk52'=>'vk52','vk53'=>'vk53','vm40'=>'vm40','vx98'=>'vx98','virg'=>'virg'
'=>'w3c ','w3c-'=>'w3c-','wapj'=>'wapj','wapp'=>'wapp','wapu'=>'wapu','wapm'=>'wapm','wig '=>'wig
','wapi'=>'wapi','wapr'=>'wapr','wapv'=>'wapv','wapy'=>'wapy','wapa'=>'wapa','waps'=>'waps','wapt'=>'wapt',
c'=>'lg-c','lg-d'=>'lg-d','lg-g'=>'lg-g','lge-'=>'lge-','libw'=>'libw','m-cr'=>'m-
cr','maui'=>'maui','maxo'=>'maxo','midp'=>'midp','mits'=>'mits','mmef'=>'mmef','mobi'=>'mobi','mot-'=>'mot-
mo'=>'t-
mo','teli'=>'teli','tim-'=>'tim-','tosh'=>'tosh','treo'=>'treo','tsm-'=>'tsm-','upg1'=>'upg1','upsi'=>'upsi
v'=>'vk-
v','voda'=>'voda','vx52'=>'vx52','vx53'=>'vx53','vx60'=>'vx60','vx61'=>'vx61','vx70'=>'vx70','vx80'=>'vx80'
check against a list of trimmed user agents to see if we find a match
      $mobile_browser = true; // set mobile browser to true
      $status = 'Mobile matched on in_array';
    break; // break even though it's the last statement in the switch so there's nothing to break away
from but it seems better to include it than exclude it

    default;
      $mobile_browser = false; // set mobile browser to false
      $status = 'Desktop / full capability browser';
    break; // break even though it's the last statement in the switch so there's nothing to break away
from but it seems better to include it than exclude it

  } // ends the switch

  // tell adaptation services (transcoders and proxies) to not alter the content based on user agent
as it's already being managed by this script
//  header('Cache-Control: no-transform'); // http://mobiforge.com/developing/story/setting-http-
headers-advise-transcoding-proxies
//  header('Vary: User-Agent, Accept'); // http://mobiforge.com/developing/story/setting-http-headers-
advise-transcoding-proxies

  // if redirect (either the value of the mobile or desktop redirect depending on the value of
$mobile_browser) is true redirect else we return the status of $mobile_browser
  if($redirect = ($mobile_browser==true) ? $mobileredirect : $desktopredirect){
    header('Location: '.$redirect); // redirect to the right url for this device
    exit;
  }else{
    return $mobile_browser; // will return either true or false
  }

} // ends function mobile_device_detect

?>
```

```php
<?php

define ('S', '/');

if($_SERVER['HTTP_HOST'] == 'localhost'){
    define('BASE', $_SERVER['DOCUMENT_ROOT'].S.'pfc');
    define('HOME', "http://localhost/pfc");
} else {
    define('BASE', $_SERVER['DOCUMENT_ROOT'].S);
    define('HOME', "http://iqualit.com");
}

include BASE.S."app/includes/mobile_device_detect.php";
#mobile_device_detect(true,true,true,true,true,true,false, HOME.S.'index.php?r=error');

//DEFINICIÓN DE CONSTANTES

define('INCLUDES', BASE.S.'app/includes');
define('THEME', BASE.S.'app/theme');

//CARGA DATOS DE LA DB
require_once INCLUDES.S."settings.php";

//CARGA LA CLASE DE LA BASE DE DATOS
require_once INCLUDES.S."class_database.php";

//CARGA FUNCIONES DE LA PLANTILLA
require_once INCLUDES.S."functions_theme.php";

//CARGA FUNCIONES GENERALES
require_once INCLUDES.S."functions_general.php";

//CARGA CLASES DE LOS PROCESOS
require_once INCLUDES.S."class_recepcio.php";
require_once INCLUDES.S."class_execucio.php";
require_once INCLUDES.S."class_acabada.php";

//CARGA CLASE PARA PDF Y ENVIAR POR EMAIL
require_once INCLUDES.S."functions_print_email.php";
require_once INCLUDES.S."pdf/fpdf.php";
require_once INCLUDES.S."email/class.phpmailer.php";

//Primero, si las cookies estan pero sin sesión, se crea una sesión.
check_cookie();
```

```php
<?php

function check_cookie(){
    if($_COOKIE['iQualit_user_id'] && $_COOKIE['iQualit_password']){

        //Antes de crear la sesión, comprueba que los datos de las cookies sean corectos
        $query = mysql_query("SELECT email FROM users WHERE user_id = '".$_COOKIE
['iQualit_user_id']."' AND password = '".$_COOKIE['iQualit_password']."' ");

        if(mysql_num_rows($query)) {
            $_SESSION['user_id'] = $_COOKIE['iQualit_user_id'];
            $_SESSION['logged_in'] = 1;
        }
    }
}

function user_validates($user_id=NULL,$project_id=NULL) {

    $session_id = $_SESSION['user_id'];

    //Si existe el project id que haga un tipo
    if($project_id && $user_id) {

        $db_query = mysql_query("SELECT id FROM project WHERE id='".$project_id."' AND user_id = '".
$user_id."'");

        $data = mysql_fetch_array($db_query);

        if ($data['id'] == $project_id) {
            if ($session_id == $user_id) {
                return 1;
            } else {
                return NULL;
            }
        } else {
            return NULL;
        }

    //Si no existe project id, pero si user_id hace este tipo
    } elseif($user_id) {
        if ($session_id == $user_id) {
            return 1;
        } else {
            return NULL;
        }

    //Si no existe ni uno ni otro es que está mal, pero igualmente
    } else {
        if($_SESSION['logged_in'] == 1) {
            return 1;
        } else {
            return NULL;
        }
    }
}

function enc($value) {
    echo urlencode($value);
}

function checked($proces) {
    if (in_array($proces, $GLOBALS['procesos_marcats'])) {
        echo 'checked';
    } else {
        echo '';
    }
}

/* project_control.php */
function titulo_procesos ($init, $proces) {
switch ($init) {
    case "PQ_01":
```

```
        if (in_array("PQ_0111", $proces) OR
            in_array("PQ_0121", $proces) OR
            in_array("PQ_0122", $proces) OR
            in_array("PQ_0127", $proces) OR
            in_array("PQ_0131", $proces) OR
            in_array("PQ_0141", $proces) OR
            in_array("PQ_0161", $proces) OR
            in_array("PQ_0162", $proces) OR
            in_array("PQ_0171", $proces) OR
            in_array("PQ_0181", $proces) OR
            in_array("PQ_0191", $proces)) {

            echo "<li class='title'>";
            echo "Moviments de terres";
            echo "</li>";
            } break;

    case "PQ_02":
        if (in_array("PQ_021A", $proces) OR
            in_array("PQ_0212", $proces) OR
            in_array("PQ_0213", $proces) OR
            in_array("PQ_0214", $proces) OR
            in_array("PQ_0215", $proces) OR
            in_array("PQ_0216", $proces) OR
            in_array("PQ_0217", $proces) OR
            in_array("PQ_0218", $proces) OR
            in_array("PQ_0219", $proces)) {

            echo "<li class='title'>";
            echo "Fonaments";
            echo "</li>";
            } break;

    case "PQ_03":
        if (in_array("PQ_0311", $proces)) {
            echo "<li class='title'>";
            echo "Fusta";
            echo "</li>";
            } break;

    case "PQ_04":
        if (in_array("PQ_0411", $proces) OR
            in_array("PQ_0413", $proces)) {
            echo "<li class='title'>";
            echo "Formigó";
            echo "</li>";
            } break;

    case "PQ_05":
        if (in_array("PQ_0501", $proces) OR
            in_array("PQ_0511", $proces)) {
            echo "<li class='title'>";
            echo "Acer";
            echo "</li>";
            } break;

    case "PQ_06":
        if (in_array("PQ_06X1", $proces) OR
            in_array("PQ_0612", $proces) OR
            in_array("PQ_0613", $proces) OR
            in_array("PQ_0616", $proces)) {
            echo "<li class='title'>";
            echo "Fabrica";
            echo "</li>";
            } break;

    case "PQ_07":
        if (in_array("PQ_0711", $proces)) {
            echo "<li class='title'>";
            echo "Prefabricada";
            echo "</li>";
            } break;
```

```
    case "PQ_08":
        if (in_array("PQ_08A1", $proces) OR
            in_array("PQ_08B1", $proces) OR
            in_array("PQ_08C1", $proces) OR
            in_array("PQ_0811", $proces) OR
            in_array("PQ_0831", $proces) OR
            in_array("PQ_0841", $proces) OR
            in_array("PQ_0851", $proces) OR
            in_array("PQ_0861", $proces) OR
            in_array("PQ_0871", $proces) OR
            in_array("PQ_0881", $proces) OR
            in_array("PQ_0891", $proces)) {
            echo "<li class='title'>";
            echo "Cobertes";
            echo "</li>";
            } break;

    case "PQ_09":
        if (in_array("PQ_0911", $proces) OR
            in_array("PQ_0921", $proces) OR
            in_array("PQ_0931", $proces) OR
            in_array("PQ_0941", $proces) OR
            in_array("PQ_0961", $proces) OR
            in_array("PQ_0971", $proces) OR
            in_array("PQ_0981", $proces)) {
            echo "<li class='title'>";
            echo "Tancaments exteriors";
            echo "</li>";
            } break;

    case "PQ_10":
        if (in_array("PQ_1021", $proces) OR
            in_array("PQ_1031", $proces)) {
            echo "<li class='title'>";
            echo "Tancaments interiors";
            echo "</li>";
            } break;

    case "PQ_13":
        if (in_array("PQ_13A1", $proces) OR
            in_array("PQ_13B1", $proces) OR
            in_array("PQ_1311", $proces) OR
            in_array("PQ_1321", $proces) OR
            in_array("PQ_1331", $proces) OR
            in_array("PQ_1341", $proces) OR
            in_array("PQ_1351", $proces) OR
            in_array("PQ_1371", $proces) OR
            in_array("PQ_1381", $proces) OR
            in_array("PQ_1391", $proces)) {
            echo "<li class='title'>";
            echo "Acabats";
            echo "</li>";
            } break;

    case "PQ_14":
        if (in_array("PQ_14A1", $proces) OR
            in_array("PQ_14B1", $proces) OR
            in_array("PQ_14C1", $proces) OR
            in_array("PQ_14D1", $proces) OR
            in_array("PQ_14E1", $proces) OR
            in_array("PQ_14F1", $proces) OR
            in_array("PQ_1421", $proces) OR
            in_array("PQ_1431", $proces)) {
            echo "<li class='title'>";
            echo "Paviments";
            echo "</li>";
            } break;

    case "PQ_15":
        if (in_array("PQ_15P1", $proces) OR
            in_array("PQ_1511", $proces) OR
```

```
        in_array("PQ_1531", $proces)) {
        echo "<li class='title'>";
        echo "Elements practicables";
        echo "</li>";
        } break;

case "PQ_16":
    if (in_array("PQ_1611", $proces) OR
        in_array("PQ_1641", $proces)) {
        echo "<li class='title'>";
        echo "Protecció";
        echo "</li>";
        } break;

case "PQ_17":
    if (in_array("PQ_1711", $proces) OR
        in_array("PQ_1741", $proces) OR
        in_array("PQ_1761", $proces) OR
        in_array("PQ_1771", $proces) OR
        in_array("PQ_1081", $proces)) {
        echo "<li class='title'>";
        echo "Sanejament";
        echo "</li>";
        } break;

case "PQ_18":
    if (in_array("PQ_1811", $proces) OR
        in_array("PQ_1821", $proces)) {
        echo "<li class='title'>";
        echo "Calefacció";
        echo "</li>";
        } break;

case "PQ_19":
    if (in_array("PQ_1911", $proces)) {
        echo "<li class='title'>";
        echo "Climatització";
        echo "</li>";
        } break;

case "PQ_20":
    if (in_array("PQ_2011", $proces) OR
        in_array("PQ_2013", $proces) OR
        in_array("PQ_2021", $proces) OR
        in_array("PQ_2031", $proces)) {
        echo "<li class='title'>";
        echo "Electricitat";
        echo "</li>";
        } break;

case "PQ_21":
    if (in_array("PQ_2111", $proces)) {
        echo "<li class='title'>";
        echo "Gas";
        echo "</li>";
        } break;

case "PQ_22":
    if (in_array("PQ_2211", $proces) OR
        in_array("PQ_2213", $proces)) {
        echo "<li class='title'>";
        echo "Aigua";
        echo "</li>";
        } break;

case "PQ_23":
    if (in_array("PQ_2311", $proces)) {
        echo "<li class='title'>";
        echo "Foc";
        echo "</li>";
        } break;
```

```
    case "PQ_24":
        if (in_array("PQ_2411", $proces)) {
            echo "<li class='title'>";
            echo "Parallamp";
            echo "</li>";
            } break;

    case "PQ_25":
        if (in_array("PQ_2511", $proces)) {
            echo "<li class='title'>";
            echo "Telecomunicació";
            echo "</li>";
            } break;

    case "PQ_26":
        if (in_array("PQ_2621", $proces)) {
            echo "<li class='title'>";
            echo "Elevadors";
            echo "</li>";
            } break;

    case "PQ_28":
        if (in_array("PQ_2811", $proces)) {
            echo "<li class='title'>";
            echo "Altres";
            echo "</li>";
            } break;

}

}

/* project_control.php */
function procesos_individuales ($proces, $init, $control_de, $project_id, $user_id) {

    for($i = 1; $i < count($proces); $i++) {
        if(substr($proces[$i], 0, 5) == $init ) {

        $proces_query = mysql_query("SELECT * FROM procesos WHERE codi = '".$proces[$i]."' ") or die
(mysql_error() );
        $dades_proces = mysql_fetch_array($proces_query); ?>

        <li class="regulable">
        <a href="processos/<?php echo $control_de; ?>.php?id=<?php echo urlencode($user_id); ?
>&project_id=<?php echo urlencode($project_id); ?>&codi=<?php echo urlencode($dades_proces['codi']); ?
>">
        <span class="name"><?php echo $dades_proces['titol']; ?></span>
        <span class="arrow"></span>
        </a>
        </li>
<?php
        }
    }
}

function valor_maxim($valor){
    switch($valor){
            case 'm2';
                $top = 20000; break;
            case 'ml';
                $top = 10000; break;
            default;
                $top = 100;
    }
    return $top;
}

function valid_email($email){
    if(eregi("^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.[a-z]{2,3})$", $email)){
        return 1;
    } else {
        return NULL;
```

```
    }
}

function user_not_allowed() {
    include THEME.S."header.php";

    top_bar('Accés denegat', HOME.S."app/login.php"); ?>

    <div id="content">
    <span class="graytitle">Error</span>
    <p class="graytext">Aquesta és una area privada. Per accedir has d'haver iniciat sessió
previament. Pots accedir al teu compte fent clic <a href="<?php echo HOME; ?>/app/login.php">aqui</a>.
    </div>
    <?php include THEME.S."footer.php";
}
```

```php
<?php

function get_header($tipo='normal') {
    if($tipo == 'normal') {
        include THEME.S."header.php";
        echo "<body>";
    } elseif($tipo == 'list') {
        include THEME.S."header.php";
        echo "<body class='list'>";
    } else {
        include THEME.S."header.php";
        echo "<body>";
    }
}


function top_bar($titulo='iQualit', $volver="main.php", $home='N', $casilla="Tornar") { ?>
    <div id="topbar">
        <div id="leftnav">
            <a href="<?php echo $volver; ?>">
                <?php if($home=='H'){ ?>
                    <img alt="home" src="theme/img/home.png"/>
                <?php } else { echo $casilla; } ?>
            </a>
        </div>
        <div id="title"><?php echo $titulo; ?></div>
    </div>
<?php
}

function top_proces_bar($codi=NULL, $volver="../main.php") {

    $query = mysql_query("SELECT titol FROM procesos WHERE codi = '".$codi."' ") or die();
    $proces = mysql_fetch_array($query);

    if ($proces) { ?>
        <div id="topbar">
            <div id="leftnav">
                <a href="<?php echo $volver; ?>">Tornar</a>
            </div>
            <div id="title"><?php echo $proces['titol']; ?></div>
        </div>
    <?php
    }
}


function error_message($content, $url) { ?>
    <span class="graytitle">Error</span>
    <p class="graytext"><b>S'han produit els seg&uuml;ents errors:</b> <?php echo $content; ?>. <a
href="<?php echo $url; ?>">Tornar enrrere</a>.</p>
<?php
}

function short_title($title) {
    if (strlen($title) > 37) {
        echo substr($title, 0, 34);
        echo '...';
    } else {
        echo $title; }
}


function get_footer(){
    include THEME.S."footer.php";
}

?>
```

```php
<?php
/*******************************************************************************
* FPDF                                                                         *
*                                                                              *
* Version: 1.6                                                                 *
* Date:    2008-08-03                                                          *
* Author:  Olivier PLATHEY                                                     *
*******************************************************************************/

define('FPDF_VERSION','1.6');

class FPDF
{
var $page;               //current page number
var $n;                  //current object number
var $offsets;            //array of object offsets
var $buffer;             //buffer holding in-memory PDF
var $pages;              //array containing pages
var $state;              //current document state
var $compress;           //compression flag
var $k;                  //scale factor (number of points in user unit)
var $DefOrientation;     //default orientation
var $CurOrientation;     //current orientation
var $PageFormats;        //available page formats
var $DefPageFormat;      //default page format
var $CurPageFormat;      //current page format
var $PageSizes;          //array storing non-default page sizes
var $wPt,$hPt;           //dimensions of current page in points
var $w,$h;               //dimensions of current page in user unit
var $lMargin;            //left margin
var $tMargin;            //top margin
var $rMargin;            //right margin
var $bMargin;            //page break margin
var $cMargin;            //cell margin
var $x,$y;               //current position in user unit
var $lasth;              //height of last printed cell
var $LineWidth;          //line width in user unit
var $CoreFonts;          //array of standard font names
var $fonts;              //array of used fonts
var $FontFiles;          //array of font files
var $diffs;              //array of encoding differences
var $FontFamily;         //current font family
var $FontStyle;          //current font style
var $underline;          //underlining flag
var $CurrentFont;        //current font info
var $FontSizePt;         //current font size in points
var $FontSize;           //current font size in user unit
var $DrawColor;          //commands for drawing color
var $FillColor;          //commands for filling color
var $TextColor;          //commands for text color
var $ColorFlag;          //indicates whether fill and text colors are different
var $ws;                 //word spacing
var $images;             //array of used images
var $PageLinks;          //array of links in pages
var $links;              //array of internal links
var $AutoPageBreak;      //automatic page breaking
var $PageBreakTrigger;   //threshold used to trigger page breaks
var $InHeader;           //flag set when processing header
var $InFooter;           //flag set when processing footer
var $ZoomMode;           //zoom display mode
var $LayoutMode;         //layout display mode
var $title;              //title
var $subject;            //subject
var $author;             //author
var $keywords;           //keywords
var $creator;            //creator
var $AliasNbPages;       //alias for total number of pages
var $PDFVersion;         //PDF version number

/*******************************************************************************
*                                                                              *
*                               Public methods                                 *
```

```
*                                                                               *
********************************************************************************/
function FPDF($orientation='P', $unit='mm', $format='A4')
{
    //Some checks
    $this->_dochecks();
    //Initialization of properties
    $this->page=0;
    $this->n=2;
    $this->buffer='';
    $this->pages=array();
    $this->PageSizes=array();
    $this->state=0;
    $this->fonts=array();
    $this->FontFiles=array();
    $this->diffs=array();
    $this->images=array();
    $this->links=array();
    $this->InHeader=false;
    $this->InFooter=false;
    $this->lasth=0;
    $this->FontFamily='';
    $this->FontStyle='';
    $this->FontSizePt=12;
    $this->underline=false;
    $this->DrawColor='0 G';
    $this->FillColor='0 g';
    $this->TextColor='0 g';
    $this->ColorFlag=false;
    $this->ws=0;
    //Standard fonts
    $this->CoreFonts=array('courier'=>'Courier', 'courierB'=>'Courier-Bold', 'courierI'=>'Courier-
Oblique', 'courierBI'=>'Courier-BoldOblique',
        'helvetica'=>'Helvetica', 'helveticaB'=>'Helvetica-Bold', 'helveticaI'=>'Helvetica-Oblique',
'helveticaBI'=>'Helvetica-BoldOblique',
        'times'=>'Times-Roman', 'timesB'=>'Times-Bold', 'timesI'=>'Times-Italic', 'timesBI'=>'Times-
BoldItalic',
        'symbol'=>'Symbol', 'zapfdingbats'=>'ZapfDingbats');
    //Scale factor
    if($unit=='pt')
        $this->k=1;
    elseif($unit=='mm')
        $this->k=72/25.4;
    elseif($unit=='cm')
        $this->k=72/2.54;
    elseif($unit=='in')
        $this->k=72;
    else
        $this->Error('Incorrect unit: '.$unit);
    //Page format
    $this->PageFormats=array('a3'=>array(841.89,1190.55), 'a4'=>array(595.28,841.89), 'a5'=>array
(420.94,595.28),
        'letter'=>array(612,792), 'legal'=>array(612,1008));
    if(is_string($format))
        $format=$this->_getpageformat($format);
    $this->DefPageFormat=$format;
    $this->CurPageFormat=$format;
    //Page orientation
    $orientation=strtolower($orientation);
    if($orientation=='p' || $orientation=='portrait')
    {
        $this->DefOrientation='P';
        $this->w=$this->DefPageFormat[0];
        $this->h=$this->DefPageFormat[1];
    }
    elseif($orientation=='l' || $orientation=='landscape')
    {
        $this->DefOrientation='L';
        $this->w=$this->DefPageFormat[1];
        $this->h=$this->DefPageFormat[0];
    }
    else
```

```
        $this->Error('Incorrect orientation: '.$orientation);
    $this->CurOrientation=$this->DefOrientation;
    $this->wPt=$this->w*$this->k;
    $this->hPt=$this->h*$this->k;
    //Page margins (1 cm)
    $margin=28.35/$this->k;
    $this->SetMargins($margin,$margin);
    //Interior cell margin (1 mm)
    $this->cMargin=$margin/10;
    //Line width (0.2 mm)
    $this->LineWidth=.567/$this->k;
    //Automatic page break
    $this->SetAutoPageBreak(true,2*$margin);
    //Full width display mode
    $this->SetDisplayMode('fullwidth');
    //Enable compression
    $this->SetCompression(true);
    //Set default PDF version number
    $this->PDFVersion='1.3';
}

function SetMargins($left, $top, $right=null)
{
    //Set left, top and right margins
    $this->lMargin=$left;
    $this->tMargin=$top;
    if($right===null)
        $right=$left;
    $this->rMargin=$right;
}

function SetLeftMargin($margin)
{
    //Set left margin
    $this->lMargin=$margin;
    if($this->page>0 && $this->x<$margin)
        $this->x=$margin;
}

function SetTopMargin($margin)
{
    //Set top margin
    $this->tMargin=$margin;
}

function SetRightMargin($margin)
{
    //Set right margin
    $this->rMargin=$margin;
}

function SetAutoPageBreak($auto, $margin=0)
{
    //Set auto page break mode and triggering margin
    $this->AutoPageBreak=$auto;
    $this->bMargin=$margin;
    $this->PageBreakTrigger=$this->h-$margin;
}

function SetDisplayMode($zoom, $layout='continuous')
{
    //Set display mode in viewer
    if($zoom=='fullpage' || $zoom=='fullwidth' || $zoom=='real' || $zoom=='default' || !is_string
($zoom))
        $this->ZoomMode=$zoom;
    else
        $this->Error('Incorrect zoom display mode: '.$zoom);
    if($layout=='single' || $layout=='continuous' || $layout=='two' || $layout=='default')
        $this->LayoutMode=$layout;
    else
        $this->Error('Incorrect layout display mode: '.$layout);
}
```

```
function SetCompression($compress)
{
    //Set page compression
    if(function_exists('gzcompress'))
        $this->compress=$compress;
    else
        $this->compress=false;
}

function SetTitle($title, $isUTF8=false)
{
    //Title of document
    if($isUTF8)
        $title=$this->_UTF8toUTF16($title);
    $this->title=$title;
}

function SetSubject($subject, $isUTF8=false)
{
    //Subject of document
    if($isUTF8)
        $subject=$this->_UTF8toUTF16($subject);
    $this->subject=$subject;
}

function SetAuthor($author, $isUTF8=false)
{
    //Author of document
    if($isUTF8)
        $author=$this->_UTF8toUTF16($author);
    $this->author=$author;
}

function SetKeywords($keywords, $isUTF8=false)
{
    //Keywords of document
    if($isUTF8)
        $keywords=$this->_UTF8toUTF16($keywords);
    $this->keywords=$keywords;
}

function SetCreator($creator, $isUTF8=false)
{
    //Creator of document
    if($isUTF8)
        $creator=$this->_UTF8toUTF16($creator);
    $this->creator=$creator;
}

function AliasNbPages($alias='{nb}')
{
    //Define an alias for total number of pages
    $this->AliasNbPages=$alias;
}

function Error($msg)
{
    //Fatal error
    die('<b>FPDF error:</b> '.$msg);
}

function Open()
{
    //Begin document
    $this->state=1;
}

function Close()
{
    //Terminate document
    if($this->state==3)
```

```
        return;
    if($this->page==0)
        $this->AddPage();
    //Page footer
    $this->InFooter=true;
    $this->Footer();
    $this->InFooter=false;
    //Close page
    $this->_endpage();
    //Close document
    $this->_enddoc();
}

function AddPage($orientation='', $format='')
{
    //Start a new page
    if($this->state==0)
        $this->Open();
    $family=$this->FontFamily;
    $style=$this->FontStyle.($this->underline ? 'U' : '');
    $size=$this->FontSizePt;
    $lw=$this->LineWidth;
    $dc=$this->DrawColor;
    $fc=$this->FillColor;
    $tc=$this->TextColor;
    $cf=$this->ColorFlag;
    if($this->page>0)
    {
        //Page footer
        $this->InFooter=true;
        $this->Footer();
        $this->InFooter=false;
        //Close page
        $this->_endpage();
    }
    //Start new page
    $this->_beginpage($orientation,$format);
    //Set line cap style to square
    $this->_out('2 J');
    //Set line width
    $this->LineWidth=$lw;
    $this->_out(sprintf('%.2F w',$lw*$this->k));
    //Set font
    if($family)
        $this->SetFont($family,$style,$size);
    //Set colors
    $this->DrawColor=$dc;
    if($dc!='0 G')
        $this->_out($dc);
    $this->FillColor=$fc;
    if($fc!='0 g')
        $this->_out($fc);
    $this->TextColor=$tc;
    $this->ColorFlag=$cf;
    //Page header
    $this->InHeader=true;
    $this->Header();
    $this->InHeader=false;
    //Restore line width
    if($this->LineWidth!=$lw)
    {
        $this->LineWidth=$lw;
        $this->_out(sprintf('%.2F w',$lw*$this->k));
    }
    //Restore font
    if($family)
        $this->SetFont($family,$style,$size);
    //Restore colors
    if($this->DrawColor!=$dc)
    {
        $this->DrawColor=$dc;
        $this->_out($dc);
```

```
    }
    if($this->FillColor!=$fc)
    {
        $this->FillColor=$fc;
        $this->_out($fc);
    }
    $this->TextColor=$tc;
    $this->ColorFlag=$cf;
}

function Header()
{
    //To be implemented in your own inherited class
}

function Footer()
{
    //To be implemented in your own inherited class
}

function PageNo()
{
    //Get current page number
    return $this->page;
}

function SetDrawColor($r, $g=null, $b=null)
{
    //Set color for all stroking operations
    if(($r==0 && $g==0 && $b==0) || $g===null)
        $this->DrawColor=sprintf('%.3F G',$r/255);
    else
        $this->DrawColor=sprintf('%.3F %.3F %.3F RG',$r/255,$g/255,$b/255);
    if($this->page>0)
        $this->_out($this->DrawColor);
}

function SetFillColor($r, $g=null, $b=null)
{
    //Set color for all filling operations
    if(($r==0 && $g==0 && $b==0) || $g===null)
        $this->FillColor=sprintf('%.3F g',$r/255);
    else
        $this->FillColor=sprintf('%.3F %.3F %.3F rg',$r/255,$g/255,$b/255);
    $this->ColorFlag=($this->FillColor!=$this->TextColor);
    if($this->page>0)
        $this->_out($this->FillColor);
}

function SetTextColor($r, $g=null, $b=null)
{
    //Set color for text
    if(($r==0 && $g==0 && $b==0) || $g===null)
        $this->TextColor=sprintf('%.3F g',$r/255);
    else
        $this->TextColor=sprintf('%.3F %.3F %.3F rg',$r/255,$g/255,$b/255);
    $this->ColorFlag=($this->FillColor!=$this->TextColor);
}

function GetStringWidth($s)
{
    //Get width of a string in the current font
    $s=(string)$s;
    $cw=&$this->CurrentFont['cw'];
    $w=0;
    $l=strlen($s);
    for($i=0;$i<$l;$i++)
        $w+=$cw[$s[$i]];
    return $w*$this->FontSize/1000;
}

function SetLineWidth($width)
```

```
{
    //Set line width
    $this->LineWidth=$width;
    if($this->page>0)
        $this->_out(sprintf('%.2F w',$width*$this->k));
}

function Line($x1, $y1, $x2, $y2)
{
    //Draw a line
    $this->_out(sprintf('%.2F %.2F m %.2F %.2F l S',$x1*$this->k,($this->h-$y1)*$this->k,$x2*$this->k,
($this->h-$y2)*$this->k));
}

function Rect($x, $y, $w, $h, $style='')
{
    //Draw a rectangle
    if($style=='F')
        $op='f';
    elseif($style=='FD' || $style=='DF')
        $op='B';
    else
        $op='S';
    $this->_out(sprintf('%.2F %.2F %.2F %.2F re %s',$x*$this->k,($this->h-$y)*$this->k,$w*$this->k,-$h*
$this->k,$op));
}

function AddFont($family, $style='', $file='')
{
    //Add a TrueType or Type1 font
    $family=strtolower($family);
    if($file=='')
        $file=str_replace(' ','',$family).strtolower($style).'.php';
    if($family=='arial')
        $family='helvetica';
    $style=strtoupper($style);
    if($style=='IB')
        $style='BI';
    $fontkey=$family.$style;
    if(isset($this->fonts[$fontkey]))
        return;
    include($this->_getfontpath().$file);
    if(!isset($name))
        $this->Error('Could not include font definition file');
    $i=count($this->fonts)+1;
    $this->fonts[$fontkey]=array('i'=>$i, 'type'=>$type, 'name'=>$name, 'desc'=>$desc, 'up'=>$up,
'ut'=>$ut, 'cw'=>$cw, 'enc'=>$enc, 'file'=>$file);
    if($diff)
    {
        //Search existing encodings
        $d=0;
        $nb=count($this->diffs);
        for($i=1;$i<=$nb;$i++)
        {
            if($this->diffs[$i]==$diff)
            {
                $d=$i;
                break;
            }
        }
        if($d==0)
        {
            $d=$nb+1;
            $this->diffs[$d]=$diff;
        }
        $this->fonts[$fontkey]['diff']=$d;
    }
    if($file)
    {
        if($type=='TrueType')
            $this->FontFiles[$file]=array('length1'=>$originalsize);
        else
```

```
            $this->FontFiles[$file]=array('length1'=>$size1, 'length2'=>$size2);
    }
}

function SetFont($family, $style='', $size=0)
{
    //Select a font; size given in points
    global $fpdf_charwidths;

    $family=strtolower($family);
    if($family=='')
        $family=$this->FontFamily;
    if($family=='arial')
        $family='helvetica';
    elseif($family=='symbol' || $family=='zapfdingbats')
        $style='';
    $style=strtoupper($style);
    if(strpos($style,'U')!==false)
    {
        $this->underline=true;
        $style=str_replace('U','',$style);
    }
    else
        $this->underline=false;
    if($style=='IB')
        $style='BI';
    if($size==0)
        $size=$this->FontSizePt;
    //Test if font is already selected
    if($this->FontFamily==$family && $this->FontStyle==$style && $this->FontSizePt==$size)
        return;
    //Test if used for the first time
    $fontkey=$family.$style;
    if(!isset($this->fonts[$fontkey]))
    {
        //Check if one of the standard fonts
        if(isset($this->CoreFonts[$fontkey]))
        {
            if(!isset($fpdf_charwidths[$fontkey]))
            {
                //Load metric file
                $file=$family;
                if($family=='times' || $family=='helvetica')
                    $file.=strtolower($style);
                include($this->_getfontpath().$file.'.php');
                if(!isset($fpdf_charwidths[$fontkey]))
                    $this->Error('Could not include font metric file');
            }
            $i=count($this->fonts)+1;
            $name=$this->CoreFonts[$fontkey];
            $cw=$fpdf_charwidths[$fontkey];
            $this->fonts[$fontkey]=array('i'=>$i, 'type'=>'core', 'name'=>$name, 'up'=>-100, 'ut'=>50,
'cw'=>$cw);
        }
        else
            $this->Error('Undefined font: '.$family.' '.$style);
    }
    //Select it
    $this->FontFamily=$family;
    $this->FontStyle=$style;
    $this->FontSizePt=$size;
    $this->FontSize=$size/$this->k;
    $this->CurrentFont=&$this->fonts[$fontkey];
    if($this->page>0)
        $this->_out(sprintf('BT /F%d %.2F Tf ET',$this->CurrentFont['i'],$this->FontSizePt));
}

function SetFontSize($size)
{
    //Set font size in points
    if($this->FontSizePt==$size)
        return;
```

```
    $this->FontSizePt=$size;
    $this->FontSize=$size/$this->k;
    if($this->page>0)
        $this->_out(sprintf('BT /F%d %.2F Tf ET',$this->CurrentFont['i'],$this->FontSizePt));
}

function AddLink()
{
    //Create a new internal link
    $n=count($this->links)+1;
    $this->links[$n]=array(0, 0);
    return $n;
}

function SetLink($link, $y=0, $page=-1)
{
    //Set destination of internal link
    if($y==-1)
        $y=$this->y;
    if($page==-1)
        $page=$this->page;
    $this->links[$link]=array($page, $y);
}

function Link($x, $y, $w, $h, $link)
{
    //Put a link on the page
    $this->PageLinks[$this->page][]=array($x*$this->k, $this->hPt-$y*$this->k, $w*$this->k, $h*$this-
>k, $link);
}

function Text($x, $y, $txt)
{
    //Output a string
    $s=sprintf('BT %.2F %.2F Td (%s) Tj ET',$x*$this->k,($this->h-$y)*$this->k,$this->_escape($txt));
    if($this->underline && $txt!='')
        $s.=' '.$this->_dounderline($x,$y,$txt);
    if($this->ColorFlag)
        $s='q '.$this->TextColor.' '.$s.' Q';
    $this->_out($s);
}

function AcceptPageBreak()
{
    //Accept automatic page break or not
    return $this->AutoPageBreak;
}

function Cell($w, $h=0, $txt='', $border=0, $ln=0, $align='', $fill=false, $link='')
{
    //Output a cell
    $k=$this->k;
    if($this->y+$h>$this->PageBreakTrigger && !$this->InHeader && !$this->InFooter && $this-
>AcceptPageBreak())
    {
        //Automatic page break
        $x=$this->x;
        $ws=$this->ws;
        if($ws>0)
        {
            $this->ws=0;
            $this->_out('0 Tw');
        }
        $this->AddPage($this->CurOrientation,$this->CurPageFormat);
        $this->x=$x;
        if($ws>0)
        {
            $this->ws=$ws;
            $this->_out(sprintf('%.3F Tw',$ws*$k));
        }
    }
    if($w==0)
```

```
        $w=$this->w-$this->rMargin-$this->x;
    $s='';
    if($fill || $border==1)
    {
        if($fill)
            $op=($border==1) ? 'B' : 'f';
        else
            $op='S';
        $s=sprintf('%.2F %.2F %.2F %.2F re %s ',$this->x*$k,($this->h-$this->y)*$k,$w*$k,-$h*$k,$op);
    }
    if(is_string($border))
    {
        $x=$this->x;
        $y=$this->y;
        if(strpos($border,'L')!==false)
            $s.=sprintf('%.2F %.2F m %.2F %.2F l S ',$x*$k,($this->h-$y)*$k,$x*$k,($this->h-($y+$h))*
$k);
        if(strpos($border,'T')!==false)
            $s.=sprintf('%.2F %.2F m %.2F %.2F l S ',$x*$k,($this->h-$y)*$k,($x+$w)*$k,($this->h-$y)*
$k);
        if(strpos($border,'R')!==false)
            $s.=sprintf('%.2F %.2F m %.2F %.2F l S ',($x+$w)*$k,($this->h-$y)*$k,($x+$w)*$k,($this->h-
($y+$h))*$k);
        if(strpos($border,'B')!==false)
            $s.=sprintf('%.2F %.2F m %.2F %.2F l S ',$x*$k,($this->h-($y+$h))*$k,($x+$w)*$k,($this->h-
($y+$h))*$k);
    }
    if($txt!=='')
    {
        if($align=='R')
            $dx=$w-$this->cMargin-$this->GetStringWidth($txt);
        elseif($align=='C')
            $dx=($w-$this->GetStringWidth($txt))/2;
        else
            $dx=$this->cMargin;
        if($this->ColorFlag)
            $s.='q '.$this->TextColor.' ';
        $txt2=str_replace(')','\\)',str_replace('(','\\(',str_replace('\\','\\\\',$txt)));
        $s.=sprintf('BT %.2F %.2F Td (%s) Tj ET',($this->x+$dx)*$k,($this->h-($this->y+.5*$h+.3*$this-
>FontSize))*$k,$txt2);
        if($this->underline)
            $s.=' '.$this->_dounderline($this->x+$dx,$this->y+.5*$h+.3*$this->FontSize,$txt);
        if($this->ColorFlag)
            $s.=' Q';
        if($link)
            $this->Link($this->x+$dx,$this->y+.5*$h-.5*$this->FontSize,$this->GetStringWidth($txt),
$this->FontSize,$link);
    }
    if($s)
        $this->_out($s);
    $this->lasth=$h;
    if($ln>0)
    {
        //Go to next line
        $this->y+=$h;
        if($ln==1)
            $this->x=$this->lMargin;
    }
    else
        $this->x+=$w;
}

function MultiCell($w, $h, $txt, $border=0, $align='J', $fill=false)
{
    //Output text with automatic or explicit line breaks
    $cw=&$this->CurrentFont['cw'];
    if($w==0)
        $w=$this->w-$this->rMargin-$this->x;
    $wmax=($w-2*$this->cMargin)*1000/$this->FontSize;
    $s=str_replace("\r",'',$txt);
    $nb=strlen($s);
    if($nb>0 && $s[$nb-1]=="\n")
```

```
        $nb--;
    $b=0;
    if($border)
    {
        if($border==1)
        {
            $border='LTRB';
            $b='LRT';
            $b2='LR';
        }
        else
        {
            $b2='';
            if(strpos($border,'L')!==false)
                $b2.='L';
            if(strpos($border,'R')!==false)
                $b2.='R';
            $b=(strpos($border,'T')!==false) ? $b2.'T' : $b2;
        }
    }
    $sep=-1;
    $i=0;
    $j=0;
    $l=0;
    $ns=0;
    $nl=1;
    while($i<$nb)
    {
        //Get next character
        $c=$s[$i];
        if($c=="\n")
        {
            //Explicit line break
            if($this->ws>0)
            {
                $this->ws=0;
                $this->_out('0 Tw');
            }
            $this->Cell($w,$h,substr($s,$j,$i-$j),$b,2,$align,$fill);
            $i++;
            $sep=-1;
            $j=$i;
            $l=0;
            $ns=0;
            $nl++;
            if($border && $nl==2)
                $b=$b2;
            continue;
        }
        if($c==' ')
        {
            $sep=$i;
            $ls=$l;
            $ns++;
        }
        $l+=$cw[$c];
        if($l>$wmax)
        {
            //Automatic line break
            if($sep==-1)
            {
                if($i==$j)
                    $i++;
                if($this->ws>0)
                {
                    $this->ws=0;
                    $this->_out('0 Tw');
                }
                $this->Cell($w,$h,substr($s,$j,$i-$j),$b,2,$align,$fill);
            }
            else
            {
```

```
                if($align=='J')
                {
                    $this->ws=($ns>1) ? ($wmax-$ls)/1000*$this->FontSize/($ns-1) : 0;
                    $this->_out(sprintf('%.3F Tw',$this->ws*$this->k));
                }
                $this->Cell($w,$h,substr($s,$j,$sep-$j),$b,2,$align,$fill);
                $i=$sep+1;
            }
            $sep=-1;
            $j=$i;
            $l=0;
            $ns=0;
            $nl++;
            if($border && $nl==2)
                $b=$b2;
        }
        else
            $i++;
    }
    //Last chunk
    if($this->ws>0)
    {
        $this->ws=0;
        $this->_out('0 Tw');
    }
    if($border && strpos($border,'B')!==false)
        $b.='B';
    $this->Cell($w,$h,substr($s,$j,$i-$j),$b,2,$align,$fill);
    $this->x=$this->lMargin;
}

function Write($h, $txt, $link='')
{
    //Output text in flowing mode
    $cw=&$this->CurrentFont['cw'];
    $w=$this->w-$this->rMargin-$this->x;
    $wmax=($w-2*$this->cMargin)*1000/$this->FontSize;
    $s=str_replace("\r",'',$txt);
    $nb=strlen($s);
    $sep=-1;
    $i=0;
    $j=0;
    $l=0;
    $nl=1;
    while($i<$nb)
    {
        //Get next character
        $c=$s[$i];
        if($c=="\n")
        {
            //Explicit line break
            $this->Cell($w,$h,substr($s,$j,$i-$j),0,2,'',0,$link);
            $i++;
            $sep=-1;
            $j=$i;
            $l=0;
            if($nl==1)
            {
                $this->x=$this->lMargin;
                $w=$this->w-$this->rMargin-$this->x;
                $wmax=($w-2*$this->cMargin)*1000/$this->FontSize;
            }
            $nl++;
            continue;
        }
        if($c==' ')
            $sep=$i;
        $l+=$cw[$c];
        if($l>$wmax)
        {
            //Automatic line break
            if($sep==-1)
```

```
            {
                if($this->x>$this->lMargin)
                {
                    //Move to next line
                    $this->x=$this->lMargin;
                    $this->y+=$h;
                    $w=$this->w-$this->rMargin-$this->x;
                    $wmax=($w-2*$this->cMargin)*1000/$this->FontSize;
                    $i++;
                    $nl++;
                    continue;
                }
                if($i==$j)
                    $i++;
                $this->Cell($w,$h,substr($s,$j,$i-$j),0,2,'',0,$link);
            }
            else
            {
                $this->Cell($w,$h,substr($s,$j,$sep-$j),0,2,'',0,$link);
                $i=$sep+1;
            }
            $sep=-1;
            $j=$i;
            $l=0;
            if($nl==1)
            {
                $this->x=$this->lMargin;
                $w=$this->w-$this->rMargin-$this->x;
                $wmax=($w-2*$this->cMargin)*1000/$this->FontSize;
            }
            $nl++;
        }
        else
            $i++;
    }
    //Last chunk
    if($i!=$j)
        $this->Cell($l/1000*$this->FontSize,$h,substr($s,$j),0,0,'',0,$link);
}

function Ln($h=null)
{
    //Line feed; default value is last cell height
    $this->x=$this->lMargin;
    if($h===null)
        $this->y+=$this->lasth;
    else
        $this->y+=$h;
}

function Image($file, $x=null, $y=null, $w=0, $h=0, $type='', $link='')
{
    //Put an image on the page
    if(!isset($this->images[$file]))
    {
        //First use of this image, get info
        if($type=='')
        {
            $pos=strrpos($file,'.');
            if(!$pos)
                $this->Error('Image file has no extension and no type was specified: '.$file);
            $type=substr($file,$pos+1);
        }
        $type=strtolower($type);
        if($type=='jpeg')
            $type='jpg';
        $mtd='_parse'.$type;
        if(!method_exists($this,$mtd))
            $this->Error('Unsupported image type: '.$type);
        $info=$this->$mtd($file);
        $info['i']=count($this->images)+1;
        $this->images[$file]=$info;
```

```
    }
    else
        $info=$this->images[$file];
    //Automatic width and height calculation if needed
    if($w==0 && $h==0)
    {
        //Put image at 72 dpi
        $w=$info['w']/$this->k;
        $h=$info['h']/$this->k;
    }
    elseif($w==0)
        $w=$h*$info['w']/$info['h'];
    elseif($h==0)
        $h=$w*$info['h']/$info['w'];
    //Flowing mode
    if($y===null)
    {
        if($this->y+$h>$this->PageBreakTrigger && !$this->InHeader && !$this->InFooter && $this->AcceptPageBreak())
        {
            //Automatic page break
            $x2=$this->x;
            $this->AddPage($this->CurOrientation,$this->CurPageFormat);
            $this->x=$x2;
        }
        $y=$this->y;
        $this->y+=$h;
    }
    if($x===null)
        $x=$this->x;
    $this->_out(sprintf('q %.2F 0 0 %.2F %.2F %.2F cm /I%d Do Q',$w*$this->k,$h*$this->k,$x*$this->k,($this->h-($y+$h))*$this->k,$info['i']));
    if($link)
        $this->Link($x,$y,$w,$h,$link);
}

function GetX()
{
    //Get x position
    return $this->x;
}

function SetX($x)
{
    //Set x position
    if($x>=0)
        $this->x=$x;
    else
        $this->x=$this->w+$x;
}

function GetY()
{
    //Get y position
    return $this->y;
}

function SetY($y)
{
    //Set y position and reset x
    $this->x=$this->lMargin;
    if($y>=0)
        $this->y=$y;
    else
        $this->y=$this->h+$y;
}

function SetXY($x, $y)
{
    //Set x and y positions
    $this->SetY($y);
    $this->SetX($x);
```

```
}

function Output($name='', $dest='')
{
    //Output PDF to some destination
    if($this->state<3)
        $this->Close();
    $dest=strtoupper($dest);
    if($dest=='')
    {
        if($name=='')
        {
            $name='doc.pdf';
            $dest='I';
        }
        else
            $dest='F';
    }
    switch($dest)
    {
        case 'I':
            //Send to standard output
            if(ob_get_length())
                $this->Error('Some data has already been output, can\'t send PDF file');
            if(php_sapi_name()!='cli')
            {
                //We send to a browser
                header('Content-Type: application/pdf');
                if(headers_sent())
                    $this->Error('Some data has already been output, can\'t send PDF file');
                header('Content-Length: '.strlen($this->buffer));
                header('Content-Disposition: inline; filename="'.$name.'"');
                header('Cache-Control: private, max-age=0, must-revalidate');
                header('Pragma: public');
                ini_set('zlib.output_compression','0');
            }
            echo $this->buffer;
            break;
        case 'D':
            //Download file
            if(ob_get_length())
                $this->Error('Some data has already been output, can\'t send PDF file');
            header('Content-Type: application/x-download');
            if(headers_sent())
                $this->Error('Some data has already been output, can\'t send PDF file');
            header('Content-Length: '.strlen($this->buffer));
            header('Content-Disposition: attachment; filename="'.$name.'"');
            header('Cache-Control: private, max-age=0, must-revalidate');
            header('Pragma: public');
            ini_set('zlib.output_compression','0');
            echo $this->buffer;
            break;
        case 'F':
            //Save to local file
            $f=fopen($name,'wb');
            if(!$f)
                $this->Error('Unable to create output file: '.$name);
            fwrite($f,$this->buffer,strlen($this->buffer));
            fclose($f);
            break;
        case 'S':
            //Return as a string
            return $this->buffer;
        default:
            $this->Error('Incorrect output destination: '.$dest);
    }
    return '';
}

/*******************************************************************************
*                                                                              *
*                               Protected methods                              *
```

```
*                                                                               *
********************************************************************************/
function _dochecks()
{
    //Check availability of %F
    if(sprintf('%.1F',1.0)!='1.0')
        $this->Error('This version of PHP is not supported');
    //Check mbstring overloading
    if(ini_get('mbstring.func_overload') & 2)
        $this->Error('mbstring overloading must be disabled');
    //Disable runtime magic quotes
    if(get_magic_quotes_runtime())
        @set_magic_quotes_runtime(0);
}

function _getpageformat($format)
{
    $format=strtolower($format);
    if(!isset($this->PageFormats[$format]))
        $this->Error('Unknown page format: '.$format);
    $a=$this->PageFormats[$format];
    return array($a[0]/$this->k, $a[1]/$this->k);
}

function _getfontpath()
{
    if(!defined('FPDF_FONTPATH') && is_dir(dirname(__FILE__).'/font'))
        define('FPDF_FONTPATH',dirname(__FILE__).'/font/');
    return defined('FPDF_FONTPATH') ? FPDF_FONTPATH : '';
}

function _beginpage($orientation, $format)
{
    $this->page++;
    $this->pages[$this->page]='';
    $this->state=2;
    $this->x=$this->lMargin;
    $this->y=$this->tMargin;
    $this->FontFamily='';
    //Check page size
    if($orientation=='')
        $orientation=$this->DefOrientation;
    else
        $orientation=strtoupper($orientation[0]);
    if($format=='')
        $format=$this->DefPageFormat;
    else
    {
        if(is_string($format))
            $format=$this->_getpageformat($format);
    }
    if($orientation!=$this->CurOrientation || $format[0]!=$this->CurPageFormat[0] || $format[1]!=$this-
>CurPageFormat[1])
    {
        //New size
        if($orientation=='P')
        {
            $this->w=$format[0];
            $this->h=$format[1];
        }
        else
        {
            $this->w=$format[1];
            $this->h=$format[0];
        }
        $this->wPt=$this->w*$this->k;
        $this->hPt=$this->h*$this->k;
        $this->PageBreakTrigger=$this->h-$this->bMargin;
        $this->CurOrientation=$orientation;
        $this->CurPageFormat=$format;
    }
    if($orientation!=$this->DefOrientation || $format[0]!=$this->DefPageFormat[0] || $format[1]!=$this-
```

```
>DefPageFormat[1])
        $this->PageSizes[$this->page]=array($this->wPt, $this->hPt);
}

function _endpage()
{
    $this->state=1;
}

function _escape($s)
{
    //Escape special characters in strings
    $s=str_replace('\\','\\\\',$s);
    $s=str_replace('(','\\(',$s);
    $s=str_replace(')','\\)',$s);
    $s=str_replace("\r",'\\r',$s);
    return $s;
}

function _textstring($s)
{
    //Format a text string
    return '('.$this->_escape($s).')';
}

function _UTF8toUTF16($s)
{
    //Convert UTF-8 to UTF-16BE with BOM
    $res="\xFE\xFF";
    $nb=strlen($s);
    $i=0;
    while($i<$nb)
    {
        $c1=ord($s[$i++]);
        if($c1>=224)
        {
            //3-byte character
            $c2=ord($s[$i++]);
            $c3=ord($s[$i++]);
            $res.=chr((($c1 & 0x0F)<<4) + (($c2 & 0x3C)>>2));
            $res.=chr((($c2 & 0x03)<<6) + ($c3 & 0x3F));
        }
        elseif($c1>=192)
        {
            //2-byte character
            $c2=ord($s[$i++]);
            $res.=chr(($c1 & 0x1C)>>2);
            $res.=chr((($c1 & 0x03)<<6) + ($c2 & 0x3F));
        }
        else
        {
            //Single-byte character
            $res.="\0".chr($c1);
        }
    }
    return $res;
}

function _dounderline($x, $y, $txt)
{
    //Underline text
    $up=$this->CurrentFont['up'];
    $ut=$this->CurrentFont['ut'];
    $w=$this->GetStringWidth($txt)+$this->ws*substr_count($txt,' ');
    return sprintf('%.2F %.2F %.2F %.2F re f',$x*$this->k,($this->h-($y-$up/1000*$this->FontSize))*
$this->k,$w*$this->k,-$ut/1000*$this->FontSizePt);
}

function _parsejpg($file)
{
    //Extract info from a JPEG file
    $a=GetImageSize($file);
```

```
    if(!$a)
        $this->Error('Missing or incorrect image file: '.$file);
    if($a[2]!=2)
        $this->Error('Not a JPEG file: '.$file);
    if(!isset($a['channels']) || $a['channels']==3)
        $colspace='DeviceRGB';
    elseif($a['channels']==4)
        $colspace='DeviceCMYK';
    else
        $colspace='DeviceGray';
    $bpc=isset($a['bits']) ? $a['bits'] : 8;
    //Read whole file
    $f=fopen($file,'rb');
    $data='';
    while(!feof($f))
        $data.=fread($f,8192);
    fclose($f);
    return array('w'=>$a[0], 'h'=>$a[1], 'cs'=>$colspace, 'bpc'=>$bpc, 'f'=>'DCTDecode', 'data'=>
$data);
}

function _parsepng($file)
{
    //Extract info from a PNG file
    $f=fopen($file,'rb');
    if(!$f)
        $this->Error('Can\'t open image file: '.$file);
    //Check signature
    if($this->_readstream($f,8)!=chr(137).'PNG'.chr(13).chr(10).chr(26).chr(10))
        $this->Error('Not a PNG file: '.$file);
    //Read header chunk
    $this->_readstream($f,4);
    if($this->_readstream($f,4)!='IHDR')
        $this->Error('Incorrect PNG file: '.$file);
    $w=$this->_readint($f);
    $h=$this->_readint($f);
    $bpc=ord($this->_readstream($f,1));
    if($bpc>8)
        $this->Error('16-bit depth not supported: '.$file);
    $ct=ord($this->_readstream($f,1));
    if($ct==0)
        $colspace='DeviceGray';
    elseif($ct==2)
        $colspace='DeviceRGB';
    elseif($ct==3)
        $colspace='Indexed';
    else
        $this->Error('Alpha channel not supported: '.$file);
    if(ord($this->_readstream($f,1))!=0)
        $this->Error('Unknown compression method: '.$file);
    if(ord($this->_readstream($f,1))!=0)
        $this->Error('Unknown filter method: '.$file);
    if(ord($this->_readstream($f,1))!=0)
        $this->Error('Interlacing not supported: '.$file);
    $this->_readstream($f,4);
    $parms='/DecodeParms <</Predictor 15 /Colors '.($ct==2 ? 3 : 1).' /BitsPerComponent '.$bpc.' /
Columns '.$w.'>>';
    //Scan chunks looking for palette, transparency and image data
    $pal='';
    $trns='';
    $data='';
    do
    {
        $n=$this->_readint($f);
        $type=$this->_readstream($f,4);
        if($type=='PLTE')
        {
            //Read palette
            $pal=$this->_readstream($f,$n);

            $this->_readstream($f,4);
        }
```

```
        elseif($type=='tRNS')
        {
            //Read transparency info
            $t=$this->_readstream($f,$n);
            if($ct==0)
                $trns=array(ord(substr($t,1,1)));
            elseif($ct==2)
                $trns=array(ord(substr($t,1,1)), ord(substr($t,3,1)), ord(substr($t,5,1)));
            else
            {
                $pos=strpos($t,chr(0));
                if($pos!==false)
                    $trns=array($pos);
            }
            $this->_readstream($f,4);
        }
        elseif($type=='IDAT')
        {
            //Read image data block
            $data.=$this->_readstream($f,$n);
            $this->_readstream($f,4);
        }
        elseif($type=='IEND')
            break;
        else
            $this->_readstream($f,$n+4);
    }
    while($n);
    if($colspace=='Indexed' && empty($pal))
        $this->Error('Missing palette in '.$file);
    fclose($f);
    return array('w'=>$w, 'h'=>$h, 'cs'=>$colspace, 'bpc'=>$bpc, 'f'=>'FlateDecode', 'parms'=>$parms,
'pal'=>$pal, 'trns'=>$trns, 'data'=>$data);
}

function _readstream($f, $n)
{
    //Read n bytes from stream
    $res='';
    while($n>0 && !feof($f))
    {
        $s=fread($f,$n);
        if($s===false)
            $this->Error('Error while reading stream');
        $n-=strlen($s);
        $res.=$s;
    }
    if($n>0)
        $this->Error('Unexpected end of stream');
    return $res;
}

function _readint($f)
{
    //Read a 4-byte integer from stream
    $a=unpack('Ni',$this->_readstream($f,4));
    return $a['i'];
}

function _parsegif($file)
{
    //Extract info from a GIF file (via PNG conversion)
    if(!function_exists('imagepng'))
        $this->Error('GD extension is required for GIF support');
    if(!function_exists('imagecreatefromgif'))
        $this->Error('GD has no GIF read support');
    $im=imagecreatefromgif($file);
    if(!$im)
        $this->Error('Missing or incorrect image file: '.$file);
    imageinterlace($im,0);
    $tmp=tempnam('.','gif');
    if(!$tmp)
```

```
        $this->Error('Unable to create a temporary file');
    if(!imagepng($im,$tmp))
        $this->Error('Error while saving to temporary file');
    imagedestroy($im);
    $info=$this->_parsepng($tmp);
    unlink($tmp);
    return $info;
}

function _newobj()
{
    //Begin a new object
    $this->n++;
    $this->offsets[$this->n]=strlen($this->buffer);
    $this->_out($this->n.' 0 obj');
}

function _putstream($s)
{
    $this->_out('stream');
    $this->_out($s);
    $this->_out('endstream');
}

function _out($s)
{
    //Add a line to the document
    if($this->state==2)
        $this->pages[$this->page].=$s."\n";
    else
        $this->buffer.=$s."\n";
}

function _putpages()
{
    $nb=$this->page;
    if(!empty($this->AliasNbPages))
    {
        //Replace number of pages
        for($n=1;$n<=$nb;$n++)
            $this->pages[$n]=str_replace($this->AliasNbPages,$nb,$this->pages[$n]);
    }
    if($this->DefOrientation=='P')
    {
        $wPt=$this->DefPageFormat[0]*$this->k;
        $hPt=$this->DefPageFormat[1]*$this->k;
    }
    else
    {
        $wPt=$this->DefPageFormat[1]*$this->k;
        $hPt=$this->DefPageFormat[0]*$this->k;
    }
    $filter=($this->compress) ? '/Filter /FlateDecode ' : '';
    for($n=1;$n<=$nb;$n++)
    {
        //Page
        $this->_newobj();
        $this->_out('<</Type /Page');
        $this->_out('/Parent 1 0 R');
        if(isset($this->PageSizes[$n]))
            $this->_out(sprintf('/MediaBox [0 0 %.2F %.2F]',$this->PageSizes[$n][0],$this->PageSizes
[$n][1]));
        $this->_out('/Resources 2 0 R');
        if(isset($this->PageLinks[$n]))
        {
            //Links
            $annots='/Annots [';
            foreach($this->PageLinks[$n] as $pl)
            {
                $rect=sprintf('%.2F %.2F %.2F %.2F',$pl[0],$pl[1],$pl[0]+$pl[2],$pl[1]-$pl[3]);
                $annots.='<</Type /Annot /Subtype /Link /Rect ['.$rect.'] /Border [0 0 0] ';
                if(is_string($pl[4]))
```

```
                    $annots.='/A <</S /URI /URI '.$this->_textstring($pl[4]).'>>>>';
                else
                {
                    $l=$this->links[$pl[4]];
                    $h=isset($this->PageSizes[$l[0]]) ? $this->PageSizes[$l[0]][1] : $hPt;
                    $annots.=sprintf('/Dest [%d 0 R /XYZ 0 %.2F null]>>',1+2*$l[0],$h-$l[1]*$this->k);
                }
            }
            $this->_out($annots.']');
        }
        $this->_out('/Contents '.($this->n+1).' 0 R>>');
        $this->_out('endobj');
        //Page content
        $p=($this->compress) ? gzcompress($this->pages[$n]) : $this->pages[$n];
        $this->_newobj();
        $this->_out('<<'.$filter.'/Length '.strlen($p).'>>');
        $this->_putstream($p);
        $this->_out('endobj');
    }
    //Pages root
    $this->offsets[1]=strlen($this->buffer);
    $this->_out('1 0 obj');
    $this->_out('<</Type /Pages');
    $kids='/Kids [';
    for($i=0;$i<$nb;$i++)
        $kids.=(3+2*$i).' 0 R ';
    $this->_out($kids.']');
    $this->_out('/Count '.$nb);
    $this->_out(sprintf('/MediaBox [0 0 %.2F %.2F]',$wPt,$hPt));
    $this->_out('>>');
    $this->_out('endobj');
}

function _putfonts()
{
    $nf=$this->n;
    foreach($this->diffs as $diff)
    {
        //Encodings
        $this->_newobj();
        $this->_out('<</Type /Encoding /BaseEncoding /WinAnsiEncoding /Differences ['.$diff.']>>');
        $this->_out('endobj');
    }
    foreach($this->FontFiles as $file=>$info)
    {
        //Font file embedding
        $this->_newobj();
        $this->FontFiles[$file]['n']=$this->n;
        $font='';
        $f=fopen($this->_getfontpath().$file,'rb',1);
        if(!$f)
            $this->Error('Font file not found');
        while(!feof($f))
            $font.=fread($f,8192);
        fclose($f);
        $compressed=(substr($file,-2)=='.z');
        if(!$compressed && isset($info['length2']))
        {
            $header=(ord($font[0])==128);
            if($header)
            {
                //Strip first binary header
                $font=substr($font,6);
            }
            if($header && ord($font[$info['length1']])==128)
            {
                //Strip second binary header
                $font=substr($font,0,$info['length1']).substr($font,$info['length1']+6);
            }
        }
        $this->_out('<</Length '.strlen($font));
        if($compressed)
```

```
            $this->_out('/Filter /FlateDecode');
        $this->_out('/Length1 '.$info['length1']);
        if(isset($info['length2']))
            $this->_out('/Length2 '.$info['length2'].' /Length3 0');
        $this->_out('>>');
        $this->_putstream($font);
        $this->_out('endobj');
    }
    foreach($this->fonts as $k=>$font)
    {
        //Font objects
        $this->fonts[$k]['n']=$this->n+1;
        $type=$font['type'];
        $name=$font['name'];
        if($type=='core')
        {
            //Standard font
            $this->_newobj();
            $this->_out('<</Type /Font');
            $this->_out('/BaseFont /'.$name);
            $this->_out('/Subtype /Type1');
            if($name!='Symbol' && $name!='ZapfDingbats')
                $this->_out('/Encoding /WinAnsiEncoding');
            $this->_out('>>');
            $this->_out('endobj');
        }
        elseif($type=='Type1' || $type=='TrueType')
        {
            //Additional Type1 or TrueType font
            $this->_newobj();
            $this->_out('<</Type /Font');
            $this->_out('/BaseFont /'.$name);
            $this->_out('/Subtype /'.$type);
            $this->_out('/FirstChar 32 /LastChar 255');
            $this->_out('/Widths '.($this->n+1).' 0 R');
            $this->_out('/FontDescriptor '.($this->n+2).' 0 R');
            if($font['enc'])
            {
                if(isset($font['diff']))
                    $this->_out('/Encoding '.($nf+$font['diff']).' 0 R');
                else
                    $this->_out('/Encoding /WinAnsiEncoding');
            }
            $this->_out('>>');
            $this->_out('endobj');
            //Widths
            $this->_newobj();
            $cw=&$font['cw'];
            $s='[';
            for($i=32;$i<=255;$i++)
                $s.=$cw[chr($i)].' ';
            $this->_out($s.']');
            $this->_out('endobj');
            //Descriptor
            $this->_newobj();
            $s='<</Type /FontDescriptor /FontName /'.$name;
            foreach($font['desc'] as $k=>$v)
                $s.=' /'.$k.' '.$v;
            $file=$font['file'];
            if($file)
                $s.=' /FontFile'.($type=='Type1' ? '' : '2').' '.$this->FontFiles[$file]['n'].' 0 R';
            $this->_out($s.'>>');
            $this->_out('endobj');
        }
        else
        {
            //Allow for additional types
            $mtd='_put'.strtolower($type);
            if(!method_exists($this,$mtd))
                $this->Error('Unsupported font type: '.$type);
            $this->$mtd($font);
        }
```

```
    }
}

function _putimages()
{
    $filter=($this->compress) ? '/Filter /FlateDecode ' : '';
    reset($this->images);
    while(list($file,$info)=each($this->images))
    {
        $this->_newobj();
        $this->images[$file]['n']=$this->n;
        $this->_out('<</Type /XObject');
        $this->_out('/Subtype /Image');
        $this->_out('/Width '.$info['w']);
        $this->_out('/Height '.$info['h']);
        if($info['cs']=='Indexed')
            $this->_out('/ColorSpace [/Indexed /DeviceRGB '.(strlen($info['pal'])/3-1).' '.($this->n
+1).' 0 R]');
        else
        {
            $this->_out('/ColorSpace /'.$info['cs']);
            if($info['cs']=='DeviceCMYK')
                $this->_out('/Decode [1 0 1 0 1 0 1 0]');
        }
        $this->_out('/BitsPerComponent '.$info['bpc']);
        if(isset($info['f']))
            $this->_out('/Filter /'.$info['f']);
        if(isset($info['parms']))
            $this->_out($info['parms']);
        if(isset($info['trns']) && is_array($info['trns']))
        {
            $trns='';
            for($i=0;$i<count($info['trns']);$i++)
                $trns.=$info['trns'][$i].' '.$info['trns'][$i].' ';
            $this->_out('/Mask ['.$trns.']');
        }
        $this->_out('/Length '.strlen($info['data']).'>>');
        $this->_putstream($info['data']);
        unset($this->images[$file]['data']);
        $this->_out('endobj');
        //Palette
        if($info['cs']=='Indexed')
        {
            $this->_newobj();
            $pal=($this->compress) ? gzcompress($info['pal']) : $info['pal'];
            $this->_out('<<'.$filter.'/Length '.strlen($pal).'>>');
            $this->_putstream($pal);
            $this->_out('endobj');
        }
    }
}

function _putxobjectdict()
{
    foreach($this->images as $image)
        $this->_out('/I'.$image['i'].' '.$image['n'].' 0 R');
}

function _putresourcedict()
{
    $this->_out('/ProcSet [/PDF /Text /ImageB /ImageC /ImageI]');
    $this->_out('/Font <<');
    foreach($this->fonts as $font)
        $this->_out('/F'.$font['i'].' '.$font['n'].' 0 R');
    $this->_out('>>');
    $this->_out('/XObject <<');
    $this->_putxobjectdict();
    $this->_out('>>');
}

function _putresources()
{
```

```
    $this->_putfonts();
    $this->_putimages();
    //Resource dictionary
    $this->offsets[2]=strlen($this->buffer);
    $this->_out('2 0 obj');
    $this->_out('<<');
    $this->_putresourcedict();
    $this->_out('>>');
    $this->_out('endobj');
}

function _putinfo()
{
    $this->_out('/Producer '.$this->_textstring('FPDF '.FPDF_VERSION));
    if(!empty($this->title))
        $this->_out('/Title '.$this->_textstring($this->title));
    if(!empty($this->subject))
        $this->_out('/Subject '.$this->_textstring($this->subject));
    if(!empty($this->author))
        $this->_out('/Author '.$this->_textstring($this->author));
    if(!empty($this->keywords))
        $this->_out('/Keywords '.$this->_textstring($this->keywords));
    if(!empty($this->creator))
        $this->_out('/Creator '.$this->_textstring($this->creator));
    $this->_out('/CreationDate '.$this->_textstring('D:'.@date('YmdHis')));
}

function _putcatalog()
{
    $this->_out('/Type /Catalog');
    $this->_out('/Pages 1 0 R');
    if($this->ZoomMode=='fullpage')
        $this->_out('/OpenAction [3 0 R /Fit]');
    elseif($this->ZoomMode=='fullwidth')
        $this->_out('/OpenAction [3 0 R /FitH null]');
    elseif($this->ZoomMode=='real')
        $this->_out('/OpenAction [3 0 R /XYZ null null 1]');
    elseif(!is_string($this->ZoomMode))
        $this->_out('/OpenAction [3 0 R /XYZ null null '.($this->ZoomMode/100).']');
    if($this->LayoutMode=='single')
        $this->_out('/PageLayout /SinglePage');
    elseif($this->LayoutMode=='continuous')
        $this->_out('/PageLayout /OneColumn');
    elseif($this->LayoutMode=='two')
        $this->_out('/PageLayout /TwoColumnLeft');
}

function _putheader()
{
    $this->_out('%PDF-'.$this->PDFVersion);
}

function _puttrailer()
{
    $this->_out('/Size '.($this->n+1));
    $this->_out('/Root '.$this->n.' 0 R');
    $this->_out('/Info '.($this->n-1).' 0 R');
}

function _enddoc()
{
    $this->_putheader();
    $this->_putpages();
    $this->_putresources();
    //Info
    $this->_newobj();
    $this->_out('<<');
    $this->_putinfo();
    $this->_out('>>');
    $this->_out('endobj');
    //Catalog
    $this->_newobj();
```

```
    $this->_out('<<');
    $this->_putcatalog();
    $this->_out('>>');
    $this->_out('endobj');
    //Cross-ref
    $o=strlen($this->buffer);
    $this->_out('xref');
    $this->_out('0 '.($this->n+1));
    $this->_out('0000000000 65535 f ');
    for($i=1;$i<=$this->n;$i++)
        $this->_out(sprintf('%010d 00000 n ',$this->offsets[$i]));
    //Trailer
    $this->_out('trailer');
    $this->_out('<<');
    $this->_puttrailer();
    $this->_out('>>');
    $this->_out('startxref');
    $this->_out($o);
    $this->_out('%%EOF');
    $this->state=3;
}
//End of class
}

//Handle special IE contype request
if(isset($_SERVER['HTTP_USER_AGENT']) && $_SERVER['HTTP_USER_AGENT']=='contype')
{
    header('Content-Type: application/pdf');
    exit;
}

?>
```

```php
<?php

class Recepcio {

    function data($codi){
        //Coge los datos de los parametros de ejecucion
        $recepcio_query = "SELECT element FROM recepcio_data WHERE codi = '".$codi."'";

        $recepcio_array = mysql_query($recepcio_query);

        $recepcio_element_data = mysql_fetch_array($recepcio_array);


        //Coge el titulo de la otra base de datos
        $titol_query = mysql_query("SELECT titol FROM procesos WHERE codi ='".$codi."'");
        $titol_array = mysql_fetch_array($titol_query);
        $titol = $titol_array['titol'];

        $array_recepcio = array(0 => $recepcio_element_data[0], 1 => $titol);

        return $array_recepcio;

    }

    function existeix_bd($codi, $project_id){
        $query = "SELECT marcats FROM recepcio WHERE codi = '".$codi."' AND project_id = '".
$project_id."'";
        $recepcio_fets = mysql_query($query);

        if (mysql_affected_rows() == 0){
            return 1;
        } else {
            return 0;
        }
    }

    function tipus_doc($valor){
        switch($valor){

            case '1a'; return
                "<p class='obligatori'>Fulls de subministrament. Etiquetat</p>"; break;
            case '1b'; return
                "<p class='voluntari'>Fulls de subministrament. Etiquetat</p>"; break;

            case '2a'; return
                "<p class='obligatori'>Garantia del fabricant</p>"; break;
            case '2b'; return
                "<p class='voluntari'>Garantia del fabricant</p>"; break;

            case '3a'; return
                "<p class='obligatori'>Declaraci&oacute; de conformitat</p>"; break;
            case '3b'; return
                "<p class='voluntari'>Declaraci&oacute; de conformitat</p>"; break;

            case '4a'; return
                "<p class='obligatori'>Certificat CE</p>"; break;
            case '4b'; return
                "<p class='voluntari'>Certificat CE</p>"; break;

            case '5a'; return
                "<p class='obligatori'>Control producci&oacute; a fabrica</p>"; break;
            case '5b'; return
                "<p class='voluntari'>Control producci&oacute; a fabrica</p>"; break;

            case '6a'; return
                "<p class='obligatori'>DITE</p>"; break;
            case '6b'; return
                "<p class='voluntari'>DITE</p>"; break;

            case '7a'; return
                "<p class='obligatori'><i>Altres</i></p>"; break;
            case '7b'; return
```

```
                "<p class='voluntari'><i>Altres</i></p>"; break;

            case '8a'; return
                "<p class='obligatori'>Adicional a la etiqueta CE</p>"; break;
            case '8b'; return
                "<p class='voluntari'>Adicional a la etiqueta CE</p>"; break;

            case '9a'; return
                "<p class='obligatori'>Segell, marca de conformitat</p>"; break;
            case '9b'; return
                "<p class='voluntari'>Segell, marca de conformitat</p>"; break;

            case '10a'; return
                "<p class='obligatori'>CCRR</p>"; break;
            case '10b'; return
                "<p class='voluntari'>CCRR</p>"; break;

            case '11a'; return
                "<p class='obligatori'>Homologaci&oacute;</p>"; break;
            case '11b'; return
                "<p class='voluntari'>Homologaci&oacute;</p>"; break;

            case '12a'; return
                "<p class='obligatori'>Marca AENOR 'N'</p>"; break;
            case '12b'; return
                "<p class='voluntari'>Marca AENOR 'N'</p>"; break;

            case '13a'; return
                "<p class='obligatori'>Assaig/s. Norma/s. Especificaci&oacute;</p>"; break;
            case '13b'; return
                "<p class='voluntari'>Assaig/s. Norma/s. Especificaci&oacute;</p>"; break;

            case '14a'; return
                "<p class='obligatori'>Certif. de origen s/especif. tecniques</p>"; break;
            case '14b'; return
                "<p class='voluntari'>Certif. de origen s/especif. tecniques</p>"; break;

            case '15a'; return
                "<p class='obligatori'>Fitxa caracteristiques tecniques</p>"; break;
            case '15b'; return
                "<p class='voluntari'>Fitxa caracteristiques tecniques</p>"; break;

            case '16a'; return
                "<p class='obligatori'>Distintius de qualitat voluntaris</p>"; break;
            case '16b'; return
                "<p class='voluntari'>Distintius de qualitat voluntaris</p>"; break;

            case '17a'; return
                "<p class='obligatori'>Avaluacions d'idoneitat tecnica</p>"; break;
            case '17b'; return
                "<p class='voluntari'>Avaluacions d'idoneitat tecnica</p>"; break;
        }
    }

    function checked($var, $codi, $project_id){
        $query = "SELECT marcats FROM recepcio WHERE codi = '".$codi."' AND project_id = '".
$project_id."'";

        $mysql_query = mysql_query($query);

        $fetch_array = mysql_fetch_array($mysql_query);

        $marcats = explode (', ', $fetch_array[0]);

        if(in_array($var, $marcats)){
            echo 'checked';
        }
    }

}

$recepcio = new Recepcio();
```

```php
<?php

class Execucio {

    function data($codi){
        //Coge los datos de los parametros de ejecucion
        $ex_param_query = "SELECT parametres, tipus_control, unitat_lot, tamany_lot FROM execucio_data
WHERE codi = '".$codi."'";
        $ex_param = mysql_query($ex_param_query);
        $array_ex = mysql_fetch_array($ex_param);

        $parametres_execucio = explode(' | ', $array_ex[0]);
        $tipus_control       = explode(' | ', $array_ex[1]);
        $unitat_lot          = $array_ex[2];
        $tamany_lot          = $array_ex[3];

        $data = array(
            'parametres_execucio' => $parametres_execucio,
            'tipus_control' => $tipus_control,
            'unitat_lot' => $unitat_lot,
            'tamany_lot' => $tamany_lot);

        return $data;
    }

    function sense_lots($project_id, $codi){

        $lots_query = mysql_query("SELECT lots FROM execucio WHERE project_id = '".$project_id."' AND
codi = '".$codi."'");

        $data = mysql_fetch_array($lots_query);

        if($data['lots'] == 0) {
            return 1;
        } else {
            return NULL;
        }
    }

    function num_lots($project_id, $codi){

        $lots_query = mysql_query("SELECT lots FROM execucio WHERE project_id = '".$project_id."' AND
codi = '".$codi."'");

        $data = mysql_fetch_array($lots_query);

        return $data['lots'];
    }

    function sense_lot_arxiu($codi, $tipo = 'nodata'){
        if ($tipo == 'nodata'){
            if(file_exists("execucio/{$codi}_sin.php")){
                include "execucio/{$codi}_sin.php";
            }
        } elseif ($tipo == 'data') {
            if(file_exists("execucio/{$codi}_data.php")){
                include "execucio/{$codi}_data.php";
            }
        }
    }

    function calcul_lots($_POST, $codi){

        $data = $this->data($codi);
        $tamany_lot = $data['tamany_lot'];

        if($_POST['submit']){
            if(is_numeric($_POST['total'])){
                $lots = $_POST['total']/$tamany_lot;
                $lots_totals = ceil($lots);
                return $lots_totals;
            } else {
```

```
                echo "Introdueix un valor numeric al formulari.";
            }
        }
    }

    function guardar_controls_fets($_POST, $project_id, $codi) {
        foreach ($_POST as $key => $value){
            if(is_int($key)){
                $key_date = $key . '_date';

                if ($_POST[$key_date]){
                    $post_array[] = $key . ' @ ' . $_POST[$key_date];
                }
            }
        }

        if($post_array){
            $fets = implode(', ', $post_array);
        } else {
            $fets = 0;
        }

        $fets_query = "UPDATE execucio SET fets = '".$fets."' WHERE project_id = '".$project_id."' AND
codi = '".$codi."'";
        mysql_query($fets_query);
    }

    function marcar_com_fet($i, $key, $project_id, $codi) {

        $query_fets = mysql_query("SELECT fets FROM execucio WHERE project_id = '".$project_id."' AND
codi = '".$codi."'");

        $data = mysql_fetch_array($query_fets);

        $fets_amb_date = explode(', ', $data[0]);

        $id = $i.$key;

        foreach ($fets_amb_date as $fet_amb_date){

            $fet_amb_date_array = explode(' @ ', $fet_amb_date);

            if ($id == $fet_amb_date_array[0]){
                $fets[] = $fet_amb_date_array[0];
                $date = $fet_amb_date_array[1];

                if (!empty($date)){ ?>
<input name="<?php echo $id; ?>" class="lot <?php echo $i; ?>"type="checkbox" checked /> Fet el dia <?
php echo $date; ?>
    <!--<a href="javascript:togglecomments(<?php echo $id; ?>2)">-->
        <img src="<?php echo HOME; ?>/app/theme/icons/schedule.png" alt="edit date">
    <!--</a>-->
<input type="hidden" name="<?php echo $id; ?>_date" value="<?php echo $date; ?>" />
                <?php } else { ?>
                <input name="<?php echo $id; ?>" class="lot <?php echo $i; ?>"type="checkbox" checked /
> Marcat com fet
        <input type="hidden" name="<?php echo $id; ?>_date" value="<?php echo date('j/m/Y'); ?>" />
                <?php }

            }

        }

    if(empty($fets)) { ?>
                    <input name="<?php echo $id; ?>" class="lot <?php echo $i; ?>"type="checkbox" />
Marcar com fet
                    <input type="hidden" name="<?php echo $id; ?>_date" value="<?php echo date('j/m/
Y'); ?>" />

    <?php }

    }
```

```
    function edit_date($id, $project_id, $codi) {
        echo 'Editar data...';
    }


    function color_fet($id, $project_id, $codi) {

        $query_fets = mysql_query("SELECT fets FROM execucio WHERE project_id = '".$project_id."' AND
codi = '".$codi."'");

        $data = mysql_fetch_array($query_fets);

        $fets_amb_date = explode(', ', $data[0]);

        foreach ($fets_amb_date as $fet_amb_date){

            $fet_amb_date_array = explode(' @ ', $fet_amb_date);

            $fets[] = $fet_amb_date_array[0];
        }

        if(in_array($id, $fets)){
            echo 'fet';
        } else {
            echo 'nofet';
        }

    }

    function titol_comprobacio($title, $curt = 'S'){
        if ($curt == 'S'){
            if (strlen($title) > 32) {
                $title = substr($title, 0, 29);
                $title .= '...';
                return $title;
            } else {
                return $title;
            }
        } else {
            return $title;
        }
    }

    function tipus_control($title){
        if (strlen($title) > 41) {
            echo substr($title, 0, 38);
            echo '...';
        } else {
            echo $title;
        }
    }

    function amb_lot_arxiu($codi, $num){
        if(file_exists("execucio/{$codi}_{$num}.php")){
            include "execucio/{$codi}_{$num}.php";
        }
    }

    function name($codi) {
        $query = mysql_query("SELECT titol FROM procesos WHERE codi = '".$codi."' ") or die();
        $proces = mysql_fetch_array($query);

        return $proces['titol'];
    }

}

$execucio = new Execucio();
```

```php
<?php

class db {

    function query($query) {
        $data = mysql_query($query) or die("MySQL Error: " . mysql_error());
        return $data;
    }

    function fetch($mysql_query) {
        $data = mysql_fetch_array($mysql_query);
        return $data;
    }

    function escape($value) {
        $data = mysql_real_escape_string($value);
        return $data;
    }
}

$db = new db();
```

```php
<?php

class Acabada {

    function data($codi, $param){
        $data_query = mysql_query("SELECT * FROM acabada_data WHERE codi = '".$codi."'");

        $data_array = mysql_fetch_array($data_query);

        $data = explode(' - ', $data_array[$param]);

        return $data;
    }

    function lotazos($codi, $project_id){

        $query = "SELECT lotazos FROM acabada WHERE codi = '".$codi."' AND project_id = '".
$project_id."'";

        $lotazos_query = mysql_query($query);

        $lotazos_array = mysql_fetch_array($lotazos_query);

        $lotazos = explode(', ', $lotazos_array[0]);

        return $lotazos;

    }

    function amb_lotazos($codi, $project_id){

        $query = "SELECT lotazos FROM acabada WHERE codi = '".$codi."' AND project_id = '".
$project_id."'";

        $lotazos_query = mysql_query($query);

        $lotazos_array = mysql_fetch_array($lotazos_query);

        if($lotazos_array == NULL){
            return NULL;
        } else {
            return 1;
        }

    }

    function cifrat($codi){
        $cifrat_query = mysql_query("SELECT cifrat FROM acabada_data WHERE codi = '".$codi."'");
        $cifrat_array = mysql_fetch_array($cifrat_query);

        $cifrat_previ = explode(' - ', $cifrat_array[0]);

        foreach ($cifrat_previ as $cp){
            $cifrat[] = explode(', ', $cp);
        }

        return $cifrat;
    }

    function checked($codi_lot, $codi, $project_id){
        $query = "SELECT lots_fets FROM acabada WHERE codi = '".$codi."' AND project_id = '".
$project_id."'";

        $mysql_query = mysql_query($query);

        $fetch_array = mysql_fetch_array($mysql_query);

        $lots = explode (', ', $fetch_array[0]);

        if(in_array($codi_lot, $lots)){
            echo 'checked';
        }
```

```
    }

    function clase($codi_lot, $codi, $project_id){
        $query = "SELECT lots_fets FROM acabada WHERE codi = '".$codi."' AND project_id = '".
$project_id."'";

        $mysql_query = mysql_query($query);

        $fetch_array = mysql_fetch_array($mysql_query);

        $lots = explode (', ', $fetch_array[0]);

        if(in_array($codi_lot, $lots)){
            echo 'fet';
        } else {
            echo 'nofet';
        }
    }

    function amb_lot_arxiu($codi){
        if(file_exists("acabada/{$codi}.php")){
            include "acabada/{$codi}.php";
        }
    }

    function sense_lot_arxiu($codi){
        if(file_exists("acabada/{$codi}_sin.php")){
            include "acabada/{$codi}_sin.php";
        }
    }

}

$acabada = new Acabada();
```

```php
<?php

require "includes/initialize.php";

$user_id = $_GET['id'];
$project_id = $_GET['project_id'];
$control_de = $_GET['control_de'];

if(user_validates($user_id, $project_id)) { ?>

<?php get_header('list'); ?>

<?php top_bar('Afegir procés', "project_control.php?id={$user_id}&project_id={$project_id}&control_de=
{$control_de}"); ?>

<div id="content">

<?php

//MARCA CHECK SI EL PROCESO YA ESTÁ SELECCIONADO

//Carga datos del projecto, donde estan los controles que existen
$project_query = $db->query("SELECT * FROM project WHERE id = '".$project_id."' ");
$project = $db->fetch($project_query);

//Crea un array con los procesos que ya existen
$procesos_marcats = explode(".", $project[$control_de]); ?>

<form method="post" action="project_control.php?id=<?php echo urlencode($user_id); ?>&project_id=<?php
echo urlencode($project_id); ?>&control_de=<?php echo $control_de; ?>" name="add_proces"
id="add_proces">

<ul>

<li class="title">01 Moviments de terres</li>

<li><p><span class="name">
<input name="PQ_0111" type="checkbox" <?php checked(PQ_0111); ?> />
Enderrocs
</span></p></li>

<li><p><span class="name">
<input name="PQ_0121" type="checkbox" <?php checked(PQ_0121); ?> />
Excavacions
</span></p></li>

<li><p><span class="name">
<input name="PQ_0122" type="checkbox" <?php checked(PQ_0122); ?> />
Reblerts
</span></p></li>

<li><p><span class="name">
<input name="PQ_0127" type="checkbox" <?php checked(PQ_0127); ?> />
Rases i pous
</span></p></li>

<li><p><span class="name">
<input name="PQ_0131" type="checkbox" <?php checked(PQ_0131); ?> />
Estrebades i apuntala...
</span></p></li>

<li><p><span class="name">
<input name="PQ_0141" type="checkbox" <?php checked(PQ_0141); ?> />
Transport de terra i runa
</span></p></li>

<li><p><span class="name">
<input name="PQ_0161" type="checkbox" <?php checked(PQ_0161); ?> />
Eixugades i esgotaments
</span></p></li>

<li><p><span class="name">
```

```
<input name="PQ_0162" type="checkbox" <?php checked(PQ_0162); ?> />
Trencaments hidràulics
</span></p></li>

<li><p><span class="name">
<input name="PQ_0171" type="checkbox" <?php checked(PQ_0171); ?> />
Sòls estabilitzats amb...
</span></p></li>

<li><p><span class="name">
<input name="PQ_0181" type="checkbox" <?php checked(PQ_0181); ?> />
Ancoratges al terreny
</span></p></li>

<li><p><span class="name">
<input name="PQ_0191" type="checkbox" <?php checked(PQ_0191); ?> />
Gestió de residus
</span></p></li>

<li class="title">02 Fonaments</li>

<li><p><span class="name">
<input name="PQ_021A" type="checkbox" <?php checked(PQ_021A); ?> />
Pilons in situ
</span></p></li>

<li><p><span class="name">
<input name="PQ_0212" type="checkbox" <?php checked(PQ_0212); ?> />
Fonaments i elements...
</span></p></li>

<li><p><span class="name">
<input name="PQ_0213" type="checkbox" <?php checked(PQ_0213); ?> />
Sabates
</span></p></li>

<li><p><span class="name">
<input name="PQ_0214" type="checkbox" <?php checked(PQ_0214); ?> />
Pous de fonamentació
</span></p></li>

<li><p><span class="name">
<input name="PQ_0215" type="checkbox" <?php checked(PQ_0215); ?> />
Murs
</span></p></li>

<li><p><span class="name">
<input name="PQ_0216" type="checkbox" <?php checked(PQ_0216); ?> />
Pantalles de formigo in...
</span></p></li>

<li><p><span class="name">
<input name="PQ_0217" type="checkbox" <?php checked(PQ_0217); ?> />
Pantalles prefabricades...
</span></p></li>

<li><p><span class="name">
<input name="PQ_0218" type="checkbox" <?php checked(PQ_0218); ?> />
Lloses de fonamentació
</span></p></li>

<li><p><span class="name">
<input name="PQ_0219" type="checkbox" <?php checked(PQ_0219); ?> />
Pilons de clavaments...
</span></p></li>

<li class="title">03 Fusta</li>

<li><p><span class="name">
<input name="PQ_0311" type="checkbox" <?php checked(PQ_0311); ?> />
Estructura de fusta
</span></p></li>
```

```
<li class="title">04 Formigó</li>

<li><p><span class="name">
<input name="PQ_0411" type="checkbox" <?php checked(PQ_0411); ?> />
Estructures de formigó
</span></p></li>

<li><p><span class="name">
<input name="PQ_0413" type="checkbox" <?php checked(PQ_0413); ?> />
Forjats amb elements pre...
</span></p></li>

<li class="title">05 Acer</li>

<li><p><span class="name">
<input name="PQ_0501" type="checkbox" <?php checked(PQ_0501); ?> />
Protecció d'estructures
</span></p></li>

<li><p><span class="name">
<input name="PQ_0511" type="checkbox" <?php checked(PQ_0511); ?> />
Estructures metàl·liques
</span></p></li>

<li class="title">06 Fàbrica</li>

<li><p><span class="name">
<input name="PQ_06X1" type="checkbox" <?php checked(PQ_06X1); ?> />
Est. de maçoneria
</span></p></li>

<li><p><span class="name">
<input name="PQ_0612" type="checkbox" <?php checked(PQ_0612); ?> />
Est. de bloc de formigó
</span></p></li>

<li><p><span class="name">
<input name="PQ_0613" type="checkbox" <?php checked(PQ_0613); ?> />
Est. de fab. de maó
</span></p></li>

<li><p><span class="name">
<input name="PQ_0616" type="checkbox" <?php checked(PQ_0616); ?> />
Est. de blocs d'argila
</span></p></li>

<li class="title">07 Prefabricada</li>

<li><p><span class="name">
<input name="PQ_0711" type="checkbox" <?php checked(PQ_0711); ?> />
Est. pref. de formigó
</span></p></li>

<li class="title">08 Cobertes</li>

<li><p><span class="name">
<input name="PQ_08A1" type="checkbox" <?php checked(PQ_08A1); ?> />
Teulada de teules
</span></p></li>

<li><p><span class="name">
<input name="PQ_08B1" type="checkbox" <?php checked(PQ_08B1); ?> />
Teulada de zinc
</span></p></li>

<li><p><span class="name">
<input name="PQ_08C1" type="checkbox" <?php checked(PQ_08C1); ?> />
Teu. de plaques asfàltiques
</span></p></li>

<li><p><span class="name">
```

```
<input name="PQ_0811" type="checkbox" <?php checked(PQ_0811); ?> />
Terrat ajardinat
</span></p></li>

<li><p><span class="name">
<input name="PQ_0831" type="checkbox" <?php checked(PQ_0831); ?> />
Terrat
</span></p></li>

<li><p><span class="name">
<input name="PQ_0841" type="checkbox" <?php checked(PQ_0841); ?> />
Cobertes translucides
</span></p></li>

<li><p><span class="name">
<input name="PQ_0851" type="checkbox" <?php checked(PQ_0851); ?> />
Teulada de fibrociment
</span></p></li>

<li><p><span class="name">
<input name="PQ_0861" type="checkbox" <?php checked(PQ_0861); ?> />
Teulada galvanitzada
</span></p></li>

<li><p><span class="name">
<input name="PQ_0871" type="checkbox" <?php checked(PQ_0871); ?> />
Teu. d'aliatges lleugers
</span></p></li>

<li><p><span class="name">
<input name="PQ_0881" type="checkbox" <?php checked(PQ_0881); ?> />
Teulada de pissara
</span></p></li>

<li><p><span class="name">
<input name="PQ_0891" type="checkbox" <?php checked(PQ_0891); ?> />
Teulada de sintètics
</span></p></li>

<li class="title">09 Tancaments exteriors</li>

<li><p><span class="name">
<input name="PQ_0911" type="checkbox" <?php checked(PQ_0911); ?> />
Tanc. elements de formigó
</span></p></li>

<li><p><span class="name">
<input name="PQ_0921" type="checkbox" <?php checked(PQ_0921); ?> />
Tancaments ceràmics
</span></p></li>

<li><p><span class="name">
<input name="PQ_0931" type="checkbox" <?php checked(PQ_0931); ?> />
Parets de carreus
</span></p></li>

<li><p><span class="name">
<input name="PQ_0941" type="checkbox" <?php checked(PQ_0941); ?> />
Paredats
</span></p></li>

<li><p><span class="name">
<input name="PQ_0961" type="checkbox" <?php checked(PQ_0961); ?> />
Murs cortina
</span></p></li>

<li><p><span class="name">
<input name="PQ_0971" type="checkbox" <?php checked(PQ_0971); ?> />
Tancaments de panells
</span></p></li>

<li><p><span class="name">
```

```
<input name="PQ_0981" type="checkbox" <?php checked(PQ_0981); ?> />
Tanc. de peçes de vidre
</span></p></li>

<li class="title">10 Tancaments interiors</li>

<li><p><span class="name">
<input name="PQ_1021" type="checkbox" <?php checked(PQ_1021); ?> />
Envans de maó
</span></p></li>

<li><p><span class="name">
<input name="PQ_1031" type="checkbox" <?php checked(PQ_1031); ?> />
En. de plaques i panells
</span></p></li>

<li class="title">13 Acabats</li>

<li><p><span class="name">
<input name="PQ_13A1" type="checkbox" <?php checked(PQ_13A1); ?> />
Teixits
</span></p></li>

<li><p><span class="name">
<input name="PQ_13B1" type="checkbox" <?php checked(PQ_13B1); ?> />
Fals sostre
</span></p></li>

<li><p><span class="name">
<input name="PQ_1311" type="checkbox" <?php checked(PQ_1311); ?> />
Enrajolats
</span></p></li>

<li><p><span class="name">
<input name="PQ_1321" type="checkbox" <?php checked(PQ_1321); ?> />
Aplacats
</span></p></li>

<li><p><span class="name">
<input name="PQ_1331" type="checkbox" <?php checked(PQ_1331); ?> />
Arrebossat
</span></p></li>

<li><p><span class="name">
<input name="PQ_1341" type="checkbox" <?php checked(PQ_1341); ?> />
Estucats i monocapes
</span></p></li>

<li><p><span class="name">
<input name="PQ_1351" type="checkbox" <?php checked(PQ_1351); ?> />
Guarnits i enlluits
</span></p></li>

<li><p><span class="name">
<input name="PQ_1371" type="checkbox" <?php checked(PQ_1371); ?> />
Revestiment flexible
</span></p></li>

<li><p><span class="name">
<input name="PQ_1381" type="checkbox" <?php checked(PQ_1381); ?> />
Revestiment lleuger
</span></p></li>

<li><p><span class="name">
<input name="PQ_1391" type="checkbox" <?php checked(PQ_1391); ?> />
Pintures
</span></p></li>

<li class="title">14 Paviments</li>

<li><p><span class="name">
<input name="PQ_14A1" type="checkbox" <?php checked(PQ_14A1); ?> />
```

```
Paviment técnic
</span></p></li>

<li><p><span class="name">
<input name="PQ_14B1" type="checkbox" <?php checked(PQ_14B1); ?> />
Pav. de peces rígides
</span></p></li>

<li><p><span class="name">
<input name="PQ_14C1" type="checkbox" <?php checked(PQ_14C1); ?> />
Paviment flexible
</span></p></li>

<li><p><span class="name">
<input name="PQ_14D1" type="checkbox" <?php checked(PQ_14D1); ?> />
Paviment continu
</span></p></li>

<li><p><span class="name">
<input name="PQ_14E1" type="checkbox" <?php checked(PQ_14E1); ?> />
Vorera (Urbanització)
</span></p></li>

<li><p><span class="name">
<input name="PQ_14F1" type="checkbox" <?php checked(PQ_14F1); ?> />
Vials (Urbanització)
</span></p></li>

<li><p><span class="name">
<input name="PQ_1421" type="checkbox" <?php checked(PQ_1421); ?> />
Subbases i recrescudes
</span></p></li>

<li><p><span class="name">
<input name="PQ_1431" type="checkbox" <?php checked(PQ_1431); ?> />
Solera de formigó
</span></p></li>

<li class="title">15 Elements practicables</li>

<li><p><span class="name">
<input name="PQ_15P1" type="checkbox" <?php checked(PQ_15P1); ?> />
Vidres
</span></p></li>

<li><p><span class="name">
<input name="PQ_1511" type="checkbox" <?php checked(PQ_1511); ?> />
Finestra i balconera
</span></p></li>

<li><p><span class="name">
<input name="PQ_1531" type="checkbox" <?php checked(PQ_1531); ?> />
Portes
</span></p></li>

<li class="title">16 Protecció</li>

<li><p><span class="name">
<input name="PQ_1611" type="checkbox" <?php checked(PQ_1611); ?> />
Baranes
</span></p></li>

<li><p><span class="name">
<input name="PQ_1641" type="checkbox" <?php checked(PQ_1641); ?> />
Senyal. de seguretat
</span></p></li>

<li class="title">17 Sanejament</li>

<li><p><span class="name">
<input name="PQ_1711" type="checkbox" <?php checked(PQ_1711); ?> />
Xarxa d'evacuació d'aigües
```

```
</span></p></li>

<li><p><span class="name">
<input name="PQ_1741" type="checkbox" <?php checked(PQ_1741); ?> />
Drenatge
</span></p></li>

<li><p><span class="name">
<input name="PQ_1761" type="checkbox" <?php checked(PQ_1761); ?> />
Inst. de rec. de residus
</span></p></li>

<li><p><span class="name">
<input name="PQ_1771" type="checkbox" <?php checked(PQ_1771); ?> />
Xarxa de clavagueram
</span></p></li>

<li><p><span class="name">
<input name="PQ_1781" type="checkbox" <?php checked(PQ_1781); ?> />
Elem. depuració d'aigües...
</span></p></li>

<li class="title">18 Calefacció</li>

<li><p><span class="name">
<input name="PQ_1811" type="checkbox" <?php checked(PQ_1811); ?> />
Instal. de calefacció
</span></p></li>

<li><p><span class="name">
<input name="PQ_1821" type="checkbox" <?php checked(PQ_1821); ?> />
Instal. de ventilació
</span></p></li>

<li class="title">19 Climatització</li>

<li><p><span class="name">
<input name="PQ_1911" type="checkbox" <?php checked(PQ_1911); ?> />
Instal. de climatització
</span></p></li>

<li class="title">20 Electricitat</li>

<li><p><span class="name">
<input name="PQ_2011" type="checkbox" <?php checked(PQ_2011); ?> />
Instal. electrica
</span></p></li>

<li><p><span class="name">
<input name="PQ_2013" type="checkbox" <?php checked(PQ_2013); ?> />
Centres de transformació
</span></p></li>

<li><p><span class="name">
<input name="PQ_2021" type="checkbox" <?php checked(PQ_2021); ?> />
Contribució fotovoltaica
</span></p></li>

<li><p><span class="name">
<input name="PQ_2031" type="checkbox" <?php checked(PQ_2031); ?> />
Enllumenat
</span></p></li>

<li class="title">21 Gas</li>

<li><p><span class="name">
<input name="PQ_2111" type="checkbox" <?php checked(PQ_2111); ?> />
Instal. de gas
</span></p></li>

<li class="title">22 Aigua</li>
```

```
<li><p><span class="name">
<input name="PQ_2211" type="checkbox" <?php checked(PQ_2211); ?> />
Instal. d'aigua
</span></p></li>

<li><p><span class="name">
<input name="PQ_2213" type="checkbox" <?php checked(PQ_2213); ?> />
Instal. solar térmica
</span></p></li>

<li class="title">23 Foc</li>

<li><p><span class="name">
<input name="PQ_2311" type="checkbox" <?php checked(PQ_2311); ?> />
Instal. de protecció al foc
</span></p></li>

<li class="title">24 Parallamp</li>

<li><p><span class="name">
<input name="PQ_2411" type="checkbox" <?php checked(PQ_2411); ?> />
Parallamps
</span></p></li>

<li class="title">25 Telecomunicació</li>

<li><p><span class="name">
<input name="PQ_2511" type="checkbox" <?php checked(PQ_2511); ?> />
Inst. de telecomunicació
</span></p></li>

<li class="title">26 Elevadors</li>

<li><p><span class="name">
<input name="PQ_2621" type="checkbox" <?php checked(PQ_2621); ?> />
Ascensors
</span></p></li>

<li class="title">28 Altres</li>

<li><p><span class="name">
<input name="PQ_2811" type="checkbox" <?php checked(PQ_2811); ?> />
Aparells sanitaris
</span></p></li>

<li class="form">
<input type="submit" name="submit" id="submit" value="Guardar dades" />
</li>

</ul>

</form>

</div>

<?php get_footer();

} else { user_not_allowed(); } ?>
```