



bluegate communications
Informationstechnologie & Consulting

bluegate communications
Inh. Ing. Michael Fürst
Hauptstr. 70a / Top 32
A - 2391 Kaltenleutgeben
E-Mail: office@bluegate.at
Tel.: +43 (0) 676 554 13 94

creditPass Bonitätsprüfung

für
xt:Commerce 3.0.4 SP2

bluegate communications
Inh. Ing. Michael Fürst
Hauptstr. 70a / Top 32
A - 2391 Kaltenleutgeben

Tel.: +43 (0) 676 554 13 94
Fax: +43 (0) 2238 20 703
E-Mail: office@bluegate.at
Web: <http://www.bluegate.at>

UID: ATU 55 89 91 02
Kammerzugehörigkeit:
Wirtschaftskammer NÖ
Handelsgericht Mödling

Bank Austria UniCredit Group
Kto. Nr.: 503 960 521 00 / BLZ 12000
IBAN: AT75 1200 0503 9605 2100
BIC: BKAUATWW



1 Systemvoraussetzungen

Für den Einsatz des creditPass Modules gelten folgende Voraussetzungen

- xt:Commerce Shop Version 3.0.4 SP2.1
- PHP5 (Unter PHP4 ist das Modul nicht lauffähig)
- PHP5 SimpleXML

2 Datensicherung

Bevor Sie diese Extension installieren, führen Sie bitte unbedingt eine Datensicherung (Backup) des bestehenden Shop Systems durch (Files und MySQL Datenbank). Für Datenverluste wird ausnahmslos keine Haftung übernommen.

3 Dateiübersicht

Für das creditPass Clearing mit xt:Commerce müssen einige Dateien geändert werden und einige neue Dateien kommen hinzu.

3.1 Modifizierte Dateien

- account_history_info.php
- checkout_confirmation.php
- checkout_payment.php
- checkout_process.php
- admin/customers.php
- admin/orders.php
- admin/includes/classes/order.php
- includes/classes/payment.php
- lang/german/admin/orders.php

3.2 Neue Dateien

- admin\creditpass.orders.php
- admin\includes\modules\export\creditpass.php
- inc\creditpass.inc.php
- inc\creditpass_checkout_confirmation.inc.php
- inc\creditpass_checkout_process.inc.php
- inc\creditpass_class_payment.inc.php
- includes\classes\bg_chunkexception.php
- includes\classes\bg_connectionexception.php
- includes\classes\bg_headerparseexception.php
- includes\classes\bg_noclearingexception.php
- includes\classes\bg_nocontentexception.php
- includes\classes\bg_novalidhttpheaderexception.php
- includes\classes\bg_settingserrorexception.php
- includes\classes\bg_wrongcustomertidexception.php
- includes\classes\bgfile.php
- includes\classes\bgheader.php
- includes\classes\bghttprequest.php
- includes\classes\bgrequest.php



- includes\classes\creditpass.php
- includes\classes\creditpass.user.inc.php
- includes\classes\creditpass.xml
- includes\modules\payment\creditpass\banktransfer.creditpass.inc.php
- lang\german\creditpass.php

4 Durchführung der Dateiänderungen

Bitte beachten Sie, dass Sie vor der Änderung an den Dateien eine Sicherheitskopie vornehmen. Die Zeilennummer in den folgenden Abbildungen beziehen sich auf ein nicht modifiziertes xt:Commerce 3.04 SP2. Änderungen sind jeweils rot umrandet.

4.1 account_history_info.php

Originaler Code:

```
67 // Order History
68 $history_block = '<table summary="order history">';
69 $statuses_query = xtc_db_query("select os.orders_status_name, osh.date_added, osh.comments from ".
    TABLE_ORDERS_STATUS." os, ".TABLE_ORDERS_STATUS_HISTORY." osh where osh.orders_id = '".(int) $_GET['order_id']'.
    "' and osh.orders_status_id = os.orders_status_id and os.language_id = '".(int) $_SESSION['languages_id']."'
    order by osh.date_added");
70 while ($statuses = xtc_db_fetch_array($statuses_query)) {
71     $history_block .= ' <tr>'. "\n". ' <td style="vertical-align:top;">';
```

```
$statuses_query = xtc_db_query("select os.orders_status_name, osh.date_added, osh.comments
from ".TABLE_ORDERS_STATUS." os, ".TABLE_ORDERS_STATUS_HISTORY." osh where osh.orders_id =
'".(int) $_GET['order_id']."' and osh.orders_status_id = os.orders_status_id and
os.language_id = '".(int) $_SESSION['languages_id']."' order by osh.date_added");
```

Geänderter Code:

```
67 // Order History
68 $history_block = '<table summary="order history">';
69 $statuses_query = xtc_db_query("select os.orders_status_name, osh.date_added, osh.comments from ".
    TABLE_ORDERS_STATUS." os, ".TABLE_ORDERS_STATUS_HISTORY." osh where osh.orders_id = '".(int) $_GET['order_id']'.
    "' and osh.orders_status_id = os.orders_status_id and os.language_id = '".(int) $_SESSION['languages_id']."'
    and osh.comments not like '%creditPass%' order by osh.date_added");
70 while ($statuses = xtc_db_fetch_array($statuses_query)) {
71     $history_block .= ' <tr>'. "\n". ' <td style="vertical-align:top;">';
```

```
$statuses_query = xtc_db_query("select os.orders_status_name, osh.date_added, osh.comments
from ".TABLE_ORDERS_STATUS." os, ".TABLE_ORDERS_STATUS_HISTORY." osh where osh.orders_id =
'".(int) $_GET['order_id']."' and osh.orders_status_id = os.orders_status_id and
os.language_id = '".(int) $_SESSION['languages_id']."' and osh.comments not like 'creditPass%'
order by osh.date_added");
```

4.2 checkout_confirmation.php

4.2.1 Änderung 1

Originaler Code:

```
125 $breadcrumb->add(NAVBAR_TITLE_1_CHECKOUT_CONFIRMATION, xtc_href_link(FILENAME_CHECKOUT_SHIPPING, '', 'SSL'));
126 $breadcrumb->add(NAVBAR_TITLE_2_CHECKOUT_CONFIRMATION);
127
128 require (DIR_WS_INCLUDES . 'header.php');
129 if (SHOW_IP_LOG == 'true') {
130     $smarty->assign('IP_LOG', 'true');
```

```
require (DIR_WS_INCLUDES . 'header.php');
```



Geänderter Code:

```
125 $breadcrumb->add(NAVBAR_TITLE_1_CHECKOUT_CONFIRMATION, xtc_href_link(FILENAME_CHECKOUT_SHIPPING, '', 'SSL'));
126 $breadcrumb->add(NAVBAR_TITLE_2_CHECKOUT_CONFIRMATION);
127
128 //require (DIR_WS_INCLUDES . 'header.php');
129 if (SHOW_IP_LOG == 'true') {
130     $smarty->assign('IP_LOG', 'true');
```

```
//require (DIR_WS_INCLUDES . 'header.php');
```

4.2.2 Änderung 2

Originaler Code:

```
205 $total_block = '<table>';
206 if (MODULE_ORDER_TOTAL_INSTALLED) {
207     $order_total_modules->process();
208     $total_block .= $order_total_modules->output();
209 }
210 $total_block .= '</table>';
211 $smarty->assign('TOTAL_BLOCK', $total_block);
212
213 if (is_array($payment_modules->modules)) {
214     if ($confirmation = $payment_modules->confirmation()) {
```

Geänderter Code:

```
205 $total_block = '<table>';
206 if (MODULE_ORDER_TOTAL_INSTALLED) {
207     $order_total_modules->process();
208     $total_block .= $order_total_modules->output();
209 }
210 $total_block .= '</table>';
211 $smarty->assign('TOTAL_BLOCK', $total_block);
212
213 /*
214  * BEGIN CREDITPASS
215  */
216 include (DIR_FS_INC . 'creditpass_checkout_confirmation.inc.php');
217 /*
218  * END CREDITPASS
219  */
220 require (DIR_WS_INCLUDES . 'header.php');
221
222 if (is_array($payment_modules->modules)) {
223     if ($confirmation = $payment_modules->confirmation()) {
```

```
/*
 * BEGIN CREDITPASS
 */
include (DIR_FS_INC . 'creditpass_checkout_confirmation.inc.php');
/*
 * END CREDITPASS
 */
require (DIR_WS_INCLUDES . 'header.php');
```



4.3 checkout_payment.php

Bitte beachten Sie bei dieser Änderung den Beistrich nach IMAGE_BUTTON_CONTINUE.

Originaler Code:

```
125 $smarty->assign('BUTTON_ADDRESS', '<a href="' . xtc_href_link(FILENAME_CHECKOUT_PAYMENT_ADDRESS, '', 'SSL') .  
    '>' . xtc_image_button('button_change_address.gif', IMAGE_BUTTON_CHANGE_ADDRESS) . '</a>');  
126 $smarty->assign('BUTTON_CONTINUE', xtc_image_submit('button_continue.gif', IMAGE_BUTTON_CONTINUE));  
127 $smarty->assign('FORM_END', '</form>');
```

```
$smarty->assign('BUTTON_CONTINUE', xtc_image_submit('button_continue.gif',  
IMAGE_BUTTON_CONTINUE));
```

Geänderter Code:

```
125 $smarty->assign('BUTTON_ADDRESS', '<a href="' . xtc_href_link(FILENAME_CHECKOUT_PAYMENT_ADDRESS, '', 'SSL') .  
    '>' . xtc_image_button('button_change_address.gif', IMAGE_BUTTON_CHANGE_ADDRESS) . '</a>');  
126 $smarty->assign('BUTTON_CONTINUE', xtc_image_submit('button_continue.gif', IMAGE_BUTTON_CONTINUE,  
    'id="button_submit"'));  
127 $smarty->assign('FORM_END', '</form>');
```

```
$smarty->assign('BUTTON_CONTINUE', xtc_image_submit('button_continue.gif',  
IMAGE_BUTTON_CONTINUE, 'id="button_submit"');
```

4.4 checkout_process.php

4.4.1 Änderung 1

Originaler Code:

```
324 xtc_db_query("update ".TABLE_ORDERS." set  
325 conversion_type = '2'  
326 where orders_id = '". $insert_id. "'");  
327 }  
328 }  
329 }  
330 }  
331 }  
332 // redirect to payment service  
333 if ($tmp)  
334 {  
335     $payment_modules->payment_action();  
336 }  
337 if (!$tmp) {  
338 }  
339 // NEW EMAIL configuration !  
340 $order_totals = $order_total_modules->apply_credit();  
341 include ('send_order.php');
```




Geänderter Code:

```
357 unset ($_SESSION['sendto']);
358 unset ($_SESSION['billto']);
359 unset ($_SESSION['shipping']);
360 unset ($_SESSION['payment']);
361 unset ($_SESSION['comments']);
362 unset ($_SESSION['last_order']);
363 unset ($_SESSION['tmp_oID']);
364 unset ($_SESSION['cc']);
365 unset ($_SESSION['creditpass']);
366 $last_order = $insert_id;
367 //GV Code Start
368 if (isset ($_SESSION['credit_covers']))
369     unset ($_SESSION['credit_covers']);
370 $order_total_modules->clear_posts(); //ICW ADDED FOR CREDIT CLASS SYSTEM
371 // GV Code End
```

```
unset ($_SESSION['creditpass']);
```

4.5 admin/customers.php

4.5.1 Änderung 1

Originaler Code:

```
56 $customers = xtc_db_fetch_array($customers_query);
57
58 $country_query = xtc_db_query("select countries_name from ".TABLE_COUNTRIES." where
countries_id = '". $customers['entry_country_id']."'");
59 $country = xtc_db_fetch_array($country_query);
```

```
$country_query = xtc_db_query("select countries_name from ".TABLE_COUNTRIES." where
countries_id = '". $customers['entry_country_id']."'");
```

Geänderter Code:

```
56 $customers = xtc_db_fetch_array($customers_query);
57
58 $country_query = xtc_db_query("select countries_name, countries_iso_code_2 from ".
TABLE_COUNTRIES." where countries_id = '". $customers['entry_country_id']."'");
59 $country = xtc_db_fetch_array($country_query);
60
61 $stat_query = xtc_db_query("select * from ".TABLE_CUSTOMERS_STATUS." where
customers_status_id = '". $customers[customers_status']."'");
62 $stat = xtc_db_fetch_array($stat_query);
```

```
$country_query = xtc_db_query("select countries_name, countries_iso_code_2 from
".TABLE_COUNTRIES." where countries_id = '". $customers['entry_country_id']."'");
```

4.5.2 Änderung 2

Originaler Code:

```
64 $sql_data_array = array ('customers_id' => xtc_db_prepare_input($customers['customers_id']), 'customers_c
65
66 $insert_sql_data = array ('currency_value' => '1.0000');
67 $sql_data_array = xtc_array_merge($sql_data_array, $insert_sql_data);
68 xtc_db_perform(TABLE_ORDERS, $sql_data_array);
```



Geänderter Code:

```
64 $sql_data_array = array ('customers_id' => xtc_db_prepare_input($customers['customers_id']), 'customers_c  
65 // BEGIN CREDITPASS  
66 $sql_data_array['customers_firstname'] = xtc_db_prepare_input($customers['entry_firstname']);  
67 $sql_data_array['customers_lastname'] = xtc_db_prepare_input($customers['entry_lastname']);  
68 $sql_data_array['delivery_firstname'] = xtc_db_prepare_input($customers['entry_firstname']);  
69 $sql_data_array['delivery_lastname'] = xtc_db_prepare_input($customers['entry_lastname']);  
70 $sql_data_array['delivery_country_iso_code_2'] = xtc_db_prepare_input($country['countries_iso_code_2']);  
71 $sql_data_array['billing_firstname'] = xtc_db_prepare_input($customers['entry_firstname']);  
72 $sql_data_array['billing_lastname'] = xtc_db_prepare_input($customers['entry_lastname']);  
73 $sql_data_array['billing_country_iso_code_2'] = xtc_db_prepare_input($country['countries_iso_code_2']);  
74 // END CREDITPASS  
75  
76 $insert_sql_data = array ('currency_value' => '1.0000');  
77 $sql_data_array = xtc_array_merge($sql_data_array, $insert_sql_data);  
78 xtc_db_perform(TABLE_ORDERS, $sql_data_array);  
79 $orders_id = xtc_db_insert_id();
```

```
// BEGIN CREDITPASS  
$sql_data_array['customers_firstname'] = xtc_db_prepare_input($customers['entry_firstname']);  
$sql_data_array['customers_lastname'] = xtc_db_prepare_input($customers['entry_lastname']);  
$sql_data_array['delivery_firstname'] = xtc_db_prepare_input($customers['entry_firstname']);  
$sql_data_array['delivery_lastname'] = xtc_db_prepare_input($customers['entry_lastname']);  
$sql_data_array['delivery_country_iso_code_2'] =  
xtc_db_prepare_input($country['countries_iso_code_2']);  
$sql_data_array['billing_firstname'] = xtc_db_prepare_input($customers['entry_firstname']);  
$sql_data_array['billing_lastname'] = xtc_db_prepare_input($customers['entry_lastname']);  
$sql_data_array['billing_country_iso_code_2'] =  
xtc_db_prepare_input($country['countries_iso_code_2']);  
// END CREDITPASS
```

4.6 admin/orders.php

4.6.1 Änderung 1

Originaler Code:

```
68 $orders_status_array[$orders_status['orders_status_id']] = $orders_status['orders_status_name'];  
69 }  
70 switch ($_GET['action']) {  
71     case 'update_order' :  
72         $oID = xtc_db_prepare_input($_GET['oID']);  
73         $status = xtc_db_prepare_input($_POST['status']);  
74         $comments = xtc_db_prepare_input($_POST['comments']);
```

Geänderter Code:

```
68 $orders_status_array[$orders_status['orders_status_id']] = $orders_status['orders_status_name'];  
69 }  
70 switch ($_GET['action']) {  
71     // BEGIN CPMANUALREQUEST  
72     case 'send_cprequest':  
73         $oID = $_GET['oID'];  
74         require DIR_FS_ADMIN.'creditpass.orders.php';  
75         xtc_redirect(xtc_href_link(FILENAME_ORDERS, 'oID='.$_GET['oID'].'&action=edit'));  
76  
77         break;  
78     // BEGIN CPMANUALREQUEST  
79     case 'update_order' :  
80         $oID = xtc_db_prepare_input($_GET['oID']);  
81         $status = xtc_db_prepare_input($_POST['status']);  
82         $comments = xtc_db_prepare_input($_POST['comments']);
```




```
// BEGIN CPMANUALREQUEST
case 'send_cprequest':
    $oID = $_GET['oID'];
    require DIR_FS_ADMIN.'creditpass.orders.php';
    xtc_redirect(xtc_href_link(FILENAME_ORDERS, 'oID='.$_GET['oID'].'&action=edit'));

    break;
// BEGIN CPMANUALREQUEST
```

4.6.2 Änderung 2

Originaler Code:

```
82     $customer_notified = '0';
83     if ($_POST['notify'] == 'on') {
84         $notify_comments = '';
85         if ($_POST['notify_comments'] == 'on') {
86             // $notify_comments = sprintf(EMAIL_TEXT_COMMENTS_UPDATE, $comments)."\n\n";
87             $notify_comments = $comments;
88         } else {
89             $notify_comments = '';
90         }

```

```
if ($_POST['notify'] == 'on') {
```

Geänderter Code:

```
90     $customer_notified = '0';
91     if ($_POST['notify'] == 'on' && striistr($comments,'creditPass') === false) {
92         $notify_comments = '';
93         if ($_POST['notify_comments'] == 'on' ) {
94             // $notify_comments = sprintf(EMAIL_TEXT_COMMENTS_UPDATE, $comments)."\n\n";
95             $notify_comments = $comments;
96         } else {
97             $notify_comments = '';
98         }

```

```
if ($_POST['notify'] == 'on' && striistr($comments,'creditPass') === false) {
```

4.6.3 Änderung 3

Originaler Code:

```
528     <tr>
529         <td class="main"><table border="1" cellspacing="0" cellpadding="5">
530             <tr>
531                 <td class="smallText" align="center"><b><?php echo TABLE_HEADING_DATE_ADDED; ?></b></td>
532                 <td class="smallText" align="center"><b><?php echo TABLE_HEADING_CUSTOMER_NOTIFIED; ?></b></td>
533             </tr>
534             <tr>
535                 <td class="smallText" align="center"><b><?php echo TABLE_HEADING_STATUS; ?></b></td>
536                 <td class="smallText" align="center"><b><?php echo TABLE_HEADING_COMMENTS; ?></b></td>
537             </tr>
538             <?php
539                 $orders_history_query = xtc_db_query("select orders_status_id, date_added, customer_notified,
540                 comments from ".TABLE_ORDERS_STATUS_HISTORY." where orders_id = '".xtc_db_input($oID)."' order by
541                 orders_status_id");
542                 if (xtc_db_num_rows($orders_history_query)) {
543                     while ($orders_history = xtc_db_fetch_array($orders_history_query)) {
```

```
$orders_history_query = xtc_db_query("select orders_status_id, date_added, customer_notified,
comments from ".TABLE_ORDERS_STATUS_HISTORY." where orders_id = '".xtc_db_input($oID)."' order
by orders_status_id");
```



Geänderter Code:

```
536 <tr>
537 <td class="main"><table border="1" cellspacing="0" cellpadding="5">
538 <tr>
539 <td class="smallText" align="center"><b><?php echo TABLE_HEADING_DATE_ADDED; ?></b></td>
540 <td class="smallText" align="center"><b><?php echo TABLE_HEADING_CUSTOMER_NOTIFIED; ?>
</b></td>
541 <td class="smallText" align="center"><b><?php echo TABLE_HEADING_STATUS; ?></b></td>
542 <td class="smallText" align="center"><b><?php echo TABLE_HEADING_COMMENTS; ?></b></td>
543 </tr>
544 <?php
545
546 $orders_history_query = xtc_db_query("select orders_status_id, date_added, customer_notified,
comments from ".TABLE_ORDERS_STATUS_HISTORY." where orders_id = '".xtc_db_input($oID)."' order by
orders_status_history_id");
547 if (xtc_db_num_rows($orders_history_query)) {
548 while ($orders_history = xtc_db_fetch_array($orders_history_query)) {
```

```
$orders_history_query = xtc_db_query("select orders_status_id, date_added, customer_notified,
comments from ".TABLE_ORDERS_STATUS_HISTORY." where orders_id = '".xtc_db_input($oID)."' order
by orders_status_history_id");
```

4.6.4 Änderung 4

Originaler Code:

```
590 <td colspan="2" align="right">
591 <?php
592 if (ACTIVATE_GIFT_SYSTEM == 'true') {
593 echo '<a class="button" href="'.xtc_href_link(FILENAME_GV_MAIL, xtc_get_all_get_params(array (
594 )
595 ?>
```

Geänderter Code:

```
598 <td colspan="2" align="right">
599 <!-- BEGIN CPMANUALREQUEST -->
600 <?php
601 if (MODULE_CREDITPASS_STATUS == 'True') {
602 echo xtc_draw_form(
603 'cprequest', FILENAME_ORDERS,
604 xtc_get_all_get_params(array('action')) . 'action=send_cprequest'
605 );
606 echo '<input type="submit" class="button" value="'.BUTTON_SEND_CPREQUEST.'">';
607 }
608 ?>
609 </form>
610 <!-- END CPMANUALREQUEST -->
611 <?php
612 if (ACTIVATE_GIFT_SYSTEM == 'true') {
613 echo '<a class="button" href="'.xtc_href_link(FILENAME_GV_MAIL, xtc_get_all_get_params(array (
614 )
615 ?>
```

```
<!-- BEGIN CPMANUALREQUEST -->
<?php
if (MODULE_CREDITPASS_STATUS == 'True') {
echo xtc_draw_form(
'cprequest', FILENAME_ORDERS,
xtc_get_all_get_params(array('action')) . 'action=send_cprequest'
);
echo '<input type="submit" class="button" value="'.BUTTON_SEND_CPREQUEST.'">';
}
?>
</form>
<!-- END CPMANUALREQUEST -->
```



4.7 admin/includes/classes/order.php

4.7.1 Änderung 1

Originaler Code:

```
44 function query($order_id) {  
45     $order_query = xtc_db_query("select customers_name,  
46                                 customers_cid,  
47                                 customers_id,  
48                                 customers_vat_id,  
49                                 customers_company,
```

Geänderter Code:

```
44 function query($order_id) {  
45     $order_query = xtc_db_query("select customers name,  
46                                 customers_firstname,  
47                                 customers_lastname,  
48                                 delivery_firstname,  
49                                 delivery_lastname,  
50                                 delivery_country_iso_code_2,  
51                                 billing_firstname,  
52                                 billing_lastname,  
53                                 billing_country_iso_code_2,  
54                                 customers_cid,  
55                                 customers_id,  
56                                 customers_vat_id,  
57                                 customers_company,
```

```
customers_firstname,  
customers_lastname,  
delivery_firstname,  
delivery_lastname,  
delivery_country_iso_code_2,  
billing_firstname,  
billing_lastname,  
billing_country_iso_code_2,
```

4.7.2 Änderung 2

Originaler Code:

```
128 $this->customer = array('name' => $order['customers_name'],  
129                        'company' => $order['customers_company'],  
130                        'csID' => $order['customers_cid'],  
131                        'vat_id' => $order['customers_vat_id'],
```

Geänderter Code:

```
136 $this->customer = array('name' => $order['customers name'],  
137                        'firstname' => $order['customers_firstname'],  
138                        'lastname' => $order['customers_lastname'],  
139                        'company' => $order['customers_company'],  
140                        'csID' => $order['customers_cid'],  
141                        'vat_id' => $order['customers_vat_id'],
```

```
'firstname' => $order['customers_firstname'],  
'lastname' => $order['customers_lastname'],
```



4.7.3 Änderung 3

Originaler Code:

```
145     $this->delivery = array('name' => $order['delivery_name'],
146                             'company' => $order['delivery_company'],
147                             'street_address' => $order['delivery_street_address'],
148                             'suburb' => $order['delivery_suburb'],
```

Geänderter Code:

```
155     $this->delivery = array('name' => $order['delivery_name'],
156                             'firstname' => $order['delivery_firstname'],
157                             'lastname' => $order['delivery_lastname'],
158                             'country_iso2' => $order['delivery_country_iso_code_2'],
159                             'company' => $order['delivery_company'],
160                             'street_address' => $order['delivery_street_address'],
161                             'suburb' => $order['delivery_suburb'],
```

```
'firstname' => $order['delivery_firstname'],
'lastname' => $order['delivery_lastname'],
'country_iso2' => $order['delivery_country_iso_code_2'],
```

4.7.4 Änderung 4

Originaler Code:

```
155     $this->billing = array('name' => $order['billing_name'],
156                             'company' => $order['billing_company'],
157                             'street_address' => $order['billing_street_address'],
158                             'suburb' => $order['billing_suburb'],
```

Geänderter Code:

```
168     $this->billing = array('name' => $order['billing_name'],
169                             'firstname' => $order['billing_firstname'],
170                             'lastname' => $order['billing_lastname'],
171                             'country_iso2' => $order['billing_country_iso_code_2'],
172                             'company' => $order['billing_company'],
173                             'street_address' => $order['billing_street_address'],
174                             'suburb' => $order['billing_suburb'],
```

```
'firstname' => $order['billing_firstname'],
'lastname' => $order['billing_lastname'],
'country_iso2' => $order['billing_country_iso_code_2'],
```

4.8 includes/classes/payment.php

4.8.1 Änderung 1

Originaler Code:

Hier ist zu beachten, dass der rot umrandete Teil aus dem originalen Programm ersetzt werden muss.

```
46     if ( (xtc_not_null($module)) && (in_array($module, '.', substr($PHP_SELF, (strpos($PHP_SELF, '.')+1)), $this->modules
47     )) ) {
48         $this->selected_module = $module;
49
50         $include_modules[] = array('class' => $module, 'file' => $module . '.php');
51     } else {
52         reset($this->modules);
53         while (list($value) = each($this->modules)) {
54             $class = substr($value, 0, strpos($value, '.'));
55             $include_modules[] = array('class' => $class, 'file' => $value);
56         }
57     }
58     // load unallowed modules into array
59     $unallowed_modules = explode(',', $_SESSION['customers_status']['customers_status_payment_unallowed'], $order->customer[
60     'payment_unallowed']);
```



```
while (list(, $value) = each($this->modules)) {  
    $class = substr($value, 0, strrpos($value, '.'));  
    $include_modules[] = array('class' => $class, 'file' => $value);  
}
```

Geänderter Code:

```
46 if ( (xrc_not_null($module) && (in_array($module . '.' . substr($PHP_SELF, (strrpos($PHP_SELF, '.')+1), $this->modules  
)) ) {  
47     $this->selected_module = $module;  
48  
49     $include_modules[] = array('class' => $module, 'file' => $module . '.php');  
50 } else {  
51     reset($this->modules);  
52  
53     /*  
54     * BEGIN CREDITPASS  
55     */  
56     include (DIR_FS_INC . 'creditpass_class_payment.inc.php');  
57  
58     /*  
59     * END CREDITPASS  
60     */  
61  
62 }  
63 // load unallowed modules into array  
64 $unallowed_modules = explode(',', $SESSION['customers_status']['customers_status_payment_unallowed'],'.').$order->customer[  
'payment_unallowed'];
```

```
/*  
 * BEGIN CREDITPASS  
 */  
include (DIR_FS_INC . 'creditpass_class_payment.inc.php');  
  
/*  
 * END CREDITPASS  
 */
```

4.8.2 Änderung 2

Diese Änderung ist optional und deaktiviert nach dem Übermitteln der Zahlungsmethode die „Weiter“ Schaltfläche auf der Seite checkout_payment.php.

Originaler Code:

```
152 $js .= "\n" . ' if (payment_value == null) { ' . "\n" .  
153     '     error_message = error_message + unescape("' . xtc_js_lang(JS_ERROR_NO_PAYMENT_MODULE_SELECTED  
154     '     error = 1;' . "\n" .  
155     ' }' . "\n\n" .  
156     ' if (error == 1 && submitter != 1) { ' . "\n" . // GV Code Start/End  
157     '     alert(error_message); ' . "\n" .  
158     '     return false;' . "\n" .  
159     ' } else { ' . "\n" .  
160     '     return true;' . "\n" .  
161     ' }' . "\n" .  
162     ' }' . "\n" .  
163     ' //--></script>' . "\n";  
164 }
```



Geänderter Code:

```
158     $js .= "\n" . ' if (payment_value == null) (' . "\n" .  
159         '     error_message = error_message + unescape("' . xtc_js_lang(JS_ERROR_NO_PAYMENT_MODULE_SELECTED  
160         '     error = 1;' . "\n" .  
161         '     )' . "\n\n" .  
162         '     if (error == 1 && submitter != 1) (' . "\n" . // GV Code Start/End  
163         '     alert(error_message);' . "\n" .  
164         '     return false;' . "\n" .  
165         '     ) else (' . "\n" .  
166         '     document.getElementById("button_submit").disabled=true;' . "\n" .  
167         '     return true;' . "\n" .  
168         '     )' . "\n" .  
169         ' )' . "\n" .  
170         ' //--></script>' . "\n";  
171     }
```

```
'     document.getElementById("button_submit").disabled=true;' . "\n" .
```

4.9 lang/german/admin/orders.php

Originaler Code:

```
114     define('TABLE_HEADING_DISCOUNT', 'Rabatt');  
115     define('ENTRY_CUSTOMERS_GROUP', 'Kundengruppe:');  
116     define('TEXT_VALIDATING', 'Nicht best&auml;tigt');  
117     ?>
```

Geänderter Code:

```
114     define('TABLE_HEADING_DISCOUNT', 'Rabatt');  
115     define('ENTRY_CUSTOMERS_GROUP', 'Kundengruppe:');  
116     define('TEXT_VALIDATING', 'Nicht best&auml;tigt');  
117  
118     // BEGIN CPMANUALREQUEST  
119     define('BUTTON_SEND_CPREQUEST', 'creditPass Abfrage');  
120     // END CPMANUALREQUEST  
121  
122  
123     ?>
```

```
// BEGIN CPMANUALREQUEST  
define('BUTTON_SEND_CPREQUEST', 'creditPass Abfrage');  
// END CPMANUALREQUEST
```



5 Sprachdatei

In der Datei `/lang/german/creditpass.php` erfolgt die Anpassung von Sprachbausteinen.

6 Installation

Für die Installation des Modules wechseln Sie in die Administrationsansicht und wählen aus dem linksseitigen Menü den Punkt „XT-Module“, in der folgenden Abbildung rot eingerahmt. Danach selektieren Sie in der Spalte Module mit einem Klick auf „creditPass Bonitätsprüfung“ das Bonitätsmodul.

The screenshot shows the xt:Commerce administration interface. The top navigation bar includes 'News', 'Support Site', 'Shop', 'Logout', and 'Credits'. The left sidebar menu has 'Kunden', 'Artikelkatalog', and 'Module' sections. The 'Module' section is expanded, and 'XT-Module' is highlighted with a red box. The main content area shows a table of XT-Modules. The 'creditPass Bonitätsprüfung' module is highlighted in yellow. A pop-up window for this module shows an 'Installieren' button and the text 'Bonitätsprüfung mit creditPass'.

Über den Button „Installieren“ erfolgt der letzte Schritt der Integration in xt:Commerce.

7 Konfiguration

Die Konfiguration des Bonitätsprüfungsmodules erfolgt an gleicher Stelle wie die Konfiguration. Statt „Installieren“ klicken Sie bitte hier auf „Start“.

7.1 Modulstatus

Mit True ist das Modul aktiv, bei False wird keine Bonitätsprüfung durchgeführt

7.2 Testmodus

Um das Modul und dessen Einstellungen zu überprüfen, kann der Testmodus verwendet werden. Es wird hierbei wie im normalen Betrieb verfahren, die Rückmeldung von creditPass ist jedoch nicht von den Personendaten abhängig, sondern wird von der Bestellsumme, inklusive MwSt. und Versandkosten, bestimmt. Es ist die letzte Stelle des Rechnungsbetrages entscheidend, bei der Eurowährung also der Cent Betrag.



Folgende Fälle können im Testmodus simuliert werden:

- 0, 3, 4, 5, 6, 7, 9 Die Transaktion wird autorisiert ⇒
z.B. 12,50 der Bestellvorgang kann normal abgeschlossen werden.
- 1 Die Transaktion wird nicht autorisiert ⇒
z.B. 12,51 Anzeige der Seite der Zahlungsvarianten. Die unter 7.6
„Fehlermeldung“ angegebene Fehlermeldung wird angezeigt
und nur die als sicher geltenden Zahlungsvarianten (Fallback)
stehen zur Auswahl.
- 2 Die Transaktion muss manuell autorisiert werden ⇒
z.B. 12,52 Es gelten die unter 7.9 „Manuelle Überprüfung von Anfragen
abweisen“ getätigten Einstellungen.
- 8 Generierung eines Systemfehlers ⇒
z.B. 12,58 Es gelten die unter 7.10 „Fehlerhafte Abfragen als unautorisiert
einstufen“ getätigten Einstellungen.

7.3 creditPass Server URL

Internetadresse unter dem das creditPass Service erreichbar ist.

7.4 Autorisierungs ID

Die mit Ihrem Vertrag mitgelieferte Benutzerkennung.

7.5 Autorisierungskennwort

Das mit Ihrem Vertrag mitgelieferte Autorisierungskennwort.

7.6 Fehlermeldung

Wird eine Zahlung abgewiesen, so wird der Käufer auf die Seite mit den Zahlungsmodalitäten geleitet und erhält den hier festgelegten Text als Hinweis, dass die Zahlart nicht akzeptiert wurde.

7.7 Server Timeout in Sekunden

Kann eine Verbindung mit dem creditPass Server hergestellt werden, aber es erfolgt keine Rückantwort, so wird die Verbindung nach der hier eingestellten Zeitspanne unterbrochen.

7.8 Admin e-Mail Adresse

An diese Mailadresse ergehen alle Systemmeldungen des creditPass Modules. Unter anderem, wenn technische Probleme vorliegen oder eine manuelle Freigabe der Zahlung notwendig ist.

7.9 Manuelle Überprüfung von Anfragen abweisen

Muss ein Auftrag manuell freigegeben werden, wird hier eingestellt ob die Zahlung abgelehnt wird (True) oder zugelassen wird (False).



7.10 Fehlerhafte Abfragen als unautorisiert einstufen

Wenn Anfragen aus technischen Gründen scheitern oder nicht durchgeführt werden können, etwa bei Verbindungsstörungen zum creditPass Server oder Nicht-erreichbarkeit des Servers, wird hier festgelegt ob die fehlgeschlagene Abfrage als negativ verlaufen (True) eingestuft wird, oder als autorisiert behandelt wird (False).

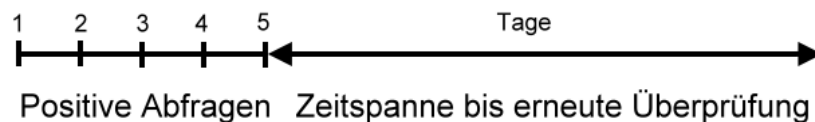
7.11 Stammkundengruppen

Zur Reduktion von Abfragen, und daher der Kosten, können ein oder mehrere Kundengruppen für die Bonitätsprüfung als Stammkunden definiert werden. Die Einstellungen können für jede Kundengruppe getrennt gehandhabt werden.

Für die Definition sind folgende Parameter notwendig

- Nummer der Kundengruppe
- Anzahl positiver Abfragen in ununterbrochener Reihenfolge zur Erlangung des Stammkundenstatus
- Zeitspanne nach der Stammkunden nach der letzten positiven Überprüfung nochmals überprüft werden
- Ein oder mehrere Auftragsstatus, welche für die Erkennung von erfolgten positiven Aufträgen herangezogen werden.

In der folgenden Grafik ist die Arbeitsweise grafisch dargestellt. Es wird zuerst überprüft ob die eingestellte Anzahl positiver Abwicklung von Aufträgen erfolgt ist. Ist dies nicht der Fall, wird eine Abfrage durchgeführt. Ist das Kriterium erfüllt, dann wird getestet ob die eingestellte Zeitspanne seit der letzten Überprüfung überschritten wurde und in diesem Fall dann ebenso eine Abfrage gemacht.



Als Sonderfall ist der Wert 0 (Null) zu sehen. In der ersten Testphase bedeutet dies, dass keine positiven Abfragen zur Erlangung des Stammkundenstatus benötigt werden. Bei der Zeitspanne hat eine Null die Folge, dass der Kunde nie mehr erneut überprüft wird.

Der Eintrag in der Stammkundendefinition ist nun folgendermaßen

`<kundengruppe>:<anzahl_ok>:<tage_abstand>:<auftragsstatus>,<auftragsstatus>`



Die Werte für die Kundengruppe, beziehungsweise den Auftragstatus, entnehmen Sie bitte der Aufzählung in Ihren Einstellungen, welche wie nebenstehend abgebildet aussehen können.

<i>Kundengruppen:</i>
0 - Admin
1 - Gast
2 - Neuer Kunde
3 - Händler
4 - Stammkunde
5 - Gruppe X
<i>Auftragsstatus</i>
1 - Offen
2 - In Bearbeitung
3 - Versendet
4 - manuell freigeben

Zur Veranschaulichung der Einstellungen einige Beispiele:

- 3:5:180:3,4 Kundengruppe 3 (Händler) ist nach 5 aufeinanderfolgenden positiven Aufträgen Stammkunde. Wird die Zeitspanne von 180 Tagen seit der letzten Überprüfung überschritten, wird erneut abgefragt. Für die positive Bewertung eines Auftrages zählen nur versendete oder manuell freigegebene Aufträge
- 4:0:90:3,4 Kundengruppe 4 (Stammkunden) werden nur alle 90 Tage überprüft. Für die positive Bewertung eines Auftrages zählen nur versendete oder manuell freigegebene Aufträge
- 4:5:0:3,4 Sonderfall: Kundengruppe 4 (Stammkunden) wird nur 5 mal getestet und dann nie wieder
- 4:0:0:3 Sonderfall: Kundengruppe 4 (Stammkunden) wird zu keinem Zeitpunkt überprüft

Werden mehrere Definition benötigt, so werden diese mit Strichpunkt voneinander getrennt.

7.12 Zahlungsarten

Für jede im System definierte Zahlungsart gibt es einen Eintrag ob für diese eine creditPass Abfrage durchgeführt werden soll und ob diese als Fallback Zahlungsvariante gehandhabt werden soll.

Fallback Zahlungsvarianten sind jene Zahlungsmethoden, welche einem potentiellen Käufer nach einer erfolgten negativen Abfrage angezeigt werden.

7.13 Debug Modus

Im Debug Modus werden im Administrationsbereich bei den Bestellungen im Kommentarfeld mehr Details der Abfrage und der Rückantwort ausgegeben.



8 Abfragedokumentation

Artikel	Artikel-Nr.	Preis (exkl.)	MwSt.	Preis (inkl.)	
1 x SimArt 1000	1	68.00 EUR	0%	68.00 EUR	68.00
					Zwischensumme: 68,00
					Selbstabholung (Selbstabholung der Ware in unserer Geschäftsstelle.): 0,00
					Summe: 68,00

hinzugefügt am:	Kunde benachrichtigt	Status	Kommentar
25.03.2009 08:56:38	✓	Offen	
25.03.2009 08:58:19	✗	Offen	creditPass Clearing OK Transaktions-ID: cpID-20090325085635 Transaktions-ID Empfang: 72e5f7076b24461a9a044ba0dcd30851 Antwortcode: 0 Antworttext: autorisiert

Zu jeder Bestellung wird im Kommentarfeld das Ergebnis einer Abfrage und auch ein mögliches negatives Ergebnis angezeigt.

Jeder Kommentar, welcher das Wort „creditPass“ enthält, wird bei der Anzeige in der Kundenhistorie ausgeblendet. Des Weiteren werden vom Administrator erstellte Kommentare die das Wort „creditPass“ enthalten, nicht an den Kunden gesendet. Unabhängig ob die Checkbox zu „Kunde benachrichtigen“ aktiviert ist oder nicht. Die Groß- und Kleinschreibung ist dabei nicht relevant.

9 Manuelle Abfrage

Bei allen Bestellungen kann der Administrator nachträglich eine Bonitätsprüfung durchführen. Für diesen Zweck gibt es bei den Bestellungen eine zusätzliche Schaltfläche, in der folgenden Abbildung rot eingrahmt, zur Abfrage. Diese Abfrage wird immer durchgeführt und ist nicht abhängig von Stammkundeneinstellungen.

Artikel	Artikel-Nr.	Preis (exkl.)	MwSt.	Preis (inkl.)	Total
1 x SimArt 1000	1	68.00 EUR	0%	68.00 EUR	68.00 EUR
					Zwischensumme: 68,00 EUR
					Summe: 68,00 EUR

hinzugefügt am:	Kunde benachrichtigt	Status	Kommentar
Keine Bestellhistorie verfügbar			

Kommentar

Status:

Kunde benachrichtigen: Kommentare mitsenden:

Die Schaltfläche wird allerdings nur dann angezeigt, wenn das creditPass Plugin aktiviert ist. Das Ergebnis der Bonitätsprüfung wird genauso wie bei einem normalen Bestellvorgang angezeigt.



bluegate communications
Informationstechnologie & Consulting

bluegate communications
Inh. Ing. Michael Fürst
Hauptstr. 70a / Top 32
A - 2391 Kaltenleutgeben
E-Mail: office@bluegate.at
Tel.: +43 (0) 676 554 13 94

10 Dokumentenversion

Stand 18. April 2011