Ce code source a été écrit par Mlle NAJMAN Sandra, étudiante en Licence Pro ATC STIC, IUT DE VELIZY

```php
<?php
/**
 * Magento
 *
 * NOTICE OF LICENSE
 *
 * This source file is subject to the Open Software License (OSL 3.0)
 * that is bundled with this package in the file LICENSE.txt.
 * It is also available through the world-wide-web at this URL:
 * http://opensource.org/licenses/osl-3.0.php
 * If you did not receive a copy of the license and are unable to
 * obtain it through the world-wide-web, please send an email
 * to license@magentocommerce.com so we can send you a copy immediately.
 *
 * DISCLAIMER
 *
 * Do not edit or add to this file if you wish to upgrade Magento to newer
 * versions in the future. If you wish to customize Magento for your
 * needs please refer to http://www.magentocommerce.com for more information.
 *
 * @category   Mage
 * @package    Mage_Sales
 * @copyright  Copyright (c) 2008 Irubin Consulting Inc. DBA Varien (http://www.varien.com)
 * @license    http://opensource.org/licenses/osl-3.0.php  Open Software License (OSL 3.0)
 */


/**
 * Sales Order PDF abstract model
 *
 * @category   Mage
 * @package    Mage_Sales
 * @author     Magento Core Team <core@magentocommerce.com>
 */
abstract class Mage_Sales_Model_Order_Pdf_Abstract extends Varien_Object
{
    public $y;
    /**
     * Item renderers with render type key
     *
     * model    => the model name
     * renderer => the renderer model
     *
     * @var array
     */
    protected $_renderers = array();

    const XML_PATH_SALES_PDF_INVOICE_PUT_ORDER_ID = 'sales_pdf/invoice/put_order_id';
    const XML_PATH_SALES_PDF_SHIPMENT_PUT_ORDER_ID = 'sales_pdf/shipment/put_order_id';
    const XML_PATH_SALES_PDF_CREDITMEMO_PUT_ORDER_ID =
```

```
'sales_pdf/creditmemo/put_order_id';

    /**
     * Zend PDF object
     *
     * @var Zend_Pdf
     */
    protected $_pdf;

    /**
     * Retrieve PDF
     *
     * @return Zend_Pdf
     */
    abstract public function getPdf();

    /**
     * Returns the total width in points of the string using the specified font and
     * size.
     *
     * This is not the most efficient way to perform this calculation. I'm
     * concentrating optimization efforts on the upcoming layout manager class.
     * Similar calculations exist inside the layout manager class, but widths are
     * generally calculated only after determining line fragments.
     *
     * @param string $string
     * @param Zend_Pdf_Resource_Font $font
     * @param float $fontSize Font size in points
     * @return float
     */
    public function widthForStringUsingFontSize($string, $font, $fontSize)
    {
        $drawingString = iconv('UTF-8', 'UTF-16BE//IGNORE', $string);
        $characters = array();
        for ($i = 0; $i < strlen($drawingString); $i++) {
            $characters[] = (ord($drawingString[$i++]) << 8) | ord($drawingString[$i
]);
        }
        $glyphs = $font->glyphNumbersForCharacters($characters);
        $widths = $font->widthsForGlyphs($glyphs);
        $stringWidth = (array_sum($widths) / $font->getUnitsPerEm()) * $fontSize;
        return $stringWidth;

    }

    /**
     * Calculate coordinates to draw something in a column aligned to the right
     *
     * @param string $string
     * @param int $x
     * @param int $columnWidth
     * @param Zend_Pdf_Resource_Font $font
     * @param int $fontSize
     * @param int $padding
     * @return int
     */
    public function getAlignRight($string, $x, $columnWidth, Zend_Pdf_Resource_Font
```

```
$font, $fontSize, $padding = 5)
    {
        $width = $this->widthForStringUsingFontSize($string, $font, $fontSize);
        return $x + $columnWidth - $width - $padding;
    }

    /**
     * Calculate coordinates to draw something in a column aligned to the center
     *
     * @param string $string
     * @param int $x
     * @param int $columnWidth
     * @param Zend_Pdf_Resource_Font $font
     * @param int $fontSize
     * @return int
     */
    public function getAlignCenter($string, $x, $columnWidth, Zend_Pdf_Resource_Font
$font, $fontSize)
    {
        $width = $this->widthForStringUsingFontSize($string, $font, $fontSize);
        return $x + round(($columnWidth - $width) / 2);
    }

    protected function insertLogo(&$page, $store = null)
    {
        $image = Mage::getStoreConfig('sales/identity/logo', $store);
        if ($image) {
            $image = Mage::getStoreConfig('system/filesystem/media', $store) .
'/sales/store/logo/' . $image;
            if (is_file($image)) {
                $image = Zend_Pdf_Image::imageWithPath($image);
                $page->drawImage($image, 330, 775, 580, 825);
            }
        }
        //return $page;
    }

    protected function insertAddress(&$page, $store = null)
    {
        /*Modif1
       encadre gris fin*/
        $page->setLineWidth(0.5);
        $page->setLineColor(new Zend_Pdf_Color_GrayScale(0.5));
        /* couleur bleue moyen*/
        $page->setFillColor(new Zend_Pdf_Color_Html('#F8F7F5'));
        $page->setFillColor(new Zend_Pdf_Color_GrayScale(0));
         $font = Zend_Pdf_Font::fontWithName(Zend_Pdf_Font::FONT_HELVETICA);

        //Applique la police et dessine du texte
        //$page->setFont($font, 18);

        //$page->drawText('FACTURE', 25, 630,'UTF-8');
        $this->_setFontRegular($page, 16);
        $font = Zend_Pdf_Font::fontWithName(Zend_Pdf_Font::FONT_HELVETICA);

        // Applique la police et dessine du texte
        $page->setFont($font, 9);
```

```
163         $page->setFillColor(new Zend_Pdf_Color_GrayScale(0));
164         $page->setFillColor(new Zend_Pdf_Color_Html('#D9E5EE'));
165
166         /* Dessine le bloc haut droite, bleu moyen et gris clair */
167           $page->setLineWidth(0.5);
168         $page->setLineColor(new Zend_Pdf_Color_GrayScale(0.5));
169           $page->setFillColor(new Zend_Pdf_Color_Html('#D9E5EE'));
170         $page->drawRectangle(340, 760, 457, 740);//Entêteg1
171         $page->setFillColor(new Zend_Pdf_Color_Html('#F8F7F5'));
172          $page->drawRectangle(457, 760, 570, 740);//Entêted1
173           $page->setFillColor(new Zend_Pdf_Color_Html('#D9E5EE'));
174           $page->drawRectangle(340, 740, 457, 720);//Entêteg2
175            $page->setFillColor(new Zend_Pdf_Color_Html('#F8F7F5'));
176          $page->drawRectangle(457, 740, 570, 720);//Entêted2
177           $page->setFillColor(new Zend_Pdf_Color_Html('#D9E5EE'));
178           $page->drawRectangle(340, 720, 457, 700);//Entêteg3
179            $page->setFillColor(new Zend_Pdf_Color_Html('#F8F7F5'));
180          $page->drawRectangle(457, 720, 570, 700);//Entêted3
181            $page->setFillColor(new Zend_Pdf_Color_Html('#D9E5EE'));
182           $page->drawRectangle(340, 700, 457, 680);//Entêteg4
183            $page->setFillColor(new Zend_Pdf_Color_Html('#F8F7F5'));
184          $page->drawRectangle(457, 700, 570, 680);//Entêted4
185           $page->setFillColor(new Zend_Pdf_Color_GrayScale(0));
186          $font = Zend_Pdf_Font::fontWithName(Zend_Pdf_Font::FONT_HELVETICA);
187          //Applique la police et dessine du texte
188     $page->setFont($font, 10);
189         $page->drawText('Date', 390, 746,'CP1252');//g1
190         $page->drawText('N° Client', 380, 726,'CP1252');//g2
191         $page->drawText('N° Commande', 368, 706,'CP1252');//g3
192           // $page->drawText(' N° Facture', 372, 686,'CP1252');//g4 dans invoice.php
193
194             /*     $date = new Zend_Date();
195                       $value = $date->get(Zend_Date::DATE_MEDIUM);
196                $page->drawText( $value, 490, 746, 'UTF-8');d1*/
197
198
    //$page->drawText($this->getCustomer($order)->getIncrementId(),435,726);//d2
199
200
201
    //$page->drawText(Mage::helper('sales')->__('').$order->getRealOrderId(), 435, 706,
    'UTF-8');
202
203
204
205          $page->setFillColor(new Zend_Pdf_Color_GrayScale(0));
206          $font = Zend_Pdf_Font::fontWithName(Zend_Pdf_Font::FONT_HELVETICA_BOLD);
207         // Applique la police et dessine du texte
208         $page->setFont($font, 14);
209          //$page->drawText('CONTACT', 307, 660,'UTF-8');
210         $page->drawText('LUMINOL', 25, 800,'UTF-8');
211         $page->setFillColor(new Zend_Pdf_Color_GrayScale(0));
212          $font = Zend_Pdf_Font::fontWithName(Zend_Pdf_Font::FONT_HELVETICA);
213         // Applique la police et dessine du texte
214         $page->setFont($font, 10);
215         $page->drawText('ADRESSE FOURNISSEUR', 25, 780,'UTF-8');
216         $page->drawText('ADRESSE FOURNISSEUR', 25, 768, 'UTF-8');
```

```
217         $page->drawText('Tél. :  XX XX XX XX XX', 25, 756, 'CP1252');
218         $page->drawText('Fax :  XX XX XX XX XX', 25, 744, 'CP1252');
219         $page->drawText('Mail : XXXXXXXXXXXXXX', 25, 732, 'CP1252');
220         $page->drawText('Site : XXXXXXXXXXXXXX', 25, 720, 'CP1252');
221         $page->drawText('SARL au capital de XXXXX Euros', 25, 700,'CP1252');
222         $page->drawText('RCS Nanterre : XXXXXXXXX', 25, 688, 'UTF-8');
223         $page->drawText('TVA : FRXXXXXXXXXX', 25, 676, 'UTF-8');
224
225          $page->setFillColor(new Zend_Pdf_Color_GrayScale(0));
226          //$font = Zend_Pdf_Font::fontWithName(Zend_Pdf_Font::FONT_HELVETICA_BOLD);
227         //$page->setFont($font, 24);
228
229         //$page->drawText('FACTURE :', 25, 635,'UTF-8');//FactureGros
230
231     /*Clauses*/
232
233
234      $font = Zend_Pdf_Font::fontWithName(Zend_Pdf_Font::FONT_HELVETICA);
235      $page->setFont($font, 6);
236
237     $this->insertFooter($page,$store);// vers ligne813
238
239
240      /*endModif1*/
241         $page->setLineWidth(0);
242        // $this->y = 820;
243          //$this->y = 730;
244          $this->y = 630;
245         foreach (explode("\n", Mage::getStoreConfig('sales/identity/address', $store
    )) as $value){
246             if ($value!==''') {
247                // $page->drawText(trim(strip_tags($value)), 30, $this->y, 'UTF-8');
248                 $this->y -=10;
249             }
250         }
251         //return $page;
252     }
253
254     /**
255      * Format address
256      *
257      * @param string $address
258      * @return array
259      */
260     protected function _formatAddress($address)
261     {
262         $return = array();
263         foreach (split('\|', $address) as $str) {
264             foreach (Mage::helper('core/string')->str_split($str, 65, true, true) as
    $part) {
265                 if (empty($part)) {
266                     continue;
267                 }
268                 $return[] = $part;
269             }
270         }
271         return $return;
```

```
272         }
273     /**
274          * Retrieve Customer if available
275          * @param Mage_Order $order
276          */
277         public function getCustomer($order)
278         {
279             if (!$this->_customer instanceof Mage_Customer_Model_Customer) {
280                 $this->_customer = Mage::getModel('customer/customer')->load($order->
    getCustomerId());
281             }
282             return $this->_customer;
283         }
284
285         /**
286          * Insert header with some numbers and dates.
287          * @param Zend_Pdf_Page $page
288          * @param Mage_Order $order
289          * @param bool $putCustomerId
290          * @param bool $putOrderId
291          */
292         protected function insertHeader(&$page, $order, $putCustomerId = true,
    $putOrderId = true)
293         {
294
295
296             if ($putOrderId) {
297                 //$page->drawText(Mage::helper('sales')->__('Order #:
    ').$order->getRealOrderId(), ;
298             }
299            // $page->drawText(Mage::helper('sales')->__('Order Date: ')
    .Mage::helper('core')->formatDate($order->getCreatedAt(), 'medium', false), 76,
    $this->y,self::PDF_CHARSET);
300
301             if ( $putCustomerId && $this->getCustomer($order)->getIncrementId() != "" ) {
302                  $font = Zend_Pdf_Font::fontWithName(Zend_Pdf_Font::FONT_HELVETICA);
303
304     // Applique la police et dessine du texte
305     $page->setFont($font, 8);
306                 $page->drawText(Mage::helper('sales')->__('Customer #: ') .$this->
    getCustomer($order)->getIncrementId(),355,726);
307
308             }
309
310                 }
311
312         protected function insertOrder(&$page, $order, $putOrderId = true)
313         {
314
315
316              $page->setFillColor(new Zend_Pdf_Color_GrayScale(0));
317              $font = Zend_Pdf_Font::fontWithName(Zend_Pdf_Font::FONT_HELVETICA);
318             $page->setFont($font, 10);
319
320
321             if ($putOrderId) {
322
```

```
323         //$page->drawText('N° de : ' . $order->getData('customer_taxvat'), 35, 660, 'CP1252');
324
325                 $page->drawText($this->getCustomer($order)->getIncrementId(),490,726);
326                             //$page->drawText(Mage::helper('sales')->__('Code Client: ') .$this->getCustomer($order)->getIncrementId(),435,726);
327                             $page->drawText($order->getRealOrderId(), 490, 706, 'UTF-8');
328
329                 //$page->drawText(Mage::helper('sales')->__('Commande : ').$order->getRealOrderId(), 355, 790, 'UTF-8');
330             }
331
332             $page->setFillColor(new Zend_Pdf_Color_Html('#D9E5EE'));
333             $page->setLineColor(new Zend_Pdf_Color_GrayScale(0.5));
334             $page->setLineWidth(0.5);
335             $h=600;//Position du bloc sous FACTURE
336                $page->drawRectangle(25, $h, 297, $h-25);//Informations facturation
337                $page->drawRectangle(297, $h, 570, $h-25);//Informations livraison
338             /* Calculate blocks info */
339
340             /* Billing Address */
341             $billingAddress = $this->_formatAddress($order->getBillingAddress()->format('pdf'));
342
343             /* Payment */
344             $paymentInfo = Mage::helper('payment')->getInfoBlock($order->getPayment())
345                 ->setIsSecureMode(true)
346                 ->toPdf();
347             $payment = explode('{{pdf_row_separator}}', $paymentInfo);
348             foreach ($payment as $key=>$value){
349                 if (strip_tags(trim($value))==''){
350                     unset($payment[$key]);
351                 }
352             }
353             reset($payment);
354
355             /* Shipping Address and Method */
356             if (!$order->getIsVirtual()) {
357                 /* Shipping Address */
358                 $shippingAddress = $this->_formatAddress($order->getShippingAddress()->format('pdf'));
359
360                 $shippingMethod  = $order->getShippingDescription();
361             }
362
363             $page->setFillColor(new Zend_Pdf_Color_GrayScale(0));
364             //$this->_setFontRegular($page);
365             $font = Zend_Pdf_Font::fontWithName(Zend_Pdf_Font::FONT_HELVETICA);
366
367             // Applique la police et dessine du texte
368             $page->setFont($font, 9);
369
370             $page->drawText(Mage::helper('sales')->__('SOLD TO:'), 35, $h-15 , 'UTF-8');
371
372             if (!$order->getIsVirtual()) {
373                 $page->drawText(Mage::helper('sales')->__('SHIP TO:'), 307, $h-15 , 'UTF-8');
```

```
374         }
375         else {
376             $page->drawText(Mage::helper('sales')->__('Payment Method:'), 307, $h-15
    , 'UTF-8');
377         }
378
379         //if (!$order->getIsVirtual()) {
380             //$y = $h-25 - (max(count($billingAddress), count($shippingAddress)) *
    10 + 5);
381         //}
382         //else {
383           //  $y = $h-25 - (count($billingAddress) * 10 + 5);
384         //
385           if (!$order->getIsVirtual()) {
386             $y = $h-25 - (max(count($billingAddress), count($shippingAddress)) * 10 +
     5);///
387             if ($order->getData('customer_taxvat'))
388             {
389             $y-=10;
390             $page->drawText('CGI article 262 Ter 1', 25, 60, 'CP1252');
391
392              }
393
394
395
396         }
397         else {
398             $y = $h-25 - (count($billingAddress) * 10 + 5);
399         }
400
401                 $pays=Mage::app()->getLocale()->getCountryTranslation($order->
    getBillingAddress()->getCountry());
402             //$page->drawText('Pays : '.$pays, 25, 70, 'CP1252');//Affichage du Pays
    en debug
403
404         if ($pays=='Guadeloupe'||$pays=='Martinique'||$pays=='Guyane'||$pays==
    'Réunion'||$pays=='Mayotte' )
405
406             {
407             $page->drawText('CGI article 294', 25, 60, 'CP1252');
408               }
409
410         //$page->setFillColor(new Zend_Pdf_Color_GrayScale(1));
411        // $page->setFillColor(new Zend_Pdf_Color_Html('#F8F7F5'));//
412         $page->setFillColor(new Zend_Pdf_Color_Html('#FFFFFF'));
413         //$page->drawRectangle(25, $h-25, 570, $y);
414         $page->drawRectangle(25, $h-25, 297, $y);//Adresse de F
415         $page->drawRectangle(297, $h-25, 570, $y);//Adresse de L
416
417         $page->setFillColor(new Zend_Pdf_Color_GrayScale(0));
418         //$page->setFillColor(new Zend_Pdf_Color_Html('#F8F7F5'));
419         $this->_setFontRegular($page,9);
420         $this->y = $h-35;//
421
422         foreach ($billingAddress as $value){
423             if ($value!==''') {
424                 $page->drawText(strip_tags(ltrim($value)), 35, $this->y, 'UTF-8');
```

```
425                 $this->y -=10;
426             }
427         }
428
429         if (!$order->getIsVirtual()) {
430
431
432             $this->y = $h-35;
433             foreach ($shippingAddress as $value){
434                 if ($value!==''') {
435                     $page->drawText(strip_tags(ltrim($value)), 307, $this->y, 'UTF-8'
    );
436                     $this->y -=10;
437                 }
438
439             }
440
441          if ($order->getData('customer_taxvat'))
442                      {
443                     $page->drawText('N° de TVA : '. $order->getData('customer_taxvat'
    ), 35, $this->y, 'CP1252');
444                  $this->y -=20;
445                   }
446         }
447    //$this->insertOrderMode(&$page, $order, $putOrderId = true) ;
448     $this->y =$y-10;
449     }
450
451     protected function insertOrderMode(&$page, $order, $putOrderId = true)
452     {
453          if (!$order->getIsVirtual()) {
454
455             $page->setFillColor(new Zend_Pdf_Color_Html('#D9E5EE'));
456             $page->setLineWidth(0.5);
457             $page->drawRectangle(25, $this->y, 297, $this->y-25);//Mode de P
458             $page->drawRectangle(297, $this->y, 570, $this->y-25);//Mode de L
459
460             $this->y -=15;
461             $font = Zend_Pdf_Font::fontWithName(Zend_Pdf_Font::FONT_HELVETICA);
462
463             // Applique la police et dessine du texte
464             $page->setFont($font, 9);
465
466             // $this->_setFontBold($page);
467             $page->setFillColor(new Zend_Pdf_Color_GrayScale(0));
468             //$page->drawText(Mage::helper('sales')->__('Mode de paiement :'), 35,
    $this->y, 'UTF-8');
469             $page->drawText(Mage::helper('sales')->__('Payment Method:'), 35, $this->
    y, 'UTF-8');
470             $page->drawText(Mage::helper('sales')->__('Shipping Method:'), 307, $this
    ->y , 'UTF-8');
471
472             $this->y -=10;
473             $page->setFillColor(new Zend_Pdf_Color_GrayScale(1));
474
475             //$this->_setFontRegular($page);
476             $font = Zend_Pdf_Font::fontWithName(Zend_Pdf_Font::FONT_HELVETICA);
```

```
477
478             // Applique la police et dessine du texte
479             $page->setFont($font, 9);
480
481             $page->setFillColor(new Zend_Pdf_Color_GrayScale(0));
482
483             $paymentLeft = 35;
484             $yPayments   = $this->y - 15;
485         }
486         else {
487             $yPayments   = 620;
488             $paymentLeft = 307;
489         }
490
491         $paymentInfo = Mage::helper('payment')->getInfoBlock($order->getPayment())
492             ->setIsSecureMode(true)
493             ->toPdf();
494         $payment = explode('{{pdf_row_separator}}', $paymentInfo);
495
496
497         foreach ($payment as $value){
498             if (trim($value)!=='') {
499                 $page->drawText(strip_tags(trim($value)), $paymentLeft, $yPayments,
    'UTF-8');//mode de paiement
500                 $yPayments -=10;
501             }
502         }
503
504         if (!$order->getIsVirtual()) {
505             $this->y -=15;
506              $shippingMethod  = $order->getShippingDescription();
507             $page->drawText($shippingMethod, 307, $this->y, 'UTF-8');//mode de
    livraison
508
509             $yShipments = $this->y;
510
511
512             $totalShippingChargesText = "(" . Mage::helper('sales')->__('Total
    Shipping Charges') . " " . $order->formatPriceTxt($order->getShippingAmount()) . ")";
513
514             $page->drawText($totalShippingChargesText, 307, $yShipments-9, 'UTF-8');
    //frais de livraison
515             $yShipments -=10;
516             $tracks = $order->getTracksCollection();
517             if (count($tracks)) {
518                 $page->setFillColor(new Zend_Pdf_Color_Rgb(0.93, 0.92, 0.92));
519                 $page->setLineWidth(0.5);
520                 $page->drawRectangle(307, $yShipments, 510, $yShipments - 10);
521                 $page->drawLine(380, $yShipments, 380, $yShipments - 10);
522                 //$page->drawLine(510, $yShipments, 510, $yShipments - 10);
523
524                 $this->_setFontRegular($page);
525                 $page->setFillColor(new Zend_Pdf_Color_GrayScale(0));
526                 //$page->drawText(Mage::helper('sales')->__('Carrier'), 290,
    $yShipments - 7 , 'UTF-8');
527                 $page->drawText(Mage::helper('sales')->__('Title'), 290, $yShipments
    - 7, 'UTF-8');
```

```
                $page->drawText(Mage::helper('sales')->__('Number'), 385, $yShipments - 7, 'UTF-8');

                $yShipments -=17;
                $this->_setFontRegular($page, 6);
                foreach ($order->getTracksCollection() as $track) {

                    $CarrierCode = $track->getCarrierCode();
                    if ($CarrierCode!='custom')
                    {
                        $carrier = Mage::getSingleton('shipping/config')->getCarrierInstance($CarrierCode);
                        $carrierTitle = $carrier->getConfigData('title');
                    }
                    else
                    {
                        $carrierTitle = Mage::helper('sales')->__('Custom Value');
                    }

                    //$truncatedCarrierTitle = substr($carrierTitle, 0, 35) . (strlen($carrierTitle) > 35 ? '...' : '');
                    $truncatedTitle = substr($track->getTitle(), 0, 45) . (strlen($track->getTitle()) > 45 ? '...' : '');
                    //$page->drawText($truncatedCarrierTitle, 307, $yShipments , 'UTF-8');
                    $page->drawText($truncatedTitle, 300, $yShipments , 'UTF-8');
                    $page->drawText($track->getNumber(), 395, $yShipments , 'UTF-8');
                    $yShipments -=7;
                }
            } else {
                $yShipments -= 7;
            }

            $currentY = min($yPayments, $yShipments);

            // replacement of Shipments-Payments rectangle block
            $page->drawLine(25, $this->y + 15, 25, $currentY);//trait vertical gauche
            $page->drawLine(25, $currentY, 570, $currentY);//trait horizontal bas
            $page->drawLine(297, $currentY, 297, $this->y + 15);//trait vertical central
            $page->drawLine(570, $currentY, 570, $this->y + 15);//trait vertical droit

            $this->y = $currentY;
            $this->y -= 15;
        }
    }


    protected function insertRectangle($page,$xg)

    {

        $page->setFillColor(new Zend_Pdf_Color_Html('#D9E5EE'));
         $page->setLineWidth(0.5);
         $page->setLineColor(new Zend_Pdf_Color_GrayScale(0.5));

            $page->drawRectangle($xg-10, $this->y+10, 490, $this->y-5);//
```

```
labelTableau  bas facture

            $page->setLineWidth(0.5);
        $page->setLineColor(new Zend_Pdf_Color_GrayScale(0.5));

        $page->setFillColor(new Zend_Pdf_Color_Html('#F8F7F5'));

         $page->drawRectangle(490, $this->y+10, 570, $this->y-5);//amountTableau bas facture

         $page->setFillColor(new Zend_Pdf_Color_GrayScale(0));

    }
    /*************************************************************/
    protected function insertTotals(&$page, $source){
        $order = $source->getOrder();
        $font = $this->_setFontBold($page,8);
        $this->y -=15;
        $xg=400;
        //$order_subtotal = Mage::helper('sales')->__('Order Subtotal:');
        /* if ($source->getAdjustmentPositive()){
            $adjustment_refund = Mage::helper('sales')->__('Adjustment Refund:');//remb de frais
            $xg= 475-$this->widthForStringUsingFontSize( $adjustment_refund, $font, 8);
        }*/

         $this->insertRectangle($page,$xg);
        $order_subtotal = 'Montant HT :';
        $page->drawText($order_subtotal, 475-$this->widthForStringUsingFontSize( $order_subtotal, $font, 8), $this->y, 'UTF-8');

        $order_subtotal = $order->formatPriceTxt($source->getSubtotal());
        $page->drawText($order_subtotal, 565-$this->widthForStringUsingFontSize( $order_subtotal, $font, 8), $this->y, 'UTF-8');
        $this->y -=15;

        if ((float)$source->getDiscountAmount()){
             $this->insertRectangle($page,$xg);

            $discount = Mage::helper('sales')->__('Discount :');
            $page->drawText($discount, 475-$this->widthForStringUsingFontSize( $discount, $font, 8), $this->y, 'UTF-8');

            $discount = $order->formatPriceTxt(0.00 - $source->getDiscountAmount());
            $page->drawText($discount, 565-$this->widthForStringUsingFontSize( $discount, $font, 8), $this->y, 'UTF-8');
            $this->y -=15;
        }

       /* if ((float)$source->getTaxAmount()){
            $fullInfo = $order->getFullTaxInfo();//
            if ( $fullInfo)
             {
                foreach ($fullInfo as $info)
                {
                    $percent = $info['percent'];
```

```
                    $amount = $info['amount'];
                    $rates = array_unique($info['rates']);
                }
                    foreach ($rates as $rate)
                     {
                       $order_tax = 'TVA ('.$rate['percent'] . "%) : " ;
                    }





             $this->insertRectangle($page,$xg);
            //$order_tax ='TVA :';
            //$order_tax = Mage::helper('sales')->__('Tax :');
            $page->drawText($order_tax, 475-$this->widthForStringUsingFontSize($order_tax, $font, 8), $this->y, 'UTF-8');
            }


            $order_tax = $order->formatPriceTxt($source->getTaxAmount());
            $page->drawText($order_tax, 565-$this->widthForStringUsingFontSize($order_tax, $font, 8), $this->y, 'UTF-8');
            $this->y -=15;

        }
*/          if ((float)$source->getShippingAmount()){
             $this->insertRectangle($page,$xg);
            $order_shipping = Mage::helper('sales')->__('Shipping & Handling:');
            $page->drawText($order_shipping, 475-$this->widthForStringUsingFontSize($order_shipping, $font, 8), $this->y, 'UTF-8');

            $order_shipping = $order->formatPriceTxt($source->getShippingAmount());
            $page->drawText($order_shipping, 565-$this->widthForStringUsingFontSize($order_shipping, $font, 8), $this->y, 'UTF-8');
            $this->y -=15;
        }

       /* if ($source->getAdjustmentPositive()){

                        $this->insertRectangle($page,$xg);
            $adjustment_refund = Mage::helper('sales')->__('Adjustment Refund:');//remb de frais
          //$page ->drawText($adjustment_refund, $xg, $this->y, 'UTF-8');

            $adjustment_refund = $order->formatPriceTxt($source->getAdjustmentPositive());
            $page ->drawText($adjustment_refund, 565-$this->widthForStringUsingFontSize($adjustment_refund, $font, 8), $this->y, 'UTF-8');
            $this->y -=15;
        }
*/
        if ((float) $source->getAdjustmentNegative()){
             $this->insertRectangle($page,$xg);
            $adjustment_fee = Mage::helper('sales')->__('Adjustment Fee:');
```

```
677             $page ->drawText($adjustment_fee, 475-$this->widthForStringUsingFontSize(
    $adjustment_fee, $font, 8), $this->y, 'UTF-8');
678
679             $adjustment_fee=$order->formatPriceTxt($source->getAdjustmentNegative());
680             $page ->drawText($adjustment_fee, 565-$this->widthForStringUsingFontSize(
    $adjustment_fee, $font, 8), $this->y, 'UTF-8');
681             $this->y -=15;
682         }
683
684 /*if ((float)$source->getTaxAmount()){*/
685             $fullInfo = $order->getFullTaxInfo();//
686             $order_tax= "TVA (0%) :";
687             if ( $fullInfo)
688              {
689                 foreach ($fullInfo as $info)
690                 {
691                     $percent = $info['percent'];
692                     $amount = $info['amount'];
693                     $rates = array_unique($info['rates']);
694                 }
695                     foreach ($rates as $rate)
696                      {
697                        $order_tax = 'TVA ('.$rate['percent'] . "%) : " ;
698                     }
699
700
701
702
703              }
704
705              $this->insertRectangle($page,$xg);
706             //$order_tax ='TVA :';
707             //$order_tax = Mage::helper('sales')->__('Tax :');
708             $page->drawText($order_tax, 475-$this->widthForStringUsingFontSize(
    $order_tax, $font, 8), $this->y, 'UTF-8');
709
710
711
712             $order_tax = $order->formatPriceTxt($source->getTaxAmount());
713             $page->drawText($order_tax, 565-$this->widthForStringUsingFontSize(
    $order_tax, $font, 8), $this->y, 'UTF-8');
714             $this->y -=15;
715
716         //}
717
718         $page->setFont($font, 8);
719
720         $order_grandtotal = Mage::helper('sales')->__('Grand Total:');
721          $this->insertRectangle($page,$xg);
722         $page ->drawText($order_grandtotal, 475-$this->widthForStringUsingFontSize(
    $order_grandtotal, $font, 8), $this->y, 'UTF-8');
723
724         $order_grandtotal = $order->formatPriceTxt($source->getGrandTotal());
725         $page ->drawText($order_grandtotal, 565-$this->widthForStringUsingFontSize(
    $order_grandtotal, $font, 8), $this->y, 'UTF-8');
726         $this->y -=15;
727     }
```

```
728
729     protected function _parseItemDescription($item)
730     {
731         $matches = array();
732         $description = $item->getDescription();
733         if (preg_match_all('/<li.*?>(.*?)<\/li>/i', $description, $matches)) {
734             return $matches[1];
735         }
736
737         return array($description);
738     }
739
740     /**
741      * Before getPdf processing
742      *
743      */
744     protected function _beforeGetPdf() {
745         $translate = Mage::getSingleton('core/translate');
746         /* @var $translate Mage_Core_Model_Translate */
747         $translate->setTranslateInline(false);
748     }
749
750     /**
751      * After getPdf processing
752      *
753      */
754     protected function _afterGetPdf() {
755         $translate = Mage::getSingleton('core/translate');
756         /* @var $translate Mage_Core_Model_Translate */
757         $translate->setTranslateInline(true);
758     }
759
760     protected function _formatOptionValue($value, $order)
761     {
762         $resultValue = '';
763         if (is_array($value)) {
764             if (isset($value['qty'])) {
765                 $resultValue .= sprintf('%d', $value['qty']) . ' x ';
766             }
767
768             $resultValue .= $value['title'];
769
770             if (isset($value['price'])) {
771                 $resultValue .= " " . $order->formatPrice($value['price']);
772             }
773             return  $resultValue;
774         } else {
775             return $value;
776         }
777     }
778
779     protected function _initRenderer($type)
780     {
781         $node = Mage::getConfig()->getNode('global/pdf/'.$type);
782         foreach ($node->children() as $renderer) {
783             $this->_renderers[$renderer->getName()] = array(
784                 'model'     => (string)$renderer,
```

```
                'renderer'  => null
            );
        }
    }

    /**
     * Retrieve renderer model
     *
     * @throws Mage_Core_Exception
     * @return Mage_Sales_Model_Order_Pdf_Items_Abstract
     */
    protected function _getRenderer($type)
    {
        if (!isset($this->_renderers[$type])) {
            $type = 'default';
        }

        if (!isset($this->_renderers[$type])) {
            Mage::throwException(Mage::helper('sales')->__('Invalid renderer model'));
        }

        if (is_null($this->_renderers[$type]['renderer'])) {
            $this->_renderers[$type]['renderer'] = Mage::getSingleton($this->
_renderers[$type]['model']);
        }

        return $this->_renderers[$type]['renderer'];
    }

    /**
     * Public method of protected @see _getRenderer()
     *
     * Retrieve renderer model
     *
     * @param string $type
     * @return Mage_Sales_Model_Order_Pdf_Items_Abstract
     */
    public function getRenderer($type)
    {
        return $this->_getRenderer($type);
    }

    /**
     * Draw Item process
     *
     * @param Varien_Object $item
     * @param Zend_Pdf_Page $page
     * @param Mage_Sales_Model_Order $order
     * @return Zend_Pdf_Page
     */
    protected function _drawItem(Varien_Object $item, Zend_Pdf_Page $page,
Mage_Sales_Model_Order $order)
    {
        $type = $item->getOrderItem()->getProductType();
        $renderer = $this->_getRenderer($type);
        $renderer->setOrder($order);
        $renderer->setItem($item);
```

```
840         $renderer->setPdf($this);
841         $renderer->setPage($page);
842         $renderer->setRenderedModel($this);
843 
844         $renderer->draw();
845 
846         return $renderer->getPage();
847     }
848 
849     protected function _setFontRegular($object, $size = 7)
850     {
851       //$font = Zend_Pdf_Font::fontWithPath(Mage::getBaseDir() .
'/lib/LinLibertineFont/LinLibertineC_Re-2.8.0.ttf');
852         $font = Zend_Pdf_Font::fontWithName(Zend_Pdf_Font::FONT_HELVETICA);
853         $object->setFont($font, $size);
854         return $font;
855     }
856 
857     protected function _setFontBold($object, $size = 7)
858     {
859        // $font = Zend_Pdf_Font::fontWithPath(Mage::getBaseDir() .
'/lib/LinLibertineFont/LinLibertine_Bd-2.8.1.ttf');
860         $font = Zend_Pdf_Font::fontWithName(Zend_Pdf_Font::FONT_HELVETICA_ITALIC);
861          $font = Zend_Pdf_Font::fontWithName(Zend_Pdf_Font::FONT_HELVETICA_BOLD);
862         $object->setFont($font, $size);
863         return $font;
864     }
865 
866     protected function _setFontItalic($object, $size = 7)
867     {
868         //$font = Zend_Pdf_Font::fontWithPath(Mage::getBaseDir() .
'/lib/LinLibertineFont/LinLibertine_It-2.8.2.ttf');
869          $font = Zend_Pdf_Font::fontWithName(Zend_Pdf_Font::FONT_HELVETICA_ITALIC);
870         $object->setFont($font, $size);
871         return $font;
872     }
873 
874     /**
875      * Set PDF object
876      *
877      * @param Zend_Pdf $pdf
878      * @return Mage_Sales_Model_Order_Pdf_Abstract
879      */
880     protected function _setPdf(Zend_Pdf $pdf)
881     {
882         $this->_pdf = $pdf;
883         return $this;
884     }
885 
886     /**
887      * Retrieve PDF object
888      *
889      * @throws Mage_Core_Exception
890      * @return Zend_Pdf
891      */
892     protected function _getPdf()
893     {
```

```
894         if (!$this->_pdf instanceof Zend_Pdf) {
895             Mage::throwException(Mage::helper('sales')->__('Please define PDF object
    before using'));
896         }
897
898         return $this->_pdf;
899     }
900
901     /**
902      * Create new page and assign to PDF object
903      *
904      * @param array $settings
905      * @return Zend_Pdf_Page
906      */
907     public function newPage(array $settings = array())
908     {
909
910         $pageSize = !empty($settings['page_size']) ? $settings['page_size'] :
    Zend_Pdf_Page::SIZE_A4;
911         $page = $this->_getPdf()->newPage($pageSize);
912         $this->_getPdf()->pages[] = $page;
913         $this->y = 800;
914
915         return $page;
916     }
917
918         /**
919      * Insert Footer with company legal and payment informations on bottom of
    document.
920      * @param Zend_Pdf_Page $page
921      * @param string $store Store Id
922      */
923      protected function insertLegals(&$page, $store = null)
924       {
925         $this->footer_y = 30;
926         $font = $this->_setFontRegular($page, 6);
927 $page->drawText("CLAUSE LEGALE.", 25, $this->footer_y + 20, 'CP1252');
928 $page->drawText("CLAUSE LEGALE." , 25, $this->footer_y +12, 'CP1252');
929
930
931       }
932
933
934
935     protected function insertFooter(&$page, $store = null) {
936 //où commence la colonne en vertical
937  $this->footer_y = 30;
938
939 //$this->insertLegals($page,$store);
940
941
942              $_column1 = "       MARQUE SOCIETE
943 ADRESSE
944 ADRESSE";
945
946
947             $_column2 = "Tél : XX XX XX XX XX / Fax : XX XX XX XX XX
```

```
948             Mail : XXXXXXXXXXXXXXXX
949                 Site : XXXXXXXXXXX";
950
951                 $_column3 = "SARL au capital de XXXXXX Euros
952         RCS Ville : XXXXXXXX
953             TVA : FRXXXXXXXXXX";
954
955
956                 // Draw line
957                 $page->setLineWidth(0.5);
958                 $page->setLineColor(new Zend_Pdf_Color_GrayScale(0.5));
959                 if ( trim($_column1) != "" || trim($_column2) != "" || trim($_column3) !=
     "" )
960                     $page->drawLine(25, $this->footer_y+9, 570, $this->footer_y+9);
961
962                 // Draw first column
963                 $this->y = $this->footer_y;
964                 $this->x = 25;
965                 $font = $this->_setFontRegular($page, 6);
966                 foreach (explode("\n", $_column1) as $value ) {
967                     //$page->drawText(trim(strip_tags($value)), $this->x, $this->y,
     'CP1252'); // 45
968                     $page->drawText($value, $this->x, $this->y, 'CP1252'); // 45
969                     $this->y -=7;//interligne colonne1
970                 }
971
972                 // Draw second column
973                 if ( trim($_column2) != "" ) {
974                     $this->y = $this->footer_y;
975                     $this->x +=185;
976                     foreach (explode("\n", $_column2) as $value){
977                        // $page->drawText(trim(strip_tags($value)), $this->x,
     $this->y,'CP1252'); // 45
978                        $page->drawText($value, $this->x, $this->y,'CP1252'); // 45
979                         $this->y -=7;//interligne col 2
980                     }
981                     $this->x +=240;//
982                 } else {
983                     $this->x +=380;//
984                 }
985
986                 // Draw third column
987                 $this->y = $this->footer_y;
988                 $font = $this->_setFontRegular($page, 6);
989                 foreach (explode("\n", $_column3) as $value ) {
990                     //$page->drawText(trim(strip_tags($value)), $this->x, $this->y,
     'CP1252'); // 45
991                     $page->drawText($value, $this->x, $this->y, 'CP1252'); // 45
992                     $this->y -=7;
993                 }
994         }
995
996
997         /**
998          * Draw lines
999          *
1000         * draw items array format:
```

```
     * lines        array;array of line blocks (required)
     * shift        int; full line height (optional)
     * height       int;line spacing (default 10)
     *
     * line block has line columns array
     *
     * column array format
     * text         string|array; draw text (required)
     * feed         int; x position (required)
     * font         string; font style, optional: bold, italic, regular
     * font_file    string; path to font file (optional for use your custom font)
     * font_size    int; font size (default 7)
     * align        string; text align (also see feed parametr), optional left, right
     * height       int;line spacing (default 10)
     *
     * @param Zend_Pdf_Page $page
     * @param array $draw
     * @param array $pageSettings
     * @throws Mage_Core_Exception
     * @return Zend_Pdf_Page
     */
    public function drawLineBlocks(Zend_Pdf_Page $page, array $draw, array $pageSettings = array())
    {
        foreach ($draw as $itemsProp) {
            if (!isset($itemsProp['lines']) || !is_array($itemsProp['lines'])) {
                Mage::throwException(Mage::helper('sales')->__('Invalid draw line data. Please define "lines" array'));
            }
            $lines  = $itemsProp['lines'];
            $height = isset($itemsProp['height']) ? $itemsProp['height'] : 10;

            if (empty($itemsProp['shift'])) {
                $shift = 0;
                foreach ($lines as $line) {
                    $maxHeight = 0;
                    foreach ($line as $column) {
                        $lineSpacing = !empty($column['height']) ? $column['height'] : $height;
                        if (!is_array($column['text'])) {
                            $column['text'] = array($column['text']);
                        }
                        $top = 0;
                        foreach ($column['text'] as $part) {
                            $top += $lineSpacing;
                        }

                        $maxHeight = $top > $maxHeight ? $top : $maxHeight;
                    }
                    $shift += $maxHeight;
                }
                $itemsProp['shift'] = $shift;
            }

            if ($this->y - $itemsProp['shift'] < 15) {
                $page = $this->newPage($pageSettings);
            }
```

```
1055
1056            foreach ($lines as $line) {
1057                $maxHeight = 0;
1058                foreach ($line as $column) {
1059                    $fontSize  = empty($column['font_size']) ? 9 : $column[
    'font_size'];
1060                    if (!empty($column['font_file'])) {
1061                        $font = Zend_Pdf_Font::fontWithPath($column['font_file']);
1062                        $page->setFont($font);
1063                    }
1064                    else {
1065                        $fontStyle = empty($column['font']) ? 'regular' : $column[
    'font'];
1066                        switch ($fontStyle) {
1067                            case 'bold':
1068                                $font = $this->_setFontBold($page, $fontSize);
1069                                break;
1070                            case 'italic':
1071                                $font = $this->_setFontItalic($page, $fontSize);
1072                                break;
1073                            default:
1074                                $font = $this->_setFontRegular($page, $fontSize);
1075                                break;
1076                        }
1077                    }
1078
1079                    if (!is_array($column['text'])) {
1080                        $column['text'] = array($column['text']);
1081                    }
1082
1083                    $lineSpacing = !empty($column['height']) ? $column['height'] :
    $height;
1084                    $top = 0;
1085                    foreach ($column['text'] as $part) {
1086                        $feed = $column['feed'];
1087                        $textAlign = empty($column['align']) ? 'left' : $column[
    'align'];
1088                        $width = empty($column['width']) ? 0 : $column['width'];
1089                        switch ($textAlign) {
1090                            case 'right':
1091                                if ($width) {
1092                                    $feed = $this->getAlignRight($part, $feed, $width
    , $font, $fontSize);
1093                                }
1094                                else {
1095                                    $feed = $feed - $this->
    widthForStringUsingFontSize($part, $font, $fontSize);
1096                                }
1097                                break;
1098                            case 'center':
1099                                if ($width) {
1100                                    $feed = $this->getAlignCenter($part, $feed,
    $width, $font, $fontSize);
1101                                }
1102                                break;
1103                        }
1104                        $page->drawText($part, $feed, $this->y-$top, 'UTF-8');
```

```
1105                        $page->setLineWidth(0.5);
1106                        $page->setLineColor(new Zend_Pdf_Color_GrayScale(0.5));
1107                        $page->drawLine(25,$this->y-$top-2, 25, $this->y-$top+10);
    //trait gauche
1108
1109                        $page->drawLine(570,$this->y-$top-2, 570, $this->y-$top+10);
    //trait droit
1110                        $top += $lineSpacing;
1111
1112                    }
1113
1114                    $maxHeight = $top > $maxHeight ? $top : $maxHeight;
1115                                    }
1116
1117                $this->y -= $maxHeight;
1118            }
1119        } $page->drawLine(25,$this->y-$top+8+$maxHeight, 570, $this->y-$top+8+
    $maxHeight);//trait bas
1120
1121
1122        return $page;
1123
1124    }
1125 }
1126
1127
```