

Specification

DRUPAL E-COMMERCE
Subscriptions, Roles and User Accounts

Version 4.7

Copyright © 2006 Synerger Pty Ltd



Document Version	Revision Date	Changes Made
4.7.1	4 Jul 2006	Initially published

CONFIDENTIAL

Contents

Introduction.....	2
Current State of E-commerce.....	3
Subscriptions	3
Roles.....	3
User Accounts.....	3
Proposed Changes.....	4
Subscriptions	4
Roles.....	4
User Accounts.....	4
Summary of Proposed Changes.....	6
Subscriptions	6
Roles.....	6
User Accounts.....	6
Other Requirements.....	7
What Can I Do To Help?.....	8

CONFIDENTIAL

Introduction

This is a specification for proposed changes to Drupal E-Commerce in the realms of *subscriptions*, *roles* and *user accounts*. To give readers some idea about what these terms mean some definitions are in order.

In much the same way someone can subscribe to a magazine, website subscriptions are a recurring membership to a site. Roles are Drupal's way of grouping access rights and tying these in with the E-commerce system means customers can purchase rights to access site features. User accounts are the rights of a user to login to a site as well as their user profile. The terms subscription, role and user account are used throughout this document with the definition just given.

The specification first explains the current state of Drupal E-Commerce (as of version 4.7) along with reasons why (where appropriate) the changes are proposed. Next, it describes the proposed changes followed by a summary of those changes is given as a quick reference. Other requirements have been stated by stakeholders and they are listed in the penultimate section. Finally, some details are given about how the reader can assist with the changes.

It is important to note that this document attempts to avoid discussion of implementation specifics unless duly warranted such as when a certain direction of implementation is advised.

Stakeholders in the E-Commerce system are asked to read through the document and provide comments.

CONFIDENTIAL

Current State of E-commerce

Drupal E-Commerce currently has some of the functionality for two of the three components. We now examine the current state of the E-Commerce system and why the changes need to be made. We look specifically at Subscriptions, Roles and User Accounts.

Subscriptions

E-Commerce currently has subscription support built-in where products can be set to have a product-specific lifetime. These features are part of the E-Commerce core or, in other words, closely tied with the core E-Commerce code.

A downside of the present implementation is renewal emails being set system-wide rather than on a per-product. As a result, it is impossible to have the first renewal email for product A sent one month before expiry and the mail for product B sent 2 days before expiry.

Another negative thing about the current subscription model is lack of recurring payment support. Support for recurring payments would make it possible to automatically charge a customer's credit card and renew their subscriptions automatically. Two methods exist for recurring payments: Paypal Subscriptions and credit card payments. At this stage we are not making any commitments to include support for recurring credit card payments. There are many security risks involved and we're yet to find an appropriate solution that will ensure people administering Drupal (those that lack experience) aren't exposed to undue financial risk. It is possible we'll build some support for this into the subscription system (an API) leaving the storage implementation up to individual sites. That is the subject of yet another specification so we won't dwell on that further here. On the other hand Paypal Subscriptions support is definitely worthwhile looking into and, while not dealt with directly in this specification, we hope the implementation of subscriptions support includes this or supports its inclusion later.

As of version 4.7, subscription expiry dates are calculated from the payment date rather than the transaction date to allow room for delays in payments such as COD or cheques.

Roles

User roles can be modified at checkout time with E-Commerce 4.7 with one limitation - the roles to be modified are set on a system-wide basis. This restriction causes the system to only allow one subscription level out-of-the-box. Sites requiring multiple subscription levels have to implement their own code for it.

Since 4.7, role changes are effected when payment is received rather than when the transaction is saved in a similar fashion to Subscriptions.

User Accounts

Unfortunately there is currently no built-in support for creating user accounts when selected (by the store admin) products are purchased. A workaround does exist. Set registrations to be available without moderation, select a role (other than Authenticated User) for change at purchase time and set the product to only be available to logged in users. This is, of course, a long way to achieve the right goal and is only possible when site owners don't want to limit site registrations to only those that have purchased the products.

There is a strong push towards having user account creation available outside E-commerce and not included in E-commerce at all. An example workflow would be to send a user to logintobogon to create an account during checkout. Of course, this means the account will be created prior to payment being received and some businesses would not like that, even if the authenticated user role has no rights whatsoever. As this is a point of contention this has still been included in the specification subject to further review through comments by stakeholders

Proposed Changes

While the following proposal for changes to the system is indeed thorough it is important that these remain the subject of scrutiny. Any valid amendment requests will be considered and incorporated if they fit with the majority of stakeholders.

A common theme to keep in mind while contemplating these changes is the need for E-commerce to have a comprehensive event driven model built in to expose events such as subscription expiry. This will allow module developers to extend built-in features easily where they see fit. One of the long term goals is to build an event model that makes Drupal surpass its competition in not just efficiency but extendibility. For this to be successful the core of E-commerce itself will need to expose the events for all workflow state alterations as well as payment state alterations. Modules such as that dealing with subscriptions will also need to fire relevant events.

Subscriptions

We propose to improve the system by adding the renewal reminder emails in as a product attribute (shown on the product edit form) in much the same way that subscription duration is currently. A reminder email format attribute is to be added to each product allowing the store admin to choose either a system-wide default format or one specific to the product. The subscription related code will be removed from the core of E-Commerce and placed into a module.

On top of that the system will require some useful handling of expirations and it appears that some type of productapi hook will need to be added to enable useful handling of the expiration event.

Roles

The roles related changes will involve adding a role selection widget to the product edit page allowing the store admin to select the roles changed for each product when they are purchased. By default, products do not modify roles. Role related code will be removed from E-Commerce core and placed into a module.

In order for the roles module to effectively process expiration events a checkbox will need to be added to select whether roles are stripped from the user upon expiration. If no roles are selected in the widget then no roles can be stripped from the user. Thus, even if this checkbox is checked when no roles are selected no effect is made on the user account.

User Accounts

User account creation is a non-trivial addition to E-Commerce because it requires integration with the registration and user systems of Drupal where security and data integrity are paramount.

A checkbox ("Purchase of this product creates a user account") to be added as a product attribute so each product can individually be marked as creational or non-creational. This is vital to having a system where subscriptions are sold alongside other tangible products.

Another checkbox ("Block expired accounts") will be required as well to ensure the user account module will deal with product expirations correctly. This checkbox must be unchecked and disabled if the product is non-creational.

Processing the purchase can be done in one of three ways.

1. The system will request a username during the checkout process. This will happen before the review page and only when the user is not logged in. At this point the username entered will be compared using the rules used in registration to ensure it's not a duplicate. If the email address and the username entered are the same as one already recorded in the system a login screen is displayed with the username already filled in. After login or entry of valid new username the checkout procedure continues to the review page. The review page will show the user the username they've selected or are using. Payment process then begins.

In the above process the time between when the username is collected and the payment is received (thus user account created) can be any length. To avoid the certain race condition it is important to have some way to prevent other users selecting the same username during this time. Ideally this would also prevent username duplication in registrations unrelated to E-Commerce. One such way would be to create a blocked account for the user with some attached data linking it to the transaction taking place.

Upon successful payment a user account is created/unblocked, the transaction is linked to that user, a registration/confirmation email is sent to them, logs entry is added to show a new account was created and they are automatically logged in going to the user profile editing page allowing them to adjust their password. Of course, the automatic login is only performed when the user is not logged in already.

2. One other possible scenario is not collecting a username during checkout and instead the user follows a link in the purchase confirmation email to a special registration page exposed by E-Commerce. This will allow email address confirmation prior to the user gaining access to the system giving an additional security precaution. This will require some special handling for expired accounts being renewed (keep in mind that expired accounts may have blocked accounts).
3. A random username and password are selected by the system at checkout. This will be something like userN or memberN where N is the userid of the new user account. If some user already has this username, the system will simply produce user<5 random chars> using user_password() to produce the 5 chars and repeat until it finds something that isn't taken already. Once the user has logged in (either automatically or later through details sent via email) they can change their username to something more desirable.

This method will require some extra handling for previous account holders wishing to reuse their account after renewing.

Out of the three possible implementations 3) is the easiest to implement and the author's preferred method.

Summary of Proposed Changes

Following is a summary listing of the proposed changes. See the previous section for more information about each one.

Subscriptions

- Add renewal reminder attribute to products
- Add 'subscription expired' hook to handle expiration
- Remove current subscription code from E-Commerce core

Roles

- Add role selection widget (attribute) to products
- Add checkbox to products to select whether roles selected are stripped on expiry
- Remove current role code from E-commerce core

User Accounts

- Add creational flag to products
- Add checkbox to products to select whether user accounts are blocked on expiry
- Either collect username during checkout or have payment process give special rights for user to access registration page or generate username and password
- Usernames can not be duplicated ensuring system integrity
- Purchase resulting in account creation is associated with new user account
- System handles users that have expired accounts

Other Requirements

Some other requirements were received in the task issue¹ on drupal.org and they are worth mentioning here. Each requirement has a comment ID, author, abridged requirement and proposed solution.

Comment ID	Author	Requirement
#11	@wpd	Buy membership but have purchase affect another user account
Solution: Introduce gift subscriptions using the coupon system otherwise this can be done once the implementation of above user account specification is completed by supplying the other user's email address at checkout and the account username/password will be sent to them.		
#14	@coupet	Recurring products have setup costs
#14	@coupet	Require a previous purchase of a specific product before allowing purchase of another
Solution: Introduce product dependencies into the system.		
#14	@coupet	Recurrence schedule selected by user
Solution: Create sub-products for each different recurrence schedule		
#14	@coupet	Trial purchases allowing different (e.g. introductory) pricing
#14	@coupet	Different user categories have different versions of the product
Solution: Introduce role-based purchase restrictions		
#14	@coupet	Each product has unlimited attributes one can collect from the user
NO SOLUTION PROPOSED YET		
#15	@coupet	Generate notices for invoice, payment due, payment overdue, service suspension, payment declined, payment received
Solution: Using the specified framework to build the support required for subscriptions, the addition of notices will be very intuitive		
#14	@coupet	Free access to a product once another product is purchased
#14	@coupet	Grant extra role for specified period of time with purchase of a product
Solution: Introduce product associations (extension of dependencies) whereby a product can be linked to other products. In the cart view, these product associations can then facilitate display of products you get for free, optional (or suggested) products, recommended products or upgrades. In order for this to work the link direction must also be set so you can have one-way dependencies and co-dependencies.		

¹ <http://drupal.org/node/53888> - move subscriptions to own module

What Can I Do To Help?

This specification is intended to give the community an idea of what is proposed for Drupal E-Commerce and to also specify what is required for inclusion of contributed code should someone want their subscription/role/user account code added to Drupal E-Commerce.

Your contribution is very important to the successful progress of Drupal E-Commerce. For progress with this specification please provide either supporting comments or clearly written addendums to this specification so we can encompass as many different configurations as possible.

CONFIDENTIAL