

# Installation von 12check für XT:Commerce

Bitte führen Sie die Installation in der hier beschriebene Reihenfolge aus!

Bitte beachten Sie: in der zip-Datei befinden sich zwei Ordner:

"xtcommerce\_install" beinhaltet alle Dateien zum Kopieren auf Ihren Server.

"xtcommerce\_muster" beinhaltet alle Dateien inkl. der fertig bearbeiteten Dateien als Muster oder einsatzbereit für einen unveränderten(!) XT:Commerce 3.04 SP 2.1 - Shop!

## 1. Voraussetzungen

12check benötigt für die Kommunikation mit dem Telefon-Server die PHP-Module „curl“ und „soap“. Diese beiden Module müssen auf Ihrem Server mit PHP installiert sein.

Dazu wird die PHP-Library „nusoap“ benötigt, die mit dieser Installation mitgeliefert wird und die von XT:Commerce gelieferte Version überschreiben soll.

Alle weiteren Bedingungen entsprechen den Laufbedingungen von XT:Commerce. Wir empfehlen Ihnen dringend die jeweils aktuellen PHP- und MySQL-Versionen zu verwenden, derzeit sind das PHP Version 5.2.8 und MySQL 5.1. 32.

## 2. Dateien kopieren

Kopieren aller Dateien entsprechend der Struktur, also mit den Verzeichnissen, aus dem Ordner "xtcommerce\_install" in Ihr XT:Commerce-Verzeichnis auf dem Server.

Die Dateien der Bibliothek "nusoap" in "includes/nusoap/lib" sollen dabei überschrieben werden.

Die mit 12check gelieferte Version aktualisiert die mit XT:Commerce ausgelieferte Version.

Das Template für die 12check-Authentifizierung wird dabei in das Verzeichnis "templates/xtc4" kopiert. Sollten Sie ein anderes Template einsetzen, passen Sie das Template dafür bitte entsprechend an.

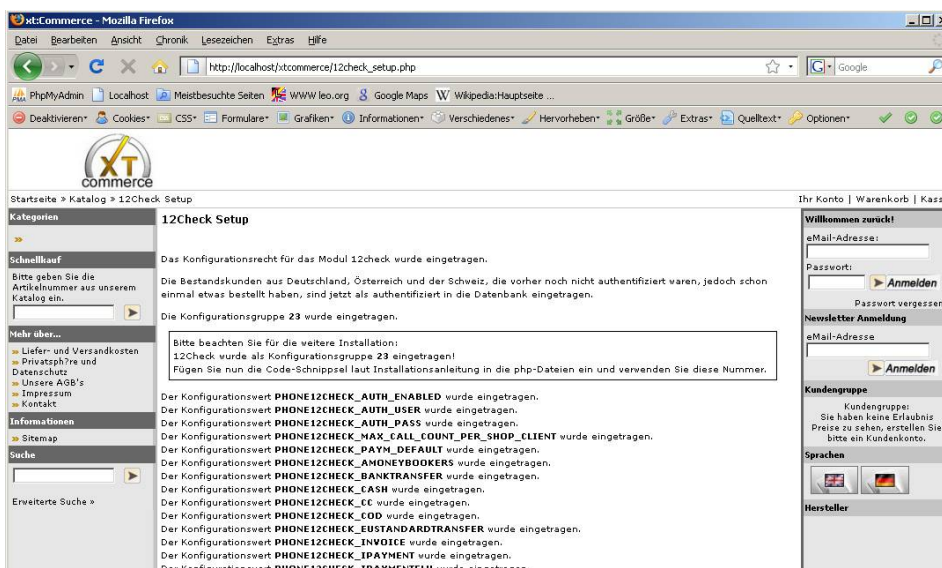
## 3. Setup

Ausführen von "12check\_setup.php".

Dieses Script legt die notwendigen Datenbankeinträge für 12check an.

Bitte merken Sie sich für die nächsten Schritte die Modulnummer für das Modul 12check.

In einem unveränderten XT:Commerce 3.04 SP 2.1 - Shop wird das die Nummer "24" sein.



Nach der einmaligen Ausführung des Scripts kann es gelöscht werden.

## 4. Bearbeiten von Sprachdateien

XT:Commerce ist auch im Admin-Bereich konsequent mehrsprachig. Dafür liegen alle Systemmeldungen und Sprachausgaben in Sprachdateien im Ordern "lang".

An einigen Stellen muss die im Setup-Programm ermittelt Modulnummer eingesetzt werden. Ersetzen Sie im Code bitte "<Modulnummer>" durch die genannte Modulnummer. (ohne Anführungszeichen)

### 4.1

in "lang\german\admin\german.php" und "lang\english\admin\english.php" einfügen:

Einzufügender Code:

```
define('BOX_CONFIGURATION_<Modulnummer>', '12check');
```

Beispiel:

```
define('BOX_CONFIGURATION_24', '12check');
```

### 4.2

in "lang\german\admin\configuration.php" und "lang\english\admin\configuration.php" einfügen: (die englische Version ist noch nicht übersetzt)

Einzufügender Code:

```
// 12check
define('PHONE12CHECK_AUTH_ENABLED_TITLE', '12Check aktiviert');
define('PHONE12CHECK_AUTH_ENABLED_DESC', 'Soll 12Check authentifizieren?');
define('PHONE12CHECK_AUTH_PARTNER_ID_TITLE', '12Check Partner-ID');
define('PHONE12CHECK_AUTH_PARTNER_ID_DESC', '12Check Partner-ID');
define('PHONE12CHECK_AUTH_USER_TITLE', '12Check Benutzername');
define('PHONE12CHECK_AUTH_USER_DESC', 'Der Benutzername bei 12Check. Dies ist der Loginname, der auch auf der Seite http://www.12check.de zur Anmeldung benötigt wird. ');
define('PHONE12CHECK_AUTH_PASS_TITLE', '12Check Kennwort');
define('PHONE12CHECK_AUTH_PASS_DESC', 'Das Passwort, dass auch auf der Seite http://www.12check.de zum Login benötigt wird. ');
define('PHONE12CHECK_AUTH_FORCE_TITLE', '12Check bei Admins IMMER durchführen');
define('PHONE12CHECK_AUTH_FORCE_DESC', 'Soll 12Check bei Administratoren für jeden Einkauf ausgeführt werden? (für Testläufe, normalerweise ausgeschaltet!)');

define('PHONE12CHECK_MAX_CALL_COUNT_PER_SHOP_CLIENT_TITLE', 'Max. Anzahl Anrufe');
define('PHONE12CHECK_MAX_CALL_COUNT_PER_SHOP_CLIENT_DESC', 'Max. Anzahl Anrufe eines neuen Kunden. ');
define('PHONE12CHECK_PAYM_DEFAULT_TITLE', 'Vorgabe für neue Bezahlmodule');
define('PHONE12CHECK_PAYM_DEFAULT_DESC', 'Ab welchem Warenwert soll eine Telefonauthentifizierung vorgenommen werden?');

define('PHONE12CHECK_AMONEYBOOKERS_TITLE', 'Moneybookers (Modul amoneybookers)');
define('PHONE12CHECK_AMONEYBOOKERS_DESC', 'Ab welchem Warenwert soll eine Telefonauthentifizierung vorgenommen werden?');
define('PHONE12CHECK_BANKTRANSFER_TITLE', 'Lastschriftverfahren (Modul banktransfer)');
define('PHONE12CHECK_BANKTRANSFER_DESC', 'Ab welchem Warenwert soll eine Telefonauthentifizierung vorgenommen werden?');
define('PHONE12CHECK_CASH_TITLE', 'Barzahlung (Modul cash)');
define('PHONE12CHECK_CASH_DESC', 'Ab welchem Warenwert soll eine Telefonauthentifizierung vorgenommen werden?');
define('PHONE12CHECK_CC_TITLE', 'Kreditkarte (Modul cc)');
define('PHONE12CHECK_CC_DESC', 'Ab welchem Warenwert soll eine Telefonauthentifizierung vorgenommen werden?');
define('PHONE12CHECK_COD_TITLE', 'Nachnahme (Modul cod)');
define('PHONE12CHECK_COD_DESC', 'Ab welchem Warenwert soll eine Telefonauthentifizierung vorgenommen werden?');
define('PHONE12CHECK_EUSTANDARDTRANSFER_TITLE', 'Nachnahme (Modul eustandardtransfer)');
define('PHONE12CHECK_EUSTANDARDTRANSFER_DESC', 'Ab welchem Warenwert soll eine Telefonauthentifizierung vorgenommen werden?');
define('PHONE12CHECK_INVOICE_TITLE', 'Rechnung (Modul invoice)');
define('PHONE12CHECK_INVOICE_DESC', 'Ab welchem Warenwert soll eine Telefonauthentifizierung vorgenommen werden?');
define('PHONE12CHECK_IPAYMENT_TITLE', 'iPayment (Modul ipayment)');
define('PHONE12CHECK_IPAYMENT_DESC', 'Ab welchem Warenwert soll eine Telefonauthentifizierung vorgenommen werden?');
define('PHONE12CHECK_IPAYMENTELV_TITLE', 'Lastschriftverfahren (Modul ipaymentelv)');
define('PHONE12CHECK_IPAYMENTELV_DESC', 'Ab welchem Warenwert soll eine Telefonauthentifizierung vorgenommen werden?');
```

```

define('PHONE12CHECK_LUUPWS_TITLE', 'LUUPAY (Modul luupws)');
define('PHONE12CHECK_LUUPWS_DESC', 'Ab welchem Warenwert soll eine Telefonauthentifizierung
vorgenommen werden?');
define('PHONE12CHECK_MONEYORDER_TITLE', 'Scheck/Vorkasse (Modul moneyorder)');
define('PHONE12CHECK_MONEYORDER_DESC', 'Ab welchem Warenwert soll eine Telefonauthentifizierung
vorgenommen werden?');
define('PHONE12CHECK_PAYPAL_TITLE', 'PayPal (Modul paypal)');
define('PHONE12CHECK_PAYPAL_DESC', 'Ab welchem Warenwert soll eine Telefonauthentifizierung
vorgenommen werden?');
define('PHONE12CHECK_UOS_GIROPAY_MODUL_TITLE', 'giropay® (Modul uos_giropay_modul)');
define('PHONE12CHECK_UOS_GIROPAY_MODUL_DESC', 'Ab welchem Warenwert soll eine
Telefonauthentifizierung vorgenommen werden?');
define('PHONE12CHECK_UOS_GP_MODUL_TITLE', 'global-paycard® (Modul uos_gp_modul)');
define('PHONE12CHECK_UOS_GP_MODUL_DESC', 'Ab welchem Warenwert soll eine Telefonauthentifizierung
vorgenommen werden?');
define('PHONE12CHECK_UOS_KREDITKARTE_MODUL_TITLE', 'Kreditkarte International (Modul
uos_kreditkarte_modul)');
define('PHONE12CHECK_UOS_KREDITKARTE_MODUL_DESC', 'Ab welchem Warenwert soll eine
Telefonauthentifizierung vorgenommen werden?');
define('PHONE12CHECK_UOS_LASTSCHRIFT_AT_MODUL_TITLE', 'Lastschrift Österreich (Modul
uos_lastschrift_at_modul)');
define('PHONE12CHECK_UOS_LASTSCHRIFT_AT_MODUL_DESC', 'Ab welchem Warenwert soll eine
Telefonauthentifizierung vorgenommen werden?');
define('PHONE12CHECK_UOS_LASTSCHRIFT_DE_MODUL_TITLE', 'Lastschrift Deutschland (Modul
uos_lastschrift_de_modul)');
define('PHONE12CHECK_UOS_LASTSCHRIFT_DE_MODUL_DESC', 'Ab welchem Warenwert soll eine
Telefonauthentifizierung vorgenommen werden?');
define('PHONE12CHECK_UOS_VORKASSE_MODUL_TITLE', 'Vorkasse (Modul uos_vorkasse_modul)');
define('PHONE12CHECK_UOS_VORKASSE_MODUL_DESC', 'Ab welchem Warenwert soll eine
Telefonauthentifizierung vorgenommen werden?');
define('PHONE12CHECK_WORLDPAY_TITLE', 'Secure Credit Card Payment (Modul worldpay)');
define('PHONE12CHECK_WORLDPAY_DESC', 'Ab welchem Warenwert soll eine Telefonauthentifizierung
vorgenommen werden?');

```

## 5. Bearbeiten von Programmdateien

### 5.1 Eintrag in "admin/includes/column\_left.php":

Tragen Sie den folgenden Code am Ende oder an der Position Ihrer Wahl ein.

Code:

```

if (($SESSION['customers_status']['customers_status_id'] == '0') && ($admin_access['configuration']
== '1')) echo '<a href="' . xtc_href_link(FILENAME_CONFIGURATION, 'gID=<Modulnummer>', 'NONSSL') .
'" class="menuBoxContentLink"> -' . BOX_CONFIGURATION_<Modulnummer>' . '</a><br>';

```

Beispiel:

```

if (($SESSION['customers_status']['customers_status_id'] == '0') && ($admin_access['configuration']
== '1')) echo '<a href="' . xtc_href_link(FILENAME_CONFIGURATION, 'gID=24', 'NONSSL') . '"
class="menuBoxContentLink"> -' . BOX_CONFIGURATION_24 . '</a><br>';

```

### 5.2 Eintrag in "includes/filenames.php":

Tragen Sie den folgenden Code am Ende oder an der Position Ihrer Wahl ein.

Code:

```

define('FILENAME_CHECKOUT_AUTHENTICATE_12CHECK', 'checkout_auth_12check.php');

```

### 5.3 Veränderung von "checkout\_payment.php"

Code Zeile 38 - 55 Original:

```

// include needed functions
require_once (DIR_FS_INC . 'xtc_address_label.inc.php');
require_once (DIR_FS_INC . 'xtc_get_address_format_id.inc.php');
require_once (DIR_FS_INC . 'xtc_check_stock.inc.php');
unset ($SESSION['tmp_oID']);
// if the customer is not logged on, redirect them to the login page
if (!isset ($SESSION['customer_id'])) {
    if (ACCOUNT_OPTIONS == 'guest') {
        xtc_redirect(xtc_href_link(FILENAME_CREATE_GUEST_ACCOUNT, '', 'SSL'));
    } else {
        xtc_redirect(xtc_href_link(FILENAME_LOGIN, '', 'SSL'));
    }
}

```

```

}

// if there is nothing in the customers cart, redirect them to the shopping cart page
if ($_SESSION['cart']->count_contents() < 1)
    xtc_redirect(xtc_href_link(FILENAME_SHOPPING_CART));

```

#### Code bearbeitet mit zwischengefügtem Abschnitt:

```

// include needed functions
require_once (DIR_FS_INC . 'xtc_address_label.inc.php');
require_once (DIR_FS_INC . 'xtc_get_address_format_id.inc.php');
require_once (DIR_FS_INC . 'xtc_check_stock.inc.php');
unset ($_SESSION['tmp_oID']);

// 12check -----
// AGB-Abfrage in Session
unset ($_SESSION['conditions']);
// Initalisierung
$_SESSION['PHONE12CHECK_USE'] = 1;
$_SESSION['PHONE12CHECK_AUTHENTICATED'] = 0;
// Übernahme der POST-Variablen bei Rücksprung
if ((isset ($_SESSION['PHONE12CHECK_POST'])) && (is_array ($_SESSION['PHONE12CHECK_POST']))) {
    foreach ($_SESSION['PHONE12CHECK_POST'] as $s12CheckReqKey => $s12CheckReqVal) {
        $_POST[$s12CheckReqKey] = $s12CheckReqVal;
    }
}
// 12check Ende -----

// if the customer is not logged on, redirect them to the login page
if (!isset ($_SESSION['customer_id'])) {
    if (ACCOUNT_OPTIONS == 'guest') {
        xtc_redirect(xtc_href_link(FILENAME_CREATE_GUEST_ACCOUNT, '', 'SSL'));
    } else {
        xtc_redirect(xtc_href_link(FILENAME_LOGIN, '', 'SSL'));
    }
}

// if there is nothing in the customers cart, redirect them to the shopping cart page
if ($_SESSION['cart']->count_contents() < 1)
    xtc_redirect(xtc_href_link(FILENAME_SHOPPING_CART));

```

#### Code Zeile 123- 130 Original:

```

$breadcrumb->add (NAVBAR_TITLE_1_CHECKOUT_PAYMENT, xtc_href_link(FILENAME_CHECKOUT_SHIPPING, '',
'SSL'));
$breadcrumb->add (NAVBAR_TITLE_2_CHECKOUT_PAYMENT, xtc_href_link(FILENAME_CHECKOUT_PAYMENT, '',
'SSL'));

$smarty->assign('FORM_ACTION', xtc_draw_form('checkout_payment',
xtc_href_link(FILENAME_CHECKOUT_CONFIRMATION, '', 'SSL'), 'post', 'onSubmit="return
check_form();"'));
$smarty->assign('ADDRESS_LABEL', xtc_address_label($_SESSION['customer_id'], $_SESSION['billto'],
true, ' ', '<br />'));
$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>');
$smarty->assign('BUTTON_CONTINUE', xtc_image_submit('button_continue.gif', IMAGE_BUTTON_CONTINUE));
$smarty->assign('FORM_END', '</form>');

```

#### Code bearbeitet mit Einfügung:

```

$breadcrumb->add (NAVBAR_TITLE_1_CHECKOUT_PAYMENT, xtc_href_link(FILENAME_CHECKOUT_SHIPPING, '',
'SSL'));
$breadcrumb->add (NAVBAR_TITLE_2_CHECKOUT_PAYMENT, xtc_href_link(FILENAME_CHECKOUT_PAYMENT, '',
'SSL'));

// 12check Bearbeitung -----
$smarty->assign('FORM_ACTION', xtc_draw_form('checkout_payment',
xtc_href_link(FILENAME_CHECKOUT_AUTHENTICATE_12CHECK, '', 'SSL'), 'post', 'onSubmit="return
check_form();"'));
// 12check Bearbeitung Ende -----

$smarty->assign('ADDRESS_LABEL', xtc_address_label($_SESSION['customer_id'], $_SESSION['billto'],
true, ' ', '<br />'));
$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>');
$smarty->assign('BUTTON_CONTINUE', xtc_image_submit('button_continue.gif', IMAGE_BUTTON_CONTINUE));
$smarty->assign('FORM_END', '</form>');

```

(Fügen Sie bitte nur die Fett und als „12check“ gekennzeichneten Code-Anteile ein!)

## 5.4 Veränderung von "checkout\_confirmation.php"

### Code Zeile 59 - 65 Original:

```
// if no shipping method has been selected, redirect the customer to the shipping method selection
page
if (!isset ($_SESSION['shipping']))
    xtc_redirect(xtc_href_link(FILENAME_CHECKOUT_SHIPPING, '', 'SSL'));

//check if display conditions on checkout page is true
if (isset ($_POST['payment']))
    $_SESSION['payment'] = xtc_db_prepare_input($_POST['payment']);
```

### Code bearbeitet mit Einfügung:

```
// if no shipping method has been selected, redirect the customer to the shipping method selection
page
if (!isset ($_SESSION['shipping']))
    xtc_redirect(xtc_href_link(FILENAME_CHECKOUT_SHIPPING, '', 'SSL'));

// 12check -----
if ((isset ($_SESSION['PHONE12CHECK_POST'])) && (is_array ($_SESSION['PHONE12CHECK_POST']))) {
    foreach ($_SESSION['PHONE12CHECK_POST'] as $s12CheckReqKey => $s12CheckReqVal) {
        $_POST[$s12CheckReqKey] = $s12CheckReqVal;
    }
}
// 12check Ende -----

//check if display conditions on checkout page is true
if (isset ($_POST['payment']))
    $_SESSION['payment'] = xtc_db_prepare_input($_POST['payment']);
```

### Code Zeile 70 - 86 Original:

```
//-- TheMedia Begin check if display conditions on checkout page is true
if (isset ($_POST['cot_gv']))
    $_SESSION['cot_gv'] = true;
// if conditions are not accepted, redirect the customer to the payment method selection page

if (DISPLAY_CONDITIONS_ON_CHECKOUT == 'true') {
    if ($_POST['conditions'] == false) {
        $error = str_replace('\n', '<br />', ERROR_CONDITIONS_NOT_ACCEPTED);
        xtc_redirect(xtc_href_link(FILENAME_CHECKOUT_PAYMENT, 'error_message=' .
urlencode($error), 'SSL', true, false));
    }
}

// load the selected payment module
require (DIR_WS_CLASSES . 'payment.php');
if (isset ($_SESSION['credit_covers']))
    $_SESSION['payment'] = 'no_payment'; // GV Code Start/End ICW added for CREDIT CLASS
$payment_modules = new payment($_SESSION['payment']);
```

### Code bearbeitet ab Zeile 75, Abschnitt tauschen:

```
//-- TheMedia Begin check if display conditions on checkout page is true
if (isset ($_POST['cot_gv']))
    $_SESSION['cot_gv'] = true;
// if conditions are not accepted, redirect the customer to the payment method selection page

// 12check Bearbeitung -----
if (DISPLAY_CONDITIONS_ON_CHECKOUT == 'true') {
    if (!$SESSION['conditions']) {
        $error = str_replace('\n', '<br />', ERROR_CONDITIONS_NOT_ACCEPTED);
        xtc_redirect(xtc_href_link(FILENAME_CHECKOUT_PAYMENT, 'error_message=' .
urlencode($error), 'SSL', true, false));
    } else {
        unset ($_SESSION['conditions']);
    }
}
// 12check Bearbeitung Ende -----

// load the selected payment module
require (DIR_WS_CLASSES . 'payment.php');
if (isset ($_SESSION['credit_covers']))
    $_SESSION['payment'] = 'no_payment'; // GV Code Start/End ICW added for CREDIT CLASS
$payment_modules = new payment($_SESSION['payment']);
```

(Fügen Sie bitte nur die Fett und als „12check“ gekennzeichneten Code-Anteile ein!)

## 5.5 Veränderung von "checkout\_process.php"

### Code Zeile 59 - 63 Original:

```
// avoid hack attempts during the checkout procedure by checking the internal cartID
if (isset ($_SESSION['cart']->cartID) && isset ($_SESSION['cartID'])) {
    if ($_SESSION['cart']->cartID != $_SESSION['cartID']) {
        xtc_redirect(xtc_href_link(FILENAME_CHECKOUT_SHIPPING, '', 'SSL'));
    }
}

// load selected payment module
require (DIR_WS_CLASSES.'payment.php');
if (isset ($_SESSION['credit_covers']))
    $_SESSION['payment'] = ''; //ICW added for CREDIT CLASS
$payment_modules = new payment($_SESSION['payment']);
```

### Code zwischengefügt an Zeile 66:

```
// avoid hack attempts during the checkout procedure by checking the internal cartID
if (isset ($_SESSION['cart']->cartID) && isset ($_SESSION['cartID'])) {
    if ($_SESSION['cart']->cartID != $_SESSION['cartID']) {
        xtc_redirect(xtc_href_link(FILENAME_CHECKOUT_SHIPPING, '', 'SSL'));
    }
}

// 12check -----
if ((isset ($_SESSION['PHONE12CHECK_POST'])) && (is_array ($_SESSION['PHONE12CHECK_POST']))) {
    foreach ($_SESSION['PHONE12CHECK_POST'] as $s12CheckReqKey => $s12CheckReqVal) {
        $_POST[$s12CheckReqKey] = $s12CheckReqVal;
    }
}
// 12check Ende -----

// load selected payment module
require (DIR_WS_CLASSES.'payment.php');
if (isset ($_SESSION['credit_covers']))
    $_SESSION['payment'] = ''; //ICW added for CREDIT CLASS
$payment_modules = new payment($_SESSION['payment']);
```

### Code Zeile 137 - 145 Original:

```
$customer_notification = (SEND_EMAILS == 'true') ? '1' : '0';
$sql_data_array = array ('orders_id' => $insert_id, 'orders_status_id' =>
$order->info['order_status'], 'date_added' => 'now()', 'customer_notified' =>
$customer_notification, 'comments' => $order->info['comments']);
xtc_db_perform(TABLE_ORDERS_STATUS_HISTORY, $sql_data_array);

// initialized for the email confirmation
$products_ordered = '';
$products_ordered_html = '';
$subtotal = 0;
$total_tax = 0;
```

### Code zwischengefügt an Zeile 141:

```
$customer_notification = (SEND_EMAILS == 'true') ? '1' : '0';
$sql_data_array = array ('orders_id' => $insert_id, 'orders_status_id' =>
$order->info['order_status'], 'date_added' => 'now()', 'customer_notified' =>
$customer_notification, 'comments' => $order->info['comments']);
xtc_db_perform(TABLE_ORDERS_STATUS_HISTORY, $sql_data_array);

// 12check -----
if (isset ($_SESSION['PHONE12CHECK_USE'])) {
    if ($_SESSION['PHONE12CHECK_USE'] == 1) {
        if ($_SESSION['PHONE12CHECK_AUTHENTICATED'] == 1) {
            // 12Check ist initialisiert und Authentifiziert - ok
            $sql_data_array = array ('orders_id' => $insert_id,
                'orders_status_id' => '59',
                'date_added' => 'now()',
                'customer_notified' => '1',
                'comments' => $_SESSION['PHONE12CHECK_COMMENT']);
            xtc_db_perform(TABLE_ORDERS_STATUS_HISTORY, $sql_data_array);
        } else {
            // 12Check ist initialisiert aber nicht Authentifiziert!
            xtc_redirect(xtc_href_link(FILENAME_SHOPPING_CART));
        }
        // 12Check ist initialisiert aber Authentifizierung nicht notwendig - ok
    }
    // 12Check Variablen aus dem Speicher entfernen
    unset ($_SESSION['PHONE12CHECK_USE']);
    unset ($_SESSION['PHONE12CHECK_AUTHENTICATED']);
}
```

```

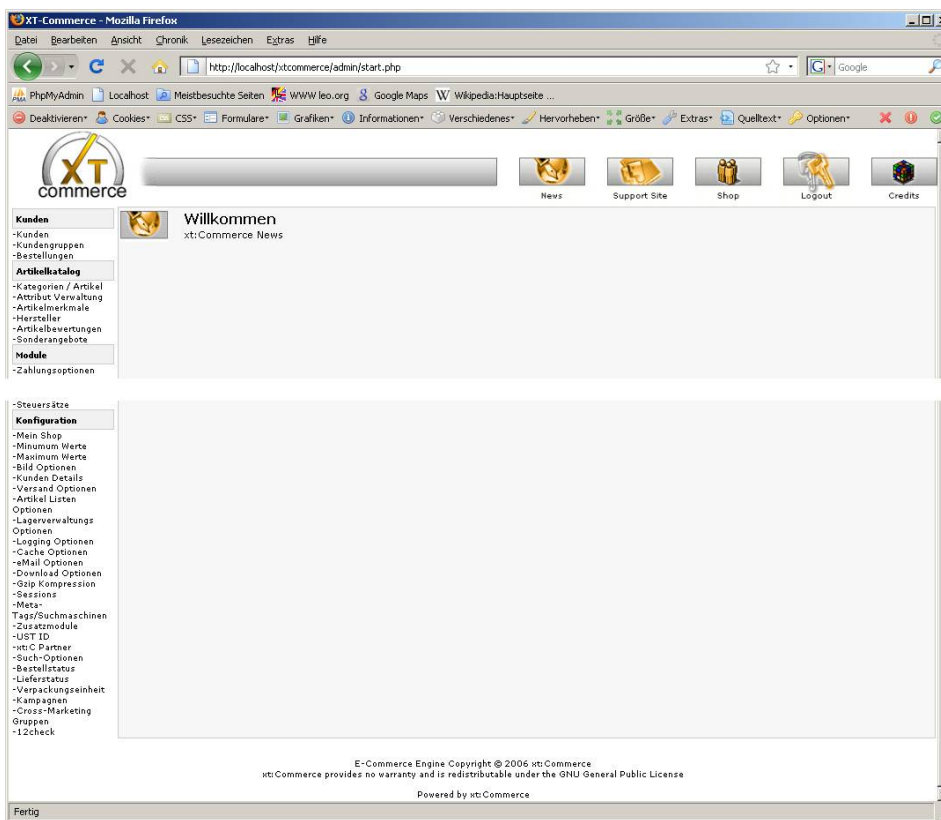
unset ($_SESSION['PHONE12CHECK_COMMENT']);
}
if (isset ($_SESSION['PHONE12CHECK_POST'])) {
    unset ($_SESSION['PHONE12CHECK_POST']);
}
// 12check Ende -----
// initialized for the email confirmation
$products_ordered = '';
$products_ordered_html = '';
$subtotal = 0;
$total_tax = 0;

```

(Fügen Sie bitte nur die Fett und als „12check“ gekennzeichneten Code-Anteile ein!)

## 6. Administration

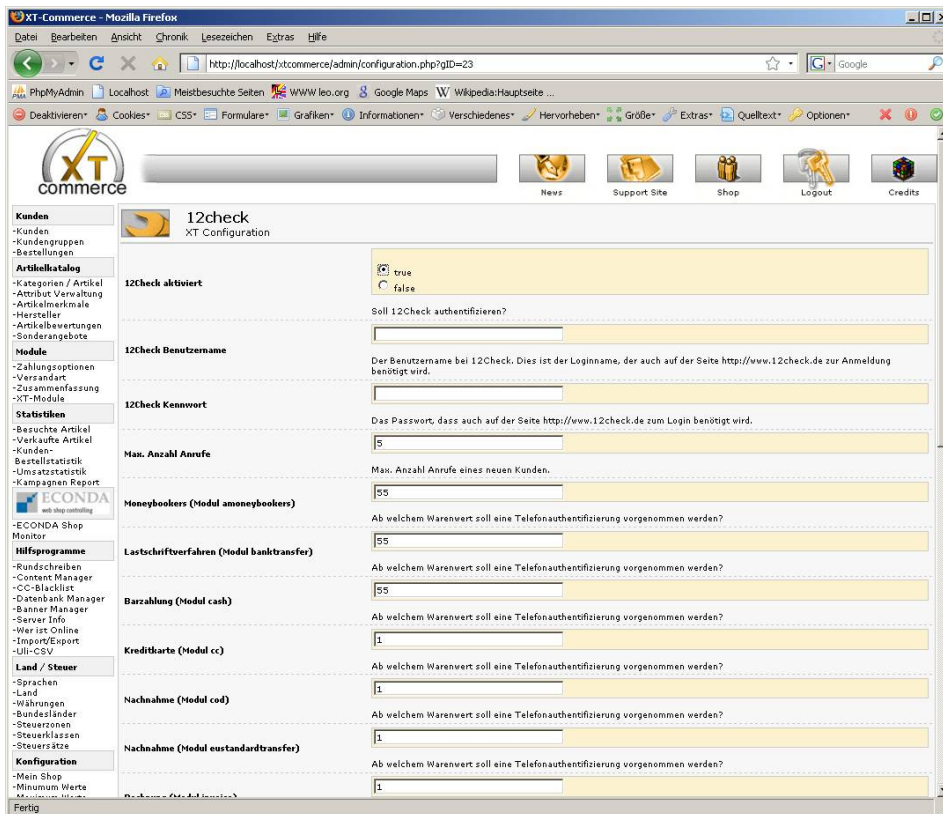
Im Admin-Bereich sollte im Menü links ganz unten der Menüpunkt "12check" erscheinen.



Darin können Sie Ihre Kundendaten für die Kommunikation mit 12check eintragen, angeben wieviele Versuche ein neuer Kunde für die Authentifizierung benötigen darf und das Modul insgesamt ein- oder ausschalten.

Für alle von XT:Commerce ausgelieferten Bezahlmodule ist ein Eintrag für die Mindesthöhe des Kaufwertes für die Authentifizierung durch 12check vorhanden.

Dazu finden Sie einen Eintrag mit einem Vorgabewert für alle 12check unbekanntenen Bezahlmodule.



Wenn Sie 12check für ein Bezahlmodul ausschalten wollen, geben Sie einfach einen unrealistisch hohen Wert wie 1000000 an.

Das Modul ist in die Rechteverwaltung von XT:Commerce integriert. Sie können das Administrationsrecht dafür wie allen für alle anderen Module zuweisen.

Wenn Sie selbst 12check selbst als Administrator testen wollen, setzen Sie in der 12check-Konfiguration den Wert "12Check bei Admins IMMER durchführen" auf "true".

Dann werden Sie (falls Sie für die Bezahlart den Schwellenwert niedrig genug eingestellt haben) immer geprüft, ansonsten merkt sich das Modul eine einmal erfolgreiche Prüfung beim Kunden und prüft nicht wieder.

12check für XT:Commerce, Version 1.09

(c) 2008-2009, das-web - Dipl.-Ing. Ulrich Heiwolt

www.das-web.de