

```
<?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/product/index.php
 * \ingroup product
 * \brief Page de définition de la fabrication
 */

$res=0;
if (! $res && file_exists(".././main.inc.php"))
    $res=@include(".././main.inc.php"); // For root directory
if (! $res && file_exists(".././../main.inc.php"))
    $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";

if ((int) DOL_VERSION > 10 && $conf->global->MAIN_MODULE_BOM ) {
    require_once DOL_DOCUMENT_ROOT."/bom/class/bom.class.php";
}

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

$langs->load("bills");
$langs->load("products");

$id=GETPOST('id', 'int');
$ref=GETPOST('ref', 'alpha');
$action=GETPOST('action', 'alpha');
$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, 'produit|service', $fieldvalue, 'product&product', '',
'', $fieldtype, $objcanvas);
```

```
$mesg = '';  
  
$object = new Product($db);  
$factory = new Factory($db);  
$productid=0;  
if ($id || $ref) {  
    $result = $object->fetch($id, $ref);  
    $productid=$object->id;  
    $id=$object->id;  
    $factory->id = $id;  
}  
  
/*  
 * Actions  
 */  
  
// add sub-product to a product  
if ( $action == 'add_prod' && $cancel <> $langs->trans("Cancel")  
&& ($user->rights->produit->creer || $user->rights->service->creer)) {  
    $error=0;  
    for ($i=0;$i<$_POST["max_prod"];$i++) {  
        //print "<br> : ".$_POST["prod_id_chk".$i];  
        if ($_POST["prod_id_chk".$i] != "") {  
            if ($factory->add_component(  
                $id, $_POST["prod_id_".$i], $_POST["prod_qty_".$i],  
                0, 0, $_POST["prod_id_globalchk".$i],  
                $_POST["descComposant".$i], $_POST["prod_order_".$i]  
            ) > 0)  
                $action = 'edit';  
            else {  
                $error++;  
                $action = 're-edit';  
                if ($factory->error == "isFatherOfThis")  
                    $mesg.=( $mesg ? "<br>" : "" ).'<div class="error">'.$langs->  
>trans("ErrorAssociationIsFatherOfThis").'</div>';  
                else  
                    $mesg.=( $mesg ? "<br>" : "" ).$factory->error;  
            }  
        } else {  
            if ($factory->del_component($id, $_POST["prod_id_".$i]) > 0)  
                $action = 'edit';  
            else {  
                $error++;  
                $action = 're-edit';  
                $mesg.=( $mesg ? "<br>" : "" ).$factory->error;  
            }  
        }  
    }  
    if (! $error) {  
        header("Location: ".$_SERVER["PHP_SELF"].'?id='.$object->id);  
        exit;  
    }  
}  
if ($cancel == $langs->trans("Cancel")) {  
    $action = '';  
    Header("Location: index.php?id=".$_POST["id"]);  
    exit;  
}  
  
if ($action == 'getfromvirtual') {  
    // on r cup re la fabrication de la composition virtuelle
```

```
$factory->clonefromvirtual();
$action="";
}

if ($action == 'getdefaultprice') {
    $factory->getdefaultprice();
    $action="";
}

if ($action == 'updateprice') {
    // on modifie les prix
    $prodsfather = $factory->getFather(); //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;

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

if ($action == 'importation') {
    $factory->importComposition(GETPOST("importexport"));
    $action="";
}

if ($action == 'remplacementproduct') {
    $factory->productChange(GETPOST("fk_product","int"),GETPOST("productchange",
"array"));
    $action="";
}

/*
 * View
 */

// search products by keyword and/or categorie
if ($action == 'search') {
    $addselected=GETPOST("addselected");
    $keysearch=GETPOST('keysearch');

    // filtre slectionn 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_extracfieldsNameInfo)
        $sql.= ' , pe.'. $conf->global->factory_extracfieldsNameInfo. ' as addinfoforecup';
    else
        $sql.= ' , \'\' as addinfoforecup';
    $sql.= ' FROM '.MAIN_DB_PREFIX.'product as p';
    $sql.= ' LEFT JOIN '.MAIN_DB_PREFIX.'product_extracfields 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-m
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 addinfoencup';
    else
        $sql.= ' , \'\' as addinfoencup';
    $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).')';
    // pour afficher les produits d'j slectionn
    $sql.= " AND pf.fk_product_father = ".$productid;
}
// $sql.= " ORDER BY p.ref ASC";

//print $sql;

$resqlsearch = $db->query($sql);
}
//print $sql;

$productstatic = new Product($db);
$form = new Form($db);

llxHeader("", "", $langs->trans("CardProduct".$product->type));

dol_htmloutput_mesg($mesg);

$head=product_prepare_head($object, $user);
$titre=$langs->trans("CardProduct".$object->type);
$picto=('product');
dol_fiche_head($head, 'factory', $titre, -1, $picto);
$linkback = '<a href="'.DOL_URL_ROOT.'/product/list.php'.(!
empty($socio)?'?socio='.$socio:').">';
$linkback.= $langs->trans("BackToList").'</a>';

if ($id || $ref) {
    if ($result) {
        $bproduit = ($object->isproduct());

        if ((int) DOL_VERSION >= 5) {
            dol_banner_tab($object, 'ref', $linkback, ($user->societe_id?0:1), 'ref');
            $cssclass='titlefield';
            print '<div class="underbanner clearboth"></div>';
        } else {
            print '<table class="border" width="100%">';
            print "<tr>";

            // Reference
            print '<td width="25%">'.$langs->trans("Ref").'</td><td>';
            print $form->showrefnav($object, 'ref', $linkback, 1, 'ref');
            print '</td></tr>';
        }
    }
}
```

```
// Libelle
print '<tr><td>'. $langs->trans("Label"). '</td>';
print '<td colspan="3">'. ($object->label ? $object->label : $object->libelle). '</td></tr>';

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

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

print '<table class="border" width="100%">';
// MultiPrix
if ($conf->global->PRODUIT_MULTIPRICES) {
    if ($socid) {
        $soc = new Societe($db);
        $soc->id = $socid;
        $soc->fetch($socid);

        print '<tr><td width="25%">'. $langs->trans("SellingPrice"). '</td>';

        if ($object->multiprices_base_type["$soc->price_level"] == 'TTC')
            print '<td>'. price($object->multiprices_ttc["$soc->price_level"]);
        else
            print '<td>'. price($object->multiprices["$soc->price_level"]);

        if ($object->multiprices_base_type["$soc->price_level"])
            print ' '. $langs->trans($object->multiprices_base_type["$soc->price_level"]);
        else
            print ' '. $langs->trans($object->price_base_type);
        print '</td></tr>';

// Prix mini
print '<tr><td>'. $langs->trans("MinPrice"). '</td><td>';
if ($object->multiprices_base_type["$soc->price_level"] == 'TTC') {
    print price($object->multiprices_min_ttc["$soc->price_level"]);
    print $langs->trans($object->multiprices_base_type["$soc->price_level"]);
} else {
    print price($object->multiprices_min["$soc->price_level"]);
    print $langs->trans($object->multiprices_base_type["$soc->price_level"]);
}
print '</td></tr>';

// TVA
print '<tr><td>'. $langs->trans("VATRate"). '</td>';
print '<td>'. vatrate($object->multiprices_tva_tx["$soc->price_level"],
true). '</td></tr>';
} else {
    for ($i=1; $i<=$conf->global->PRODUIT_MULTIPRICES_LIMIT; $i++) {
        // TVA
        if ($i == 1) {
            // We show only price for level 1
            print '<tr><td>'. $langs->trans("VATRate"). '</td>';
            print '<td>'. vatrate($object->multiprices_tva_tx[1], true). '</td></tr>';
        }
        print '<tr><td width="25%">'. $langs->trans("SellingPrice"). ' '. $i. '</td>';
    }
}
```

```
if ($object->multiprices_base_type["$i"] == 'TTC')
    print '<td>'.price($object->multiprices_ttc["$i"]);
else
    print '<td>'.price($object->multiprices["$i"]);

if ($object->multiprices_base_type["$i"])
    print ' '.$langs->trans($object->multiprices_base_type["$i"]);
else
    print ' '.$langs->trans($object->price_base_type);
print '</td></tr>';

// Prix mini
print '<tr><td>'. $langs->trans("MinPrice").' '.$i.'</td><td>';
if ($object->multiprices_base_type["$i"] == 'TTC')
    print price($object->multiprices_min_ttc["$i"]).' '.$langs->trans($object->
>multiprices_base_type["$i"]);
else
    print price($object->multiprices_min["$i"]).' '.$langs->trans($object->
>multiprices_base_type["$i"]);
    print '</td></tr>';
}
} else {
// TVA
print '<tr><td width="25%">'. $langs->trans("VATRate").'</td>';
print '<td>'.vatrate($object->tva_tx.($object->tva_npr?'*':''),
true).'</td></tr>';

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

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

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

// on indique la nature du produit
print '<tr><td>'. $langs->trans("Nature").'</td>';
print '<td>'. $object->getLibFinished().'</td></tr>';

print '<tr><td>'. $langs->trans("PhysicalStock").'</td>';
print '<td>'. $object->stock_reel.'</td></tr>';
print '</table>';
dol_fiche_end();

// indique si on a déjà une composition de présente ou pas
$compositionpresente=0;

$head=factory_product_prepare_head($object, $user);
```

```

$titre=$langs->trans("Factory");
$picto="factory@factory";
dol_fiche_head($head, 'composition', $titre, -1, $picto);

// pour connaitre les produits composé du produits
$prodsfather = $factory->getFather(); //Parent Products

// pour connaitre les produits composant le produits
$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;

/* ***** */
/*
/* Importation / d'une composition
/*
/* ***** */
if ($action == 'importexport') {
    /*
    * Import/export customtabs
    */
    print_fiche_titre($langs->trans("ImportExportComposition"), '', '');

    print '<form action="'. $_SERVER["PHP_SELF"].' method="POST">';
    print '<input type="hidden" name="token" value="'. $_SESSION['newtoken'].'">';
    print '<input type="hidden" name="action" value="importation">';
    print '<input type="hidden" name="id" value="'.GETPOST("id").'">';
    print '<input type="hidden" name="backtopage" value="'. $backtopage.'">';

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

    print '<tr><td><span class="fieldrequired">'. $langs->trans("FillImportExportData").'</span></td></tr>';
    print '<td><textarea name="importexport" cols=132 rows=20>';
    print $factory->getExportComposition($prods_arbo);
    print '</textarea></td></tr>';
    print '</table>';
    print '<br><center>';
    print '<input type="submit" class="button" value="'. $langs->trans("LaunchImport").'">';
    print '</center>';
    print '</form>';
} else {
    print_fiche_titre($langs->trans("FactorisedProductsNumber").' :
'.count($prods_arbo), '', '');

    // List of subproducts
    if (count($prods_arbo) > 0) {
        $compositionpresente=1;
        print '<b>'. $langs->trans("FactoryTableInfo").'</b>';
        print '<table class="border" >';
        print '<tr class="liste_titre">';
        print '<th class="liste_titre" width=10px></th>';
        print '<th class="liste_titre" width=100px align="left">'. $langs->trans("Ref").'</th>';
        print '<th class="liste_titre" width=200px align="left">'. $langs->trans("Label").'</th>';
        print '<th class="liste_titre" width=50px align="center">'. $langs->trans("QtyNeed").'</th>';
        // on affiche la colonne stock même si cette fonction n'est pas active
        print '<th class="liste_titre" width=50px align="center">'. $langs->trans("RealStock").'</th>';
    }
}

```

```

        print '<th class="liste_titre" width=100px align="center">'. $langs-
>trans("QtyOrder").'</th>';
        if ($conf->stock->enabled) { // we display vwap titles
            print '<th class="liste_titre" width=100px align="right">'. $langs-
>trans("UnitPmp").'</th>';
            print '<th class="liste_titre" width=100px align="right">'. $langs-
>trans("CostPmpHT").'</th>';
        } else {
            // we display price as latest purchasing unit price title
            print '<th class="liste_titre" width=100px align="right">'. $langs-
>trans("FactoryUnitHA").'</th>';
            print '<th class="liste_titre" width=100px align="right">'. $langs-
>trans("FactoryCostHA").'</th>';
        }
        print '<th class="liste_titre" width=100px align="right">'. $langs-
>trans("FactoryUnitPriceHT").'</th>';
        print '<th class="liste_titre" width=100px align="right">'. $langs-
>trans("FactorySellingPriceHT").'</th>';
        print '<th class="liste_titre" width=100px align="right">'. $langs-
>trans("ProfitAmount").'</th>';

        print '</tr>';
        $mntTot=0;
        $pmpTot=0;

        foreach ($prods_arbo as $value) {
            //var_dump($value);
            // verify if product have child then display it after the product name
            $tmpChildArbo=$factory->getChiltsArbo($value['id']);
            $nbChildArbo="";
            if (count($tmpChildArbo) > 0) $nbChildArbo=" (" .count($tmpChildArbo).)";

            print '<tr><td>';
            print "<a href='#line".$objjp->rowid."'
onclick=\$('<td align="left">'. $factory->getNomUrlFactory($value['id'], 1,
'index').$nbChildArbo;
            print $factory->PopupProduct($value['id']);
            print '</td>';

            print '<td align="left" title="'. $value['description'].'">';
            print $value['label'].'</td>';
            print '<td align="center">'. $value['nb'];
            if ($value['globalqty'] == 1)
                print "&nbsp;G";
            print '</td>';
            $price=$value['price'];
            $pmp=$value['pmp'];

            if ($conf->stock->enabled) {
                // we store vwap in variable pmp and display stock
                $productstatic->fetch($value['id']);

                if ($value['fk_product_type']==0) {
                    // if product
                    $bAllService=false;
                    $productstatic->load_stock();
                    print '<td align=center>'. $factory->getUrlStock($value['id'], 1,
$productstatic->stock_reel).'</td>';
                    $nbcmde=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) {
            $objp = $db->fetch_object($resql);
            if ($objp->nbCmdFourn)
                $nbcmdde=$objp->nbCmdFourn;
        }
        print '<td align=right>'. $nbcmdde.'</td>';
    } else // no stock management for services
        print '<td></td>';
}
else // no stock management for services
    print '<td></td>';

    print '<td align="right">'.price($pmp).'</td>'; // display else vwap or else
latest purchasing price
    print '<td align="right">'.price($pmp*$value['nb']).'</td>'; // display
total line
    print '<td align="right">'.price($price).'</td>';
    print '<td align="right">'.price($price*$value['nb']).'</td>';
    print '<td align="right">'.price(($price-$pmp)*$value['nb']).'</td>';

    $mntTot=$mntTot+$price*$value['nb'];
    $pmpTot=$pmpTot+$pmp*$value['nb']; // sub total calculation

    print '</tr>';
    print "<tr style='display:none' class='detaillignecomposition".$objp->rowid.">";
    print '<td colspan=2>'. $langs->trans("Position")." :
".$value['ordercomponent'].'</td>';
    print '<td colspan=9>'. $langs->trans("InfoAnnexes")." :
".$value['description'].'</td>';
    print '</tr>';
    //var_dump($value);
    //print '<pre>'. $productstatic->ref.'</pre>';
    //print $productstatic->getNomUrl(1).'<br>';
    //print $value[0]; // This contains a tr line.
}
print '<tr class="liste_total">';
print '<td colspan=7 align=right >'. $langs->trans("Total").'</td>';
print '<td align="right" >'.price($pmpTot).'</td>';
print '<td ></td>';
print '<td align="right" >'.price($mntTot).'</td>';
print '<td align="right" >'.price($mntTot-$pmpTot).'</td>';
print '</tr>';
print '</table>';
print '<br>';
}
}

// Number of parent products

$tblBomList = array();
if ((int) DOL_VERSION > 10 && $conf->global->MAIN_MODULE_BOM ) {

    $product_bom = new Product($db);
    $bom_static = new BOM($db);
    // on ne sélectionne que les bom actif
```

```
        $tblBomList = $bom_static->fetchAll('', '', 0, 0,
            array('fk_product'=>$id)
        );
var_dump($id);
var_dump($tblBomList);
var_dump($prodsfather);
    }

    print_fiche_titre($langs->trans("ParentComposedProductsNumber").' :
'.(count($prodsfather)+count($tblBomList)), '', '');

    if (count($tblBomList) > 0) {
        print "<br>";
        print '<b>'.$langs->trans("BOMListTableInfo").'</b><br>';
        print '<table class="border" >';
        print '<tr class="liste_titre">';
        print '<td class="liste_titre" width=100px align="left">'.$langs-
>trans("RefBom").'</td>';
        print '<td class="liste_titre" width=100px align="left">'.$langs-
>trans("RefProduct").'</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("Stock").'</td>';
        print '<td class="liste_titre" width=100px align="right">'.$langs-
>trans("UnitPmp").'</td>';
        print '<td class="liste_titre" width=100px align="right">'.$langs-
>trans("SellingPriceHT").'</td>';
        print '</tr>';
        foreach ($tblBomList as $value) {
        }
        print '</table>';
        print "<br>";
    }

    if (count($prodsfather) > 0) {
        print '<b>'.$langs->trans("FactoryParentTableInfo").'</b>';
        // on ajoute la possibilité de remplacer un produit par un autre
        print '<form action="index.php?id='.$id.'" method="post">';
        print '<input type="hidden" name="token" value="'.$_SESSION['newtoken'].'">';
        print '<input type="hidden" name="action" value="remplacementproduct">';

        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">'.$langs-
>trans("RealStock").'</td>';
        print '<td class="liste_titre" width=100px align="right">'.$langs-
>trans("UnitPmp").'</td>';
        print '<td class="liste_titre" width=100px align="right">'.$langs-
>trans("SellingPriceHT").'</td>';
        print '<td class="liste_titre" width=50px align="right">'.$langs-
>trans("Sel").'</td>';

        print '</tr>';
        foreach ($prodsfather as $value) {
            print '<tr>';
            print '<td>'.$factory->getNomUrlFactory($value['id'], 1, 'index');
            print $factory->PopupProduct($value['id']);
            print '</td>';
        }
    }
}
```

```
print '<td>'. $value["label"]. '</td>';;
$productstatic->fetch($value["id"]);
if ($value['fk_product_type']==0) {
    if ($conf->stock->enabled) {
        $productstatic->load_stock();
        print '<td align=center>'. $productstatic->stock_reel. '</td>';
    } else
        print '<td align=center></td>';
} else {
    // no stock managment for the services
    print '<td></td>';
}
print '<td align="right">'. price($productstatic->pmp). '</td>';
print '<td align="right">'. price($productstatic->price). '</td>';
print '<td align="right"><input type=checkbox name="productchange[]"
value="'. $value["id"]. '"></td>';

    print '</tr>';
}
print '<tr class="liste_titre"><th align=right colspan=2>';
print $langs->trans("AlternativeProduct"). '&nbsp;:;&nbsp;';
print '</th>';
print '<td colspan=2>';
print $form->select_produits($id, 'fk_product', '');
print '</td>';
print '<td align=right colspan=2>';
print '<input type="submit" class="button" value="'. $langs-
>trans("ChangeProduct"). '">';
print '</td></tr>';
print '</table>';

}

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

    print '<form action="index.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>';

    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).'
```

```
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="index.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">'. $langs->
>trans("SellPrice").' </th>';
    if ($conf->stock->enabled)
    print '<th class="liste_titre" align="right">'. $langs->
>trans("Stock").' </th>';
    print '<th class="liste_titre" align="right">'. $langs->
>trans("QtyOrder").' </th>';
    print '<th class="liste_titre" align="right">'. $langs->
>trans("Quantity").' </th>';
    print '<th class="liste_titre" align="center">'. $langs->
>trans("AddDel").' </th>';
    print '<th class="liste_titre" align="right">'. $langs->trans("Global").' </th>';
    print '</tr>';

    if ($resqlsearch) {
        $num = $db->num_rows($resqlsearch);
        $i=0;
        $var=true;

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

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

            print '<td><label id=line' . $objp->rowid .'">';
            print $factory->getNomUrlFactory($objp->rowid, 1, 'index', 24,
$productstatic->ref);
            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='#line".$objp->rowid."'
onclick=\"$( '.detailligne".$i.'').toggle();\" >";
print img_picto("", "edit_add")."</a>&nbsp;";
print $labelto show.'</td>';
if ($factory->is_sousproduit($id, $objp->rowid)) {
    $addchecked = ' checked="checked" ';
    $qty=$factory->is_sousproduit_qty;
    $descComposant=$factory->is_sousproduit_description;
    $ordercomponent=$factory->is_sousproduit_ordercomponent;
    $qtyglobal=$factory->is_sousproduit_qtyglobal;
} else {
    $addchecked = '';
    $descComposant = $objp->addinforecup;
    $qty="1";
    $ordercomponent="0"; // par d faut pas d'ordre
    $qtyglobal=0;
}
print '<td align="right">'.price($objp->pmp).'</td>';
print '<td align="right">'.price($objp->price).'</td>';

if ($conf->stock->enabled) {
    $productstatic->load_stock();
    print '<td align="right">'. $productstatic->stock_reel.'</td>';
} else
    print '<td ></td>';

if ($objp->fk_product_type==0) { // if product
    $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";
    $sql.= " and cofd.fk_product=".$objp->rowid;
    //print $sql;
    $resql = $db->query($sql);
    if ($resql) {
        $objpcmd = $db->fetch_object($resql);
        if ($objpcmd->nbCmdFourn)
            $nbcmd=$objpcmd->nbCmdFourn;
    }
    print '<td align="right">'. $nbcmd.'</td>';
}
else // no stock management for services
    print '<td></td>';

print '<td align="right" >';
print '<input type="text" size="3" name="prod_qty_'.$i.'"
value="'.$qty.'">';
print '</td><td align="center">';
print '<input type="checkbox" '.$addchecked.' name="prod_id_chk_'.$i.'"
value="'.$objp->rowid.'">';
print '<input type="hidden" name="prod_id_'.$i.'" value="'.$objp-
>rowid.'">';
print '</td>';
print '<td align="right">';

print $form->selectyesno('prod_id_globalchk_'.$i, $qtyglobal, 1);
print '</td></tr>';

if ($bc[$var]=='class="pair"')
```

```
        print "<tr style='display:none' class='pair detailligne".$i.">";
    else
        print "<tr style='display:none' class='impair detailligne".$i.">";
    print '<td ></td>';
    print '<td valign=top align=right>'.$langs->trans("Position").' : </td><td
valign=top>';
        print '<input type="text" size="2" name="prod_order_'.$i.'"
value="'.$ordercomponent.'">';
        print '</td>';
        print '<td valign=top align=right>'.$langs->trans("InfoAnnexes").' :
</td><td colspan=6>';
        print '<textarea name="descComposant'.$i.'" wrap="soft" cols="80"
rows="'.ROWS_2.'">';
        print $descComposant.'</textarea>';
        print '</td>';
        print '</tr>';

        $i++;
    }
} else
    dol_print_error($db);

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

if ($num > 0) {
    print '<br><center>';
    print '<input type="submit" class="button" value="'.$langs->
>trans("Add").'/'.$langs->trans("Update").'>';
    print ' &nbsp; &nbsp; <input type="submit" name="cancel" class="button"
value="'.$langs->trans("Cancel").'>';
    print '</center>';
}
print '</form>';
}
}
}
}

/* Barre d'action */
print '<div class="tabsAction">';
$object->fetch($id, $ref);

if ($action == '' && $bproduit) {
    if ($user->rights->factory->creer) {
        //Le stock doit être actif et le produit doit être identifié manufacturé
        if ($conf->stock->enabled && $object->finished == '1') {
            print '<a class="butAction" href="index.php?action=edit&id='.$productid.'">';
            print $langs->trans("EditComponent").'</a>';
            print '<a class="butAction"
href="index.php?action=importexport&id='.$productid.'">';
            print $langs->trans("ImportExportComposition").'</a>';

            if ($compositionpresente) {
                // gestion des prix
                print '<a class="butAction"
href="index.php?action=getdefaultprice&id='.$productid.'">';
                print $langs->trans("GetDefaultPrice").'</a>';
                print '<a class="butAction"
href="index.php?action=adjustprice&id='.$productid.'">';
                print $langs->trans("AdjustPrice").'</a>';
            } else {
                //uniquement si les produits virtuels sont actifs
                if (! empty($conf->global->PRODUIT_SOUSPRODUITS)) {
```

```
        print '<a class="butAction"
href="index.php?action=getfromvirtual&id='.$productid.'">';
        print $langs->trans("GetFromVirtual").'</a>';
    }
} else
    print $langs->trans("NeedFinishedProductAndStockEnabled");
}
}
print '</div>';
llxFooter();
$db->close();
```