

Emails 3.1.0
















- [Overview](#)
- [Emails](#)
 - [Annotations](#)
 - [Template driven](#)
 - [CMS driven](#)
 - [Mixed approach](#)
 - [Configuring recipients and suppressing emails SaaS](#)

Overview

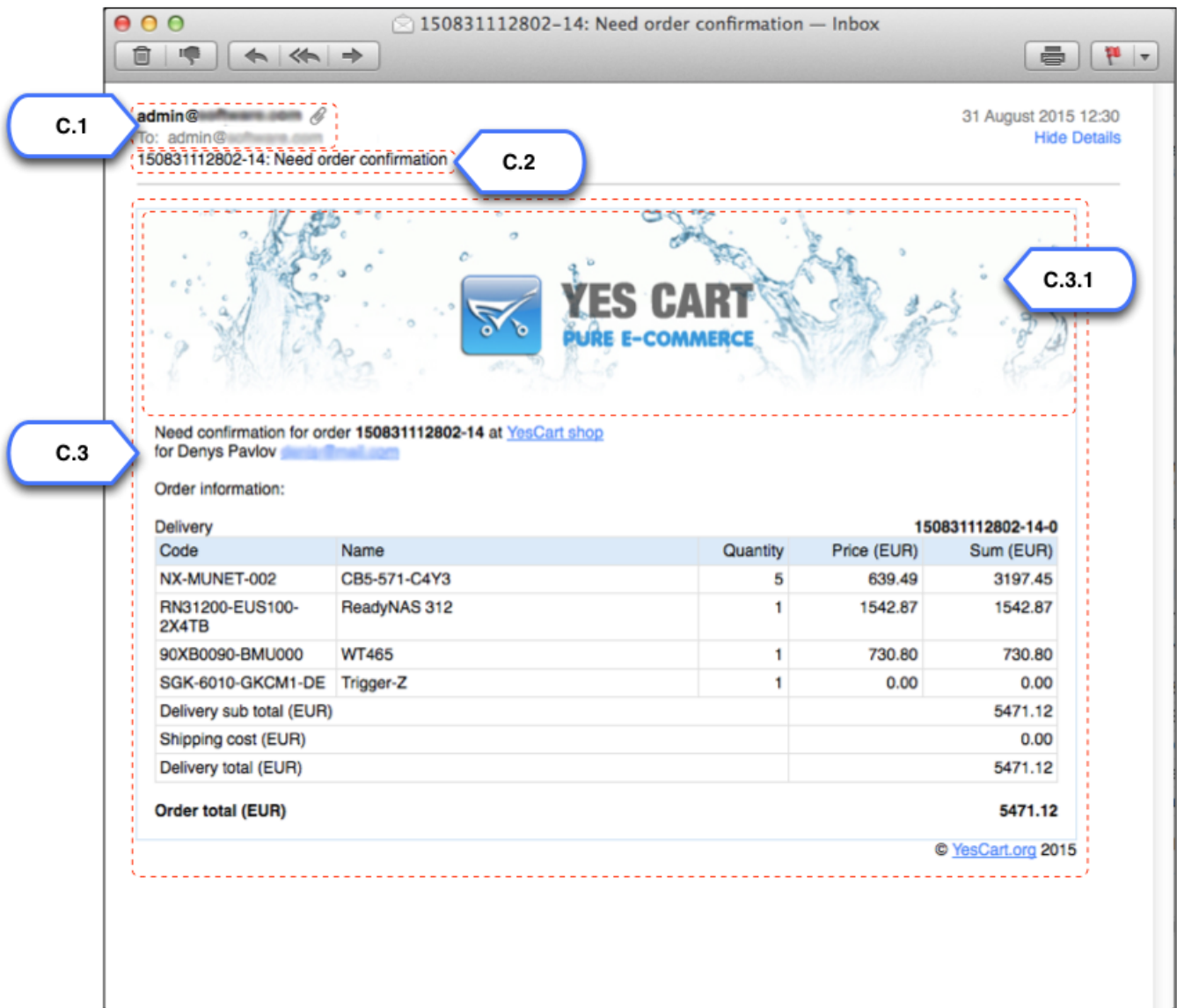
Emails can be configured as templates much in the same way as normal web page templates. **Alternatively they can be managed via CMS** by creating content objects that hold email body, property configurations and embeddable media files.

Default theme implementation provides following email templates:

Theme	Template	Condition	Customer	Callcenter	System
admin	adm-passwd	Admin password email			✓
default	adm-cant-allocate-product-qty	Failed standard allocation		✓	
default	adm-order-canceled	Order has been cancelled		✓	
default	adm-order-confirmed	Offline payment order is confirmed by call centre operative		✓	
default	adm-order-delivery-allocated	Allocation for delivery has been completed, delivery can be packed		✓	
default	adm-order-delivery-inprogress	Shipping is in progress for a delivery		✓	
default	adm-order-delivery-inprogress-wait	Online payment on delivery mode, where something prevents payment from completion. Delivery is waiting for payment to be collected before proceeding with shipping.		✓	
default	adm-order-delivery-packing	Delivery is being assembled at warehouse, marked by warehouse operative		✓	
default	adm-order-delivery-ready	Delivery ready for shipment, marked by warehouse operative		✓	
default	adm-order-delivery-ready	Delivery ready for shipment, marked by warehouse operative. However shipping requires payment to be processed.		✓	
default	adm-order-new	New order has been placed.		✓	
default	adm-order-payment-confirmed	Call centre operative confirmed payment and order can be processed.		✓	
default	adm-order-returned	Call centre operative processed order return.		✓	
default	adm-order-shipping-completed	Call centre operative marked that delivery has been completed. ⚠ This can also be done using third party delivery tracking services.		✓	
default	adm-order-wait-confirmation	New offline payment order is placed and awaits action by call centre operative.		✓	
default	adm-payment	Online pre-payment for an order has been successfully processed.		✓	
default	adm-payment-failed	Online pre-payment for an order has failed.		✓	
default	adm-payment-shipped	Online payment on delivery for an order has been successfully processed.		✓	
default	adm-payment-shipped-failed	Online payment on delivery for an order has failed.		✓	
default	adm-refund	Online refund for an order has been successfully processed.		✓	

default	adm-refund-failed	Online refund for an order has failed.			
default	adm-newsletter-request	Notification sent via Newsletter form .			
default	adm-contactform-request	Notification sent via Contact us form .			
default	customer-change-password	Customer password reset.			
default	customer-registered	Customer registration confirmation.			
default	order-canceled	Order cancellation.			
default	order-confirmed	Order confirmation for offline payments.			
default	order-delivery-readytoshipping	Order is ready to be shipped.			
default	order-delivery-shipped	Order is shipped.			
default	order-new	New order placement.			
default	order-returned	Order returned.			
default	order-shipping-completed	Shipping completed when marked by call centre operative.  This can also be done using third party delivery tracking services.			
default	payment	Online payment confirmation (pre-paid).			
default	shipment-complete	Online payment confirmation (payment on delivery).			

Emails



Annotations

Element	Template	Configurations ¹	Content include ²	Description
C.1 To/From		From email is defined at shop level using "SHOP_ADMIN_EMAIL". Emails addressed to call centre also use shop level using "SHOP_ADMIN_EMAIL". Emails to customer use customer email from customer profile		To/From settings
C.2 Subject		Template driven emails will use property file with key "subject". CMS driven emails will use property config content with key "subject".		Subject of the email. Order updates emails are prefixed with order number.

C.3 Message body				Message body can contain HTML or plain text message. Body templates features Groovy support so dynamic elements can be present in body as well. Message body can be defined in templates or as shop content with special URI.
C.3.1 Embedded objects				Message body allows to define placeholders for embedded objects such as images. This technique allows to provide all media together with the message thus providing a complete look without need for internet connection.

¹Configurations refer to [shop](#) and [category](#) custom attributes that define behaviour of components

²[Content](#) include refers to shop URI suffix for content. For example if shop has code "SHOP10" and content include is specified as "custom_content" then URI of the content include that will be rendered is "SHOP10_custom_content"

Template driven

Template driven emails work much in the same way as regular page templates and are subject to the same inheritance rules as regular templates. Therefore email templates can be defined per theme. Note that content managed email templates have higher priority than regular theme chain (see section below).

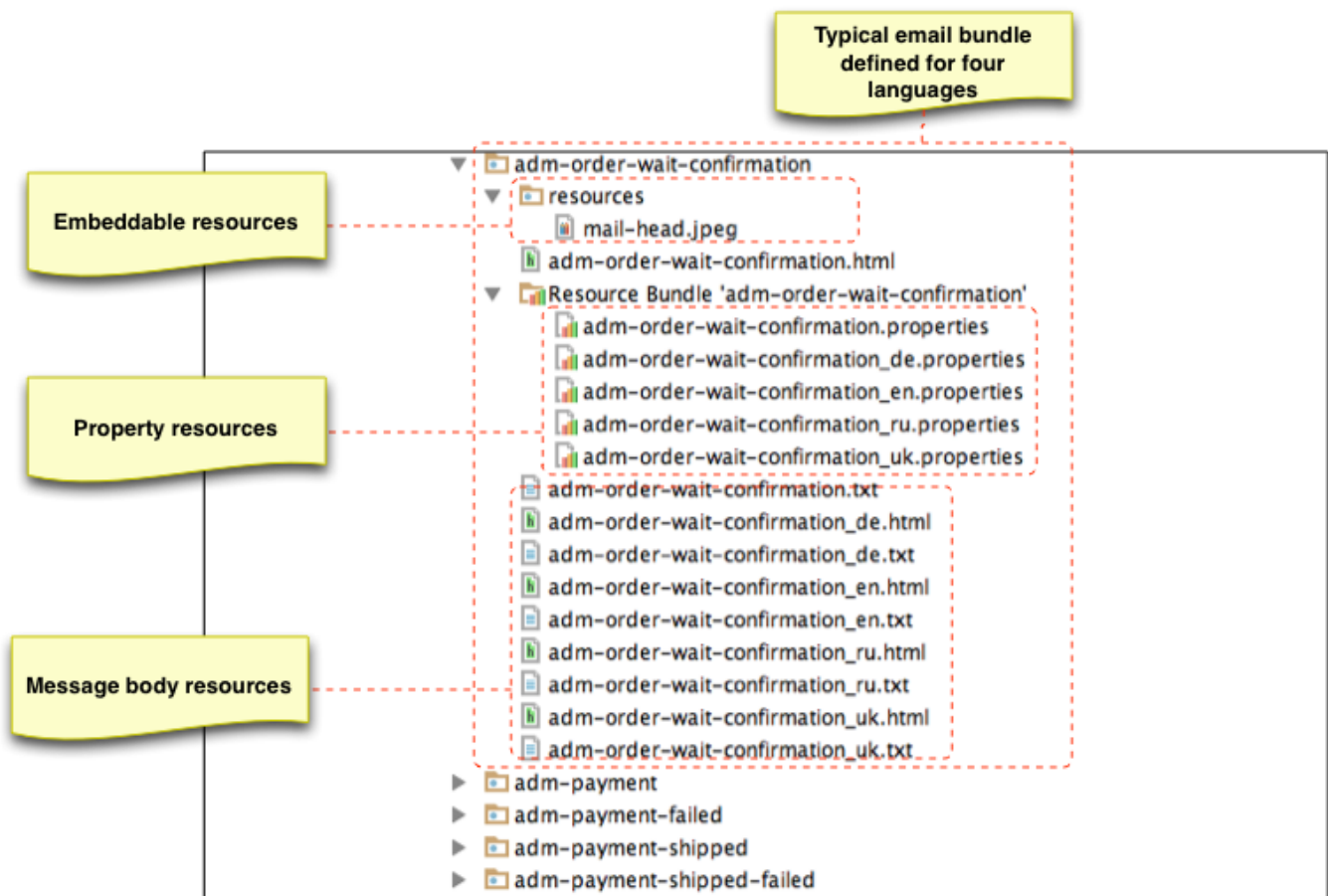
All theme templates are defined within "theme/mail" module. Each email template bundle consists of a number of resources:

- Embedded resources contained within resources sub directory. Each embeddable resource file name must match the placeholder name in the email body template.
- Property resource which defines static configurations for template (usually subject). The property files must be named with same base name as template bundle followed by underscore and then by language code. For example if bundle is "adm-cant-allocate-product-qty" then for English version there must be a file named "adm-cant-allocate-product-qty_en.properties".
- HTML body resource which defines the message body for HTML messages. HTML format is preferred method as it allows greater flexibility in styling emails and using multimedia. The HTML file must be named with same base name as template bundle followed by underscore and then by language code. For example if bundle is "adm-cant-allocate-product-qty" then for English version there must be a file named "adm-cant-allocate-product-qty_en.html".
- Plain text body resource which defines the message body for plain text messages. This is somewhat obsolete format but it is there if someone needs to use it. Plain text file must be named with same base name as template bundle followed by underscore and then by language code. For example if bundle is "adm-cant-allocate-product-qty" then for English version there must be a file named "adm-cant-allocate-product-qty_en.txt".

Ensure that embeddable files do not exceed the size of TMAILPART.PART_DATA column size otherwise Mail composer will fail to save the message.

Typical structure of email template bundle for four languages is depicted in figure 1.

Figure 1: Email theme template bundle "adm-order-wait-confirmation"



Contents of the property file is a key value pair list that can be seen in example 1.

Example 1: Properties "adm-order-wait-confirmation"

```
subject=Need order confirmation
from=shopadmin@demo.yes-cart.com
```

HTML message body file is the actual email itself and represents a mix of Groovy scripts and HTML (see example 2, note that some parts are skipped by '...').

Example 2: HTML "adm-order-wait-confirmation"

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>YesCart pure e-commerce</title>
</head>
<body>

<table width="700" cellpadding="0" cellspacing="0">
  <tr>
    <td style="border: 1px solid #CEE3F6; border-bottom: 0px;">
      <!-- Embedded resource
```

```

-->
    </td>
</tr>
<tr>
    <td style="border: 1px solid #CEE3F6; border-top: 0px;">
        <table width="100%" cellpadding="0" cellspacing="0">
            <tr>
                <td width="2%">&nbsp;</td>
                <td>
                    Need confirmation for order <b>${root.ordernum}</b>
at <a href="<%out.print(shop.getDefaultShopUrl())%>">${shop.name}</a> <br>
                    for $customer.firstname $customer.lastname
$customer.email
                <table width="100%" cellpadding="0"
cellspacing="0">
                    <tr>
                        <td>&nbsp;</td>
                    </tr>
                    <tr>
                        <td colspan="2" align="left">Order
information:</td>
                    </tr>
                    <%
                    def totalSum = 0;
                    for (Object delivery : root.getDelivery()) {
                    def deliverySum = 0;
                    %>
                    <tr>
                        <td>Delivery</td>
                        <td
align="right"><b>${delivery.deliveryNum}</b></td>
                    </tr>

                    ... <!-- Loop over deliveries within order to list
items -->
                    <%
                    totalSum += (delivery.price + deliverySum);
                    }
                    %>
                    <tr>
                        <td>&nbsp;</td>
                    </tr>
                    <tr>
                        <td><b>Order total
(<%=root.currency%>)</b></td>
                        <td
align="right"><b><%=totalSum.setScale(2,
BigDecimal.ROUND_HALF_UP)%></b></td>
                    </tr>
                    </table>
                </td>
            </tr>
        </table>

```

```
        </td>
    </tr>
```

```
...
```

```
</table>
</body>
</html>
```

Example above shows how image can be embedded into email using:

```

```

Note how "mail-head_jpeg" represents file name of the resource with full stop '.' replaced by underscore '_'.

Using Groovy constructs allows to reference various variables within template to include customer data and order details within the email body.

CMS driven

CMS allows to mimic the same structure for email template resources as in template driven approach. The configurations are exactly the same as for file templates but are defined within content body of each resource.

Before attempting theme chain file system resource look up mail composer will try to resolve content object by its URI first. Email resource content object has the following URI naming convention:

HTML, Plain text and Properties

```
[SHOP CODE]_mail_[BUNDLE NAME].[EXTENSION]
```

Embedded file

```
[SHOP CODE]_mail_[BUNDLE NAME]_[FILENAME]
```

For example for email bundle "adm-order-wait-confirmation" for "SHOP10" the following resources apply:

Resource name	Content SEO URI	Notes
adm-order-wait-confirmation.html	SHOP10_mail_adm-order-wait-confirmation.html	Use content body
adm-order-wait-confirmation.txt	SHOP10_mail_adm-order-wait-confirmation.txt	Use content body
adm-order-wait-confirmation.properties	SHOP10_mail_adm-order-wait-confirmation.properties	Use content body
resources/mail-head.jpeg	SHOP10_mail_adm-order-wait-confirmation_mail-head.jpeg	Use "CATEGORY_IMAGE0" attribute

Note that content does not need the _[language] suffix as content is a localisable object. Therefore language specific email template should go to language specific content cody attribute.

Typical structure of email template bundle managed via CMS is depicted in figure 2.

Figure 2: Email CMS template bundle "adm-order-wait-confirmation"

Customer Service
PIM & Master Catalog
Content

SHOP10: YesCart shop
SHOP20: Electronics
SHOP30: Textiles

Typical email bundle defined for four languages

Categories

Categories Add

Embeddable resource

Summary SEO Content Images Attributes

URI SHOP10_mail_adm-order-wait-confirmation_mail-head

Window title

Add locale Remove locale

Locale Display value

Category image

Value shop_header_SHOP10_mail_adm-order-wait

Alternative text

Add locale Remove locale

Locale Display value

Title

Add locale Remove locale

Locale Display value

Load Save Cancel

Customer Service
PIM & Master Catalog
Content

SHOP10: YesCart shop
SHOP20: Electronics
SHOP30: Textiles

Typical email bundle defined for four languages

Categories

Categories Add

Message body resource

Summary SEO Content Images Attributes

URI SHOP10_mail_adm-order-wait-confirmation.html

Window title

Add locale Remove locale

Locale Display value

Content body (en)

Content body in English (en). 1st 4000 characters

Value

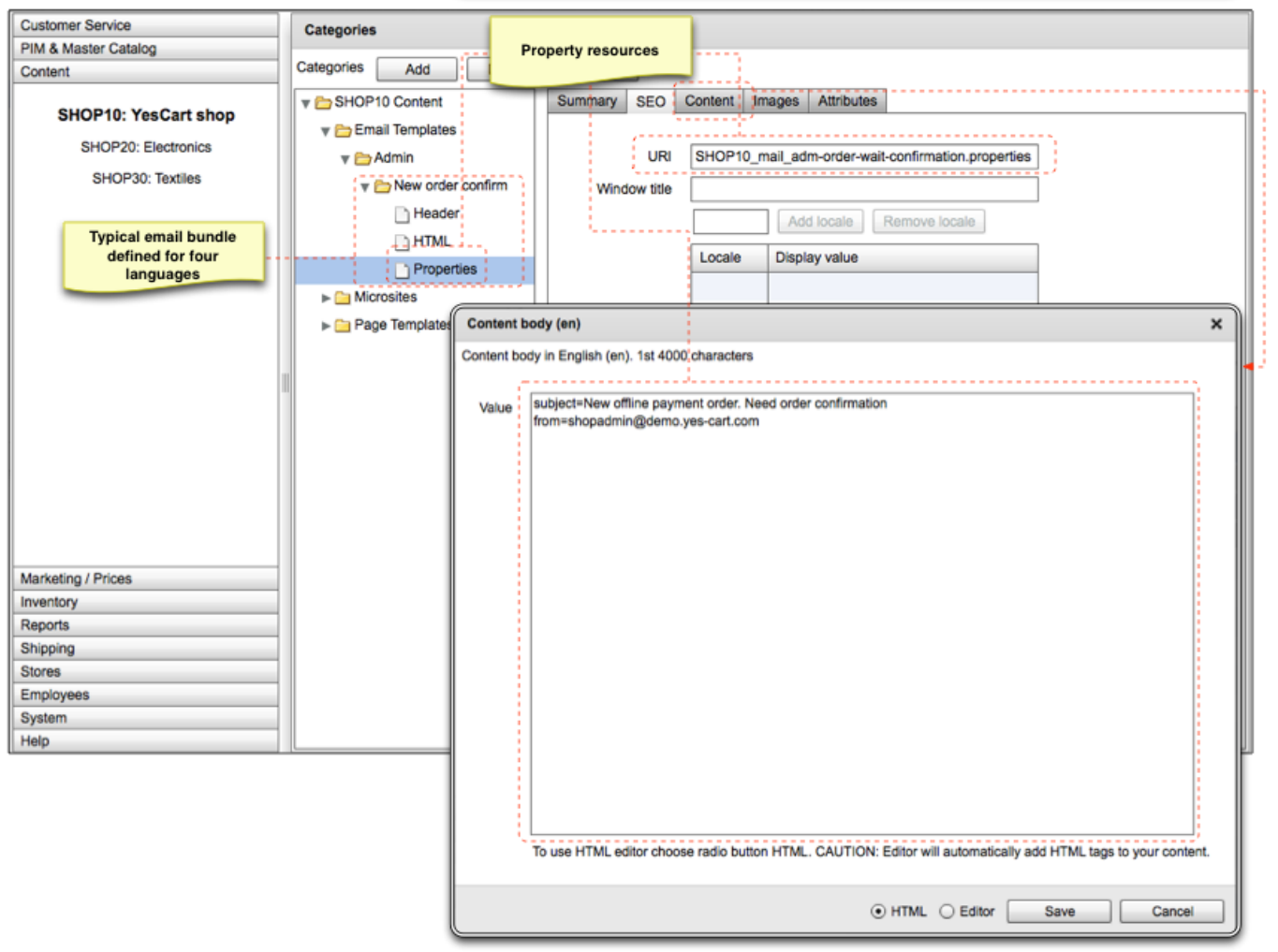
```
<body>
<table width="700" cellpadding="0" cellspacing="0">
<tr>
<td style="border: 1px solid #CEE3F6; border-bottom: 0px;">

</td>
</tr>
<tr>
<td style="border: 1px solid #CEE3F6; border-top: 0px;">
<table width="100%" cellpadding="0" cellspacing="0">
<tr>
<td width="2%">&nbsp;</td>
<td>
This is new offline payment order. Need confirmation for order <b>$root.orderum</b> at <a href="<%out.print(shop.getDefaultShopUri())%>">$shop.name</a> <br>
for $customer.firstname $customer.lastname $customer.email
<table width="100%" cellpadding="0" cellspacing="0">
<tr>
<td>&nbsp;</td>
</tr>
<tr>
<td colspan="2" align="left">Order information:</td>
</tr>
</table>
</td>
</tr>
</table>

```

To use HTML editor choose radio button HTML. CAUTION: Editor will automatically add HTML tags to your content.

HTML Editor Save Cancel



It can be observed that content can be arranged into hierarchy thus allowing better management of email templates. Content can be named to business user convenience as only requirement is correct SEO URI.

Email templates are for internal use only so all content has to have **include** template variation

Resulting email from the above configurations can be seen in figure 3.

Figure 3: CMS driven email

150901145111-15: New offline payment order. Need order confirmation — Inbox


admin@yesware.com

1 September 2015 16:00

To: admin@yesware.com

Hide Details

150901145111-15: New offline payment order. Need order confirmation



This is new offline payment order. Need confirmation for order **150901145111-15** at [YesCart shop](#) for Bob Doe [\[redacted\]](#)

Order information:

Delivery **150901145111-15-0**

Code	Name	Quantity	Price (EUR)	Sum (EUR)
ME181C-A1-BK	ME181C-A1-BK	1	2389.50	2389.50
SGK-6010-GKCM1-DE	Trigger-Z	1	0.00	0.00
Delivery sub total (EUR)				2389.50
Shipping cost (EUR)				0.00
Delivery total (EUR)				2389.50
Order total (EUR)				2389.50

© [YesCart.org](#) 2015

Mixed approach

Email templates were designed to promote as much code reuse as possible by resolving each resource individually. This means that **any part of the bundle can be either theme resource file on file system or CMS content with SEO URI defined**. Mail composer service will resolve each resource to create a composite view.

With this in mind it is possible to define base template in themes and then override properties and header image in CMS, or use header and/or properties from themes and define message body in CMS. There is no right or wrong approach and business users should use most convenient method inline with business individual requirements.

Configuring recipients and suppressing emails SaaS

By default all emails are sent to shop admin email specified in the [shop](#) attributes. In **SaaS** edition of the platform shop has three additional attributes:

Attribute	Purpose	Example
Mail: Shop admin email map	Allows to define emails of recipients of email for specific template	<p>For example snippet below allows to direct new orders to fulfilment department and all confirmed payment notifications to accounting.</p> <pre>adm-order-new=fulfilment@shop.com,</pre>
Mail: Shop admin email CC map	Allows to define additional recipients, so that copies of email can be sent. This is especially useful for reseller sites, where copy of confirmed order is sent to the fulfilment company	<p>For example which configuration below copies of new and confirmed orders are sent to orders@mainshop.com (Note that an email is also sent to recipients in "Mail: Shop admin email map" as usual or to "Shop admin email" if not specified)</p> <pre>adm-order-new=orders@mainshop.com, adm-order-payment-confirmed=orders@mainshop.com</pre>
Mail: Shop customer disabled emails map	Allows to suppress email notifications to customers. This is useful when you wish to disable some email notifications.	<p>For example the following configuration disabled all shipping notifications:</p> <pre>order-delivery-readytoshipping=true, order-delivery-shipped=true, order-shipping-completed=true</pre>