

```
<?php
/* Copyright (C) 2001-2007 Rodolphe Quiedeville <rodolphe@quiedeville.org>
 * Copyright (C) 2004-2011 Laurent Destailleur <eldy@users.sourceforge.net>
 * Copyright (C) 2005 Eric Seigne <eric.seigne@ryxeo.com>
 * Copyright (C) 2005-2012 Regis Houssin <regis.houssin@capnetworks.com>
 * Copyright (C) 2006 Andre Cianfarani <acianfa@free.fr>
 * Copyright (C) 2011 Juanjo Menent <jmenent@2byte.es>
 * Copyright (C) 2013-2020 Charlene BENKE <charlie@patas-monkey.com>
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

/**
 * \file htdocs/factory/fiche.php
 * \ingroup factory
 * \brief Page des Ordres de fabrication sur la fiche produit
 */

$res=@include("../main.inc.php"); // For root directory
if (! $res && file_exists($_SERVER['DOCUMENT_ROOT']."/main.inc.php"))
    $res=@include($_SERVER['DOCUMENT_ROOT']."/main.inc.php"); // Use on dev env only
if (! $res) $res=@include("../../main.inc.php"); // For "custom" directory

require_once DOL_DOCUMENT_ROOT."/core/lib/product.lib.php";
require_once DOL_DOCUMENT_ROOT."/product/class/product.class.php";
require_once DOL_DOCUMENT_ROOT."/product/stock/class/entrepot.class.php";
require_once DOL_DOCUMENT_ROOT."/categories/class/categorie.class.php";

require_once DOL_DOCUMENT_ROOT."/core/class/html.formfile.class.php";
require_once DOL_DOCUMENT_ROOT."/core/class/extrafields.class.php";
require_once DOL_DOCUMENT_ROOT."/core/lib/date.lib.php";
require_once DOL_DOCUMENT_ROOT."/core/lib/files.lib.php";

dol_include_once('/factory/class/factory.class.php');
dol_include_once('/factory/core/lib/factory.lib.php');

if (! empty($conf->global->FACTORY_ADDON)
    && is_readable(dol_buildpath("/factory/core/modules/factory/".$conf->global-
>FACTORY_ADDON.".php")))
    dol_include_once("/factory/core/modules/factory/".$conf->global-
>FACTORY_ADDON.".php");

$langs->load("bills");
$langs->load("products");
$langs->load("stocks");
$langs->load("factory@factory");

$id=GETPOST('id', 'int');
$ref=GETPOST('ref', 'alpha');
$action=GETPOST('action', 'alpha');

$fk_product=GETPOST('fk_product');
```

```
$qty_planned=GETPOST('qty_planned');
$fk_entrepot=GETPOST('fk_entrepot');

$confirm=GETPOST('confirm', 'alpha');
$cancel=GETPOST('cancel', 'alpha');
$keysearch=GETPOST('keysearch');
$parent=GETPOST('parent');

// Security check
//if (! empty($user->societe_id)) $socid=$user->societe_id;
$fieldvalue = (! empty($id) ? $id : (! empty($ref) ? $ref : ''));
$fieldtype = (! empty($ref) ? 'ref' : 'rowid');
$result = restrictedArea($user, 'factory');

$msg = '';

$product = new Product($db);
$factory = new Factory($db);
$entrepot = new Entrepot($db);
$extrafields = new ExtraFields($db);

$form = new Form($db);

if ($action=="create"){

    // on vérifie qu'un produit ou un BOM a été sélectionné
    if (GETPOST("productid") > 0 || GETPOST("bomid") > 0) {
        // on redirige vers la creation sur la fiche produit
        $szurl="product/fiche.php?action=createof";
        $szurl.="&id=".GETPOST("productid");
        $szurl.="&bomid=".GETPOST("bomid");
        $szurl.="&entrepotid=".GETPOST("entrepotid");
        $szurl.="&nbToBuild=".GETPOST("nbToBuild");
        $szurl.="&plannedstarthour=".GETPOST("plannedstarthour");
        $szurl.="&plannedstartmin=".GETPOST("plannedstartmin");
        $szurl.="&plannedstartday=".GETPOST("plannedstartday");
        $szurl.="&plannedstartmonth=".GETPOST("plannedstartmonth");
        $szurl.="&plannedstartyear=".GETPOST("plannedstartyear");
        $szurl.="&plannedendhour=".GETPOST("plannedendhour");
        $szurl.="&plannedendmin=".GETPOST("plannedendmin");
        $szurl.="&plannedendday=".GETPOST("plannedendday");
        $szurl.="&plannedendmonth=".GETPOST("plannedendmonth");
        $szurl.="&plannedendyear=".GETPOST("plannedendyear");
        $szurl.="&workloadhour=".GETPOST("workloadhour");
        $szurl.="&workloadmin=".GETPOST("workloadmin");

        header("Location: ".$szurl);

        exit;
    } else {
        $msg = '<div class="error">'. $langs->trans("NeedBOMorProductSelected").'</div>';
        $action="add";
    }
}

$productid=0;
if ($id || $ref) {
    // l'of, l'entrepot et le produit associé
    $result = $factory->fetch($id, $ref);
    if (!$id) $id = $factory->id;
}
```

```
$result = $product->fetch($factory->fk_product);
$productid= $factory->fk_product;

$result = $entrepot->fetch($factory->fk_entrepot);
$entrepotid= $factory->fk_entrepot;

//var_dump($factory);
}

// Initialize technical object to manage hooks of thirdparties. Note that conf-
>hooks_modules contains array array
$hookmanager->initHooks(array('factorycard', 'globalcard'));

$parameters = array('product' => $product);
// Note that $action and $object may have been modified by some hooks
$reshook = $hookmanager->executeHooks('doActions', $parameters, $factory, $action);
if ($reshook < 0) setEventMessages($hookmanager->error, $hookmanager->errors,
'errors');

/*
 * Actions
 */

if (empty($reshook)) {
// mise à jour des composants
if ($action == 'add_prod'
&& $cancel <> $langs->trans("Cancel")
&& $factory->statut == 0 // seulement si le module est à l'état brouillon
&& $user->rights->factory->creer ) {
$error=0;
for ($i=0;$i < $_POST["max_prod"];$i++) {
//print "<br> ".$i.: " ".$_POST["prod_id_chk".$i];
if ($_POST["prod_id_chk".$i] != "") {
if ($factory->add_componentOF(
$id,
$_POST["prod_id_".$i],
$_POST["prod_qty_".$i], 0, 0,
$_POST["prod_id_globalchk".$i],
$_POST["descComposant".$i]
) > 0)
$action = 'edit';
else {
$error++;
$action = 're-edit';
if ($factory->error == "isFatherOfThis")
$msg = '<div class="error">'.$langs-
>trans("ErrorAssociationIsFatherOfThis").'</div>';
else
$msg=$factory->error;
}
} else {
if ($factory->del_componentOF($id, $_POST["prod_id_".$i]) > 0)
$action = 'edit';
else {
$error++;
$action = 're-edit';
$msg=$product->error;
}
}
}
}
if (! $error)
$action="";
```

```
    } elseif ($action == 'confirm_clone' && $confirm == 'yes' && $user->rights->factory->creer) {
        // Action clone object
        if (1 == 0 && ! GETPOST('clone_content') && ! GETPOST('clone_receivers'))
            setEventMessage($langs->trans("NoCloneOptionsSpecified"), 'errors');
        else {
            if ($factory->id > 0) {
                // Because createFromClone modifies the object, we must clone it so that we
                // can restore it later
                $orig = dol_clone($factory);

                $result=$factory->createFromClone($fk_product, $qty_planned, $fk_entrepot);
                if ($result > 0) {
                    header("Location: ".$_SERVER['PHP_SELF'].'?id='.$result);
                    exit;
                } else {
                    setEventMessage($factory->error, 'errors');
                    $mesg= '<div class="error">'.$factory->error.'</div>';
                    $factory = $orig;
                    $action='';
                }
            }
        }
    }
    } elseif ($cancel == $langs->trans("Cancel")) {
        $action = '';
        Header("Location: fiche.php?id=".$_POST["id"]);
        exit;
    } elseif (substr($action, 0, 7) == 'setExFi' && $user->rights->factory->creer) {
        $keyExFi= substr($action, 7);
        $res=$factory->fetch_optionals($factory->id, $extralabels);
        $factory->array_options["options_".$keyExFi]=$_POST["options_".$keyExFi];
        $factory->insertExtraFields();
        $action = "";
    } elseif ($action == 'seteditdatestartmade') {
        $datestartmade=dol_mktime(
            '23', '59', '59',
            $_POST["datestartmademonth"],
            $_POST["datestartmadeday"],
            $_POST["datestartmadeyear"]
        );
        // $factory->fetch($id);
        $result=$factory->set_datestartmade($user, $datestartmade);
        if ($result < 0)
            dol_print_error($db, $factory->error);
        $action = "";
    } elseif ($action == 'seteditdatestartplanned') {
        $datestartplanned=dol_mktime(
            '23', '59', '59',
            $_POST["datestartplannedmonth"],
            $_POST["datestartplannedday"],
            $_POST["datestartplannedyear"]
        );
    };

    // $factory->fetch($id);
    $result=$factory->set_datestartplanned($user, $datestartplanned);
    if ($result < 0)
        dol_print_error($db, $factory->error);
    $action = "";

    } elseif ($action == 'seteditdateendplanned') {
        $dateendplanned=dol_mktime(
            '23', '59', '59',
            $_POST["dateendplannedmonth"],
            $_POST["dateendplannedday"],
```

```
        $_POST["dateendplannedyear"]
    );

    // $factory->fetch($id);
    $result=$factory->set_dateendplanned($user, $dateendplanned);
    if ($result < 0)
        dol_print_error($db, $factory->error);
    $action = "";

} elseif ($action == 'seteditdurationplanned') {
    $dateendplanned =
GETPOST("duration_plannedhour")*3600+GETPOST("duration_plannedmin")*60;;

    // $factory->fetch($id);
    $result=$factory->set_durationplanned($user, $dateendplanned);
    if ($result < 0)
        dol_print_error($db, $factory->error);
    $action = "";

} elseif ($action == 'setdescription') {
    // $factory->fetch($id);
    $result=$factory->set_description($user, $_POST["description"]);
    if ($result < 0)
        dol_print_error($db, $factory->error);
    $action = "";

} elseif ($action == 'setentrepot') {
    // $factory->fetch($id);
    $result=$factory->set_entrepot($user, GETPOST("fk_entrepot"));
    if ($result < 0)
        dol_print_error($db, $factory->error);
    $action = "";

} elseif ($action == 'setquantity') {
    // $factory->fetch($id);
    $result=$factory->set_qtyplanned($user, GETPOST("qty_planned"));
    if ($result < 0) dol_print_error($db, $factory->error);
    $action = "";
} elseif ($action == 'bulddoc') {
    /*
     * Generate order document
     * define into /core/modules/factory/modules_factory.php
     */

    // Save last template used to generate document
    if (GETPOST('model')) $factory->setDocModel($user, GETPOST('model', 'alpha'));

    // Define output language
    $outputlangs = $langs;
    $newlang='';
    if ($conf->global->MAIN_MULTILANGS && empty($newlang) && !
empty($_REQUEST['lang_id']))
        $newlang=$_REQUEST['lang_id'];
    if ($conf->global->MAIN_MULTILANGS && empty($newlang))
        $newlang=$factory->client->default_lang;
    if (! empty($newlang)) {
        $outputlangs = new Translate("", $conf);
        $outputlangs->setDefaultLang($newlang);
    }
    $result=factory_create($db, $factory, $factory->modelpdf, $outputlangs,
$shidedetails, $shidedesc, $hideref);

    if ($result <= 0) {
        dol_print_error($db, $result);
    }
}
```

```
        exit;
    }
    $action = "";
} elseif ($action == 'cancelof') {
    $factory->statut = 3;
    $sql = "UPDATE ".MAIN_DB_PREFIX."factory ";
    $sql.= " SET fk_statut =3";
    $sql.= " WHERE rowid = ".$id;
    if ($db->query($sql)) {
        // Call trigger
        $result=$factory->call_trigger('FACTORY_CANCEL', $user);
        if ($result < 0) $error++;
        // on supprime les mouvements de stock ??
    }
    $action="";
} elseif ($action == 'remove_file') {
    // Remove file in doc form
    $langs->load("other");
    $upload_dir = $conf->factory->dir_output;
    $file = $upload_dir . '/' . GETPOST('file');
    $ret = dol_delete_file($file, 0, 0, 0, $factory);
    if ($ret) setEventMessage($langs->trans("FileWasRemoved", GETPOST('urlfile')));
    else setEventMessage($langs->trans("ErrorFailToDeleteFile", GETPOST('urlfile')),
'errors');
    $action="";
} elseif ($action == 'clone') {

    // pour l'affichage soit de la liste des produits, soit la recherche produit
    if (! empty($conf->use_javascript_ajax) && ! empty($conf->global-
>PRODUIT_USE_SEARCH_TO_SELECT)) {
        $placeholder='';
        $selected_input_value=$product->ref;

        // mode=1 means customers products

        $urloption='htmlname='.'fk_product'.'&outjson=1&price_level=0&type=&mode=1&status=1&f
placeholder="'. $langs->trans("RefOrLabel").'";
        if (! empty($conf->global->MAIN_HTML5_PLACEHOLDER)) $placeholder='
        if (! empty($conf->global->PRODUIT_CUSTOMER_PRICES) && !empty($socid))
        $urloption.='&socid='.$socid;
        $productselectlist= ajax_autocompleter(
            $factory->fk_product, 'fk_product',
            DOL_URL_ROOT.'/product/ajax/products.php',
            $urloption, $conf->global->PRODUIT_USE_SEARCH_TO_SELECT, 0
        );
        $productselectlist.= '<input type="text" size="20" name="search_fk_product"
id="search_fk_product"';
        $productselectlist.= ' value="'. $selected_input_value.' '.$placeholder.' />';
    } else
        $productselectlist= $form->select_produits_list($factory->fk_product,
'fk_product', '');

    // Create an array for form
    $formquestion = array(
        // 'text' => $langs->trans("ConfirmClone"),
        // array('type' => 'checkbox', 'name' => 'clone_content', 'label' => $langs-
>trans("CloneMainAttributes"), 'value' =>
        // 1),
        array ( 'type' => 'text',
            'size' => 3,
            'name' => 'qty_planned',
            'label' => $langs->trans("QtyPlannedToBuild"),
            'value' => $factory->qty_planned
```

```
    ),
    array ( 'type' => 'other',
            'name' => 'fk_product',
            'label' => $langs->trans("SelectProduct"),
            'value' => $productselectlist
    ),
    array ( 'type' => 'other',
            'name' => 'fk_entrepot',
            'label' => $langs->trans("SelectWarehouse"),
            'value' => select_entrepot_list($factory->fk_entrepot, 'fk_entrepot', 0, 1)
    )
);

// Confirmation du clonage de l'OF
$formconfirm= $form->formconfirm(
    $_SERVER["PHP_SELF"] . '?id=' . $factory->id,
    $langs->trans('CloneOF'),
    $langs->trans('ConfirmCloneOF'). $factory->ref,
    'confirm_clone', $formquestion, 'yes', 1, 240
);
} elseif ($action == 'updateprice') {
    // on modifie les prix
    // $prodsfather = $factory->getFather($factory->id); //Parent Products
    // $factory->get_sousproduits_arbo();
    // // Number of subproducts
    // $prods_arbo = $factory->get_arbo_each_prod();
    // // something wrong in recurs, change id of object
    // $factory->id = $id;
    $prods_arbo= $factory->getChildsOF($factory->id);

    // List of subproducts
    if (count($prods_arbo) > 0) {
        foreach ($prods_arbo as $value)
            $factory->updateOFprices(
                $value['id'], GETPOST("prod_pmp_". $value['id']),
                GETPOST("prod_price_". $value['id'])
            );
    }
    $action="";
} elseif ($action == 'getdefaultprice') {
    $factory->getdefaultprice(1); // mode factorydet
    $action="";
}

// Clone confirmation
if (! $formconfirm) {
    $parameters = array('lineid' => $lineid);
    // Note that $action and $object may have been modified by hook
    $formconfirm = $hookmanager->executeHooks('formConfirm', $parameters, $factory,
    $action);
}
}

/*
 * View
 */

$titre=$langs->trans("Factory");

llxHeader("", "", $langs->trans("CardFactory"));

if ($id || $ref) {
```

```
$extralabels=$extrafields->fetch_name_optionals_label('factory');

$res=$factory->fetch_optionals($factory->id, $extralabels);

$formfile = new FormFile($db);
$head=factory_prepare_head($factory, $user);
$picto="factory@factory";

dol_fiche_head($head, 'factoryorder', $titre, -1, $picto);

// Print form confirm
if ($formconfirm)
    print $formconfirm;

if ((int) DOL_VERSION >= 6)
    $urllink='list.php';
else
    $urllink='list-old.php';

$linkback = '<a href="'. $urllink. '?restore_lastsearch_values=1' . (!
empty($productid) ? '&productid=' . $productid : '') . '">' . $langs
>trans("BackToList") . '</a>';

if ((int) DOL_VERSION >= 5) {

    // factory card

    $morehtmlref='<div class="refidno">';

    // ajouter la date de création de l'OF

    // Ref product
    $morehtmlref.='<br>'. $langs->trans('Product') . ' : ' . $product->getNomUrl(1)."
". $product->label;
    if (empty($conf->global->MAIN_DISABLE_OTHER_LINK))
        $morehtmlref.=' (<a href="'. $urllink. '?productid='. $factory-
>fk_product.'">' . $langs->trans("OtherFactory"). '</a>')';

    // ref storage
    // rendre modifiable
    $morehtmlref.='<br><table class="nobordernopadding" width="100%"><tr><td>' . $langs
>trans("Warehouse"). '</td>';
    if ($action != 'editstock' && $factory->statut == 0) {
        $morehtmlref.='<td align="right"><a
href="'. $_SERVER["PHP_SELF"]. '?action=editstock&id='. $factory->id.'">';
        $morehtmlref.='img_edit($langs->trans('Modify'), 1). '</a> : </td>';
    }
    $morehtmlref.='<td>';
    if ($action == 'editstock') {
        $morehtmlref.='<form name="editstock"
action="'. $_SERVER["PHP_SELF"]. '?id='. $factory->id.'" method="post">';
        $morehtmlref.='<input type="hidden" name="token"
value="'. $_SESSION['newtoken']. '">';
        $morehtmlref.='<input type="hidden" name="action" value="setentrepot">';
        $morehtmlref.= select_entrepot_list($factory->fk_entrepot, 'fk_entrepot', 1, 1)
        $morehtmlref.='<input type="submit" class="button" value="'. $langs-
>trans('Modify'). '">';
        $morehtmlref.='</form>';
    } else {
        if ($factory->fk_entrepot >0)
            $morehtmlref.=$entrepot->getNomUrl(1)." - ".$entrepot->lieu." (" . $entrepot-
>zip.")" ;
    }
}
```



```
if (empty($conf->global->MAIN_DISABLE_OTHER_LINK))
    $morehtmlref.=' (<a href="'. $surlink.'?entrepotid='.$factory-
>fk_entrepot.'">'. $langs->trans("OtherFactory").'</a>');

$morehtmlref.='</td></tr>';
$morehtmlref.='</table>';

$morehtmlref.='</div>';

dol_banner_tab($factory, 'ref', $linkback, 1, 'ref', 'ref', $morehtmlref);

print '<div class="fichecenter">';
print '<div class="underbanner clearboth"></div>';

} else {

print '<table class="border" width="100%">';

// Reference OF
print '<tr><td width="20%">'. $langs->trans("Ref").'</td><td colspan=3>';
print $form->showrefnav($factory, 'ref', $linkback, 1, 'ref');
print '</td></tr>';

// produit
print '<tr><td width="20%">'. $langs->trans("Product").'</td><td colspan=3>';
print $product->getNomUrl(1);
print '</td></tr>';

// Lieu de stockage
print '<tr><td><table class="noborder nopadding" width="100%"><tr><td>'. $langs-
>trans("Warehouse").'</td>';
if ($action != 'editstock' && $factory->statut == 0) {
    print '<td align="right"><a
href="'. $_SERVER["PHP_SELF"].'?action=editstock&id='.$factory->id.'">';
    print img_edit($langs->trans('Modify'), 1).'</a></td>';
}
print '</tr></table></td><td colspan="3">';
if ($action == 'editstock') {
    print '<form name="editstock" action="'. $_SERVER["PHP_SELF"].'?id='.$factory-
>id.'" method="post">';
    print '<input type="hidden" name="token" value="'. $_SESSION['newtoken'].'">';
    print '<input type="hidden" name="action" value="setentrepot">';
    print select_entrepot_list($factory->fk_entrepot, 'fk_entrepot', 1, 1);
    print '<input type="submit" class="button" value="'. $langs->trans('Modify').'">';
    print '</form>';
} else {
    if ($factory->fk_entrepot > 0)
        print $entrepot->getNomUrl(1)." - ".$entrepot->lieu." (".$entrepot->zip.)" ;
}
print '</td></tr>';
print '</table>';
}

print '<table class="border" width="100%">';
print '<tr><td width=30% valign=top>';
print '<table id="tablelines" class="noborder noshadow" width="100%">';

print '<tr class="liste_titre nodrag nodrop"><th align=left colspan=2>';
print $langs->trans("ProductsAdditionalInfos");
print '</b></th></tr>';

print '<tr><td>'. $langs->trans("VATRate").'</td>';
```

```
print '<td>'.vatrate($product->tva_tx.($product->tva_npr?'*':''), true).'</tr>

// Price
print '<tr><td>'. $langs->trans("SellingPrice").'</td><td>';
if ($product->price_base_type == 'TTC') {
    print price($product->price_ttc).' ' . $langs->trans($product->price_base_type);
    $sale="";
} else {
    print price($product->price).' ' . $langs->trans($product->price_base_type);
    $sale=$product->price;
}
print '</td></tr>';

// Price minimum
print '<tr><td>'. $langs->trans("MinPrice").'</td><td>';
if ($product->price_base_type == 'TTC')
    print price($product->price_min_ttc).' ' . $langs->trans($product->price_base_type)
else
    print price($product->price_min).' ' . $langs->trans($product->price_base_type);
print '</td></tr>';

// stock physique
print '<tr><td>'. $langs->trans("PhysicalStock").'</td>';
$product->load_stock();
print '<td>'. $product->stock_reel.'</td></tr>';

// Status (to sell)
print '<tr><td>'. $langs->trans("Status").' (' . $langs->trans("Sell").')</td><td
colspan="2">';
print $product->getLibStatut(2, 0);
print '</td></tr>';

// Status (to buy)
print '<tr><td>'. $langs->trans("Status").' (' . $langs->trans("Buy").')</td><td
colspan="2">';
print $product->getLibStatut(2, 1);
print '</td></tr>';

print '</table></td>';

print '<td width=70% valign=top>';
print '<table id="tablelines" class="noborder noshadow" width="100%">';

print '<tr class="liste_titre nodrag nodrop"><th align=left colspan=2><b>';
print $langs->trans("OFAdditionalInfos");
print '</b></td></tr>';

// pour gerer le bom que depuis la v11 (stable)
if ((int) DOL_VERSION > 10 && $conf->global->MAIN_MODULE_BOM ) {
    require_once DOL_DOCUMENT_ROOT."/bom/class/bom.class.php";
    $bom_static = new BOM($db);
    $bom_static->fetch($factory->fk_bom);
    print '<tr><td valign=top >'. $langs->trans("BOMLink").'</td ><td colspan="3"
valign=top>';
    if ($factory->fk_bom >0)
        print $bom_static->getNomUrl(1);
    print '</td></tr>';
}

// Date start planned
print '<tr><td valign=top ><table class="nobordernopadding" width="100%"><tr>';
```

```
print '<td align=left >'. $langs->trans("FactoryDateStartPlanned");
if ($action != 'editdatestartplanned' && $factory->statut < 2) {
    print '<td valign=top align="right">';
    print '<a
href="'. $SERVER["PHP_SELF"].'?action=editdatestartplanned&id='.$factory->id.'">';
    print img_edit($langs->trans('Modify'), 1). '</a></td>';
}
print '</tr></table></td ><td colspan="3" valign=top>';
if ($action == 'editdatestartplanned') {
    print '<form name="editdatestartplanned"
action="'. $SERVER["PHP_SELF"].'?id='.$factory->id.'" method="post">';
    print '<input type="hidden" name="token" value="'. $SESSION['newtoken'].'">';
    print '<input type="hidden" name="action" value="seteditdatestartplanned">';
    print $form->select_date($factory->date_start_planned, 'datestartplanned', 0, 0,
'', "datestartplanned");
    print '<input type="submit" class="button" value="'. $langs->trans('Modify').'">';
    print '</form>';
} else
    print dol_print_date($factory->date_start_planned, 'day');
print '</td></tr>';

// Date start made
print '<tr><td valign=top ><table class="nobordernopadding" width="100%"><tr>';
print '<td align=left><b>'. $langs->trans("DateStartMade"). '</b><br></td>';

// c'est la saisie de cette date qui conditionne la validation ou pas de l'OF
if ($action != 'editdatestartmade' && $factory->statut < 2) {
    print '<td valign=top align="right">';
    print '<a
href="'. $SERVER["PHP_SELF"].'?action=editdatestartmade&id='.$factory->id.'">';
    print img_edit($langs->trans('Modify'), 1). '</a></td>';
}
print '</tr></table></td ><td colspan="3" valign=top>';
if ($action == 'editdatestartmade') {
    print '<form name="editdatestartmade"
action="'. $SERVER["PHP_SELF"].'?id='.$factory->id.'" method="post">';
    print '<input type="hidden" name="token" value="'. $SESSION['newtoken'].'">';
    print '<input type="hidden" name="action" value="seteditdatestartmade">';
    print $form->select_date($factory->date_start_made, 'datestartmade', 0, 0, '',
"datestartmade");
    print '<input type="submit" class="button" value="'. $langs->trans('Modify').'">';
    print '</form>';
}
else
    print dol_print_date($factory->date_start_made, 'day');
// pour gérer la mise en forme
if ($factory->date_start_made)
    print '<br>';
else
    print "<b><font color=red>". $langs->trans("DateStartMadeInfo"). "</font></b>";
print '</td></tr>';

// Date end planned
print '<tr><td><table class="nobordernopadding" width="100%">';
print '<tr><td>'. $langs->trans("FactoryDateEndPlanned"). '</td>';
if ($action != 'editdateendplanned' && $factory->statut == 0) {
    print '<td align="right">';
    print '<a
href="'. $SERVER["PHP_SELF"].'?action=editdateendplanned&id='.$factory->id.'">';
    print img_edit($langs->trans('Modify'), 1). '</a></td>';
}
print '</tr></table></td><td colspan="3">';
```

```
if ($action == 'editdateendplanned') {
    print '<form name="editdateendplanned"
action="'.$_SERVER["PHP_SELF"].'?id='.$factory->id.'" method="post">';
    print '<input type="hidden" name="token" value="'.$_SESSION['newtoken'].'">';
    print '<input type="hidden" name="action" value="seteditdateendplanned">';
    print $form->select_date($factory->date_end_planned, 'dateendplanned', 0, 0, '',
"dateendplanned");
    print '<input type="submit" class="button" value="'. $langs->trans('Modify').'">';
    print '</form>';
} else
    print dol_print_date($factory->date_end_planned, 'day');

print '</td></tr>';

print '<tr><td><table class="nobordernopadding" width="100%"><tr><td>'. $langs->
>trans("QuantityPlanned").'</td>';
    if ($action != 'editquantity' && $factory->statut == 0) {
        print '<td align="right"><a
href="'.$_SERVER["PHP_SELF"].'?action=editquantity&id='.$factory->id.'">';
        print img_edit($langs->trans('Modify'), 1).'</a></td>';
    }
    print '</tr></table></td><td colspan="3">';
    if ($action == 'editquantity') {
        print '<form name="editquantity" action="'.$_SERVER["PHP_SELF"].'?id='.$factory->
>id.'" method="post">';
        print '<input type="hidden" name="token" value="'.$_SESSION['newtoken'].'">';
        print '<input type="hidden" name="action" value="setquantity">';
        print '<input type="text" name="qty_planned" value="'. $factory->qty_planned.'">';
        print '<input type="submit" class="button" value="'. $langs->trans('Modify').'">';
        print '</form>';
    } else
        print $factory->qty_planned;

// Planned workload
print '<tr><td><table class="nobordernopadding" width="100%"><tr><td>'. $langs->
>trans("FactoryDurationPlanned").'</td>';
    if ($action != 'editdurationplanned' && $factory->statut == 0) {
        print '<td align="right"><a
href="'.$_SERVER["PHP_SELF"].'?action=editdurationplanned&id='.$factory->id.'">';
        print img_edit($langs->trans('Modify'), 1).'</a></td>';
    }
    print '</tr></table></td><td colspan="3">';
    if ($action == 'editdurationplanned') {
        print '<form name="editdurationplanned"
action="'.$_SERVER["PHP_SELF"].'?id='.$factory->id.'" method="post">';
        print '<input type="hidden" name="token" value="'.$_SESSION['newtoken'].'">';
        print '<input type="hidden" name="action" value="seteditdurationplanned">';
        print $form->select_duration('duration_planned', $factory->duration_planned, 0,
'text');
        print '<input type="submit" class="button" value="'. $langs->trans('Modify').'">';
        print '</form>';
    }
    else
        print convertSecondToTime($factory->duration_planned, 'allhourmin');
    print '</td></tr>';

// Other attributes
$parameters = array( 'colspan' => ' colspan="3"' );
// Note that $action and $object may have been modified by
$reshook = $hookmanager->executeHooks('formObjectOptions', $parameters, $factory,
$action);
// hook
if (empty($reshook) && ! empty($extrafields->attribute_label)) {
```

```
foreach ($extrafields->attribute_label as $key=>$label) {
    $value=(isset($_POST["options_".$key])?$_POST["options_".$key]:$factory->array_options["options_".$key]);

    print '<tr><td><table class="nobordernopadding"
width="100%"><tr><td>'.$label.'</td>';
    if ($action != 'ExFi'.'.$key && $factory->statut == 0) {
        print '<td align="right"><a
href="'.$_SERVER["PHP_SELF"].'?action=ExFi'.'.$key.'&id='.$factory->id.'">';
        print img_edit($langs->trans('Modify'), 1).'</td>';
    }
    print '</tr></table></td><td colspan="3">';
    if ($action == 'ExFi'.'.$key) {
        print '<form name="ExFi'.'.$key.'"
action="'.$_SERVER["PHP_SELF"].'?id='.$factory->id.'" method="post">';
        print '<input type="hidden" name="token" value="'.$SESSION['newtoken'].'">';
        print '<input type="hidden" name="action" value="setExFi'.'.$key.'">';
        print $extrafields->showInputField($key, $value);
        print '<input type="submit" class="button" value="'.$langs->trans('Modify').'">';
        print '</form>';
    } else
        print $extrafields->showOutputField($key, $value);

    print '</td></tr>'. "\n";
}

print '</table></td></tr>';

// Description
print '<tr><td valign=top >';
print '<table class="nobordernopadding" width="100%"><tr>';
print '<td valign=top >'.$langs->trans("Description").'</td>';
if ($action != 'editdescription' && ($factory->statut == 0 || $user->rights->factoryupdate)) {
    print '<td align="right" ><a
href="'.$_SERVER["PHP_SELF"].'?action=editdescription&id='.$factory->id.'">';
    print img_edit($langs->trans('Modify'), 1).'</td>';
}
print '</tr></table></td><td >';
if ($action == 'editdescription') {
    print '<form name="editdescription" action="'.$_SERVER["PHP_SELF"].'?id='.$factory->id.'" method="post">';
    print '<input type="hidden" name="token" value="'.$SESSION['newtoken'].'">';
    print '<input type="hidden" name="action" value="setdescription">';
    require_once DOL_DOCUMENT_ROOT.'/core/class/doleditor.class.php';
    $doleditor=new DolEditor("description", $factory->description, '', '100',
'dolibarr_notes', 'In', 0, true, true, '20', '100');
    print $doleditor->Create(1);

    // print '<textarea name="description" wrap="soft" cols="120"
rows="'.ROWS_4.'">'.$factory->description.'</textarea>';
    print '<input type="submit" class="button" value="'.$langs->trans('Modify').'">';
    print '</form>';
} else
    print str_replace(array("\r\n", "\n"), "<br>", $factory->description);
print '</td></tr>';
print '</table>';

print '</td></tr></table>';
print '<br>';
```

```
// indique si on a d[un]j[un] une composition de pr[es]ente ou pas
$compositionpresente=0;

$prods_arbo = $factory->getChildsOF($id);

// on travaille avec les valeurs conserv[ees]
if (false) {
    $factory->id = $product->id;
    $factory->get_sousproduits_arbo();
    // Number of subproducts
    $prods_arbo = $factory->get_arbo_each_prod();
    // something wrong in recurs, change id of object
    $factory->id = $product->id;
}

print_fiche_titre($langs->trans("FactorisedProductsNumber").' : '.count($prods_arbo)
'', '');

// List of subproducts
if (count($prods_arbo) > 0) {
    $compositionpresente=1;
    print '<table class="border" >';
    print '<tr class="liste_titre">';
    print '<td class="liste_titre" width=100px align="left">'.$langs-
>trans("Ref").'</td>';
    print '<td class="liste_titre" width=200px align="left">'.$langs-
>trans("Label").'</td>';
    print '<td class="liste_titre" width=50px align="center">'.$langs-
>trans("QtyNeedOF").'</td>';
    // on affiche la colonne stock m[eme] si cette fonction n'est pas active
    print '<td class="liste_titre" width=50px align="center">'.$langs-
>trans("QtyOfWarehouse").'</td>';
    print '<td class="liste_titre" width=100px align="center">'.$langs-
>trans("QtyOrder").'</td>';
    if ($user->rights->factory->showprice) {
        if ($conf->stock->enabled) { // we display vwap titles
            print '<td class="liste_titre" width=100px align="center">'.$langs-
>trans("UnitPmp").'</td>';
            print '<td class="liste_titre" width=100px align="center">'.$langs-
>trans("CostPmpHT").'</td>';
        } else { // we display price as latest purchasing unit price title
            print '<td class="liste_titre" width=100px align="center">'.$langs-
>trans("UnitHA").'</td>';
            print '<td class="liste_titre" width=100px align="center">'.$langs-
>trans("CostHA").'</td>';
        }
        print '<td class="liste_titre" width=100px align="right">'.$langs-
>trans("SellPrice").'</td>';
        print '<td class="liste_titre" width=100px align="right">'.$langs-
>trans("FactorySellingPriceHT").'</td>';
        print '<td class="liste_titre" width=100px align="right">'.$langs-
>trans("UnitProfitAmount").'</td>';
        print '<td class="liste_titre" width=100px align="right">'.$langs-
>trans("ProfitAmount").'</td>';
    }

    print '</tr>';
    $mntTot=0;
    $pmpTot=0;
    $btopChild = false;
    foreach ($prods_arbo as $value) {
        // verify if product have child then display it after the product name
```

```
$tmpChildArbo=$factory->getChilddsArbo($value['id']);
$nbChildArbo="";
if (count($tmpChildArbo) > 0) {
    $nbChildArbo=" (" .count($tmpChildArbo).")";
    $btopChild = true;
}
$nbcolspan=3;
print '<tr>';
print '<td align="left">'. $factory->getNomUrlFactory($value['id'], 1,
'fiche'). $nbChildArbo;
print $factory->PopupProduct($value['id']);
print '</td>';
print '<td align="left" title="'. $value['description']. '>';
if ($value['description'])
    print "<a href=# onclick=\"\$('.'.detailligneComp".$value['id'].").toggle();\"
>.img_picto(\"\", \"edit_add\").\"</a>&nbsp;";
print $value['label']. '</td>';
print '<td align="center">'. $value['qtyplanned'];
if ($value['globalqty'] == 1)
    print "&nbsp;G";
print '</td>';

if ($conf->stock->enabled) {
    $nbcolspan+=1;
    if ($value['fk_product_type']==0) {
        // if product
        $nbcolspan+=1;
        $product->fetch($value['id']);
        $product->load_stock();
        if ((! empty($conf->productbatch->enabled)) && $product->hasbatch())
        {
            $details= $product->stock_warehouse[1]->detail_batch;

            print '<td align=center>';

            if ($details<0) dol_print_error($db);
            foreach ($details as $pdluo)
            {
                //print 'Caducidad '. dol_print_date($pdluo->eatby,'day') .',';
                //print 'Venta máxima '. dol_print_date($pdluo->sellby,'day') .',';
                print 'Lote '. $pdluo->batch. ',';
                print ' Stock '. $pdluo->qty;
                print '<br>';
            }
            print '</td>';
        }
        else {
            print '<td align=center>' . $factory->getUrlStock($value['id'], 1,
$product->stock_reel) . '</td>';
        }
        $nbcmd=0;
        // on regarde si il n'y pas de commande fournisseur en cours
        $sql = 'SELECT DISTINCT sum(cofd.qty) as nbCmdFourn';
        $sql.= " FROM ".MAIN_DB_PREFIX."commande_fournisseurdet as cofd";
        $sql.= " LEFT JOIN ".MAIN_DB_PREFIX."commande_fournisseur as cof ON
cof.rowid = cofd.fk_commande";
        $sql.= " WHERE cof.entity = ".$conf->entity;
        $sql.= " AND cof.fk_statut = 3";
        $sql.= " and cofd.fk_product=".$value['id'];
        //print $sql;
        $resql = $db->query($sql);
        if ($resql) {
            $objpcmd = $db->fetch_object($resql);
            if ($objpcmd->nbCmdFourn)

```

```
        $nbcmdde=$objpcmd->nbCmdFourn;
    }
    print '<td align=right>'. $nbcmdde. '</td>';
}
else // no stock management for services
    print '<td></td>';
}
if ($user->rights->factory->showprice) {
    $nbcspan+=6;
    // display else vwap or else latest purchasing price
    print '<td align="right">'.price($value['pmp'], 0, '', 1, 2, 2). '</td>';
    $qtyvalue=($value['globalqty']==1? 1 : $factory->qty_planned);
    print '<td align="right">'.price($value['pmp']*$value['nb']*$qtyvalue, 0, '',
1, 2, 2). '</td>';
    print '<td align="right">'.price($value['price'], 0, '', 1, 2, 2). '</td>';
    print '<td align="right">'.price($value['price']*$value['nb']*$qtyvalue, 0, '
1, 2, 2). '</td>';
    print '<td align="right">'.price(($value['price']-$value['pmp'])*$value['nb']
0, '', 1, 2, 2). '</td>';
    print '<td align="right">'.price(($value['price']-
$value['pmp'])*$value['nb']*$qtyvalue, 0, '', 1, 2, 2). '</td>';
    $mntTot = $mntTot + $value['price'] * $value['nb'] * $qtyvalue;
    $pmpTot = $pmpTot + $value['pmp'] * $value['nb'] * $qtyvalue; // sub total
calculation
}
print '</tr>';
if ($value['description']) {
    print "<tr style='display:none' class='detailligneComp'.".$value['id']. ">";
    print '<td></td><td colspan='. $nbcspan. '>'. $value['description']. '</td>';
    print '</tr>';
}
}
// only if not canceled
if ($user->rights->factory->showprice && $factory->qty_planned > 0) {
    print '<tr class="liste_total">';
    print '<td colspan=5 align=right >'. $langs->trans("Total"). '</b></td>';
    print '<td align="right" ><b>'.price($pmpTot/($factory->qty_planned), 0, '', 1,
2, 2). '</b></td>';
    print '<td align="right" ><b>'.price($pmpTot, 0, '', 1, 2, 2). '</b></td>';
    print '<td align="right" ><b>'.price($mntTot/($factory->qty_planned), 0, '', 1,
2, 2). '</b></td>';
    print '<td align="right" ><b>'.price($mntTot, 0, '', 1, 2, 2). '</b></td>';
    print '<td align="right" ><b>'.price(($mntTot-$pmpTot)/($factory->qty_planned),
0, '', 1, 2, 2). '</b></td>';
    print '<td align="right" ><b>'.price(($mntTot-$pmpTot), 0, '', 1, 2,
2). '</b></td>';
    print '</tr>';
}
print '</table>';
}
if ($btopChild) print '<b>'. $langs->trans("FactoryTableInfo"). '</b><BR>';

/* Gestion de la composition □ chaud */
if ($action == 'search') {
    $addselected=GETPOST("addselected");
    $keysearch=GETPOST('keysearch');

    // filtre s□lectionn□ on filtre
    $sql = 'SELECT DISTINCT p.rowid, p.ref, p.label, p.price, p.fk_product_type as
type, p.pmp';
    if ($conf->global->factory_extrafieldsNameInfo)
        $sql.= ' , pe.'. $conf->global->factory_extrafieldsNameInfo. ' as addinfoecup';
```



```
else
    $sql.= ' , "" as addinfoforecup';
    $sql.= ' FROM '.MAIN_DB_PREFIX.'product as p';
    $sql.= ' LEFT JOIN '.MAIN_DB_PREFIX.'product_extrafields as pe ON p.rowid =
pe.fk_object';
    $sql.= ' LEFT JOIN '.MAIN_DB_PREFIX.'categorie_product as cp ON p.rowid =
cp.fk_product';
    $sql.= ' WHERE p.entity IN ('.getEntity("product", 1).')';
    $sql.= ' AND p.rowid <> '.$productid; // pour ne pas afficher le produit lui-
me
    if ($keysearch != '') {
        $sql.= ' AND (p.ref LIKE \'%'.$keysearch.'%\'';
        $sql.= ' OR p.label LIKE \'%'.$keysearch.'%\'';
    }
    if ($conf->categorie->enabled && $parent != -1 and $parent) {
        $sql.= " AND cp.fk_categorie = " . $db->escape($parent) . " ";
    }

    if ($addselected) {
        $sql.= ' UNION SELECT DISTINCT p.rowid, p.ref, p.label, p.price,
p.fk_product_type as type, p.pmp';
        if ($conf->global->factory_extrafieldsNameInfo)
            $sql.= ' , pe.'. $conf->global->factory_extrafieldsNameInfo .' as addinfoforecup'
        else
            $sql.= ' , "" as addinfoforecup';
        $sql.= ' FROM '.MAIN_DB_PREFIX.'product as p';
        $sql.= ' LEFT JOIN '.MAIN_DB_PREFIX.'product_extrafields as pe ON p.rowid =
pe.fk_object';
        $sql.= ' , '.MAIN_DB_PREFIX.'product_factory as pf WHERE pf.fk_product_children
= p.rowid';
        $sql.= ' AND p.entity IN ('.getEntity("product", 1).')';
        $sql.= " AND pf.fk_product_father = " . $productid; // pour afficher les
produits d'j s'lectionn's
    }

    $resql = $db->query($sql);

    $productstatic = new Product($db);
}

$rowspan=1;
if ($conf->categorie->enabled)
    $rowspan++;

if ($action == 'edit' || $action == 'search' || $action == 're-edit' ) {
    print '<br>';
    print_fiche_titre($langs->trans("ProductToAddSearch"), '', '');
    print '<form action="fiche.php?id='.$id.'" method="post">';
    print '<table class="border" width="50%"><tr><td>';
    print '<table class="nobordernopadding" width="100%">';

    print '<tr><td>';
    print '<input type="hidden" name="token" value="'.$_SESSION['newtoken'].'">';
    print $langs->trans("KeywordFilter").' &nbsp; ';
    print '</td>';
    print '<td><input type="text" name="keysearch" value="'. $keysearch .'>';
    print '<input type="hidden" name="action" value="search">';
    print '<input type="hidden" name="id" value="'. $id .'>';
    print '</td>';
    print '<td rowspan="'. $rowspan .' " valign="bottom">';
    print '<input type="checkbox" name=addselected '. ($addselected?"checked":"").'
value="1">';
    print $langs->trans("AddSelectectProduct").'<br>';
    print '<input type="submit" class="button" value="'. $langs->trans("Search").'>';
}
```

```
print '</td></tr>';
if ($conf->categorie->enabled) {
    print '<tr><td>'. $langs->trans("CategoryFilter").' &nbsp; </td>';
    print '<td>'. $form->select_all_categories(0, $parent). '</td></tr>';
}

print '</table>';
print '</td></tr></table>';
print '</form>';

if ($action == 'search') {
    print '<br>';
    print '<form action="fiche.php?id='.$id.'" method="post">';
    print '<input type="hidden" name="token" value="'.$_SESSION['newtoken'].'">';
    print '<input type="hidden" name="action" value="add_prod">';
    print '<input type="hidden" name="id" value="'.$id.'">';
    print '<table class="noborder nopadding" width="100%">';
    print '<tr class="liste_titre">';
    print '<th class="liste_titre">'. $langs->trans("Ref"). '</th>';
    print '<th class="liste_titre">'. $langs->trans("Label"). '</th>';
    print '<th class="liste_titre" align="right">'. $langs->trans("BuyPrice"). '</th>';
    print '<th class="liste_titre" align="right">'. $langs->
>trans("SellPrice"). '</th>';
    print '<th class="liste_titre" align="center">'. $langs->trans("AddDel"). '</th>';
    print '<th class="liste_titre" align="right">'. $langs->trans("Quantity"). '</th>';
    print '<th class="liste_titre" align="right">'. $langs->trans("Global"). '</th>';
    print '</tr>';
    if ($resql) {
        $num = $db->num_rows($resql);
        $i=0;
        $var=true;

        if ($num == 0)
            print '<tr><td colspan="4">'. $langs->trans("NoMatchFound"). '</td></tr>';

        while ($i < $num) {
            $objp = $db->fetch_object($resql);
            $var=!$var;
            print "\n<tr ". $bc[$var]. ">";
            $productstatic->id=$objp->rowid;
            $productstatic->ref=$objp->ref;
            $productstatic->libelle=$objp->label;
            $productstatic->type=$objp->type;

            print '<td>'. $factory->getNomUrlFactory($objp->rowid, 1, 'index', 24);
            print $factory->PopupProduct($objp->rowid, $i);
            print '</td>';
            $labeltoshow=$objp->label;
            //if ($conf->global->MAIN_MULTILANGS && $objp->labelm) $labeltoshow=$objp-
>labelm;

            print '<td>';
            print "<a href=# onclick=\"$( '.detailligne'.".$i." ).toggle();\"
>".img_picto("", "edit_add"). "</a>&nbsp;";
            print $labeltoshow. '</td>';
            if ($factory->is_sousproduitOF($id, $objp->rowid)) {
                $addchecked = ' checked="checked"';
                $qty = $factory->is_sousproduit_qty;
                $descComposant = $factory->is_sousproduit_description;
            } else {
                $addchecked = '';
                $qty = "1";
                $descComposant = '';
            }
        }
    }
}
```

```
        print '<td align="right">'.price($objp->pmp).'
```

```
        print '<tr><td >'. $langs->trans("BOMList").'</td><td align=left>';

        print $form->selectarray("bomid", $bomArray, "", 1);
        print '</td></tr>';
    }
    print '<tr><td >'. $langs->trans("Products").'</td><td align=left>';
}
else
    print '<tr><td class="fieldrequired">'. $langs->trans("Products").'</td><td
align=left>';
// seulement les produits fabricables
$factoryproductarray = $factory->getListProductWithComposition();
print $form->selectarray("productid", $factoryproductarray, "", 1);

print '</td></tr>';

print '<tr><td width=250px>'. $langs->trans("EntrepotStock").'</td><td >';
print select_entrepot_list("", "entrepotid", 0, 1);
print '</td></tr>';
print '<tr><td>'. $langs->trans("QtyToBuild").'</td>';
print '<td ><input style="text-align:right;" type="text" name="nbToBuild" size=5
value="'.GETPOST("nbToBuild").'">';
print '</td></tr>';

print '<tr><td>'. $langs->trans("FactoryDateStartPlanned").'</td>';
print '<td >';
$plannedstart=dol_mktime(
    GETPOST('plannedstarthour', 'int'), GETPOST('plannedstartmin', 'int'), 0,
    GETPOST('plannedstartmonth', 'int'), GETPOST('plannedstartday', 'int'),
    GETPOST('plannedstartyear', 'int')
);
print $form->select_date(
    (GETPOST("plannedstart")? $plannedstart:''), 'plannedstart',
    1, 1, '', "plannedstart"
);
print '</td></tr>';

print '<tr><td>'. $langs->trans("FactoryDateEndBuildPlanned").'</td>';
print '<td >';
$plannedend=dol_mktime(
    GETPOST('plannedendhour', 'int'), GETPOST('plannedendmin', 'int'), 0,
    GETPOST('plannedendmonth', 'int'), GETPOST('plannedendday', 'int'),
    GETPOST('plannedendyear', 'int')
);
print $form->select_date(
    (GETPOST("plannedend")? $plannedend:''), 'plannedend',
    1, 1, '', "plannedend"
);
print '</td></tr>';

print '<tr><td>'. $langs->trans("FactoryDurationPlanned").'</td>';
print '<td >';
print $form->select_duration(
    'workload', (int) GETPOST("workloadhour")*3600+ (int) GETPOST("workloadmin"
"int")*60, 0, 'text'
);
print '</td></tr>';

print '</table>';
print '</td></tr>';

print '<tr><td colspan=2 align=center>';
print '<br><input type="submit" class="button" name="verifyof" value="'. $langs->
>trans("VerifyQty").'">';
```

```
print '</td></tr>';
print '</table>';
print '</form>';
}

/* Barre d'action */
if ($action == '' ) {
    print '<div class="tabsAction">';

    $parameters = array();
    // Note that $action and $object may have been
    $reshook = $hookmanager->executeHooks('addMoreActionsButtons', $parameters, $factory, $action);
    // modified by hook
    if (empty($reshook)) {

        if ($user->rights->factory->creer && $factory->statut == 0) {
            print '<a class="butAction" href="fiche.php?action=edit&id='.$id.'">'. $langs->trans("ChangeGlobalQtyFactory").'</a>';
            print '<a class="butAction" href="fiche.php?action=getdefaultprice&id='.$id.'">'. $langs->trans("GetDefaultPrice").'</a>';
            print '<a class="butAction" href="fiche.php?action=adjustprice&id='.$id.'">'. $langs->trans("AdjustPrice").'</a>';
            print '<br>';
        }

        print '<div class="inline-block divButAction">';
        if ($user->rights->factory->send)
            print '<a class="butAction" href="'.$_SERVER["PHP_SELF"].'?id='.$id.'&action=present&mode=init">';
        else
            print '<a class="butActionRefused" href="#">';
        print $langs->trans('SendByMail').'</a>';
        print '</div>';

        if ($user->rights->factory->creer)
            print '<a class="butAction" href="fiche.php?action=clone&id='.$id.'">'. $langs->trans("CloneOF").'</a>';

        if ($user->rights->factory->annuler && $factory->statut == 0)
            print '<a class="butAction" href="fiche.php?action=cancelof&id='.$id.'">'. $langs->trans("CancelFactory").'</a>';

        print '</div>';

        print '<div class="fichecenter"><div class="fichehalfleft">';
        print '<br><br>';
        /*
        * Documents generes
        */
        $comref = dol_sanitizeFileName($factory->ref);
        // $file = $conf->factory->dir_output.'/'.$comref.'/'.$comref.'.pdf';
        // $relativepath = $comref.'/'.$comref.'.pdf';
        $filedir = $conf->factory->dir_output.'/'.$comref;
        $urlsource=$_SERVER["PHP_SELF"]."?id=".$factory->id;
        $genallowed=$user->rights->factory->creer;
        $delallowed=$user->rights->factory->delete;
        $somethingshown=$formfile->show_documents(
            'factory', $comref, $filedir, $urlsource,
            $genallowed, $delallowed, $factory->modelpdf,
            1, 0, 0, 28, 0, '', '', '', ''
        );
    }
};
```

```
/*
 * Linked object block
 */
if ((int) DOL_VERSION >= 5)
    $somethingshown = $form->showLinkedObjectBlock($factory, "");
else
    $somethingshown = $factory->showLinkedObjectBlock();

print '</div><div class="fichehalfright"><div class="ficheaddleft">';

// List of actions on element
include_once DOL_DOCUMENT_ROOT.'/core/class/html.formactions.class.php';
$formactions=new FormActions($db);
$somethingshown=$formactions->showactions($factory, 'factory', $socid);

    print '</div></div>';
}
print '</div>';
} elseif ($action == 'adjustprice') {
    print '<br>';
    print_fiche_titre($langs->trans("AdjustPrice"), '', '');

    print '<form action="fiche.php?id='.$id.'" method="post">';
    print '<input type="hidden" name="token" value="'.$_SESSION['newtoken'].'">';
    print '<input type="hidden" name="action" value="updateprice">';
    print '<input type="hidden" name="id" value="'.$id.'">';
    print '<table class="nobordernopadding" >';
    print '<tr class="liste_titre">';
    print '<th width="100px" class="liste_titre">'.$langs->trans("Ref").'</th>';
    print '<th width="200px" class="liste_titre">'.$langs->trans("Label").'</th>';
    print '<th width="50px" class="liste_titre" align="right">'.$langs->
>trans("Quantity").'</th>';
    print '<th width="100px" class="liste_titre" align="right">'.$langs->
>trans("InfoPMP").'</th>';
    print '<th width="100px" class="liste_titre" align="right">'.$langs->
>trans("InfoCostPrice").'</th>';
    print '<th width="100px" class="liste_titre" align="right">'.$langs->
>trans("BuyPrice").'</th>';
    print '<th width="100px" class="liste_titre" align="right">'.$langs->
>trans("InfoSellPrice").'</th>';
    print '<th width="100px" class="liste_titre" align="right">'.$langs->
>trans("SellPrice").'</th>';

    print '</tr>';

    $productstatic = new Product($db);

    foreach ($prods_arbo as $value) {
        $productstatic->id=$value['id'];
        $productstatic->fetch($value['id']);
        $productstatic->type=$value['type'];
        $var=!$var;

        print "\n<tr ". $bc[$var].">";

        print '<td>'.$factory->getNomUrlFactory($value['id'], 1, 'fiche', 24).'</td>';
        $labeltoshow=$productstatic->label;

        print '<td>'.$labeltoshow.'</td>';

        //var_dump($value);
        $qty=$value['qtyplanned'];
```

```
print '<td align="right">'. $qty. '</td>';

print '<td align="right">'. price($productstatic->pmp). '</td>';
print '<td align="right">'. price($productstatic->cost_price). '</td>';
print '<td align="right">';
print '<input type="text" size="5" name="prod_pmp_'. $value['id']. '"
value="'. price2num($value['pmp']). '"></td>';

print '<td align="right">'. price($productstatic->price). '</td>';
print '<td align="right">';
print '<input type="text" size="5" name="prod_price_'. $value['id']. '"
value="'. price2num($value['price']). '"></td>';

print '</tr>';
}

print '</table>';
print '<input type="hidden" name="max_prod" value="'. $i. '">';

print '<br><center>';
print ' <input type="submit" class="button" value="'. $langs->trans("Update"). '">';
print ' &nbsp; &nbsp; &nbsp;';
print ' <input type="submit" class="button" name="cancel" value="'. $langs-
>trans("Cancel"). '">';
print '</center>';
print '</form>';
}

/*
 * Add file in email form
 */
if (GETPOST('addfile')) {

    // Set tmp user directory TODO Use a dedicated directory for temp mails files
    $vardir=$conf->user->dir_output."/". $user->id;
    $upload_dir_tmp = $vardir.'/temp';

    dol_add_file_process($upload_dir_tmp, 0, 0);
    $action = 'presend';
}

/*
 * Remove file in email form
 */
if (GETPOST('removedfile')) {

    // Set tmp user directory
    $vardir=$conf->user->dir_output."/". $user->id;
    $upload_dir_tmp = $vardir.'/temp';

    // TODO Delete only files that was uploaded from email form
    dol_remove_file_process(GETPOST('removedfile'), 0);
    $action = 'presend';
}

//
/*
 * Send mail
 */

if ($action == 'send' && ! GETPOST('addfile') && ! GETPOST('removedfile') && !
GETPOST('cancel')) {
    $langs->load('mails');
```

```
if ($id > 0) {
    $ref = dol_sanitiseFileName($factory->ref);
    $file = $conf->factory->dir_output.'/'.$ref.'/'.$ref.'.pdf';
    if (is_readable($file)) {
        if (GETPOST('sendto')) {
            // Le destinataire a ete fourni via le champ libre
            $sendto = GETPOST('sendto');
            $sendtoid = 0;
        } elseif (GETPOST('receiver') != '-1') {
            // Recipient was provided from combo list
            if (GETPOST('receiver') == 'thirdparty') {
                // Id of third party
                $sendto = $factory->client->email;
                $sendtoid = 0;
            } else {
                // Id du contact
                // $sendto = $factory->client->contact_get_property(GETPOST('receiver'),
                'email');
                $sendtoid = GETPOST('receiver');
            }
        }
    }

    if (dol_strlen($sendto)) {
        $langs->load("commercial");

        $from = GETPOST('fromname') . ' <' . GETPOST('frommail') . '>';
        $replyto = GETPOST('replytoname') . ' <' . GETPOST('replytomail') . '>';
        $message = GETPOST('message');
        $sendtoocc = GETPOST('sendtoocc');
        $deliveryreceipt = GETPOST('deliveryreceipt');

        if ($action == 'send') {
            if (dol_strlen(GETPOST('subject')))
                $subject=GETPOST('subject');
            else
                $subject = $langs->transnoentities('Order').' '.$factory->ref;
            $actiontypecode='AC_COM';
            $actionmsg = $langs->transnoentities('MailSentBy').' '.$from.' '.$langs->
            >transnoentities('To').' '.$sendto.".\n";
            if ($message) {
                $actionmsg.=$langs->transnoentities('MailTopic').": ".$subject.".\n";
                $actionmsg.=$langs->transnoentities('TextUsedInTheMessageBody').":.\n";
                $actionmsg.=$message;
            }
            $actionmsg2=$langs->transnoentities('Action'.$actiontypecode);
        }

        // Create form object
        include_once DOL_DOCUMENT_ROOT.'/core/class/html.formmail.class.php';
        $formmail = new FormMail($db);

        $attachedfiles=$formmail->get_attached_files();
        $filepath = $attachedfiles['paths'];
        $filename = $attachedfiles['names'];
        $mimetype = $attachedfiles['mimes'];

        // Send mail
        require_once DOL_DOCUMENT_ROOT.'/core/class/CMailFile.class.php';
        $mailfile = new CMailFile(
            $subject, $sendto, $from, $message, $filepath,
            $mimetype, $filename, $sendtoocc, '', $deliveryreceipt, -1
        );
    }
}
```



```
if ($mailfile->error)
    $mesg='<div class="error">'.$mailfile->error.'</div>';
else {
    $result = $mailfile->sendfile();
    if ($result) {
        $mesg = $langs->trans(
            'MailSuccessfullySent',
            $mailfile->getValidAddress($from, 2),
            $mailfile->getValidAddress($sendto, 2)
        ); // Must not contains "
        $error=0;

        // Initialisation donnees
        $factory->sendtoid = $sendtoid;
        $factory->actiontypecode = $actiontypecode;
        $factory->actionmsg = $actionmsg;
        $factory->actionmsg2 = $actionmsg2;
        $factory->fk_element = $factory->id;
        $factory->elementtype = $factory->element;

        // Appel des triggers
        include_once DOL_DOCUMENT_ROOT . '/core/class/interfaces.class.php';
        $interface=new Interfaces($db);
        $result=$interface->run_triggers('FACTORY_SENTBYMAIL', $factory, $user,
$langs, $conf);
        if ($result < 0) {
            $error++; $this->errors=$interface->errors;
        }
        // Fin appel triggers

        if ($error) {
            dol_print_error($db);
        } else {
            // Redirect here
            // This avoid sending mail twice if going out and then back to page
            //header('Location:
' . $_SERVER["PHP_SELF"] . '?id=' . $id . '&mesg=' . urlencode($mesg));
            //exit;
        }
    } else {
        $langs->load("other");
        $mesg='<div class="error">';
        if ($mailfile->error) {
            $mesg.=$langs->trans('ErrorFailedToSendMail', $from, $sendto);
            $mesg.='<br>'.$mailfile->error;
        } else {
            $mesg.='No mail sent. Feature is disabled by option
MAIN_DISABLE_ALL_MAILS';
        }
        $mesg.='</div>';
    }
}
} else {
    $langs->load("other");
    $mesg='<div class="error">'.$langs->trans('ErrorMailRecipientIsEmpty').'
!</div>';
    $action='presend';
    dol_syslog('Recipient email is empty');
}
} else {
    $langs->load("errors");
    $mesg='<div class="error">'.$langs->trans('ErrorCantReadFile', $file).'</div>';
    dol_syslog('Failed to read file: '.$file);
}
}
```

```
    } else {
        $langs->load("other");
        $mesg='<div class="error">'. $langs->trans('ErrorFailedToReadEntity', $langs->trans("Order")). '</div>';
        dol_syslog($langs->trans('ErrorFailedToReadEntity', $langs->trans("Order")));
    }

    $action="";
}

dol_htmloutput_mesg($mesg);

/*
 * Action present
 */
if ($action == 'present') {
    $ref = dol_sanitizeFileName($factory->ref);

    $fileparams = dol_most_recent_file($conf->factory->dir_output . '/' . $ref, preg_quote($ref, '/'));
    $file=$fileparams['fullname'];

    // Build document if it not exists
    if (! $file || ! is_readable($file)) {
        // Define output language
        $outputlangs = $langs;
        $newlang='';
        if ($conf->global->MAIN_MULTILANGS && empty($newlang) && ! empty($_REQUEST['lang_id']))
            $newlang=$_REQUEST['lang_id'];
        //if ($conf->global->MAIN_MULTILANGS && empty($newlang)) $newlang=$factory->client->default_lang;
        if (! empty($newlang)) {
            $outputlangs = new Translate("", $conf);
            $outputlangs->setDefaultLang($newlang);
        }

        $result=factory_create($db, $factory, $factory->modelpdf, $outputlangs, $shidedetails, $shidedesc, $hideref);

        if ($result <= 0) {
            dol_print_error($db, $result);
            exit;
        }
        $fileparams = dol_most_recent_file($conf->factory->dir_output.'/' . $ref, preg_quote($ref, '/'));
        $file=$fileparams['fullname'];
    }
    var_dump($file);

    print '<br>';
    print_titre($langs->trans('SendFactoryByMail'));

    // Cree l'objet formulaire mail
    include_once DOL_DOCUMENT_ROOT.'/core/class/html.formmail.class.php';
    $formmail = new FormMail($db);
    $formmail->fromtype = 'user';
    $formmail->fromid = $user->id;
    $formmail->fromname = $user->getFullName($langs);
    $formmail->frommail = $user->email;
    $formmail->withfrom=1;
}
```

```
// on recupere les contacts de l'entrepot
$liste=array();
foreach ($factory->contact_entrepot_email_array() as $key=>$value)
    $liste[$key]=$value;

$formmail->withto=GETPOST('sendto')?GETPOST('sendto'):$liste;
$formmail->withtocc=$liste;
$formmail->withtoccc=$conf->global->MAIN_EMAIL_USECCC;
$formmail->withtopic=$langs->trans('SendFactoryRef', '__FACTORYREF__');
$formmail->withfile=2;
//$formmail->withmaindocfile=1;
$formmail->withbody=1;
$formmail->withdeliveryreceipt=1;
$formmail->withcancel=1;
// Tableau des substitutions
$formmail->substit['__FACTORYREF__']=$factory->ref;
$formmail->substit['__SIGNATURE__']=$user->signature;
//$formmail->substit['__REFCLIENT__']=$factory->ref_client;
$formmail->substit['__PERSONALIZED__']='';
//$formmail->substit['__CONTACTCIVNAME__']='';

$custcontact='';
$contactarr=array();
$entrepotStatic=new Entrepot($db);
$entrepotStatic->fetch($factory->fk_entrepot);
$entrepotStatic->element='stock'; // bug dolibarr corrigé dans les prochaines
versions
$contactarr=$entrepotStatic->liste_contact(-1, 'external');
if (is_array($contactarr) && count($contactarr)>0) {
    foreach ($contactarr as $contact) {
        if ($contact['libelle'] == $langs->trans('TypeContact_entrepot_external')) {
            $contactstatic=new Contact($db);
            $contactstatic->fetch($contact['id']);
            $custcontact=$contactstatic->getFullName($langs, 1);
        }
    }

    if (!empty($custcontact)) {
        $formmail->substit['__CONTACTCIVNAME__']=$custcontact;
    }
}

// Tableau des parametres complementaires
$formmail->param['action']='send';
$formmail->param['models']='factory_send';
$formmail->param['factoryid']=$id;
$formmail->param['returnurl']=$_SERVER["PHP_SELF"].'?id='.$id;

// Init list of files
if (GETPOST("mode")== 'init') {
    $formmail->clear_attached_files();
    $formmail->add_attached_files($file, basename($file), dol_mimetype($file));
    $formmail->param['fileinit'] =array($file);
}
// Show form
$formmail->show_form();

print '<br>';
}

llxFooter();
$db->close();

print '<script>$(function){';
```

```
print '$(".tiptipimg").tipTip({maxWidth: "auto", edgeOffset: 10});';  
print '});</script>';
```