learn drupal with forcontu

Expert in Drupal

Course in creation and administration of Websites with Drupal 7

7

Intermediate

Fran Gil

www.forcontu.com

Expert in Drupal 7

Intermediate

Course in creation and administration
of Websites with Drupal 7

Fran Gil

learn drupal with
forcontu
www.forcontu.com

Expert in Drupal 7. Intermediate

Course in creation and administration of Websites with Drupal 7

Learn Drupal with Forcontu Collection

ISBN-13 (Printed edition): 978-84-942763-0-9
ISBN-13 (Electronic edition, PDF): 978-84-942763-1-6

Author: Fran Gil Rodríguez

With the assistance of the team of tutors of Forcontu: Iván Méndez López, Miguel Ángel Naranjo Moreno and José Miguel Rodríguez Arias

Cover design: Ateigh Design
Marketing and Distribution Manager: Laura Mª Fornié Alonso

Published by Forcontu S.L.

For further information about the content of this book or the distribution channels, please write directly to info@forcontu.com or visit the Website www.forcontu.com.

You can find complementary resources to this book at www.forcontu.com.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Presentation

One of the great challenges facing us to use Drupal is, without doubt, **overcome its steep learning curve**. Mastering Drupal is a costly goal and requires effort and dedication. With more than 3 years of experience in Drupal 6 training, Forcontu launches the course **Expert in Drupal 7**, a training plan improved and expanded to train Web development professionals in Drupal 7.

The Course **Expert in Drupal 7** is provided online. It is guided and supervised directly by the team of tutors of Forcontu.

The course **Expert in Drupal 7** is divided into 3 modules or training levels (beginner, intermediate and advanced) and **requires programming knowledge**. As an alternative, for those people who do not have programming skills, there is the possibility of studying only the beginner and intermediate levels.

Many companies and organizations are betting on Drupal as a solution for their Websites development, and on **Forcontu** to train their working team to the highest level. More than 1,000 people have already tested our courses and materials, both Drupal 6 and Drupal 7.

Fran Gil Rodríguez
Co-founder and CEO of Forcontu

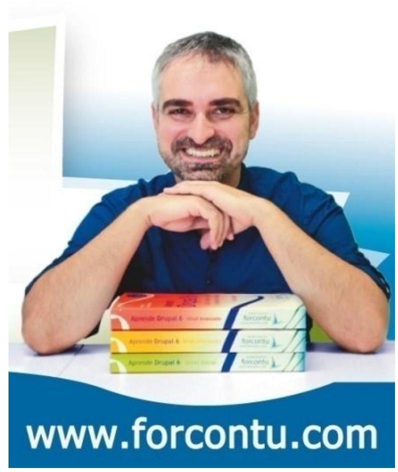
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

About the author

Fran Gil Rodríguez
Co-founder and CEO of Forcontu

fran.gil@forcontu.com
@frankgil76



Fran Gil Rodríguez is a Technical Telecommunications Engineer from the University of Las Palmas de Gran Canaria (2002), holding a Master's degree in the Management and Administration of Information Technologies and Systems from the UOC (2006) and a University Master's degree in Multidisciplinary Computing, specialising in Electronic Teaching and Learning from the University of Alcalá de Henares, Madrid (2009).

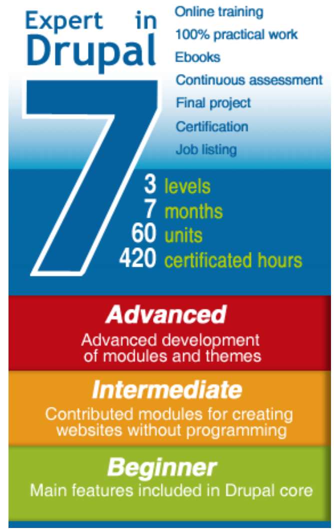
His professional career was initially associated with the University of Las Palmas de Gran Canaria (ULPGC), first as the coordinator of the team for developing the ULPGC Website (2002 to 2006) and then as Head of projects and coordinator of the ITC development department ULPGC S.L. (2006 to 2009). He has participated in numerous projects for developing applications and Websites, as head of Project, an analyst and developer, working with Drupal since 2006.

2009 saw the creation of the Forcontu project, a company specialising in Website development and advanced training in Drupal. Following the success of the Drupal 6 training plan, launched in 2009, the author presents us with an enlarged and improved Drupal 7 training plan: Expert in Drupal 7.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

0 Expert in Drupal 7

0.1 Course description



Teaching method

The Expert in Drupal 7 course is a 100% online course with supervised practice labs. In addition to the virtual classroom, students will be provided with web hosting for their own Drupal installation.
Through the virtual classroom, the course tutor will be in contact with students, guiding them in their learning process and proposing the work necessary for passing the course. Students will hand in their work directly in their assigned web space, **building their own web site** following the proposed activities.

The student will be able perform activities at any time of the day, in order to adapt the training to his/her professional or private needs.

Length and modality

The course is made up of **60 units** and divided into **3 modules**: beginner, intermediate and advanced. After completing all three modules, students will develop and complete a final project. The total course duration is 7 months with 420 hours of certified work.

While this formative experience requires a significant amount of effort, you will save a considerable amount of time when compared to studying Drupal on your own.

Learning objectives

- The main objective of Expert in Drupal 7 is to offer students professional training in design, development, implementation and management of complex websites, using Drupal. The course materials are designed to train students how to complete successful implementation of professional websites using Drupal. However, to reach these objectives, the following secondary objectives must be achieved:

- **Acquire a good knowledge in Drupal.** Having a good base is essential for defining a good strategy for the installation, configuration and programming of features in Drupal. In this course we place special emphasis on the need to study Drupal in an orderly and guided fashion, carrying out work ranging in difficulty from low to high. For this reason we believe that all of the activities proposed in the training plan are important for building a good solid foundation, regardless of the student's background.
- **Knowing Drupal architecture in depth**. Knowing Drupal architecture at the programming level is essential in understanding how the modules interact with the system, as well as developing modules that adapt to the core and the rest of the installed modules on a Website in a proper manner.
- **Essential additional modules.** The Drupal core is only a starting point; it will always be necessary to add additional modules. The most important modules that are offered will be studied in this course. Some of these modules have become essential, because they have been

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

used in prior versions to Drupal 7, having enjoyed widespread acceptance in the community for years. Other modules created specifically for Drupal 7 will also be studied.

- **Developers' tools**. We will study some of the programming and system administration tools with the aim of speeding up the work with Drupal.
- **Adaptation to new versions and modules.** The student will have the necessary skills to adapt to the continuous changes of version and to manage the installation and configuration of new modules contributed by the community successfully.
- **Modules Programming**. When the available modules are not enough to achieve the functional requirements of our project, we will need to implement new modules. We go into greater depth in the API of Drupal and the Drupal programming style to develop professional modules.
- **Learn how to contribute to the community**. The Drupal success would not have been possible if it did not have thousands of enthusiasts, users and developers, who have shared their developments, knowledge and experiences with the community. This is why we believe that, wherever possible, we must contribute to the community, whether sharing and maintaining developed modules, reporting bugs and solutions, or helping other users.

What does the course cover?

Through this course you will learn to install, configure and administrate Websites with Drupal 7. You will also be able to develop new functionalities that allow adapting a Website to the required functional requirements. Becoming an expert in Drupal 7 you will be trained to deal with, individually and in team, the design and implementation of complex Websites. The acquired training will be helpful in getting you into the labor market, improving your career or even undertaking your own project based on web technologies.

Who should attend?

The Expert in Drupal 7 course is aimed at web site developers with programming skills that require the development of professional Web projects with Drupal 7, either for third parties or in order to carry out their own business ideas.

Prior Knowledge

No specific academic qualification is required to gain entry to the course. The beginner and intermediate levels can be studied in their entirety with no programming or database management knowledge. Only a minimum knowledge of computer basics and Internet user-level is needed.

Technical requirements

It is necessary to have a computer with Internet connection and web browser to take this course. During the course it will be necessary to install an FTP client (FileZilla or similar).

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

0.2 Course structure and content

What the course includes

The Expert in Drupal 7 course includes the following services and materials:

- **Materials organized in English.** There is a lot of information on Drupal available in books, blogs, forums, etc. At Forcontu, we have developed organized training materials consisting of different learning subjects to ensure the professional training in Drupal (compulsory work, practical cases, enhancement activities, etc). The course includes **3 Expert in Drupal 7 books** in Web page format and an eBook (PDF).

- **Web hosting** for the training plan activities and the final project undertaking.

- **Continuous assessment** of the work carried out during the course, providing feedback on the possible mistakes.

- **Practical examination after each module**

- **Final Project evaluation.**

- **Follow-up on the student's progress**. Tutors will carry out continuous monitoring of the student and will answer the doubts raised in the classroom forums.

- **Certificate for passing part of the module.**

- **Expert in Drupal 7 Certificate**. After successfully passing the training modules and the end of course project.

- **Letter of introduction**, for students who have completed the course Expert in Drupal 7 with a grade of Excellent.

- Consultancy after the course in order to **migrate your Website** (final work) to your personal hosting site.

Module I: Beginner

Beginner level is aimed at those people with or without technical training, who want to start creating Websites. It is also very useful for companies or professionals who wish to create their professional or personal Website without depending on external assistance. At this level, the aim is for the student to acquire sound basic knowledge in order to tackle the following educational levels with guarantees.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

This module will be studied for 4 weeks and it will certify 60 hours of work.

1. Introduction to Drupal
2. Drupal installation
3. Administration area
4. Content management
5. Content types
6. Menu management
7. Blocks management
8. Themes
9. Extending functionality with modules
10. Management of users, roles and permissions
11. Taxonomy
12. Text formats and WYSIWYG editors
13. Blogs
14. Forums
15. Structured documents: Books
16. Media Galleries
17. Searches
18. Languages and translation
19. Access statistics
20. Website management, maintenance and updating

Final module exam

Module II: Intermediate

The **Intermediate level** complements the beginner level. It is aimed at people who already had previous contact with Drupal and want to understand it in depth and make better use of its full potential. Completing this level is very useful for building professional Websites with features that go far beyond company webs or personal blogs. Programming knowledge is not required to study intermediate level.

This module will be studied for 8 weeks and it will certify 120 hours of work.

21. Creating forms with Webform
22. Advanced content types
23. Views
24. Automatic image processing
25. Extending menus with additional modules
26. Extending blocks with additional modules
27. Panels
28. Display Suite
29. Customizing Themes
30. Actions and Triggers
31. Rules
32. Defining contexts
33. Extending users
34. Workflow
35. Multilingual sites
36. E-commerce: Introduction to Ubercart and Drupal Commerce
37. SEO Positioning with Drupal
38. Social tools: voting and sharing content
39. Newsletter Subscription
40. Other Features

Final module exam

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Module III: Advanced

Advanced level is intended for PHP and MySQL developers who have also had a prior contact with Drupal. This level is designed for Website developers who work for for third parties or for those entrepreneurs with an advanced technical level who wish to launch their own business ideas.

This module will be studied for 12 weeks and it will certify 180 hours of work.

41. Drupal development best practices
42. Tools for developers
43. Drupal Architecture
44. Creating modules
45. Database access and database schemas
46. Menu system
47. Creating forms
48. Programming blocks
49. Programming users and permissions
50. Programming content types
51. Programming entities and fields
52. Programming actions and triggers
53. Working with Files
54. Search system
55. Translating modules
56. Creating themes
57. Using jQuery and Ajax
58. Features
59. Installation Profiles
60. Contributing to the Drupal Community

Final module exam

Final Project

The final project may be taken during the course, but there will be 4 weeks available after the three modules to complete and submit it. This Project will certify 60 hours of work.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

0.3 Methodology, materials and evaluation

Methodology

The **Expert in Drupal 7 is an online supervised course**. The student must carry out the proposed activities on the hosted website, which will be corrected and assessed by the tutor on an ongoing basis, as they are handed in. The e-learning platform will give access to the course contents and the necessary tools for facilitating communication among the participants in the course, the correction of activities and the answering queries.

The schedule includes detailed plans of the work to be carried out, continuous monitoring of the student's progress and evaluations of the student's activities and work completed.

The course is arranged based on a recommended timetable for handing in work, although the students can follow their own pace of work and hand them in throughout the course, but they must always do this before the end of each module. The student can work on his/her web host and in the classrooms used in any schedule, although the questions expressed in the forums and correcting work will generally be done during business hours. Our tutors' full-time commitment ensures flexible communication and corrected activities handed back on a daily basis.

E-learning platform

Once registered for the course, the student will be added onto the educational platform and placed in the classroom corresponding to the module to be studied.

Tutors and students may communicate freely through **forums** (in a way that is public to the entire group), or by **internal e-mail** (privately, only for the stated recipients) for each course. By default, a duplicate of these communications will be sent to the external electronic e-mail address provided by the student.

Students in the same group can contact each other and share their experiences and the results of their work and proposed materials.

The classroom is arranged in the following way:

- **Guide Forum** (Tutor's Forum). Only tutors can publish on this forum, and this will be used to report general issues concerning the course or other issues that may be of interest to the Group.

- **Learning Forum** (Course Forum). Both tutors and students will be able to post on this forum, and it will be used to discuss issues related to Drupal and the course. The students may raise their questions about the work and other general enquiries about Drupal here.

- **Share Forum** (general forum). This is an open forum where students and tutors can share other experiences aside from the course and Drupal.

- **Inbox**. The internal mail enables messages to be sent to other classmates and tutors.

- **Participants**. List of participants on the course, including tutors and students.

- **News.** The latest posts added by tutors in the Forum guide will be published in this block.

- **Upcoming events.** This block shows upcoming events related to course planning. Participants can create private events.

- **Calendar.** Displays events in timetable form.
- **Online Users.** A participants' list shows who is connected at any one time.

Profile (administration). Allows you to administer the user profile: modify the login password, modify the personal data, change the photo, etc.

Educational materials

Materials are written in English and consist of:

- EBooks of the **Learn Drupal with Forcontu set,** corresponding to the registered level. The books will be available in the classroom in digital format (html or PDF).
- **Web Hosting** for doing the work and the student's final project.
- **Links** to external resources.
- **Activities**. Each unit will include activities. That it must be carried out in the time set so the tutor can assess them later on.

Evaluation

During the course each piece of work will be assessed, corrected and rated by the tutors, providing continuous feedback to the student.

At the end of each module the student will get a **continuous assessment rating** based upon the individual marks obtained in their work. To achieve each module it will be also necessary to pass a practical examination. This examination is necessary to validate the mark obtained in the continuous assessment so the student will have only a pass or fail grade. Once the module has been passed, a partial certificate of accomplishment of the module will be issued.

Once the three modules of the course have been passed, the student will have to carry out a Final Project, consisting of the development and documentation of a Website. By passing all the formative modules and the Final Project the student will obtain the **certificate of Expert in Drupal 7.**

Upon completion the course the **certificates of follow-up and accomplishment of each passed module and the certificate of Expert in Drupal 7** will be delivered by ordinary post detailing, among other things, the number of working hours, the content of the course and the final mark obtained.

The **final grade of Expert in Drupal 7** will be obtained by scoring the marks attained in each of the modules and final project, having regarded to the following percentages:

- Module I: Beginner (15%)
- Module II: Intermediate (30%)
- Module: III Advanced (40%)
- Final Project (15%)

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Drupal Certification with Forcontu

Companies increasingly demand professional Web developers, experts in Drupal, and they usually ask for a training certificate and/or experience in Drupal. Another new development introduced into our training plans is the new model of **Drupal certification with Forcontu.**

All training progress with Forcontu will be available on a unique URL that you will be provided with at the beginning of the course. As you pass the modules or additional activities of the lifelong learning Plan, these achievements will be reflected directly on your **Drupal certification with Forcontu.**

If you would like, you can include this URL in your curriculum vitae so the companies that are in your level of knowledge of Drupal can collate the information provided there in.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Table of contents

Presentation iii
About the author v
Expert in Drupal 7 vii
Course description vii
Course structure and content ix
Methodology, materials and evaluation xii

Table of contents xv

Unit 21. Creating forms with Webform 1
21.1. Installation and configuration of Webform 2
21.2. Form creation with Webform 5
21.3. Email configuration 21
21.4. Form settings 24
21.5. Results analysis 27
21.6. Permissions 28
21.7. Validation of form fields 29
21.8. Use of taxonomy in selection lists 33

Unit 22. Advanced content types 35
22.1. Content types definition 36
22.2. Fields in core 38
22.3. Multimedia fields 47
22.4. Date fields 49
22.5. Fields to reference other entities 52
22.6. Field display 54
22.7. Automatic URL aliases using Pathauto 57
22.8. Field and node permission control 65
22.9. Additional modules 69

Unit 23. Views 75
23.1. Views module introduction 76
23.2. View creation 80
23.3. Basic settings 84
23.4. Fields 92
23.5. Sort criteria 97
23.6. Filter criteria 100
Practical Case 23.1. Create the Latest movies view 103
23.7. Contextual filters 111
Practical Case 23.2. Creating a view with contextual filters 114
Practical Case 23.3. Creating a view with default values in the contextual filters 119
23.8. Relationships 123
Practical Case 23.4. Adding related fields 124
23.9. Other advanced options 127
23.10. Permissions 128
23.11. Additional modules 129

Unit 24. Automatic image processing 135
24.1. Image styles 136
24.2. Image styles in Fields 141
24.3. Image styles in Views 142
24.4. Additional modules 144

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Unit 25. Extending menus with additional modules 147
25.1. Breadcrumb Display 148
25.2. Custom breadcrumbs 151
25.3. Menu Link Improvements 153
25.4. Building submenus with Menu Block 158
25.5. Menu Link Access control...161
25.6. Improving Menu display...162

Unit 26. Extending blocks with additional modules.. 169
26.1. Additional elements in blocks 170
26.2. Improving block display 176
26.3. Access control and block visibility 179
26.4. Other modules..181

Unit 27. Panels 185
27.1. Introduction to the Panels module 186
27.2. Panel Pages 188
Practical case 27.1 Create a panel with arguments 201
Practical case 27.2 Create a panel with two variants...208
27.3. Managing panels...212
27.4. Mini panels..218
Practical case 27.3 Create a mini panel..223
27.5. Panel nodes...227
27.6. Integrating Views into Panels 230
Practical case 27.4 Integrate a view into a panel 232
27.7. Additional modules..238

Unit 28. Display Suite 243
28.1. Introduction to Display Suite 244
28.2. Displaying entities...247
28.3. Fields...257
28.4. Extras 263
28.5. Displaying views 266
28.6. Displaying forms 268
28.7. Displaying search results 272

Unit 29. Customizing themes 277
29.1. Theme structure 278
29.2. Theme template files...280
29.3. Introduction to web layout with HTML and CSS 287
29.4. Base themes and sub-themes..304
Practical case 29.1 Creation of a sub-theme with Zen 304

Unit 30. Actions and Triggers 315
30.1. Introduction to Actions and Triggers 316
30.2. Simple and Advanced Actions...318
30.3. Triggers 322

Unit 31. Rules 323
31.1. Installing and Configuring Rules 324
31.2. Rules and Events 326
Practical Case 31.1 Creating a Rule...336
31.3. Components 342
Practical Case 31.2 Create a Component 347
Practical Case 31.3 Reuse a Component..354
31.4. Planning..357
31.5. Integration of Webform With Rules..359

Unit 32. Defining contexts.. 361
32.1. Installing and Configuring Context 362
32.2. Create contexts...363
32.3. Defining conditions..367

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

32.4. Defining reactions 373
32.5. Using context to change design 376
32.6. Additional Modules 379

Unit 33. Extending users 383
33.1. User Images 384
33.2. Multiple User Profiles 392
33.3. Access Roles and Permissions 396
33.4. User Login 405
33.5. Massive Operations over Users 412

Unit 34. Workflow 415
34.1. Reviewing content with Revisioning 416
34.2. Planning content with Scheduler 425
34.3. Access to Premium Content 429

Unit 35. Multilingual Sites 433
35.1. Language Configuration 434
35.2. Internationalization Module (i18n) 438
35.3. Mulitlingual Variables 440
35.4. Content Translation 442
35.5. Menu Translation 448
35.6. Block Translation 454
35.7. Content type and Field translation 455
35.8. Taxonomy Translation 457
35.9. Views Translation 459
35.10. Additional Modules 460

Unit 36. E-commerce: Introduction to Ubercart and Drupal Commerce 463
36.1. The Ubercart module 464
36.2. Ubercart: Managing and Configuring the Store 466
36.3. Ubercart: Products 478
36.4. Ubercart: Purchases and orders 487
36.5. Ubercart: Conditional actions 491
36.6. Ubercart: Other functionality 492
36.7. The Drupal Commerce module 502
36.8. Commerce: Administration and configuration of the store 504
36.9. Commerce: Products 512
36.10. Commerce: Purchases and orders 515
36.11. Commerce: Other functionalities 519

Unit 37. SEO Positioning with Drupal 521
37.1. Introduction to SEO Positioning 522
37.2. Information For Search Engines 523
37.3. Content Structure 529
37.4. Analysis and Optimization 534

Unit 38. Social tools: voting and sharing content 545
38.1. Content voting 546
Practical case 38.1 Voting content with Fivestar 546
Practical case 38.2 Voting contents with Rate 553
38.2. User scoring 558
38.3. Share in social media 563
38.4. Communication tools 568

Unit 39. Newsletter Subscription 575
39.1. Simplenews installation and Configuration 576
39.2. Subscribing to Newsletters 581
39.3. Create and Send Newsletters 588

Unit 40. Other Features 591
40.1. Content Presentation 592

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

40.2. Mega menus....................606
40.3. Scheduling events with Calendar....................610
40.4. Administration Modules....................613

Annex A. Configuration of the hosting and necessary tools 617
A.1. Hosting dashboard....................617
A.2. Managing databases with phpMyAdmin....................622
A.2. Upload and download files via FTP....................624
A.3. Uncompress files with 7zip....................628
A.4. Editing text files with Notepad++....................629

Annex B. Links and resources of interest 631
B.1. forcontu.com....................631
B.2. Official Drupal page, drupal.org....................632
B.3. drupalmodules.com....................632
B.4. Drupal Association....................633
B.5. Drupal Group on Google....................634
B.6. Translations of Drupal, localize.drupal.org....................634
B.7. Drupal API, api.drupal.org....................635
B.8. Drupal paid-for themes....................635
B.9. Generic texts for tests....................635

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

21 Creating forms with Webform

The Webform module allows us to create forms in Drupal in a quick and relatively straightforward way. No programming or HTML knowledge is required to use Webform.

We can use Webform to create contact or data submission forms, surveys, questionnaires, etc.

Administrators can view all the forms completed by visitors and site users and export the results to an Excel-compatible spread sheet or CSV for further analysis. Forms can also be configured so that both administrators and users receive an email of its contents once it has been filled in.

In this unit we will learn how to configure Webform and how to create forms with different sections (texts, option lists, dates, etc). Lastly we introduce some additional modules related to Webform that allow us to add more functions.

Comparative D7/D6

Webform versions 2 and 3

In this unit we study Webform version 3. We can not speak of significant differences between Drupal 6 and Drupal 7, but major changes between versions 2 and 3 of the module, both available in Drupal 6 and 7.

In the Drupal 6 course we studied Webform version 2, so in this unit we will see some important changes.

Unit contents

21.1 Installation and configuration of Webform 2
21.2 Form creation with Webform..5
21.3 Email configuration..21
21.4 Form settings..24
21.5 Results analysis..27
21.6 Permissions..28
21.7 Validation of form fields...29
21.8 Use of taxonomy in list selection..................................33

21

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

21.1 Installation and configuration of Webform

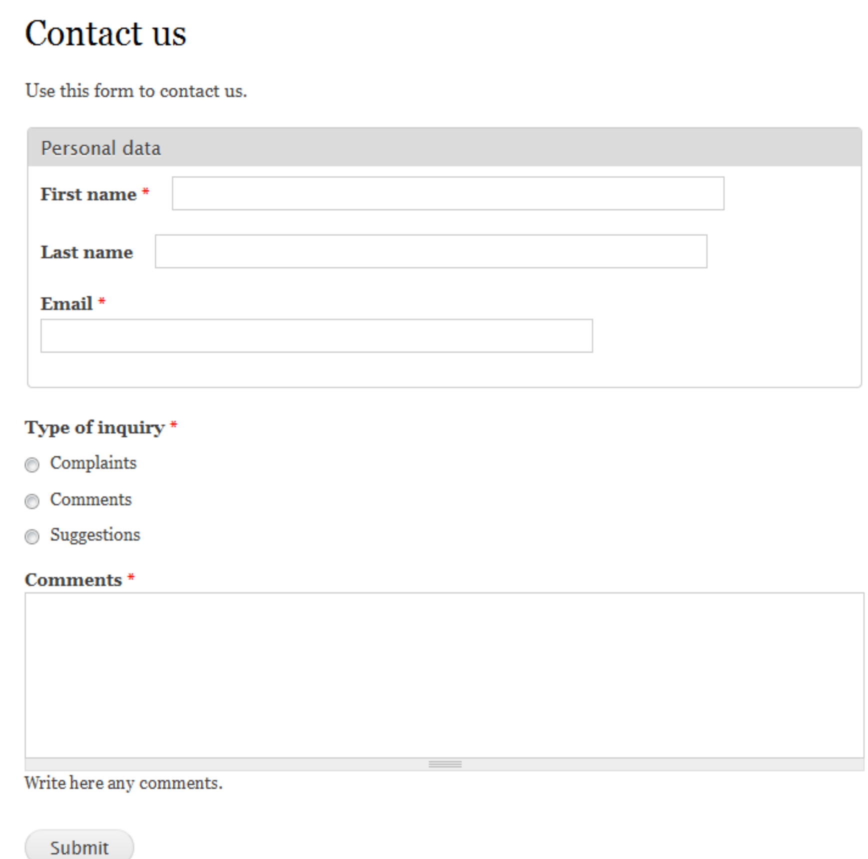
The **Webform** module is the ideal tool for the creation of all types of forms: contact forms, questionnaires, order forms, reservation forms, etc. This F21.1 **Figure** shows a typical contact form as an example of what can be created in Webform.

F21.1

Form created in Webform

Example form created in Webform.

Webform allows for the creation of forms with no programming knowledge.



The Webform module is available at:

http://drupal.org/project/webform

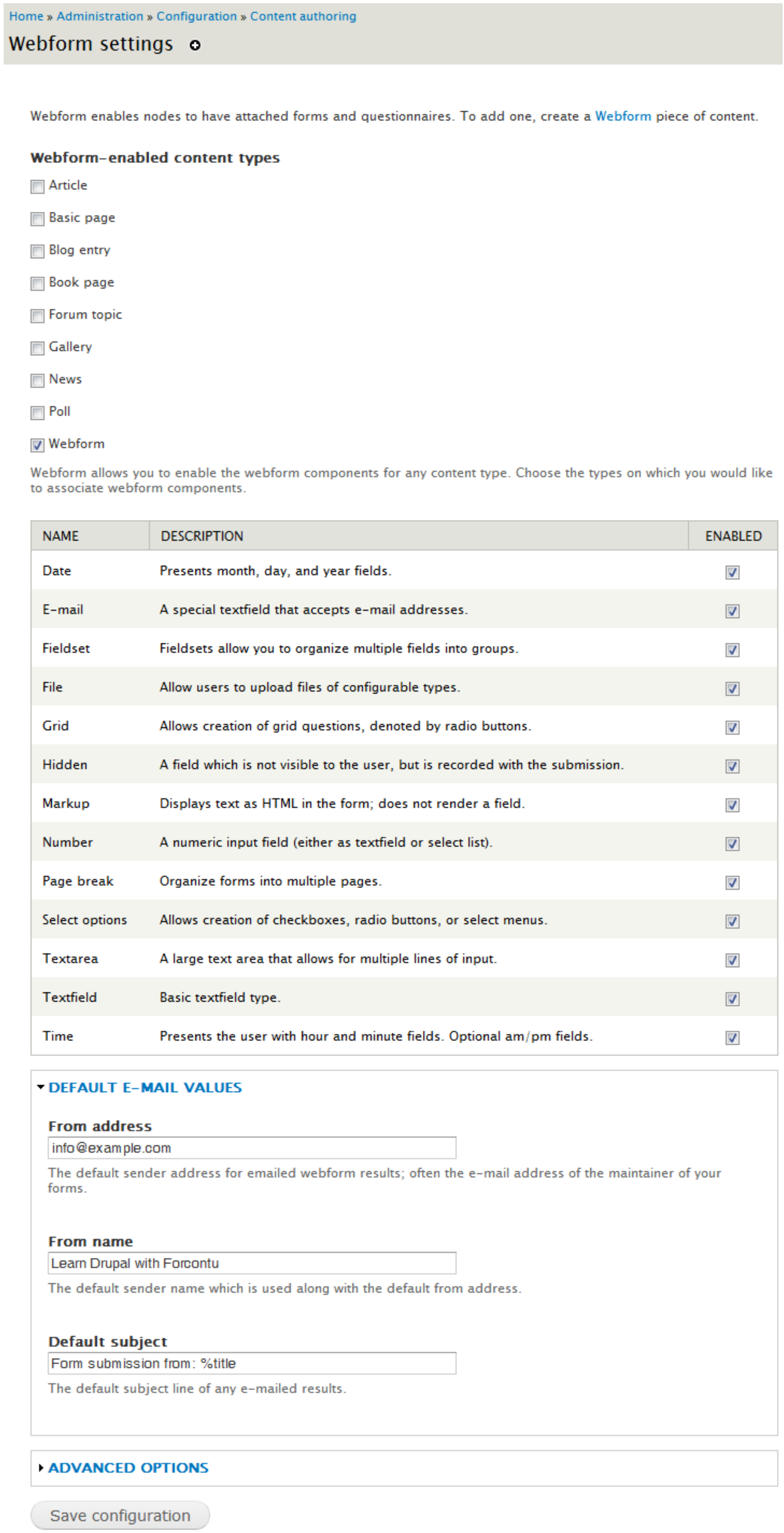
Once installed and activated, we configure the general settings of Webform by accessing:

URL Webform Configuration
/admin/config/content/webform

Administration ⇨ Configuration ⇨ Content authoring ⇨ Webform settings

Here we find the following options: F21.2

- **Webform-enabled content types**. For the content types shown here we activate the Webform form components. New content types do not need to be activated because the module creates the Webform type that is used to create forms.


Home » Administration » Configuration » Content authoring

Webform settings

Webform enables nodes to have attached forms and questionnaires. To add one, create a Webform piece of content.

Webform-enabled content types

- [ ] Article
- [ ] Basic page
- [ ] Blog entry
- [ ] Book page
- [ ] Forum topic
- [ ] Gallery
- [ ] News
- [ ] Poll
- [x] Webform

Webform allows you to enable the webform components for any content type. Choose the types on which you would like to associate webform components.

NAME	DESCRIPTION	ENABLED
Date	Presents month, day, and year fields.	☑
E-mail	A special textfield that accepts e-mail addresses.	☑
Fieldset	Fieldsets allow you to organize multiple fields into groups.	☑
File	Allow users to upload files of configurable types.	☑
Grid	Allows creation of grid questions, denoted by radio buttons.	☑
Hidden	A field which is not visible to the user, but is recorded with the submission.	☑
Markup	Displays text as HTML in the form; does not render a field.	☑
Number	A numeric input field (either as textfield or select list).	☑
Page break	Organize forms into multiple pages.	☑
Select options	Allows creation of checkboxes, radio buttons, or select menus.	☑
Textarea	A large text area that allows for multiple lines of input.	☑
Textfield	Basic textfield type.	☑
Time	Presents the user with hour and minute fields. Optional am/pm fields.	☑

▾ DEFAULT E-MAIL VALUES

From address
info@example.com
The default sender address for emailed webform results; often the e-mail address of the maintainer of your forms.

From name
Learn Drupal with Forcontu
The default sender name which is used along with the default from address.

Default subject
Form submission from: %title
The default subject line of any e-mailed results.

▸ ADVANCED OPTIONS

Save configuration

F21.2

Webform settings

From module configuration you can choose the content types on which you would like to associate webform components.

You can also choose which content types are available

Lastly you can set the default email values for sending emails related to the forms.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Webform components**. Here we see a list of the available form components. From this option we can deactivate components that we do not want available when creating forms. In the next section we find out what each of these components is used for.

- **Default e-mail values**. Allows us to define the predetermined format for sending emails related to the forms. Later, while creating forms, we can make specific changes to each one of them.

- **Advanced options**: F21.3

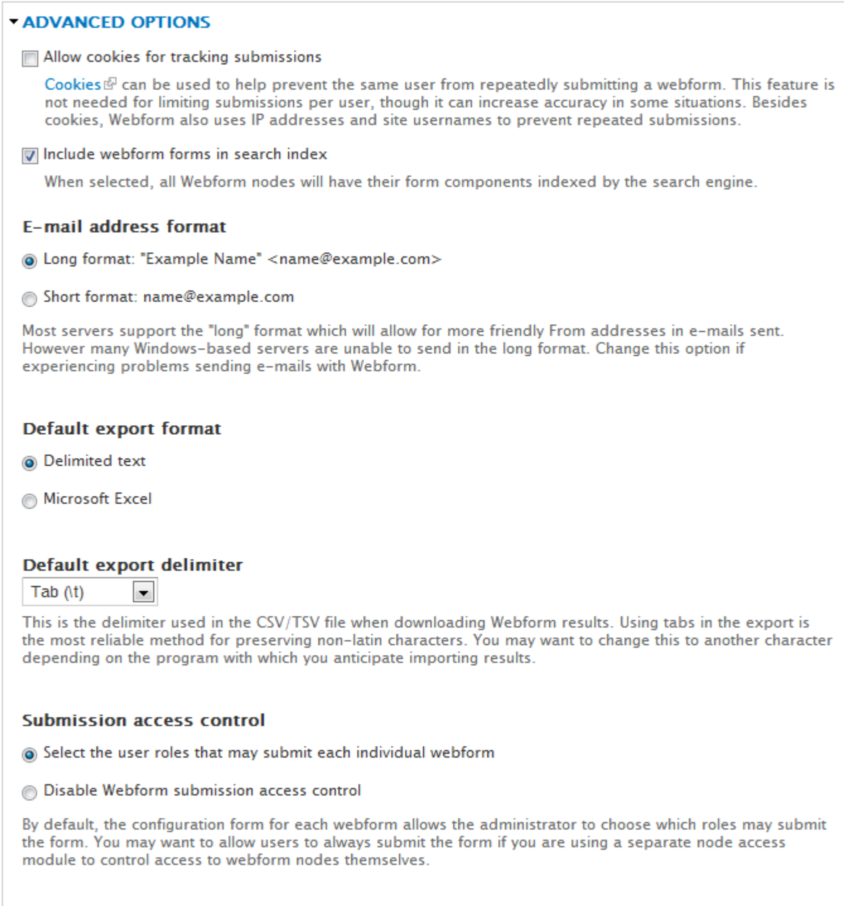
    - **Allow cookies for tracking submissions**. It allows us to use **cookies in order to** prevent a single user from submitting a form more than once. Webform also uses other methods to avoid repeated submissions, including username and IP address tracking.

    - **E-mail address format**. We can choose between **Long Format**, which includes a name, as well as an email address, and **Short Format**, with just an email address. The long format may not be compatible with all servers, especially those based on Windows.

    - **Default export format** and **Default export delimiter**. Both these options allow us to configure data export, which occurs in CSV format. The export delimiter is the character that separates the fields that make up the form, allowing the downloaded file to be easily read by other programs.

    - **Submission access control**. Allows the administrator to choose which roles may submit the form (default and recommended option). Each form displays which roles may submit the form.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079


▾ ADVANCED OPTIONS

☐ Allow cookies for tracking submissions

Cookies can be used to help prevent the same user from repeatedly submitting a webform. This feature is not needed for limiting submissions per user, though it can increase accuracy in some situations. Besides cookies, Webform also uses IP addresses and site usernames to prevent repeated submissions.

☑ Include webform forms in search index

When selected, all Webform nodes will have their form components indexed by the search engine.

E-mail address format

◉ Long format: "Example Name" <name@example.com>

○ Short format: name@example.com

Most servers support the "long" format which will allow for more friendly From addresses in e-mails sent. However many Windows-based servers are unable to send in the long format. Change this option if experiencing problems sending e-mails with Webform.

Default export format

◉ Delimited text

○ Microsoft Excel

Default export delimiter

Tab (\t)

This is the delimiter used in the CSV/TSV file when downloading Webform results. Using tabs in the export is the most reliable method for preserving non-latin characters. You may want to change this to another character depending on the program with which you anticipate importing results.

Submission access control

◉ Select the user roles that may submit each individual webform

○ Disable Webform submission access control

By default, the configuration form for each webform allows the administrator to choose which roles may submit the form. You may want to allow users to always submit the form if you are using a separate node access module to control access to webform nodes themselves.

F21.3

Advanced Options

In the Advanced Options tab you can set other Webform settings. For example, allowing cookies which prevent repeated form submission, choosing the format of the email address used in the email component and setting which roles may submit the form.

Form creation with Webform

21.2

Once we have configured the module we can create forms. The module adds Webform content types, accesed at: F21.4

Administration ⇨ Content ⇨ Add content ⇨ Webform

URL Create Webform
/node/add/webform

In the first stage of content creation only the Standard node components are available: title, body (displayed before the form components), text format and other option groups.

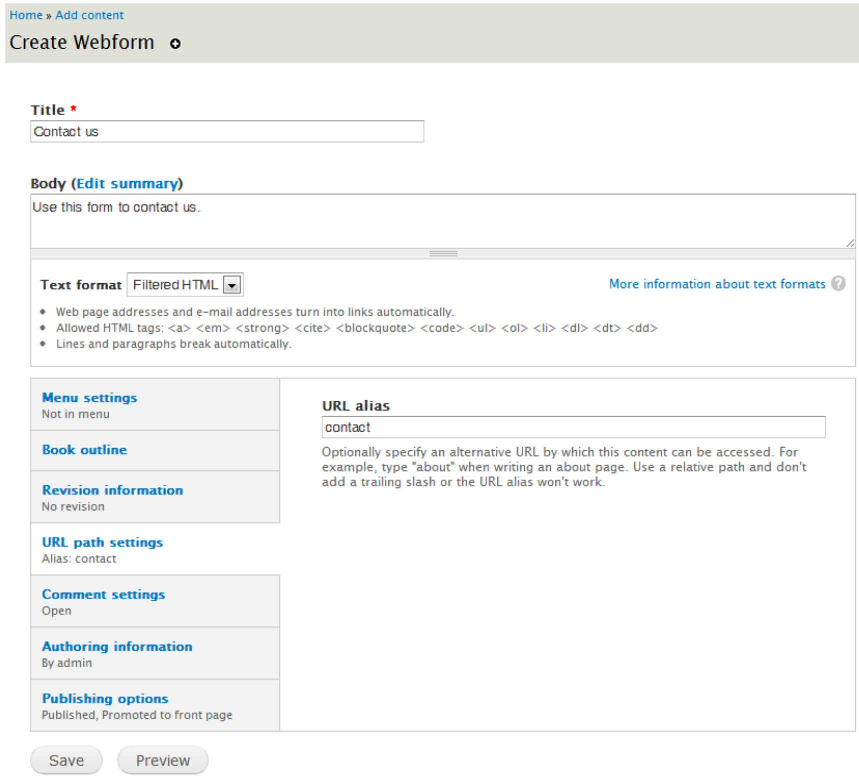
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F21.4

Create Webform

In the first stage of form creation you must choose the components shared by every node: title, body, text format, URL alias, etc.

The body is displayed before the form components: You can use it as an introduction or as an explanation of the purpose of the form.

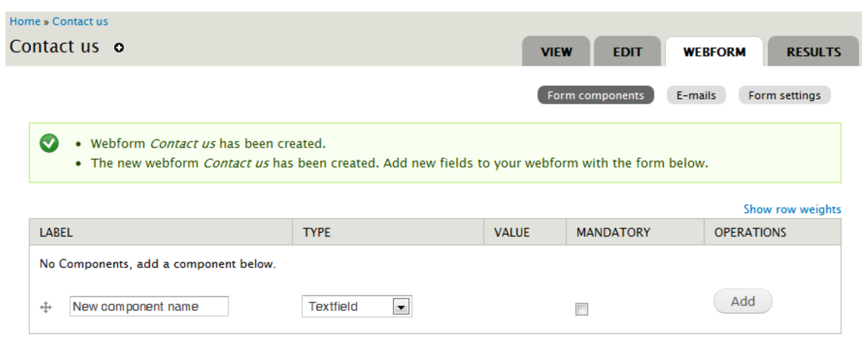


Once the node is set up, create and manage the form components from the Webform tab. **F21.5**

F21.5

Webform tab

From the Webform tab, configure the form and add components.



Within the **Webform** tab there are three sub-tabs:

- **Form components**. This tab is active by default. From here we can add, delete or edit form components.

- **E-mails**. Allows us to establish different email addresses for alerts or copies of submitted forms (for example we can send a copy of a form to the user that submits it, and/or to the site administrator).

- **Form settings.** Here we can specify configuration settings for the form. For example, closing down the form to prevent more submissions, or limiting submissions to certain roles.

In this section we continue to look at the form components, leaving the two sub-tabs until later sections.

Form components

Once we create the node, we add form components from the **Form components** tab. F21.5

We choose only the component types that we want to add, assigning them a name and clicking on the **Add** button. We can also indicate whether a component is **Mandatory,** meaning that the form cannot be submitted unless it is filled in.

In the next section we briefly describe the different types of form components available in the **Webform** module:

- **Date**. Displays day, month and year fields.

- **Email**. A text field that allows the users to type in their email address. This component has extra options that allow it to check whether an email address is valid.

- **Fieldset**. Allows several similar fields to be grouped together in a single block. For example, we can create a block of fields called *Personal Information* that groups together the *Name, Surname* and *Address* components.

- **File**. Attaches a file to the form.

- **Grid**. Allows the creation of tables with several questions that share the same set of answers. An example of this is the standard customer satisfaction survey where each question has a range of answers from 1 (very poor) to 5 (excellent). In the first column the questions are displayed, while the other columns contain all the possible answers (selected using **radio** buttons).

- **Hidden**. An element of the form invisible to users (and that cannot be modified by them) that is sent along with the rest of the form. Its purpose is to include further information in the form that is not visible to the submitter. For example, it can be used to store information such as an identification number (for example, a course or user number).

- **Markup**. Shows a block of text (formatted) along with the form. This is used to include additional content. As this is an informative component, it is not interactive.

- **Page break**. Allows the form to be divided over several pages or into stages.

- **Select options**. Use this to create lists of element selections, which can appear in the *checkbox* format (there are several options), as *radio buttons* (just one option), or as selection lists or *select* (single or multiple selection options).

- **Textarea**. Includes a text area of several lines long in the form.

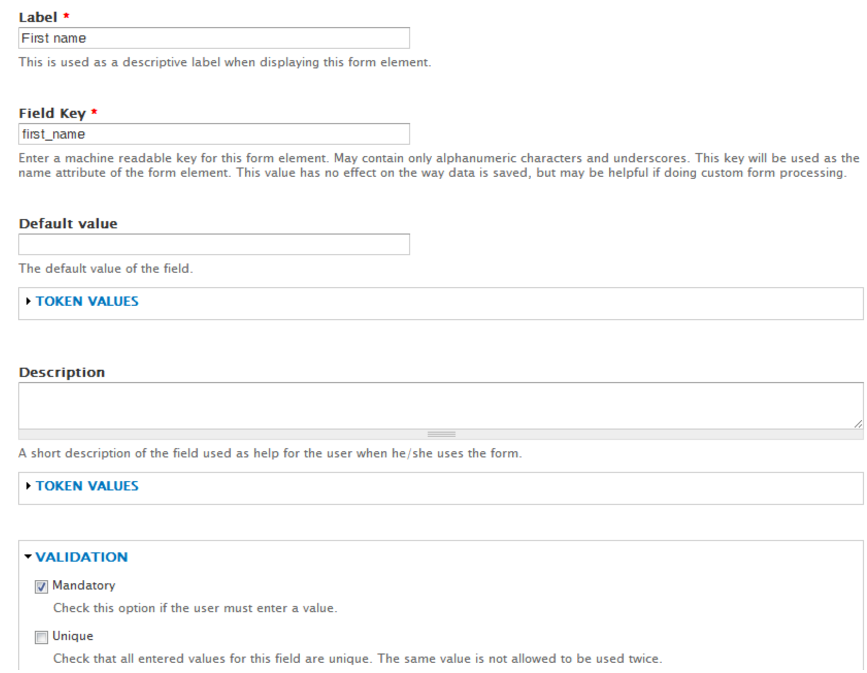
- **Textfield**. A single line text area.

- **Time**. Allows users to fill in the time in hours and minutes.

Once we click on the **Add** button, the field configuration options appear. These options vary depending on the component type.

Common options to all components

These are, in summary, the available options for each basic component. F21.6



F21.6

Common Options to all components

This figure shows some of the options common to almost all form component types.

- **Label**. The field name, seen by the user next to the field (such as *Name, Surname, Telephone*, etc.).

- **Field Key**. Name of the system or identifier used internally by Drupal. It identifies the field and must be an un-spaced text containing lower-case letters, numbers and underscores.

- **Default value**. Introduces a default value in a field which can be changed by the end user.

- **Description**. A field for additional information that explains to the user how to fill out the field.

- **Mandatory**. Shows whether a field is mandatory or not. A form cannot be sent if any of its mandatory fields are empty.

- **Unique**. Prevents identical replies in a field set, even if they are entered by different users. For example, when asking for email addresses, it can be set to stop the same address being used for two different form submissions.

Some components also include the <**Token> Values** field. If we open these sections we will see the tokens which are strings that, once entered in the fields, will be replaced by their corresponding values. For example, if we input the string "Welcome to %site" in the Description field, the users will see the token %site replaced by the site name (For example, "Welcome to Learn Drupal with Forcontu").

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

In the **Display** section we will find the following options:

- **Label display**. Defines where the main field text is displayed: Above, Inline (on the left) or None (no text displayed).

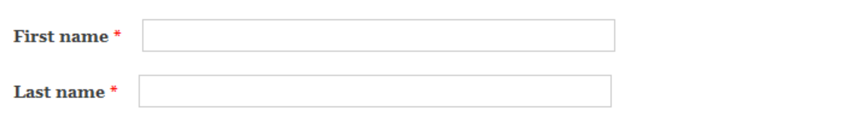
- **Disabled**. Allows a field to be blocked. The field is displayed but we can´t change its value. Useful for inputting default values that cannot be modified by the end user.

- **Private**. Private fields are only displayed to users with permission to see the submitted answers.

In the **Position** section we can group fields together in blocks (within a Fieldset component). We can also change the order of fields within a Fieldset.

Textfield component

The **Textfield** component allows simple text inputs that are one line long and unformatted. The **Figure** F21.7 shows two Textfield components as they are shown to the user when filling out the form. Fields marked with an asterisk (*) must be filled in (they have been tagged as **Mandatory**).



F21.7

Textfields

Examples of **Textfields** shown to the end user.

The specific configuration options of the Textfields F21.8 are:

- **Maxlength**. The maximum number of characters that can be inputted in the field.

- **Width**. Defines the width of the displayed field box (in number of characters).

- **Text label placed to the left of the textfield**. Allows a text to be displayed on the left of the field.

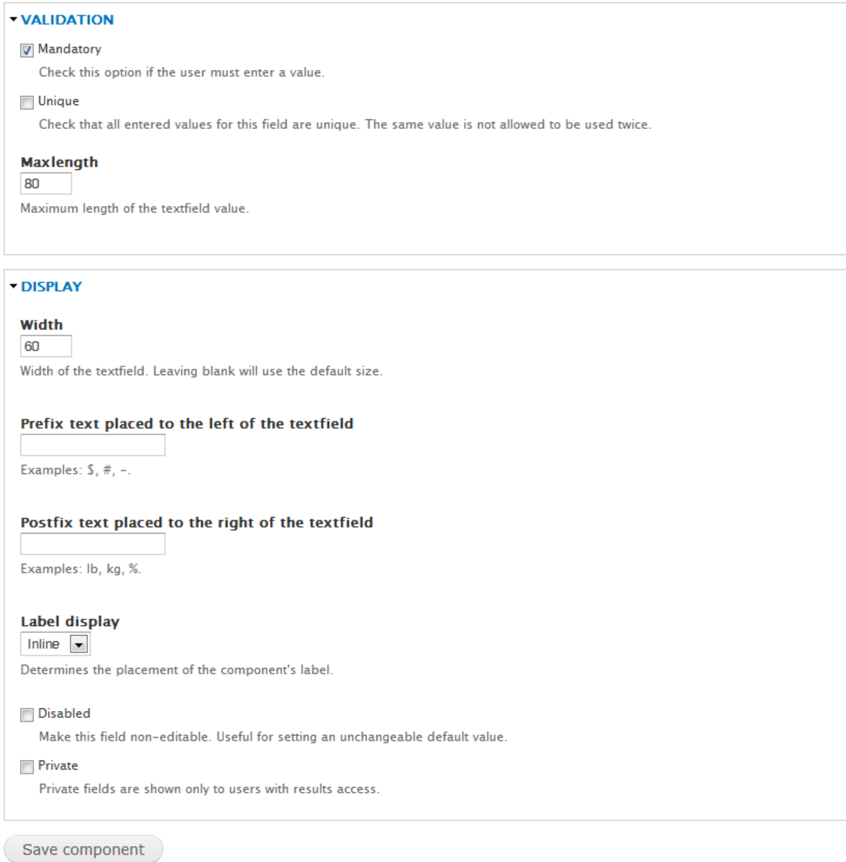
- **Text label placed to the right of the textfield**. Allows a text to be displayed on the right of the field. Used, for example, for units of currency, weight, measurements, etc.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F21.8

Textfield. Configuration

Specific configuration settings of the **Textfield** components.



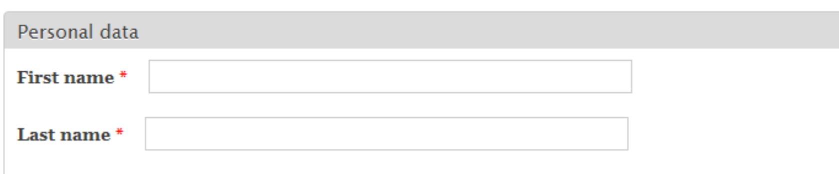
Fieldset component

The **Fieldset** component allows several fields to be grouped together in a single block (a *fieldset* in HTML). This is useful for grouping together related fields. In the **Figure** F21.9 we see the result of grouping the *Name* and *Surname* fields together in a *Personal Information* **Fieldset** block.

F21.9

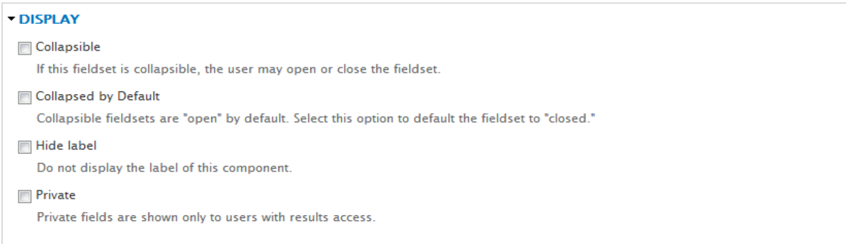
Fieldset

Example of the use of the Fieldset component to group together other components.



The configuration options for **Fieldset** components are as follows: F21.10

- **Collapsible**. Indicates whether a group of options can be collapsed or not.
- **Collapsed by Default**. If a group of options is collapsible this allows the group to be displayed collapsed by default.
- **Hide label**. If activated the field label will be not displayed.

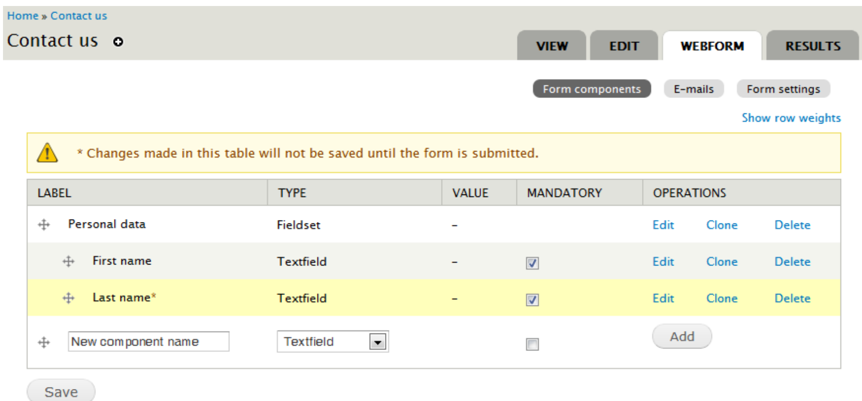


F21.10

Fieldset. Configuration

Configuration settings for a **Fieldset** component

Components grouped within a Fieldset are displayed within the **Form components** tab. There, we can reorder the fields by clicking on a component and dragging it to its final position, inside or outside the **Fieldset**. If we alter the order of components we must save the changes. F21.11



F21.11

Fieldset. Component order

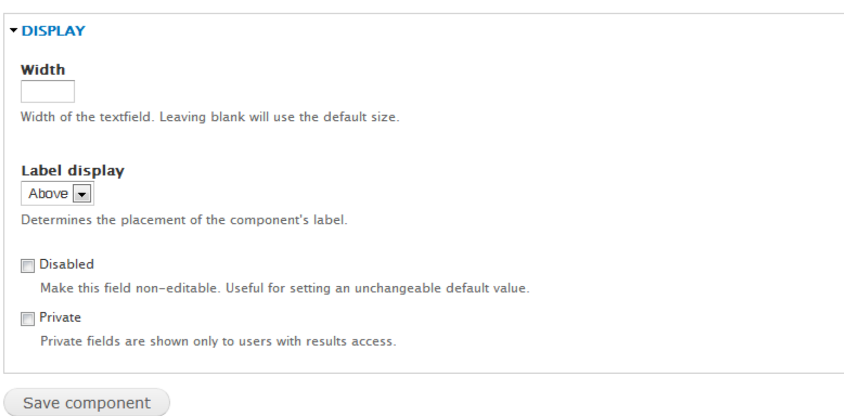
Components can be ordered or grouped together within a Fieldset by dragging and dropping.

E-mail component

The **E-mail** component is similar in appearance to the **Textfield** component. The main difference is that, on form submission, the inputted email address format is verified. The specific configuration options of the **E-mail** component are as follows: F21.12

- **User email as default**. Uses the email entered during registration for registered site users. The user can change this address when they fill in the form.

- **Width.** Defines the width of the field (the number of characters).



F21.12

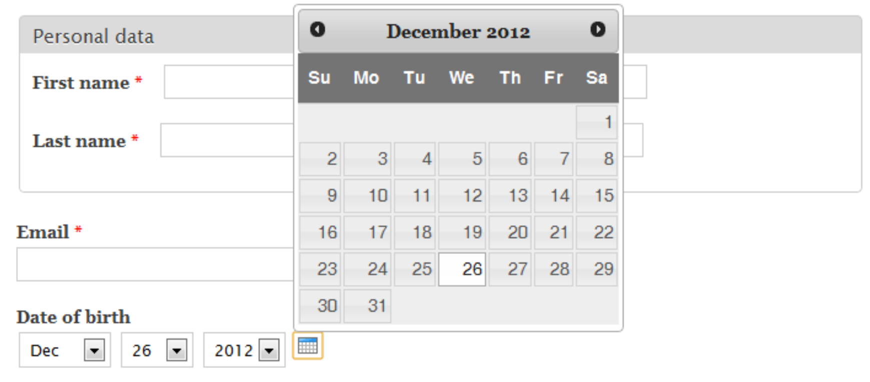
E-mail. Configuration

Configuration settings for the E-mail form component.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Date component

The **Date** component allows us to input a date in separate day, month and year fields. The user can choose values from the selection lists of the days, months and years, or from a calendar when it is available. F21.13



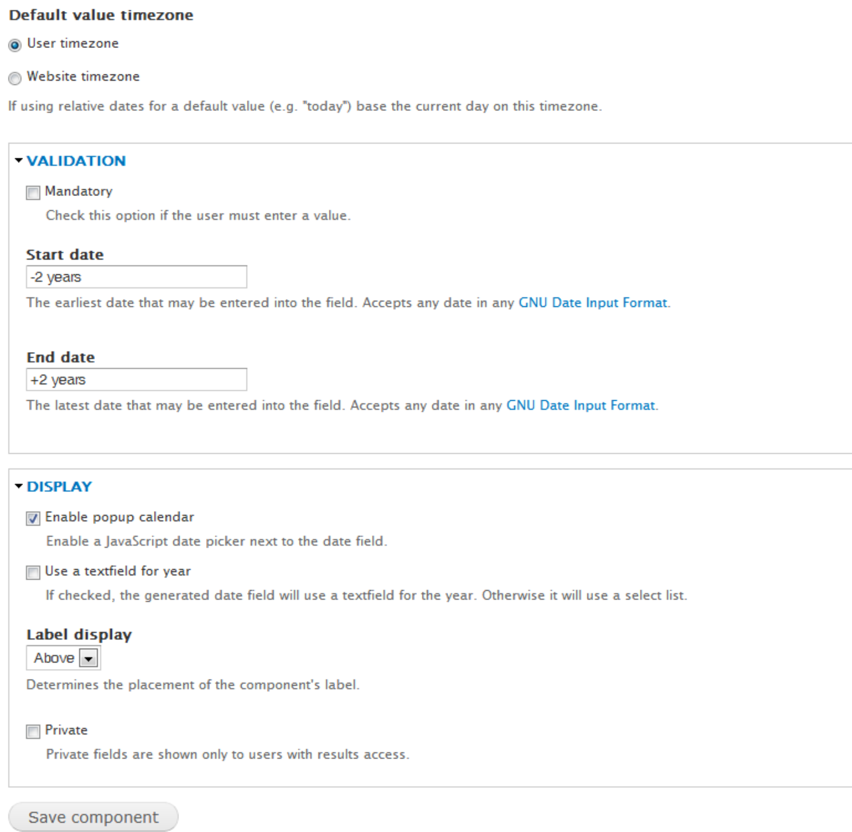
F21.13

Date

Example of a date component. It is possible to activate a date selection calendar in the Date components.

The specific options of the Date component are as follows: F21.14

- **Default value timezone**. Allows to introduce a predetermined time zone value, either the timezone specified by the user (in user profile) or the website's timezone
- **Start/end date**. Start and end year in the year selection field. Different formats are allowed, such as relative values to the current year (-2 years, +5 years)
- **Enable popup calendar**. If this option is checked, a button next to the field which allows to select the date directly from a pop-up calendar will appear.
- **Use a textfield for year**. If this is selected, the year field will appear as a manual fill in text field rather than as an option select list.



F21.14

Date. Configuration

Configuration settings for the Date form component.

Time component

The **Time** component displays the time, divided into hours and minutes. **F21.15**

F21.15

Time

Examples of text fields for the **Time** component, as displayed in the form.

The options specific to the **Time** component are as follows: **F21.16**

- **Default value timezone**. Allows a default timezone to be displayed in the field, either the timezone specified by the user (in their user profile) of the websites default timezone.

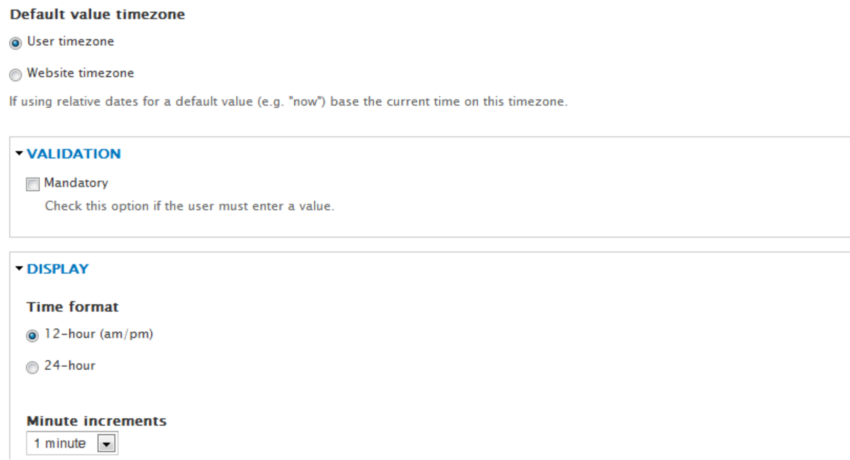
- **Time format**. Displays the time either in 12- or 24-hour format.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F21.16

Time. Configuration

Configuration settings for the **Time** form component.



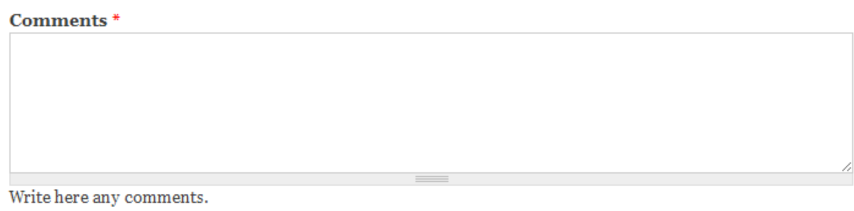
Textarea component

The **Textarea** component allows text entries that can be several lines long. F21.17

F21.17

Textarea

Example of a text field for the **Textarea** component, as displayed in the form.

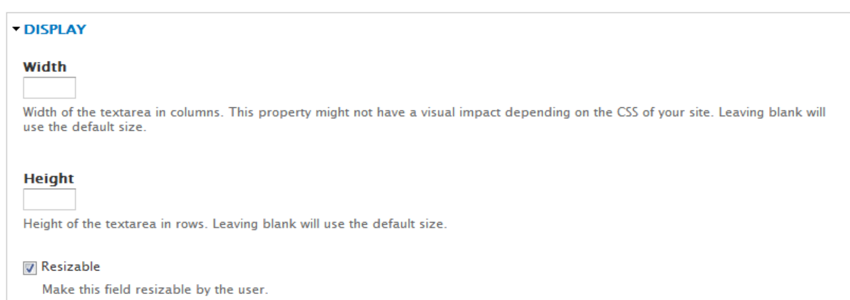


The specific options for the **Textarea** component are: F21.18

F21.18

Textarea. Configuration

Configuration settings for the **Textarea** form component.



- **Width**. Defines the width of the field (in number of characters)
- **Height**. Defines the number of lines in the text field.
- **Resizable**. If checked this option allows the user to expand or reduce the size of the text field.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Select options component

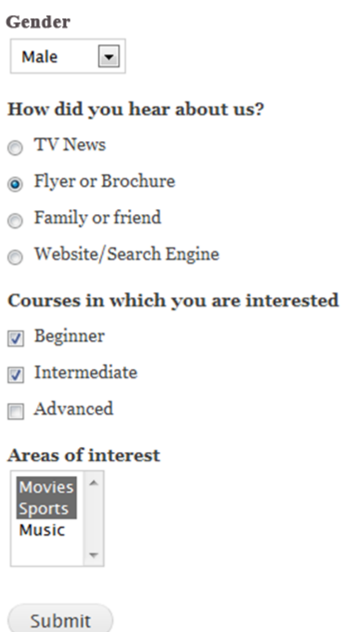
The **Select options** component let us to create element selection lists. It allows, depending on the configuration preset, four different types of selection lists: F21.19

- **Select list**. The elements are displayed as a collapsible list. The user can choose only one option (for example the *Gender* field).

- **Radio buttons**. Also a group of elements from which the user can choose only one. In this case the elements are displayed as radio buttons (in the example, the *How did you hear about us?* field).

- **Checkboxes**. The elements are displayed as checkboxes or selection boxes, with several options available (in the example, the *Courses in which you are interested* field).

- **Select list (multiple)**. A multiple selection list with several elements available for selection. In these lists the user must use the Control button to choose several elements (in the example, the *Areas of interest* field).



F21.19

Select options

Examples of lists created with the **Select options** component. The list types possible are:

- Select list (Gender)
- Radio buttons (How did you hear about us?)
- Checkboxes (Courses in which you are interested)
- Select list multiple (Areas of interest)

The following are the configuration options available, with explanations of how to configure specific options to get each type of list selection: F21.20

- **Multiple**. Allows the user to choose multiple options. If multiple is selected, the lists can be displayed as **checkboxes** and **select list (multiple)**. If unselected, the lists possible are **select list** and **radio buttons**.

- **Options**. All list elements can be displayed, one per line. The elements must be specified as key/value. The key part can only contain alphanumeric characters and underscores. For example:

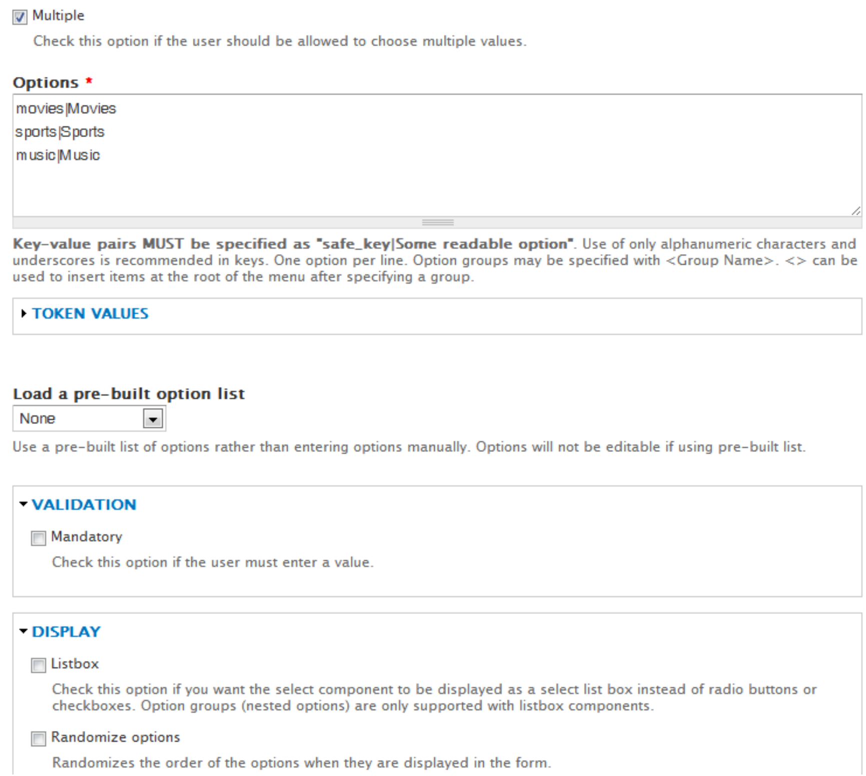
    - movies|Movies
    - sports|Sports

- music|Music

or

- 1|Movies
- 2|Sports
- 3|Music

This is useful when processing the information received from form submissions. Normally the internal values of the fields are processed (generally numeric) rather than the text displayed for the user.



F21.20

Select options. Configuration

Configuration settings for a **Select options** form component.

- **Load a pre-built option list.** Allows pre-built option lists to be loaded (days of the week, countries, etc.).

- **Listbox.** Shows that we want to display the elements as a list selection.

  Different combinations of the Multiple and Listbox fields allows us to generate the four list types mentioned above.

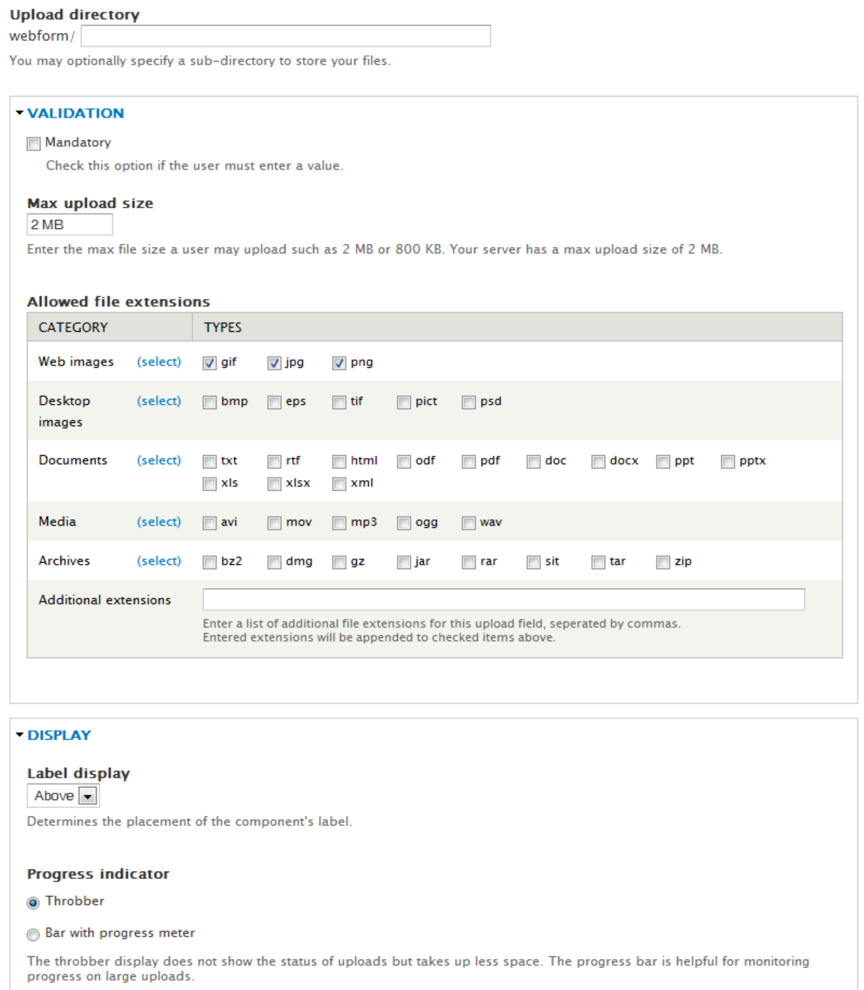
  - **Select list**. Multiple deactivated, Listbox activated.
  - **Radio buttons**. Multiple deactivated, Listbox deactivated.
  - **Checkboxes**. Multiple activated, Listbox deactivated.
  - **Select list (multiple)**. Multiple activated, Listbox activated.

- **Randomize options.** If activated, this option displays the options in a random order rather than in the order defined in the Options field.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

File component

The **File** component allows the end user to attach files to the form. The specific configuration options of the **File** component are as follows: F21.21



F21.21

File. Configuration

Configuration settings for a **File** form component.

- **Upload directory**. Files are saved by default in sites/default/files/webform, but a specific subdirectory can also be chosen.

- **Allowed file extensions**. Allows the type of accepted file extensions uploaded to the form to be defined.

- **Max upload size**. Indicates the maximum file size (in KB) that can be attached to the form.

- **Additional extensions**. Allows other extensions not included in the previous categories (separated by commas).

Each **File** component allows a single file to be attached to the form. For multiple file attachments we need to create multiple **File** components.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Markup component

The **Markup** component allows to include a formatted text in the form (plain text, HTML or even PHP logic). Remember that if we want to use PHP code, we must choose the PHP code text format. This is only available if the PHP Filter core module is activated.

The text will be displayed (or executed, in the case of PHP logic) where we choose in the form. It cannot be modified by the user who is filling out the form. The specific configuration options of the **Markup** component are as follows: F21.22

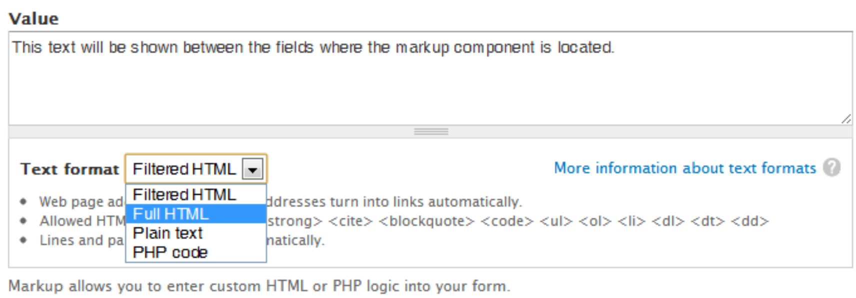
- **Value**. Field content. Can be HTML and/or PHP, depending on the text format selected. Tokens can be used in this field.

- **Text format**. Here we can choose any of the text formats available on the site.

F21.22

Markup. Configuration

Configuration settings for the **Markup** form component.



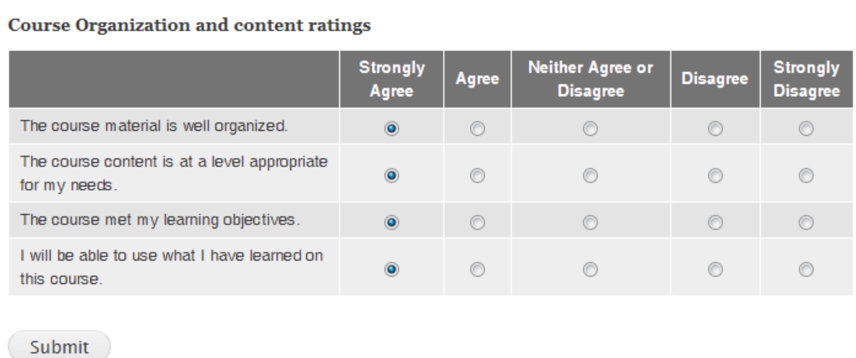
Grid component

The **Grid** component allows the creation of tables containing various questions which share the same possible answers. A typical example is a customer satisfaction survey, where each question can be answered by selecting a value between 1 (very poor) and 5 (excellent). In the first column the questions are displayed, while the rest of the columns display all the potential answers (as radio buttons). F21.23

F21.23

Grid

Examples questions in **Grid** mode.

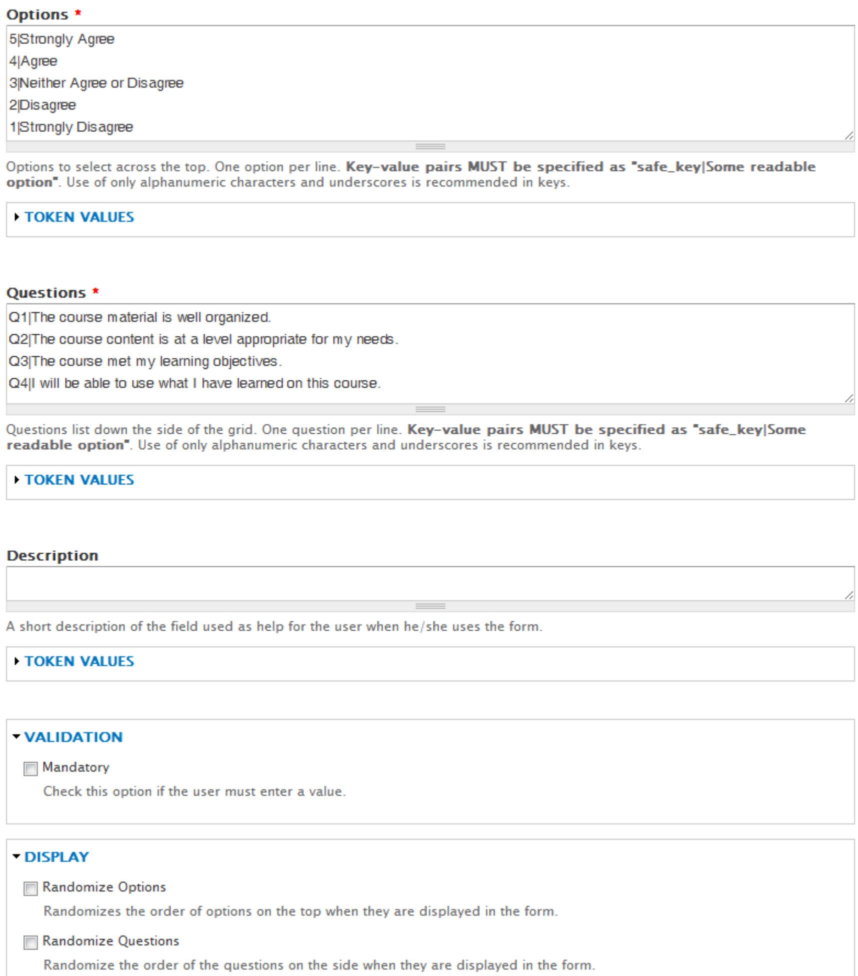


The specific configuration options of the **Grid component** are: F21.24

- **Options**. We write here the answers or values that the form user must choose between. The elements must be specified as key/value pairs. In the key, only alphanumeric characters and underscores are allowed.

- **Questions**. We write here the questions that are displayed in the column and which share the answers written in Options. The elements must be specified as key/value pairs. In the key only alphanumeric characters and underscores are permitted.

- **Randomize Options and/or Questions**. When activated, it displays the elements in a random order.



F21.24

Grid. Configuration

Configuration settings for the **Grid** form component.

Hidden component

The **Hidden** component allows to send hidden information with the form. For example, the username of the submitter, the IP, the date, etc. These values can be created by the tokens or fixed values included in the field by default.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Page break component

The **Page break** component allows to divide a form into several pages or stages. It has no specific configuration options and is simply inserted in the form at the point where a page break is required.

Once a page break is inserted the form is divided into separate stages. For the user to continue filling out the form they must click on the **Next Page** button. On the last page the user must click on the **Submit** button to finish off and submit the form.

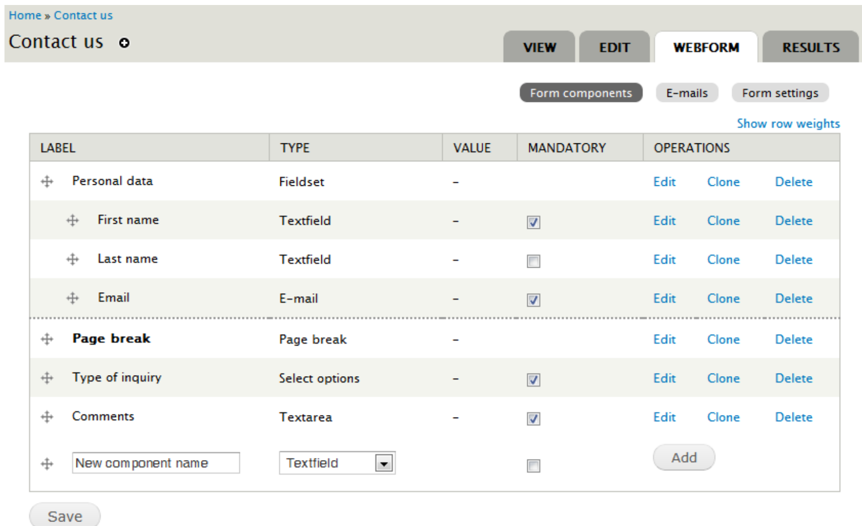
Page breaks are displayed in the form's component list as a dashed line. F21.25

F21.25

Example of a Page break component

You can divide a form across several pages, especially useful for long format forms.

The **Page break** is highlighted as a discontinuous line in the form component list.

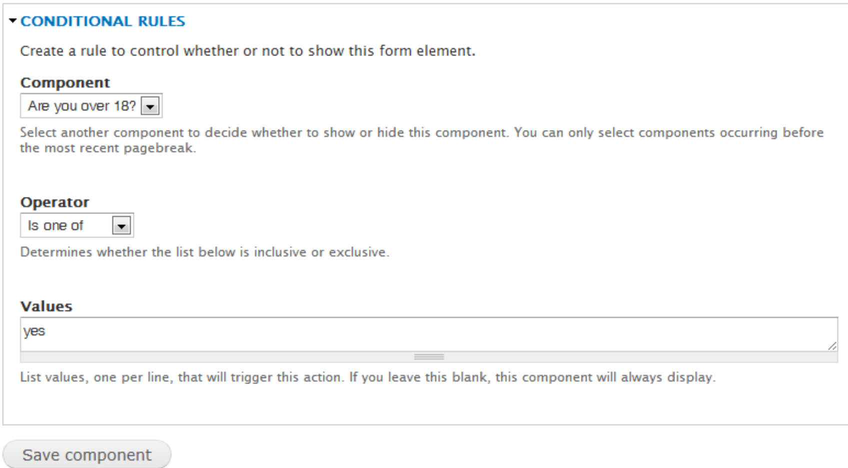


Conditional rules

The **Page break** component is particularly useful in the case of long format forms and we want to group fields together on separate pages in order to make filling out the form easier for the user.

Webform also includes an additional functionality that is only available when we use page breaks. When dividing a form we can use **conditional rules** to display or hide certain fields, based on the values inputted by the user in the previous steps. F21.26

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



F21.26

Conditional rules example

The conditional rules are only available for forms divided into several pages.

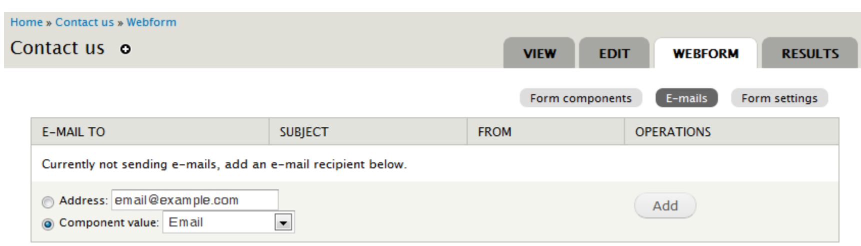
The field in which the conditional rule is displayed only appears if the user has replied "Yes" to the "Are you over 18?" question in the previous step.

Email configuration

21.3

Webform allows a very flexible configuration for sending email notifications when the user submits a form. For example, we can send an email to the submitting user thanking them for their interest and showing them a copy of the submitted form. At the same time we can send an email to the administrator telling them that a user has submitted a form.

Once we create a form we can configure email options from the E-mails sub-tab (Within the **Webform** tab). **F21.27**



F21.27

E-mails tab

You can program different email alerts, which are sent when a user submits a form.

For adding instructions for sending an email we can specify in this screen a specific email address or a component´s value. In this example **Component value: Email** has been selected. This means that an email is sent to the address specified by the user in the *Email* field.

Clicking on **Add** takes us to the email send configuration: **F21.28**

- **E-mail to address**. Choosing **Custom** allows us to pick any email address while the **Component** option allows us to send an email to the address specified by the user in one of the email (or text) components of the form.

- **E-mail subject**. This can either be a default value, personalized, or be input automatically from any of the form components.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **E-mail from address**. This option includes the email address that the receiver sees as the sender's address. It can also be a default value, be personalized, or filled in automatically from a form component.

- **E-mail from name**. This is the name associated with the sender that is displayed to the receiver. It can also be a default value, personalized, or filled in automatically from a form component.

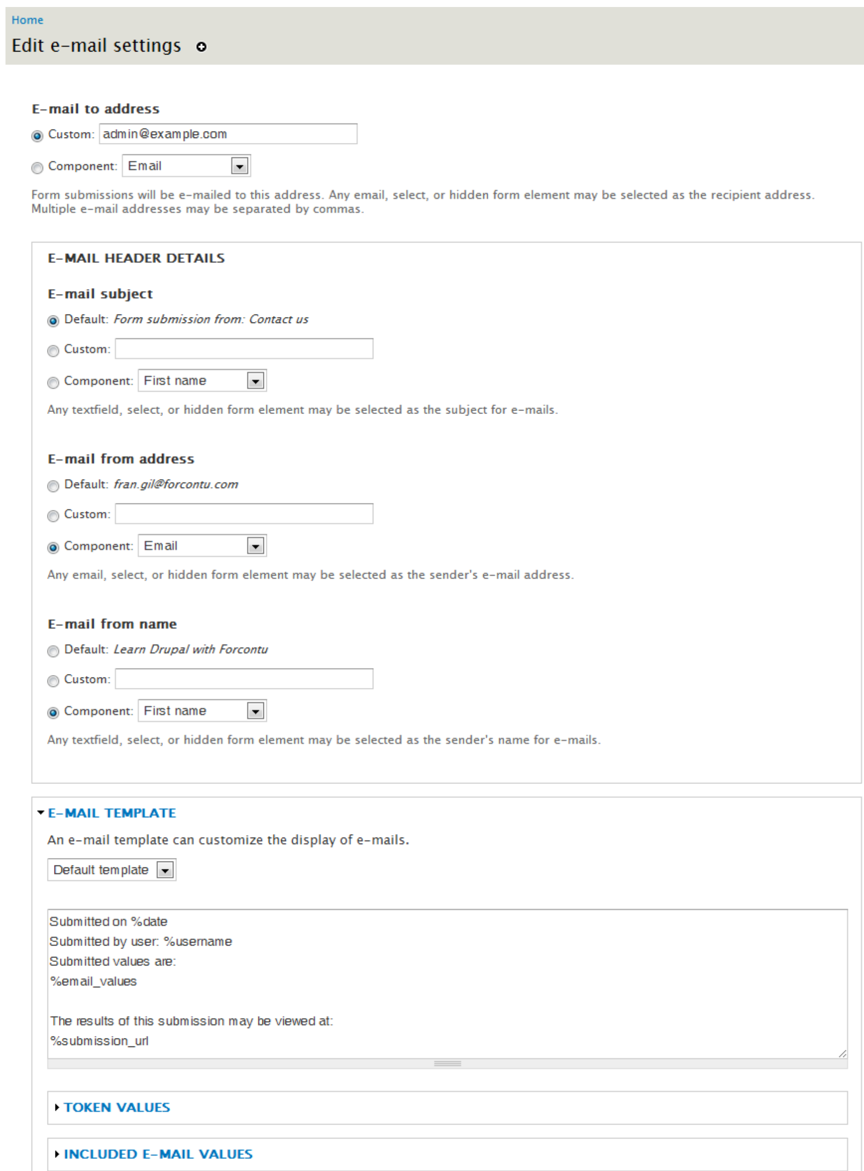
- **E-mail template**. Here the choice is between the default template and a personalized option. The sent message can be modified and available Tokens used.

F21.28

Configuration of email

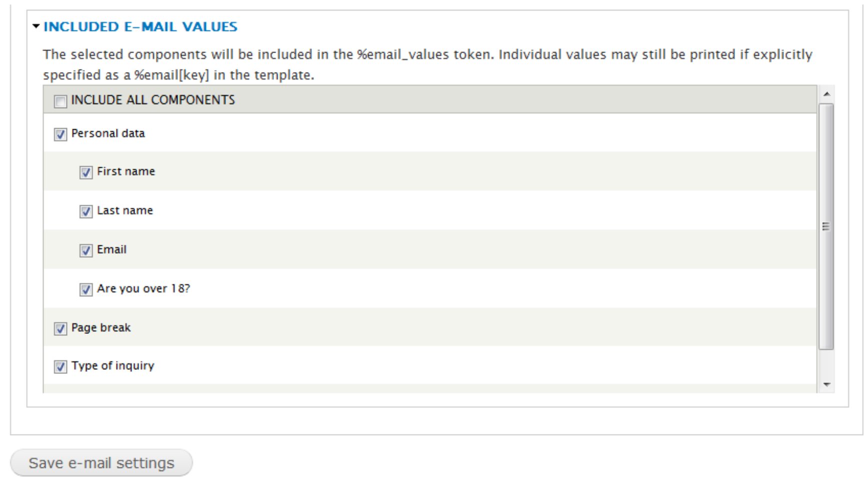
Configuration options for the sending emails.

Use fixed or dynamic settings, making use of the values inputted by the form user (email address, name, etc.)



- **Included e-mail values**. In the email template that is sent, the **%email_values** token in available for its use. It contains all the values entered into the form.

This section allows us to define which fields are displayed in the email when using the **%email_values** token. We can choose to display all values by checking the **Include all components** option, or choose individual components to display. For example, in an email reply to a user we should at least exclude the Hidden components. F21.29



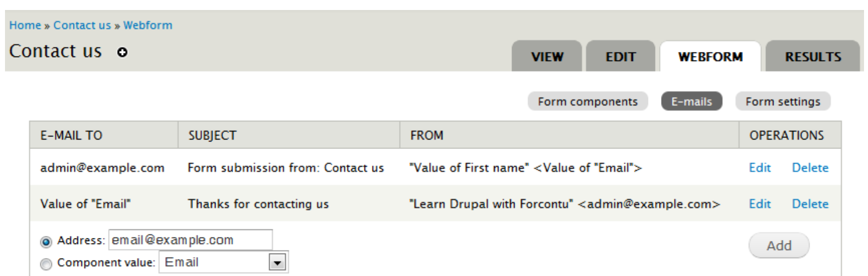
F21.29

Configuration of email sending

In Included Email Values you define which form components are displayed in emails.

These values are only available in the email template when the %email_values token is enabled (by default).

Once we click on **Save e-mail settings** we move back to the **E-mails** tab where a list appears of all programmed email sends. By following the same procedure we can configure other email alerts. These emails are sent to their addressee every time a user submits a completed form. F21.30



F21.30

E-mails

Displays the email addresses selected for the form.

Addressees receive emails when a form is submitted.

21.4 Form settings

From the **Form settings** sub-tab we can configure other options related to form submissions.

At this point, it is very important to check whether the selected options are compatible with the desired behaviour of the form because/since in this section we configure important factors such as which users have permission to submit the form.

The available options are: F21.31

- **Confirmation message**. The message displayed to the user once they submit the form.

- **Redirection location**. We can choose between three options that take place once the form has been submitted. Confirmation message: the user sees a page displaying the message inputted in the previous section. Using Personalized URL we can select a URL to which the user is redirected after submission. Using No redirect the user sees the refreshed form page after submitting the form.

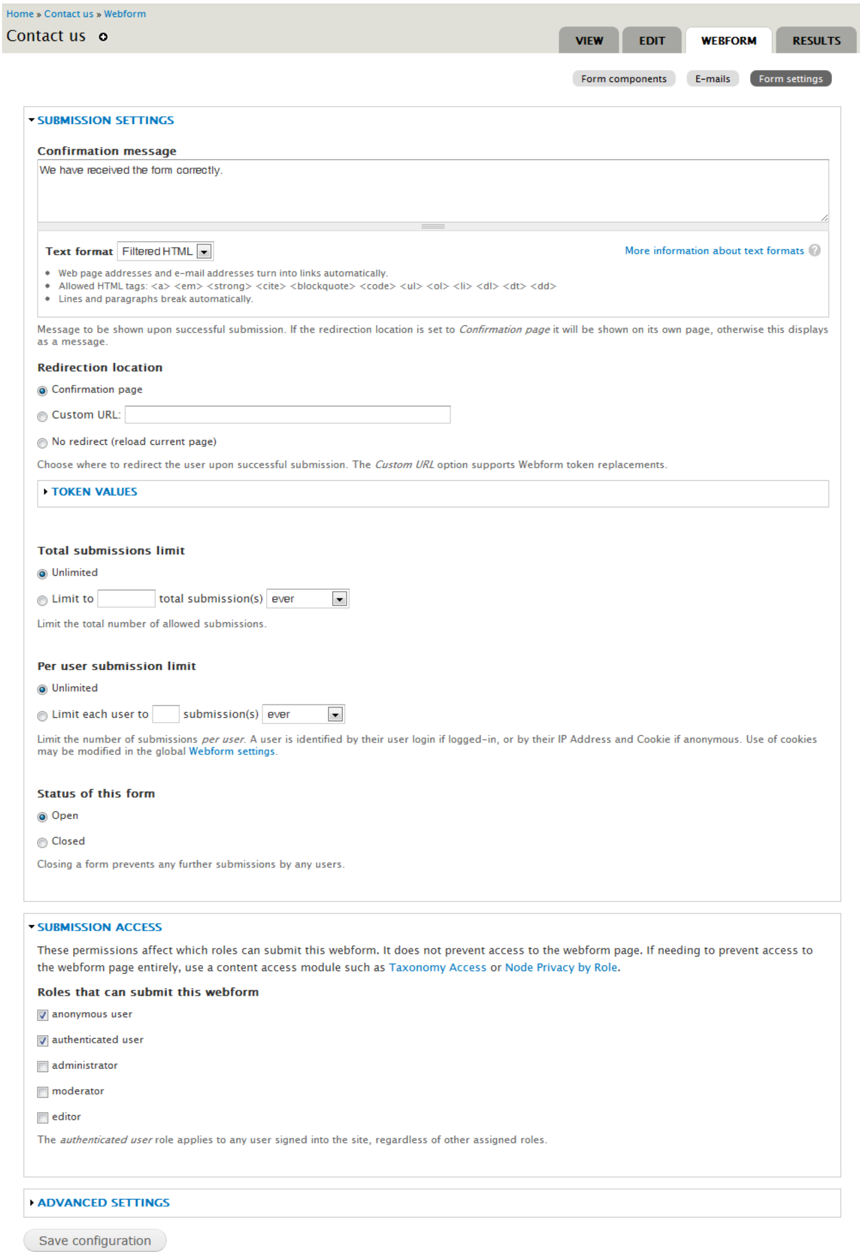
- **Submissions limit**. Defines the number of times a specific user can submit the same form. The limit can be temporal (for example, each user can only submit the form once per day). To control individual users Webform uses user registration data as well as the IP and cookies (if they are enabled in the global module configuration).

- **Status of this form**. This can be open or closed. A closed form is displayed to the user but cannot be submitted.

- **Roles that can submit this webform**. Indicates the roles needed by a user to be able to submit the form.

  Bear in mind that this permission only prevents form submission but not form display. This means that all users are able to view and fill out the form. If a user lacks the required roles, as selected here, they see an error message and the form won´t be submitted.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



F21.31

Form settings

It allows you to adjust other specific form settings (confirmation message, total submissions limit, status of this form, which roles can submit the webform, etc.)

- **Advanced settings**: **F21.32**

  - **Available as block**. Activating this option creates a block that contains the form. This allows the form to be placed anywhere in a website.

  - **Show complete form in teaser**. Normally a teaser node view displays just a portion of its content. When the node is a form we may not want to display it partially. Check this box to display the entire form even if it is being displayed as a *teaser*.

  - **Show "Save draft" button**. Allows users to save a draft of a form as they fill it out. The user can then return to the form later on to complete it and submit. This option is very useful for

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

long format forms that require extensive time to complete and submit. (for example a survey that is several pages long).

- **Automatically save as draft between pages**. Check this to automatically save a draft every time a user moves from one page of the form to the next.

- **Show the previous submissions notifications**. If checked, it displays a reminder to users who have already filled in the form. This option does not prevent a user from resubmitting the form. Resubmissions are controlled via the **Submissions limit** option.

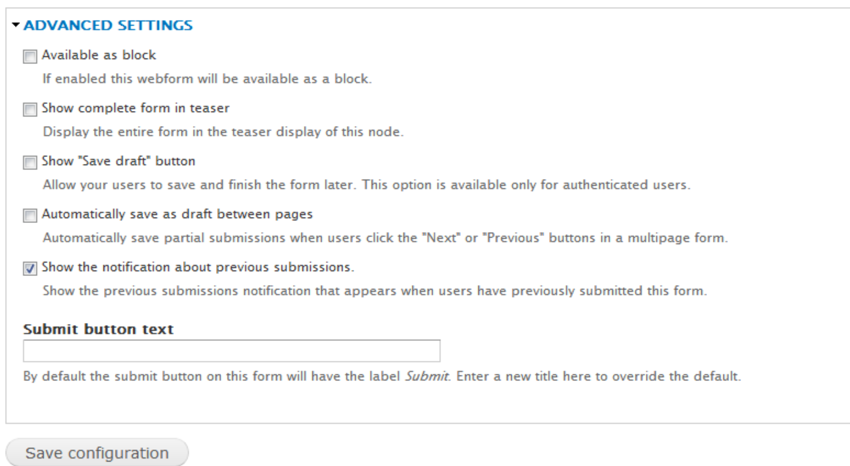
- **Submit button text**. By default the form submission button displays the text *Submit.* To change this, specify an alternative text (for example; Send submission, Request registration, etc.)

F21.32

Advanced settings

Advanced form configuration options.

Among other options this allows you to enable drafts so that a user can save an incomplete form.



Results analysis

21.5

Webform stores all user submitted forms in its database. To check them, access the **Results** tab found in each Webform node (provided that your username has the required permissions).

Within **Results** are the following additional tabs:

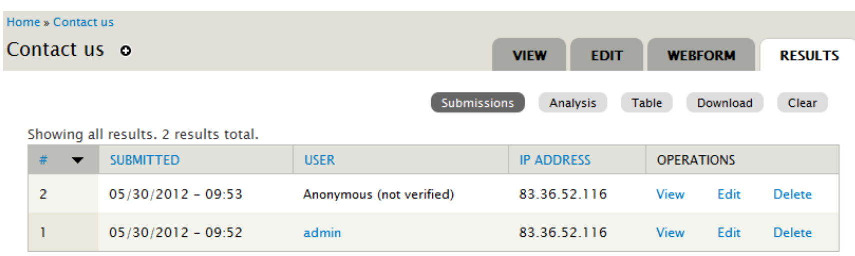
- **Submissions.** Displays all the user´s submissions, showing the date and time, username and IP address. From this list we can view, edit and delete every submission. F21.33

- **Analysis.** Displays some statistics obtained from the submitted forms including useful information such as the number of users that have left blank fields, or the number of users that selected a set option, such as a list type field. F21.34

- **Table.** Displays all the values inputted into submitted forms. Allows a quick visualization of the inputted information, even if the display table is difficult to interpret when we have a large number of form submissions.

- **Download**. Allows us to download a CSV format file of the submitted forms, letting us work on the collected information using a third-party program (such as Open Office's Calc or Microsoft Office´s Excel).

- **Clear**. This option allows us to clear all submitted forms with a single action.



F21.33

Submissions

Displays all the forms submitted by users.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F21.34

Submissions analysis

Statistical analysis of each component using the values inputted by users.

Home » Contact us » Results

Contact us

VIEW | EDIT | WEBFORM | RESULTS

Submissions | Analysis | Table | Download | Clear

Q	RESPONSES	
1	**First name**	
	Left Blank	0
	User entered value	2
	Average submission length in words (ex blanks)	2.00
2	**Last name**	
	Left Blank	0
	User entered value	2
	Average submission length in words (ex blanks)	1.50
3	**Email**	
	Left Blank	0
	User entered value	2
	Average submission length in words (ex blanks)	3.50

21.6 Permissions

Webform adds the following permissions: F21.35

F21.35

Permissions added by the Webform modules

Permissions available for the configuration of functionalities within the Webform module.

PERMISSION	ANONYMOUS USER	AUTHENTICATED USER	ADMINISTRATOR
Webform			
Access all webform results Grants access to the "Results" tab on all webform content. Generally an administrative permission.	☐	☐	☑
Access own webform results Grants access to the "Results" tab to the author of webform content they have created.	☐	☐	☑
Edit all webform submissions Allows editing of any webform submission by any user. Generally an administrative permission.	☐	☐	☑
Delete all webform submissions Allows deleting of any webform submission by any user. Generally an administrative permission.	☐	☐	☑
Access own webform submissions	☐	☐	☑
Edit own webform submissions	☐	☐	☑
Delete own webform submissions	☐	☐	☑

- **Access all webform results**. This permission allows access to the **Results** tab to all Webform nodes.
- **Access own webform results**. The user can only access the **Results** tab of forms that they create.
- **Edit all webform submissions**. Allows any user to edit all webform submissions.
- **Delete all webform submissions**. Allows any webform submission to be deleted.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Access own webform submissions**. Registered users can view their own webform submissions.
- **Edit own webform submissions**. Registered users can edit their own webform submissions.
- **Delete own webform submissions**. Registered users can delete their own webform submissions.

Upon creating new Webform content, the corresponding permissions from the **Node** module are also applicable. F21.36

PERMISSION	ANONYMOUS USER	AUTHENTICATED USER	ADMINISTRATOR
Webform: Create new content	☐	☐	☑
Webform: Edit own content	☐	☐	☑
Webform: Edit any content	☐	☐	☑
Webform: Delete own content	☐	☐	☑
Webform: Delete any content	☐	☐	☑

F21.36

Permissions of Node

Node permissions related to Webform content.

Validation of form fields

21.7

As well as the required field validations, Webform includes some basic validations, such as checking whether a name or email address is correctly formatted.

The **Webform Validation** module allows additional validations to be incorporated into the form fields.

The module is available at:

http://drupal.org/project/webform_validation

Once installed and activated the module adds an additional sub-tab, **Form validation**, within the **Webform** tab of every Webform node.

To add a validation rule simple click on the available rules and configure them, choosing all the fields it will change. A rule can be applied to several fields.

The available rules are as follows: F21.37

- **Numeric values**. An inputted value must be numerical. We can also opt to define the range of permitted values.
- **Minimum length**. The minimum number of characters that a field must contain.
- **Maximum length**. The maximum number of characters that a field can contain.
- **Minimum number of words**. The minimum number of words that a field must contain.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Maximum number of words**. The maximum number of characters that a field can contain.

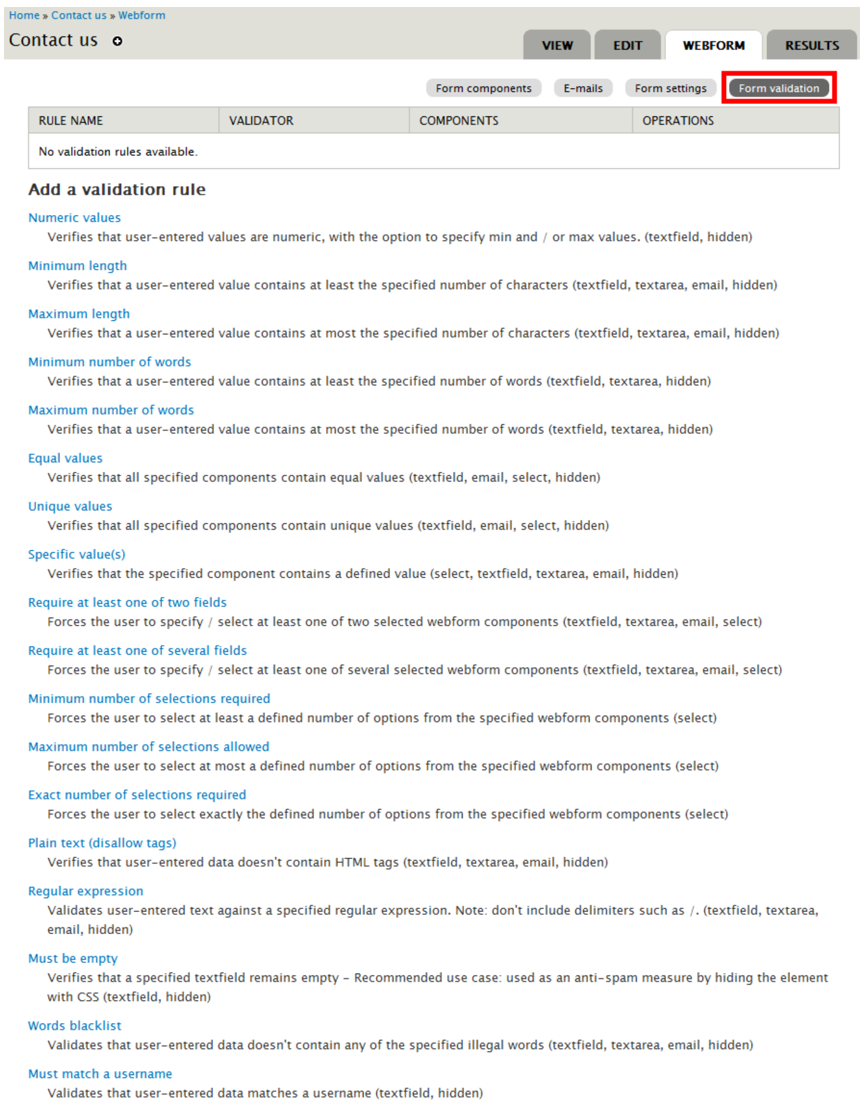
- **Equal values**. Checks that the values inputted into different fields are identical. For example when we want a user to repeat their email address or password.

- **Unique values**. Makes sure that selected components do not contain repeated values.

F21.37

Form validation

From the new sub-tab you can create validations for form components.



- **Specific value(s)**. Checks whether a selected component contains a specific value.

- **Require at least one of two fields**. The user must choose at least one of the two available components covered by the rule.

- **Require at least one of several fields**. The user must choose at least one of the available components covered by the rule.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Minimum number of selections required**. In selection lists we can indicate the minimum number of elements a user must select.

- **Maximum number of selections allowed**. In selection lists we can indicate the maximum number of elements a user can select.

- **Exact number of selections required**. In selection lists we can indicate the exact number of elements a user must select.

- **Plain text (disallow tags)**. Verifies that the inputted text does not contain HTML tags.

- **Regular expression**. Allows a field to be validated using regular expressions.

- **Must be empty**. Verifies that a field is empty.

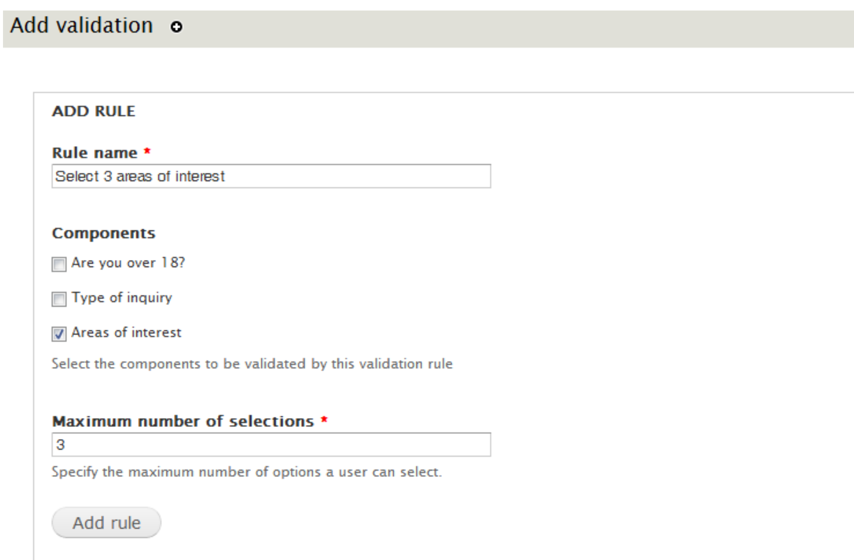
- **Words blacklist**. Verifies that a field contains no blacklisted words.

- **Must match a username**. Checks whether an inputted value matches a site username.

Creating three example rules:

- **Select 3 areas of interest**. We will use the component Description field to indicate that the user has to select 3 elements from the multiple choice list Areas of interest.
  From Form validation choose a rule **Maximum number of selections allowed**.

  In rule configuration select the **Areas of interest** component, and in **Number of selections**, select 3. F21.38



F21.38
Rule validation example

In these rules we indicate that in the component Areas of interest, (multiple-choice list) the user must select a maximum of 3 options.

We used the rule Maximum number of allowed rule.

- **Valid number of children**. The F21.39 number of children is an open text field so we must make sure that the inputted value is valid. Using the **Numeric values** rule, select the corresponding component and specify the range **0|10**. This means that the inputted value can only be between 0 and 10, both inclusive.

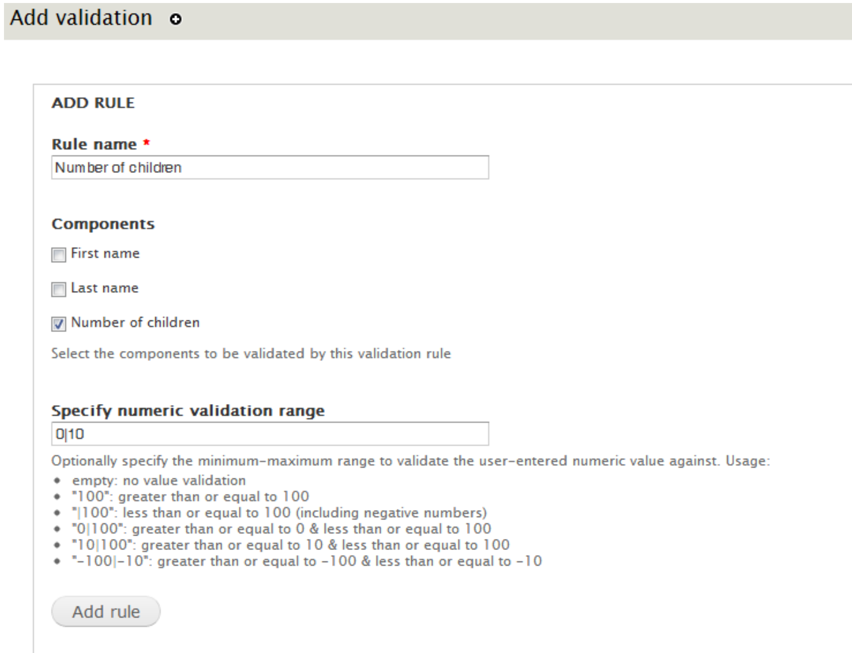
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F21.39

Example of validation rule

In this rule we indicate that the value introduced in the text component Number of children must be numerical and between 0 and 10 inclusive.

We used Numeric values rule.

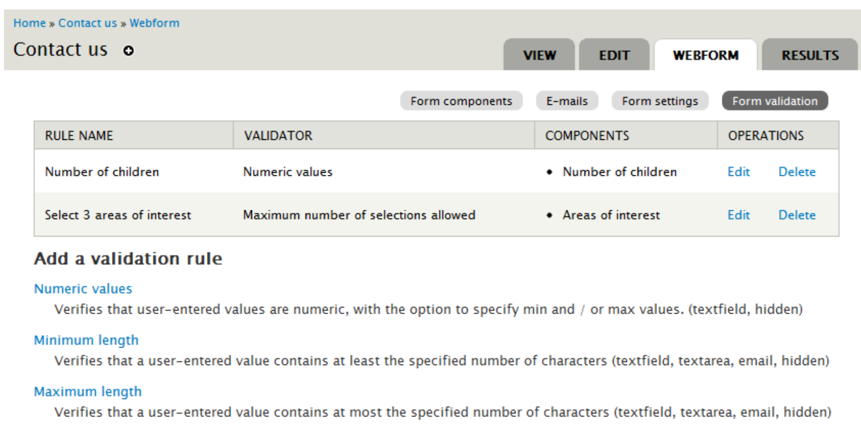


The list of added rules is displayed in the **Form validation** tab. F21.40

F21.40

Validation rules

List of form validation rules.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Use of taxonomy in selection lists

21.8

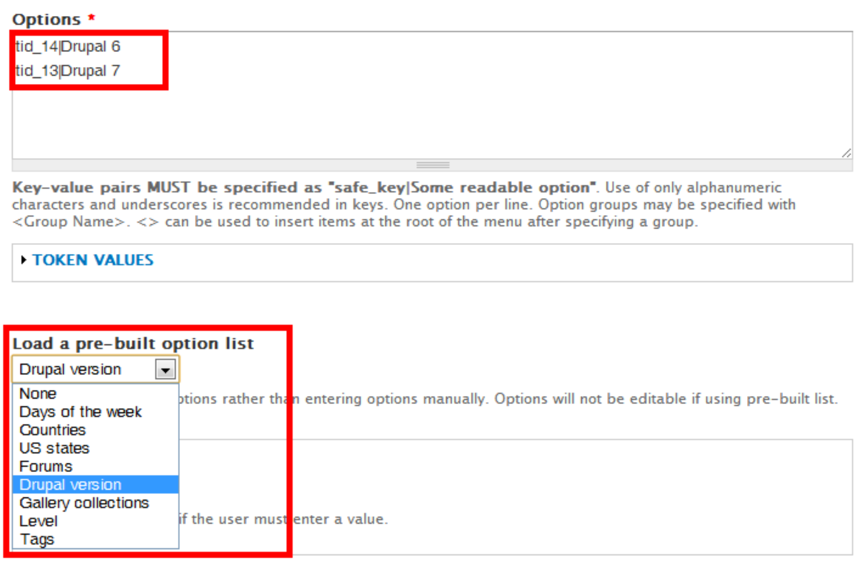
Upon adding a **Select options** component to a form, **we can load a pre-built option list**, such as the days of the week or countries, etc.

The **Webform Term Options** module allows us to use the vocabulary and defined terms of the site taxonomy as lists of predefined options.

The module is available at:

http://drupal.org/project/webform_term_opts

Once installed, when we create a **Select options** component, all the site vocabularies are displayed as lists in the **Load a pre-build option list. The selected list loads** in the **Options** field and its values cannot be modified. For each option, the **key** corresponds to the **term id (tid)** and the **value** to its name (for example, **tid_13|Drupal 7**). F21.41



F21.41

Load a pre-build option list

The Webform Term Options module allows us to create lists of options from the taxonomy terms.

The available vocabularies of the site are displayed as pre-created option lists.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

22 Advanced content types

In **Unit 5** of the beginner level we saw how to create new content types. Content types are entities, which mean that we can add additional fields of different types (text, image, file, numeric, date, etc.).

In this unit we explore advanced configuration options for content types and especially for fields. Much functionality can be applied to other entities.

As well as reviewing the types of field available in the core, we look at the modules that add further field types, such as multimedia elements, dates and references to other entities.

We also look at how to automatically generate URL aliases by following replacement patterns.

Finally we explore how to control permissions on individual fields and on nodes.

Comparative D7/D6

CCK vs. entities and fields

In Drupal 6 we used the CKK module (Content Construction Kit) to add additional fields to the content types.

As we know, in Drupal 7 the concept of the entity was introduced along with the ability to group fields. This means that CKK is no longer used and has become part of the core via the entities and fields.

In Drupal 7 we find new contributed modules which add additional fields to the entities.

Content templates

In Drupal 6 we worked with the Content templates (Contemplate) module to change the content type templates via the site interface.

In Drupal 7, due to internal changes to node presentation, the module is not available. In this level we will see how to modify template files to improve content presentation.

Unit contents

22.1 Content types definition...36
22.2 Fields in core...38
22.3 Multimedia fields...47
22.4 Date fields...49
22.5 Fields to reference other entities.................................52
22.6 Field display...54
22.7 Automatic URL Aliases using Pathauto..........................57
22.8 Field and node permission control................................65
22.9 Additional modules...69

22

22.1 Content types definition

In the beginner level (**Unit 5**) we studied how to define content types and add extra fields. In this unit we will look at the available core fields in more detail and at the extra fields that we can install via the contributed modules.

One of the first steps before starting to build a website is to define the content types, their fields and the relationships between them.

For example, in this unit we define the content types needed to create an **online video store**. What follows is a simplified definition, focusing on teaching the main contents of this course.

Content types for an online video store

In the following sections we will build the content types that act as a base for the display of a movie catalogue for our online video store:

- **Movie**. Displays information about the movies available in the video store.
- **Participant**. Participant refers to the people associated with a movie, such as actors, directors or writers. Since a person can have several roles in one or more movies, we opt to create a content type that is common to all of them. As we shall see, the role reference will be made from the Movie content type.

The **Movie** content type fields are:

- **Title**. The page title corresponds to the title of the movie.
- **Synopsis**. Corresponds to the node´s body field.
- **Cover**. Cover image for the movie.
- **Country**. We will choose a country from the taxonomy term list.
- **Year**. Indicates the year the movie was released.
- **Director**. It will be a reference to a Participant node. A movie can have one or several directors.
- **Cast**. A reference to a Participant node. A movie can have one or several actors.
- **Genre**. It will be a vocabulary that allows for multiple term selections.
- **Rating**. It will be a vocabulary that allows for a single term selection.
- **Running time**. A numerical field (whole numbers) that indicates the running time of the movie.
- **Rent price**. A numerical field (decimal) that indicates the cost of renting the movie.
- **Available from**. Date when movie was released.
- **Trailer**. The trailer of the movie.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

The **Participant content type fields** are:

- **Name**. Stage name of the participant
- **Biography**. Corresponds to the body field of the content type.
- **Photo**.
- **Date of birth**.
- **Country of birth**. Chose a country from the taxonomy terms list.

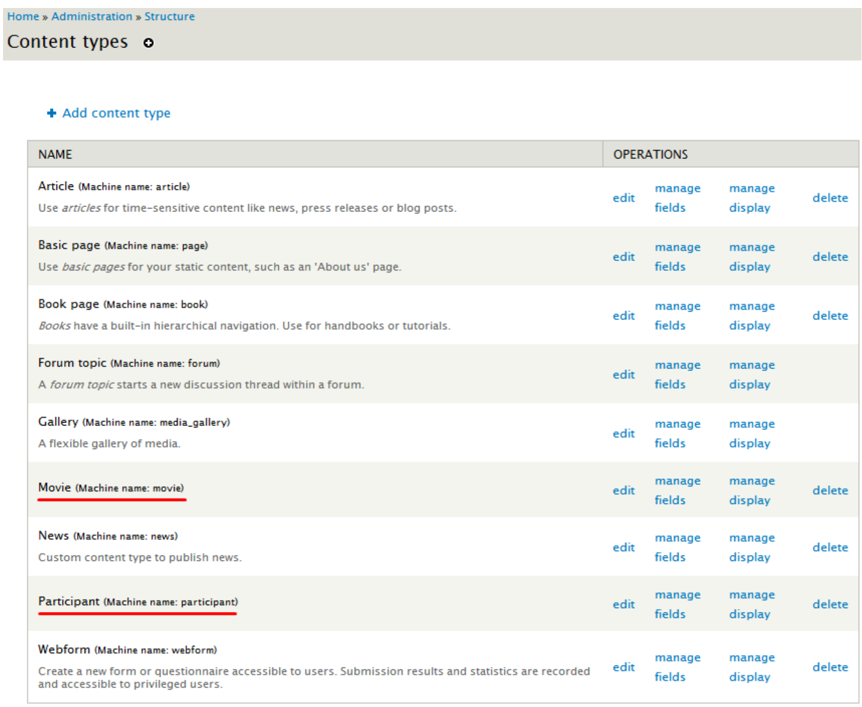
A movie can be connected to several participants, be they actors or directors. As we find out in **Unit 23**, this connection allows us to obtain different **views** or displays of the information. For example, we can present a list of all the movies in the video store, or only those that contain a certain actor or director.

Add new content types

To create a new content type go to: F22.1

Administration ⇨ Structure ⇨ Content types

URL Content types
/admin/structure/types



F22.1
Content types
Shows the list of content types available on the site.
You can modify an existing content type or add new content types.

When adding new content types we have to check that the associated system name is correctly written. For the content type **Movie** it will be "**movie**" and for the content type **Participant** we must use "**participant**".

Configure the **Movie** content type with the following options:

- Machine name: **movie**.
- The title field label is **Title** (Submission form settings).
- By default nodes will be published and won't be promoted to front page (Publishing options).

- Author and date of publication data is not displayed (Display settings).
- Comments opened (Comment settings).

Configure the **Participant** content type with the following options:

- Machine name: **participant**.
- The title field label is **Name** (Submission form settings).
- By default nodes will be published and won't be promoted to front page (Publishing options).
- Author and date of publication data is not displayed (Display settings).
- Comments closed (Comment settings).

For more information about content type configuration, see units 4 and 5 of the beginner level. In the following sections we look at how to add extra fields as needed.

22.2 Fields in core

In **Unit 5** of the beginner level we saw how to add core fields in the content types. By core fields we mean fields generated by the core modules of Drupal.

To create a new field we go to the content type editor, **Manage fields** tab, and input the following information: F22.2

- **Label**. Name that is displayed to the user of the node creation form (for this content type).

- **Machine name**. **Name associated with the field**. Drupal uses this name internally and adds the prefix **field_**. Only a-z characters, numbers (0-9) and underscores (_) are permitted. This value cannot be modified later on.

- **Type of data to store**. A range of data types are available for the new field. The available core fields are as follows:

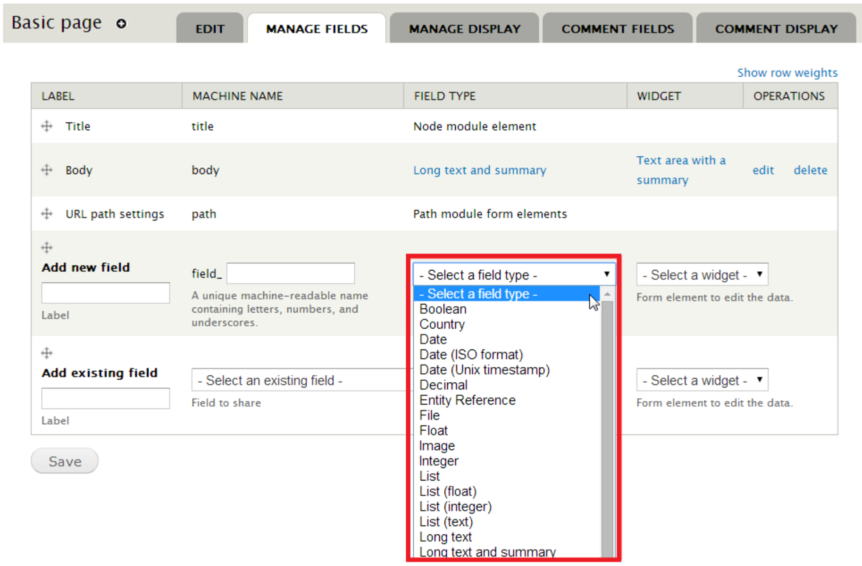
    - **File**, allows a file to be attached to the node. We can choose which file extensions are permitted.
    - **Boolean**, a field that stores two values (true/false, on/off).
    - **Float**, **Decimal** and **Integer**, allow numerical values in different formats.
    - **Image**, allows an image to be added to the form. We can choose the permitted file extensions for the image, as well as other conversion options.
    - **List (float), List (integer) and List (text)** allow lists of numbers of texts.
    - **Term reference**, allows a term selector to be added to the taxonomy vocabulary.
    - **Text, Long text, Long text and summary**, allow different types of text field.

NOTE

Field names

Use prefixes when assigning an internal field name. For example, to create a "Name" field for use in a Curriculum Vitae (CV) content type, call it **field_cv_name**, rather than **field_name**. This prevents later confusion when reusing existing fields in different content types, or when creating fields with the same label.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



F22.2

Field types available in the core

The displayed fields come by default in the core Fields module.

In the intermediate level we look at extra modules that complement the core, adding new fields and configuration and display options.

- **Form element to edit the data (Widget)**. Define the element type or form control that is displayed to the user during the new node creation process. For example, a text area, a selection list, a set of radio buttons, etc. The available elements depend on the field type selected and on the extra fields installed. The selected control can be modified later on when the field is edited, via the **Widget type** tab.

Once the values are introduced to create any of these fields we need to click on the save button. Depending on the type of field we are adding, the system displays a configuration form related to the new field.

In the field **Edit** tab we find two option blocks. The first block (***Content type* Settings**) includes configuration options specific to the content type where the field has been created or inserted. Bear in mind that the same field can be reused in various different content types (and in other entities). This block therefore allows us to modify the same field for different content types. The second block (***Field name* field settings**) includes general field options which are applied to all content types (and in general to all entities) where the field is used.

Some configuration options that are shared by almost all fields are:

Content type settings. The settings indicated here only affect the particular content type to which the new field is added.

  - **Label**. Allows the selection of a field label name to be displayed in the creation form or node editor.
  - **Required field**. Check this option to make the field mandatory.
  - **Help text**. The help text displayed to the user.
  - **Default value**. Defines the default value contained in the field when a new node is created.

- **Field settings**. The settings indicated in this section affect the field across any content type in which it is present. As previously mentioned, different content types (and, in general, entities) can share the same field. As such, these options are shared by all the content types that

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

share the field.

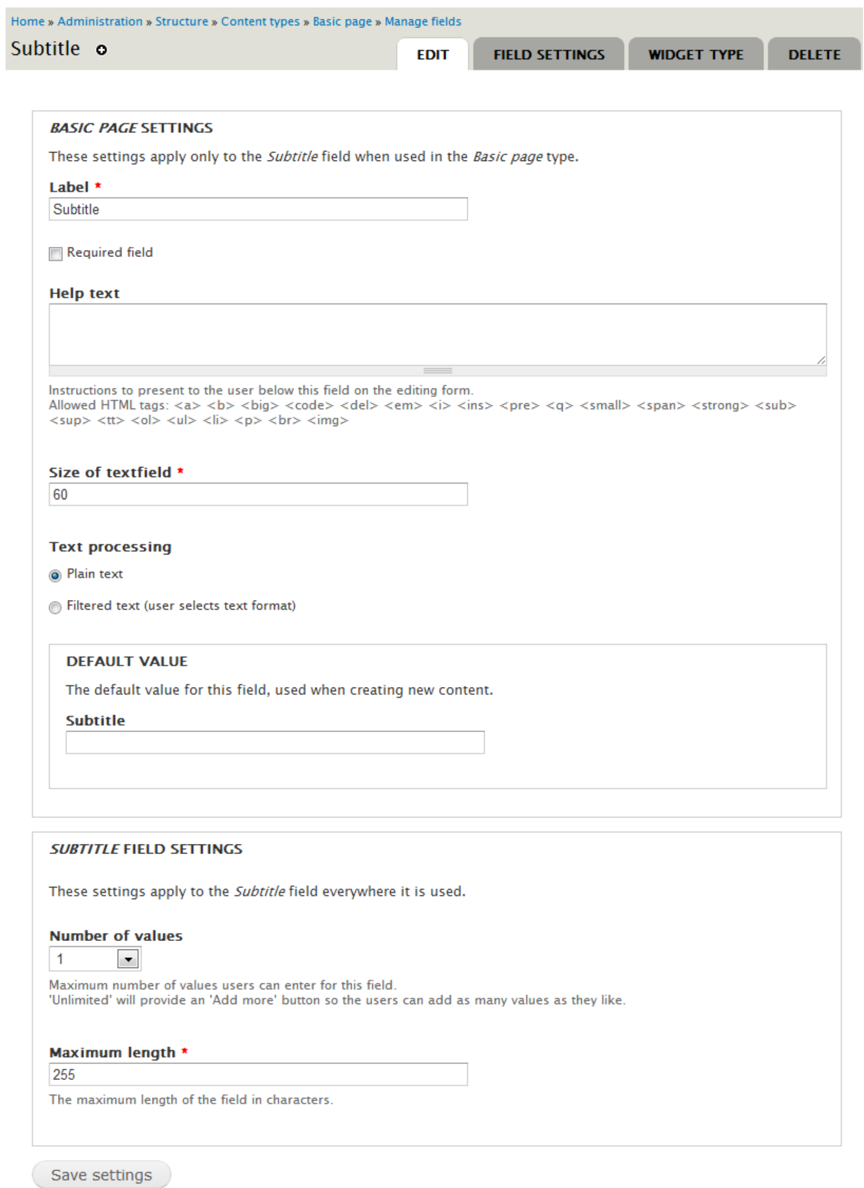
- **Number of values**. Determines the maximum number of values that the user can add. This can be defined as unlimited, which brings up an Add more field.

Next we look at the specific options included in core fields and what the field types are.

Text fields

In the core there are three available text field types: **Text**, **Long text** and **Long text and summary**.

F22.3
Text type fields
There are three types of text field: Text, Long text and Long text and summary.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

The **Text** field has the following unique settings options: F22.3

- **Size of textfield**. Indicates the size, in characters, of the text field.
- **Text processing**. We select this option if the text is unformatted or filtered, depending on the available formats.
- **Maximum length**. Determines the maximum number of characters that the inputted value can contain.

The **Long text** field has the following unique settings options:

- **Rows**. Indicates the size of textboxes in the rows.
- **Text processing**. We select this option if the text is unformatted or filtered, depending on the available formats.

The **Long text and summary**, as well as the options available for the **Long text** format, includes:

- **Summary input**. Allows the user who creates a node to write a summary. If no summary is provided, pages that display a node summary show an automatically generated text.

For **Movie** and **Participant** content types we don't need to add extra **Text** fields.

Numeric fields

The numeric types available in the core are: **Integer**, **Decimal** and **Float**. We also include the **Boolean** type.

The **Integer** field has the following unique settings options:

- **Minimum and Maximum**. Allows us to indicate a range of permitted values. If we leave these values blank no limits are established.
- **Prefix and Suffix**. Allows us to add a prefix or suffix to the text, for example to allow us to associate a currency or measurement symbol to the value. To separate the text from the field value we must add a space before or after, as appropriate.
- **Default value**. Allows us to add an initial value for the field so that the user can modify it when creating the node.

In the **Decimal** field we also find the Integer field option, as well as the following:

- **Precision**. Total number of digits.
- **Scale**. Number of decimal digits.
- **Decimal marker**. Digit that is used as a decimal separator.

The **Float** field only allows us to specify a **Decimal marker** type, along with the content type options available for the previous contents.

The **Boolean** field admits two values (0/1). When creating a field we can choose two different types of control:

- **Check boxes/radio buttons**. Displays all possible options as radio buttons.
- **Single on/off checkbox**. Displays a single selection box. The selection box corresponds to the field value 1.

Among the available **Boolean** field settings options are:

- **Default value**. Allows us to indicate an initial value for the field, which the user can modify when creating a node.
- **On/Off values**. By default the values are 1 and 0, respectively. These values can be modified when creating a node.

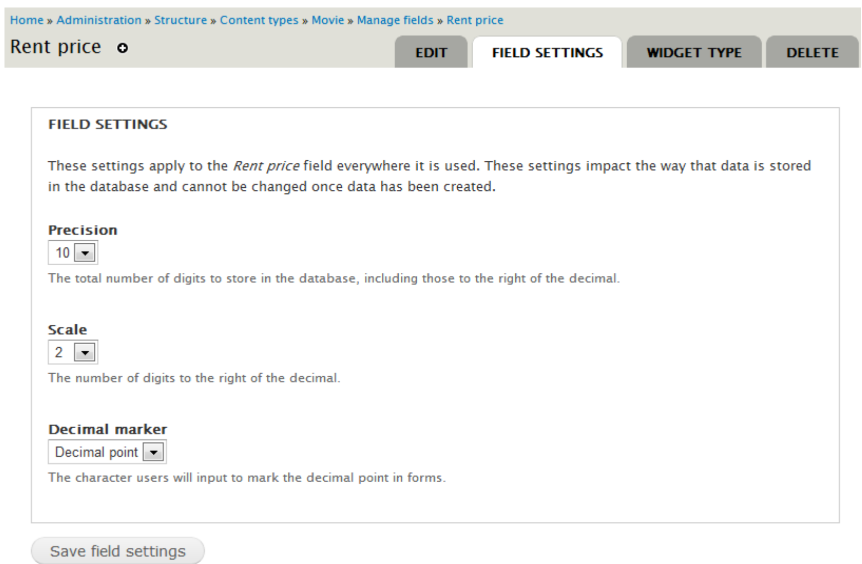
For **Movie** type content we must add the following additional numeric fields:

- **Running time** (field_movie_running_time). **Integer** field. We don't need to indicate a minimum and maximum, but we can add the **Suffix "minutes"**.
- **Rent price** (field_movie_rent_price). **Decimal** field. In precision we select 10 and in scale 2. As a decimal marker we select "decimal point". F22.4

F22.4

Numeric fields

The numeric field types available in the core are: **Whole number**, **Decimal number** and **floating comma**. We also include **Boolean**.



List fields

Three list fields exist, depending on the element type: **List (float)**, **List (integer)** and **List (text)**.

When creating a **List** type field (of any of the three types) we must select the control display from the following options:

- **Select list**. Displays the options as a list. The user can select an option.
- **Check boxes/radio buttons**. Displays the options as checkboxes or radio buttons, depending on whether the user can select several options or a single option, respectively.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Among the **List** field settings options are:

- **Allowed values list**. This field indicates the options, one per line, following the **key|label** format. The key is the internal value, while the label is the text string that is displayed to the user. For example: 1|red, 2|blue, 3|green (one option per line).

Although the **Movie** and **Participant** content types display lists of options (Genre, Rating, Country) we don't use this type of field. Instead we use **Term reference** type fields as explained below.

Term reference field

The **Term reference** field allows us to add selection lists that make use of the terms included in a vocabulary taxonomy.

Consequently the taxonomy vocabularies must already be created and include their corresponding terms.
We create the following vocabulary and terms structure, which is then applied to the **Movie** and **Participant** content types:

- **Movie Genre**:
    - Action
    - Adventure
    - Animation
    - Comedy
    - Crime
    - Documentary
    - Drama
    - Fantasy
    - Horror
    - Sci-Fi
    - Thriller

- **Rating**:
    - Rate pending.
    - Suitable for all audiences.
    - Not suitable for children under 7.
    - Not suitable for children under 12.
    - Not suitable for children under 16.
    - Not suitable for children under 18.

- **Country**: Includes a short list of countries (United States, United Kingdom, Spain, etc.).

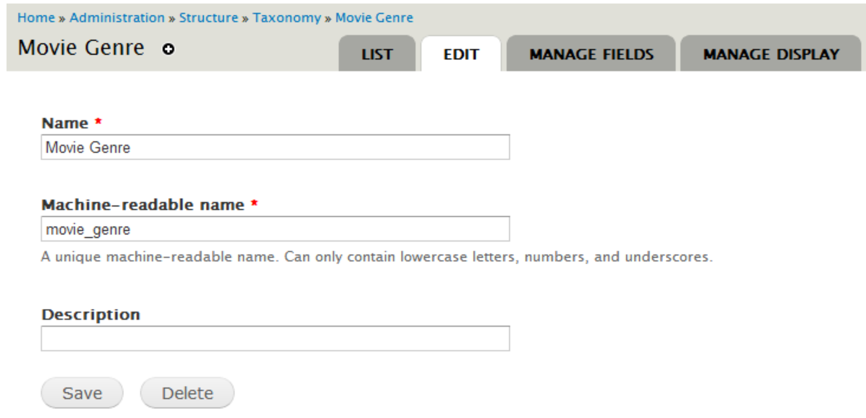
As each vocabulary is created we check that the system name corresponds to the following values: **movie_genre**, **rating**, **country**. F22.5

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F22.5

Vocabulary creation

You create the Movie genre vocabulary and rewrite and modify the system name.

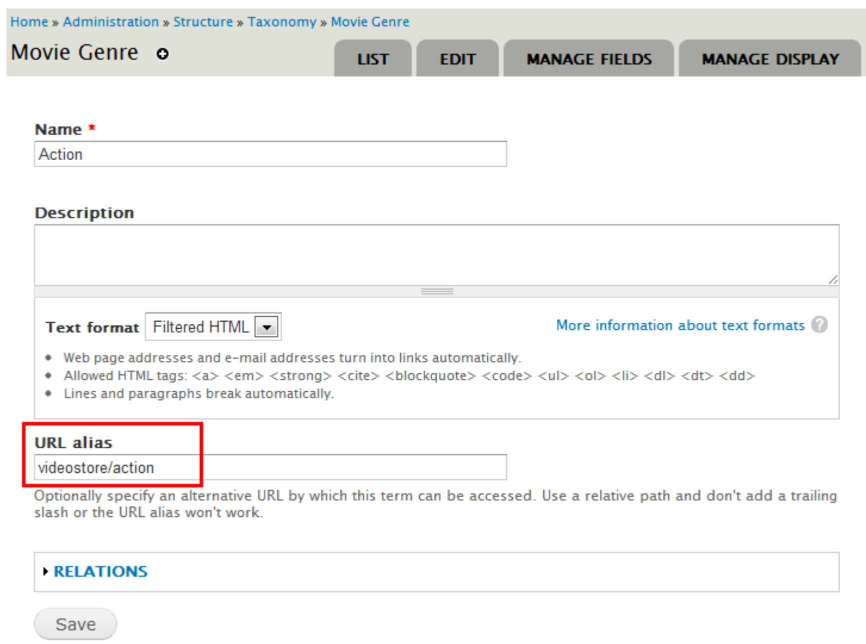


Within the **Movie Genre, add** for each term an URL alias (for example: videostore/action, videostore/horror, etc.). We will use these URLs to display all movies of a particular genre. **F22.6**

F22.6

Adding vocabulary terms

You create the taxonomy terms associated with the Film genre vocabulary.



Once the vocabularies mentioned above are created, we will add the **Term reference** fields.

As these are lists, the Term reference fields allow us to select between the following controls:

- **Select list**. Displays the options as a list. The user can select an option from the list.
- **Check boxes/radio buttons**. Displays the options as checkboxes or radio buttons, depending on whether the user can select several options or just a single option.
- **Autocomplete term widget** (tagging). Displays a text box where vocabulary terms can be inputted manually. When the system locates a term it autocompletes so the user does not have to enter the complete term.

Once added, we must select the **Vocabulary** associated with the field and the

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

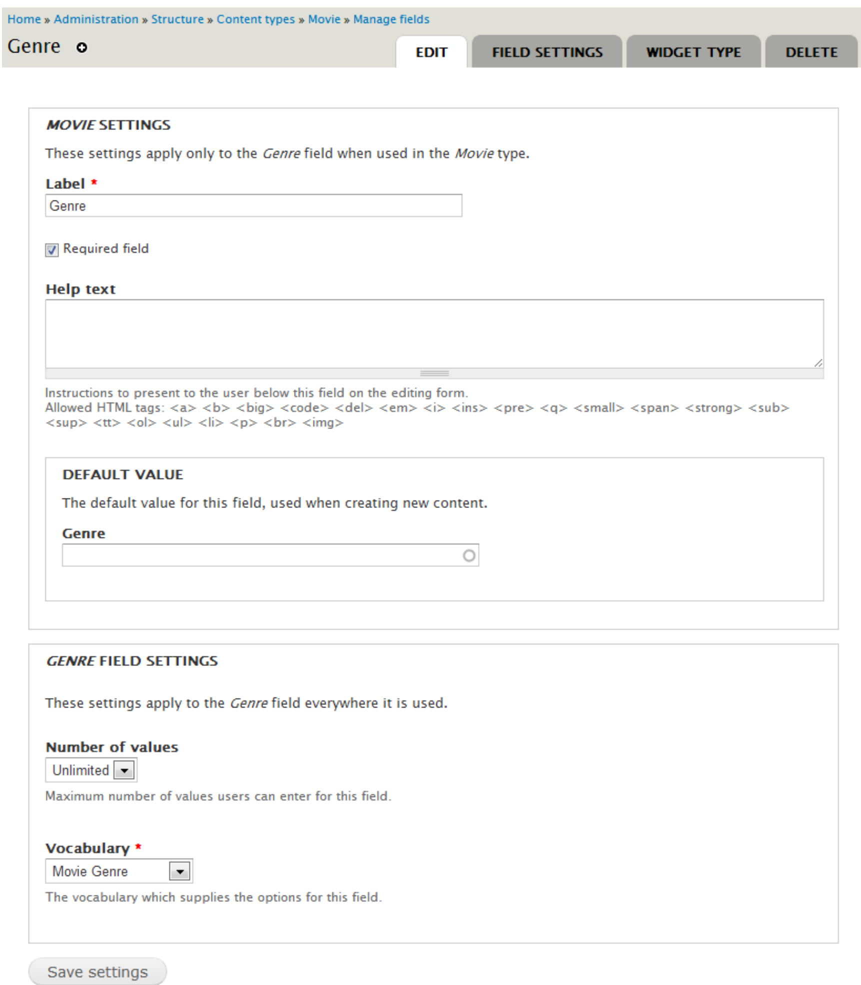
permitted **Number of values**. F22.7

For **Movie** type content we create the following fields:

- **Genre** (field_movie_genre), with the associated **Movie Genre** vocabulary. A movie has at least one associated genre but may have several. We assign the **Autocomplete term widget control**.
- **Rating** (field_movie_rating), with the associated **Rating** vocabulary. We use a selection list. Each movie must have a single associated value (mandatory).
- **Country** (field_movie_country), which uses the **Country** vocabulary. We use a selection list. Each movie can have several assigned countries.

For **Participant** type content we create the field:

- **Country of birth** (field_participant_country), with the associated **Country** vocabulary. We use a selection list. Each participant must have a single assigned value (mandatory).



F22.7

Term references

When adding a Term references field we indicate the vocabulary from where the elements list is extracted.

File and Image fields

The **File** field allows files to be attached to the node. The **Image** field is very similar but it´s used specifically for images.

Some of the fields they share are:

- **Allowed file extensions**. The extensions must be written without the dot, separated by a space or a comma.
- **File directory**. Allows us to choose a subdirectory where uploaded files are saved (within the main files directory of the site).
- **Maximum upload size**. Can be used to specify different sizes (KB, MB)
- **Progress indicator**. Displays a progress bar during file upload and the percentage left to load
- **Upload destination**. Determines whether the files are public (**Public files**) or private (**Private files**).

In **Files** we can also select:

- **Enable *Display* field**. Allows us to choose whether the uploaded files are displayed together with the node presentation or not. If activated we can also indicate that the Files are to be displayed by default.
- **Enable *Description* field**. Allows us to add a description in a text field to any uploaded file.

The settings options unique to **Images** are:

- **Maximum/Minimum image resolution**. Allows us to specify the maximum and minimum size limits, expressed as **width x height** (in pixels).
- **Enable Alt/Title field**. Allows us to complete the *alt* and *title* attributes that are added to the *img* label in the HTML code.
- **Preview image style**. Allows us to indicate the image style in preview mode. The default value is *thumbnail*, which is a predefined style that displays a miniature of the image.
- **Default image**. Allows us to select a default image.

- Returning to our content types, we must create the following **Image** fields:

- **Cover** (field_movie_cover), for the **Movie** content type.
- **Photo** (field_participant_photo), for the **Participant** content type.

In both fields the permitted extensions are: png, gif, jpg, jpeg. Uploaded files are stored in the file "videostore".

From **Manage Display** we select **Image style** "**medium**" for the default display, with no link (**Link image to Nothing**).

With the fields available in the core we have created many of the fields defined in the content type structure. In the following sections we see how, via additional modules, we can add other field types, such as videos, dates and relationships between nodes.

Multimedia fields

22.3

In **Unit 16** of the Beginner Level we installed the **Media** module as a base for publishing multimedia content. (videos, images, audio, documents, etc.).

The Media module consists of two modules, **Media** and **Media Internet Sources**. The second allows us to integrate multimedia content from external providers, such as videos from YouTube.

The **Media Internet Sources** module acts as a base that allows other modules to integrate with defined content providers. If we want to include YouTube videos we need the **Media: YouTube** module, and if we want to display Flickr we need the **Media: Flickr** module. There are "**Media:**" modules for many content providers. Furthermore, each module has its own configuration options based on the information required by the content provider.

This combination of modules can also be used to create multimedia fields identified in the field selector as **File**. **Note that the old *Multimedia asset* type field is now deprecated so don't use it**.

To create the video field that displays the **trailer** of the **Movie**, we install the **Media**, **Media Internet Sources** and **Media: YouTube** modules, available at:

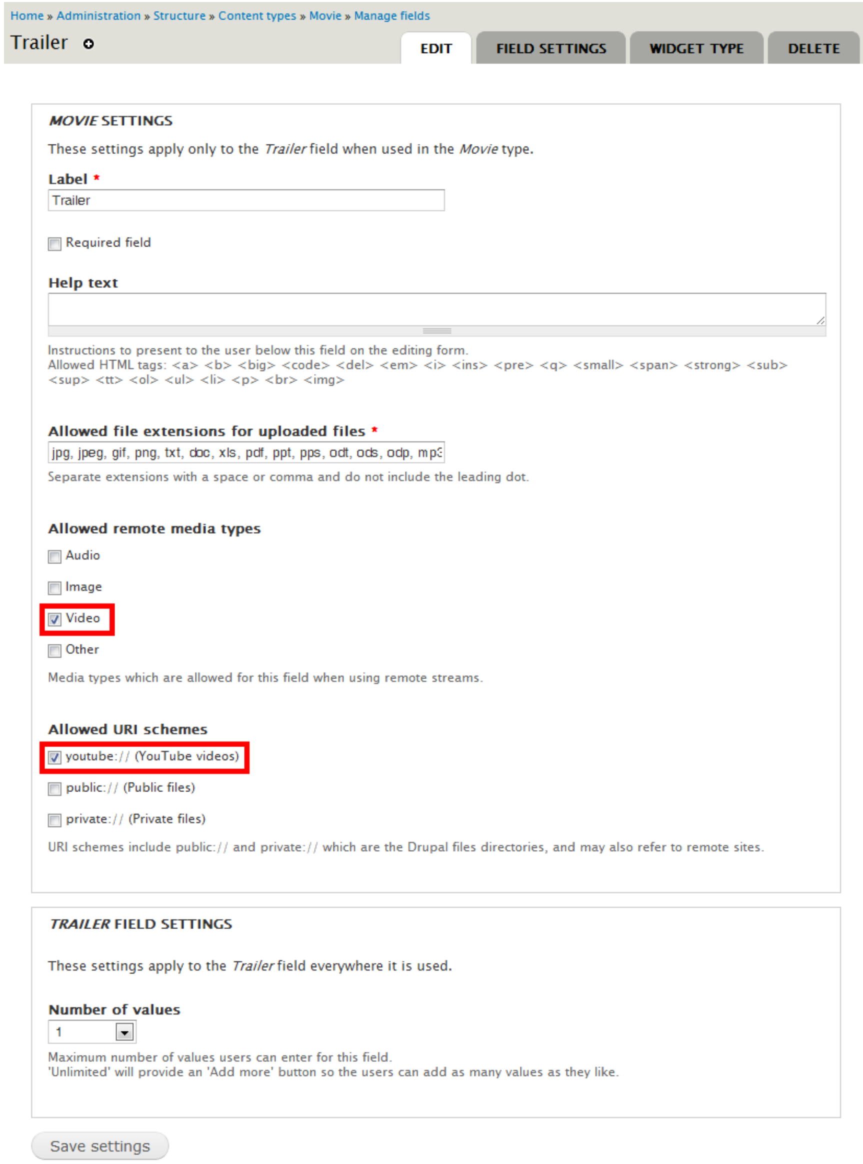
http://drupal.org/project/media
http://drupal.org/project/media_youtube

In the **Movie** content type we add a **File** type field, called **Trailer** (field_movie_trailer), with the control **Media file selector**. In the field options we select the **Video** type and the source **YouTube**. F22.8

F22.8

Media field

The Multimedia asset field allows you to add videos, images, audio, etc. With additional modules you can include multimedia resources from external sources such as Youtube, Flickr, etc.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Date fields

22.4

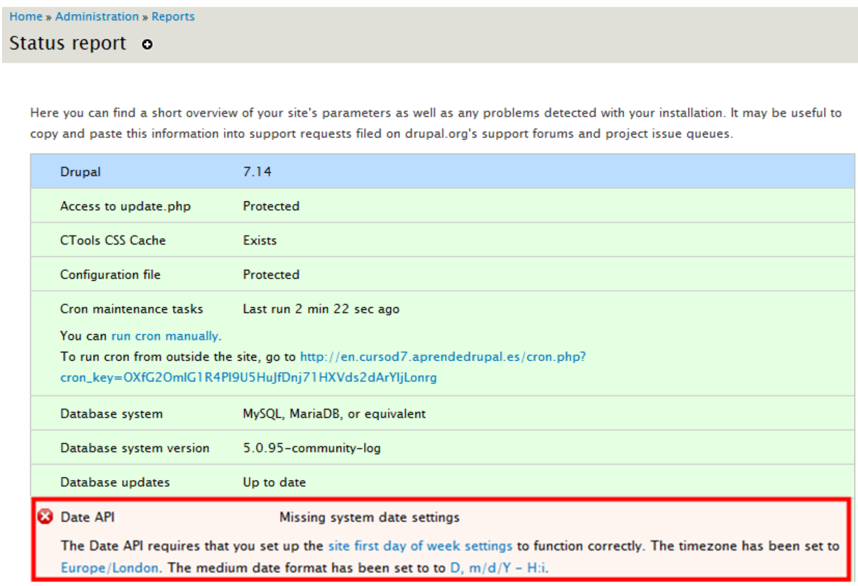
The **Date** module allows us to add Date type fields to entities. We can download it here:

http://drupalmodules.com/module/date

The Date module consists of a set of modules. For the moment we will use the following:

- **Date**
- **Date API**
- **Date Popup**

On installing Date, the system displays an error message, which we can view within Status report. F22.9



Home » Administration » Reports

Status report

Here you can find a short overview of your site's parameters as well as any problems detected with your installation. It may be useful to copy and paste this information into support requests filed on drupal.org's support forums and project issue queues.

Drupal	7.14
Access to update.php	Protected
CTools CSS Cache	Exists
Configuration file	Protected
Cron maintenance tasks	Last run 2 min 22 sec ago You can run cron manually. To run cron from outside the site, go to http://en.cursod7.aprendedrupal.es/cron.php?cron_key=OXfG2OmIG1R4PI9U5HuJfDnj71HXVds2dArYIjLonrg
Database system	MySQL, MariaDB, or equivalent
Database system version	5.0.95-community-log
Database updates	Up to date
Date API	Missing system date settings The Date API requires that you set up the site first day of week settings to function correctly. The timezone has been set to Europe/London. The medium date format has been set to to D, m/d/Y - H:i.

F22.9

Date module

On installing the Date module you must fill in the system date setting.

To solve this, we simple configure some of the module settings, such as the first day of the week and the time zone.

To configure these settings go to: F22.10

Administration ⇨ Configuration ⇨ Regional and language ⇨ Regional settings

URL Regional settings
admin/config/regional/settings

Choose the first day of the week (Sunday or Monday), the country and the corresponding time zone. Bear in mind that the time zone is not defined by the location of the server, but is the time zone where the website is based.

Once we save the configuration the error message disappears.

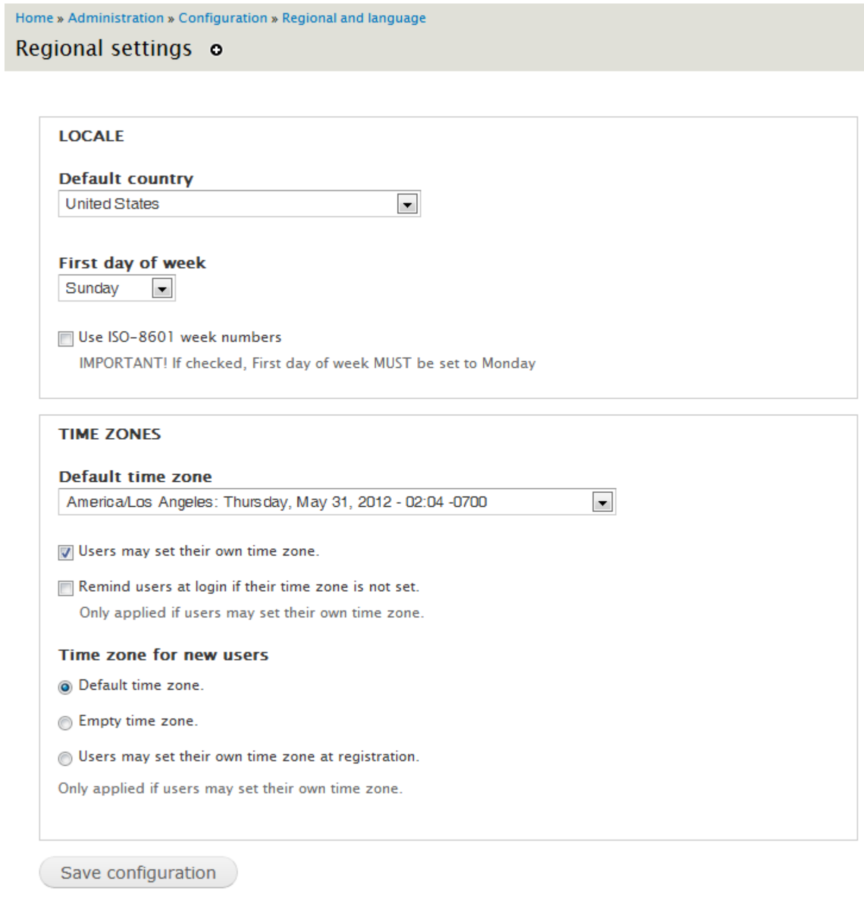
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F22.10

Regional settings

In Regional settings you must select the first day of the week and the default time zone.

Even if the existing values are correct, you must save the settings to get rid of the error message displayed by the Date module.



The available date type fields are: Date, Date (ISO format) and Date (UNIX timestamp). The format type only affects the format in which the date is stored in the database.

Among the field options we find:

- **Date attributes to collect**. Controls the granularity of the date. If we only need the year, we will select only Year. For a birthday select Year, Month and Day. For more specific dates, such as a publication date, we can also include Hour and Minute (and even Second).
- **Collect an end date**. Activating this option gives two dates, a Start date and an End date. Useful, for example, for defining the start and end dates of an event.
- **Time zone handling**. Allows us to configure the time zone to be used in the date field.

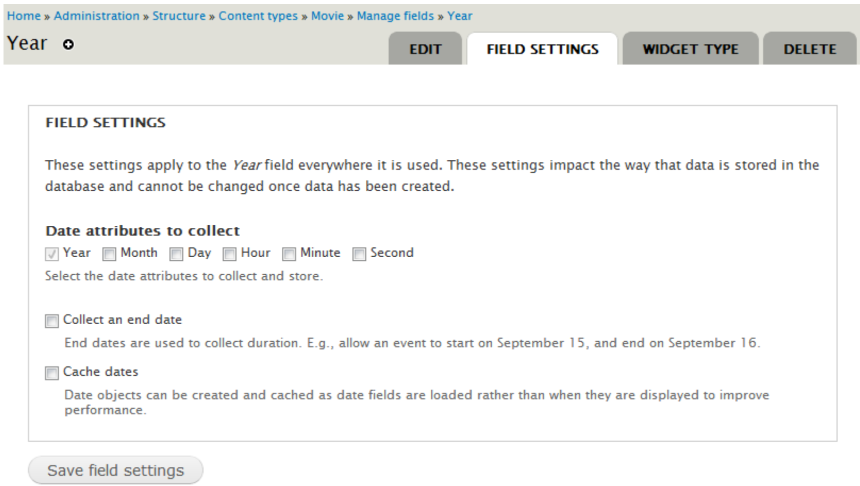
- In the **Movie** type content we create the following **Date** type fields:

- **Year** (field_movie_year). Only asks for the Year date value. **F22.11** The field control type is **Select list**.
- **Available from** (field_movie_available_from). Asks for the Year, Month and Day values. The field control type is **Pop-up calendar**.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

In the **Participant** content type we create the field:

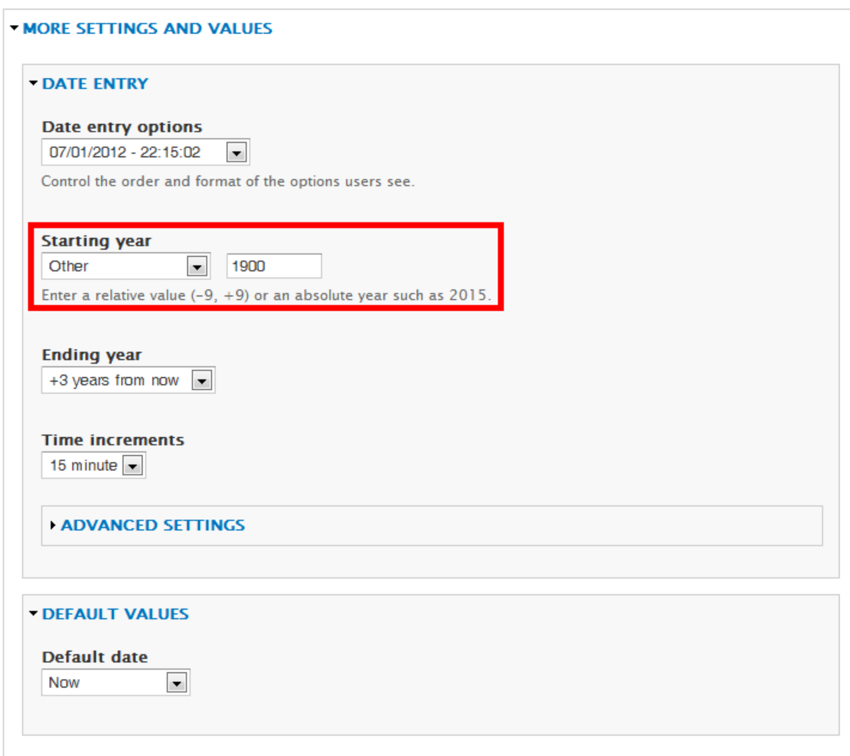
- **Date of birth** (field_participant_birth). Asks for the Year, Month andDay values. The field control type is **Select list**.



Date field

We select the date attributes that are included in the field (year, month, day, hour, minute and/or second).

Once the new field is saved, we can configure other values, such as the first and last year to be displayed in the year selector. This can be a relative value (+3 years from now, -9 years from now) or an absolute value (1900-2015). F22.12



Date field

You can choose the starting and ending year of the selector, making them either absolute values or relative values in respect to the current date.

In the **Date** type fields we can chose between these **widgets**:

- **Select list**. Displays the day, month and year fields as a drop-down list (when these values are required for the date).

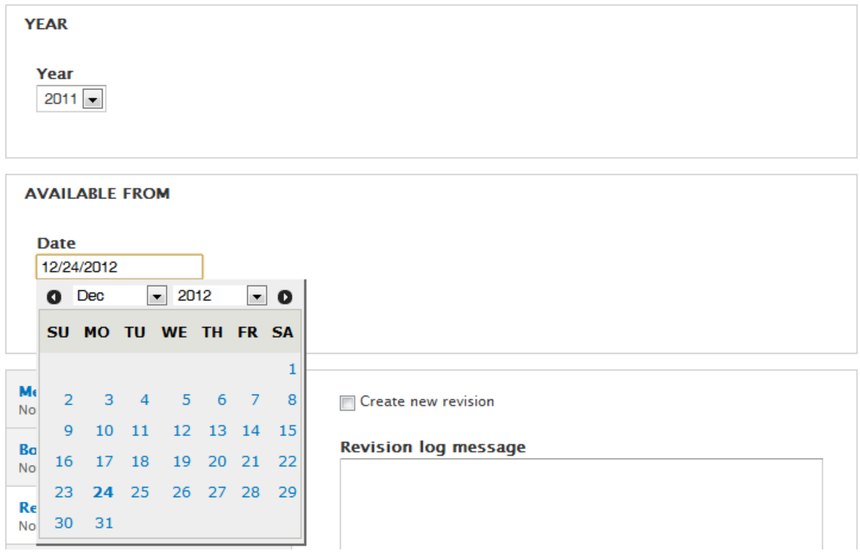
- **Text field**. Creates a single text field for date input.
- **Pop-up calendar**. Allows a pop-up calendar for date selection.

The **Select list** and **Pop-up calendar** controls are shown in **Figure** F22.13 .

F22.13

Date controls

You can input dates via a text box, selection list or a calendar.



22.5 Fields to reference other entities

The **Entity Reference** module allows us to add fields which reference other entities.

A reference to another entity establishes a logical link between two entities. For example, a course (Course content type) can be related to its pupils (entity Users) via an Enrolment field.

In the same way, in our example, we can establish an Actor link between a Movie and a Participant. We can also create a Director field to link both entities.

The **Entity Reference** module is available at:

http://drupal.org/project/entityreference

The module depends on the **Entity API**, available at:

http://drupal.org/project/entity

Alter installing and activating these modules we have a new field type available: **Entity Reference**. We can add it to any entity from **Manage fields**.

The Entity Reference field behaves like an element list where the available elements for selection are named entities. In our example, for the **Cast** field and the **Director** field (both created in the **Movie** type content), all the **Participant** type nodes are displayed.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

The widgets that we can use in an Entity reference field are similar to those used in the selection lists and taxonomy term references: Autocomplete, Autocomplete (tags style), Select list and Check boxes/radio buttons.

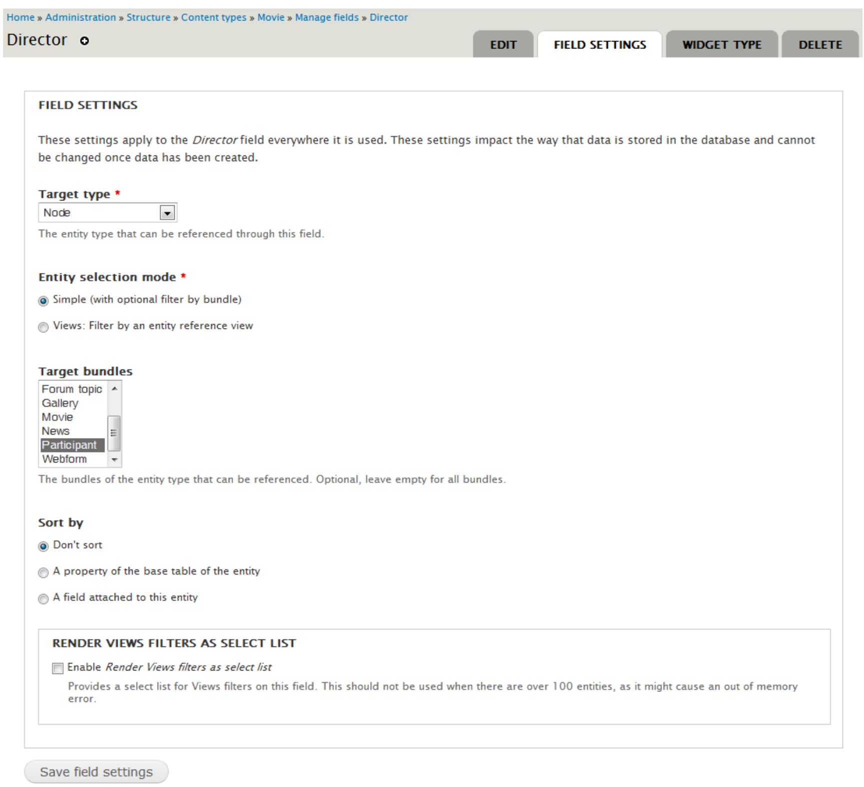
The unique settings options to the **Entity reference** field are:

- **Target type**. Entity type referred to by the field (Comment, Node, File, Taxonomy term, Taxonomy vocabulary, User).

- **Target bundles**. Different elements are displayed, depending on the selected destination type. Of these we can select none (equivalent to all), one or several. If the selected destination type is **Node** then all content types will be displayed in the elements. If we select **Taxonomy term**, all vocabularies are displayed; etc.

We need to create the following **Entity reference** fields in **Movie** content type: F22.14

- **Cast** (field_movie_cast). We use the **Autocomplete** widget. The target type is **Node** and we select the **Participant** target bundle. The field can contain an unlimited number of values.
- **Director** (field_movie_director). We use the **Autocomplete** control. The target type is **Node** and we have to select the **Participant** target bundle. The field can contain an unlimited number of values.



F22.14

Referentes to other entities

You can create fields that reference other entities.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

22.6 Field display

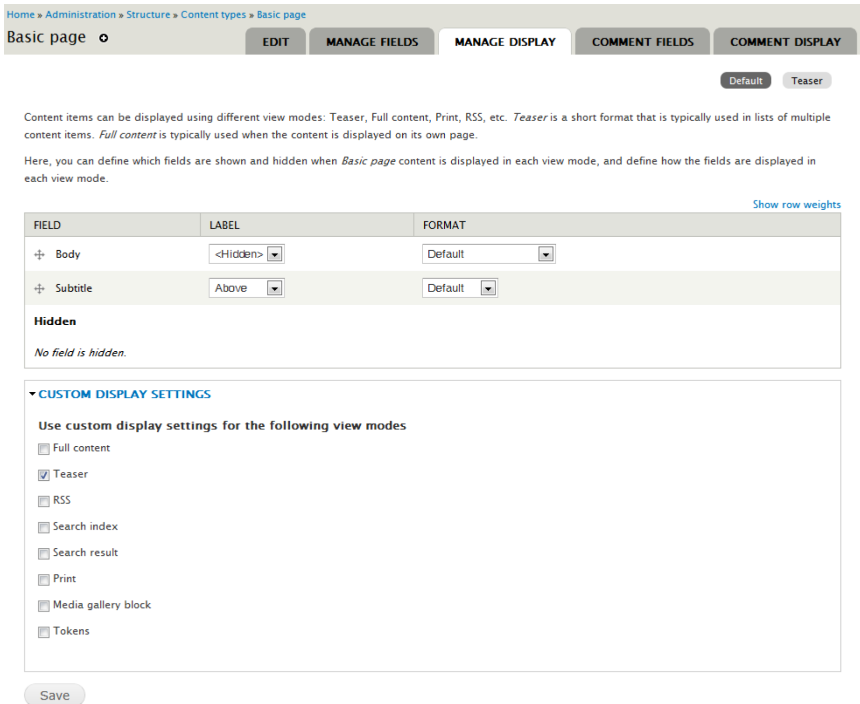
We can configure how the fields display from **Manage Display**: F22.15

- **Order of fields**. Establishes the order of field display.
- **Label position**. Selects whether the field is displayed Above, Inline or is Hidden (<Hidden>).
- **Format**. The format depends on the field type. Selecting <Hidden> means the field doesn't display. The fields that do not allow us to select a specific format have at least a Visible/Hidden option.

F22.15

Manage display

From the Manage display tab we can configure how the fields will be displayed when the node is viewed.



Initially we will modify the default display options (**Default tab**), but we can also configure the display for other node presentations (Full content, Teaser, RSS, Search index, etc.). To activate these tabs we must activate each display in **Custom display settings**.

Figure F22.16 shows the display of a **Movie** type node. We can get to this "design" by modifying the display settings for each field. As shown later in this level, we can markedly improve the node display by modifying the node pattern and the CSS style sheets.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Matrix

During the year 1999, a man named Thomas Anderson (also known as Neo), lives an ordinary life. A software techie by day and a computer hacker by night, he sits alone at home by his monitor, waiting for a sign, a signal - from what or whom he doesn't know - until one night, a mysterious woman named Trinity seeks him out and introduces him to that faceless character he has been waiting for: Morpheus. A messiah of sorts, Morpheus presents Neo with the truth about his world by shedding light on the dark secrets that have troubled him for so long.

Country:
United States

Year:
1999
Director:
Andy Wachowski
Larry Wachowski
Cast:
Keanu Reeves
Genre:
Sci-Fi

Rating:
General Audiences. All Ages Admitted.

Running time:
115minutes
Rent price:
2.95
Available from:
Sunday, January 1, 2012
Trailer:

Add new comment

F22.16

Node display

Display of a Movie node type. We get to this "design" by modifying the display settings for each field. As shown later in this level, we can markedly improve node display by modifying the node pattern and the CSS style sheets.

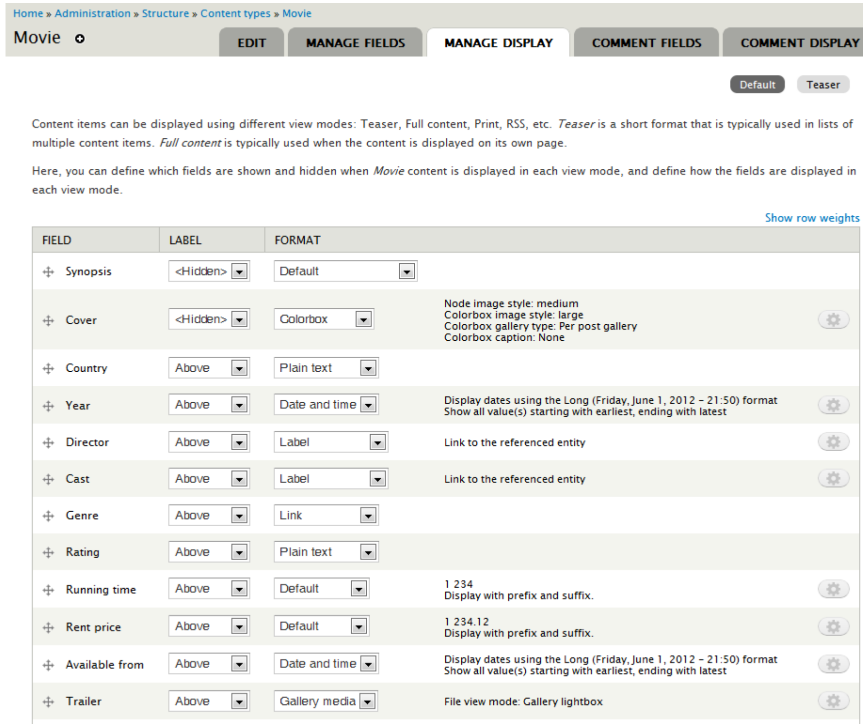
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Now we will explain the display options used in the **Movie** content type and we will configure it from **Manage display**: F22.17

F22.17

Manage display

From the Manage display tab you can modify the display of each field. The available options depend on the field type.



- **Synopsis**. The label must be hidden (Hidden).
- **Cover**. The label must be hidden (Hidden) and we select the Colorbox format. Colorbox is available if we have previously installed the Colorbox module (**Unit 16**). In content **image style** we select "medium", and the Colorbox image style must be "large". With these particular settings, when we click on a movie cover image, it opens as an enlarged image.
- **Country**. Displays as **Plain text**.
- **Year**. Shows the **Date and time** format. In this case only the year is shown.
- **Director**. Displays the **Label** (**Link to the referenced entity**) format. Clicking on the director name link takes us to the participants node.
- **Cast**. Displays the **Label** (**Link to the referenced entity**) format. Clicking on the actor name link takes us to the participants node.
- **Genre**. Displays the **Link** format. Clicking on a term name takes us to a page which displays all nodes with the same assigned term.
- **Rating**. Displays as **Plain text**.
- **Running time**. We select **Display with prefix and suffix**.
- **Rent price**. We select **Display with prefix and suffix**.
- **Available from**. The format is **Date and time**, in **Long** format.
- **Trailer**. To display the Youtube video embedded in the page, we select the **Gallery media** format with the **Gallery lightbox** setting. For this setting option to be available we must have previously installed the **Media Gallery** module (Unit 16).

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Automatic URL aliases using Pathauto

22.7

The **Pathauto** module automatically generates URLs so that there is no need to fill out the **URL path settings** field every time we create a node.
Normally the aliases are generated using information in the node (for example, from the title). The module does a character conversion to prevent the URL from containing special characters, spaces, etc.

For example, for **Movie** content type we can automatically create the alias types:

- videostore/movie/matrix
- videostore/movie/day-earth-stood-still

And for **Participant** type content:

- videostore/participant/keanu-reeves
- videostore/participant/scott-derrickson

As we shall see, it is possible to define replacement patterns to generate the aliases depending on the content type, the categories, the users, etc.

The **Pathauto** module is available at:

http://drupal.org/project/pathauto

It requires the **Token** module, available at:

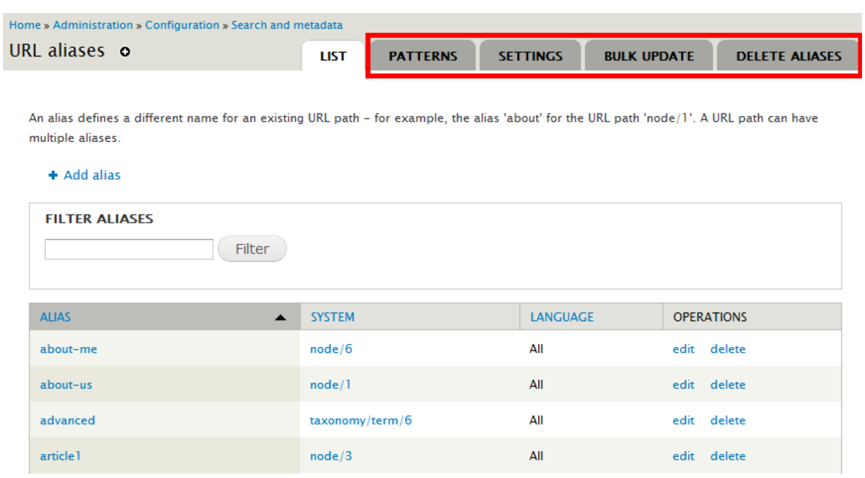
http://drupal.org/project/token

Once installed and activated, we can configure the automatic aliases of the site from **URL aliases**:

Administration ⇨ Configuration ⇨ Search and metadata ⇨ URL aliases

URL URL aliases
/admin/config/search/path

The Pathauto module has added four additional tabs: **Patterns**, **Settings**, **Bulk update** and **Delete aliases**. We will review them one by one. F22.18



F22.18
Pathauto
Additional tabs added by the Pathauto module in URL aliases administration.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

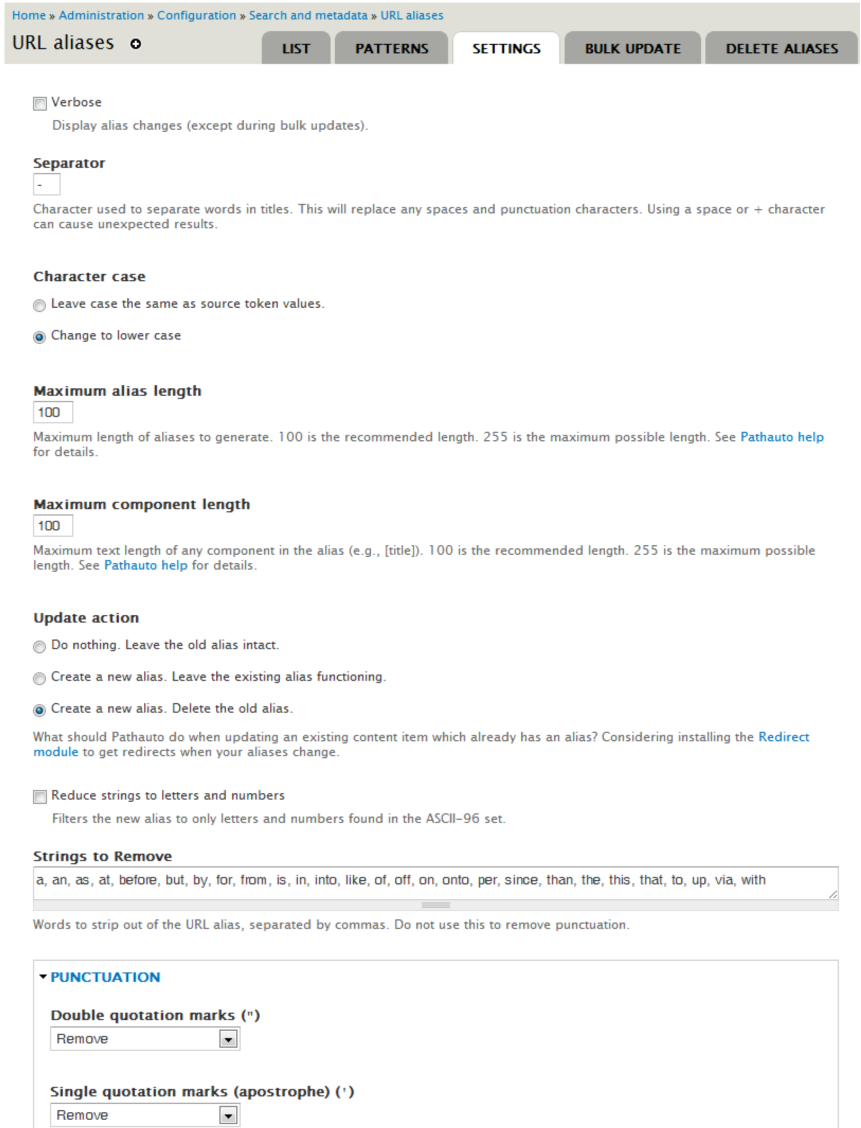
Settings

From the **Settings** tab we set the basic conversion options used to obtain the URLs: F22.19

F22.19

Pathauto. Settings

Basic options that are applied during URL creation.



- **Verbose**. If we activate this option, every time we create an automatic alias (for example, when creating content), it displays a message showing the generated alias.
- **Separator**. Word separator character. The default value is a hyphen. The system substitutes spaces and punctuation marks with the chosen separator symbol.
- **Character case**. By default we select lower case to convert all URLs to lower case. The use of upper case in URLs is not recommended.
- **Maximum alias length**. A maximum length of 100 characters is advisable, although the maximum length permitted is 255.
- **Maximum component length**. A URL can be generated from several components, for example title and date. Here we define the maximum length per component. The maximum length of the URL is defined by the previous field.

- **Update action**. Defines what happens when content is updated (no change to previous alias, create a new alias but leave the previous one active, or create a new alias and delete the previous one).
- **Reduce strings to letters and numbers**. All characters are filtered leaving only letters and numbers as defined by the ASCII-96 set of characters.
- **Strings to remove**. Allows us to include a list of words to be deleted from the URL. This is used, for example, to remove articles and prepositions. We can also leave this field empty so that no strings are removed.
- **Punctuation**. Allows us to decide what to do with special characters. The options for each one are: Remove, Replace by separator or No action (keep the character in the URL).

Transliteration

If we generate a URL and still find that it contains special characters, such as accents, we can replace them with the equivalent character accent marks (replace á for a, é for e, ñ for n, etc.). To do this, we install the **Transliteration** module, available at:

http://drupal.org/project/transliteration

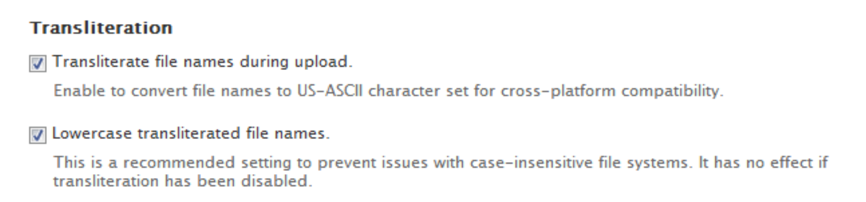
The **Transliteration** module has two functions:

1. It converts file names when they are uploaded to the site.
2. It substitutes URL special characters (characters such as ñ, etc.)

For the first case we go to: F22.20

Administration ⇨ Configuration ⇨ Media ⇨ File system

We simple check the options in the **Transliteration** section to modify file names as they are uploaded to the site. As well as replacing special characters we can also choose to transliterate all characters to lower case. This option is recommended to improve the compatibility across systems.



From the Transliteration tab, on the same page, we can see which uploaded files need to be transliterated. We can perform the transliteration action for all required files. F22.21

Comparing D7/D6
Pathauto
In Drupal 6 special character transliteration was carried out by creating the file i18n-ascii.txt. In Drupal 7 you only have to install the Transliteration module and activate the Pathauto options tab.

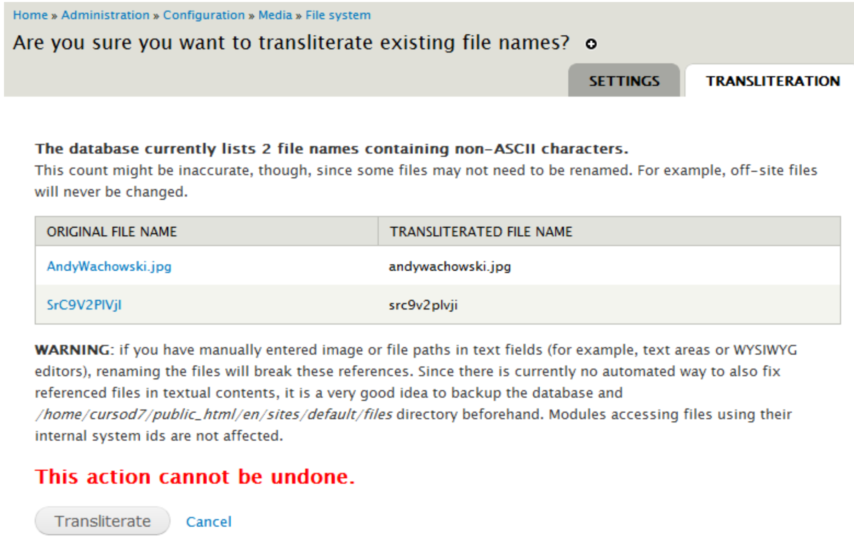
URL File system
/admin/config/media/file-system

F22.20
File transliteration
Allows you to replace special characters with their standard equivalents (a for á, e for é, n for ñ, etc.).

F22.21

File transliteration

List of files awaiting transliteration.



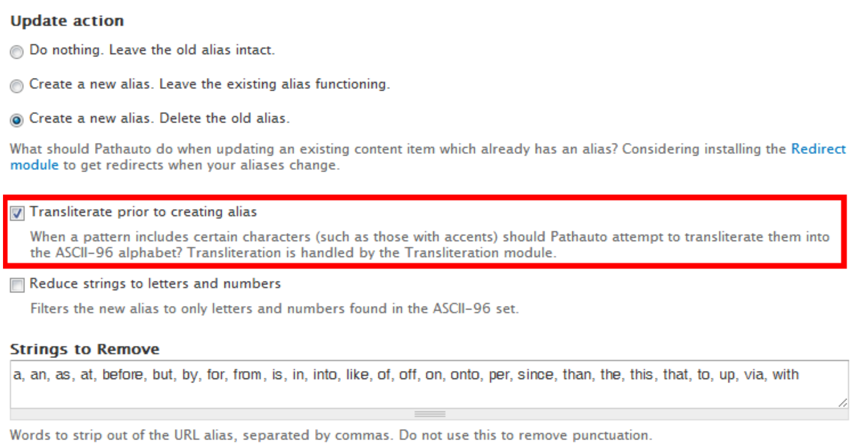
To perform the transliteration during URL conversion, we return to the Pathauto (URL aliases) **Settings** tab A new option appears after installing the module:

- **Transliterate prior to creating alias**. All we have to do to apply the indicated changes is activate this option (á for a, é for e, ñ for n, etc.). **F22.22**

F22.22

Transliterating URL aliases

On applying the transliteration to the URL aliases you can change characters with accents for their accent-free equivalents (á for a, é for e, ñ for n, etc.).



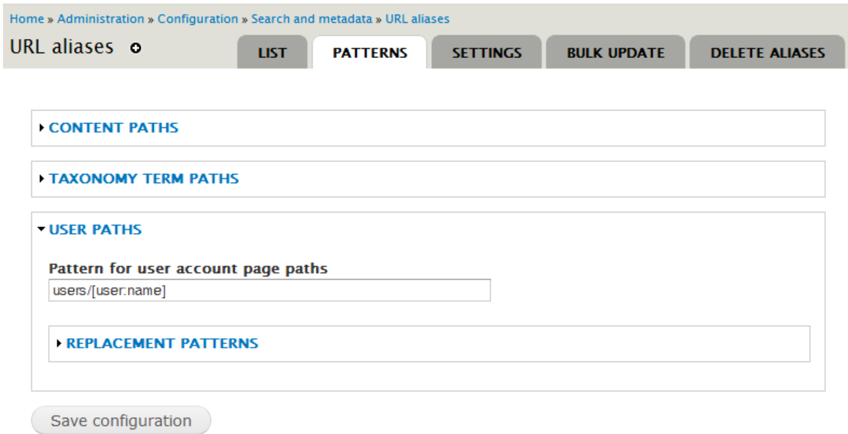
The current settings including the removal of articles and prepositions, allows us to make string conversions.

- Original string: **The Day the Earth Stood Still**
- Generated string: **day-earth-stood-still**

Patterns

Once the basic module settings are configured we must define the path patterns for the different site elements (content paths, user paths and taxonomy term paths).

We will define patterns from the **Patterns** tab: F22.23



F22.23

Pathauto. Patterns

Defining URL alias patterns for the different entities.

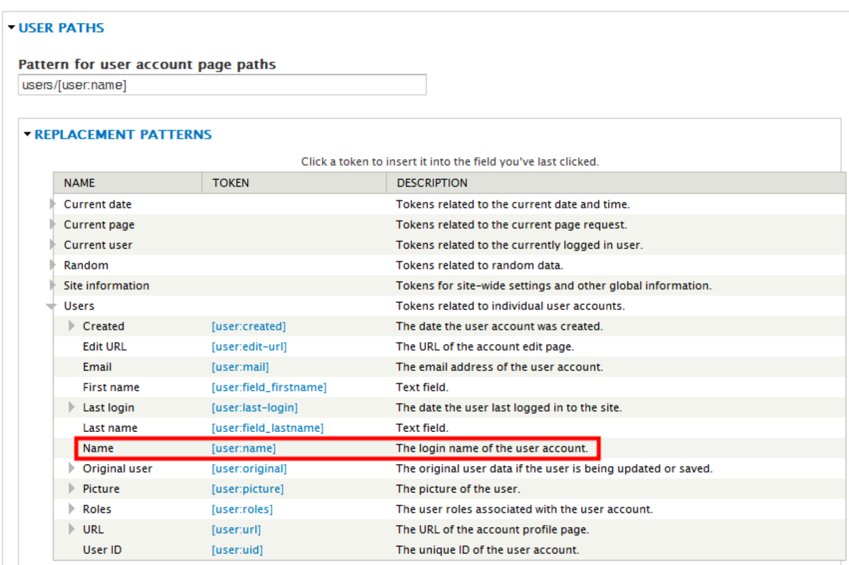
For example, for the User´s paths (user/1, user/2, etc.), the module uses the default pattern:

- users/[user:name]

This allows us to generate these URLs types:

- users/frankgil
- users/laurafornie

The string [user:name] is a **replacement pattern**. The available replacement patterns depend on the entity to which the paths are applied (content, terms and users), and on the fields created, etc. F22.24



NAME	TOKEN	DESCRIPTION
Current date		Tokens related to the current date and time.
Current page		Tokens related to the current page request.
Current user		Tokens related to the currently logged in user.
Random		Tokens related to random data.
Site information		Tokens for site-wide settings and other global information.
Users		Tokens related to individual user accounts.
Created	[user:created]	The date the user account was created.
Edit URL	[user:edit-url]	The URL of the account edit page.
Email	[user:mail]	The email address of the user account.
First name	[user:field_firstname]	Text field.
Last login	[user:last-login]	The date the user last logged in to the site.
Last name	[user:field_lastname]	Text field.
Name	[user:name]	The login name of the user account.
Original user	[user:original]	The original user data if the user is being updated or saved.
Picture	[user:picture]	The picture of the user.
Roles	[user:roles]	The user roles associated with the user account.
URL	[user:url]	The URL of the account profile page.
User ID	[user:uid]	The unique ID of the user account.

F22.24

Pathauto. Replacement patterns

In URL alias patterns you can use any replacement pattern that is available.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

We can indicate a default path pattern and individual patterns for each content type.

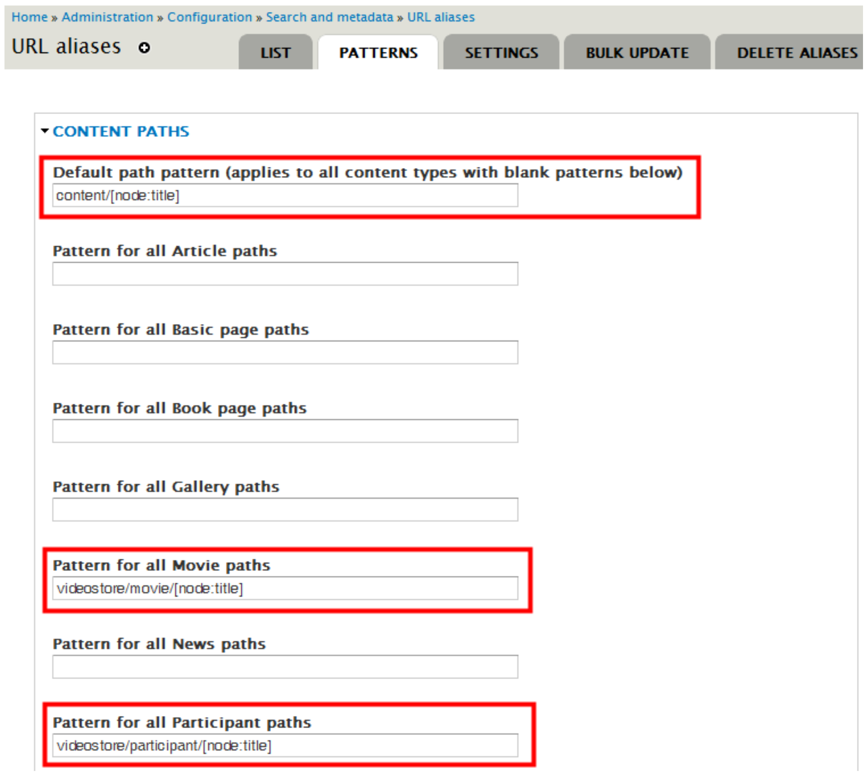
The default pattern for any node is **content/[node:title]**. It is advisable to modify or delete it as soon as we install **Pathauto**. Otherwise it will apply itself, as a massive update, to all contents that do not have a specific defined F22.25 pattern.

F22.25

Patterns for content paths

We have defined the patterns for Participant and Movie contents.

The default pattern for all nodes is ontent/[node:title]. It is advisable to modify or delete it as soon as we install **Pathauto**. Otherwise it will apply itself, as a huge update, to all contents that do not have a specific defined pattern.



The pattern that we will apply to **Movie** content type to generate the following URLs:

- videostore/movie/matrix
- videostore/movie/day-earth-stood-still

is:

- videostore/movie/[node:title]

because the movie title matches the node title.

The pattern we apply to **Participant** content type to generate the following URLs:

- videostore/participant/keanu-reeves
- videostore/participant/scott-derrickson

is:

- videostore/participant/[node:title]

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

because the participant name matches the node title.
If we want to generate the following URL type:

- videostore/movies/2008/day-earth-stood-still

we would use the pattern:

- videostore/movies/**[node:field_movie_year]**/**[node:title]**

Display the **Replacement patterns** tab and navigate through the available patterns to locate the most suitable pattern for the required path.

Automatic URL alias generation

URL aliases are generated automatically when we create a new content, taxonomy term or user.

On creating a content the **Generate automatic URL alias** option displays within **URL path settings**. It is activated by default. If we want to define a manual URL for a specific content, all we have to do is deactivate this option and fill out the **URL alias** field. F22.26

When we edit a content in the **URL alias** field it displays the automatically generated URL. We can also modify this URL by first deactivating the **Generate automatic URL alias** option.

F22.26
Generate automatic URL alias
The automatically generated URL alias displays when you create or edit a node.

Bulk update

From the **Bulk update** tab we update all the selected entity type paths (content, taxonomy terms and/or users). F22.27

Bear in mind that the URL alias update occurs following the defined patterns and the settings options indicated in **Settings**. If we have selected the option "Create a new alias. Delete the old alias” in **Update action**, all element´s URLs are will be updated and the old aliases deleted. This combination can be useful for reconverting all site contents once we have correctly defined all the URL alias patterns.

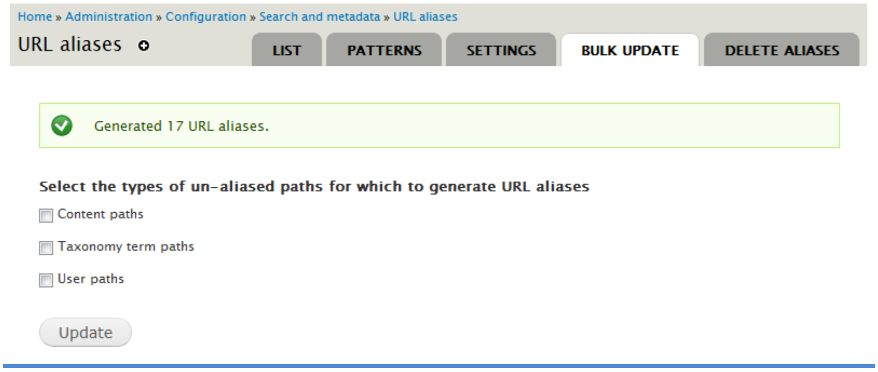
On carrying out a massive update (**Update button**) the system tell us how many URL aliases are generated.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F22.27

Bulk update

Selecting the entity types, all the aliases are generated automatically, following the specified configurations you have set.



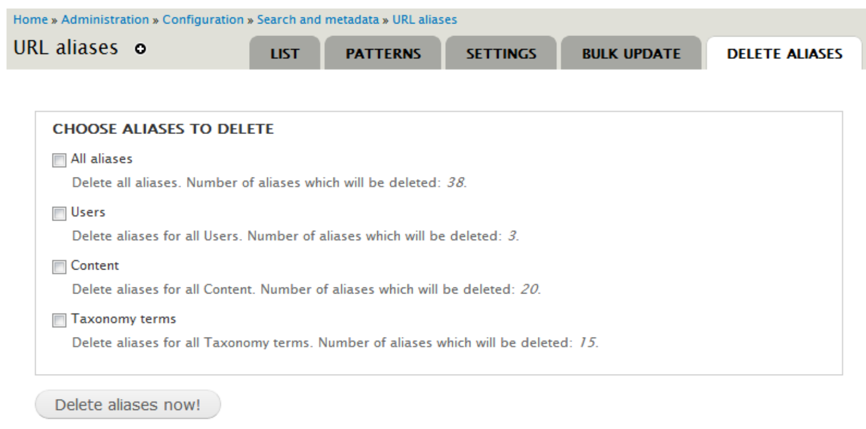
Delete aliases

From the **Delete aliases** tab we can delete all the aliases or only those which belong to a specific entity type (Users, Content and/or Taxonomy terms). **F22.28**

F22.28

Delete aliases

Allows you to delete URL aliases in batches, depending on the selected entity.



Field and node permission control 22.8

In this section we look at contributed modules that allow us to control content access.

With **Field Permissions**, we can control the individual permissions of the fields added to entities.

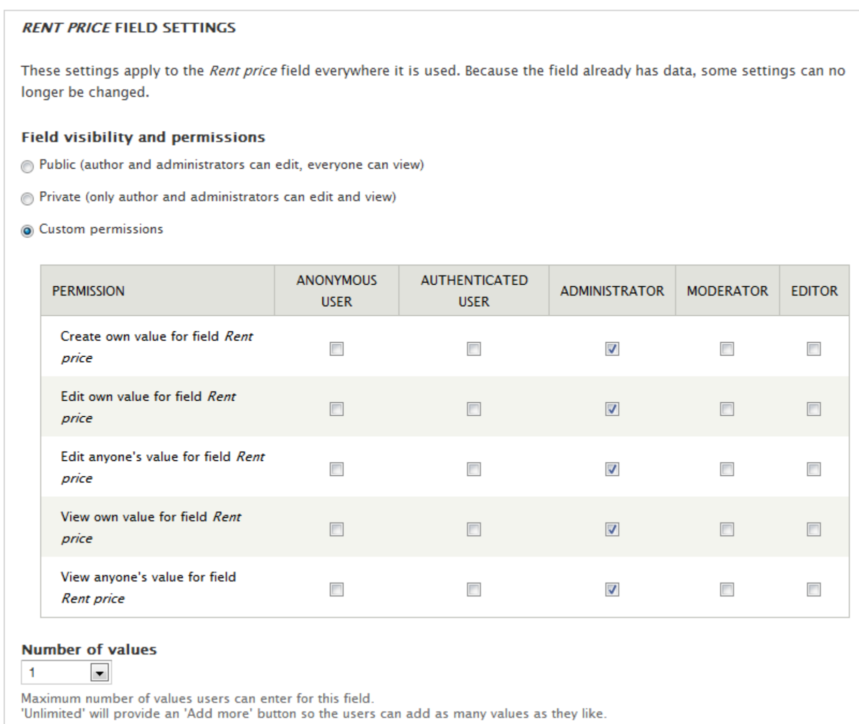
With **Taxonomy Access Control** we can control the access to contents via the node taxonomy terms.

Field Permissions module

The **Field Permissions** module allows us to control permissions for creating, updating and/or seeing entity fields.

http://drupal.org/project/field_permissions

Once installed and activated we can define individual permissions for each field. We can do this from the field Edit option, accessing the corresponding entity. F22.29



RENT PRICE FIELD SETTINGS

These settings apply to the *Rent price* field everywhere it is used. Because the field already has data, some settings can no longer be changed.

Field visibility and permissions

- Public (author and administrators can edit, everyone can view)
- Private (only author and administrators can edit and view)
- Custom permissions

PERMISSION	ANONYMOUS USER	AUTHENTICATED USER	ADMINISTRATOR	MODERATOR	EDITOR
Create own value for field *Rent price*	☐	☐	☑	☐	☐
Edit own value for field *Rent price*	☐	☐	☑	☐	☐
Edit anyone's value for field *Rent price*	☐	☐	☑	☐	☐
View own value for field *Rent price*	☐	☐	☑	☐	☐
View anyone's value for field *Rent price*	☐	☐	☑	☐	☐

Number of values

1

Maximum number of values users can enter for this field.
'Unlimited' will provide an 'Add more' button so the users can add as many values as they like.

F22.29
Field permissions
With the Field Permissions module you can establish individual permissions for each field.

The field can be set up as:

- **Public**. The field can be edited by all users with administrative permissions. Any user can view the field.
- **Private**. Only the author and the administrators can view and edit the field.
- **Custom permissions**. We can establish, for each site role, the following permissions:

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Create own value for field *Field-name***. Allows the users to create fields of this type. The users need permissions to create the corresponding entity. If a user has permissions for creating an entity but not to create their own field value, the field does not display.
- **Edit own value for field *Field-name***. Allows the users to edit their own field.
- **Edit anyone's value for field *Field-name***. Allows the users to edit the field value created by any other user.
- **View own value for field *Field-name***. Allows the users to see the fields that they create.
- **View anyone's value for field *Field-name***. Allows the users to see the field value created by any other user.

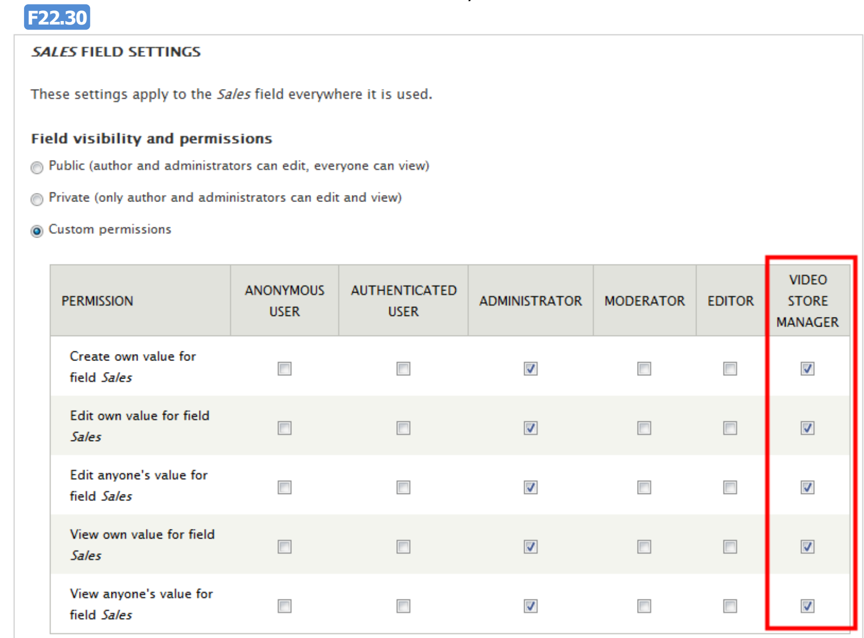
As an example we will add a new integer field, called **Sales** (field_movie_sales), to the **Movie** type content. This field stores the number of sales of the movie.

We create a **video store manager** role with the following permissions:

- Create, edit and delete Movie (and Participant) type contents. We establish the permissions corresponding to the Node module.
- Create, edit and view the **Sales** field. The Sales field cannot be seen by any other site user except administrators.
- Administer (**Media**) and **Media Gallery** multimedia contents.
- Add multimedia contents from remote services (**Media Internet Sources**).

With this configuration a user with the **video store manager** role can create video store content and edit the **Sales** field, which is hidden from all other users. F22.30

F22.30
Sales field permissions
The video store manager role can create video store contents and edit the Sales field, which is hidden from all other users.


SALES FIELD SETTINGS

These settings apply to the *Sales* field everywhere it is used.

Field visibility and permissions

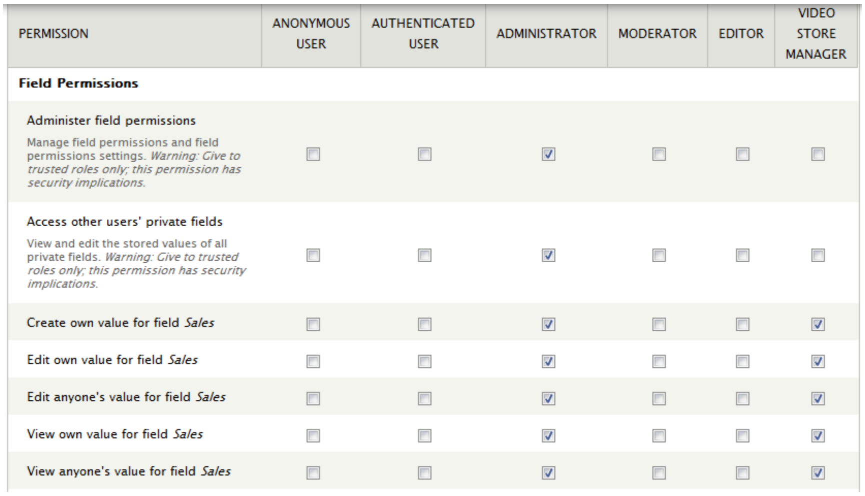
- ( ) Public (author and administrators can edit, everyone can view)
- ( ) Private (only author and administrators can edit and view)
- (•) Custom permissions

PERMISSION	ANONYMOUS USER	AUTHENTICATED USER	ADMINISTRATOR	MODERATOR	EDITOR	VIDEO STORE MANAGER
Create own value for field *Sales*	☐	☐	☑	☐	☐	☑
Edit own value for field *Sales*	☐	☐	☑	☐	☐	☑
Edit anyone's value for field *Sales*	☐	☐	☑	☐	☐	☑
View own value for field *Sales*	☐	☐	☑	☐	☐	☑
View anyone's value for field *Sales*	☐	☐	☑	☐	☐	☑

In addition to the field edition, the **custom permissions** of each field are also available in the site permissions administration page, within the corresponding permissions of the **Field Permissions** module.
The module also adds two fixed permissions: F22.31

- **Administer field permissions**. Administer field permissions.
- **Access other users' private fields**. Access other users' private fields.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



F22.31

Field Permissions module permissions

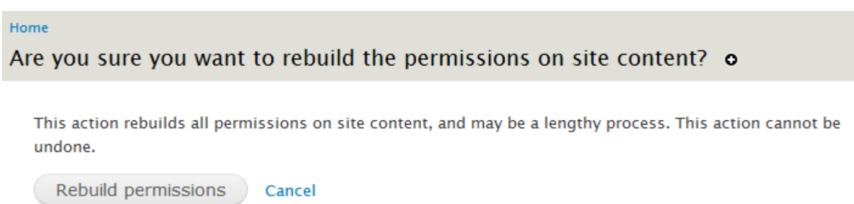
The specific permissions of each field are shown within the module permissions, next to the site permissions.

Taxonomy Access Control module

With **Taxonomy Access Control** we can control the access to contents using the node's taxonomy terms. This module is available at:

http://drupalmodules.com/module/taxonomy-access-control

On installing and activating the module the system requests **Rebuild permissions** (The content access permissions need to be rebuilt). This warning message includes the link we click on to rebuild the permissions (/admin/reports/status/rebuild). F22.32



F22.32

Rebuild permissions

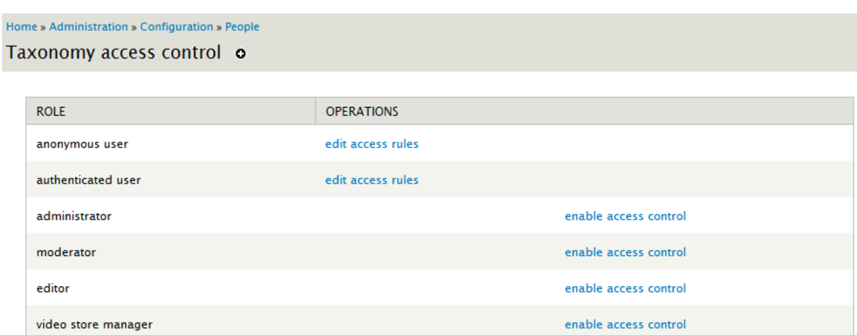
Rebuild permissions request.

For configuring the module permission we must go to:

Administration ⇨ Configuration ⇨ People ⇨ Taxonomy access control

URL Taxonomy Access Control
/admin/config/people/taxonomy_access

The roles for those who already have taxonomy access control enabled show the link **edit access rules**, while for the others, we must first enable it (*enable access control* link). F22.33



F22.33

Taxonomy access control

Node access control via the taxonomy terms associated with the contents.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

When editing each role we can assign, to each term, the options **A** (Allow), **I** (Ignore), **D** (Deny), to the following permissions:

- **View**. The users can see nodes containing this term.
- **Update**. The users can edit nodes containing this term.
- **Delete**. The users can delete nodes containing this term.

We can also activate and deactivate these permissions:

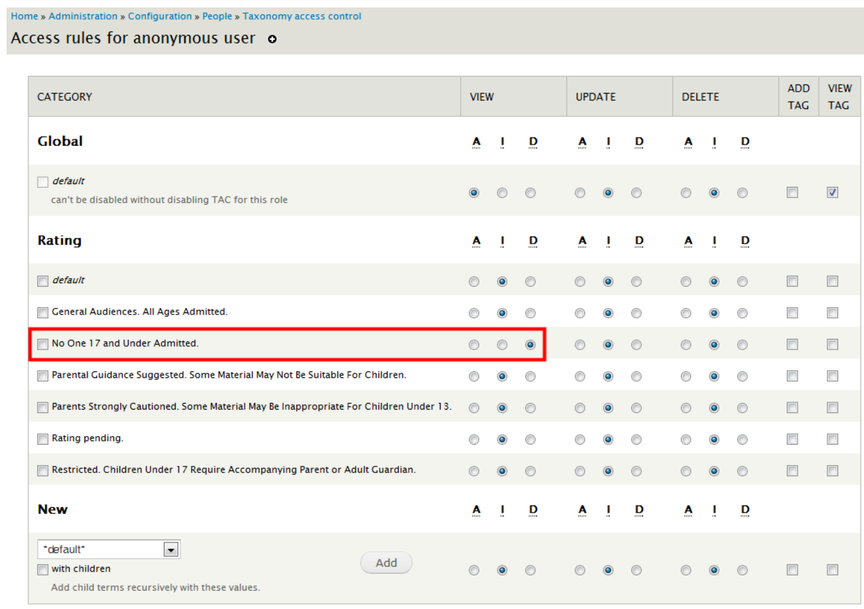
- **Add tag**. The tag is available to be assigned to the node, when this is created or edited.
- **View tag**. The tag is displayed in the terms lists and nodes. This does not necessarily mean that the user has access to the nodes. This permission depends on **View**.

As an example we have added the term "Not suitable for children under 17" from the **Rating** vocabulary and configured the **anonymous user** role to stop it from seeing movies with this rating. (value **D** in the **View** field). To allow other users to view the contents with this term we do not need to configure the settings, as they take the values by default. F22.34

F22.34

Taxonomy permissions

We can add taxonomy terms and, for each site role, establish the appropriate permissions.



Taxonomy permissions configuration can be more or less complicated depending on the access control level required for the site.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Additional modules

22.9

In this section we briefly introduce other contributed modules that act on contents and on entities in general.

Countries module

The **Countries** module has a range of functionalities relating to countries. It consists of the **Countries** and **Countries configuration** modules, and is available at:

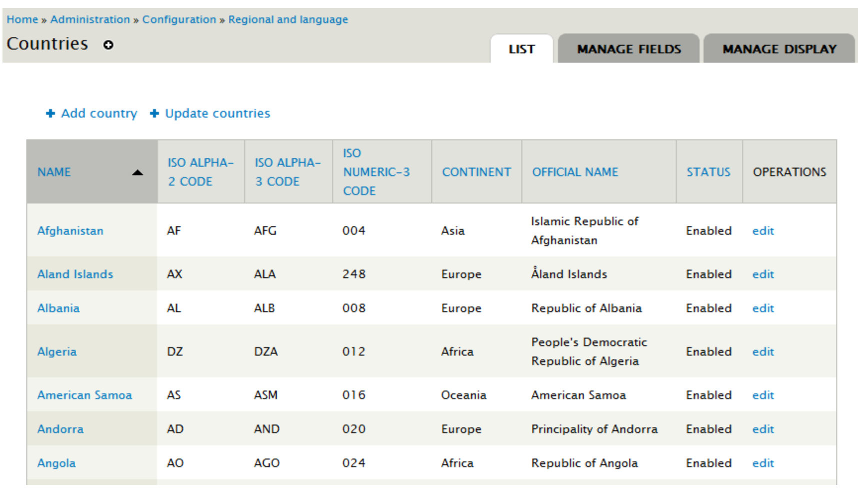
http://drupal.org/project/countries

Access to module configuration:

Administration ⇨ Configuration ⇨ Regional and language ⇨ Countries

URL Countries
/admin/config/regional/countries

Countries are treated as an entity, which means that we can manage fields and their display. F22.35



F22.35
Countries module
The Countries module includes a database of countries and a set of functionalities for use in the site.

The module also adds a new field type, called **Country**. This field allows us to add a country selection list of all the countries available in the database to any entity.

The specific configuration options of the **Country** field are: F22.36

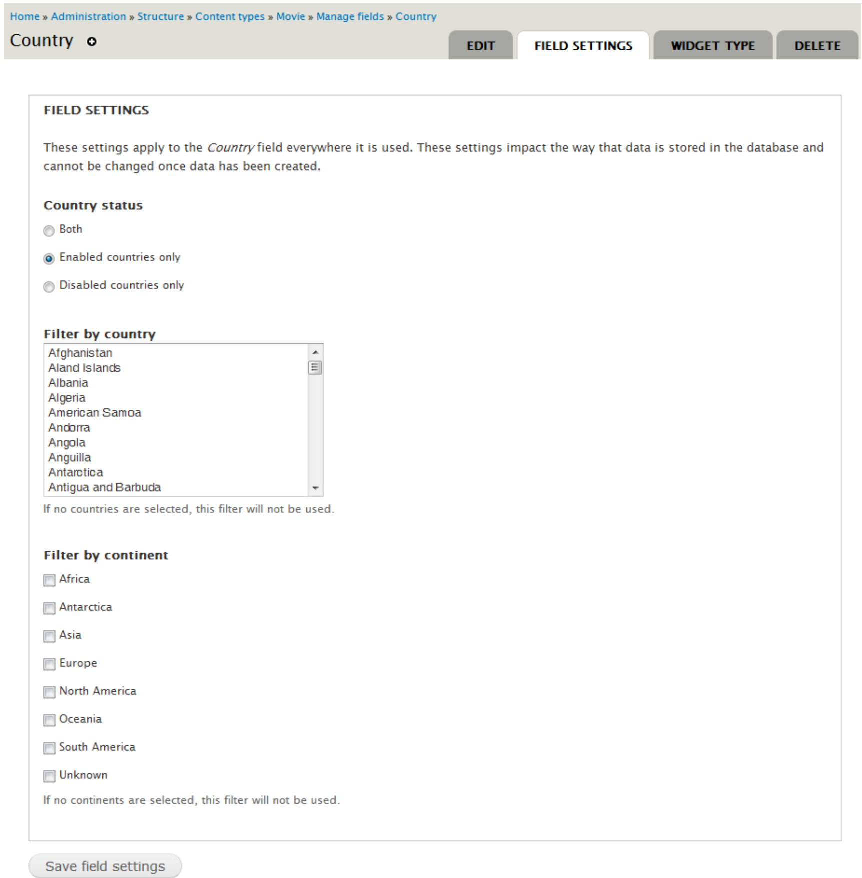
- **Country status**. We can choose between:
  - Both (all countries in the database).
  - Enabled countries only.
  - Disabled countries only.
- **Filter by country**. We can select the countries displayed in the list.
- **Filter by continent**. We can select the continents. Only countries from selected continents are displayed.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F22.36

Country field

In the Country field you can filter by country and/or continent.



CCK List module

The CCK List module adds a field that transforms the inputted content into a HTML list.

The module is available at:

http://drupal.org/project/cck_list

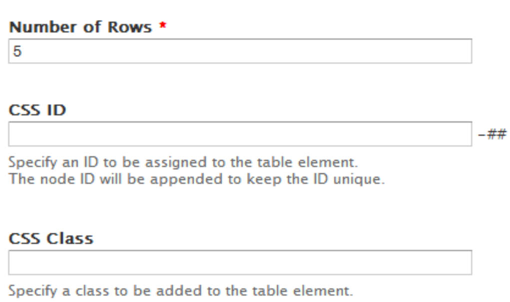
Once installed we find it in the modules list with the name **List**, within the CCK modules group. The **List** field that is now available should not be confused with the list type fields that we have already covered, and which allow us to choose between a list of elements. The **List** field added by the **CCK List** is a **Text area** where the user who creates the node can add a list of elements (one per line). These elements are displayed in the node as a list of HTML elements, applying the installed theme styles.

F22.37

CCK List module

Allows us to create a list that displays as an HTML list.

You can apply styles by indicating an ID or CSS Class.

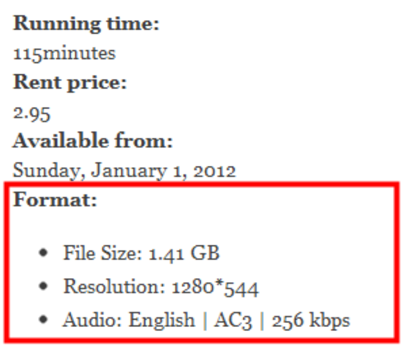


Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Once the field is added we can configure: F22.37

- **Number of Rows** in the text area.
- **CSS ID**. Allows us to specify an ID that is used to apply styles in an HTML list.
- **CSS Class**. Allows us to specify a CSS class that is used to apply styles to an HTML list.

As an example we have added a **Format** field (field_movie_format) in the **Movie** content type. Here we add a list of technical specifications from the movie file, as text. F22.38



F22.38
List
Input the list into a text area field. It displays as an HTML list.

Node Clone module

The **Node Clone** module allows us to make a copy of any node created in the site.

The module is available at:

http://drupal.org/project/node_clone

Although we can clone complex nodes that contain additional fields, it is necessary to check that the content type cloning process have finished without any errors.

Module configuration:

Administration ⇨ Configuration ⇨ Content authoring ⇨ Node clone module

URL Node clone
/admin/config/content/clone

Some of the most important options are:

- **Method to use when cloning a node**. Allows us to select between:
  - **Pre-populate the node form fields**. Fills out the node form fields. The node is only created if saved.
  - **Save as a new node then edit**. Creates the node and then opens if for editing.
- **Confirmation mode when using the "Save as a new node then edit" method**. Allows us to select between:
  - **Require confirmation**.
  - **Bypass confirmation**.
- **Clone menu links**.
- **Reset publishing options to defaults**. We choose the content types that we want the option to apply to.
- **Omitted content types**. If we see that some content types are not clonable with **Node clone**, we omit them by activating them in this line. It is important to identify the content types that do not clone correctly.

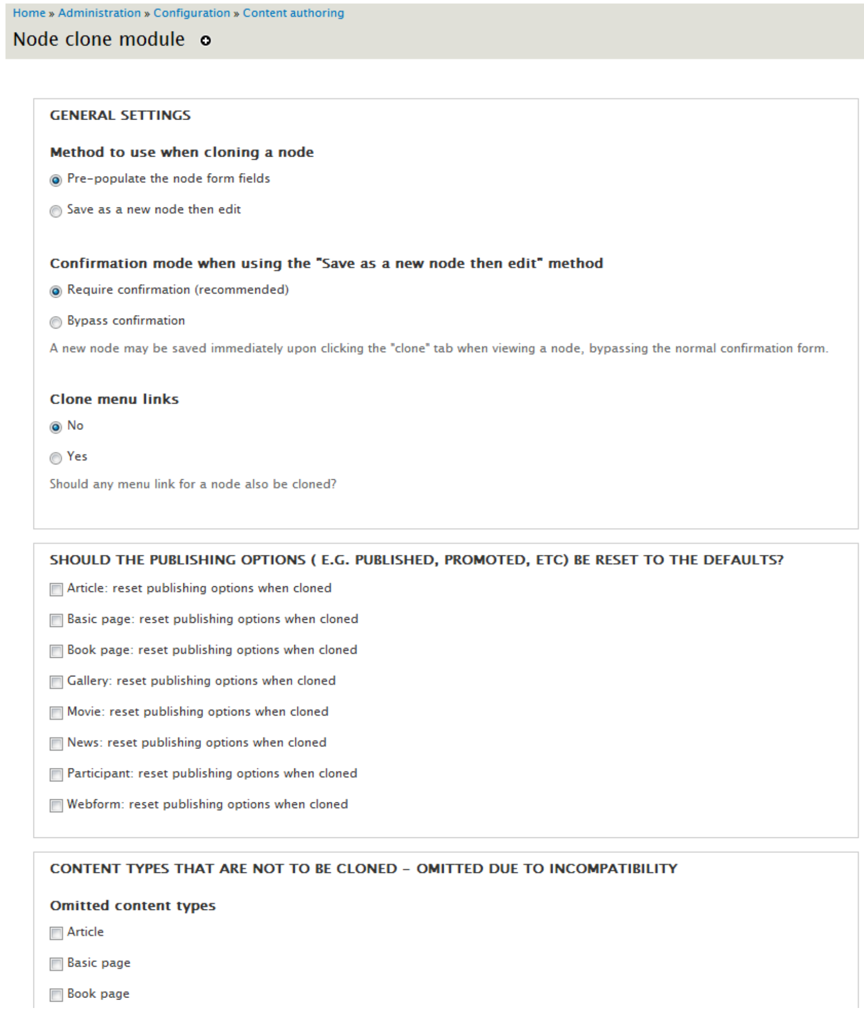
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F22.39

Node clone module

Node clone module´s configuration options.

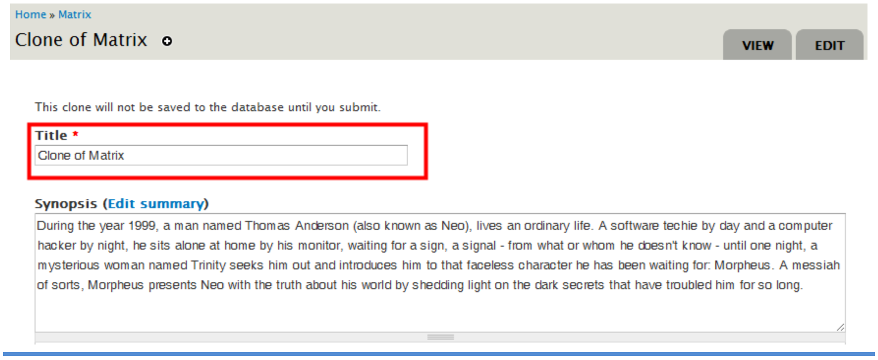
On cloning a node we can opt to fill out the fields without saving the node, or create and save the node before opening it for editing.



When the user has the required permissions to clone nodes (see node permissions), a "Clone content" link displays in clonable nodes.

Clicking on the link brings up a new node, which depending on the method used, may or may not be saved. Either way a node edit form displays, allowing us to make the required modifications. The cloned node title begins with "**Clone of**", followed by the original node title. F22.40

Upon cloning a node we must take care to modify the URL alias because two nodes can´t have the same URL alias.



F22.40

Node clone

The cloned node title begins with "**Clone of**", followed by the original node title.

Use the Node clone module to clone any content type. Always check that the process works correctly, especially when the node contains content special fields.

This module is particularly useful when we need to clone contents with very similar values. For example, we can clone forms created with Webform, saving a lot of time when replicating forms that only require minor modifications.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

23 Views

The **Views** module is probably the most useful tool for creating dynamic pages (called **views**) fed with content from the site.

To fully understand what can be achieved by using views, consider these examples. With views we can create:

- A page which displays all the nodes with a particular content type. Also, we can add search filters that help the user to find interesting content.
- A block containing a list of the latest users registered in the site.
- A data table with sortable fields. Data can be extracted, for example via a set of nodes containing a particular content type.
- A block which contains the most visited content.
- A grid which contains a selection of images.

In this unit we will study the **Views** module in depth, using step-by-step practical examples as well as theoretical study of the different functionalities and configuration options that this module offers. We will also look at extra modules that add further functionalities to views.

Comparative D7/D6

Views 3.x

Drupal 7 contains the 3.x version of the Views module. There are important changes compared to Drupal 6 and the 2.x version.

In Drupal 7, along with changes to the interface, you find different available fields and configuration options.

Unit contents

23.1 Views module introduction....................................76
23.2 View creation...80
23.3 Basic settings...84
23.4 Fields..92
23.5 Sort criteria..97
23.6 Filter criteria...100
23.7 Contextual filters..111
23.8 Relationships...123
23.9 Other advanced options.......................................127
23.10 Permissions..128
23.11 Extra modules...129

23

23.1 Views module introduction

The **Views** module is probably the most useful and powerful tool for building dynamic pages (called **Views**) that contain site content.

To fully understand what can be achieved with views, consider the following examples. With views we can create:

- A page that displays all the nodes of a particular content type. We can also add search filters that help the user to find interesting content.
- A block containing a list of the latest registered users in the site.
- A data table with sortable fields. The data can be extracted, for example via a set of nodes containing a particular content type.
- A block containing the most visited content.
- A grid containing a selection of images.

Returning to our **online video store** application, we can add the following views.

- A list of the latest movies added to the video store.
- A list of the most popular movies.
- A list of upcoming movie premieres.
- A list of all movies featuring an actor or director.
- A grid displaying the price of the latest movies added to the video store.

The **Views** module combines perfectly entities with fields, allowing us to increase the number of display options for dynamic content.
The **Views** module is available at:

http://drupal.org/project/views

The module is made up of two submodules:

- **Views**. The main module, needed to create views.
- **Views UI**. Provides the user interface required to administer views. Without this module we cannot create or edit views.

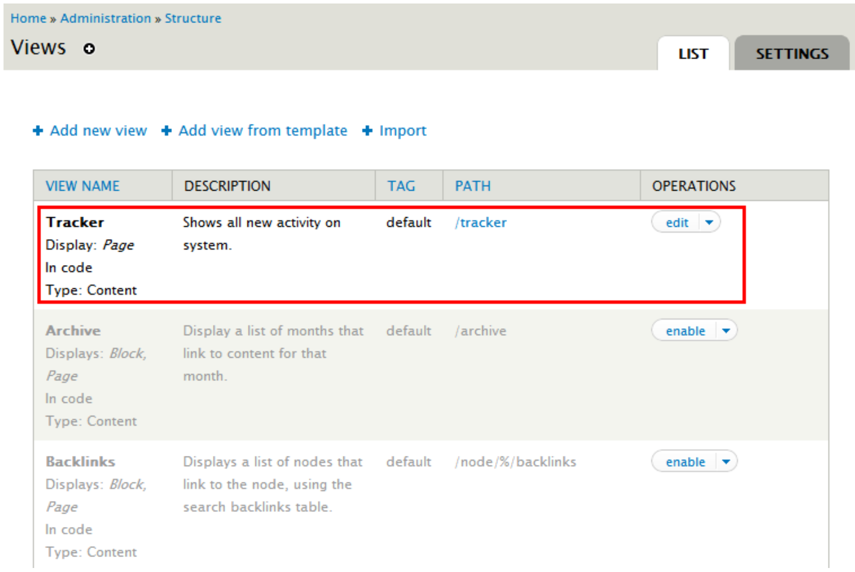
To start working with views we first need to activate both modules. We make the configuration from:

URL Views
/admin/structure/views

Administration ⇨ Structure ⇨ Views

In this list some views appear as deactivated. These views come from other installed modules and can be activated and used in the website. However, for now we focus on creating new views, leaving the others deactivated. F23.1

To see a view example we activate Tracker, one of the predefined views. This view creates a page with the URL alias **/tracker** which displays a list of the latest content published on the site (**Recent posts**).



VIEW NAME	DESCRIPTION	TAG	PATH	OPERATIONS
Tracker Display: *Page* In code Type: Content	Shows all new activity on system.	default	/tracker	edit
Archive Displays: *Block*, *Page* In code Type: Content	Display a list of months that link to content for that month.	default	/archive	enable
Backlinks Displays: *Block*, *Page* In code Type: Content	Displays a list of nodes that link to the node, using the search backlinks table.	default	/node/%/backlinks	enable

Once a view is activated, we go straight to the page via the view URL, **/tracker**. When loading this page the F23.2 system sometimes displays a **Page not found error**. This due to the system cache and can be fixed in two ways:

- **Edit and save the view** (edit one view option, even if no changes are made, to activate the **Save** button).
- **Clear all caches** from the **Performance** configuration option.

Type	Title	Author	Replies	Last Post
Movie	Street Kings	admin		Sat, 06/02/2012 - 13:41
Movie	The Social Network	admin		Sat, 06/02/2012 - 13:38
Movie	The Devil's Advocate	admin		Sat, 06/02/2012 - 13:36
Movie	Zombieland	admin		Sat, 06/02/2012 - 13:32
Participant	David Ayer	admin		Sat, 06/02/2012 - 13:25
Participant	Taylor Hackford	admin		Sat, 06/02/2012 - 13:24
Participant	David Fincher	admin		Sat, 06/02/2012 - 13:24

F23.2
Recent post view
Example of a view that displays the latest content Published on the site.

Settings

Before learning how to create **Views**, we access the **Settings** tab to review some of the general configuration options that affect views management. Some of the basic settings are: F23.3

- **Show filters on the list of views**.

- **Show advanced help warning**. A module called **Advanced help** allows developers to include advanced help in the modules. If we do not install the module, **Views** continually tells us about it, as **Advanced help** gives us more detailed help for views. To avoid this message from displaying constantly, either install?? **Advanced help** or deactivate this

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

option.

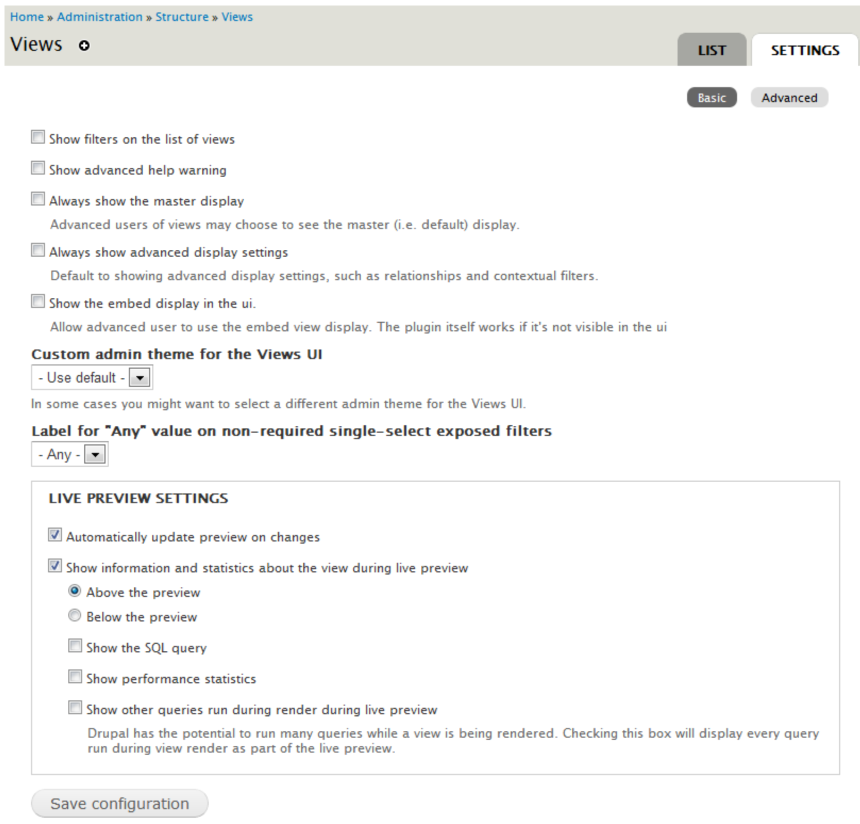
- **Label for "Any" value**. The value that displays in single selection lists. The value <Any> is not translatable so we must choose another available value, such as, - Any -, if we want another language to display in the lists.

- **Live preview settings**. Allows us to configure the behaviour of the **Preview**. For example, we can configure it to display the generated SQL query.

F23.3

Views. Basic settings

In the tools tab you can configure some general options that affect views administration.



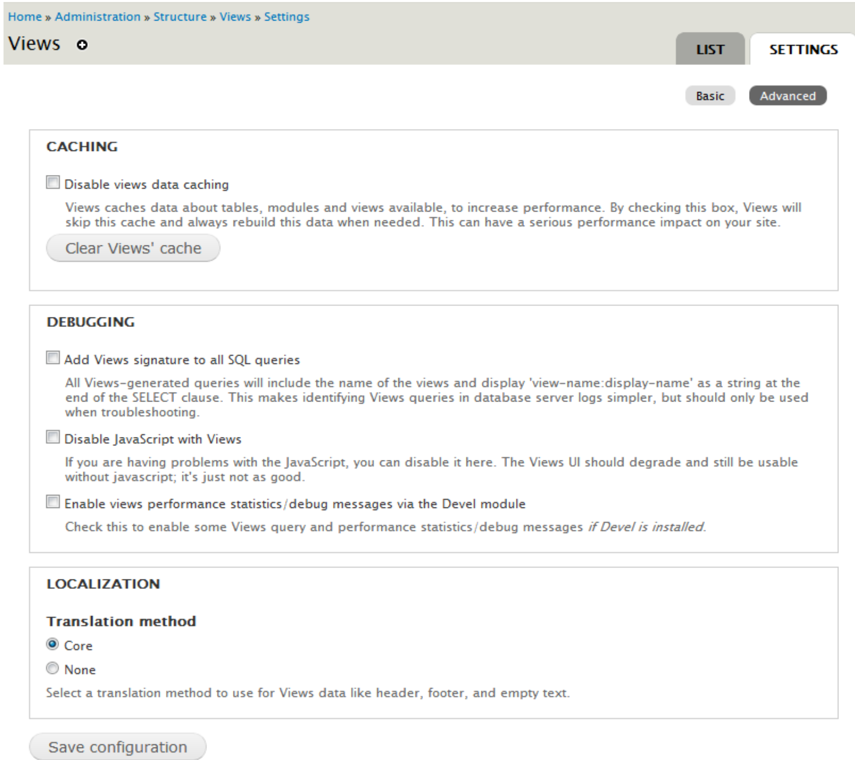
Among the advanced options (Advanced tab) we have: F23.4

- **Disable views data caching**. This option can affect site performance so we recommend not activating it. It is advisable to deactivate it during site development to avoid cache- and refresh-related problems.

- **Clear Views' cache**. Clears all information stored in the cache to speed up views. We use this option when changes applied to a view do not show up on the final page generated by the view.

- **Debugging settings**. During the implementation of views we may need to activate some of these options. The interface uses JavaScript, which can be deactivated via this option. The editing area of views continues to work afterwards, but without the added functionalities that Javascript offers.

- **Localization**. Allows us to select a translation method. We use the default setting offered by the core.



F23.4

Views. Advanced settings

From Advanced settings you can access other options such as Clear Views' cache.

23.2 View creation

URL Add new view
/admin/structure/views/add

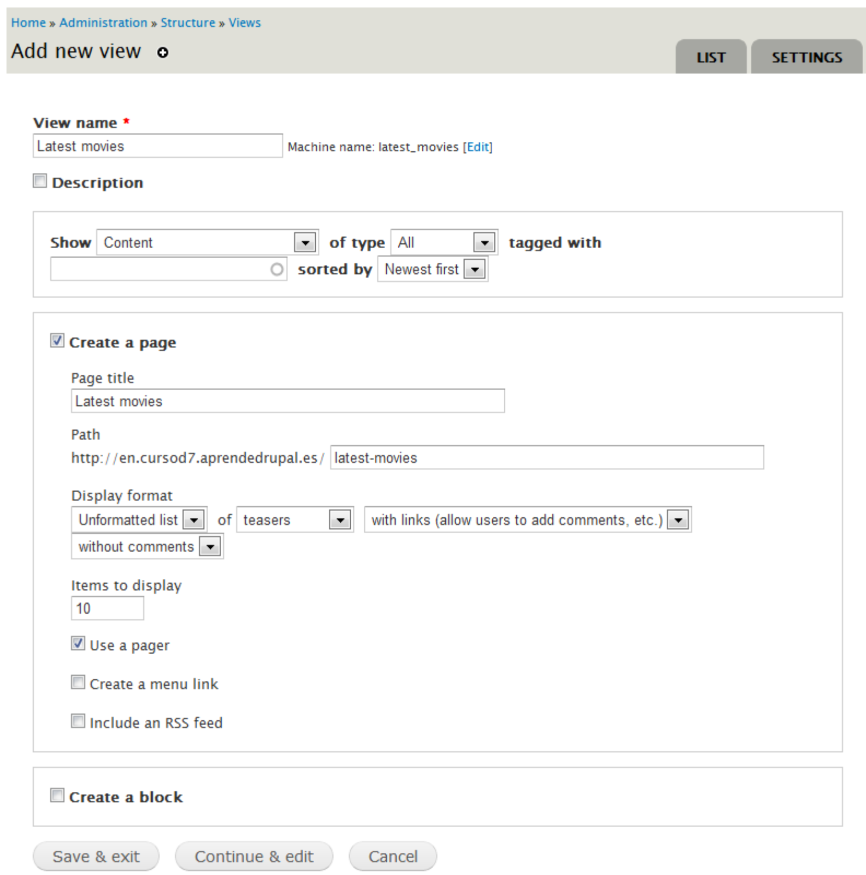
To create a new view we go to: F23.5

Administration ⇨ Structure ⇨ Views [Add new view]

To define the view we must indicate which element types will be displayed (content, users, comments, taxonomy terms, etc.). The fields available later on depend on this selection.

We also chose whether the view will be displayed on a page and/or in a block, which are the two default display types.

F23.5
Add new view
Upon creating a new view you can chose between a wide range of options. These can be modified after editing the view.



At this first step we complete the following options:

- **View name**. The view name shown in the administration lists. It can contain any characters.
- **Machine name**. The name that the system displays when the view name is completed. Special characters, such as accents are substituted for underscores. This may mean that we have to edit and rewrite the system name.
- **Description**. Adds a descriptive text.

- **Show**. In this section we chose the element type (Content, Files, Users, etc.). Depending on the selected type we can build different views, as what we are choosing is the main element that the view uses for database SQL queries. This choice is vital as the configuration options and available fields depend on the view type that we select. Once we chose a view type it cannot be modified. A mistake means we start building the view again from scratch. Some of the view types available for selection are:

    - **Content**. Allows us to work on any content type or site node.
    - **Files**. Allows us to create a view based on files uploaded to the site.
    - **Comments**. Creates a view with comments published in the nodes.
    - **Locale source**. Allows us to create a view with translation texts.
    - **Content revisions**. Allows us to create a view based on node revisions.
    - **Taxonomy terms.** Creates a view that uses taxonomy terms associated with the nodes.
    - **Users**. Allows us to create a view for displaying the users registered on the site.

Installed modules may add new element types that can be used in views, such as:

- **Webform Submissions**. This type is added by the **Webform** module and allows us to create a view based on form submissions and answers.
- **Countries**. Added by the **Countries** module and allows us to create a view based on the countries stored in the database.

Depending on the type selected, additional configuration options are displayed. For example, if we choose **Content**, the view allows us to choose a specific content type or even a taxonomy term to filter the results. These extra configuration options can be modified later on.

- **Create a page**. Using this option we will indicate whether the view is presented on a page. We can configure, among other options, the title, the URL alias and the display format. Later on we cover what display formats are available.

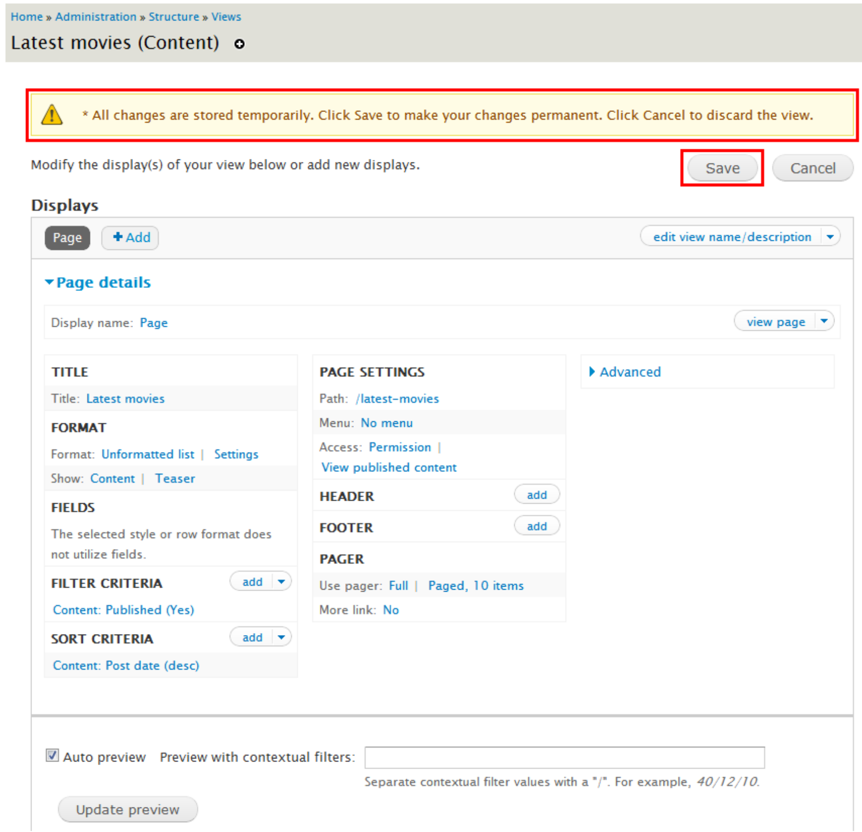
- **Create a block**. Using this option we will indicate whether the view is presented in a block. We can configure the title, the display format and the number of elements per page.

Once these fields are completed we click on **Continue & edit** to move on to view editing. The view is not stored as a final version until we click on **Save** in the edit window.

F23.6

Views. Editing

The fields are not saved until we click on the Save button.



The edit window of a view is made up of the following sections or configuration blocks. We study these in detail in the following sections. F23.6

- **Save and Cancel buttons**. Displayed in the top right-hand corner and only when changes are available to be saved. The Cancel button removes any changes made since the last save.

- **Displays**. The available displays are shown in tabs. The **+Add** button allows us to add new displays to the view. For example, the same view can generate a page and a block (or several of these displays). In fact, the view cannot be used unless we create at least one display. The most frequently used displays are page and block, although we can also choose others, such as RSS news channel.

- **Actions bar**. Next to the **Edit view name/description** option there is a “hidden” toolbar with extra functionalities. We display it by clicking on the right hand arrow. The extra options are: **analyze view, clone view, export view, reorder displays** and **delete view**.

  - **Edit view name/description**. Allows us to modify the **View name** and the **Description**. Field. We can also add a **tag** to categorize the view in the administration pages.
  - **Analyze view**. Finds potential errors in the view.
  - **Clone view**. Generates a complete copy of the view.
  - **Export view**. Generates PHP code which we use to export the view to another web page or to include it in a module (only for advanced users).
  - **Reorder displays**. Reorders the displays within the view.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Delete view**. This option is available once we save the view.

- **Display details**. Allows us to configure the selected display. On the right of the Display name there is a new pop-up menu with functions that can applied to the display: View display, Clone display, Delete display and Disable display.

- **Preview**. Under the view edit area we can preview of the current display. We can choose auto preview, which displays any added field in the refreshed preview. We can also create a preview by clicking on the **Update preview** button.

Within the **Details** of each display, we find configuration options grouped together in:

- **Basic settings**. A set of options that allow the display to be configured (**field 23.3**).

- **Fields**. Here we add fields that are shown in the view (**field 23.4**).

- **Filter criteria**. Allows the results displayed in the view to be filtered. It is also possible to make form elements (displayed filters) available to the user. With these the user can filter the results. (**field 23.5**).

- **Sort criteria**. Defines the criteria by which the elements displayed in the view can be sorted (**field 23.6**).

Expanding the **Advanced** tab (in the **Details** of each display) displays advanced configuration options:

- **Contextual filters**. Allows us to add filters based on context (fields, URL parameters, current user, etc.) (**section 23.7**).

- **Relationships**. Increases the number of queries in a view by adding fields from other related Drupal elements. For example, for a view that displays nodes we can establish a relationship with the user who created the nodes to display fields related to the users. (**section 23.8**).

- **Other advanced settings**. In **Other** we see other configuration options for the display (**field 23.9**).

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

23.3 Basic settings

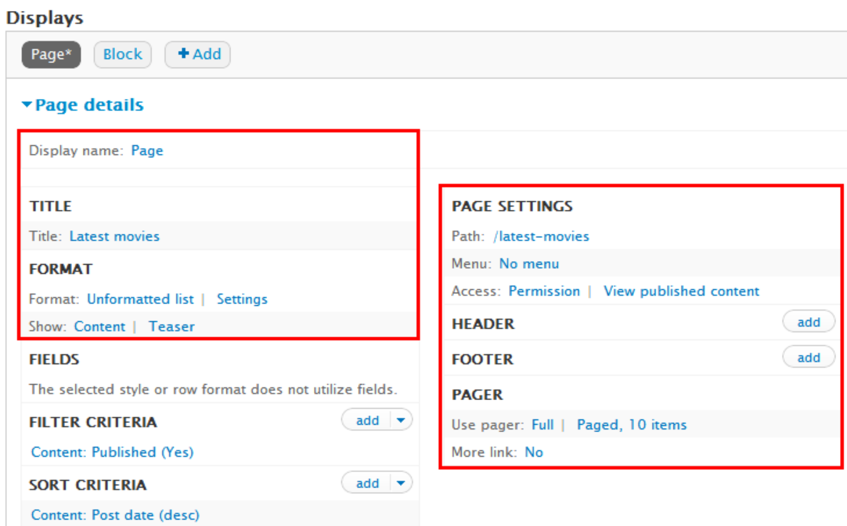
In this section we present the basic options that can be applied to each display in the view. We also look at **Page settings** and **Block settings**, depending on the type of display. F23.7

F23.7

Basic settings

Configuration options for the selected display.

Some of these options depend on the type of display (Page, block, etc).



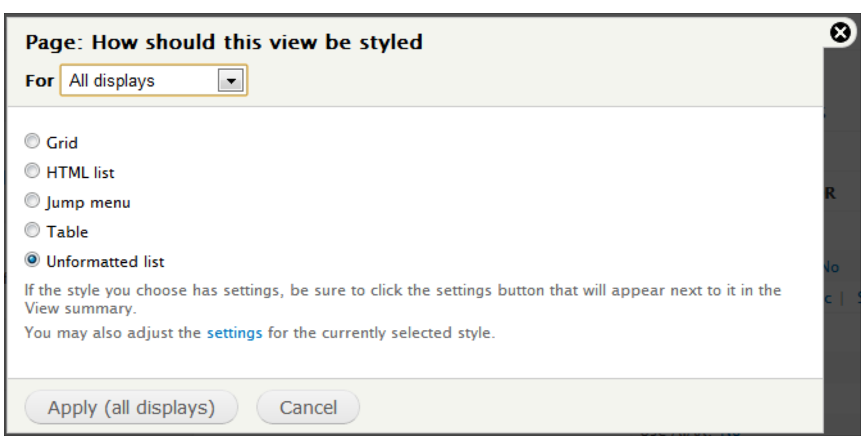
- **Display name**. This is the name associated with the display. It is only used in the view administration interface. It is advisable to change the default name to another one related to the content of the view (List of latest movies, Latest users, etc.).
- **Title**. The title displayed in the view. Depending on the display type it can be a page title or a block title, etc.
- **Format**. The following formats and styles are initially available: **Grid, HTML list, Unformatted list, Jump menu** and **Table**. Once the **Format** is selected we can access its settings by clicking on the **Settings** link that is displayed to its right. Below we see an example of each display format and the options available for each one. F23.8
- **Show**. Allows us to choose between **Fields** or **Content**.

F23.8

Views formats

The available formats are: Grid, HTML list, Unformatted list, Jump menu and Table.

As you can see, installing other extra modules adds new format types and styles to the list.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Grid format

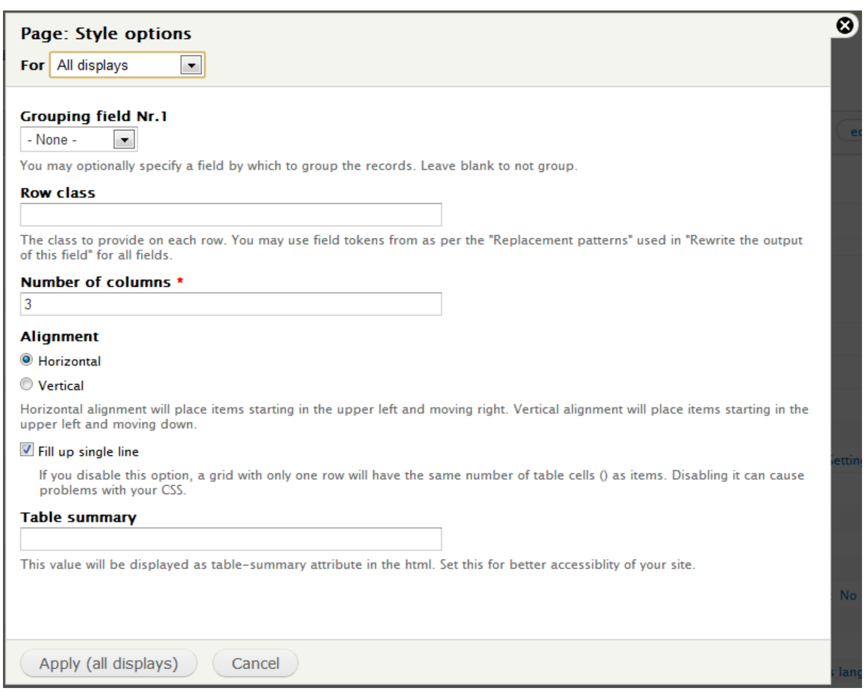
Creates a grid of elements. It is Useful for displaying elements in several columns/rows, such as a gallery of photos. In the **Figure** we see a 3-column grid showing the **Latest movies**. F23.9

F23.9
Grid format
Shows the results as a grid.

In the grid format we can configure the number of columns and the layout. If we choose a horizontal layout the view elements are aligned from left to right (and from top to bottom), while if we choose a vertical layout the elements are aligned from top to bottom (and from left to right). F23.10

The **Fill up single line** fills up the row boxes even if it does not have all the elements.

We can also choose a type for each row, with which we can apply styles from the CSS style sheet. Finally, the **Table summary** completes the table-summary attribute in the HTML code.



Grid format. Settings
You can choose the number of columns and the alignment.

HTML list format

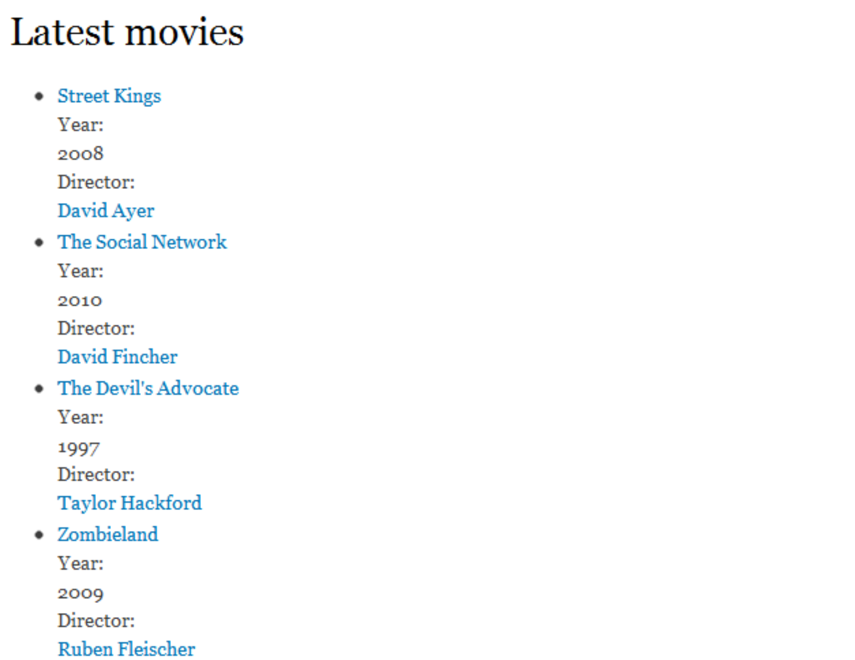
Creates a list of elements using the corresponding HTML tags (<ul>, <li>, etc.). We can choose whether the list is ordered (<ol>: 1, 2, 3 ...) or without a defined order (<ul>). We can also use one of the defined fields to create register bundles and add CSS classes to different elements (row, list and list element).

F23.11 F23.12

F23.11

HTML list format

Displays the results as an HTML list.

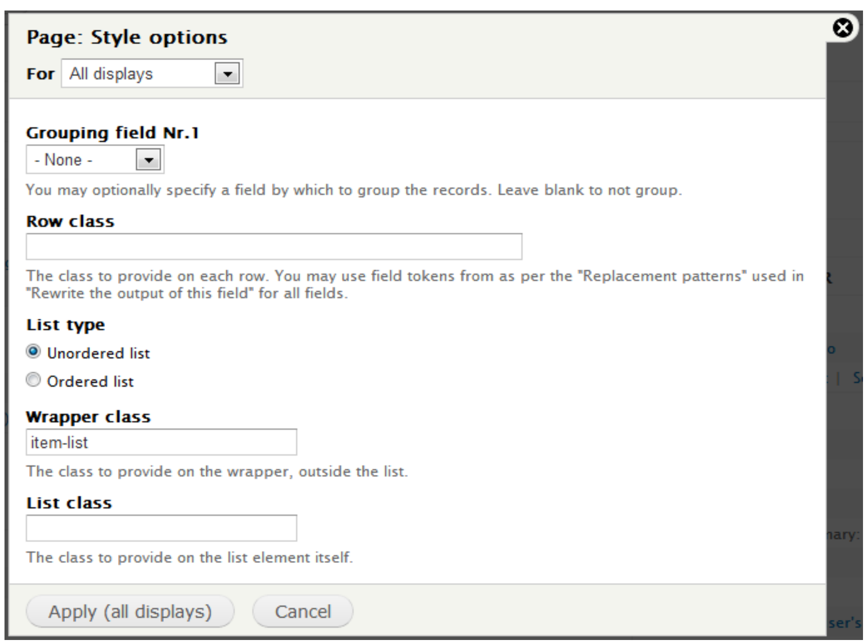


F23.12

HTML list format. Settings

The list can be an ordered list (OL) or an unordered list (UL).

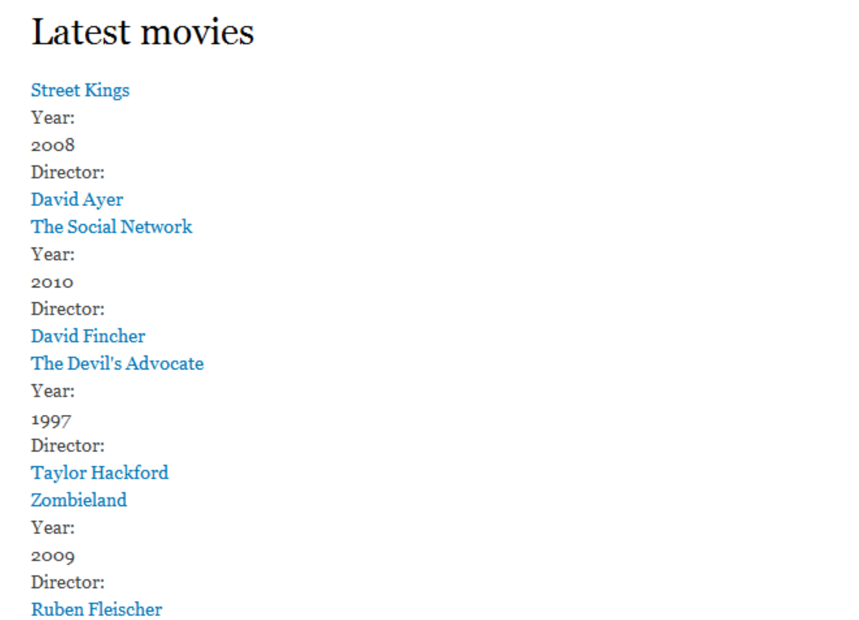
You can also decide the CSS class applied on each list element (ul, li, etc.).



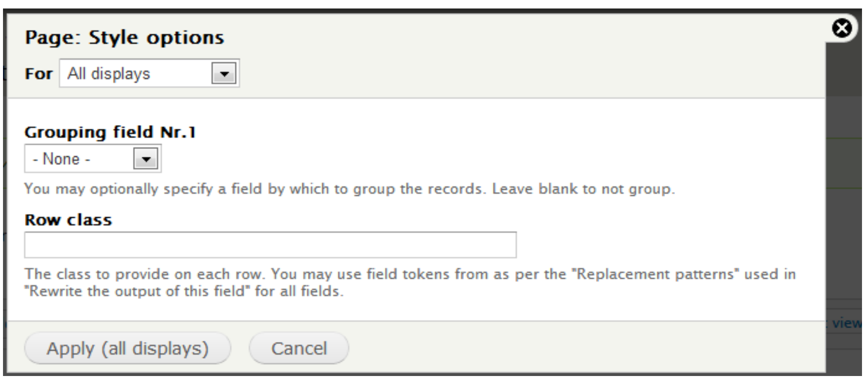
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Unformatted list format

Creates an unformatted display using <div> tags. We can apply CSS styles using the generated <div> tags. F23.13 F23.14



F23.13
Unformatted list format
Displays the results with DIV tags. The styles can be applied from the CSS style sheet.



F23.14
Unformatted list format. Settings
For this format you can only indicate the CSS class that is applied to each row.

Jump menu format

Displays an element selection field with the view results. The user can select which element they want to **Go** (or jump) to. F23.15 F23.16

F23.15
Jump menu format
Displays a content selection list and a button to load the selected content.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F23.16

Jump menu format. Settings

Allows you to select the field where each element path is stored. This is the path that will be skipped when selecting an element.

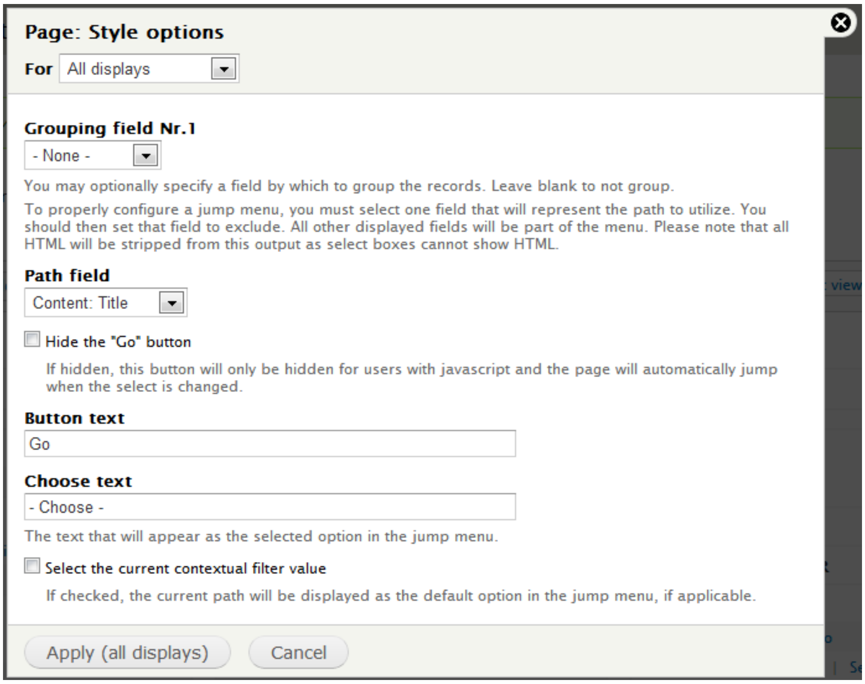


Table format

Creates a table where the columns correspond to fields. The tables allow us to order them automatically, add headings, etc.

F23.17

Table format

Shows the elements in table format.

Latest movies

	Year	Director
Street Kings	2008	David Ayer
The Social Network	2010	David Fincher
The Devil's Advocate	1997	Taylor Hackford
Zombieland	2009	Ruben Fleischer
The Day the Earth Stood Still	2008	Scott Derrickson
Matrix	1999	Andy Wachowski, Larry Wachowski

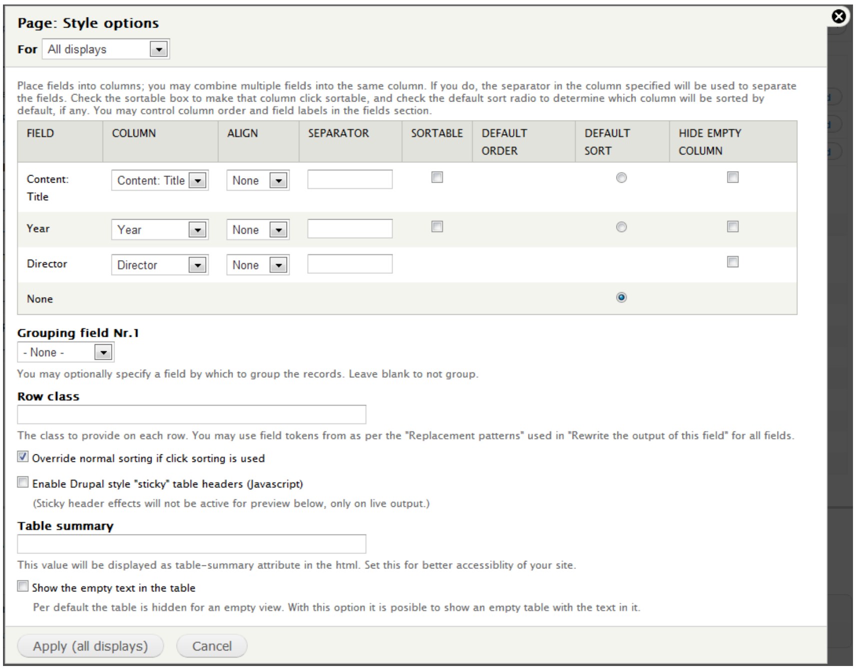
From Settings in Table format we can configure: **F23.18**

- **Column** where the field is placed. This allows us to group several fields in the same column. In these cases we can define the **Separator** used between fields (for example a comma, dash, etc.).
- **Align**. The permitted alignments are left, centre and right.
- **Sortable**. Allows us to define whether the contents of a column are sortable. When a column is sortable, the title becomes a clickable link that allows the user to change the order of its contents (ascending/descending).
- **Default order**. Defines which column sets the order of the table.
- **Override normal sorting if click sorting is used**. If this option is not activated the order defined in the view prevails.
- **Enable Drupal style "sticky" table headers**. A JavaScript effect that keeps the table header fixed in place even if the user scrolls down to view lower rows. A similar effect to the one we have already seen in the

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

site permissions administration table.

- **Default sort method**. Applies the displayed order either in ascending or descending order if we select the default sort for any of the columns.
- **Row class**. Allows us to choose a class name for applying CSS styles.
- **Table summary**. Fills out the table-summary attribute in the HTML code.
- **Show the empty text in the table**. By default the text is not displayed if the view returns no results. Activating this option displays the table with the corresponding text indicating that there are no results.



F23.18
Table format. Settings
We can make individual columns sortable so that their order can be changed via their headers (ascending o descending).

Page settings

If we choose the Page presentation type, the **Page settings** are displayed: **F23.19**

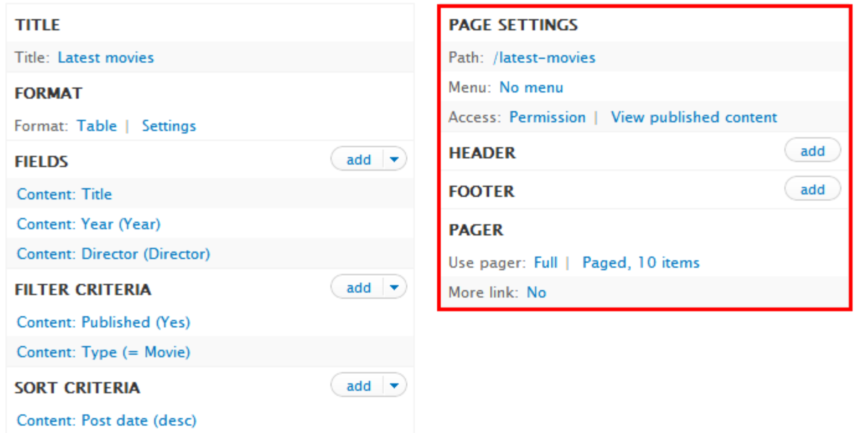
- **Path**. URL alias of the page where the view will be displayed.
- **Menu**. We can associate the page with a menu element.
- **Access**. Defines view access via permissions or roles.
- **Header** and **Footer**. Allows us to add header and footer blocks with different content and configurations. There are two different types:
  - **Text area**. Adds a formatted text. We can use tokens.
  - **View area**. Allows us to insert another view into the header or footer of the main view.

- **Pager**. Allows us to configure how the elements will be displayed on a page when they exceed the maximum defined number of elements permitted. The available pager options are:
  - **Display all items**, no pager.
  - **Display a specified number of items**, no pager.
  - **Paged output**, mini pager.
  - **Paged output**, full pager.

F23.19

Page settings

Settings options for the Page display.



Block settings

If we choose the Block display type, the **Block settings** options appear. The new options available in Page display type are: F23.20

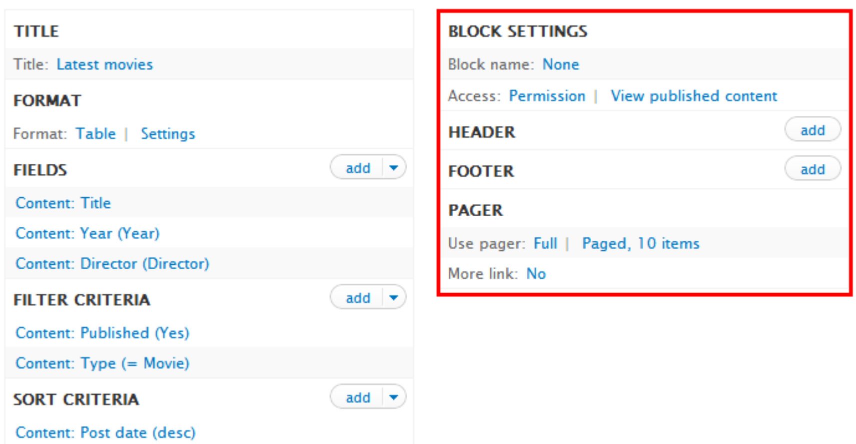
- **Block name**. The name displayed in the block administration area.
- **More link**. A **More** link connects the block to the view page. In general the block displays a smaller set of results than the page. The **more** link thus allows the user to access the rest of the view results.

The **Access**, **Header**, **Footer** and **Pager** options are similar to those on the presentation Page.

F23.20

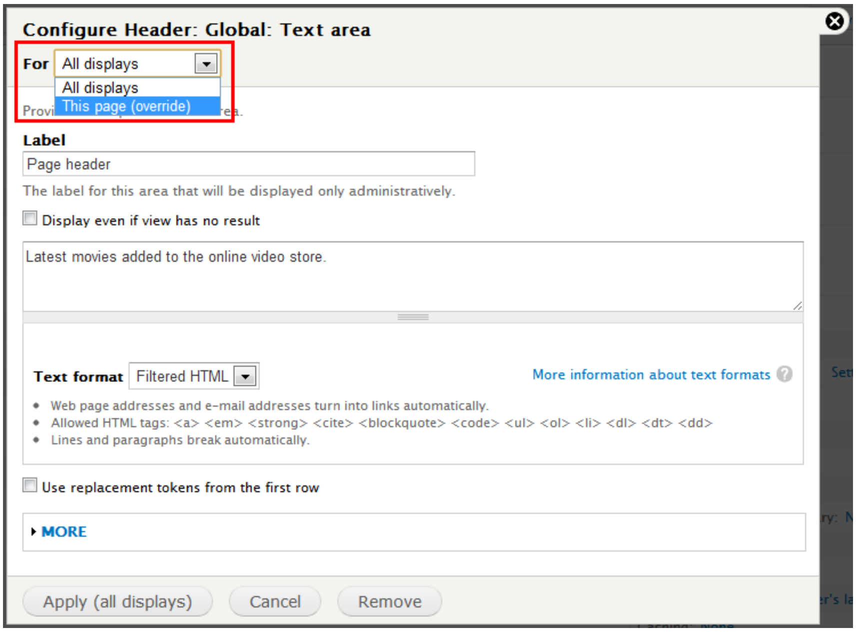
Block options

Settings options for the display of a Block.



Override settings

In general, the common options that we configure in the view affect **All displays**. If we want to apply a different configuration to a single display, we must select the "**override**" option. In this way we can configure a specific value for the selected display.

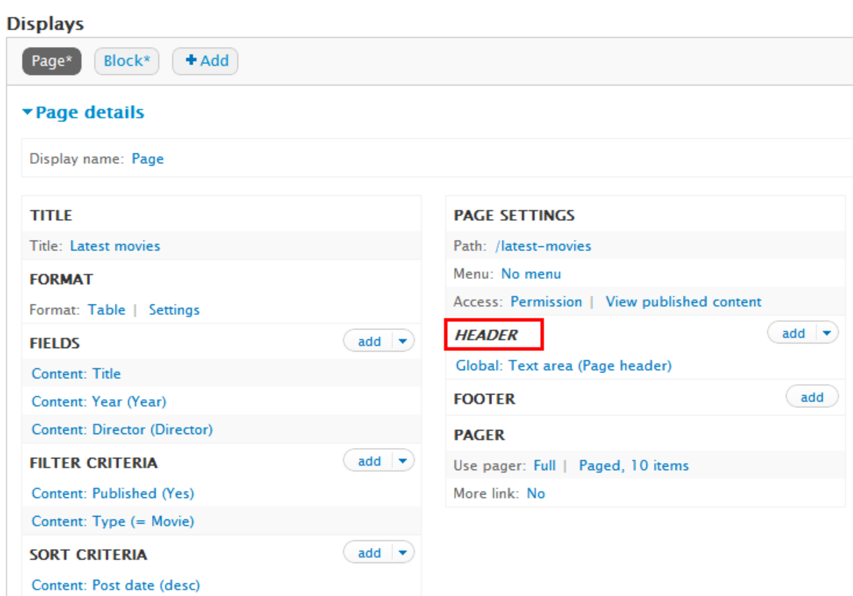


F23.21

Override settings

We can apply specific for each display via the Override option.

The overridden options in the display are shown in cursive (italic). **F23.22**



F23.22

Override settings

Overridden options are displayed in cursive (italic).

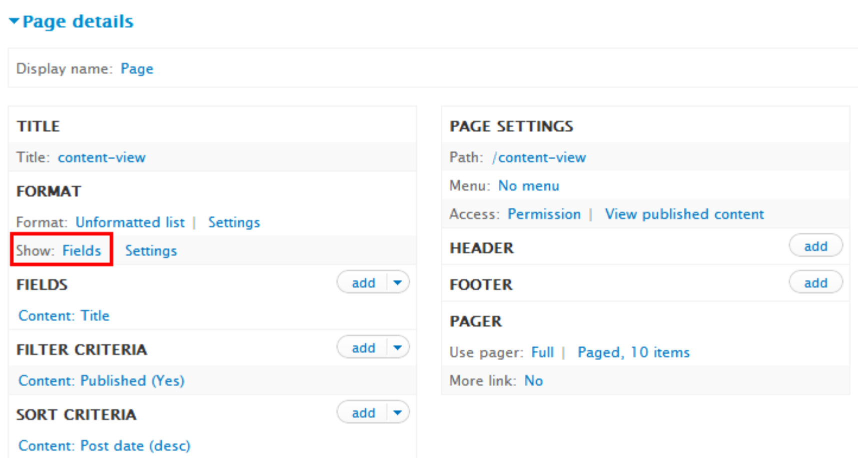
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

23.4 Fields

In this section we will learn how to add **fields** to the views (**Fields** options).

The **Add fields** option is only available if we select **Fields** in the display **Format** options. There is no need to select this option for the **Table** format as this only uses fields. F23.23

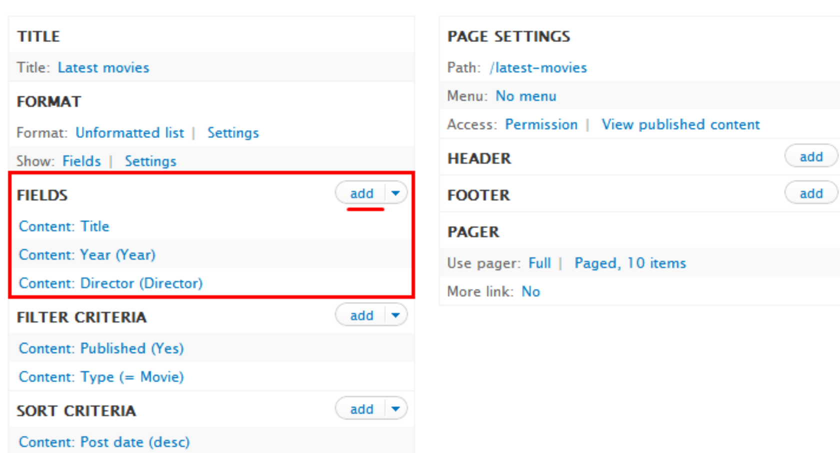
F23.23
Fields
Allows you to add fields to views.



Clicking on the **add** button in the **Fields** group takes us to a list of available fields. This list varies depending on the type of view selected and on the installed modules as well as on the fields created on the site.

For example, if the view is a Content field and we want to display a list containing movie information (content type Movie) we can add generic fields (such as node title, publication date, etc.) and/or specific fields of the content type (country, year, cover, actors, directors, etc.).

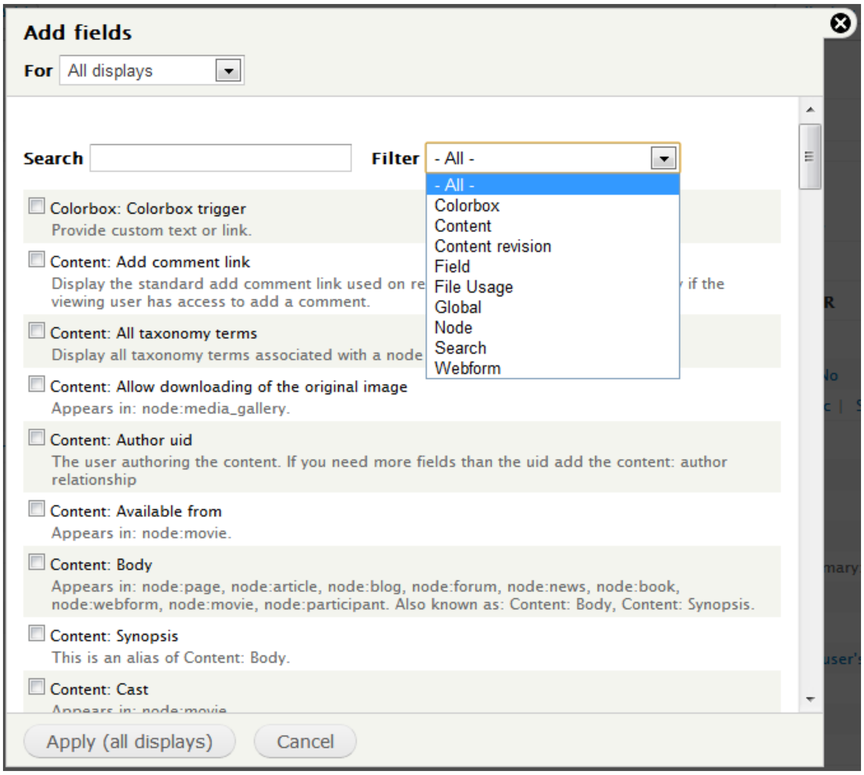
F23.24
Add fields
Add fields by clicking on add.



The **Filter** selector F23.25 allows us to filter fields by the group that they belong to. This is useful as the number of fields available on a site increases. We find among others, the following groups:

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Content**. Fields related to the nodes (route, title, type, body, publisher, summary, sent date, last modified, etc.) and the fields created on the site.
- **Global**. General use fields (text or personalized links, mathematical formula, view result position).
- **Taxonomy**. Taxonomy-related fields (vocabulary name, terms, etc.).
- **User**. User-related fields (user name, roles).



F23.25
Add fields
You can filter by field type. Once you find the correct field, select it and click on Apply.

After choosing the fields that we want to add to the view we click Apply (or Add and configure fields).

Upon adding a field, specific configuration options appear. These vary depending on the type of field selected.

For example, upon adding the **Content: Title** (node title) field we can configure the following options: F23.26

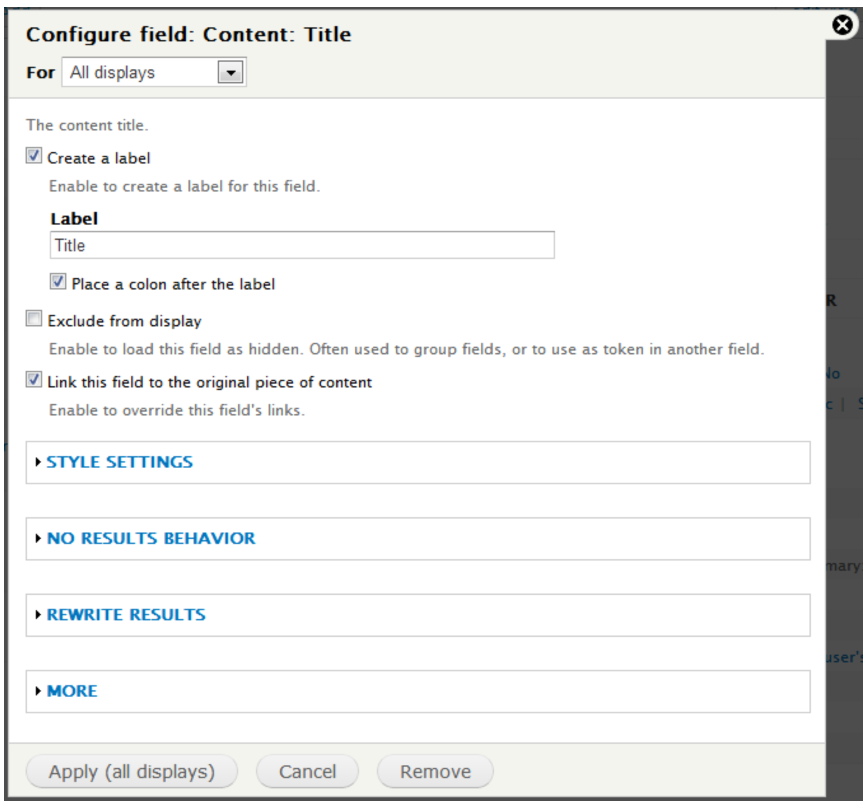
- **Label**. Adds a label, normally it identifies the field content. For example, title, synopsis, subtitle, etc.
- **Exclude from display**. The field is available in the view but not displayed. Hidden fields can be used in groups along with other fields or as tokens for other fields (see the rewrite option for this field).
- **Link this field to the original piece of content**. Links the field with the node that it refers to.
- **Style settings**. Allows us to make changes to the HTML tags and CSS class used to display the field.
- **No results behaviour**. Indicates how the field behaves if it returns no results or is empty.
- **Rewrite results**. Some options included in this field are (new options appear when the main options are activated):
    - **Rewrite the output of this field**. Allows us to rewrite the

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

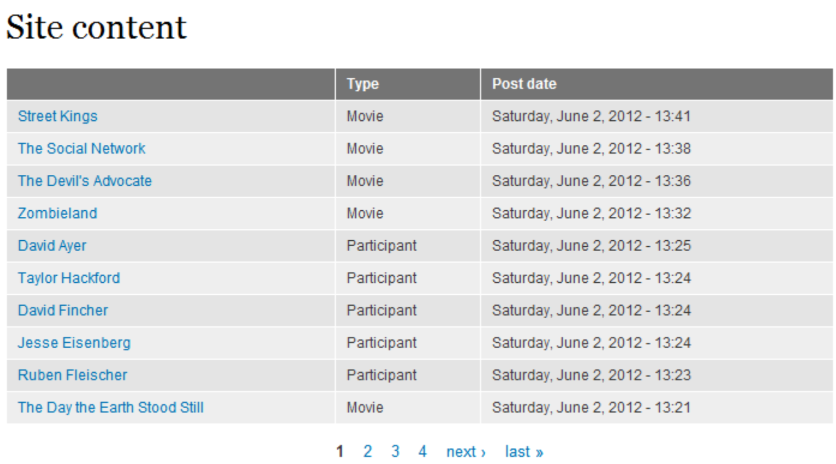
field value using HTML and replacement values. Among the available values in the replacement patterns are fields added to the view **before the actual field**, even hidden fields.

- **Output this field as a link**. The field value is outputted as a link. We can choose a link and the link properties. The URL can be generated dynamically via the replacement patterns.
- **Trim this field to a maximum length**. Allow us to trim the field output by configuring options such as **Trim only on a word boundary** or **Add an ellipsis**.
- **Strip HTML tags**. Removes the HTML tags from the field. We can choose which tags not to strip.
- **Remove whitespace**. Removes any white spaces before and after the field value.
- **Convert newlines to HTML
 tags**.
- **More**. Other options, such as changing the field name (Administrative title) in the administration pages.

F23.26
Field settings
Example of a field setting. The options may vary depending on the type of field.



In the Figure F23.27 we see a simple **Content** view formatted as a table. It shows the fields **Content: Title** (linked to the node), **Content: Type** and **Content: Post date**. No extra filters or order criteria have been added so all published content is displayed in all content types across the site.



Site content

	Type	Post date
Street Kings	Movie	Saturday, June 2, 2012 - 13:41
The Social Network	Movie	Saturday, June 2, 2012 - 13:38
The Devil's Advocate	Movie	Saturday, June 2, 2012 - 13:36
Zombieland	Movie	Saturday, June 2, 2012 - 13:32
David Ayer	Participant	Saturday, June 2, 2012 - 13:25
Taylor Hackford	Participant	Saturday, June 2, 2012 - 13:24
David Fincher	Participant	Saturday, June 2, 2012 - 13:24
Jesse Eisenberg	Participant	Saturday, June 2, 2012 - 13:24
Ruben Fleischer	Participant	Saturday, June 2, 2012 - 13:23
The Day the Earth Stood Still	Movie	Saturday, June 2, 2012 - 13:21

1 2 3 4 next › last »

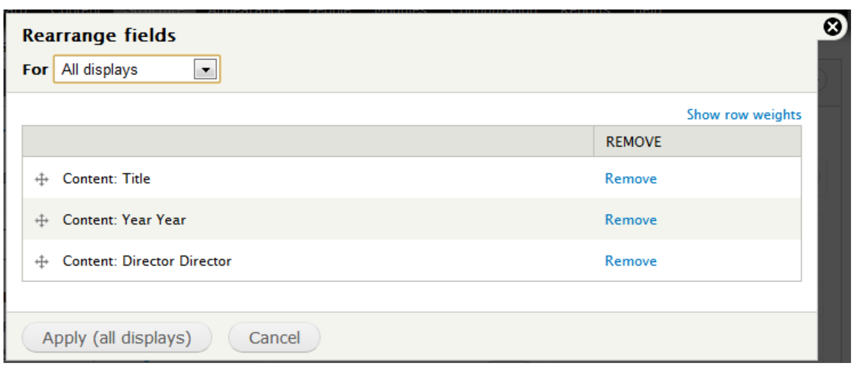
F23.27

Example view

Example of a view in Table format.

Rearrange fields

Once we add the fields, we can reorder or even delete fields from the **rearrange** option (available in the same drop down menu as **Add**). F23.28



F23.28

Rearrange fields

From the rearrange option you can change the order of fields or delete them.

As we have already stated, the correct distribution of fields may be needed both for presentation purposes and in cases when fields depend on others, for example, when **tokens** are used. A field can use values related to fields as tokens only if they are created before it. It will not "see" later fields.

Override fields

Just like we override the general settings options, we can override fields and their options for a view display. In this way, displays of the same view can have the same fields configured in different ways, or can have different fields.

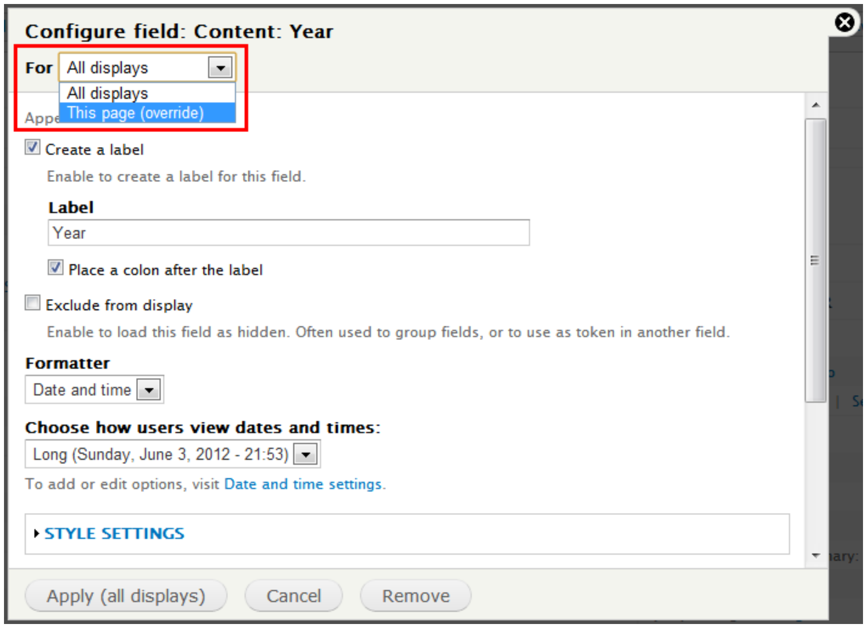
To change the configuration of a field for a particular display simple select the display using the **override** option at the top of the field settings window. F23.29

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F23.29

Override field configuration

You can also override fields, which allows you to have different settings for the same field in view displays.

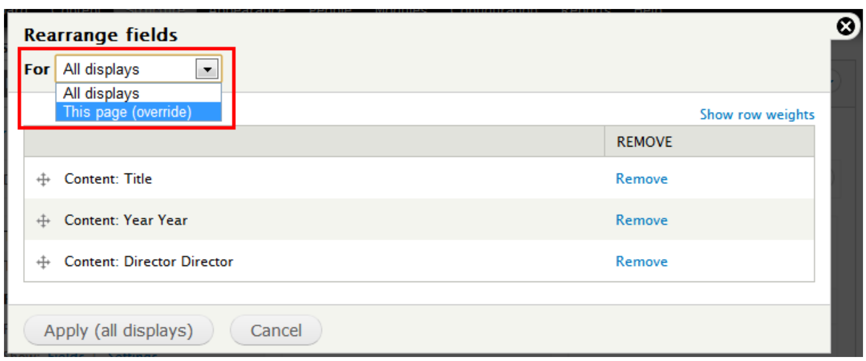


If we want to change the field order or delete a field for a particular display, we go to the rearrange window and select the display that we want to override (from the header). **F23.30**

F23.30

Rearrange fields

You can also override fields from rearrange fields, which allows you to have a different field order in each display and even to delete fields.



Sort criteria

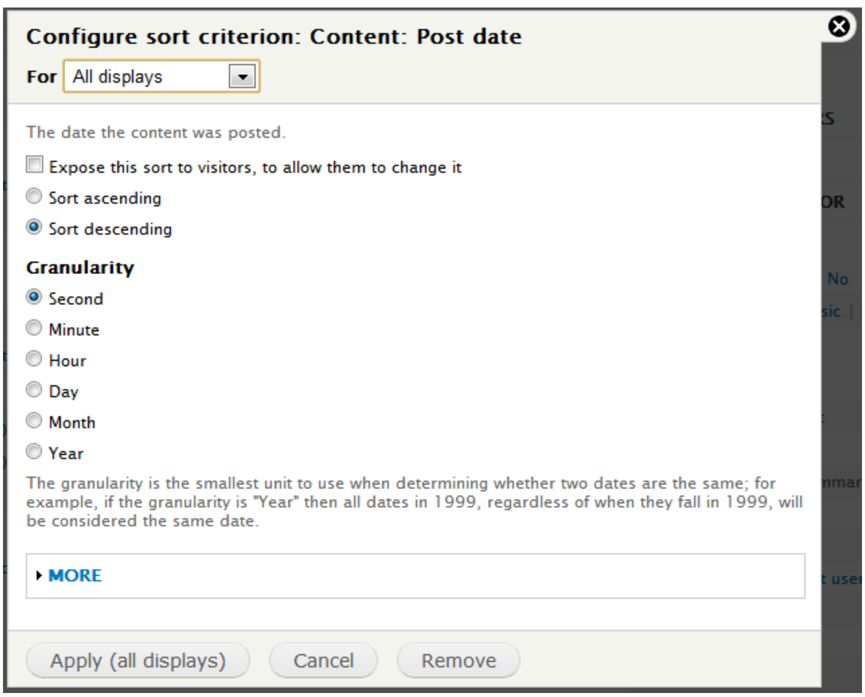
23.5

We can apply sort criteria using any available field even if they have not been added to the view. For example, we can sort by the date of publication of nodes (**Content: Post Date**), even if we have not added the field to the view.

In **Sort criteria** we add new fields by clicking on **add**. By default some sort criteria that have already been created may appear, depending on the options selected when the view was created.

As a general rule, adding a field brings up the following choices: F23.31

- **Sorting method**. Can be **Ascending** or **Descending**. For a date, **Ascending** displays the oldest elements first. For a text field, **Ascending** displays the elements in alphabetical order.
- **Granularity**. In date fields we can even choose which elements of the date to use (year, month, day, hour, minute or second), and which to leave out.



F23.31

Configure sort criteria

Each field can have different sort criterion depending on the values that it stores.

A display can have several sort criteria. The sort criteria are applied depending on the established order (which can be modified via **rearrange**). The first criterion is applied and the second only comes into play if several results are the same.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

For example, in the **Figure** F23.32 we see the results of applying the following sort criteria:

- Content: Year (desc)
- Content: Title (asc)

The movies are first sorted by the Year field in descending order (the most recent movies first). When two movies have the same year, they are sorted via the Title field, in ascending (alphabetical) order.

F23.32

Sort example

A display of content sorted by the Year field in descending order.

Latest movies

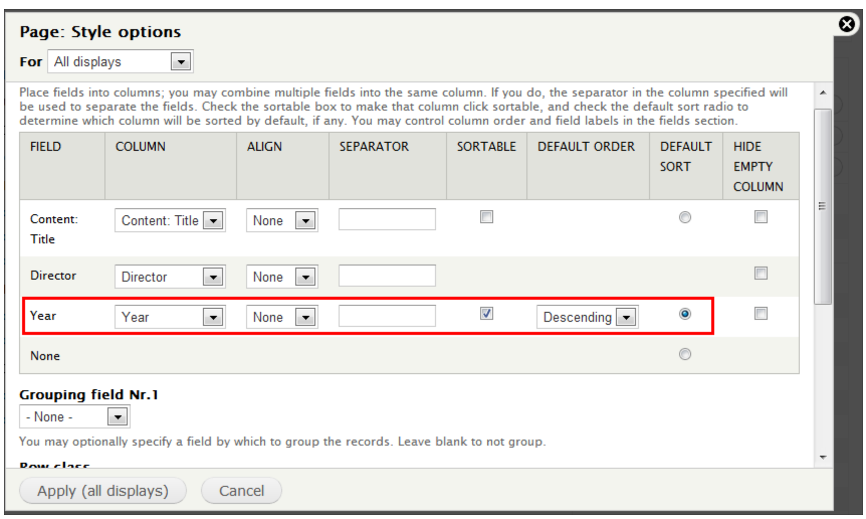
	Director	Year
The Social Network	David Fincher	2010
Zombieland	Ruben Fleischer	2009
Street Kings	David Ayer	2008
The Day the Earth Stood Still	Scott Derrickson	2008
Matrix	Andy Wachowski, Larry Wachowski	1999
The Devil's Advocate	Taylor Hackford	1997

When using the **Table** format we can configure the column headers as sortable column clicks. We do this from the **Settings** of the **Table**, selecting which columns are **Sortable**.

F23.33

Sortable columns

You can configure the table columns as sortable.



Sortable columns are indicated by an arrow, either pointing upwards or downwards depending on the selected setting (ascending or descending respectively). Clicking on the title column , we can change the sort method. F23.34

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Latest movies

	Director	Year
The Social Network	David Fincher	2010
Zombieland	Ruben Fleischer	2009
The Day the Earth Stood Still	Scott Derrickson	2008
Street Kings	David Ayer	2008
Matrix	Andy Wachowski, Larry Wachowski	1999
The Devil's Advocate	Taylor Hackford	1997

F23.34

Sortable column

The sortable column header displays a visible filter that allows its contents to be sorted in either ascending or descending order.

Sort criterion exposed to visitors

The sort criterion can be exposed to visitors, allowing them to decide how to sort the results displayed in a view.

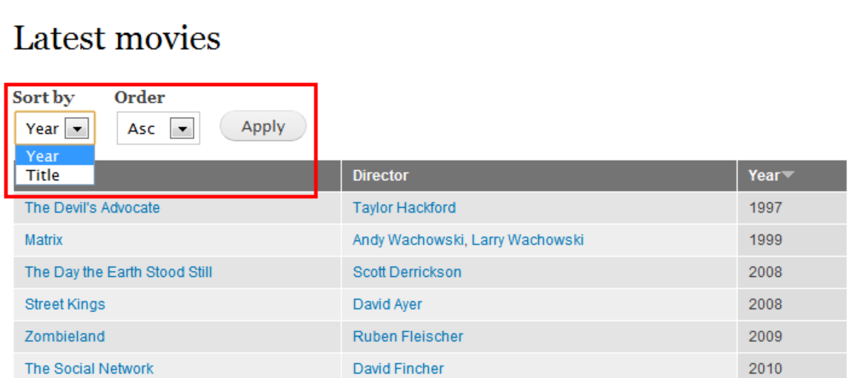
From Configure site criterion choose the **Expose this sort to visitors, to allow them to change it** option. F23.35

F23.35

Exposed sort

You can expose the sort to visitors, allowing them to select the sort method for the results displayed in the view.

In **Figure** F23.36 we see the sort criterion exposed to visitors. They can sort by the **Add** field and by the **Year** field, choosing whether the sort is in ascending or descending order.



	Director	Year
The Devil's Advocate	Taylor Hackford	1997
Matrix	Andy Wachowski, Larry Wachowski	1999
The Day the Earth Stood Still	Scott Derrickson	2008
Street Kings	David Ayer	2008
Zombieland	Ruben Fleischer	2009
The Social Network	David Fincher	2010

F23.36

Exposed sort examples

In this view the Year and Title sort criteria have been exposed. As well as giving visitors the option to sort via different fields, they can also choose whether to view the results in ascending or descending order.

23.6 Filter criteria

Filters allow us to define the conditions for displaying specific elements in a view. For example, we can add a filter that only displays Movie nodes, or only movies released in the year 2012 (or both).

Filters can be hidden, and therefore always applied to the results, or exposed to visitors, allowing them to decide which filters to apply. This second option turns the view into a simple search engine. For example, we can add an exposed year filter (2008, 2009, 2010, 2011, 2012) that allows visitors to choose the year and only view nodes published in that year.

To add a filter to the view we click on the **Add** button of the **Filter criteria** section, and then select the field to be used to execute the filter.

For example, if we add the **Content: Type** field, the following configuration options appear: F23.37

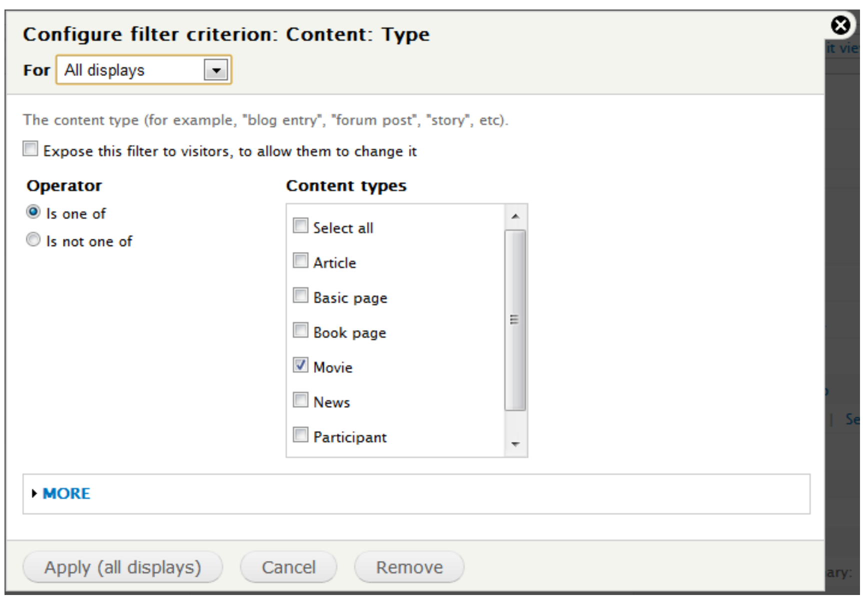
- **Operator**. Allows us to choose the type of filter to apply (is one of, is not one of).
- **Content types**. Defines the content types that are affected by the filter.

If selected:

- **Operator**: Is one of.
- **Content types**: Movie.

We are indicating that only the **Movie content** type will be displayed.

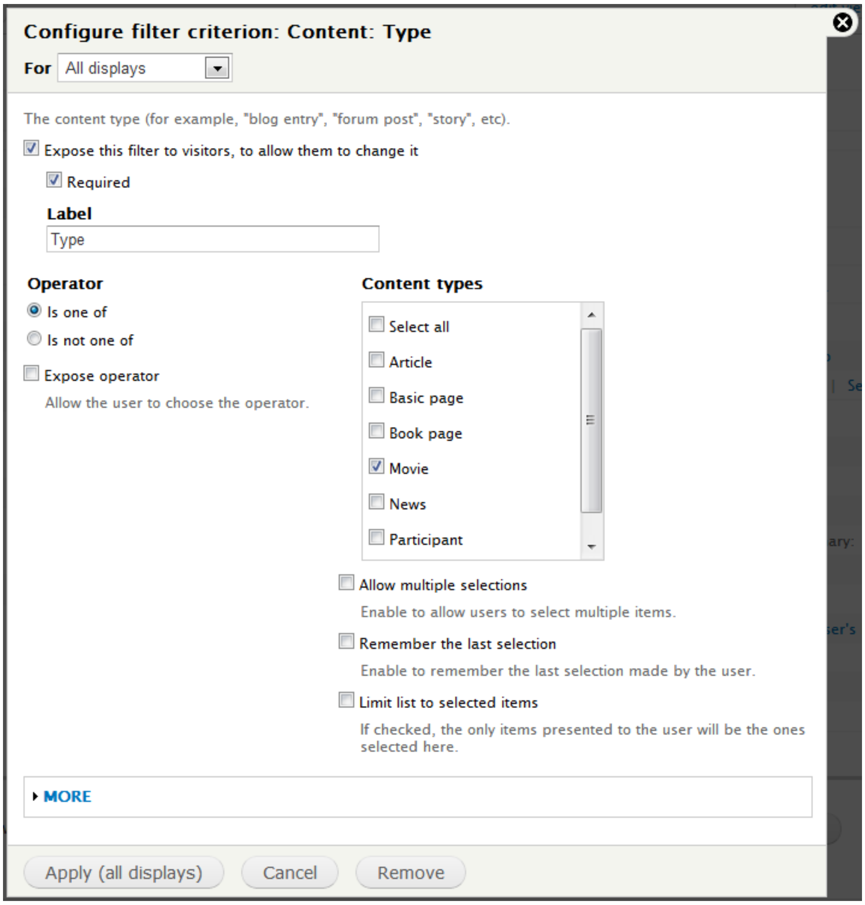
F23.37
Applying filters
You can give each filter a specific configuration by



The **Expose** option (**Expose this filter to visitors, to allow them to change it**), makes the filter visible in the view. This means that visitors can choose between the values that we have activated.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

On activating the **Expose** F23.38 option new setting options appear:



F23.38

Exposing filters

The setting options for an exposed filter can also vary depending on the properties of the field it is applied to.

- **Required**. If this option is not activated, an “any” option is added so that the visitor can choose a filter value, or not.
- **Label**. Text that is displayed in the view to identify the filter. In general this must be modified to make its content suitable for the user.
- **Expose operator**. The user can also select which operator to apply.
- **Allow multiple selections**. If activated the user can choose several values (displayed as a multiple selection list). If not activated the filter displays as a single selection list.
- **Remember the last selection**. If activated this applies the last selected filter when the user returns to the view.
- **Limit list to selected items**. Displays only the selected items to the user. In this example only the selected nodes are displayed in the selection list.

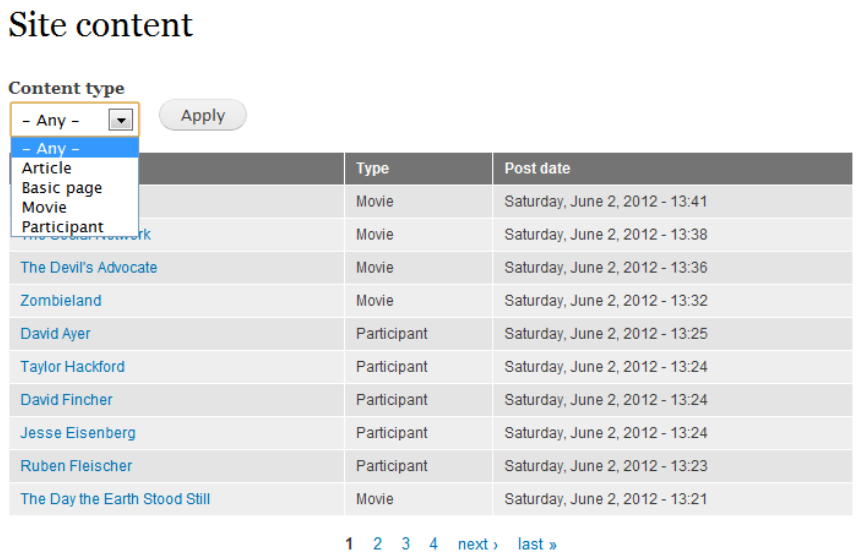
In the Figure F23.39 we see a table of contents generated by a **Content type** exposed filter (we have modified the label). The user can select a content type to display the corresponding content.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F23.39

View with exposed filter

An exposed filter is nothing more than a form element that allows the visitor to make selections. The displayed elements depend on the filters that are applied.



Y (AND) and O (OR) operators

By default, filter criteria are applied using AND logic, meaning that only results that match all criteria are displayed:

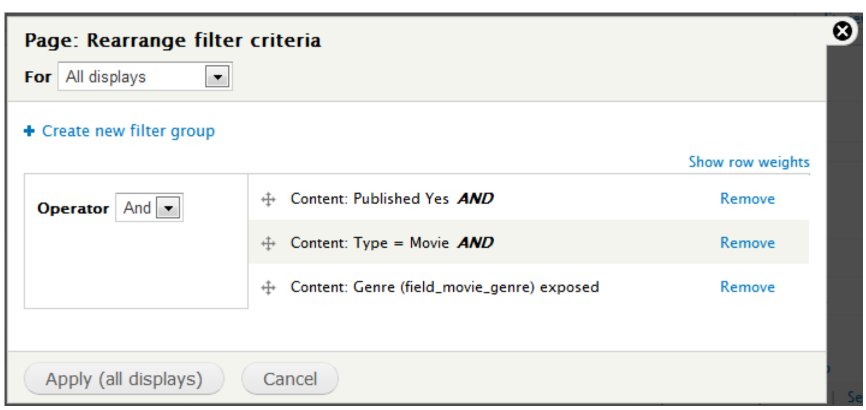
criteria1 **AND** criteria2 **AND** criteria3

Clicking on the options list displays the **and/or** option alongside the default **And** option. From this window we can specify the Operator to be used by the filters. **F23.40**

F23.40

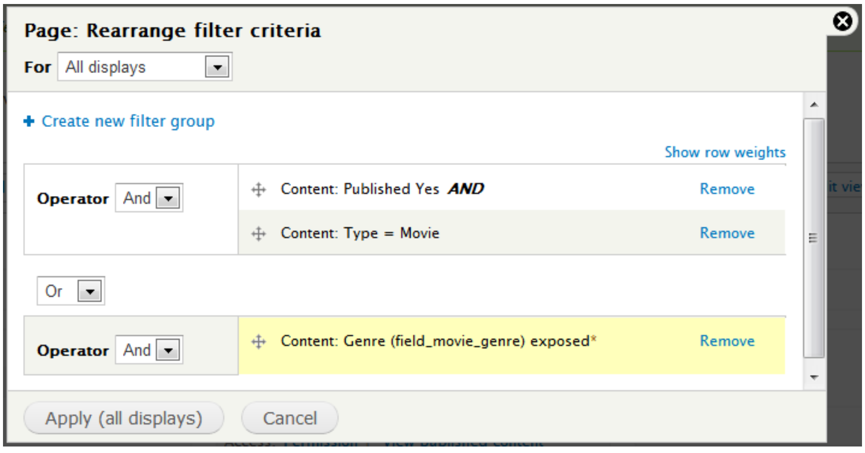
Filter operators

Filters can use a AND operator (the results must match all criteria) or O (the results must match at least one criteria).



We can also separate filters into filter groups, allowing us to specify a different operator for each group and create sets of criteria. **F23.41**

For example: (criteria1 **AND** criteria2) **OR** (criteria3 **AND** criteria4)



F23.41

Filter groups

Via filter groups you can combine different AND and OR operators.

Practical case 23.1 Create the Latest movies view

In this practical example we create, step-by-step, a view that displays on a page a list of movies (content type **Movie**). The view has the following characteristics:

- **View name**: Latest movies
- **Machine name**: latest_movies
- **URL alias:** videostore/latest-movies
- **Page title**: Latest movies
- **View type**: Content
- **Display**: Page
- **Format**: Table (therefore, we will work with fields)

- **Page settings**:
  - Five elements displayed per page. Use a pager.
  - The **header** displays the text: "On this page you can view the latest movies added to the online video store".

- **Fields**:
  - Movie title (linked to the node)
  - Country
  - Genre
  - Year
  - Director/s
  - Cast

- **Order criteria**:
  - The movies are ordered by node publication date, in descending order (the latest movies first).
  - The table is sortable via the Title and Year columns.

- **Filter criteria**:
  - Only Movie content types are displayed.
  - An exposed filter allows sorting via Genre.
  - An exposed filter allows sorting via movie Title.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Step 1. View creation and initial configuration

The first step is to create the view. To do this we go to: F23.42

URL Add new view
/admin/structure/views/add

Administration ⇨ Structure ⇨ Views [Add new view]

First we indicate the name of the view (**Latest movies**) and modify the system name (**latest_movies**). We select display **Movie** type **Content** ordered by **Newest first**.

We create a **Page** titled **Latest movies** and the URL alias **videostore/latest-movies**.

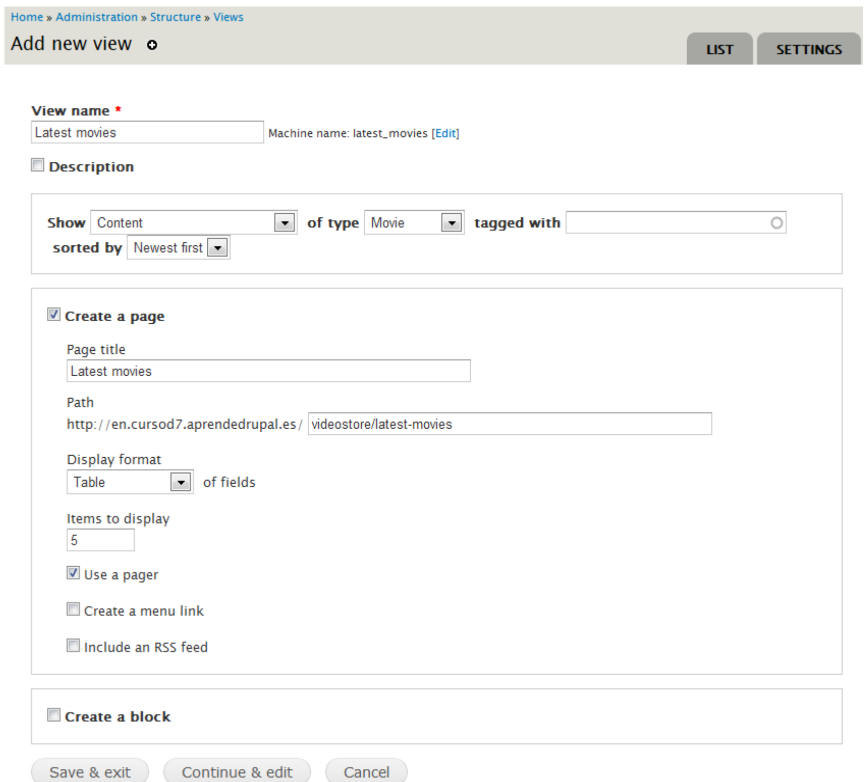
The display Format is a **Table** that displays **5 items**.

We click on **Continue & edit** to continue building the view. Bear in mind that the view has not yet been saved.

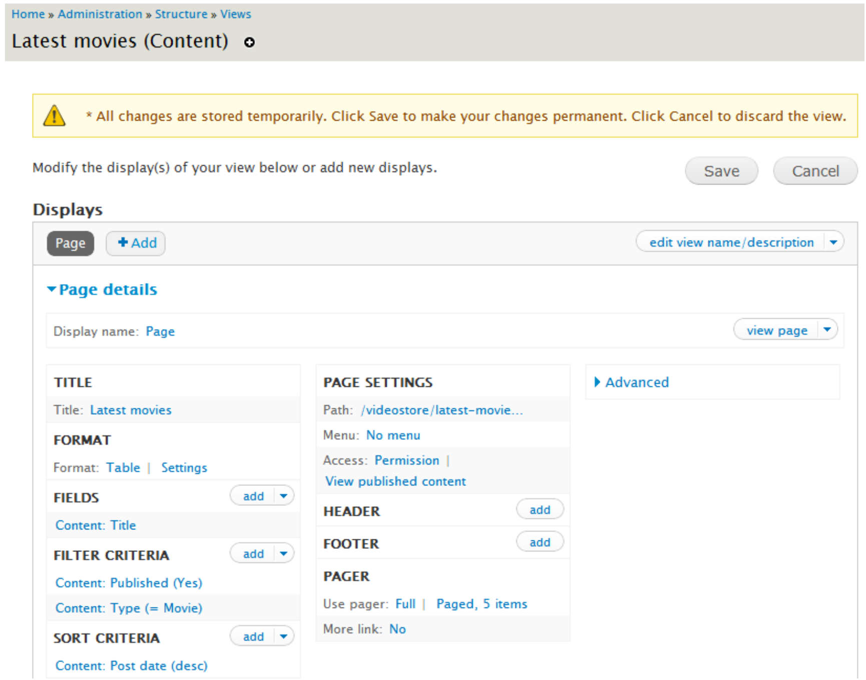
F23.42

Practical example 23.1
Creating a view

When creating a view you can also create the page display.



In the view edit we can see that many settings options have already been filled in with the information that we provided upon creating the view. F23.43



F23.43

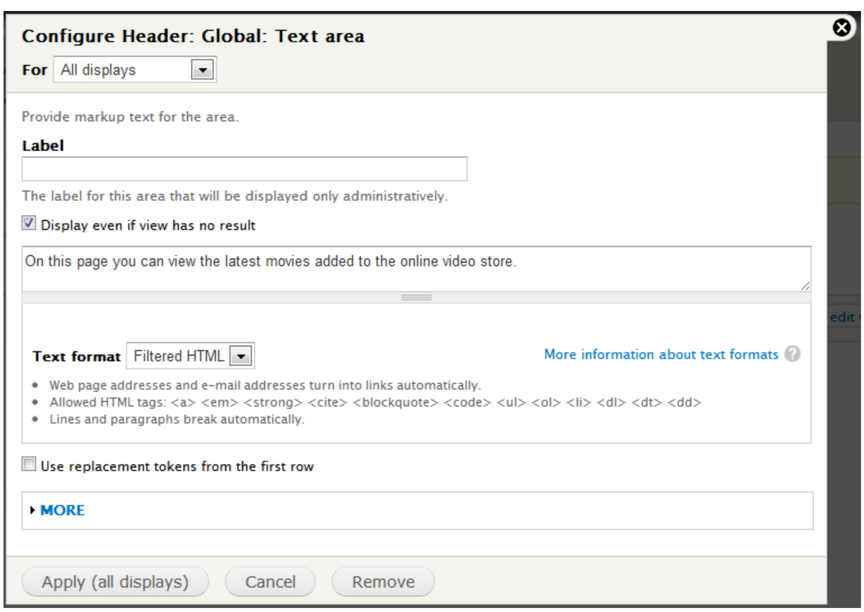
Practical case 23.1
Page editing

The view is not created until you save it at least once.

The only remaining general configuration step is to write the header. We click on add (header), and add a header of the type:

- Global: Text area

Once added we can write the header text: "On this page you can view the latest movies added to the online video store". The text is to be displayed even when the view contains no results so we activate this option. The header does not have a label. F23.44



F23.44

Practical case 23.1
Header settings

Add a Text area header.

Before continuing with the next step, save the view by clicking on Save.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

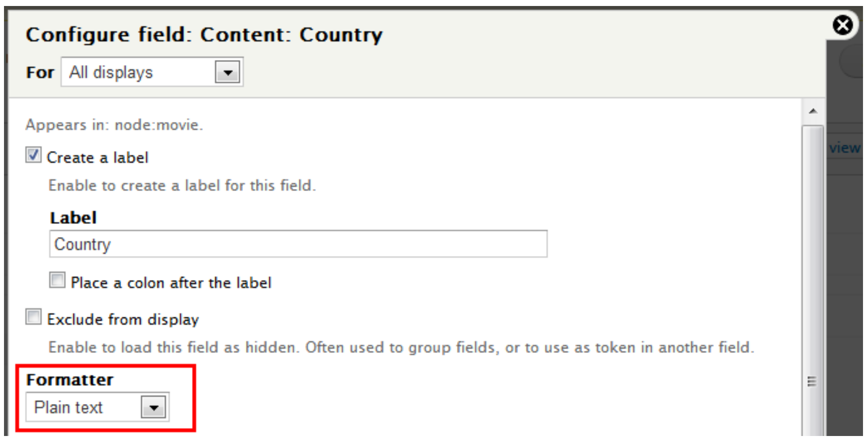
Step 2. Adding fields

We add the following fields to the view:

- **Movie title** (linked to node). This field corresponds to the node title.
- **Country**.
- **Genre**.
- **Year**.
- **Director/s**.
- **Cast**.

The **Movie title** field is already added by default to the view (**Content: Title**). Accessing its settings we create the label Title (**Create a label**). We must also check that the **"Link this field to the original piece of content"** option is activated.

To add the **Country** field we click on **add** (fields) and locate the **Content: Country** field. In the field configuration we change the **Formatter**, selecting **Plain text**. We do not opt to display the colon after the label. F23.45

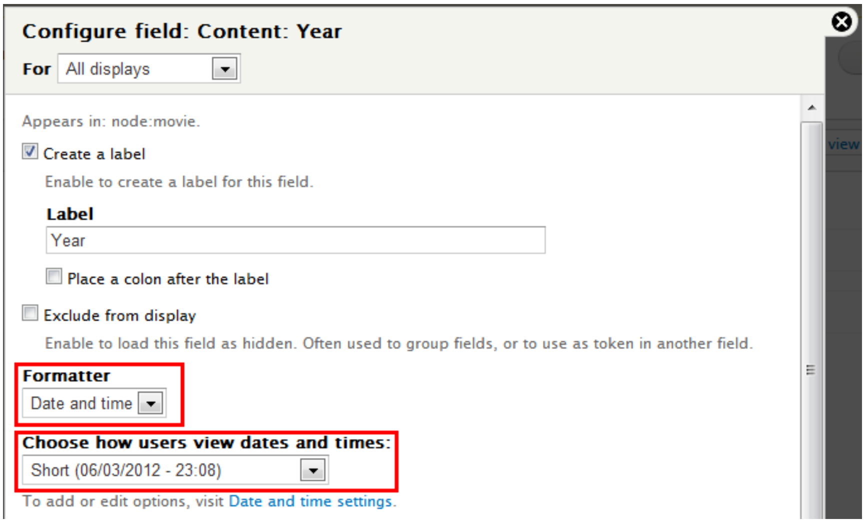


F23.45

Practical case 23.1
Formatter
Selecting the format for the field display.

The next field is **Genre** (**Content: Genre**). Again we select **Plain text** in the **Formatter**. We do not opt to display the colon after the label.

In the **Year** (**Content: Year**) field we find specific options for date fields. In the Formatter we can choose between "Date and time" or "Time ago". In this case we select Date and time. In the date format we select Short, even though only the year will be displayed in this field whichever date format we select (short, medium, long). F23.46

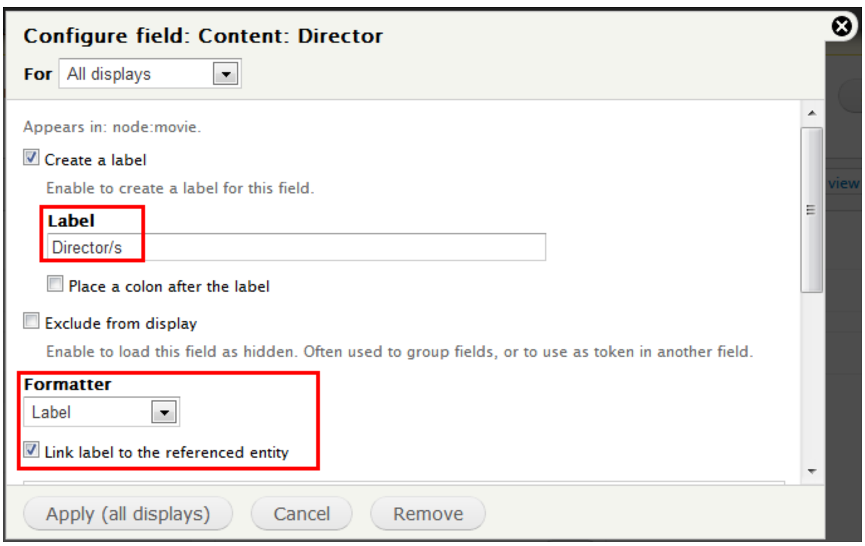


F23.46

Practical case 23.1
Date format

In date fields you can select the date format: short, medium or long.

For the **Director/s** (**Content: Director**) **F23.47** field we modify the (**Director/s**) label and, in **Formatter,** select **Label**. We link to the referenced node by activating "**Link label to the referenced entity**".



F23.47

Practical case 23.1
Referenced entity field

In the fields that refer to another entity you can link the label to the referenced entity.

We set the same configuration options for the **Cast** (**Content: Cast**) field.

At any time we can see how the field looks from the preview or directly from its page (**videostore/latest-movies**).

Step 3. Adding the sort criteria

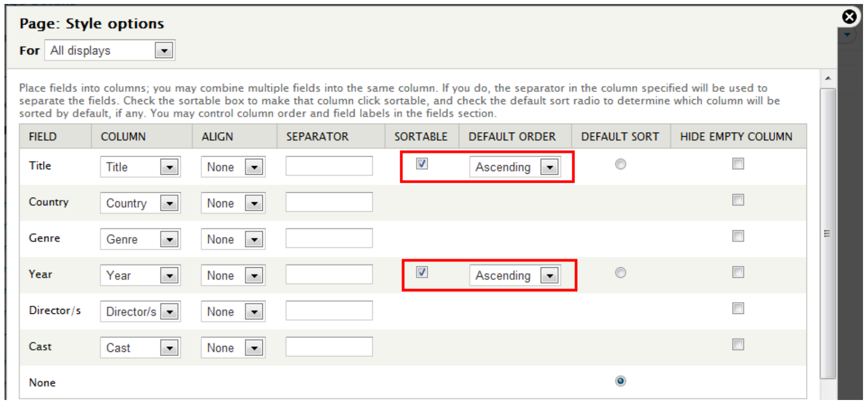
Because we set the view to display the latest Published nodes first when we set up the view, the **Sort criteria** are already correctly configured: **Content: Post date (desc)**. We do not need to make any changes to these criteria, although we can access their configuration options to check that everything is set correctly.

Next we configure the **Title** and **Year** columns to be sortable. We do this from the **Settings** of the **Table** format. **F23.48**

F23.48

Practical case 23.1
Sortable columns

You can configure some of the table columns to be sortable. You can also set the default order.



Now the titles of the **Title** and **Year** columns are links that make the columns click sortable. F23.49

F23.49

Practical case 23.1
View display

The current status of the view.

Latest movies

On this page you can view the latest movies added to the online video store.

Title	Country	Genre	Year	Director/s	Cast
Matrix	United States	Sci-Fi	1999	Andy Wachowski, Larry Wachowski	Keanu Reeves
Street Kings	United States	Action, Drama	2008	David Ayer	Keanu Reeves
The Day the Earth Stood Still	United States	Sci-Fi	2008	Scott Derrickson	Keanu Reeves
The Devil's Advocate	United States	Drama, Thriller, Horror	1997	Taylor Hackford	Keanu Reeves
The Social Network	United States	Drama	2010	David Fincher	Jesse Eisenberg

1 2 next › last »

Step 4. Adding filter criteria

To complete the view we must add filters that give us the following:

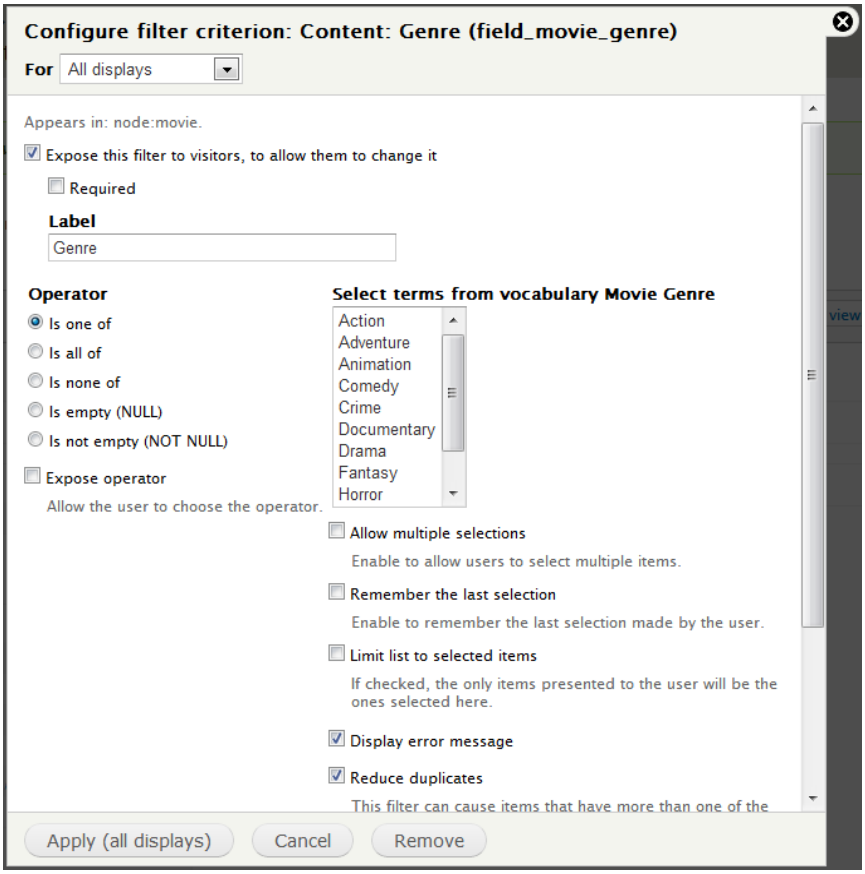
- Only Movie type content is displayed.
- An exposed filter allows the user to select the genre.
- An exposed filter searches by movie title.

The first filter has already been applied, as it was set when we created the view. It displays identified as: **Content: Type (= Movie)**. A further filter, **Content: Published (Yes)**, has also been added by default, meaning that only published content is displayed.

Now we create the **Genre** filter. From add (in **Filter criteria**) we add the **Genre** (**Content: Genre**) field. Remember that Genre is a **Term reference** field that displays a list of taxonomy terms associated with a vocabulary.

We have to set the **Selection type**, Dropdown or Autocomplete. Choose the first, **Dropdown**. Next we configure the filter, exposing it to visitors and selecting **Operator**: **Is one of**. We do not need to select any terms as the user has access to them all (We do not activate the **Limit list to selected items** option). We do select **Reduce duplicates**. F23.50

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

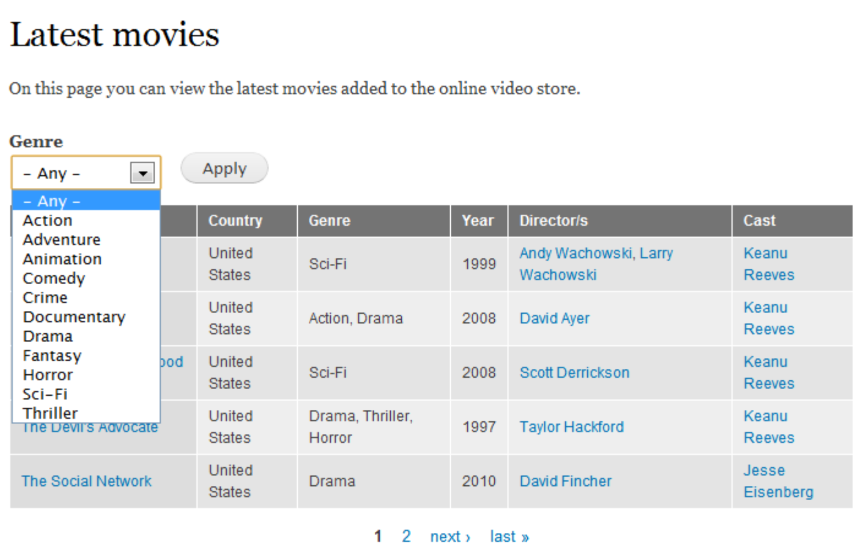


F23.50

Practical case 23.1
Filter criterion

Criterion filter options when using the Genre field.

The exposed **Genre** filter is seen in action in **Figure**. F23.51



F23.51

Practical case 23.1
View with exposed filter

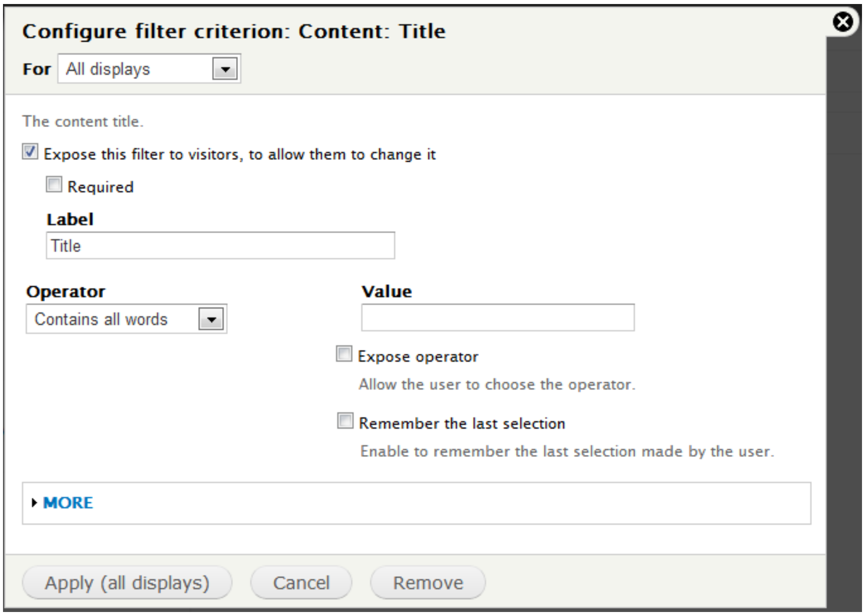
The exposed filter allows you to select the movie genre

Lastly we add a filter that allows Title (**Content: Title**) field text searches. Once the filter is added we expose it and select the "**Contains all word**" **operator**. F23.52

F23.52

Practical case 23.1
Filter criterion

You configure a filter that searches Title field texts.

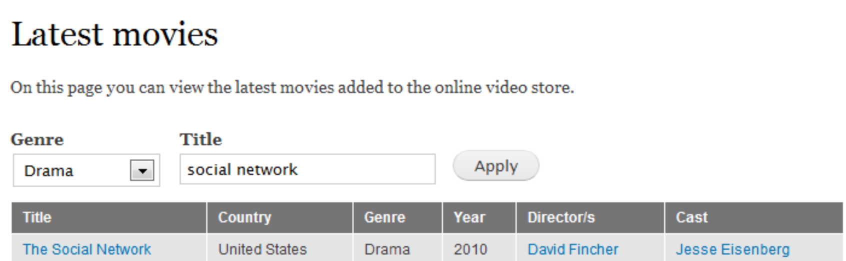


The Title filter allows us to write any word (complete) that is contained in the title. The filters work together, so that only results that comply with the unexposed filters and the values inputted by the user in the exposed filters are displayed. **F23.53**

F23.53

Practical case 23.1
Final result

The final view contains the Genre and Title filters.



Contextual filters

23.7

With **contextual filters** we can give instructions to the views via the URL.

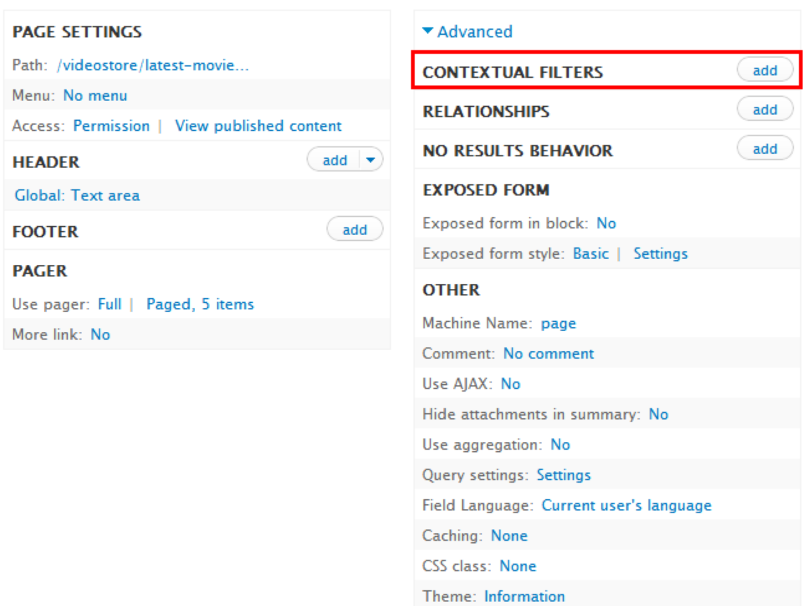
For example, given the following view, which displays all the video store's movies:

- videostore/movies

we can add a genre value to the URL, meaning that only movies of that genre are displayed:

- videostore/movies/**drama**
- videostore/movies/**horror**

To add a contextual filter we must go to the **Advanced** tab and click on **add**, selecting the field that is to act as a filter. F23.54



F23.54

Contextual filters

Contextual filters appear in the Advanced tab. Using the add button you can add contextual filters.

Contextual filter settings are divided into two groups of options: F23.55

- **When the filter value is NOT in the URL**. Determines how the view acts when the expected URL value is not found. Some options are **Display all results for the specified field**, **Show "Page not found"** or **Provide default value**.

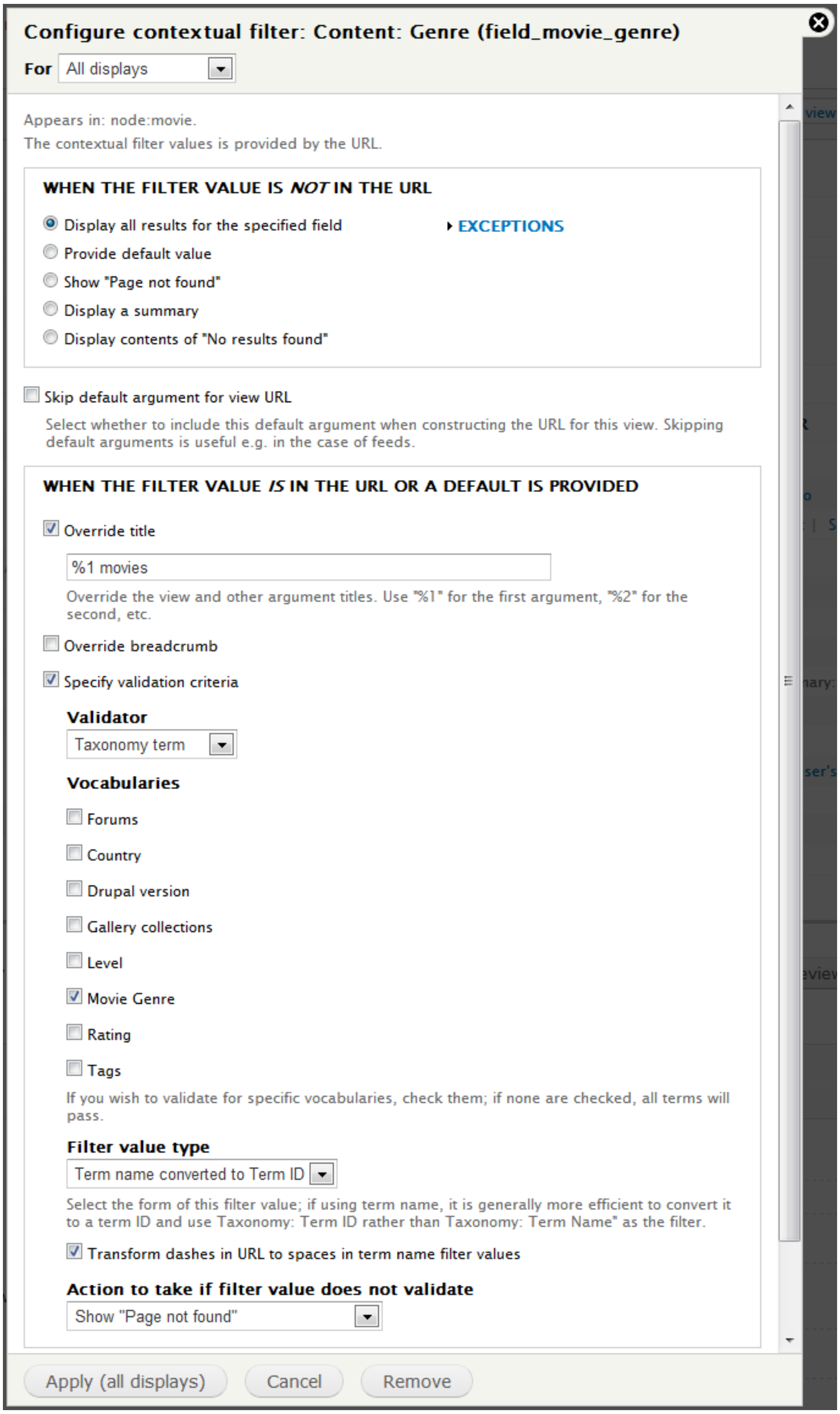
- **When the filter value IS in the URL or a default is provided**. When the filter has a value, either from the URL or via a default value, we can:

  - **Override title**. Allows us to personalize the view title (For example, **Action** Movies, **Horror** Movies).
  - **Specify validation criteria**. Allows us to make sure that the filter value is valid.

F23.55

Contextual filters. Configuration

Contextual filter configuration depends on the field type. In general you can set how the filter acts if the URL value is not found (display all results, default value, show "page not found") and when it is found (override the title and specify validation criteria).



When we add contextual filters via URL we must edit the view path and add the % symbol to every position that corresponds with a filter. For example:

- videostore/movies/%

Where the % corresponds with a general value:

- videostore/movies/drama

In cases where several contextual filters via URL exist, these must be in the same order as the % symbols. For example:

- videostore/movies/%/%

Where the first % refers to the genre and the second to the country:

- videostore/movies/drama/united-states

In the **practical example 23.2** we see how to create a list of movies by genre, depending on the genre value indicated in the URL (For example, **videostore/movies/drama**). F23.56

Drama movies

Title	Country	Genre	Year	Director/s	Cast
Street Kings	United States	Action, Drama	2008	David Ayer	Keanu Reeves
The Social Network	United States	Drama	2010	David Fincher	Jesse Eisenberg
The Devil's Advocate	United States	Drama, Thriller, Horror	1997	Taylor Hackford	Keanu Reeves

F23.56

View with contextual filter

Results of the view after loading the URL: videostore/movies/drama

Contextual filters with predefined values

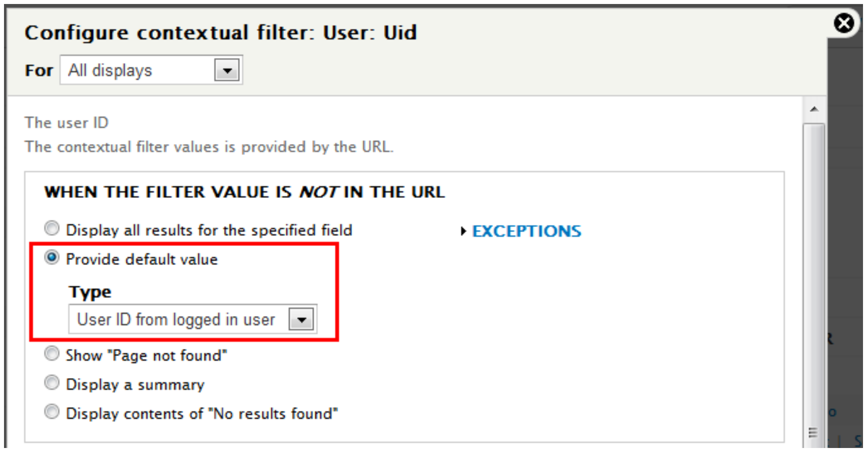
Contextual filter predefined values can be used for different purposes. Some examples of their uses are:

- To provide a default value, generally fixed, when no value is provided by the URL.
- To obtain a filter based on the user who loads the page.
- To obtain a filter based on the author of the node who loads the page.
- To obtain a filter based on the node that is being used.

Suppose that we want to create a block containing information about a user profile that shows each user their own information.

To achieve this we can create a contextual filter by selecting the **User: Uid** field. The identifier value is not passed on via the URL. Instead, we choose **Provide a default value: User ID from logged in user**. F23.57

In this simple way we get the values returned by the view (for example name, surnames and user image) to match the identified user on the site. In the **practical example 23.3** we see this example of the use of predetermined values in contextual filters in detail.



F23.57

Configuration of contextual filters

Configuration of contextual filters using predetermined values. In this case you don't use the URL to pass on the filter value.

Practical case 23.2 Creating a view with contextual filters

In this practical example, we create, step-by-step, a view that displays a list of movies by genre on the page. Instead of creating an exposed filter in the view the genre comes from the URL, with the format:

- videostore/movies/**drama**
- videostore/movies/**horror**

The view has the following characteristics:

- **View name**: Movies by genre
- **Machine name**: movies_by_genre
- **URL alias:** videostore/movies/%
- **Page title**: Movies by genre
- **View type**: Content
- **Display**: Page
- **Format**: Table (because of this, we work with fields)

- **Page settings**:
    - Five elements displayed per page, including page elements.

- **Fields**:
    - Movie title (linked to node)
    - Country
    - Genre
    - Year
    - Director/s
    - Cast

- **Order criteria**:
    - The movies are sorted by node date of publication; in descending order (the latest movies first).

- **Filter criteria**:
    - Only Movie content types are displayed.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Contextual filters**:
    - The movie genre is passed on via the URL. Only movies which correspond with the indicated genre are displayed.

Step 1. Creating the view and initial configuration

The first step is to create the view from here:

Administration ⇨ Structure ⇨ Views [Add new view]

URL Add new view
/admin/structure/views/add

First we indicate the name of the view (**Movies by genre**) and change the system name (**movies_by_genre**). We choose to display **Content** of the **Movie** type, sorted by **Newest first**.

We create a **Page** display titled **Movies by genre** with the URL alias **videostore/movies/%**.

The % is the space reserved for the contextual filter, which we add later on.

The display Format is a **Table** that displays **5 items**.

We click on **Continue & edit** to continue building the view.

Step 2. Adding fields

We add the following fields to the view:

- **Movie title** (linked to node). This field corresponds to the node title.
- **Country**.
- **Genre**.
- **Year**.
- **Director/s**.
- **Cast**.

To add and configure the fields see **Step 2** of the **practical example 23.1**.

Step 3. Sorting and filtering

As we already set the view to display the latest published nodes first when we created the view, the criteria: **Content: Post date (desc)**, already appears correctly configured, in **Sort criteria**.

The required filter also appears as specified during view creation. It is called: **Content: Type (= Movie)**. The **Content: Published (Yes)** filter is there by default so only published content is displayed.

Step 4. Contextual filters

We add a contextual filter that matches the values in the URL. This filter corresponds to **Movie** content, which in turn refers to the **Movie Genre** vocabulary terms.

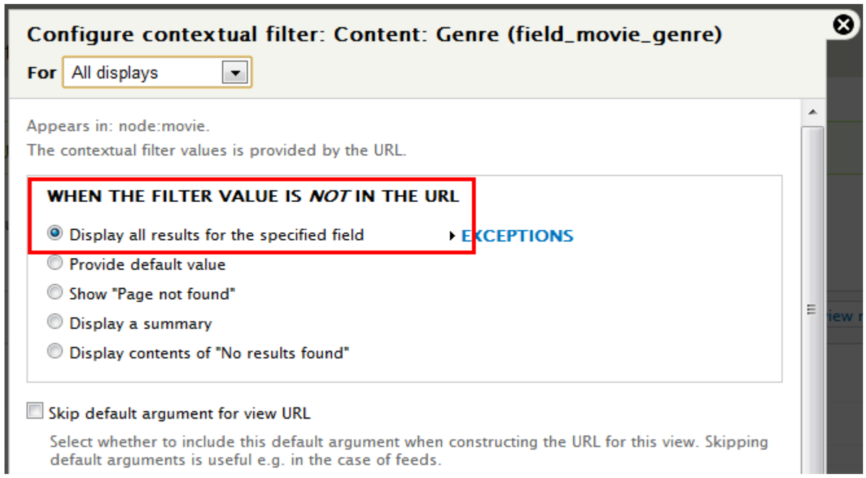
In the **Advanced** tab click on **add** (in **Contextual filters**). Locate and select the **Content: Genre** field.

In the **When the filter value is NOT in the URL** section we choose the value **"Display all results for the specified field"**. F23.58

F23.58

Practical case 23.2
Contextual filter
Configuring a contextual filter when it is not in the URL.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



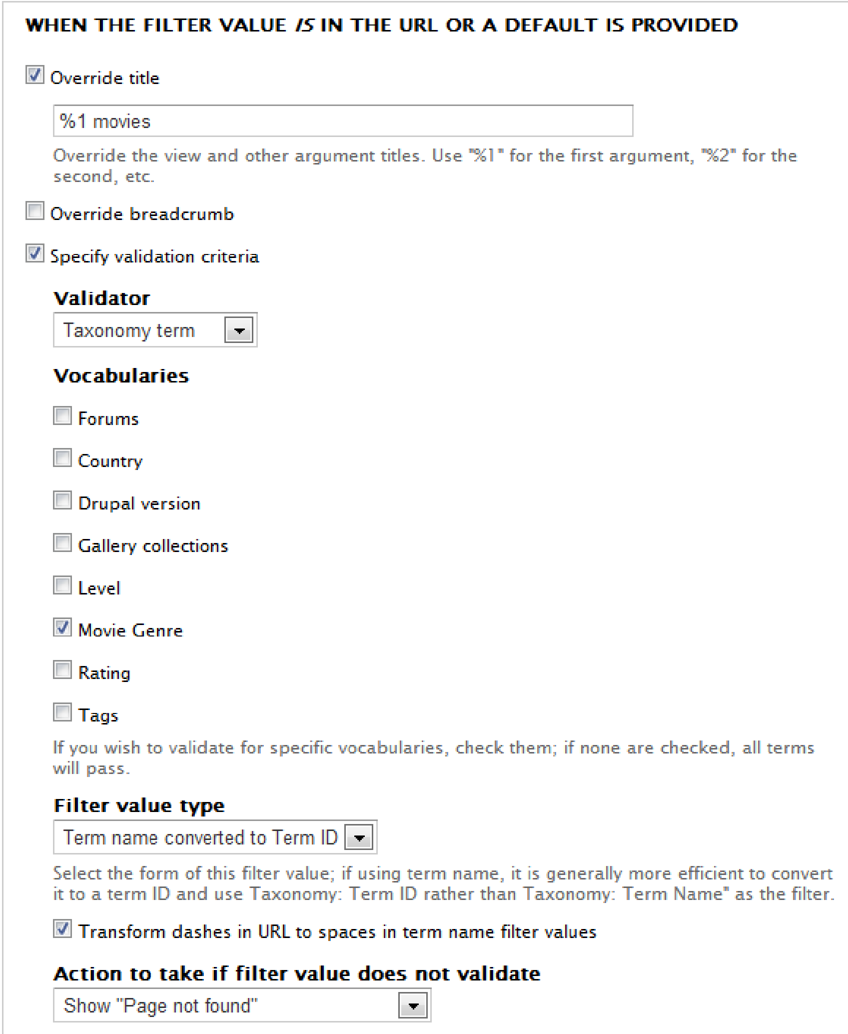
When the filter value IS in the URL: F23.59

- **Override title**. Allows us, via a replacement pattern, to create dynamic view titles, depending on the context. In this case, we use "%1 movies", with %1 being the first (and only) setting passed on to the view.

  For example, For the URL: "videostore/movies/horror", the page title: "Horror movies".

- **Specify validation criteria**. Checks the specified value. In this chase we check that it is a vocabulary term in the movie Genre vocabulary.

  - In **Validator** we select: **Taxonomy term**.
  - In **Vocabularies** we select: **Movie Genre**.
  - In **Filter value type** we select: **Term name converted to Term ID**. This option allows us to select a term name for the URL but to work internally with the term ID.
  - We activate the **Transform dashes in URL to spaces in term name filter values** option. This allows us to write a URL where the genre is several words long, for example: /videostore/movies/sci-fi (separating the words with dashes).
  - Lastly, if the filter value does not pass validation, a **Page not found error** is displayed.



F23.59

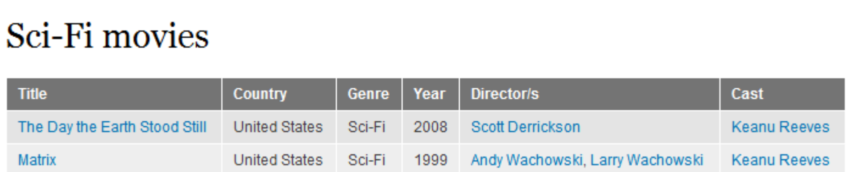
Practical case 23.2
Contextual filter

Configuration of the contextual filter. Because it is a field that refers to a taxonomy term you choose a validation criteria in the validator.

Indicate the vocabulary and the filter value: Term name converted to term ID.

In the **Figure** F23.60 we see the result of applying the filter via the URL:

- videostore/movies/sci-fi



Sci-Fi movies

Title	Country	Genre	Year	Director/s	Cast
The Day the Earth Stood Still	United States	Sci-Fi	2008	Scott Derrickson	Keanu Reeves
Matrix	United States	Sci-Fi	1999	Andy Wachowski, Larry Wachowski	Keanu Reeves

F23.60

Practical case 23.2
Resulting view

The resulting view after applying the filter via the URL: videostore/movies/sci-fi.

Step 5. Display all the results

Here we configure the contextual filter so that it allows us to display all movies independent of genre.

Within **When the filter value is NOT in the URL**, we go to the **Exceptions** options. F23.61

By default the exception value is **all**. To display all the results we use the URL: videostore/movies/all.

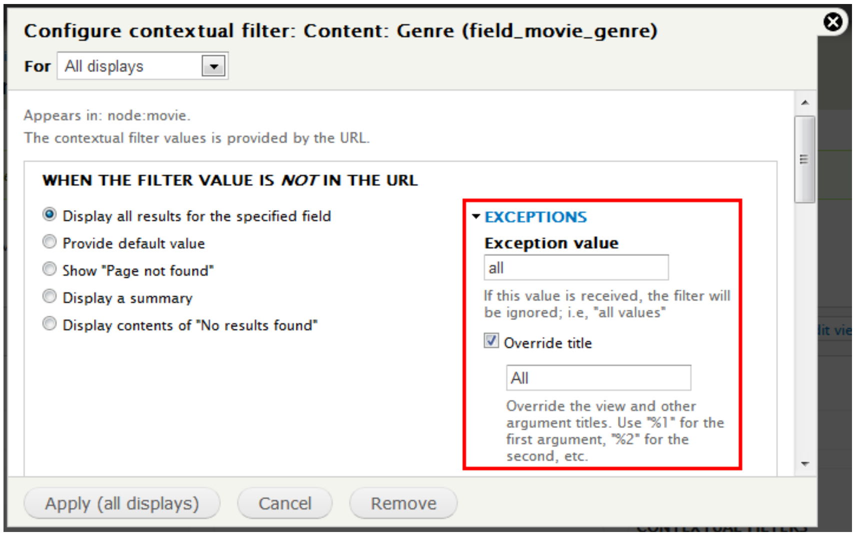
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

In the practical example we modify the exception value (all) and override the title (All movies).

F23.61

Practical example 23.2
Contextual filter exceptions

Apply the exception value, so that when you choose the all setting, all results are displayed.



With this configuration the URL **videostore/movies/all** displays all the movies. F23.62

F23.62

Practical example 23.2
All results

Upon applying the URL videostore/movies/all all movies are displayed. The title display is overridden for the exception.


All movies

Title	Country	Genre	Year	Director/s	Cast
Street Kings	United States	Action, Drama	2008	David Ayer	Keanu Reeves
The Social Network	United States	Drama	2010	David Fincher	Jesse Eisenberg
The Devil's Advocate	United States	Drama, Thriller, Horror	1997	Taylor Hackford	Keanu Reeves
Zombieland	United States	Action, Comedy, Horror	2009	Ruben Fleischer	Jesse Eisenberg
The Day the Earth Stood Still	United States	Sci-Fi	2008	Scott Derrickson	Keanu Reeves

1 2 next › last »

Practical case 23.3 Creating a view with default values in the contextual filters

In this practical example we create a block containing user profile information. Each user can only see their own information.

The view has the following characteristics:

- **View name**: My profile
- **Machine name**: my_profile
- **View type**: Users
- **Display**: Block
- **Format**: Unformatted list (of Fields)
- **Block settings**:
    - Block title/name: My profile
    - One element displayed per page, without pager.
    - The block is accessible to a registered user.
- **Fields**:
    - User name
    - User image
    - Name
    - Surnames
    - Roles
- **Sort criteria**:
    - Not required (only one result displayed).
- **Filter criteria**:
    - Not required. Only the contextual filter used.
- **Contextual filters**:
    - Only user information for the logged-in user is displayed.

Step 1. View creation and initial configuration

The first step is to create the view. For this go to: F23.63

Administration ⇨ Structure ⇨ Views [Add new view]

URL Add new view
/admin/structure/views/add

First we indicate the view name (**My profile**) and modify the system name (**my_profile**). We choose to display the view type, **Users** and sort by **Newest first**.

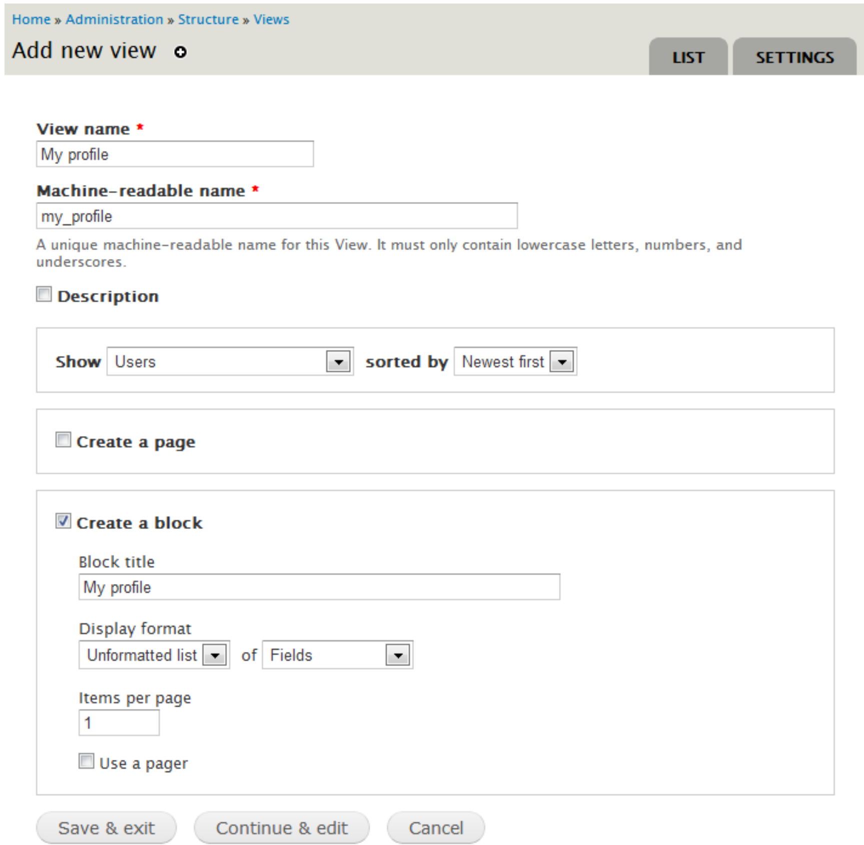
We create a **Block** display titled **My profile**.

As we are only displaying one result, that of the current user, we select **1 item per page**.

F23.63

Practical case 23.3
View creation

Creating the My Profile view as a block.

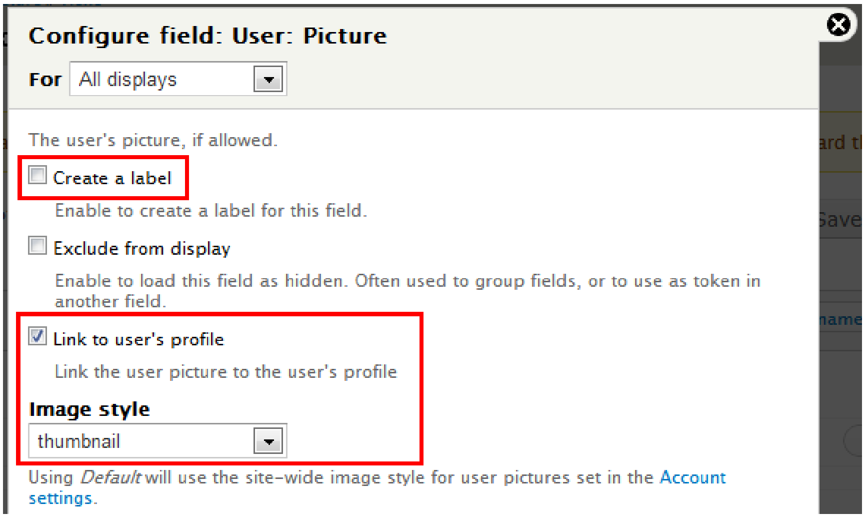


Step 2. Add fields

Having selected a **Users** view type, the fields we can add are related to the **User** entity. We add the following fields:

- **User: Name**. The user name with which the user accesses the website. We select the option "**Link this field to its user**". This link takes us to the user's profile.

- **User: Picture**. The profile image that the user has uploaded to their profile. We select "**Link to user's profile**" and in **Image style**, the **thumbnail** style. We delete the label for this field (deactivate **Create a label**). F23.64



F23.64

Practical case 23.3
Image Field

The image field links to the user profile. The image displays as a thumbnail.

- **User: First name** and **User: Last name**. These fields are extra fields added to the user entity. If not already created we create them now before continuing with the view.

- **User: Roles**. In **Display type** we select **Simple separator**, with the **Separator** comma (,). F23.65

F23.65

Practical case 23.3
Field Roles

The user roles are displayed as a list of elements separated by a comma (,).

Step 3. Contextual filter

To get the view block results to display only the fields for the current user we must add a contextual filter, selecting the field **User: Uid**.

In this instance, when the filter value is not available, we select **Provide default value: User ID from logged in user**. F23.66

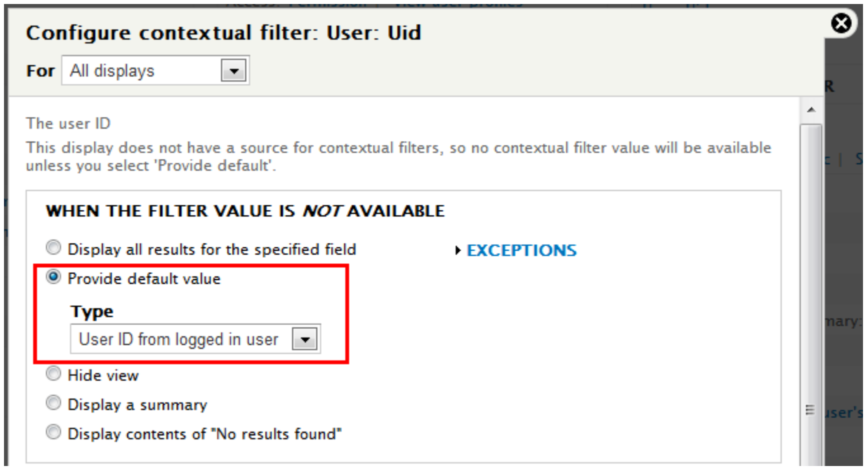
In this simple way we get the values displayed by the view to correspond only to the actual logged-in user.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F23.66

Practical case 23.3
Configuration of the contextual filter

Provide the default value: User ID from logged in user.



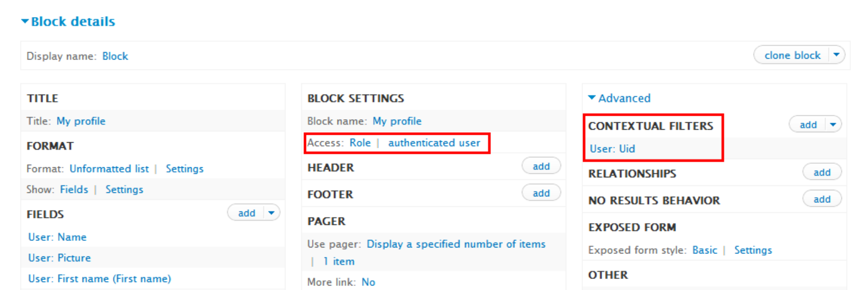
Step 4. Block activation

Before activating the block we must configure view access permissions. In **Block settings**, we select **Access: Role**. The role that can view the results is **authenticated user**. F23.67

F23.67

Practical case 23.3
Access control

Select access control by Role. Only authenticated users can access the view.



Once the view is saved the block is available in the Block administration area under the name **My profile**. We activate the block in the right side column and, to view the results, access the site as any logged-in user other than the user administrator.

The displayed block always corresponds to the user currently logged into the website. F23.68

F23.68

Practical case 23.3
Active block

Once the block is activated its displays to any logged-in user.

The information displayed is that of the logged-in user.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Relationships

23.8

Via relationships we can increase the number of queries made in a view. This is because it allows us to access related element fields that we initially are unable to access.

For example, in a view that displays content, we can display extra information about users (First name and Last name, for example) by establishing a relationship with the user who is the node author.

The **Views** module contains some of the most used relationships by default. This means that we do not need to add them to the view. If we added extra fields to the user profile (User entity), we do need to establish a relationship with the content author in order to gain access to the user's extra fields.

To add a relationship first go to the **Advanced** group and then click on **add** (within **Relationships**). Select one of the available fields to establish a relationship. F23.69

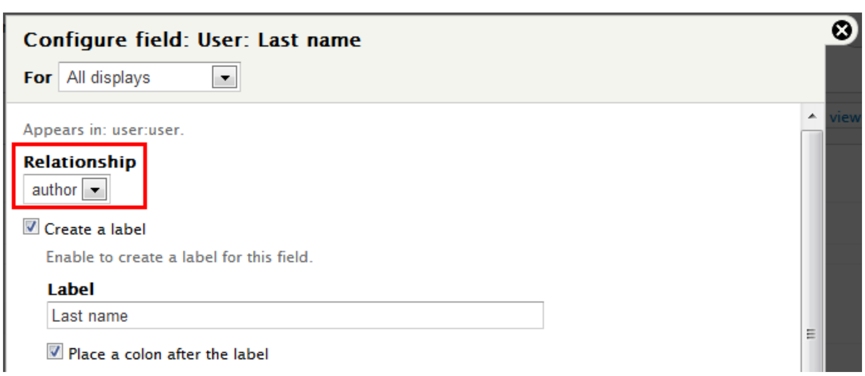
F23.69

Create a relationship

With relationships you can access extra fields.

For configuration purposes only one label is needed to identify the relationship along with the option **Require this relationship**. If this is activated the view only displays related elements.

Once a relationship is added new fields become available. Upon adding one of these fields we must indicate the relationship that we are using. F23.70



F23.70

Related field

Related fields are fields that are not accessible if you do not add the relationship In these fields you select the highlighted relationship.

To understand relationships better we will use a practical example.

Practical case 23.4 Adding related fields

To study the use of relationships, we will create a Content type view that requires us to establish a relationship with the author of the nodes in order to access extra user information.

The view has the following characteristics:

- **View name**: Site content
- **Machine name**: site_content
- **URL alias:** site-content
- **Page title**: Site content
- **View type**: Content
- **Display**: Page
- **Format**: Table (because of this, we work with fields)

- **Page settings**:
  - Five elements displayed per page, incorporating pager elements.

- **Fields**:
  - Content title (linked to node)
  - Content type
  - First name and Last name (grouped in a single column).
  - Post date
  - Content editing link

- **Order criteria**:
  - The contents are ordered by node publication date, in descending order (with the latest published content displayed first).

- **Filter criteria**:
  - Only published content is displayed.

Step 1. View creation and initial configuration

The first step is to create the view. Go to:

URL Add new view
/admin/structure/views/add

Administration ⇨ Structure ⇨ View [Add new view]

First we indicate the view name (**Site content**) and the system name (**site_content**). We select the view type, **Content**, of type **All** and sorted by **Newest first**.

We choose a **Page** type display titled **Site content** with the URL alias **/site-content**.

The **Display format** is **Table** and it displays **5 items**.

We click on **Continue & edit** to continue building the view.

Step 2. Adding unrelated fields

Unrelated fields are those with no direct relationship to the content:

- **Content: Title**. Added when we define the view. We create a Title label and link it the field value to its node.

- **Content: Type**.

- **Content: Post date**. We choose the short date format.
- **Node: Edit link**.

Step 3. Adding relationships and related fields

We establish the relationship via the **Content: Author** field. We find the field among the available relationships (**add** link in **Relationships**). The relationship identifier is **author**. F23.71

F23.71

Practical example 23.4
Adding relationships

Adding the author relationship via the Content: Author field.

Once we create the **author (Content: Author)** relationship, we return to the **Fields** section to add the missing fields; those related to the user who is the author of the content:

- **Author's first name** (**User: First name**).

- **Author's last name** (**User: Las name**).

In both fields the **Relationship** option is displayed, allowing us to select the **author** relationship created previously.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F23.72

Practical example 23.4
Related fields

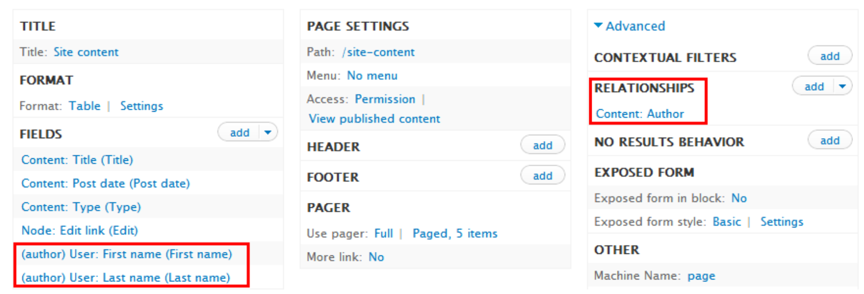
The fields User: Name and User: Surnames are fields related by the author relationship.

In the fields list, those fields that use a relationship are preceded by the relationship name.

F23.73

Practical example 23.4
Related fields

Related fields display along with the other fields, but indicaated by the relationship (author) as a prefix.



Step 4. Grouping Name and Last Name in the same column

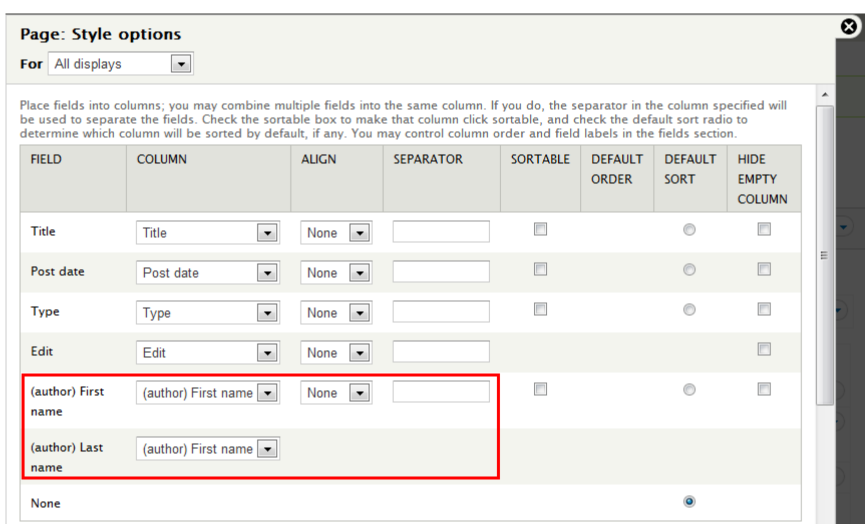
We reorder the fields and then go to the **Table** format **Options**. F23.74

We Modify the Surnames field column, giving it the same name as the Name field. In the Name field Separator we include a space ( ). In this column the name and surnames display together, separated by a space.

F23.74

Practical example 23.4
Grouping fields

Grouping fields in a single table column.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

The final result is shown in **Figure** F23.75 .

Site content

Title	Post date	Type	Edit	First name
Street Kings	06/02/2012 - 13:41	Movie	edit	Fran Gil
The Social Network	06/02/2012 - 13:38	Movie	edit	Fran Gil
The Devil's Advocate	06/02/2012 - 13:36	Movie	edit	Laura Fornié
Zombieland	06/02/2012 - 13:32	Movie	edit	Fran Gil
David Ayer	06/02/2012 - 13:25	Participant	edit	Fran Gil

1 2 3 4 5 6 7 next › last »

F23.75

Practical example 23.4
Final result

Final result of a view where you have used a relationship to have access to the name and last names of the author of each content.

Other advanced options

23.9

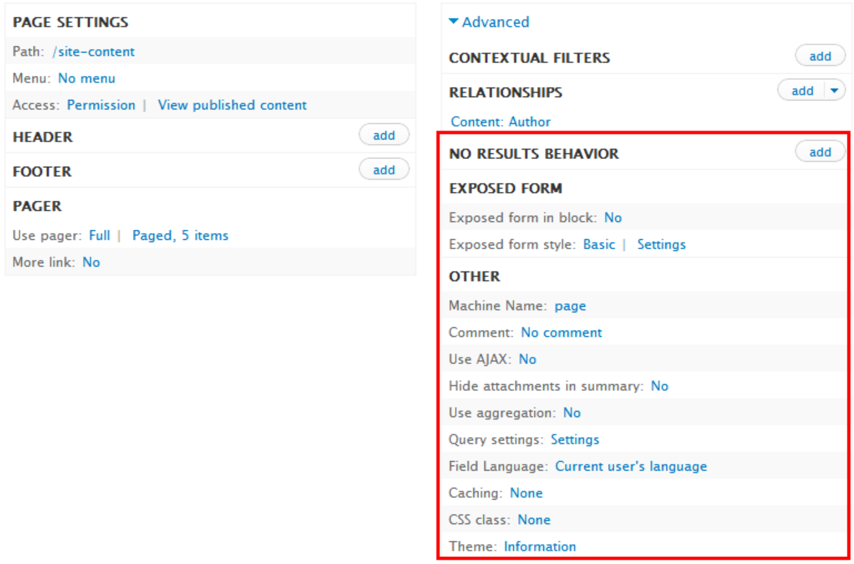
Within the Advanced tab we find other configuration options for the display: F23.76

- **No results behavior**. Allows us to add a **Text area** or **View area** that displays when the view shows no results.
- **Exposed form in block**. This option allows us to separate the views' exposed filters in an independent block. The filters do not display in the view and the block must be activated and configured manually from the block administration area.
- **Machine name**. Display system name. Made from the display name.
- **Comment**. A comment about the display just for administrative use. The final user does not see this content.
- **Use AJAX**. Uses AJAX for paging and the sorting of tables and exposed filters. We must bear in mind that, even if this makes navigation easier, the view results come from the first query and are not updated. Therefore, this option should only be used when the results do not update in "real time".
- **Hide attachments in summary**. Used when a contextual filter summary drops down.
- **Use aggregation**. Available for aggregation if activated.
- **Query settings**. We can't deactivate the SQL rewrite. Within options we can indicate that the results are Unique to prevent duplication.
- **Field language**. Specifies which language is displayed in the field.
- **Caching**. Allows us to activate the view cache.
- **CSS class**. We can indicate a CSS class that applies display styles to the view.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Theme**. Information about the templates (tpl.php files) that can be used to modify the view display.

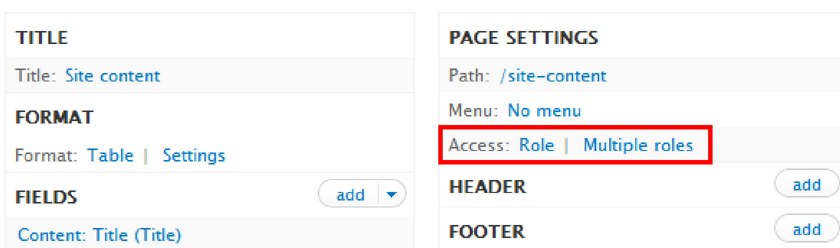
F23.76
Other options
Other view configuration options



23.10 Permissions

We have already seen how to apply access restrictions to each view display. We can do this for a permission or for one or more roles.

F23.77
View access control
You can control access by role or by permission.



The **Views** module also adds two general use permissions:

- **Administer views**. Only users with this permission can administer views (create, modify and delete).

- **Bypass views Access control**. A user with this permission can see all views, independent of any existing independent access control.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

PERMISSION	ANONYMOUS USER	AUTHENTICATED USER	ADMINISTRATOR
Views			
Administer views Access the views administration pages.	☐	☐	☑
Bypass views access control Bypass access control when accessing views.	☐	☐	☑

F23.78

Views module permissions

The module adds the Administer view and Bypass views Access controls permissions.

Additional modules

23.11

Views can be improved via additional modules. In this section we describe some of them in brief.

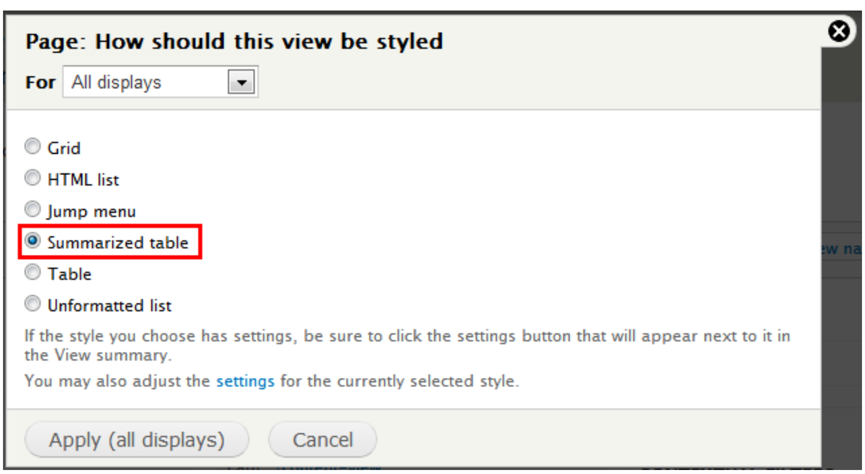
Views Summarize

The **Views Summarize** module adds a new display format called **Summarized table**. This new format allows us to add summaries and totals to the view results. For example, the sum total of a numerical field, the average of a field, etc.

The module is available at:

http://drupal.org/project/views_summarize

Once installed, the new **Summarized table** format is available when a view is created or edited. F23.79



F23.79

Views Summarize module

The Views Summarize module das a new views format.

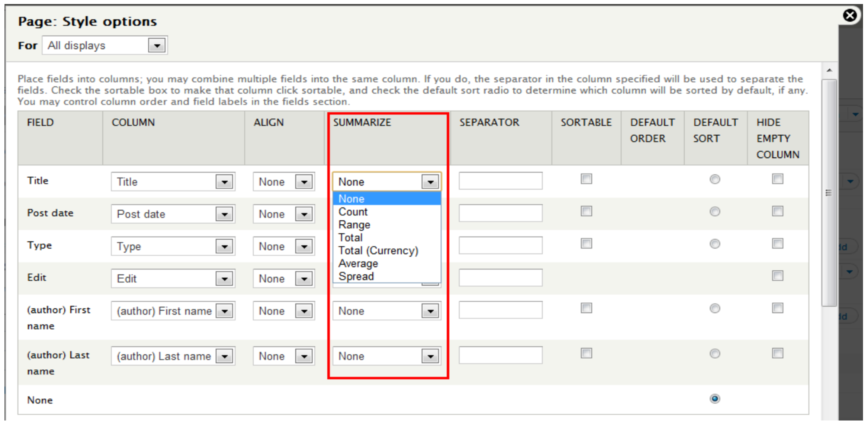
In the format **Settings**, which are inherited by all Table format options, a new column called **Summarize** is displayed. Here we can select the type of operation that is carried out on each field. F23.80

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F23.80

Views Sumaries module

In each column of the table you can indicate which operation is to be applied to all the listed values (add, count, average value, etc.).



The final result is displayed at the end of each column. F23.81

F23.81

Views Sumaries module

Example of module application. Capture taken from the official module page..

Nibh	367	27	27	Restricted
Ullamcorper	8403	34	34	Open
Nimis Patria	9558	56	56	Restricted
Cui Tation	3558	1	1	Secret
Duis Brevitas Typicus	3630	82	82	Restricted
Quia	7571	99	99	Classified
Vel Decet Eu	5536	31	31	Restricted
Dolor Oppeto Euismod	4959	57	57	Restricted
Lenis Eros Brevitas	1443	23	23	Restricted
Wisi Duis Dolor	6319	75	75	Restricted
Vicis Natu Similis	291	66	66	Secret
	Average: 4964.65	**Total:** 4915	**Range:** Min 0, Max 100	**Spread:** Open 28, Classified 20, Secret 26, Restricted 26

Views Bulk Operations (VBO)

The **Views Bulk Operations** (VBO) module adds checkbox-style selection boxes to the view results. The user can then carry out bulk operations on selected results.

The Views Bulk Operations module is available at:

http://drupal.org/project/views_bulk_operations

Once installed and activated we can add a new selection field in the view, identified as **Bulk operations**. In a Content style view we can add the field **Bulk operations: Content**, and in a Users view, the field **Bulk operations: User**.

If you have previously worked with VBO in Drupal 6, you must bear in mind one important difference: In Drupal 6 the VBO functionality is applied via the display format rather than via a field.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Upon adding a **Bulk operations** field we can configure the following options: F23.82

- **Enable "Select all items on all pages"**.
- **Force single**. Changes the checkbox selectors for radio buttons, meaning that only one value can be chosen.
- **Selected operations**. Selected operations. We choose the operations that can be carried out on selected content: Change content author, delete content, publish or unpublish content, etc.

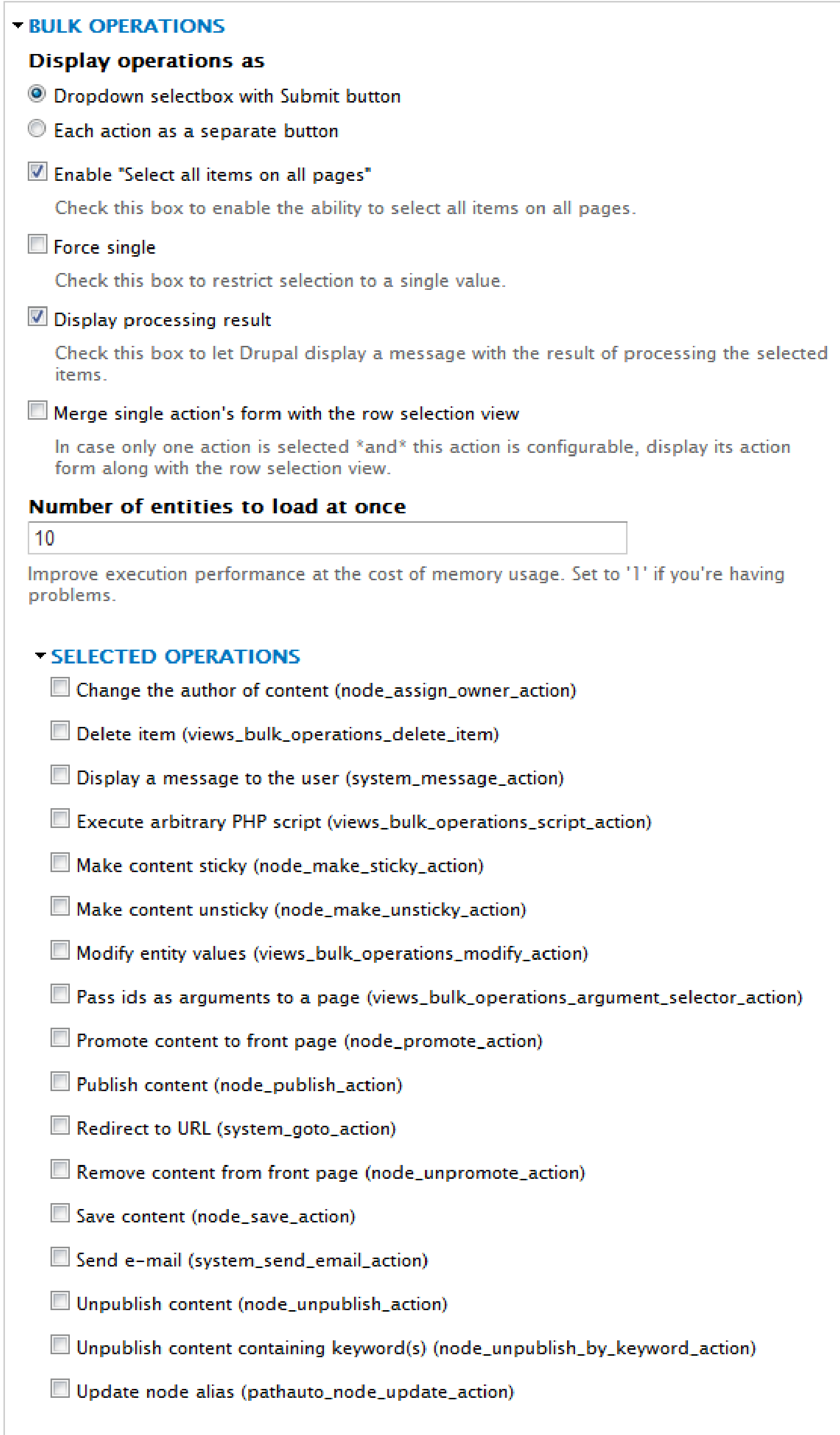
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F23.82

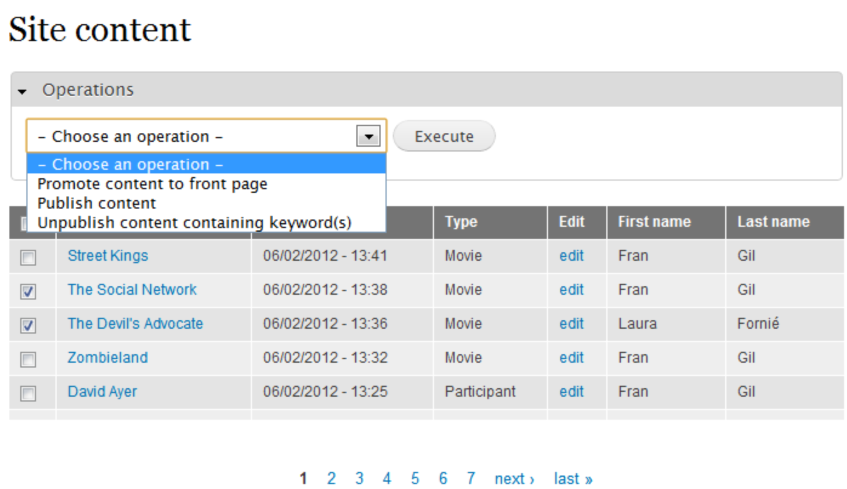
Views Bulk Operations

Configuration of a VBO type field.

The module allows you to add VBO type fields. These fields add a selector to the results, initially in the form of a checkbox. The user can choose between the results and then apply any of the available operations.



The resulting view displays a selection element next to each result and a new operation filter. The indicated operation is executed on the selected elements. F23.83



F23.83

Example of a view with a VBO field

Example of a view with a VBO field with three operations

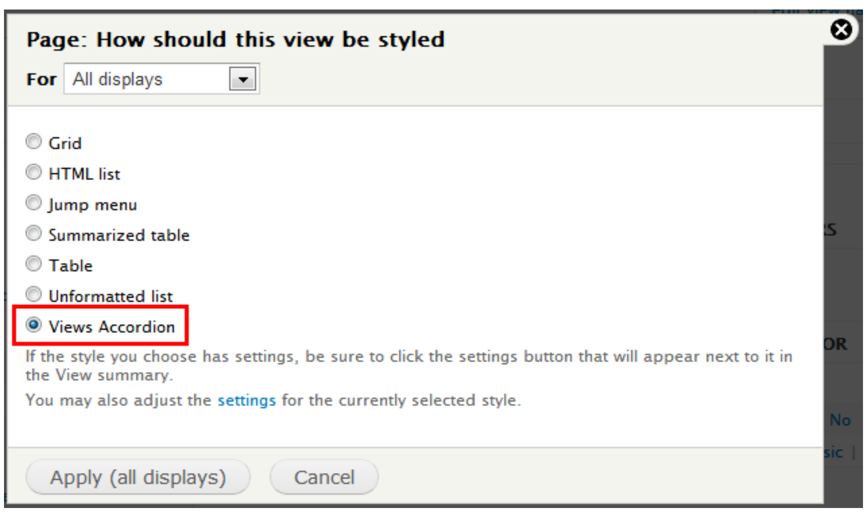
Views Accordion

The **Views Accordion** module adds a new display format to the views, called **Views Accordion**. Applying this style displays the contents as an accordion with JQuery: All the results are folded until the title is clicked (or the first field value).

The Views Accordion module is available at:

http://drupal.org/project/views_accordion

Once installed, we can choose the **Views Accordion** format or style. **F23.84**



F23.84

Views Accordion module

Adds a new style that displays the results as an accordion.

Among other options we can configure the following:

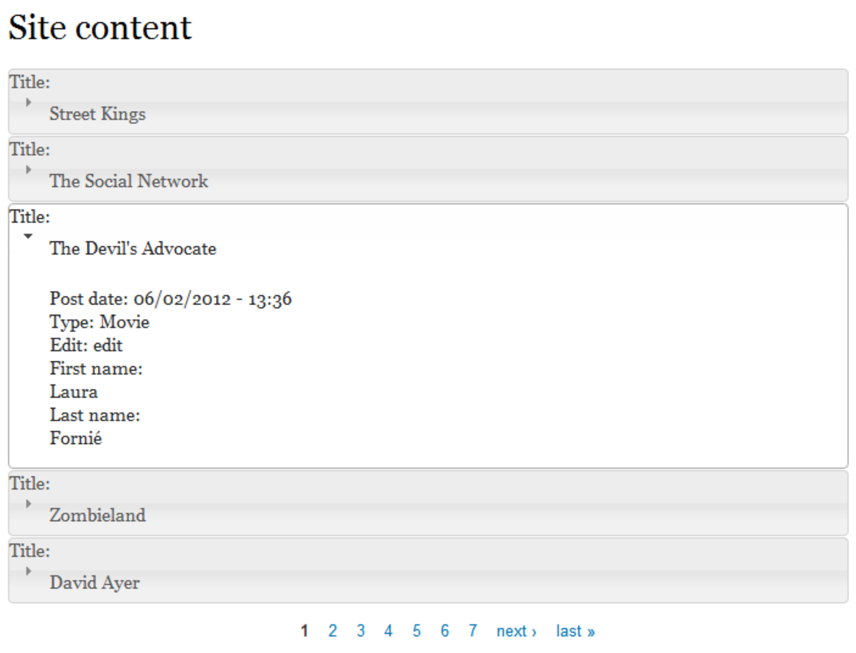
- **Row to display opened on start**. This is by default the first row (Row 1), but we can also select None so that all the results are displayed folded.

- **Animation effect**. The module includes different effects that are applied when the elements Fold or unfold.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Event**. By default it is Click, but it can also be executed by a Mouseover.

In the **Figure** F23.85 we see an example of a view folded like an accordion. Clicking on any of the titles open the new element and closes the previous one.



F23.85
Views Accordion module
Display of a view in accordion mode. The results remain closed until you click on the title.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

24 Automatic image processing

Automatic image processing is essential to maintain a homogenous and well-structured appearance across all images on a site. For example, if we publish a list of people with their photos, we want all the images to be the same size.

Drupal allows us to modify images uploaded to the site automatically, meaning that we don't have to pre-process them with a third-party application. This applies to images that form part of the site content but not to images that are embedded in the site's theme as part of its general design.

The Image module, included in the core, allows us to make automatic modifications to images. Complementing this module are others that add extra effects that can be applied to images.

We will also learn how to integrate automatic image modifications with content types (and entities) and views.

Comparative D7/D6

ImageCache

In **Drupal 7** the tools that allow automatic image processing are included in the core via the Image module.

In previous versions of Drupal this functionality was included in the site using the **ImageCache** module, which is no longer needed. We can however, install other modules related to ImageCache that add functionalities to image processing in Drupal 7.

The Image module in Drupal 7 works in a similar way to the ImageCache module from previous versions. Now, instead of presets, we talk about **Image Styles**, and instead of actions, **Effects**.

Unit contents

24.1 Image styles..136
24.2 Image styles in Fields...141
24.3 Image styles in Views..142
24.4 Additional modules...144

24

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

24.1 Image styles

In **Drupal 7** the tools that permit automatic image processing and modification are included in the core via the **Image** module.

In previous versions of Drupal this functionality was included in the site using the **ImageCache** module, which is no longer needed. We also can install, as this unit teaches us, other modules related to ImageCache. These add extra image processing functionalities in Drupal 7.

The Drupal 7 Image module works in a similar way to ImageCache in previous versions. We will cover **styles**, which are groups of actions or effects that are applied to images. For example, we can indicate that images must be reduced to a set size, eliminating any excess pixels. Once these styles are created we need to define where and on which photos they are to be applied. For example, we want a specific style to be applied to all the user profile images on the site.

To create new styles or configure existing styles, we go to: F24.1

URL Image styles
/admin/config/media/image-styles

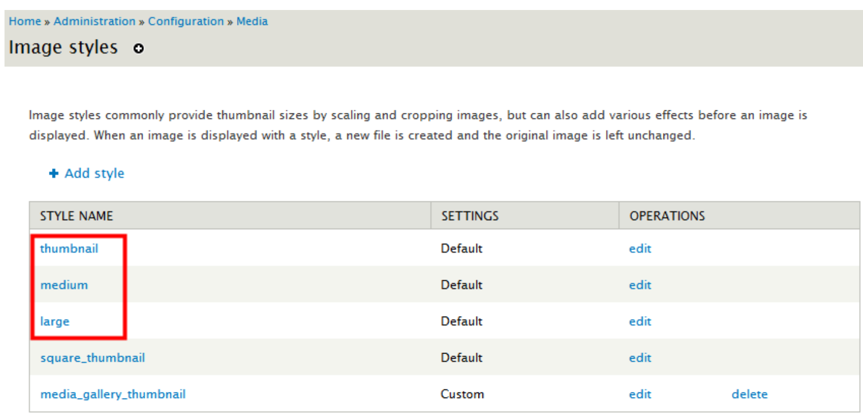
Administration ⇨ Configuration ⇨ Media ⇨ Image styles

By default the available styles are **thumbnail**, **medium** and **large**. Other styles may be displayed, added by extra modules that are already installed.

F24.1

Image styles

List of image styles available on the site. Each style consists of a set of effects that are applied to the image.



Add a new style

To create a new style we click on the **Add style** link, then give the style a valid name (using alphanumeric characters, underscores and/or hyphens).

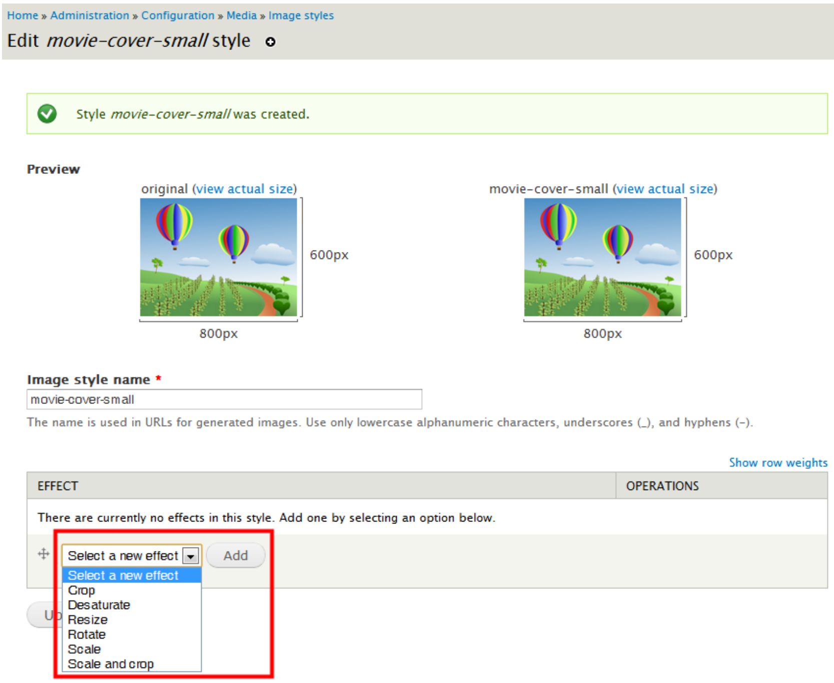
After inputting the style name a list of the available effects is displayed. F24.2

F24.2

Available effects

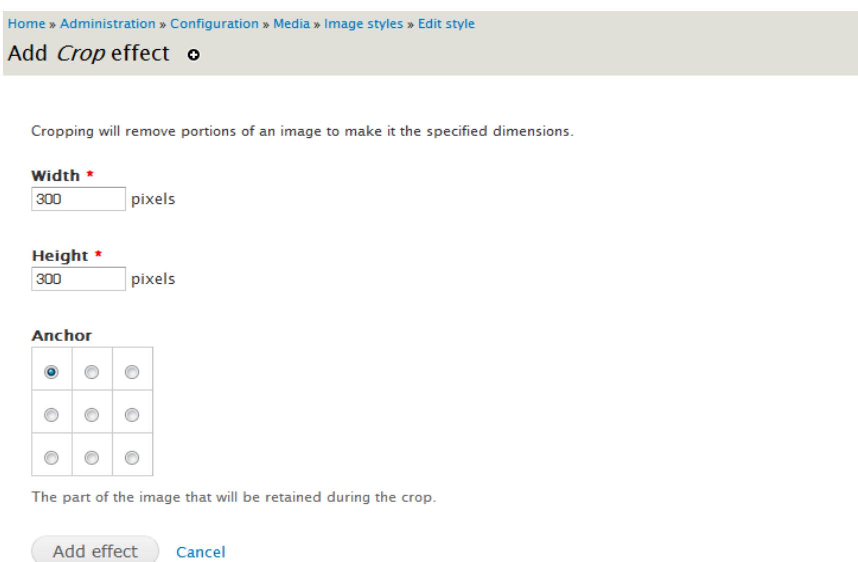
The effects available depend on the modules installed. You must bear in mind that the order the effects are applied in affects the final appearance of the images the style is applied to.

At the top of the page a preview appears that displays an example image and an image showing the result of the effects applied by the style. The image preview is displayed in miniature but can be viewed full size by clicking on the "**view actual size**" links. Initially the two preview images are identical as no effects are applied unless activated.



The available effects are:

- **Crop**. Crops the image to the size specified in pixels (width and height). We can decide what part of the image is preserved and what part is cropped away. F24.3
  In **Figure** F24.4 we see the image, cropped from the top left corner, with a final height and width of 300x300px.



F24.3

Crop effect

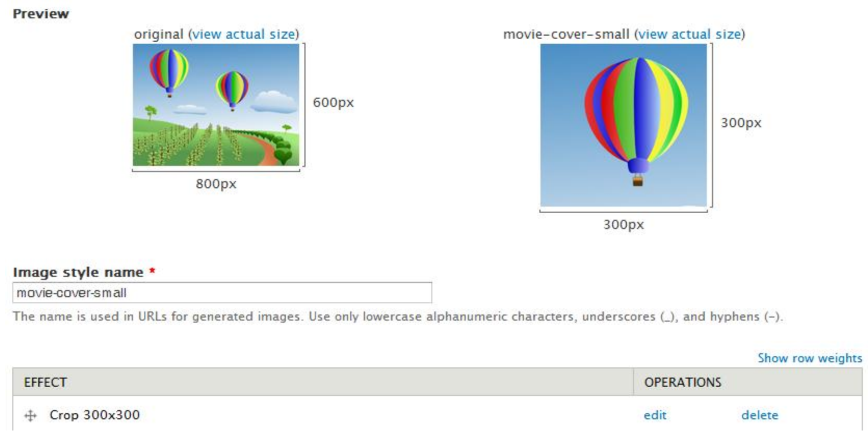
Crop effect settings. You can choose an anchor point: It defines the part of the image that is retained during cropping.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F24.4

Crop effect

Preview of an image cropped to 300x300px with the anchor point set in the top left corner.

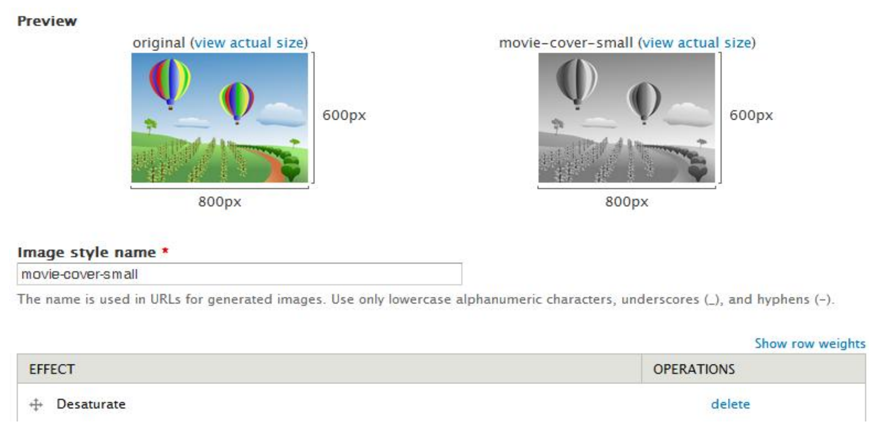


- **Desaturate**. Converts the image to greyscale. F24.5

F24.5

Desaturate effect

The desaturate effect removes all colours from the image, converting it to greyscale.



- **Resize**. Allows us to change the width and height of the image to specified values. The resize effect does not maintain image proportions and can cause the image to appear stretched or elongated. F24.6

F24.6

Resize effect

Changes the width and height of the image without maintaining its proportions. In consequence the image can appear distorted (stretched or elongated).

- **Rotate**. Allows us to rotate the image by a specified angle. A positive value rotates clockwise, a negative value anticlockwise. F24.7

F24.7

Rotate effect

Rotates the image by the specified amount.

- **Scale**. Allows us to rescale the image to a specified width and height. Unlike the Resize effect, Scale maintains the images' original proportions. If we only specify one value (width or height) the new image maintains its original proportions. If we specify both values the image adjusts to the new scale but keeps its original proportions. The option "Allow upscaling" allows an image to be increased in scale to the specified dimension (it may lose quality). F24.8

F24.8

Scale effect

Changes the image scale but maintains its proportions.

- **Scale and crop**. Crops and rescales an image to exact dimensions without losing its original proportions. Useful, for example, when images are to be displayed together in a gallery and need to be the same height and width. The image is first scaled and then cropped on its longest side (centred). F24.9

F24.9

Scale and crop

Scales the image to the specified dimensions and then crops away any extra pixels. This allows you to create an image of exact dimensions and to maintain its proportions.

Combining effects in a style

We can add as many effects as are required into a single style.

The order of the effects determines the order of execution: It is important to get the order right.

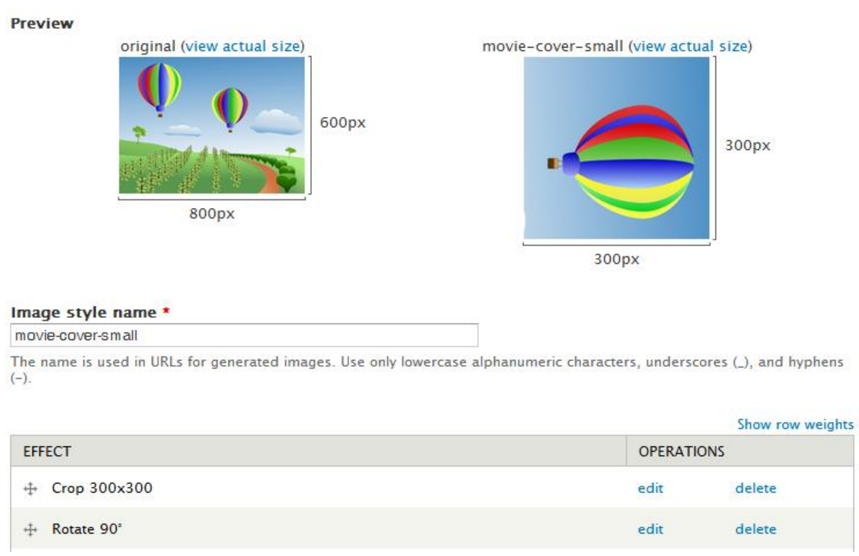
In **Figure** F24.10 we see the result of applying effects:

- **Crop 300x300px** (anchored in the top left corner).
- **Rotate 90º**.

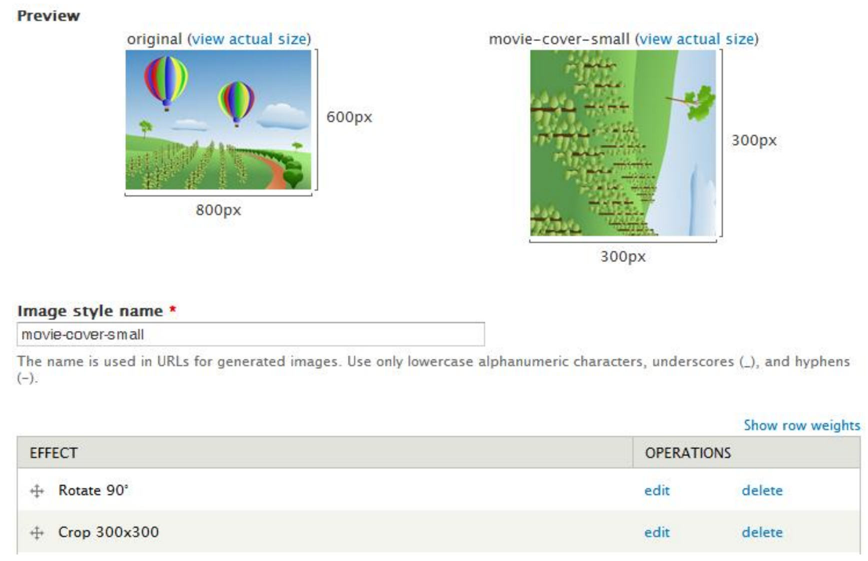
F24.10

Combined effects

You can apply several effects in the same style. In this example first the Crop effect (300x300px) and then Rotate 90º.



And in **Figure** F24.11 we see the result of applying the same effects but in a different order.



F24.11

Combining effects

The order of the effects determines the result of applying a style.

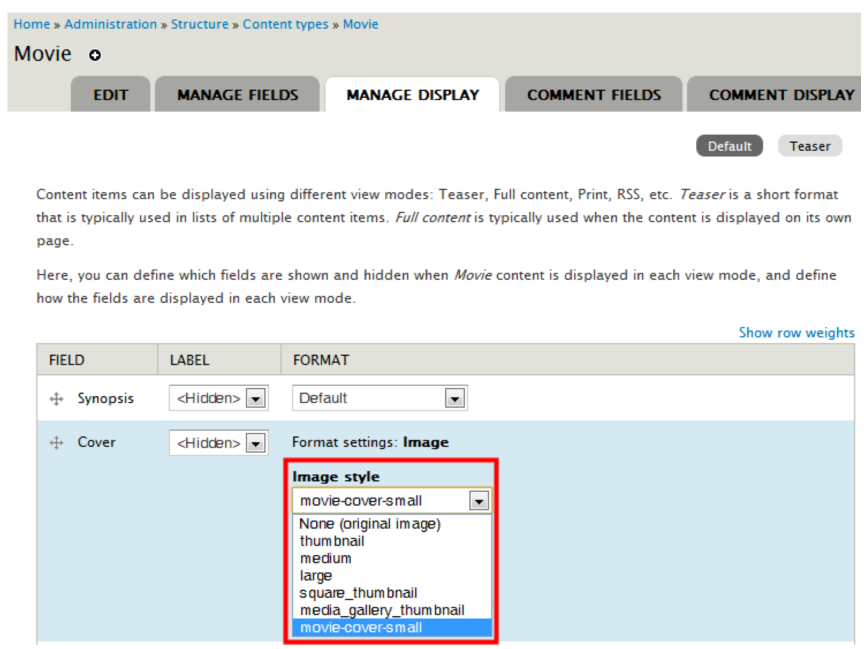
In this example the order of the effects is reversed. First we rotate the image by 90º and then crop to 300x300px.

Image styles in Fields

24.2

In both the beginner and intermediate levels we have seen how to create Image fields in entities. Now that we know how to create styles we can apply them to image fields created in content types (and other entities).

To apply a style to a field we go to the content type **Manage display** tab. In the **Image style** drop down menu we can select any previously created F24.12 styles.



F24.12

Image styles in fields

You can select any of the image styles available when you select the Image format.

In some cases, such as when we use the lightbox module (Colorbox type or Lightbox2), TWO styles are required. One is applied to the image in the node and the other is applied when the lightbox window is opened.

It is important to bear in mind that the style is applied to the image when it is displayed but does not change the original image uploaded to the site. This allows us to change styles at any time without having to re-upload the images.

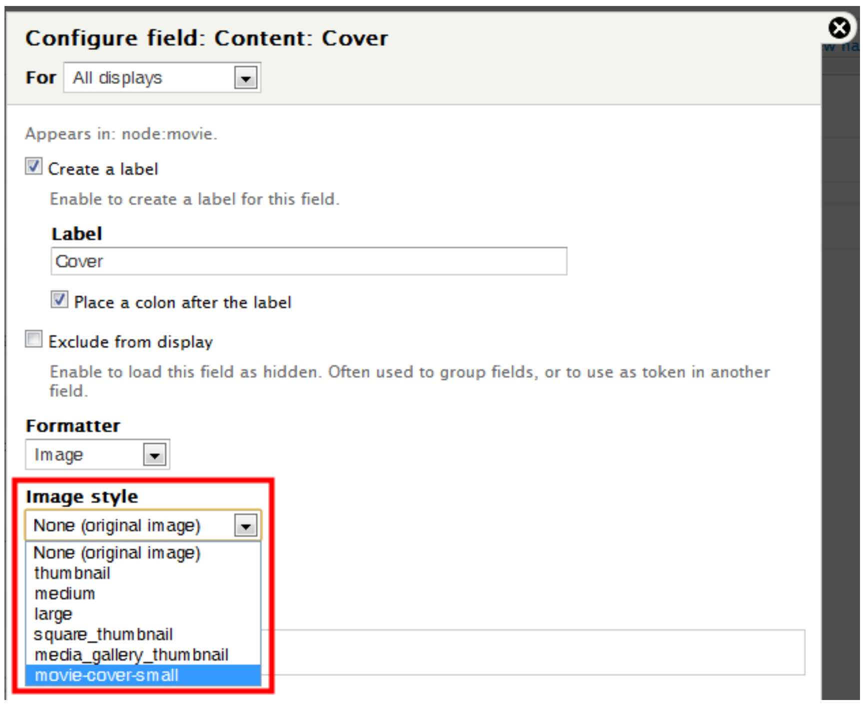
24.3 Image styles in Views

In views we can also add image fields and associate image styles with them. As well as selecting the available **Formatter** (Image, Colorbox, Lightbox2, etc.), we can indicate the **Image style** to be applied. F24.13

F24.13

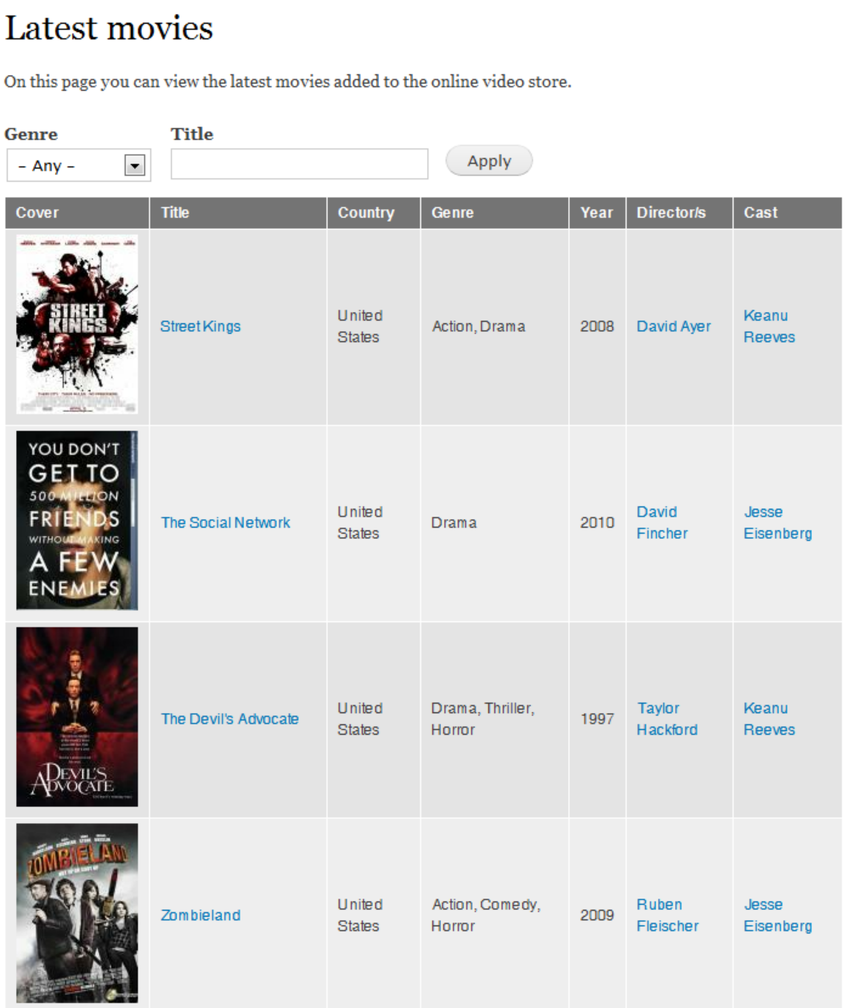
Image styles in views

In image fields added to views you can also indicate the image style to apply.



In **Figure** F24.14 we see the **Latest movies** view. We have added a field containing a thumbnail of the movie cover image, and applied the **Scale and crop (100x150px)** effect.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



Latest movies

On this page you can view the latest movies added to the online video store.

Genre – Any –   **Title**   Apply

Cover	Title	Country	Genre	Year	Director/s	Cast
	Street Kings	United States	Action, Drama	2008	David Ayer	Keanu Reeves
	The Social Network	United States	Drama	2010	David Fincher	Jesse Eisenberg
	The Devil's Advocate	United States	Drama, Thriller, Horror	1997	Taylor Hackford	Keanu Reeves
	Zombieland	United States	Action, Comedy, Horror	2009	Ruben Fleischer	Jesse Eisenberg

F24.14

Example of a view with images

The Latest movies view with a style applied to the images.

24.4 Additional modules

The additional modules that we are about to cover add new effects that can be applied to image styles. Most of the modules are identified by the name **ImageCache**: The name of the module that acted as a base for automatic image treatment in previous versions of Drupal.

Imagecache Proportions module

The **Imagecache Proportions** module allows us to select, within image fields, different image styles depending on whether the image is vertical, horizontal or squared.

The Imagecache Proportions module is available at:

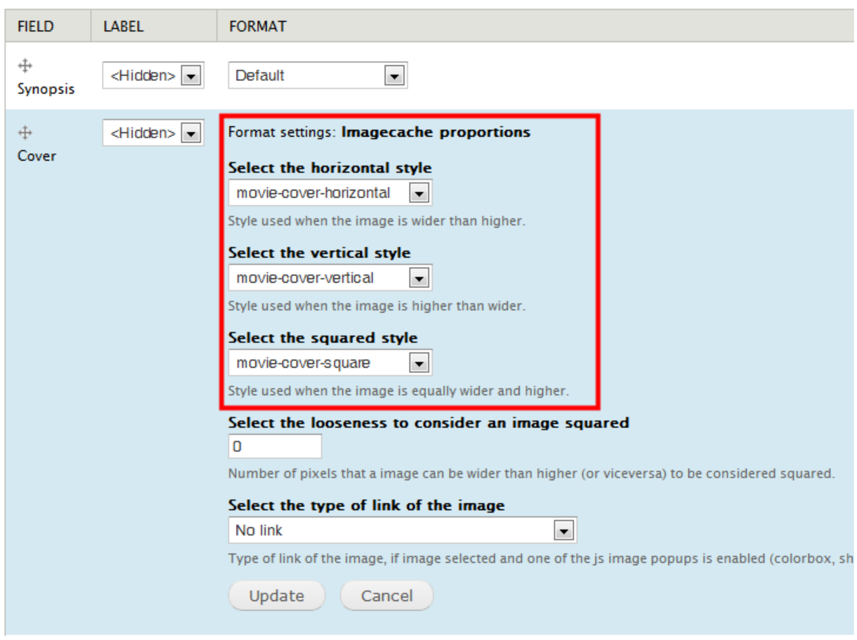
http://drupal.org/project/imagecache_proportions

To apply this functionality we first create the three image styles. From the **Manage display** tab for the content type (or entity), we select the **Imagecache proportions format**. Amongst the configuration options of this new format we can select different styles for each image style: horizontal, vertical and squared. F24.15

F24.15

Imagecache Proportions module

Allows you to apply different image styles within a field depending on whether the images are horizontal, vertical or squared.



Imagecache Actions module

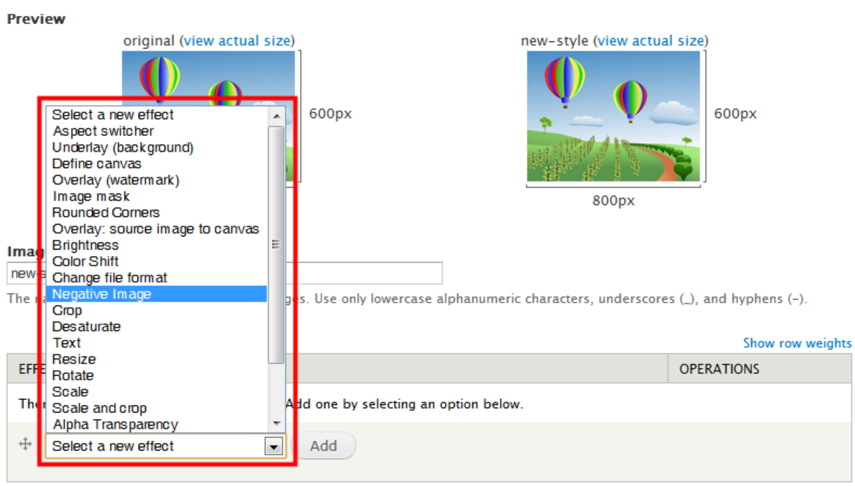
The **Imagecache Actions** module adds new effects that can be applied in image styles.

The Imagecache Actions module is available at:

http://drupal.org/project/imagecache_actions

The module includes the following submodules:

- **Imagecache Actions**. Acts as a base for the other modules.
- **Image Text Effects**. Adds text to the images.
- **ImageCache Autorotate**. Allows us to rotate images.
- **ImageCache Canvas Actions**. Adds actions over the image canvas, allowing us to add watermarks, backgrounds effects, etc.
- **ImageCache Color Actions**. Adds effects related to the colour, brightness and transparency of the images.
- **ImageCache Custom Actions**. Allows us to manipulate images directly via PHP. We leave this module deactivated.
- **ImageCache_actions Test Suite**. Adds a set of example styles.



F24.16
Imagecache Actions module
Adds new effects that you can apply to any image style, combining them with each other, and with the effects provided in the module core.

Among the effects added by **Imagecache Actions** are: F24.16

- **Brightness**. Allows us to adjust the image brightness.
- **Underlay** (**background**). Allows us to add a background image.
- **Color Shift**. Allows us to add a colour filter to the image.
- **Rounded Corners**. Allows us to add rounded corners to an image.
- **Overlay** (**watermark**). Places a second image over the original as a watermark. We must indicate the location of the image to be added (which must be in the files directory), the image position, and its

opacity.

- **Text**. Allows us to add static or dynamic text (using PHP).

In **Figure** F24.17 we see the **corners_combo**, the example style that is added by the module.

F24.17

Imagecache Actions Module

Extra Effects applied with Imagecache Actions.

The corners_combo image style included in the example module.

Preview

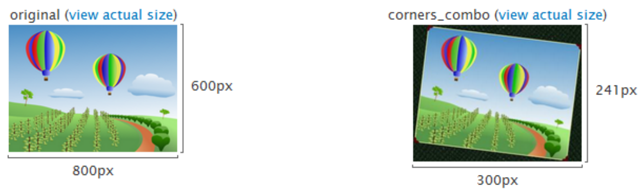


Image style name

corners_combo

This image style is being provided by *imagecache_testsuite* module and may not be renamed.

Show row weights

EFFECT	WEIGHT	OPERATIONS
Define canvas left:4 right:4 top:4 bottom:4 #e5e8a1 **under** image		
Define canvas left:4 right:4 top:4 bottom:4 Transparent **under** image		
Overlay (watermark) **sites/all/modules/imagecache_actions/tests/corner-tl.png** x:left, y:top alpha:100%		
Overlay (watermark) **sites/all/modules/imagecache_actions/tests/corner-tr.png** x:right, y:top alpha:100%		
Overlay (watermark) **sites/all/modules/imagecache_actions/tests/corner-bl.png** x:left, y:bottom alpha:100%		
Overlay (watermark) **sites/all/modules/imagecache_actions/tests/corner-br.png** x:right, y:bottom alpha:100%		
Rotate 7°		
Scale width 300		
Underlay (background) xpos: , ypos: alpha:100%. file: sites/all/modules/imagecache_actions/tests/background-2.jpg, dimensions:original		

Override defaults

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

25 Extending menus with additional modules

In the Beginners Level we studied the basics of creating menus in Drupal. In this unit we add to the menu functionalities using the following extra modules:

- **Menu Breadcrumb** allows breadcrumbs for any site menu.
- **Custom breadcrumbs**, allows us to create customized breadcrumbs depending on the content type.
- **Menu Attributes**, adds additional attributes to the menu links.
- **Special Menu Items**, allows us to create menu elements without links, for use as separators and categories.
- **Menu Block**, creates menu blocks from other menus (submenus).
- **Menu per role** and **Menu item visibility**, allows us to display menu items to specific users depending on their roles.
- **Nice menus**, creates drop-down menus.
- **Menu Icons**, allows us to add icons and images to menu links.
- **Administration Menu**, provides a drop-down menu with all options, sub-options and administration menu tabs.

Comparative D7/D6

All the modules in this unit have a corresponding version in Drupal 6 with almost identical functionalities.

Some of the modules you will study are not yet available as stable versions. They may therefore produce unwanted functionalities or be affected by other installed modules.

Unit contents

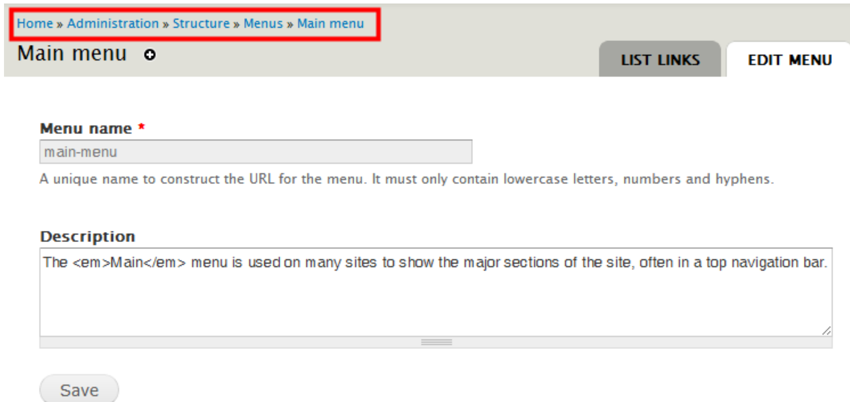
25.1 Breadcrumb Display....................148
25.2 Custom breadcrumbs....................151
25.3 Menu link improvements....................153
25.4 Building submenus with Menu Block....................158
25.5 Menu link access control....................161
25.6 Improving menu display....................162

25

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

25.1 Breadcrumb Display

Breadcrumbs provide the user with information about the structure of the site and its exact location, allowing them to access pages and previous sections with ease. F25.1



F25.1

Breadcrumbs

Breadcrumbs provide information about the location of the page in relation to the site's menu structure.

There are two main types of breadcrumb:

- **Location breadcrumbs,** connected to the site content structure. Provide information about page location and are independent of user navigation. When a specific page is loaded the breadcrumb structure is always the same (provided the site structure remains unaltered).

- **Dynamic breadcrumbs,** connected to navigation. Normally used for sites where information is located in different places and has multiple access pathways. In this case the displayed pathway is the one followed by the user. It allows the user to reverse their steps to return to previously visited pages.

The functionality of breadcrumbs (**location breadcrumbs**) in Drupal is described first. These show the location of a loaded page as part of a static information structure.

Drupal only shows breadcrumbs for pages that are linked to the Administration menu.

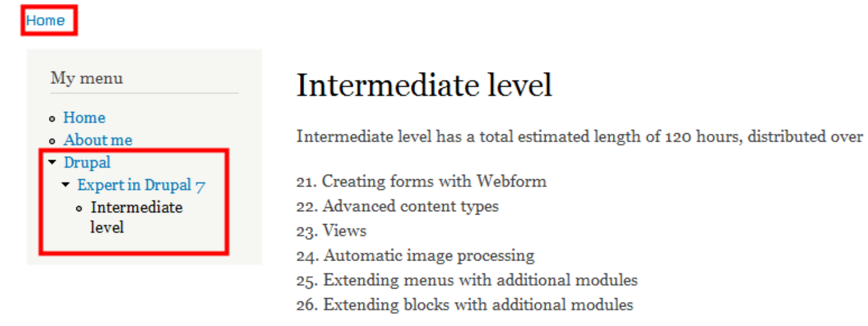
Menu Breadcrumb module

The **Menu Breadcrumb** module allows breadcrumbs to be displayed for any site menu.

The module is available at:

http://drupal.org/project/menu_breadcrumb

We can see how breadcrumbs are displayed for a page linked from the *My Menu* menu, without the Menu breadcrumb module being installed in the site. As we can see the breadcrumbs only show the link to Home. F25.2

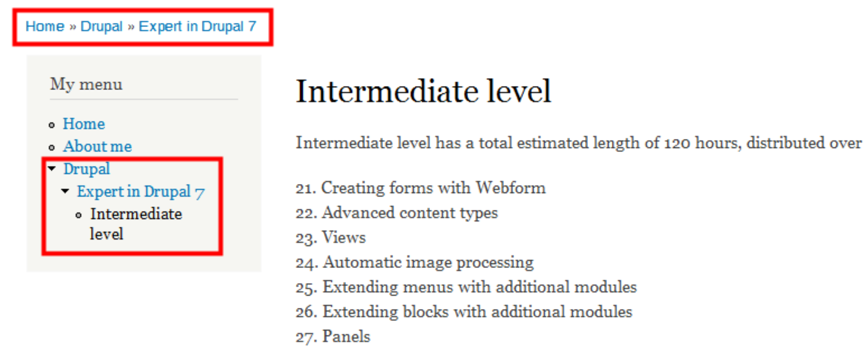


F25.2

Default breadcrumbs

Initially breadcrumbs are only available for content linked from the administration menu. For other content the only link displayed is one back to the main site page (for example, Home).

However, if we install and activate the **Menu breadcrumb** module, breadcrumbs display the complete menu pathway, from *My Menu* to the loaded page. **F25.3**



F25.3

Breadcrumbs with Menu Breadcrumb

Once you configure the module, breadcrumbs are available for content linked to menus other than the administration menu.

We access the module administration menu here:

Administration ⇨ Configuration ⇨ User interface ⇨ Menu Breadcrumb

URL Menu Breadcrumb
/admin/config/user-interface/menu-breadcrumb

The available options are: **F25.4**

- **Use menu the page belongs to for the breadcrumb**. This option uses the menu a page belongs to for the breadcrumb. It is the default option as it provides the module's main functionality.

- **Append page title to breadcrumb**. Adds the title of the loaded page to the breadcrumbs. The title will be displayed without a link.

- **Appended page title as an URL**. This complements the previous option, adding the loaded page title to the breadcrumbs, but with a link to the page.

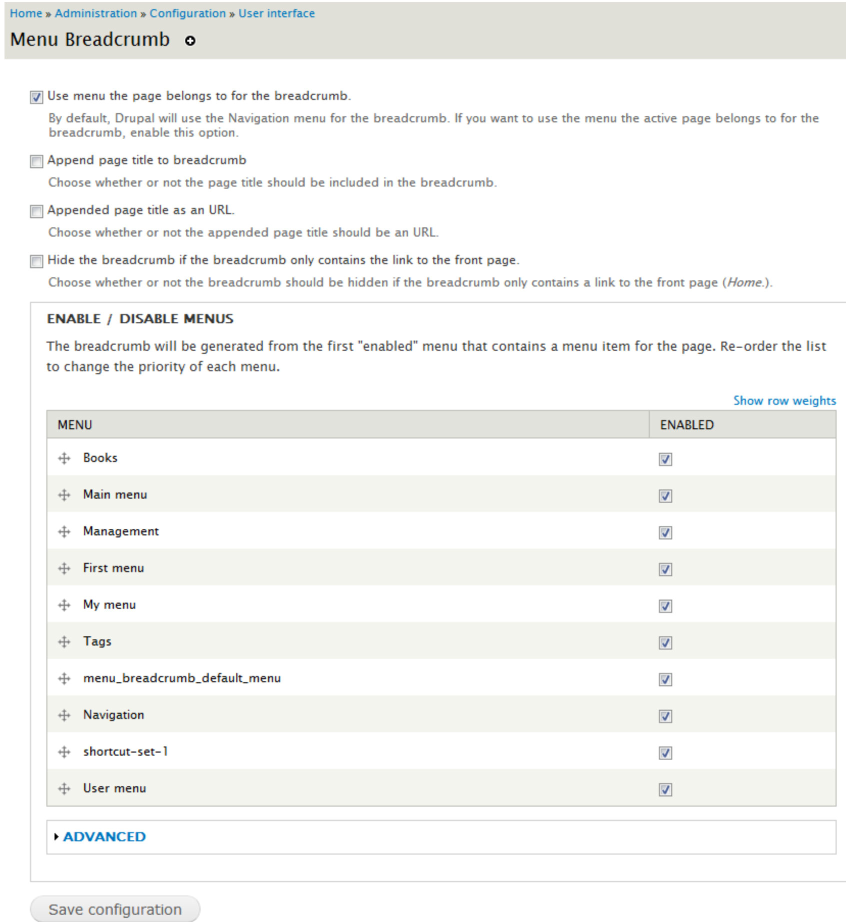
- **Hide the breadcrumb if the breadcrumb only contains the link to the front page**. No breadcrumbs are displayed for beginner level options and pages that are not linked to a menu. Unless we activate this option a link to the Home page is always displayed.

- **Enable/disable menus**. Allows us to activate and deactivate breadcrumbs for each menu available on the site. When a node is linked to several menus, the breadcrumb displayed will correspond to the first active menu found. From this option we can also reorder the menus to indicate breadcrumb order of preference.

F25.4

Menu breadcrumb. Settings

Choose which menus to activate breadcrumbs for. You can also indicate whether the title of the loaded page is included in the breadcrumbs.



Once the module is configured, for all content linked from the activated menus, the breadcrumbs display correctly.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Custom breadcrumbs

25.2

Until now we have seen how to get Drupal to display breadcrumbs correctly when viewed content is linked from a menu. If we need specific content to show customized breadcrumbs, we use the **Custom breadcrumbs** module.

The **Custom breadcrumbs** module allows us to create customized breadcrumbs depending on the content type. The module is available at:

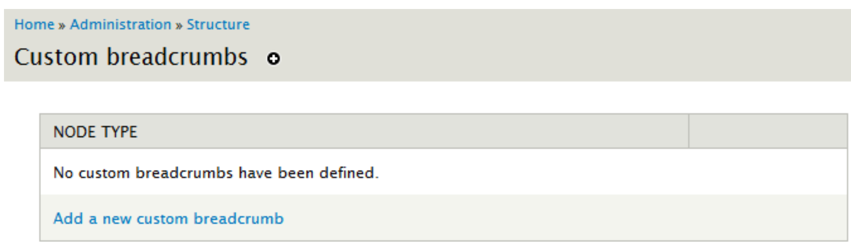
http://drupal.org/project/custom_breadcrumbs

Alter installation and activation, we begin configuration at:

Administration ⇨ Structure ⇨ Custom breadcrumbs

URL Custom breadcrumbs
/admin/structure/custom_breadcrumbs

We can choose how the breadcrumbs behave depending on node type. For each content type we can opt for a different configuration by clicking on **Add a new custom breadcrumb**. F25.5



F25.5
Custom breadcrumbs
With the Custom breadcrumbs module you can add customized breadcrumbs, independent of whether the page is linked to the site menu or not.

On the following page we can specify the following: F25.6

- **Node type.** The content type containing the nodes that the breadcrumb configuration is applied to.

- **Breadcrumb visibility.** Here we can insert PHP code (without the start and end labels *<?php ?>*) that returns (TRUE) or (FALSE). If the code returns true the configuration is applied to the breadcrumbs. If it returns false, then the configuration is not applied. This field is optional and if we leave it empty then custom breadcrumbs are always displayed. PHP code should only be used by developers with programming knowledge.

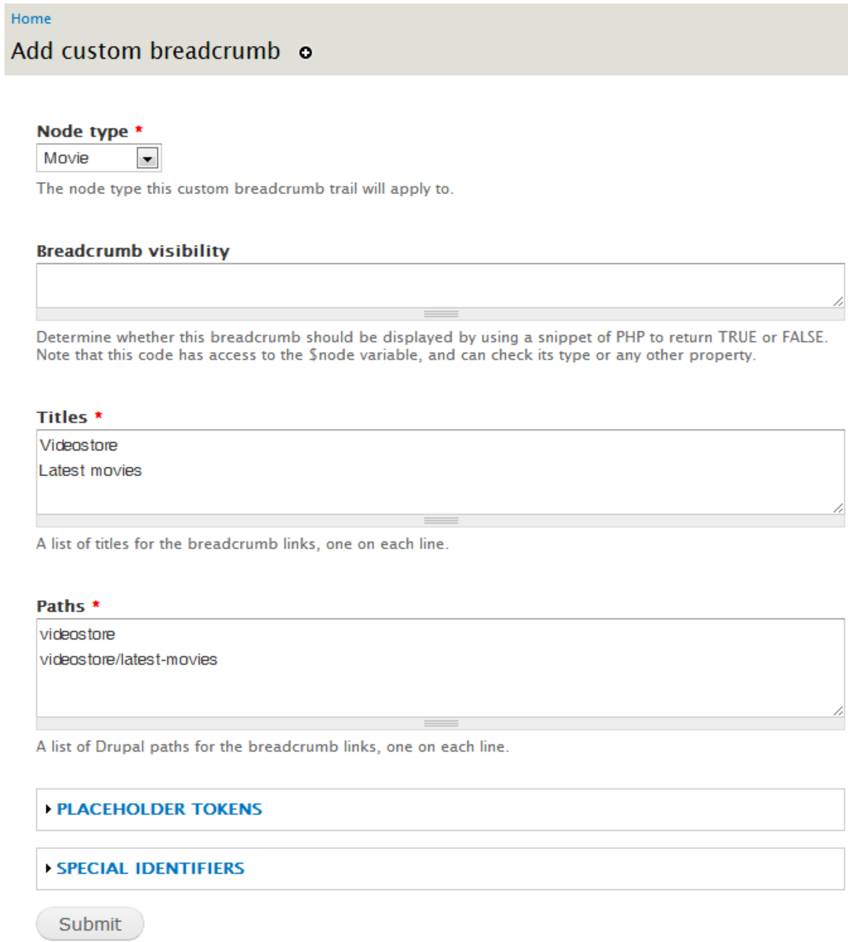
- **Titles.** The titles that appear in the breadcrumbs, as they are seen by the user. We can choose a list, with each element as a line. This displays the breadcrumbs as successive elements.

- **Paths.** For each title inserted in the previous section, we must indicate a system route. For example, we can link to the URL of a view that we created in a previous unit, such as **Latest movies**.

F25.6

Add custom breadcrumb

When adding a new custom breadcrumb (Add custom breadcrumb) chose the content type that it applies to, the visibility (Using PHP) and the paths for the breadcrumb links.

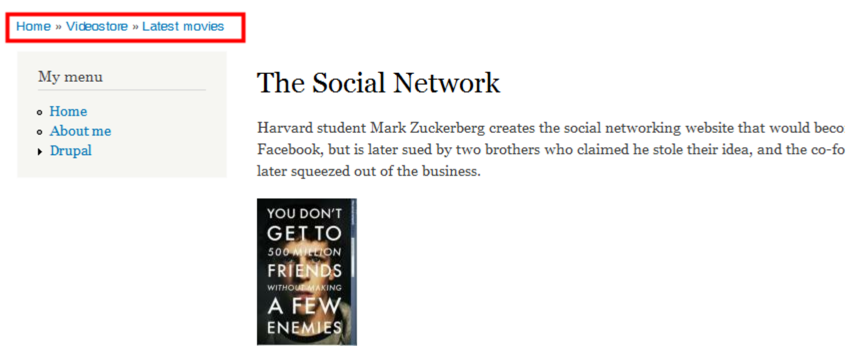


Once the configuration is saved we can check that the custom breadcrumb is working by accessing any node of the type **F25.7** specified.

F25.7

Custom breadcrumbs

The custom breadcrumbs you generate display based on the configuration specified in Figure 25.6.



Menu Link Improvements

25.3

By default, all links open up in the same navigation window in Drupal. In some instances it may be useful that a link opens up in a new window, generally when it come to external links. It may also be necessary to personalize the menu link styles, in order to differentiate them from the rest through CSS styles. In this paragraph, we will study two modules that add additional attributes to the menu links: **Menu Attributes** and **Special Menu Items**.

Menu Attributes Module

The **Menu Attributes** module allows additional attributes to be added to the elements of a menu. Internally, these attributes are added to the link tag (<a>) generated for the menu element.

The module is available at:

http://drupal.org/project/menu_attributes

This module allows additional attributes to be added to the elements of a menu. It includes the following additional attributes that will be added to the **<a>** tag of the menu element: F25.8

- **Title** (**title** attribute). Allows a description of the link to be added through the **title** attribute, which will be displayed when the cursor is placed over the link.

  <a **title="Link 1"** href="/article1">Article 1

- **Id** (**id** attribute). Allows an **id** to be added to the option for the use of CSS or jQuery (JavaScript). The **id** attribute assigns an identifier to the associated element. This should be the only identifier in the document, and may be used to refer to that element (for example, from CSS). For example:

  <a **id="link1"** href="/article1">Article 1

- **Name** (**name** attribute). Allows the value of the **name** attribute to be assigned to the menu element link. It gives a unique name to the element so that it may subsequently be located. It is used to insert page markers in a document. For XHTML compatibility, it is recommended that the **id** attribute is used in place of **name**. For example:

  <a **name="articles"**>Articles

- **Relationship** (**rel** attribute). Allows the value of the **rel** attribute to be assigned to the menu element link. The **rel** attribute describes the relationship that exists between the current page and the destination page or file. In other words, it defines the significance of the destination document in relation to the current document. For example:

  <a **rel="next"** href="/article2">Next article

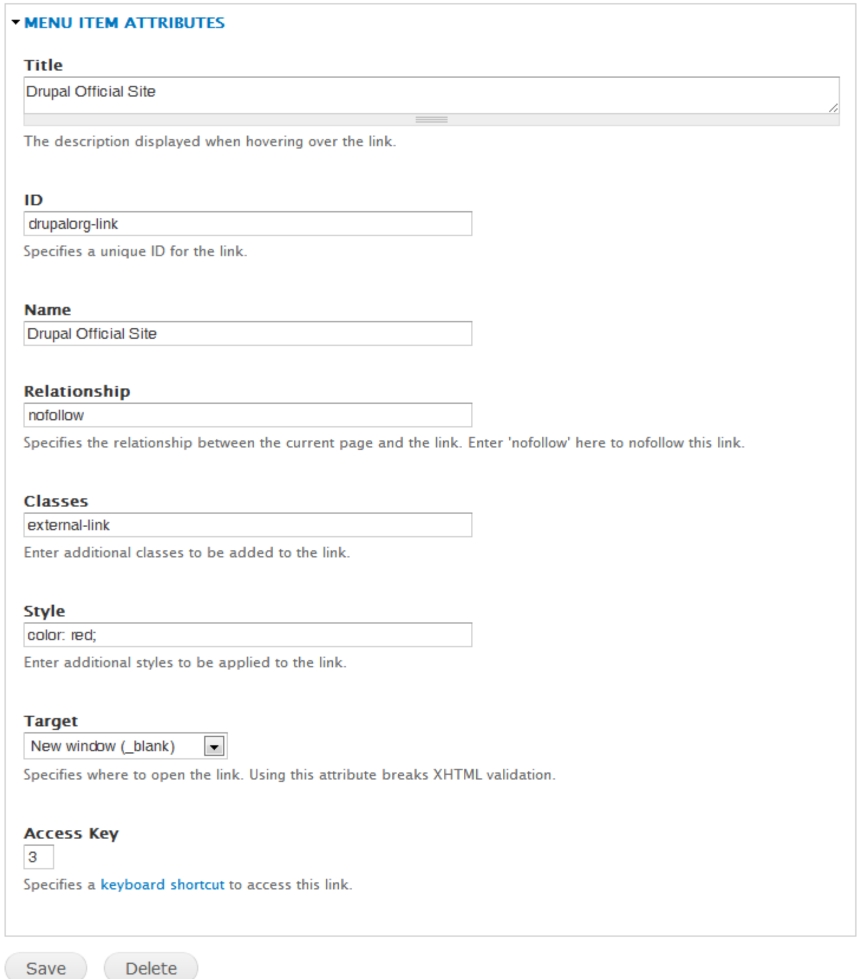
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Other typical values of this attribute are: start, next, prev, contents, index, chapter, section, subsection, help, etc.

F25.8

Menu Attributes Module

This module allows additional attributes to be added to the elements of a menu that will be applied to the <a> tag of the generated link.



- **Classes** (**class** attribute). Allows the assigning of one or several class names for the application of CSS styles, that will be included in the **class** attribute of the link. If several classes are assigned, they should be separated by spaces.

  <a **class="menu-style"** href="/article1">Article 1

- **Style** (**style** attribute). Allows the direct specification of CSS styles, making use of the **style** attribute of the link.

  <a **style="color: #0000FF; font-size: 12pt"** href="/article1">Article 1

- **Target** (**target** attribute). Adds the **target** attribute of the links, indicating the page (or frame) where the page referred to by the menu element should be displayed. It is a very useful option for linking external pages to the site while opening them in a new window. For example, the value **_blank** in the **target** attribute allows the page to open up in a anew window:

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Article 1

- **Access key** (**accesskey** attribute). Favors accessibility by adding keyboard shortcuts for the different menu elements. It establishes a relationship between the menu element and a character key, allowing the user to activate it with that key (usually when used while hitting the "alt" key simultaneously).

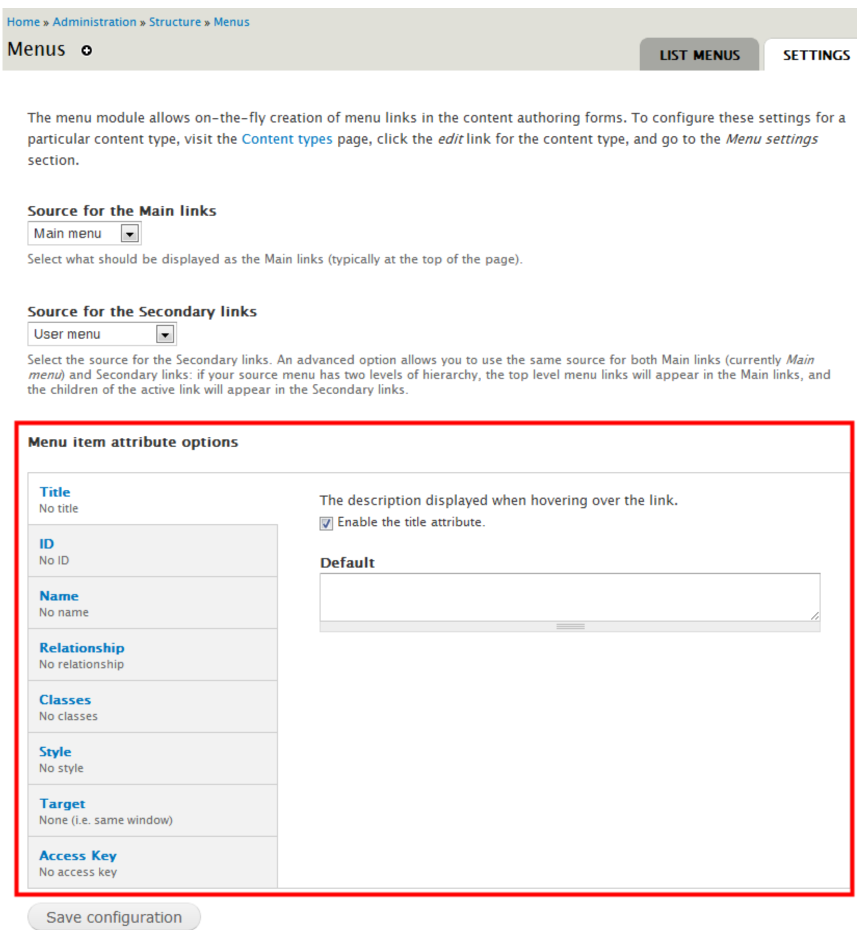
  <a **accesskey="a"** href="/article1">Article 1

The **Menu Attributes** module has a **configuration area** from where we can select what attributes will be available. We can also assign a predetermined value to each attribute.

The configuration of the Attributes Menu module can be F25.9 found in:

Administration ⇨ Structure ⇨ Menus [Settings tab]

URL Menu Attributes
/admin/structure/menu/settings



F25.9

Menu Attributes Module Settings

Configuration of the Menu Attributes module allows activation or deactivation of the options we want available when editing the menu elements.

Lastly, we show the HTML code generated for a link menu that has been assigned the values of the **Figure**: F25.8

```
<a href="http://drupal.org" title="Drupal Official Site"
id="drupalorg-link" name="Drupal Official Site" rel="nofollow"
class="external-link" style="color: red;" target="_blank"
accesskey="3">Drupal</a>
```

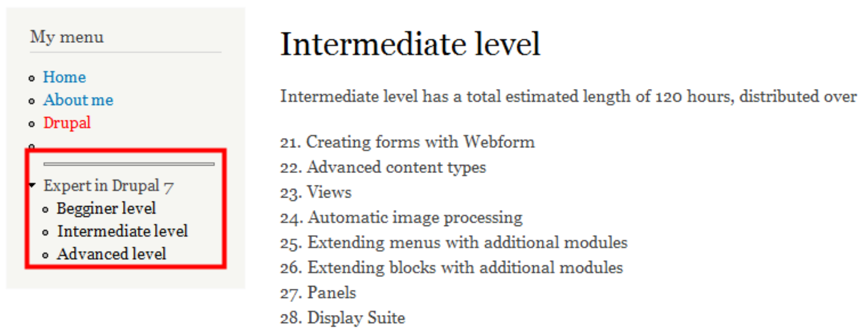
Special Menu Items Module

The **Special Menu Items** allows menu elements to be created without links, for use as separators and categories. F25.10

F25.10

Special Menu Items Module Presentation

The Special Menu Items Module allows elements to be created without a link or separators in the menu.



The module is available at:

http://drupal.org/project/special_menu_items

Once activated and installed we can access the module configuration from:

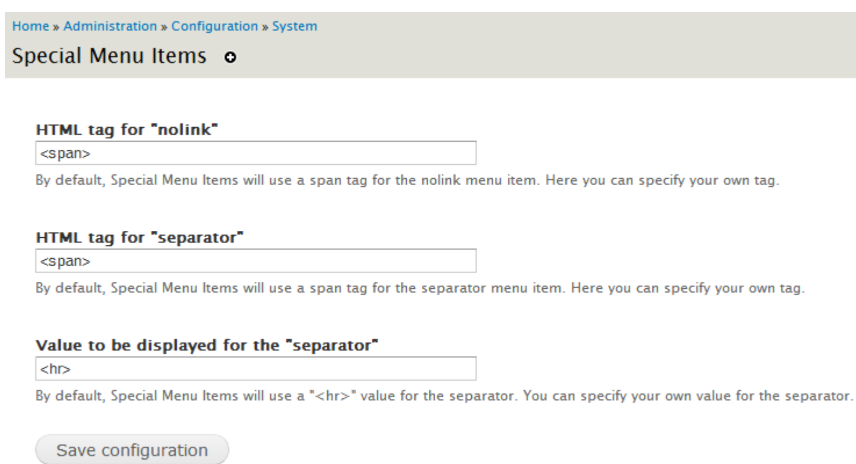
URL Special Menu Items
/admin/config/system/special_menu_items

Administration ⇨ Configuration ⇨ System ⇨ Special Menu Items

F25.11

Special Menu Items Module Settings

The Special Menu Items allows elements to be created without links <nolink> or separators <separator> in the menu.



The available options are: F25.11

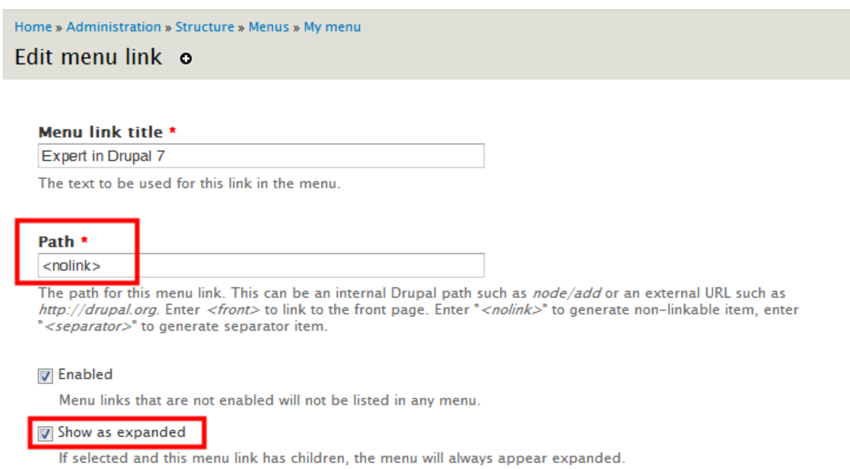
- **HTML tag for "nolink".** Assigns a different HTML tag (by default <span>) in order to define personalized CSS styles.
- **HTML tag for "separator".** Assigns a different HTML tag (by default <span>) in order to define personalized CSS styles.
- **Value to be displayed for the "separator".** By default the separator will be displayed as a horizontal line <hr>, making it possible to change it for a different value.

At first it will not be necessary to modify these values.

We have seen the configuration options of the module, but how do you use them? When creating a menu element (within any menu in the site), the following values can be shown in **Path**: F25.12

- **<nolink>.** Creates a menu element without a link.
- **<separator>.** Creates a separator.

When we create a menu element without a link (**<nolink>** or **<separator>**), it is necessary to activate the **Show as expanded** option, otherwise the menu elements nested within this element will never be displayed (children).



F25.12

Special Menu Items Module Application

When we create a menu element without a link (<nolink> and <separator>), it is necessary to activate the "expanded" option, otherwise the menu elements nested within this element will never be displayed.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

25.4 Building submenus with Menu Block

The **Menu Block** module allows menu blocks to be created from other menus, staring from any of its elements. This is how a submenu is created from a menu.

The module is available at:

http://drupal.org/project/menu_block

This module allows menu blocks to be created from other menus, staring from any of its elements. This is how a submenu is created from a menu.

The creation and management of the menu blocks is done from the block administration area, link **Add menu block**:

URL Add Menu block
/admin/structure/block/
add-menu-block

Administration ⇨ Structure ⇨ Blocks [Add menu block]

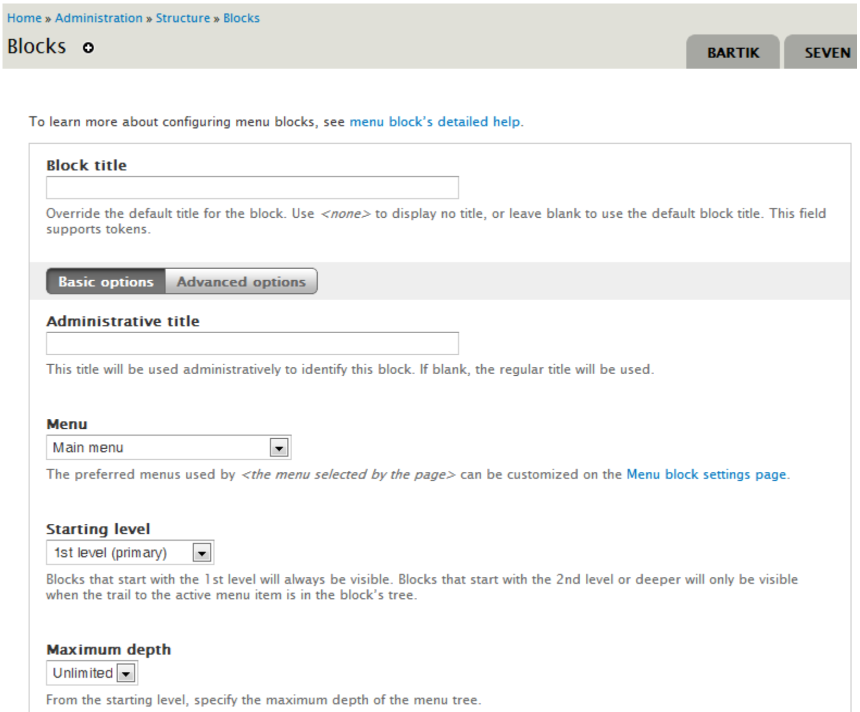
The basic options available are: F25.13

- **Administrative title**. Name of the block on the block administration list.
- **Menu.** Allows menu selection from which the menu block will be generated.
- **Starting level**. If the 1st level is selected, the block will always be displayed, otherwise it will only be displayed when it corresponds with the function of the selected level and from the page of the site that is open.
- **Maximum depth**. Number of menu levels that will be displayed in the block.

Emphasis is on the following options:

- **Expand all children**. Expands all of the menu elements.
- **Sort**. Organizes the menu element tree.
- **Fixed parent item**. Allows you to leave from any menu element. The values **Starting level** and **Maximum depth** will be considered from this principal element, and only the elements below it will be displayed.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



F25.13

Menu Block Module

Add a new menu block

The Menu Block module allows menu blocks to be created from other menus, starting from any of its elements.

Usage Example

In order to better understand the functionality of the module, lets look at an example of how to use it.

Imagine a **Videostore** menu with the following structure:

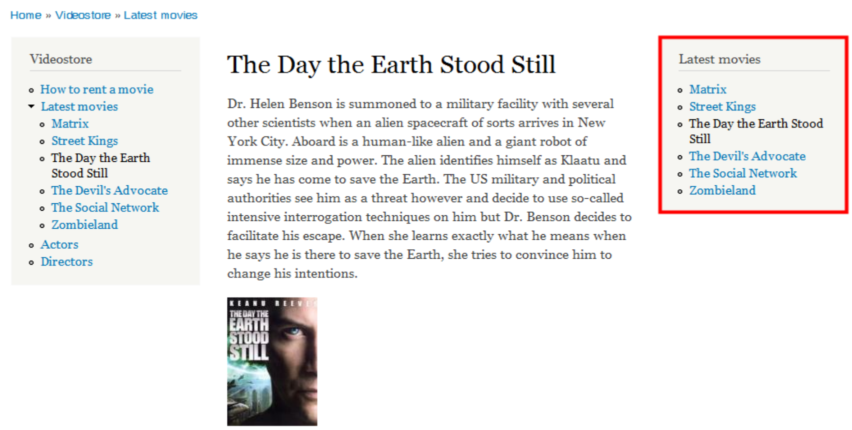
- How to rent a movie
- Latest movies
  - **Street Kings**
  - **The Devil's Advocate**
  - **Matrix**
  - **...**
- Actors
- Directors
- ...

We want to extract the **Latest movies** from this menu, so that they are displayed as an independent block when we load a movie node. The result we want to reach is shown in the **Figure**. F25.14

F25.14

Menu Block Module Example

There is an example of the Latest Movies block in the Figure, which is a sub-menu of the Videostore menu. This sub-menu will only be displayed when you navigate through one of its links.



To achieve this we must create a **menu block** (**Add menu block**), and we will select the following options: **F25.15**

- Administrative title: Latest Movies Submenu.
- Menu: Videostore.
- Starting level: 2nd level (secondary).

F25.15

Menu Block Module Example

In order to create the Latest Movies submenu, the 2nd menu level of the Videostore must be selected.

Once it's created, we will activate the **Latest Movies Submenu** block, assigning it to the "Sidebar second" region (or equivalent). With this, we will have a submenu with all the movies which will only be displayed when we navigate through the submenu content.

Menu Link Access control

25.5

In this paragraph we will study two modules that control the access to menus and their links, according to the users role.

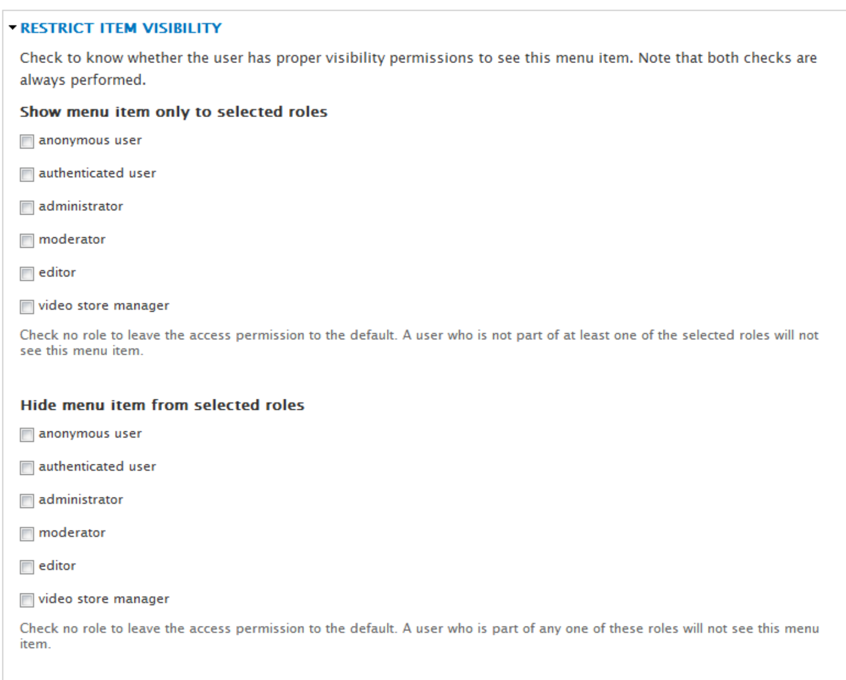
Menu Per Role Module

The **Menu per rol** module allows the menu elements to be displayed according to the function of the role of the user that visits the page. Keep in mind this module does not control access to the linked page. It will only display or hide the menu elements.

The module is available at:

http://drupal.org/project/menu_per_role

The configuration options will be displayed in the editing of each menu link, under the **Restrict Item Visibility** options group. We can select the roles that can look at the element (Show) or the roles for which they should be hidden. (Hide). F25.16



▾ RESTRICT ITEM VISIBILITY

Check to know whether the user has proper visibility permissions to see this menu item. Note that both checks are always performed.

Show menu item only to selected roles

- [ ] anonymous user
- [ ] authenticated user
- [ ] administrator
- [ ] moderator
- [ ] editor
- [ ] video store manager

Check no role to leave the access permission to the default. A user who is not part of at least one of the selected roles will not see this menu item.

Hide menu item from selected roles

- [ ] anonymous user
- [ ] authenticated user
- [ ] administrator
- [ ] moderator
- [ ] editor
- [ ] video store manager

Check no role to leave the access permission to the default. A user who is part of any one of these roles will not see this menu item.

F25.16

Menu Per Role Module

The Menu per role module allows the menu links to be displayed only to users with specific roles.

This includes the possibility of hiding elements to specific roles.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

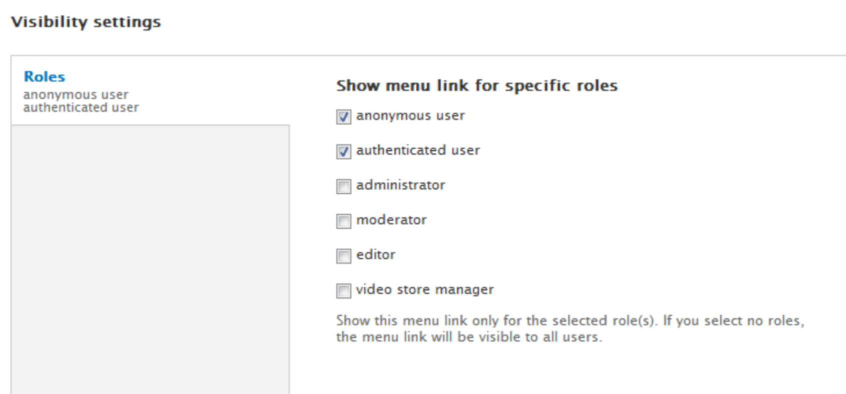
Menu Item visibility module

The Menu item visibility module is very similar to Menu per role, although it only allows selection of the roles the menu link will be displayed for.

The module is available at:

http://drupal.org/project/menu_item_visibility

The configuration of the module is done through the editing of each menu link, within **Visibility settings** (**Roles**). We simply have to select the roles for which the links will be displayed. If we do not select any roles, the link will be visible to all users. F25.17



F25.17
Menu Item Visibility Module
The Menu item visibility module allows menu elements to be displayed only to users with specific roles.

This module applies practically the same function as the Menu per role.

Both modules offer the same functionality, which is why only one should be installed on your site.

25.6 Improving Menu display

In this paragraph we will analyze modules that improve the menu presentation.

Nice menus Module

Creates dropdown menus. The module is available at:

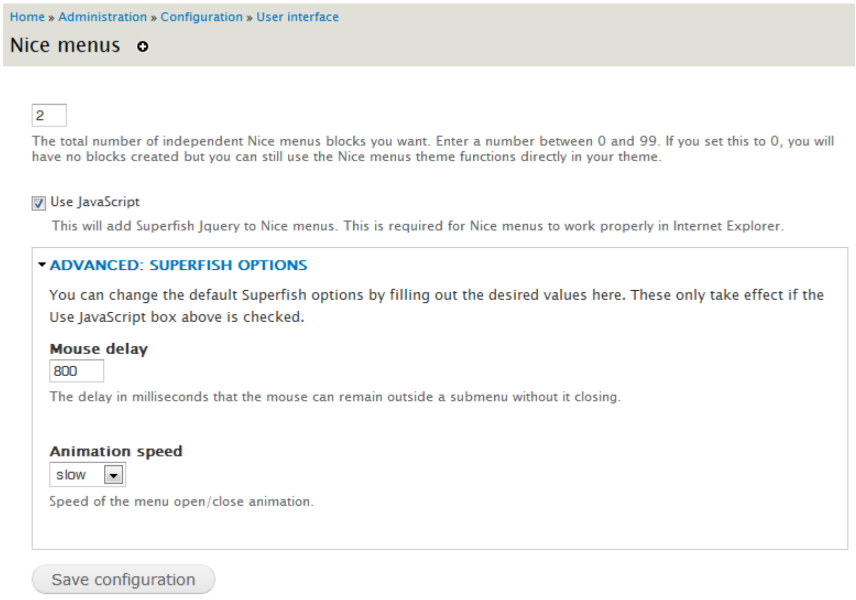
http://drupal.org/project/nice_menus

The menu will be configured from:

Administration ⇨ Configuration ⇨ User interface ⇨ Nice menus

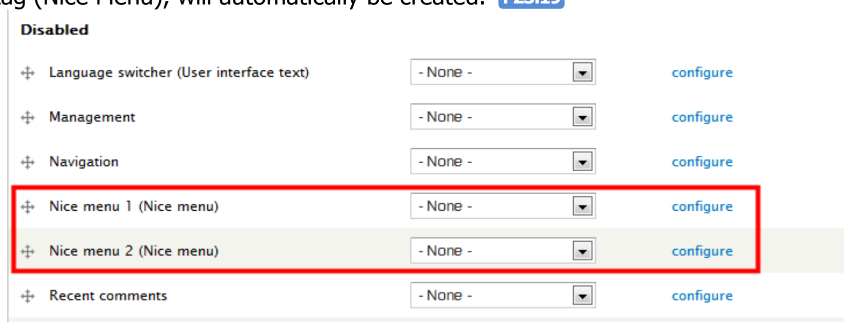
URL Nice menus
/admin/config/user-interface/nice_menus

Configuration of the module allows for the selection of the number of blocks for dropdown menus that will be created in the block administration area, with a maximum of 99. By activating the use of javascript, we can incorporate animated effects when expanding and retracting. F25.18



F25.18
Nice Menus Module Configuration
Nice Menus allows creating up to 99 dropdown menu blocks.

Once these parameters have been adjusted, the indicated number of Nice Menu blocks, that are available in the block administration area and identified with the tag (Nice Menu), will automatically be created. **F25.19**



F25.19
Nice Menus Module Added Blocks
Blocks added by Nice Menus appear directly in the block administration area identified with the "(Nice Menu)" tag after the menu name.

Accessing the configuration of each block allows modification of the menu as well selection of the menu (of those previously created in the site) that we want to create in dropdown, in addition to its direction, which can be: **F25.20**

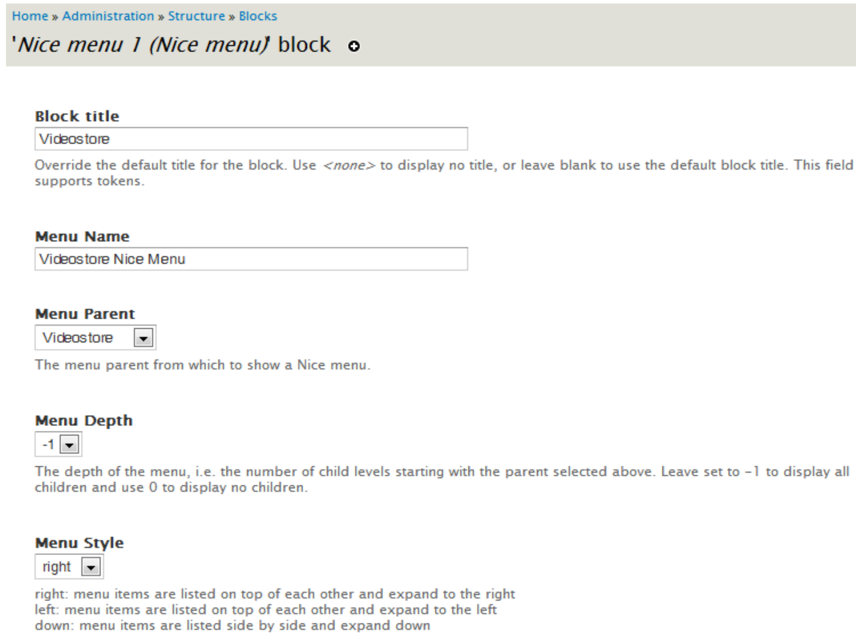
- **right.** Expands the elements of the menu to the right (for a menu placed in the left column of the site).
- **left.** Expands the elements of the menu to the left (for a menu placed in the right column of the site).
- **down.** Expands the elements of the menu down (for a horizontal menu placed, for example, in the header).

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F25.20

Nice Menus Module Configuration of the block menu

Accessing the configuration of the Nice Menu block, allows selection of the menu that we desire to convert into dropdown as well as its direction.



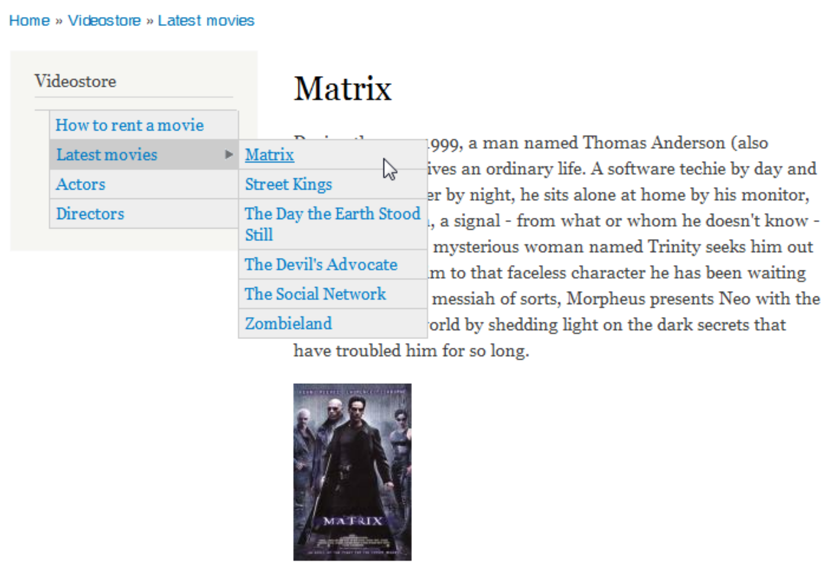
Keep in mind that once a menu such as a Nice Menu has been created and activated, the "normal" menu must be deactivated, otherwise both will be displayed.

Finally, there is an F25.21 example of a right expanding dropdown menu using the Nice Menus Module in the **Figure.**

F25.21

Nice Menus Module Dropdown menu example

Example of a dropdown menu making use of the Nice Menus module



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Menu Icons module

The **Menu Icons** module allows icons or images to be added to the menu links. The module is available at:

http://drupal.org/project/menu_icons

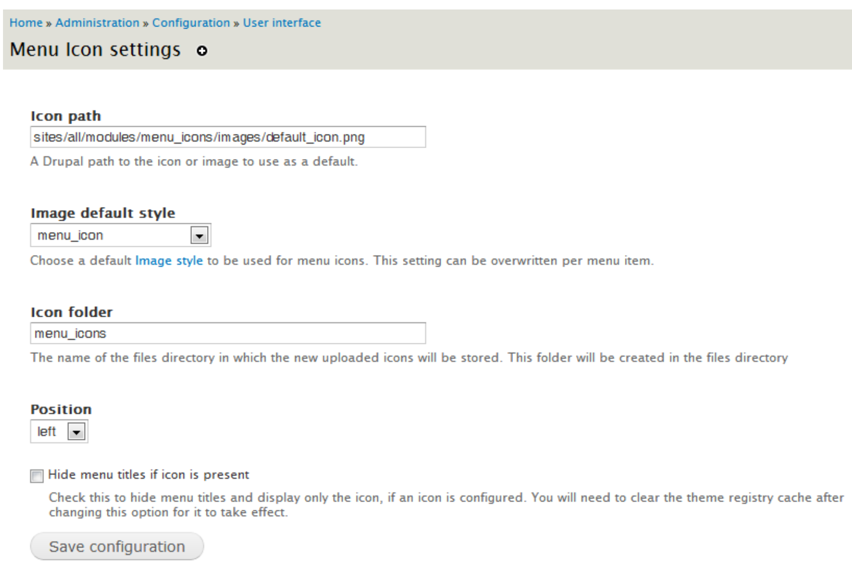
Once installed and activated, the configuration can be accessed from:

Administration ⇨ Configuration ⇨ User interface ⇨ Menu Icon Settings

URL Menu Icon Settings
/admin/config/user-interface/menu_icons

The module configuration options are: F25.22

- **Icon path**. Path to the image that will be used by default.
- **Image default style**. Image styles can be applied to the icons associated to the links. A default image style can be selected from this option.
- **Icon folder**. File where uploaded images will be stored.
- **Position**. The position of the image can be to the left or right of the menu link.



F25.22
Menu Icons Module Configuration
The Menu Icons module allows images or icons to be added to the link menu.

Image Styles can be applied to the uploaded images.

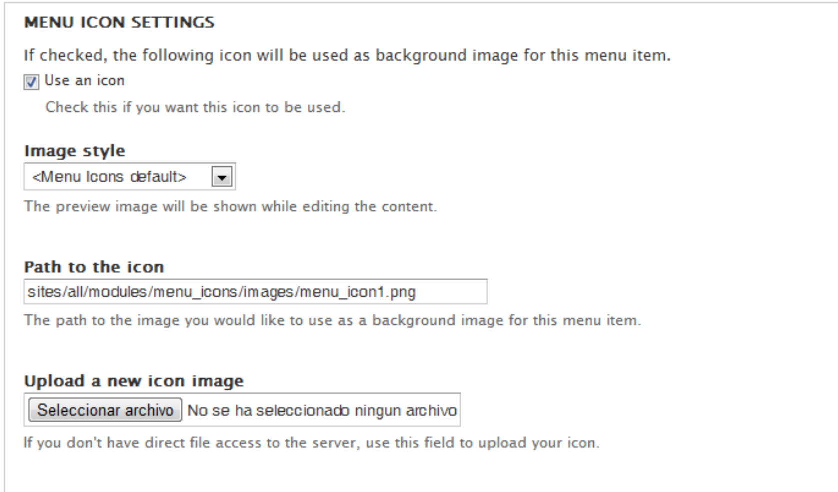
In editing each menu link, the **Menu icon settings** options can be found, allowing configuration of the icon for that particular element: F25.23

- **Use an icon**. Activating this option enables an icon for the menu link.
- **Image style**. Allows selection of an image style that can be different from the one defined in the general configuration of the module.
- **Path to the icon**. It will not be necessary to select a direct path if an image is uploaded next.
- **Upload a new icon image**. Allows upload of an image that can be used as an icon.

F25.23

Menu Icons Module Icon in menu link

Configuration of the menu link. Once specified that we want to use an icon in this menu link, the image may be uploaded and the image style to be applied may be selected.

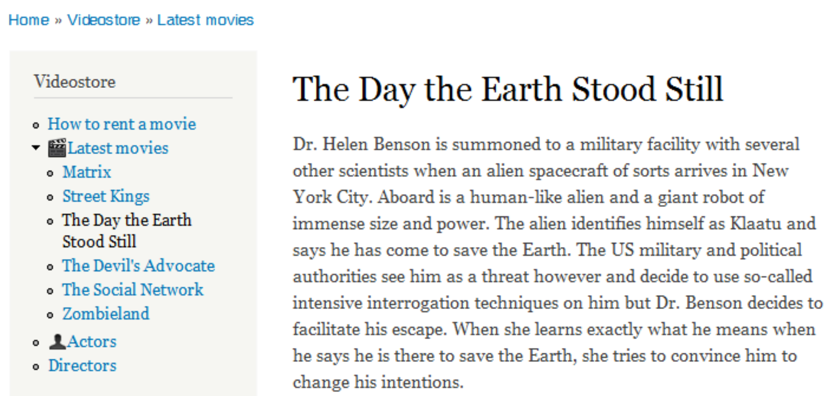


In the links where this option has been activated, the corresponding icon will be displayed. **F25.24**

F25.24

Menu Icons Module Example of a menu with icons

Example with a menu with icons. As seen in the Figure, the icons may also be applied to the site breadcrumbs.



Administration Menu Module

Although Drupal 7 offers a superior administration menu, the **Administration Menu** provides a more complete dropdown menu, with all the administration options, sub-options, and tabs.

The module is available at:

http://drupal.org/project/admin_menu

Once installed and activated, the configuration can be accessed from:

URL Administration menu
/admin/config/administration/admin_menu

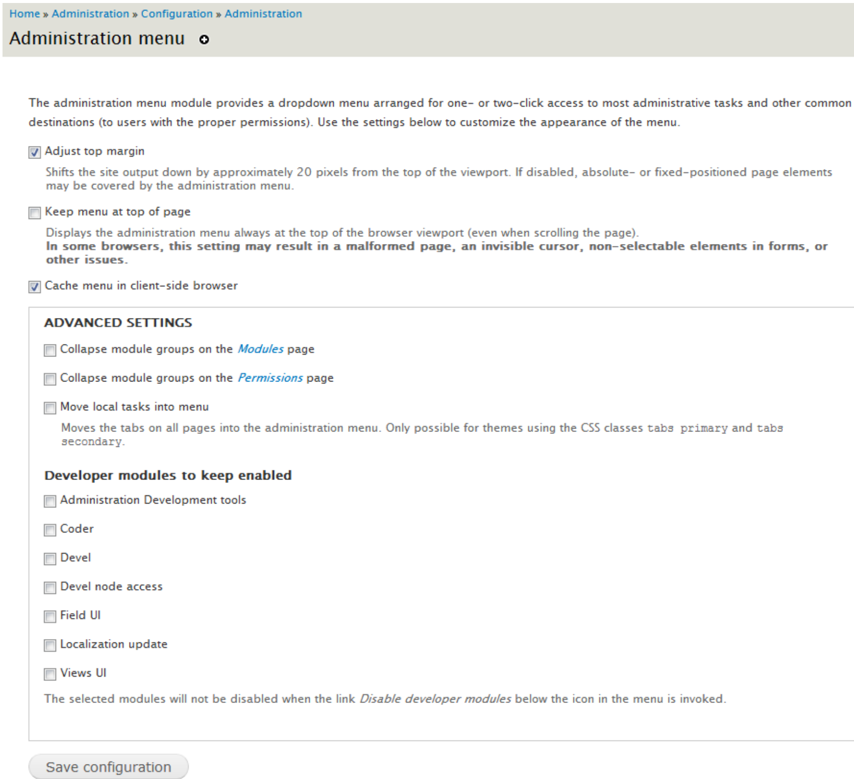
Administration ⇨ Configuration ⇨ Administration ⇨ Administration menu

Two of the basic administration configuration menu options are: **F25.25**

- **Adjust top margin**. Adds a width of 20px so that the administration menu does not overlap with the site.

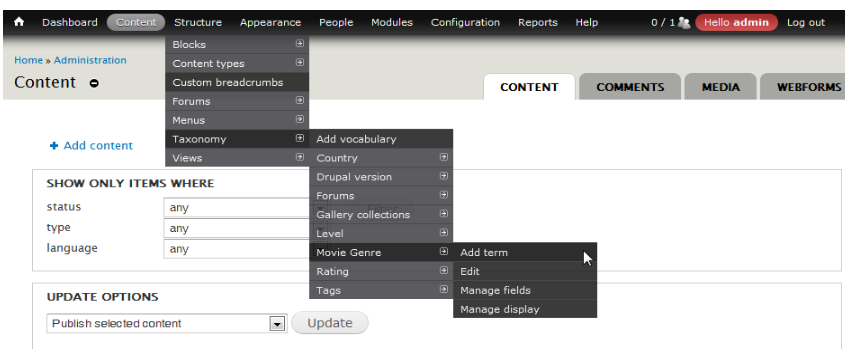
- **Keep menu at top of page**. Maintains the menu visible every time, even when scrolling down the page.

To deactivate the original administration menu, the **Toolbar** menu has to be deactivated from the Drupal core.



F25.25
Administration Menu Module Settings
Displays the administration menu in the page header in the form of a dropdown menu.

Figure F25.26 shows the administration menu.



F25.26
Administration Menu Module Settings
The Toolbar module must be deactivated from the Drupal core so that the administration menu will not be displayed by default.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

26 Extending blocks with additional modules

In the Beginner Level we studied the basics of creating and configuring block visibility. In this unit we expand our knowledge of the core block functionalities using the following additional modules:

- **Block Title Link**, changes block titles into links.
- **Block Subtitle**, adds a subtitle field to blocks.
- **Image Block**, creates image blocks.
- **Block More Link**, allows us to add an extra "see more" link to blocks.
- **Block Class**, allows us to add CSS classes to blocks. We can then apply different styles to those that are in the theme style pages.
- **Block tab**, groups all blocks from a region and displays them as tabs.
- **Block Visibility by Term**, controls block visibility via the taxonomy terms of the displayed content.
- **Block Refresh**, allows us to configure a block to update at regular, defined intervals, using jQuery/AJAX.
- **Multiblock**, to create multiple copies of a block. We can then distribute these blocks in different regions and change their individual settings.
- **Block reference**, adds a block reference field to entities.

Comparative D7/D6

Most of the modules in this unit have a corresponding version in Drupal 6 and work in a very similar way.

Some of the modules we study are not yet available as stable versions. They may produce unwanted effects or be affected by other installed modules.

Unit contents

26.1 Additional elements in blocks....................................170
26.2 Improving block display...176
26.3 Access control and block visibility.............................179
26.4 Other modules..181

26

26.1 Additional elements in blocks

We start this unit by looking at some of the modules that add additional elements to blocks.

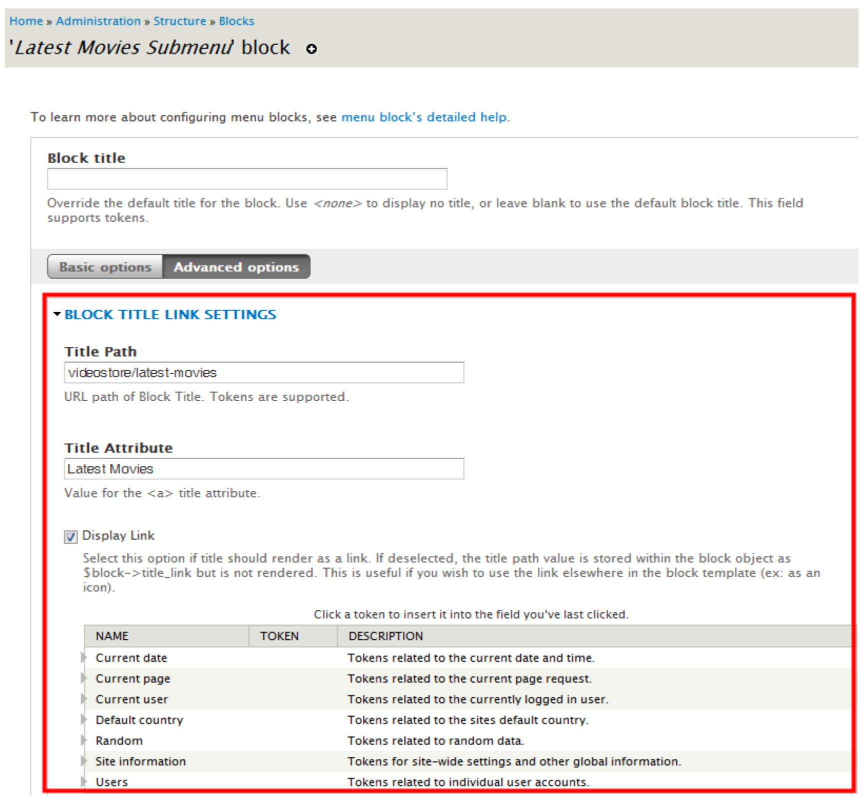
Block Title Link module

The **Block Title Link** module converts the block title into a link. The module is available at:

http://drupal.org/project/block_titlelink

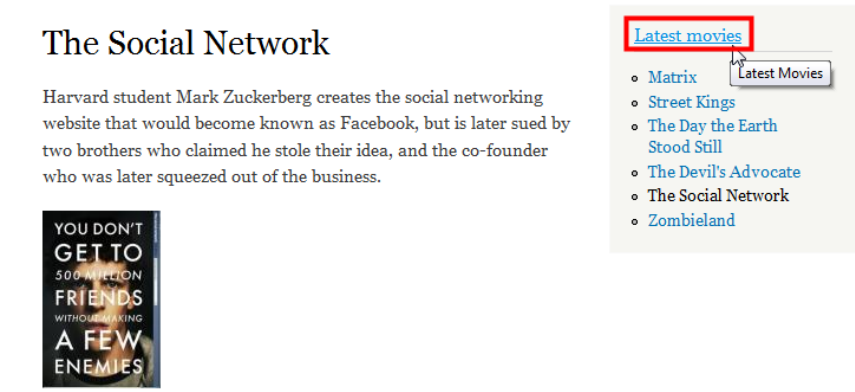
The settings are within a block of options called **Block title link settings** that appears when we edit a block. We can choose the link path and the content of the **title** attribute associated with the <a> link label.
The link text is **Block title.** F26.1

F26.1
Block Title Link module. Settings
The **Block Title Link** module turns a title into a link.



In the **Figure** F26.2 we see a **Latest Movies** block with the title converted into a link to the **videostore/latest-movies** view.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



F26.2

Block Title Link module. Display

Example of how the **Block Title Link** module works. The block title is now a link.

Block Subtitle module

The **Block Subtitle** module adds a subtitle field to blocks. The module is available at:

http://drupal.org/project/block_subtitle

Once installed and activated, we must make a small modification to the theme template we are using (in the file **block.tpl.php**):

```
<?php if ($block->subtitle): ?>
  <h3 class="subtitle"><?php print $block->subtitle; ?></h3>
<?php endif; ?>
```

Although we will study theme file structures later on, the installation of this block acts as an initial introduction to template modification.

Before moving on we must consider this important warning: **The core themes must never be modified**, as any changes are lost when Drupal is updated. Extra themes added to the site (in /sites/default/themes or /sites/all/themes) can be modified and, as we shall see, often need to be modified to adapt their design: We can modify their templates (archivos .tpl.php) and the style sheets (archivos .css).

As we have to modify the default theme of the site, the first thing to check is whether it is a core theme or not. If it is a core theme we download it (/themes) and upload it to the additional themes file (/sites/all/themes). Even though the theme is now duplicated the site starts to use the theme located in the **sites** file. We therefore avoid modifying the original core theme and make any changes to the themes located in the **sites** file.

After this, we locate the theme template file, **block.tpl.php**. It is normally in the root theme directory. Some themes, such as **Bartik**, have a **templates** folder containing all the available theme templates.

The theme file **block.tpl.php** is not a mandatory theme file so we may not find it in the theme's file. When this occurs we locate the generic theme file located in the **/modules/block** file, and copy it (download and upload) to the theme file that we are modifying.

We add the code above after the title and before the content. As an example we show the code added in the generic theme file provided by the **block** module:

```
<?php

/**
 * @file
 * Default theme implementation to display a block.
 * ...
 */
?>
<div id="<?php print $block_html_id; ?>" class="<?php print
$classes; ?>"<?php print $attributes; ?>>

<?php print render($title_prefix); ?>
<?php if ($block->subject): ?>
  <h2<?php print $title_attributes; ?>><?php print $block->subject
?></h2>
<?php endif;?>
  <?php print render($title_suffix); ?>

<?php if ($block->subtitle): ?>
  <h3 class="subtitle"><?php print $block->subtitle; ?></h3>
<?php endif; ?>

  <div class="content"<?php print $content_attributes; ?>>
    <?php print $content ?>
  </div>
</div>
```

Once this change is made we empty the site cache from:

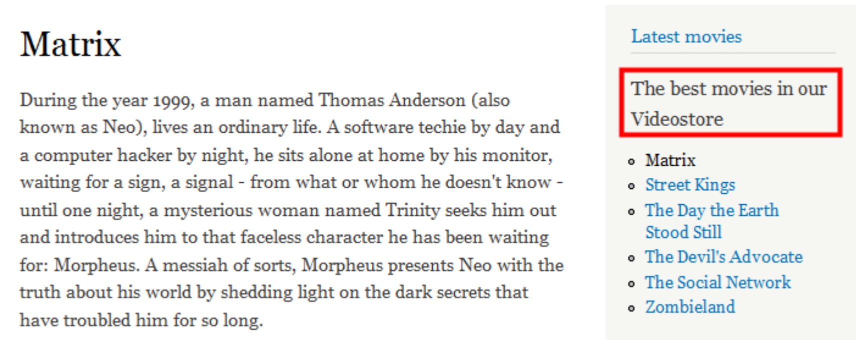
URL Performance
/admin/config/development/performance

Administration ⇨ Configuration ⇨ Development ⇨ Performance

Click on **Clear all caches**. After these changes we can add subtitles to site blocks from each block's settings. F26.3

F26.3
Block Subtitle settings. Settings
The **Block Subtitule** module adds a display field to blocks.

In the **Figure** F26.4 we see an example block with a subtitle.



F26.4
Block Subtitle module. Display
Example of a block with a subtitle, added using the Block Subtitle module.

Image Block module

The **Image Block** module allows us to create blocks containing images. We can also apply any image style previously created in the site.

The module is available at:

http://drupal.org/project/imageblock

Once installed and activated, we create new blocks with images from block administration. F26.5

F26.5
Image Block module. Configuration
The **Image Block** module allows you to create blocks containing an image.

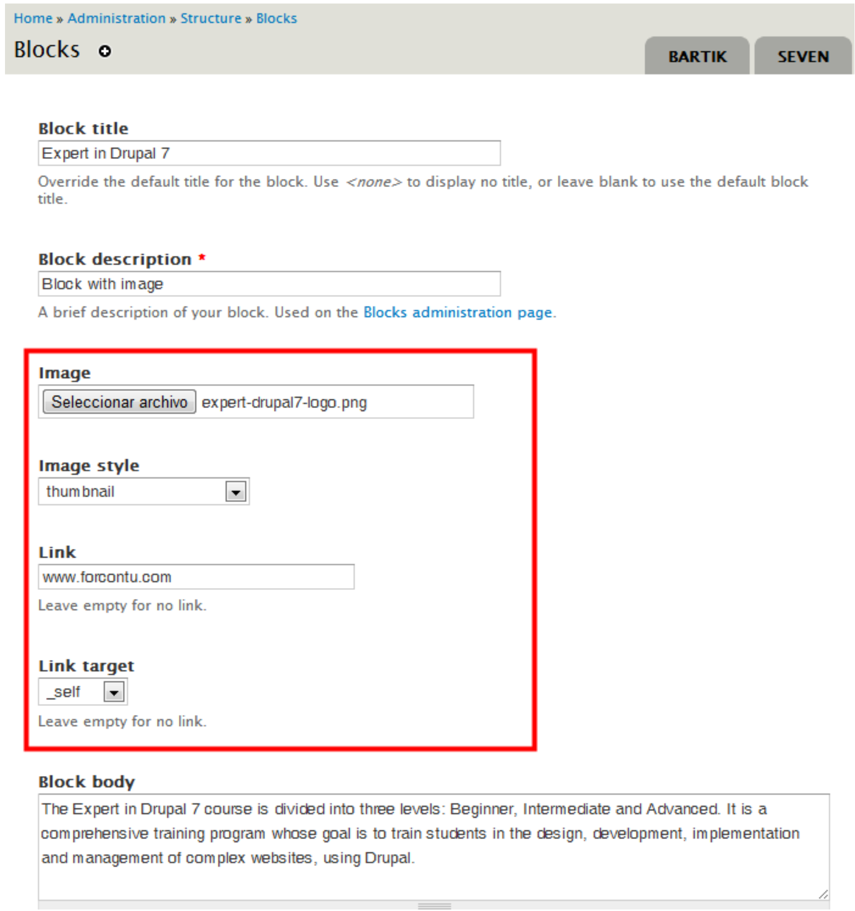
Image blocks have, as well as the standard fields for all blocks, the following extra fields: F26.6

- **Image**. Allows us to upload an image.
- **Image style**. Selects an image style that is applied to transform the uploaded image.
- **Link**. If we specify a path the image becomes a link.
- **Link target**. We choose where the page opens (for example, _self to open it in the same window and _blank to open a new window).

F26.6

Image Block module. Settings

Images in blocks created with **Image Block** have an associated image style and, optionally, an associated link.

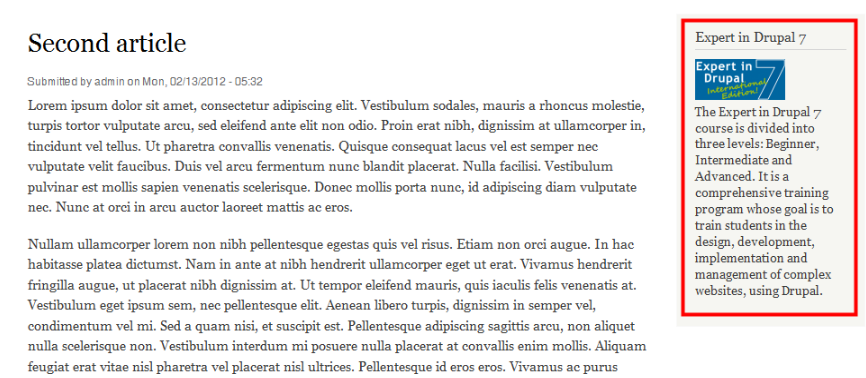


In the **Figure** **F26.7** we see an example image block.

F26.7

Image Block module. Display

Example of an image block created with **Image Block**.



Block More Link module

The **Block More Link** module allows us to add an extra link, typically a "see more" link, to blocks.

The module is available at:

http://drupal.org/project/block_morelink

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

As with **Block Subtitle**, we must add a small segment of code to the template file, **block.tpl.php**:

```
<?php if ($block->more): ?>
  <?php print $block->more ?>
<?php endif;?>
```

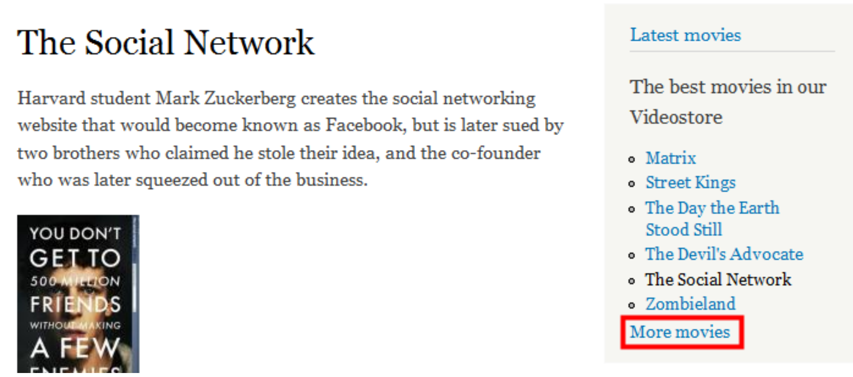
For example, in the generic block.tpl.php:

```
<?php

/**
 * @file
 * Default theme implementation to display a block.
 * ...
 */
?>
<div id="<?php print $block_html_id; ?>" class="<?php print
$classes; ?>"<?php print $attributes; ?>>

<?php print render($title_prefix); ?>
<?php if ($block->subject): ?>
  <h2<?php print $title_attributes; ?>><?php print $block->subject
?></h2>
<?php endif;?>
  <?php print render($title_suffix); ?>

  <div class="content"<?php print $content_attributes; ?>>
    <?php print $content ?>
  </div>

  <?php if ($block->more): ?>
    <?php print $block->more ?>
  <?php endif;?>

</div>
```

Once the code is added to the template and the caches are emptied, we can add an extra link to a block from its settings page. F26.8

F26.8
Block More Link module. Settings
The **Block More Link** module allows you to add an extra link to a block, typically a "see more" link.

In the **Figure** F26.9 we see an example of an extra link added to a block.



F26.9

Block More Link module. Display

An extra "More movies" link, created with the Block More Link module.

26.2 Improving block display

We continue introducing additional modules, this time related to block display.

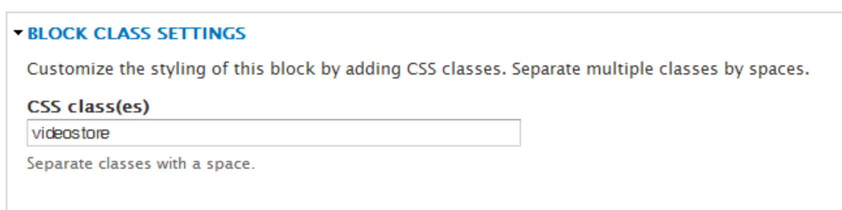
Block Class module

The **Block Class** module adds CSS classes to blocks, allowing us to apply different styles to theme style sheets.

The Block Class module is available at:

http://drupal.org/project/block_class

Once installed and activated, we can add CSS classes from block settings (**Block class settings**). F26.10 To add one or more classes, we separate them with a space.



F26.10

Block Class module

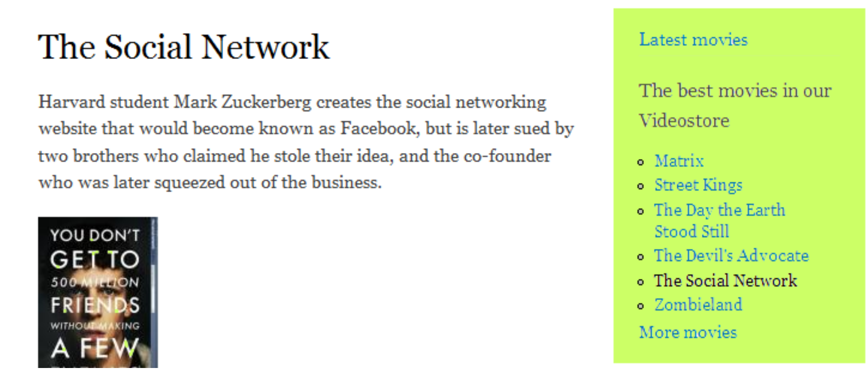
With **Block Class** you can add CSS classes to blocks.

The CSS classes must be correctly defined in the style.css file (the name of the main style file varies depending on the selected theme). In **Unit 29** we learn more about the use of CSS styles.

For example, we can add a class called **videostore** to modify the background colour of the **Latest Movies** block: F26.11

```
.videostore {
  background-color: #ccff66;
}
```


Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



F26.11

Block Class module
Example of a block with a modified background colour created by adding a class with **Block Class**.

As well as adding the class you must also modify the corresponding style sheets.

Block tab module

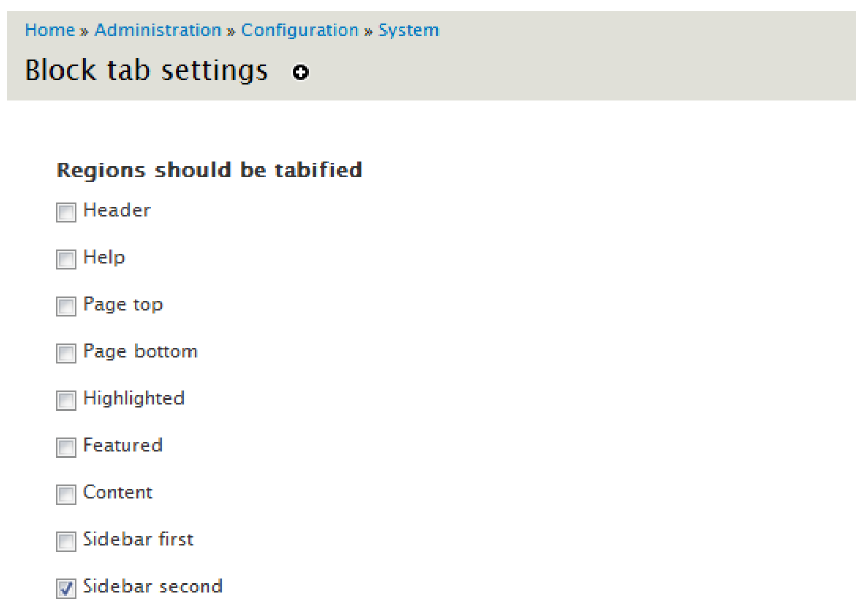
The **Block tab** module displays all the blocks assigned to a region as tabs. The module is available at:

http://drupal.org/project/block_tab

Once installed and activated, we indicate which regions are converted to tabs from: F26.12

Administration ⇨ Configuration ⇨ System ⇨ Block tab settings

URL Block tab settings
/admin/config/system/block_tab



F26.12

Block tab module. Settings
With **Block tab** you can transform blocks in the same region into tabs.

From settings you select the regions to be converted into tabs.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

In **Figure** F26.13 we see an example of blocks converted into tabs.

F26.13
Block tab module. Display
Example of the use of **Block tab**. Blocks in the right hand column have been grouped in tabs.

Note
Some of the modules we have looked at may produce unexpected results when used together.

Access control and block visibility

26.3

In the Beginner Level we studied block **Visibility settings**. These options establish the visibility of blocks on the site, depending on the user, the page and the settings. Here we review the options covered previously:

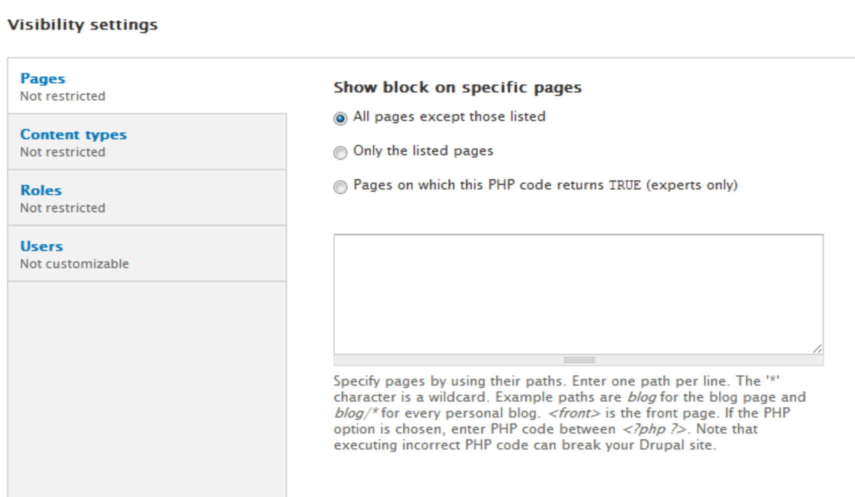
- **Pages**. Allows us to display a block on specific pages. We must choose between the following options: F26.14

    - Display on all pages except those listed on this page.
    - Display only the listed pages
    - Display only pages on which this PHP code returns TRUE (PHP mode, for experts only).

    For the first two options we must indicate the names of the pages in the **Pages** text box (one page per line). Also, we can use the asterisk ***** to indicate any character. For example, we can use the following page names:

    - **node/1**. Refers to page **node/1**.
    - **<front>**. Refers to the start page or site homepage.
    - **blog***. Refers to all pages that start with blog, even if they contain subsequent characters.
    - **about-me**. Refers to the page with the URL alias **about-me**.

    If we don't want to choose a specific page, we select the first option and leave the **Pages** field empty. This means that the block displays all pages.



F26.14
Visibility by page
You can display a block for a specific set of pages, and choose exceptions.

Expert users can use PHP code to create advanced conditions for block visibility.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Content types.** Allows us to display blocks when the user is viewing the selected content types. The site content types are shown here. F26.15

F26.15

Visibility by content type

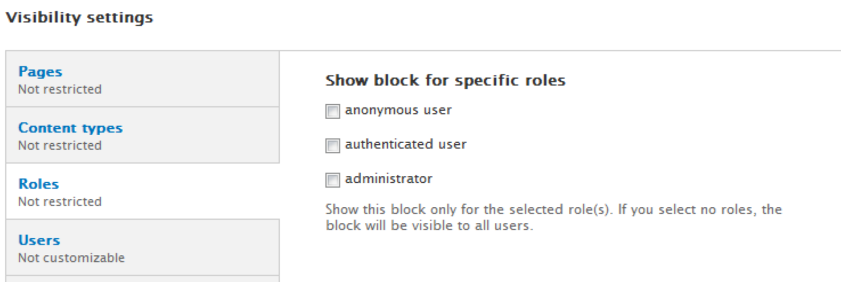
You choose specific content types and the block only appears when nodes of that type are displayed.

- **Roles.** Allows us to show the block to specific roles. If no role is selected the block is visible to all users. F26.16

F26.16

Visibility by role

Shows the block only to the selected roles.

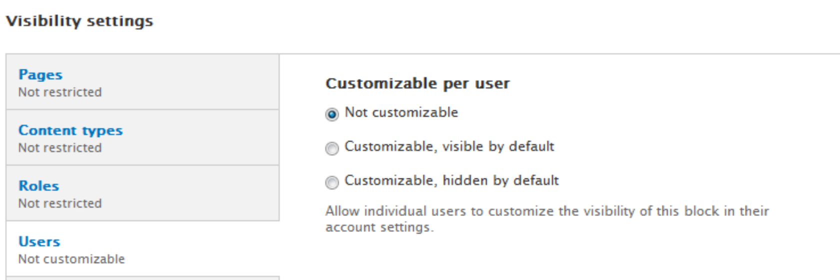


- **Users.** We must choose one of the following options: F26.17
  - **Not customizable.** The users cannot control the visibility of this block.
  - **Customizable, visible by default.** Displays the block by default, but allows the user to hide it.
  - **Customizable, hidden by default.** Hides the block by default, but allows the user to display it.

F26.17

Customizable per user

Allows the user to decide whether to see a block or not.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

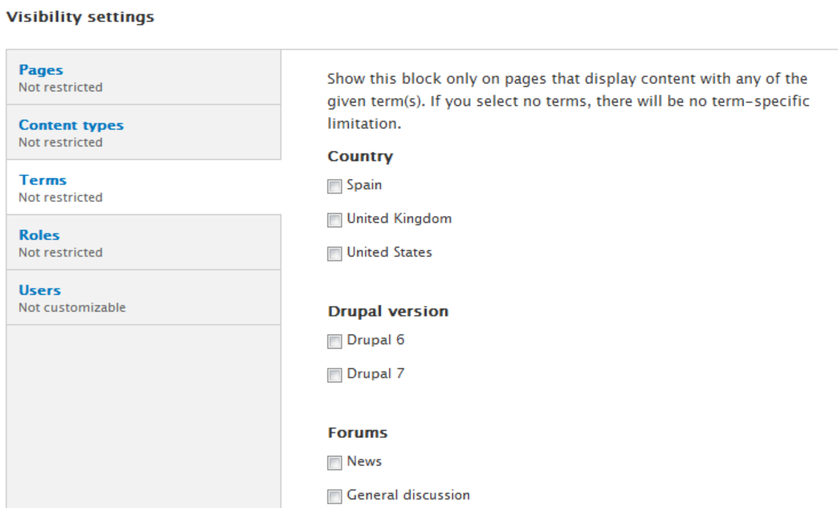
Block Visibility by Term module

The **Block Visibility by Term** module adds a new group of visibility options, **Terms**, which allow us to select taxonomy terms that render the block visible.

The module is available at:

http://drupal.org/project/block_term

In the **Figure** F26.18 we see the Visibility settings for taxonomy terms. The block only displays when the loaded page contains one or more of the selected taxonomy terms. If no terms are selected then taxonomy terms are not used to govern the visibility of the block.



F26.18
Block Visibility by Term module
The Block Visibility by Term module allows you to control block visibility using the taxonomy terms visible in the displayed node.

Other modules 26.4

In this last section we look at other general-purpose modules relating to blocks.

Block Refresh module

The Block Refresh module allows us to configure a block to refresh automatically after a defined period has passed. Because the block refreshes using jQuery/AJAX, the whole page does not refresh.

The module is available at:

http://drupal.org/project/block_refresh

We can activate automatic block refresh from its settings (Block refresh settings), and choose the period between each refresh (in seconds). We can also activate a button allowing the user to refresh the block's contents.

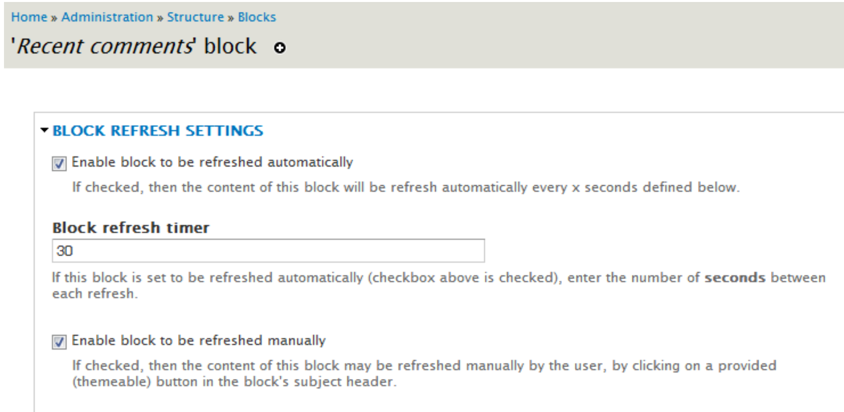
This functionality is only useful for blocks with content that changes regularly and over short periods. For example, the **recent comments** block on a website that receives multiple comments. F26.19

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F26.19

Block Refresh module

Allows you to configure a block so that it refreshes automatically at regular intervals. Blocks are refreshed with jQuery/AJAX so that only the block, and not the whole page, is refreshed.



Multiblock module

By default each block can only be assigned to one site region and with a single configuration. The Multiblock module allows us to create multiple versions, or instances, of a block with different settings, and to assign these "copies" to different regions.

The Multiblock module is available at:

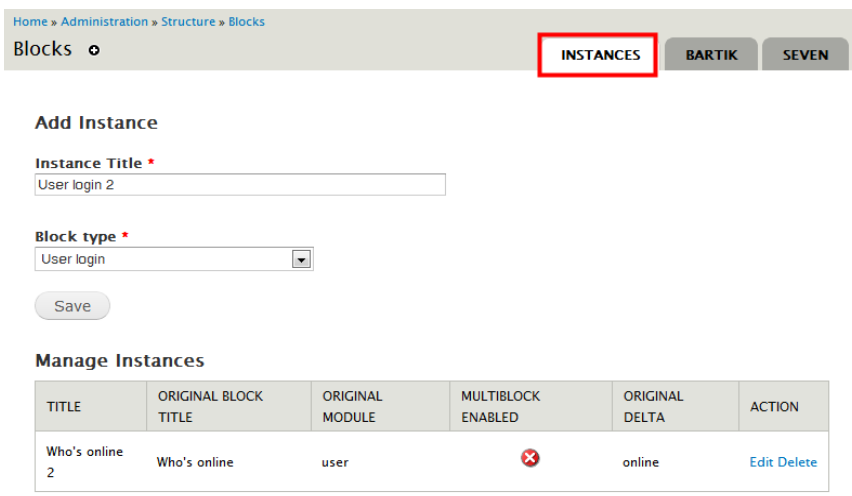
http://drupal.org/project/multiblock

To create versions of blocks we access the **Instances** tab in Block administration. All we need to do is indicate a title for the instance. This becomes the title of the new block and of the block we want to clone. In the same tab we see all created instances and whether they have been activated. Instances can be treated as independent blocks and we can assign them to different regions and give them different visibility controls. **F26.20**

F26.20

Multiblock module

The Multiblock module allows you to create multiple instances of a block, configure their settings individually, and to assign the "copies" to different regions.



Block reference module

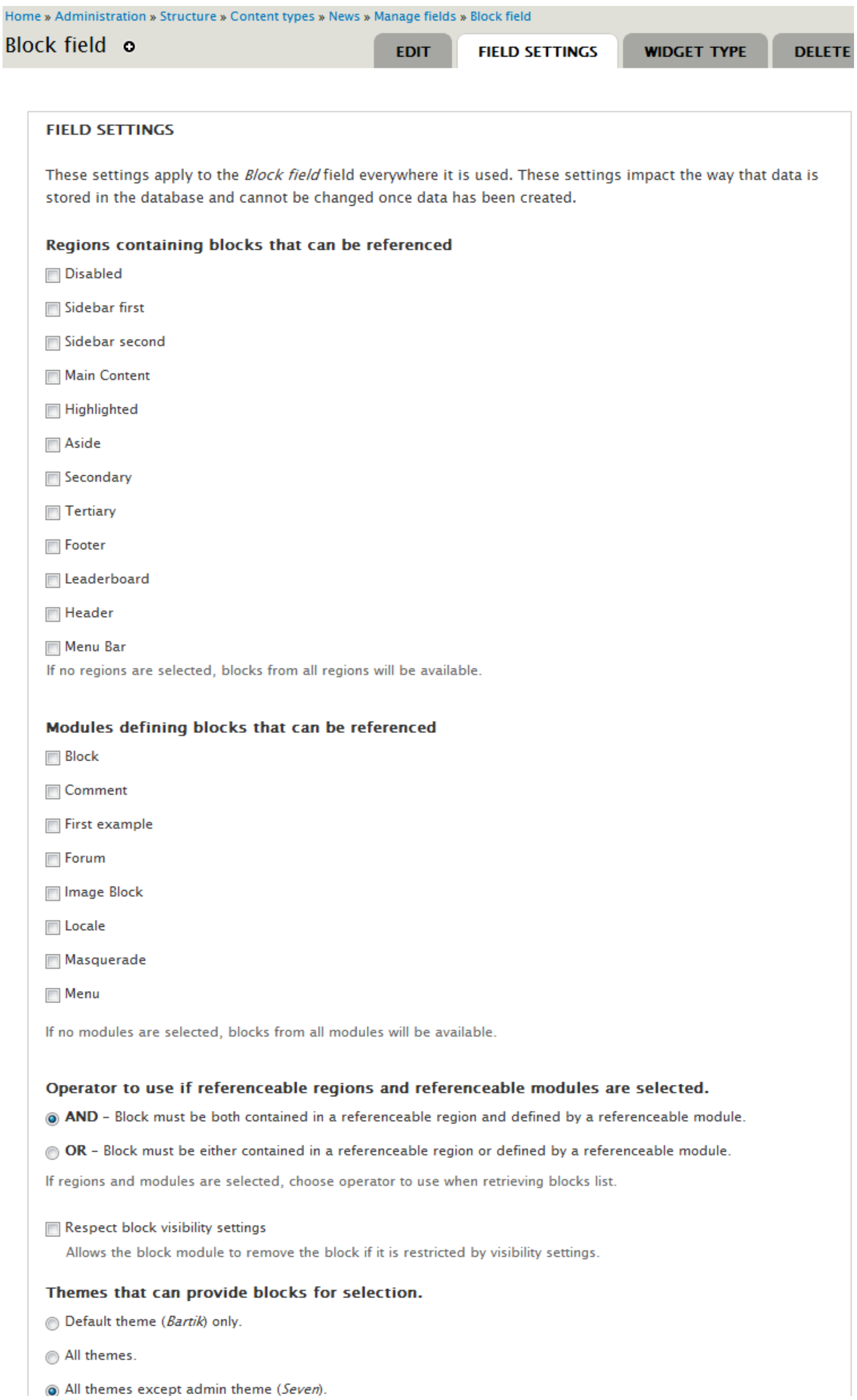
The **Block reference** module adds a block reference field that we can assign to any entity (for example, to content types). This field displays the contents of the block.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

The module is available at:

http://drupal.org/project/blockreference

Once we add the **Block reference** field type, we control the block types that appear in the node by indicating the regions where each one can be active, and specify the modules that can generate blocks. If we don't specify any blocks, all available blocks are displayed. F26.21



F26.21

Block reference module

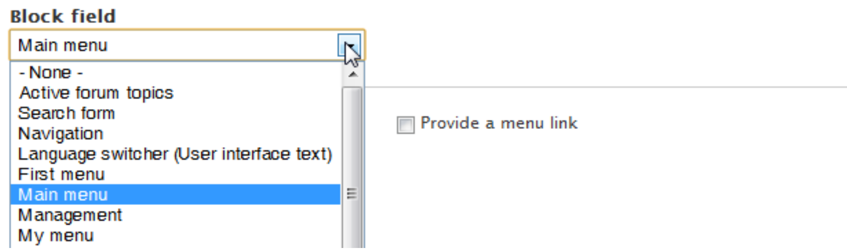
The **Block reference** module adds a block reference field that you can assign to any entity (for example, to content types).

When creating a node with a block reference field, we choose one of the available blocks. F26.22

F26.22

Block reference module

Block type selection in content types where a Block reference field has been added.



27 Panels

Panels is one of the most flexible and powerful modules available in Drupal. It allows you to build pages using various blocks or regions, called panels, where we can display different site content (nodes, views, blocks, menus, etc.). This allows us to create pages or blocks using different elements.

Panels is the ideal module for creating the home page for a site, or the landing pages for individual sections on the site.

With Panels, you can manage a variety of templates with different layouts, making it easy to create pages that display content in different ways.

Panels can also work together with the Views module, giving it added functionality.

This unit includes four case studies, which explain how to use the different types of panels.

To conclude this unit, we look at two additional modules that complement Panels:

- **Panels Extra Layouts Module**, which adds new layouts to the default ones included in the module.
- **Panels Breadcrumbs Module**, which allows us to assign custom breadcrumbs for each panel.

Comparative D7/D6

Panels

Panels operates in a similar way in both Drupal 7 and Drupal 6. But keep in mind the addition of new elements in Drupal 7, such as entities and new field types, and other elements applicable to Views 3.

Unit contents

27.1 Introduction to the Panels module 186
27.2 Panel pages 188
27.3 Managing panels 212
27.4 Mini panels 218
27.5 Panel nodes 227
27.6 Integrating Views into Panels 230
27.7 Additional modules 238

27

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

27.1 Introduction to the Panels module

With the Panels module, you can build layout designs using "subregions," which can appear as a page on the site or appear in any region, in the same manner as blocks.

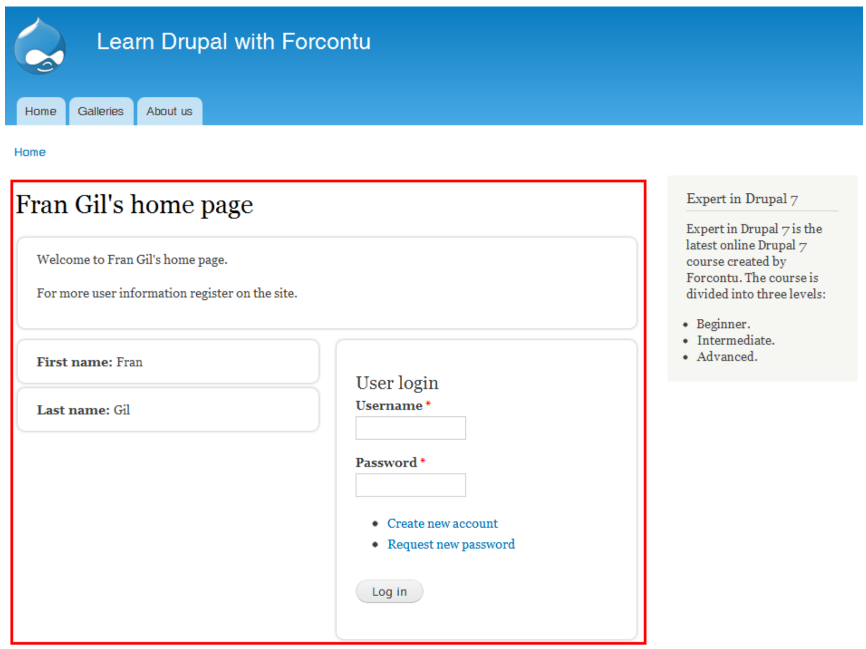
One implementation of the Panels is to build a home page for a site, incorporating several dynamic sections. Figure F27.1 shows an example of a page containing two columns with different regions, displaying both static content and dynamic content (views). This method of building content with other content is what we will build using Panels.

F27.1

Panels

An example of a panel made up of different regions and elements.

This example will be explored in the case studies later in the unit.



The Panels module is available here:

http://drupal.org/project/panels

Panels can be used to create three types of panels:

- **Panel page**. This is the fullest implementation of panels. A panel page operates like any other page, but we can use arguments to display dynamic content. As shown below, we open new possibilities in page construction by using arguments to reference other content in our Drupal site (NID, users, etc.).

- **Mini panel**. Mini panels are displayed as blocks. These blocks can function as normal blocks, appear within other regions, or appear within other more complex panels.

- **Panel node**. Similar to panel pages, panel nodes are more limited. The panel node contains different regions, but it cannot receive arguments. Panel nodes operate like nodes and appear in search results.

The Panels module relies on other modules, and the types of panels that are

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

available on a site will depend on which modules are enabled:

- **Panels.** This is the base module that takes care of managing the other modules that help panels do their job. This module does not include a user interface to create panels, so we will have to enable additional modules for this task.

- **Mini Panels.** This module allows us to create mini panels that can be configured as blocks within the regions of our theme. Mini panels also can be used within other panels.

- **Panel nodes.** This module creates panels as nodes. Compared to panel pages, this method operates on a more limited scope, but we can still apply the same options and functionality inherent to nodes, such as allowing comments or including the content in search results.

  To achieve this functionality, the module generates a new content type to the site, enabling us to create panels that operate as nodes.

- **Panels In-Place Editor.** This module enables site developers to perform certain edits directly from the final panel display, without having to navigate constantly to the panels management page.

The **Panels** module requires the **Chaos tool suite** module (known as **Ctools**), which is a suite of libraries that can be used by other modules. The Chaos tool suite module is composed of individual modules, specifically the following ones which are required by Panels:

- **Chaos tools.** This is the core of Ctools and provides the basic tools needed for many other modules, including the Panels module.

- **Page Manager.** This module provides the visual tools that the site developer will use to build and manage panel pages. This module **is required** for the case studies below.

- **Views content panes.** This module allows panels to display views.

The **Chaos tool suite (CTools)** module is available here:

http://drupal.org/project/ctools

Once installed, we will activate the following modules under **Panels**: Mini panels, Panel nodes and Panels. Under **Ctools**, we will activate: Chaos tools, Page manager and Views content panes.

27.2 Panel pages

The method for creating a **Panel page** is the most complete of all, and it will help us understand the general method of creating panels and the possibilities that open up in Drupal when we use them.

The method is very similar to create a **mini panel** or a **panel node**, but, as these are more simplified options, there are some limitations which we will explore below.

To create or edit panels, navigate to: F27.2

URL Panels
/admin/structure/panels

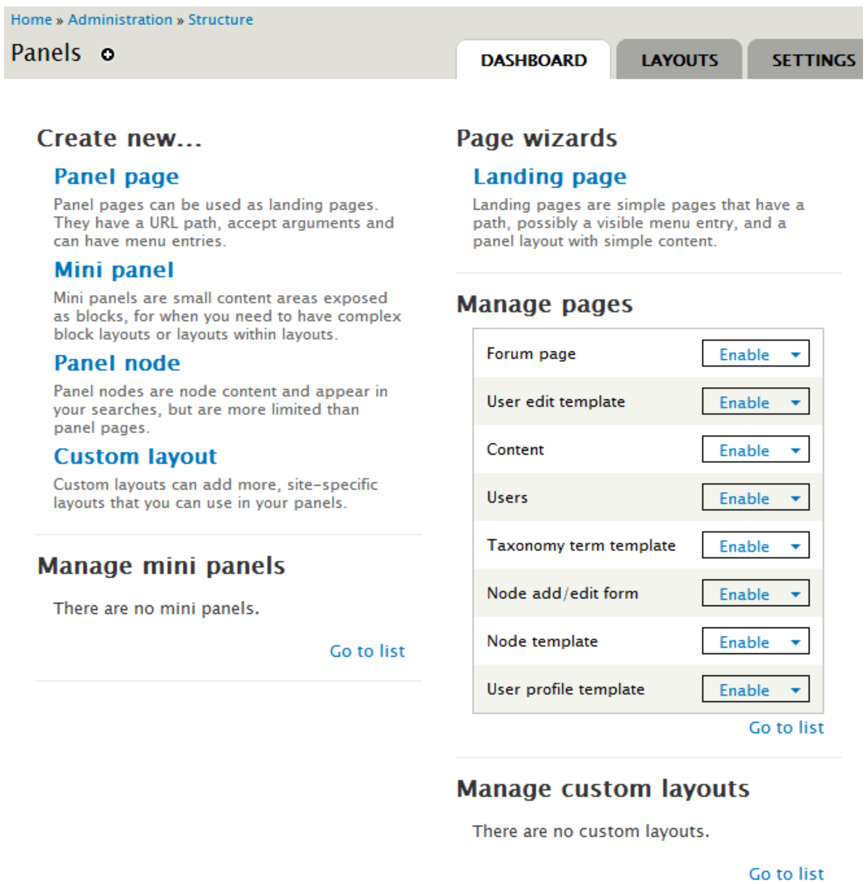
Administration ⇨ Structure ⇨ Panels

F27.2

Managing panels

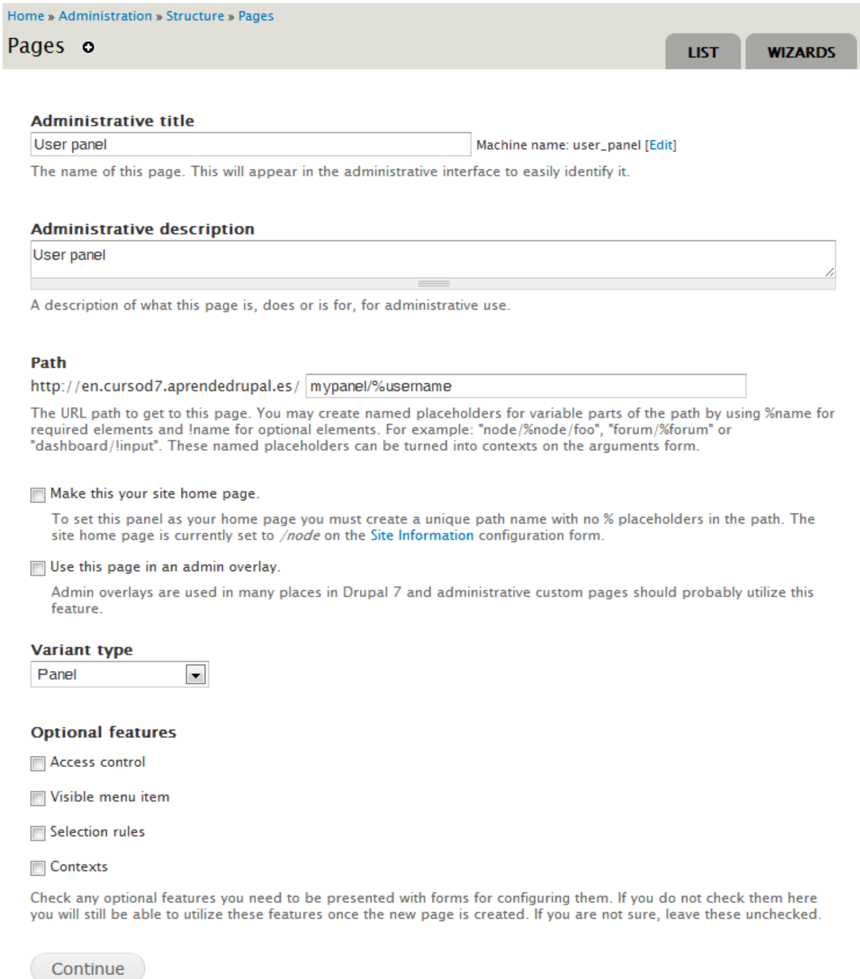
The panel management page provides all the options we need to create and edit panels that we have created.

The page has links to create panel pages, mini panels and panel nodes.



From the panel management page, we can create new panels and edit existing ones. First, we will create a new Panel page by clicking the link for **Create new... Panel page**.

Create and configure a panel page



F27.3

Create a panel page

Form to create a panel page.

If the panel will require arguments, we indicate them in the Path.

The options for configuring a new panel page are: **F27.3**

- **Administrative title**. This is the name that is displayed on the page management list. It is best to use short, descriptive names, to help identify pages.

- **Administrative description**. This is a brief description of the page to aid site management.

- **Path.** This is the URL to the new page. Optional or required arguments can be defined, which will be included in the URL, in the following format:

    - **mypanel/%argument**. The **%** symbol is used to indicate required arguments, which must appear in the URL. If the argument is not present, Drupal will display a 404 error or “not found” page.

    - **mypanel/!argument**. The **!** symbol is used to indicate optional arguments. If the argument is not present, a panel configuration error will be shown.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

For example, we can use **mypanel/%username** with a panel to show information about the user (user image, contact information, most recent comments, etc.). The panel employs the *username* argument to identify which user is being referenced.

In this example, the argument is required, meaning the site will generate an error page if an argument is not provided. When we use **mypanel/!username** and the username argument is not specified, however, the panel will simply ignore it. As shown below, there are a few options for instructing the panel how to proceed, if an argument is not specified in the URL.

- **Make this your site home page**. This option allows you to set the panel as the home page of the site. Another way to activate this setting is through the **Default front page** setting, at:

  Administration ⇨ Configuration ⇨ System ⇨ Site information

- **Use this page in an admin overlay.** As we've seen, the Overlay module enables us to access the management settings as an overlay. When this setting is enabled, the page will be displayed as an overlay using the **Overlay** module, whenever it is available. This option should be enabled only when the panel is being used for site management and is accessible to site managers.

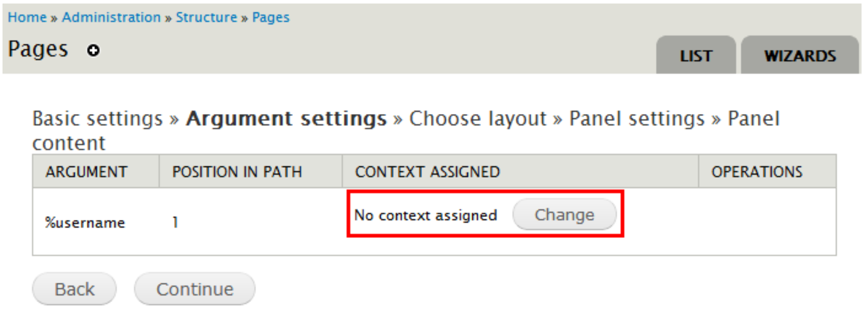
- **Variant type.** The two variant types are **Panel**, which creates a Panel page, and HTTP response code, which always generates the designated HTTP code (403, 404, or 301 redirect). This latter option is an advanced implementation that is not covered in this course, as we will focus on the **Panel** option.

- **Optional features.** With these settings, we can control different features for the new page:
  - **Access control**. Designate access control for the page (by role, permissions, etc.).
  - **Visible menu item**. We can link the panel page to a menu item.
  - **Selection rules.** We can identify a series of rules or conditions to determine whether the page is displayed or not. For example, we can tell the page to load only for certain languages.
  - **Contexts.** This setting is related to arguments, allowing us to define "what happens" with arguments specified in the URL **Path**.

Even if we do no select any of these options at first, we can change them later. When any one of these settings is enabled, we will be asked to configure additional settings during the following steps.

Define arguments (assign context)

When we click **Continue**, the settings are saved and we continue to the next step to create and configure the page. If the path includes arguments, the next step will ask us to configure these arguments. F27.4



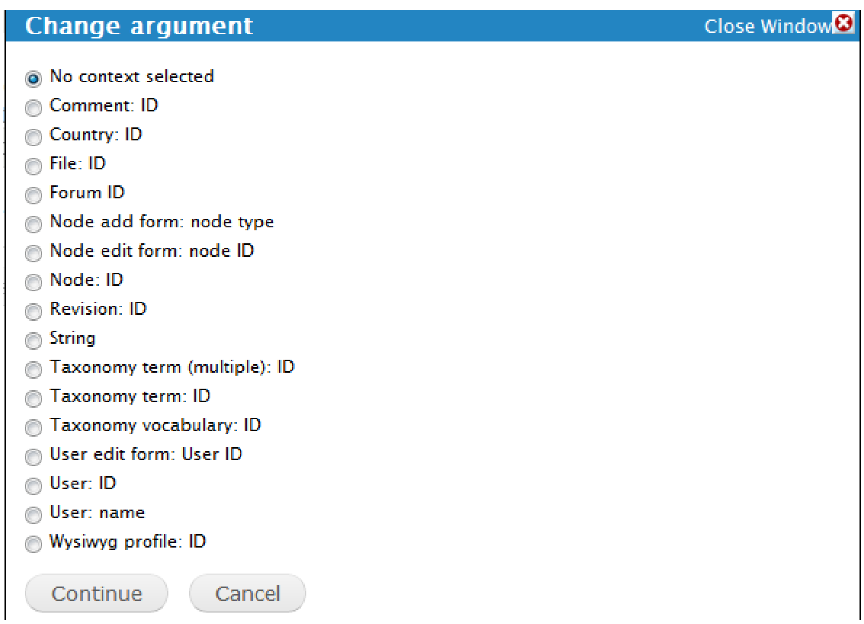
F27.4

Argument settings

Configure arguments that were defined in the path. Each argument should have a context assigned to it.

To assign a context to each argument, click Change. Here, we select the value that corresponds to the intended meaning for the argument. Some of the options available are: F27.5

- **Node: ID**. The argument will represent the node ID (nid).
- **User: ID**. The argument will represent the user ID (uid).
- **User: name**. The argument will represent the user name.
- **Comment: ID**. The argument will represent the comment ID.
- **Taxonomy term: ID**. The argument will represent a taxonomy term.



F27.5

Change context for an argument

We can assign context, which controls the meaning of each argument in the path.

Each of the elements in Drupal has a unique ID associated with it, which corresponds to its location in the database.

If a node has the URL node/54, then its identifier is 54. This identifier is called node ID, or nid. Users also have unique numerical identifiers, as do comments and taxonomy terms. While the user name is not a numeric value, it is also unique and is used as an identifier.

Therefore, in order to configure the panel URL to include an argument that specifies a node ID (using paths such as mypanel/%nid, node/%nid or node/%nid/*extra*), we select the ***Node: ID*** setting.

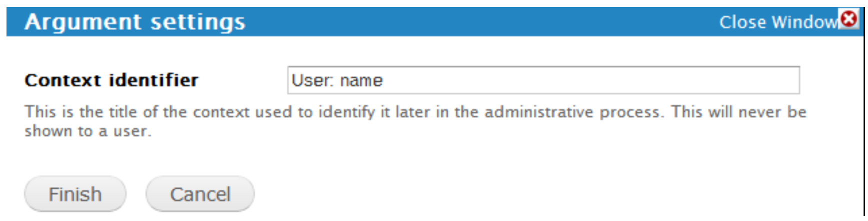
If we want to create a panel that shows information about a user, whose name is specified as an argument in the URL (for example, mypanel/%username), we select the ***User: name*** setting.

Once we make a selection, we are asked to type a context identifier. This name is used for site administration and is not shown to users. F27.6

F27.6

Context identifier

We can change the name for the assigned context.



Panel layout

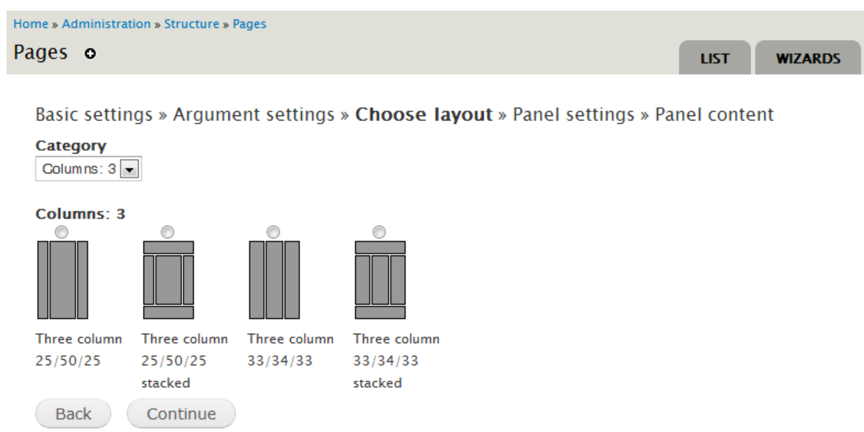
After assigning context for each argument, we click on **Continue** and come to the **Panel layout** setting. F27.7 There are several templates at our disposal, each with a different structure of rows and columns which we can use to display content inside the panel as needed.

F27.7

Choose layout

Several template designs are available by default.

We can select a category based on the number of columns needed, and then selet a template.



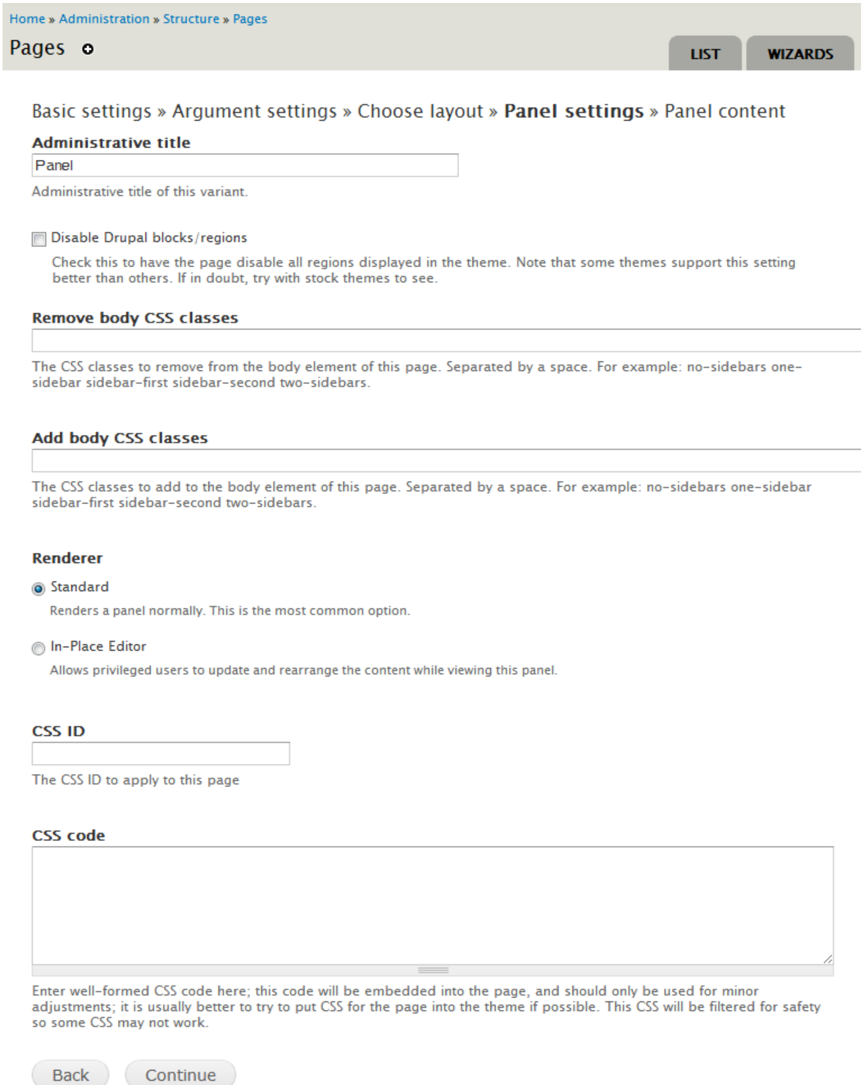
Panel settings

After selecting a layout, the next step is to configure the settings that will determine the display of the panel page. The options are: F27.8

- **Disable Drupal blocks/regions.** When this box is checked, the other blocks and regions in them will not be displayed. This is especially useful, for example, to create a home page for the site.

- **CSS ID.** We can designate a CSS ID for the panel, which is used when writing the CSS code to style the panel.

- **CSS code.** We can insert CSS code that will be embedded into the page. This is not intended for large blocks of code, rather it provides an opportunity to quickly test out changes to the panel display. This code will then be transferred to our CSS style sheet.



F27.8

Panel settings

After choosing the layout, we can configure additional settings, such as “Disable Drupal blocks/regions” to remove site blocks and have the panel display as a complete page.

This option is useful for creating panels that will be used as the site home page or as the landing pages for site sections.

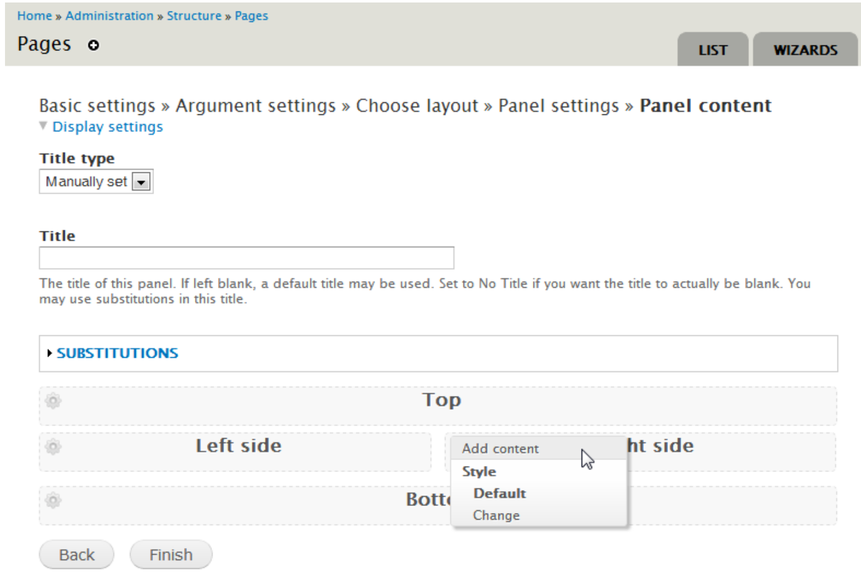
Adding content to the panel

The final step in creating a Panel page is to select the content that will be displayed in each of the panel sections for the layout we have chosen. F27.9

F27.9

Add content

In this step, we access the configurations to place content in each of the panel regions.



From the Panel content page, we can configure the page title, which can be set to use the available substitution tokens (using the Substitutions tool).

Each of the panel regions contains a gear icon in the upper left. The first option, when clicking the icon, is Add Content, which opens a window showing the available content to place. F27.10

F27.10

Add content. Choose elements

This window shows the elements that we can insert into the panel, arranged by category.

We can add many different page elements, from among all the content generated on the site. These elements may be generated by users, or by the installed modules. There are different categories of content, as well, based on the entities referenced in the panel context.

Add content Windows includes the following categories:

- **Activity.** Blocks related to site activity: Active themes, Recent comments, Who's online, etc.

- **Custom blocks.** Choose one of the blocks we have created on the site.

- **Widgets.** Choose a widget, such as the search form, language selector, or session start.

- **Page elements**. This includes elements related to pages, such as site name, page title, site logo, primary navigation links, etc.

- **Form**. Form fields. This includes fields related to entity references. For example, if we have set user as the context reference (with *username* or *user ID*), we can select the fields that are created by the **user** entity.

- **Menus**. Add any of the menus that have been created for the site.

- **Miscellaneous**. Other content elements.

- **User**. Fields related to the user entity. This category is only visible when the user entity is referenced in the context panel.

- **User (tokens)**. Replacement tokens for users. This category is only visible when the user entity is referenced in the context panel.

- **Node**. Elements related to nodes, such as title, body, or creation date. This includes all fields that we have defined on the site for different content types. This category is only visible when the node entity is referenced in the context panel

- **Node (tokens).** Replacement tokens for nodes. This category is only visible when the node entity is referenced in the context panel.

- **Views**. Select any of the views that are active on the site. Once selected, we will configure how to **display** content generated by the view.

  This category is only visible when we activate the **Views content panes** module. Also we must have created and activated at least one view.

In addition to the above categories, there are two more options:

- **Existing node.** This allows us to include an existing node into the panel. If we have defined relationships, the node may be referenced by the context.

- **New custom content.** This allows us to add new content, similar to creating a node.

Based on the elements we choose, the system will ask us configure additional settings.

One option that is shared by many elements is **Override title**, which allows us to select a different title for the element.

If the element has a context reference, we must indicate the relationship. For example, if we add the **User picture** element, from the **User** category, then we must define the context that is linked to that element, which is one of the defined user types. F27.11

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F27.11

Choose context for the element

For elements with context references, we must indicate the relationship.

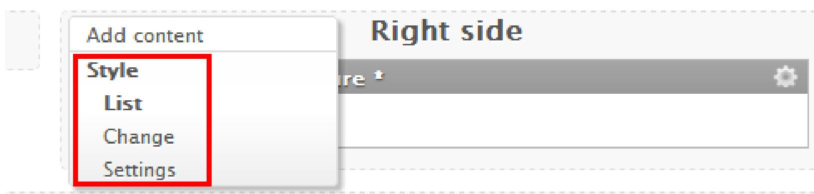
If we add a view as our page element, then we must define how to display the content in the panel.

Once the content is added and we return to the panels content page, we can edit the **Style** for the panel region, by clicking **Change**. F27.12

F27.12

Change Style

Change the Style of the region or the entire panel.



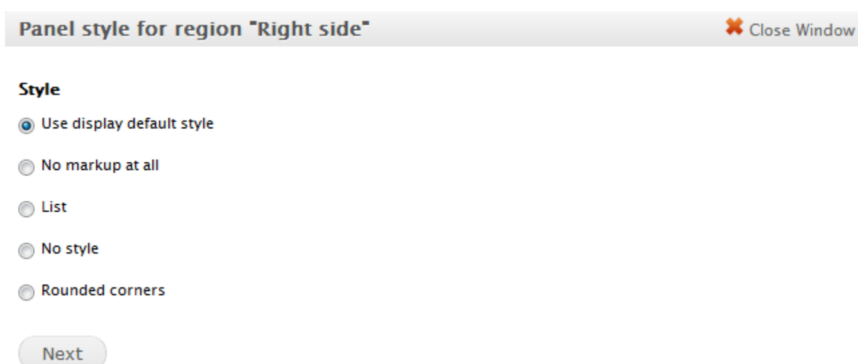
The style options are: F27.13

- Use display default style
- No markup at all
- List
- No style
- Rounded corners

F27.13

Change style

Available options.



Under the **Settings** option, we have additional configuration options that control the style. For example, if we select List, we can choose between an *ordered list* and an *unordered list*. For the Rounded corners option, we can define whether the box is drawn for the entire region or for each of the elements contained in the panes. It's important to remember that a region may contain various content elements, which can be reorganized by dragging and dropping their icons.

Once we have inserted content into the regions, we can access the configuration options for each content element, by clicking on the settings icon, in the upper right. F27.14

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F27.14

Content settings

After inserting content into the panels, we can configure settings for each element by clicking on the icon in the upper righthand corner.

The dropdown menu for each content element includes:

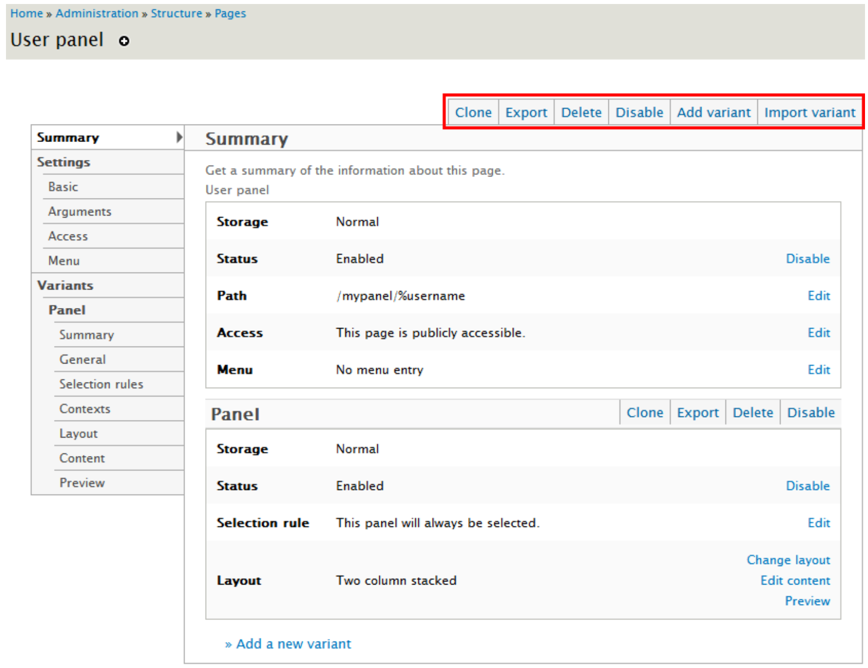
- **Disable this panel**. Disable the content from the panel.
- **Settings**.
- **CSS properties**.
- **Style**.
- **Visibility rules**. This option allows us to control visibility based on role, permissions, URL, personalized PHP code, etc.
- **Locking**. Define whether content should be locked (*Immovable*) or if it should be moved among different regions on the page (*No lock* or *Regions*).
- **Caching**. Enable or disable caching for the content.
- **Remove content**. Delete the content from the panel. The original content is not affected, but this option removes the content element and the configuration options for the panel that were created by adding **New custom content**.

Once we have added content to each of the panels, click Save to complete the process of creating a new panel.

Panel configuration menu

Each panel has its own **configuration menu** where we can edit parameters.

At the top, we have a menu bar with the following operations: F27.15



F27.15

Panel configuration menu

The panel configuration menu allows us to adjust settings.

From this page, we can configure other settings for the panel. Click **Save** to save any changes.

The top menu includes operations such as Clone, Export, Delete, etc.

- **Clone**. Create a copy of the panel with identical settings and content elements in a few short steps. This is helpful when you need to create panels that have many settings in common.

- **Export**. Export code from the panel. This code can then be inserted into a module or into another panel, with the **Import variant** operation. This enables us to make backup copies of a panel, in the form of a text file containing the generated code, which can then be imported into other panels or imported to another Drupal site.

- **Delete**. Delete the panel. With this option, panels cannot be recovered.

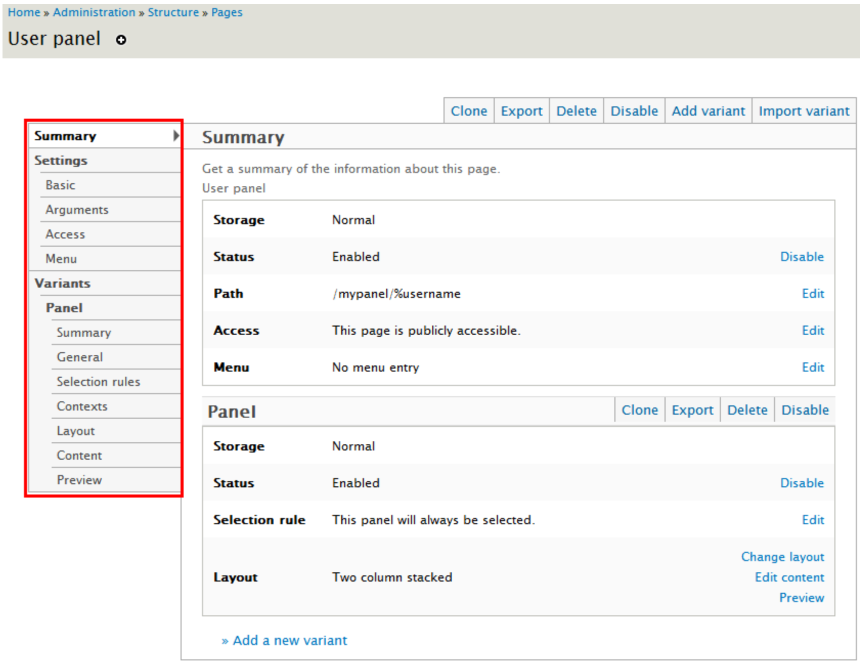
- **Disable**. This operation blocks site users from accessing the page, but the configuration settings remain stored in the system. If a panel is disabled, the menu option changes to **Enable**, and the panel can be reactivated.

- **Add variant.** One panel page may refer to several variants or Panel configurations, allowing us to maintain various displays for the page. Multiple variants can be created for a panel, much like multiple displays can be created for an individual view.
  Selection rules will determine which variant will be shown to the user when he or she visits the page. For example, we can create a specific variant for users of a certain role. Later, we will see how to configure **Selection rules** for each variant.

- **Import variant.** Import a variant that was exported earlier using the **Export** operation. We can import variants created by other panels, or by other Drupal sites.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

The left menu includes the following options, some of which we saw earlier when we created the panel: F27.16



F27.16

Panel configuration menu

The left menu displays options for basic settings and variants.

- **Summary**. Basic information about the panel configuration. We also have links to quickly access panel settings that we can edit (URL, access, menu, layout, etc.)

- **Settings**. Global options for the panel, categorized by:

  - **Basic**. Basic options for the page (administrative title, URL, etc.).

  - **Arguments**. Configure arguments that are passed to the page URL. We can use this to define the context for each argument, as we did above.

  - **Access**. Create rules for page access. For example, we can create a rule so that the page is only visible to users with certain roles or permissions.

  - **Menu**. Create a menu link for the page.

- **Variants**. These configuration options are used to control each variant for the panel. After following the steps above to create a panel, we have defined the first variant. The options are:

    - **Summary**. Basic information about the configuration of the variant.

    - **General**. General options for the variant.

    - **Selection rules**. Create rules to define under what conditions the variant is displayed. During initial setup, the page will display the first variant that meets the pre-defined criteria.

    - **Contexts**. Add contexts or functions for the relevant arguments. The context can refer to a node, the edit page for a content type (create or edit a node), a user, etc. After adding the context, we can select the relevant elements to add to the panel content, from the **Content** setting. From the **Relationships** option, we can link related elements to the context. For a context that references users, we can link to other elements, such as files, nodes or comments created by the user, by adding the relevant relationship.
      Whereas **views** relies on arguments and relationships, panels uses contexts and relationships.

    - **Layout**. Control the layout or template for the panel variant.

    - **Content**. Control the content displayed in the variant.

    - **Preview**. Show a preview of the panel variant. For pages that require arguments in the path, we can define those terms before generating the preview.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Practical case 27.1 Create a panel with arguments

In this exercise, we will show the steps to create a panel page that displays information related to users, which we will define in the URL. Some of the characteristics of this page are:

- **Administrative title**: User homepage
- **Machine name**: user_homepage
- **URL path**: user-homepage/%username
- **Argument**: Username (required)
- **Variant**: Registered users (this option is only visible to registered users)
- **Layout**: Two columns bricks
- **Panel title**: *FirstName LastName*'s homepage.
- **First row** (1 column): Heading for the view
- **Second row** (2 columns):
    - Left: Username, First name, Last name and Email.
    - Right: user roles, last login, account creation date.

Step 1. Create the panel

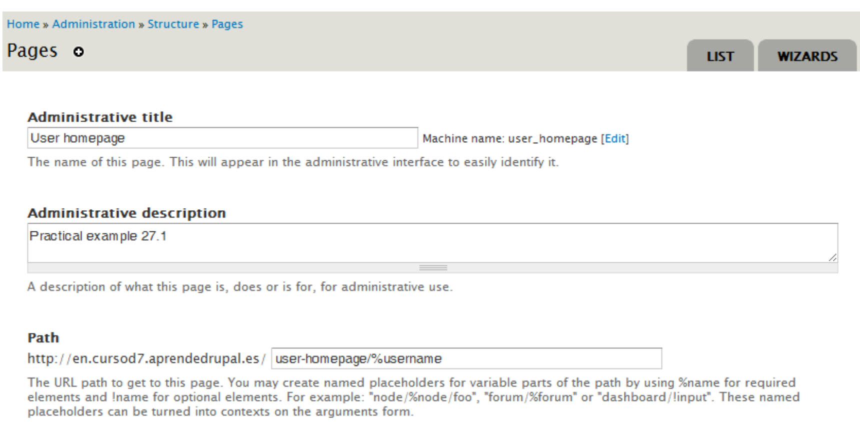
To being the process of creating a panel page, navigate to: F27.17

Administration ⇨ Structure ⇨ Panels [Create new… Panel page]

URL Panels
/admin/structure/pages/add

The administrative title will be **User homepage** (and machine name is **user_homepage**).

In this case, the argument is required, so we use the % symbol. The URL for the panel becomes: **user-homepage/%username** (we can use another name for the argument, but it's important to remember to assign it to the context).



F27.17
Practical case 27.1
Add an argument to the panel
The first step is to define the argument by typing it into the **Path**. We use the % symbol before the argument to indicate that it is required.

Do not check any of the boxes under **Optional features**. We can configure these options after we have created the panel. Click **Continue** to advance to the next step in creating the panel page.

Step 2. Define the context

Next we will define the context for the %username argument in the URL. Click on Change and select the **User: name** option. Now the page will expect that any value in the first position of the URL will refer to a user name. F27.18

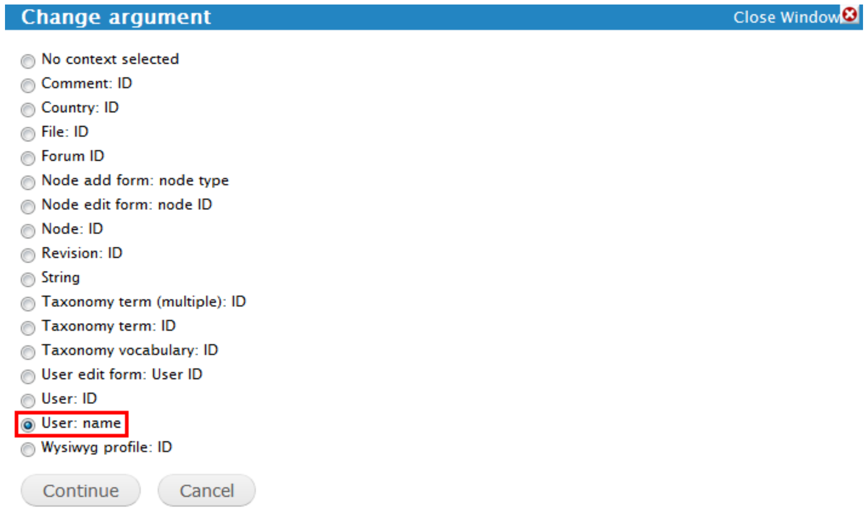
F27.18

Practical case 27.1

Define the context for the argument

Define the context.

Here, we configure User: name as the argument.



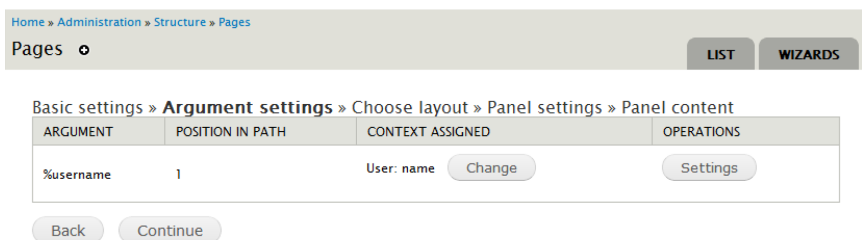
We don't need to change the name of the context (User: name). The argument settings and the assigned context will appear in the **Argument settings**. Click **Continue** to advance to the next step. F27.19

F27.19

Practical case 27.1

Define the context for the argument

The list of argument settings shows the argument and the context that is assigned to it.



Step 3. Choose layout

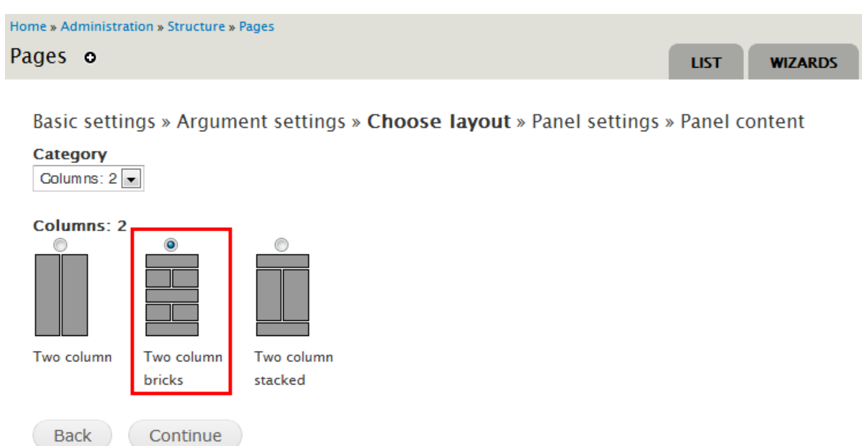
Change the layout to "Two column bricks". F27.20

F27.20

Practical case 27.1

Choose layout

Choose the "Two column bricks" option.



After selecting the layout, we are asked to configure settings for the variant. For **Administrative title**, type: "**Registered users**". Leave the other options unchanged, and click **Continue**.

Step 4. Panel title

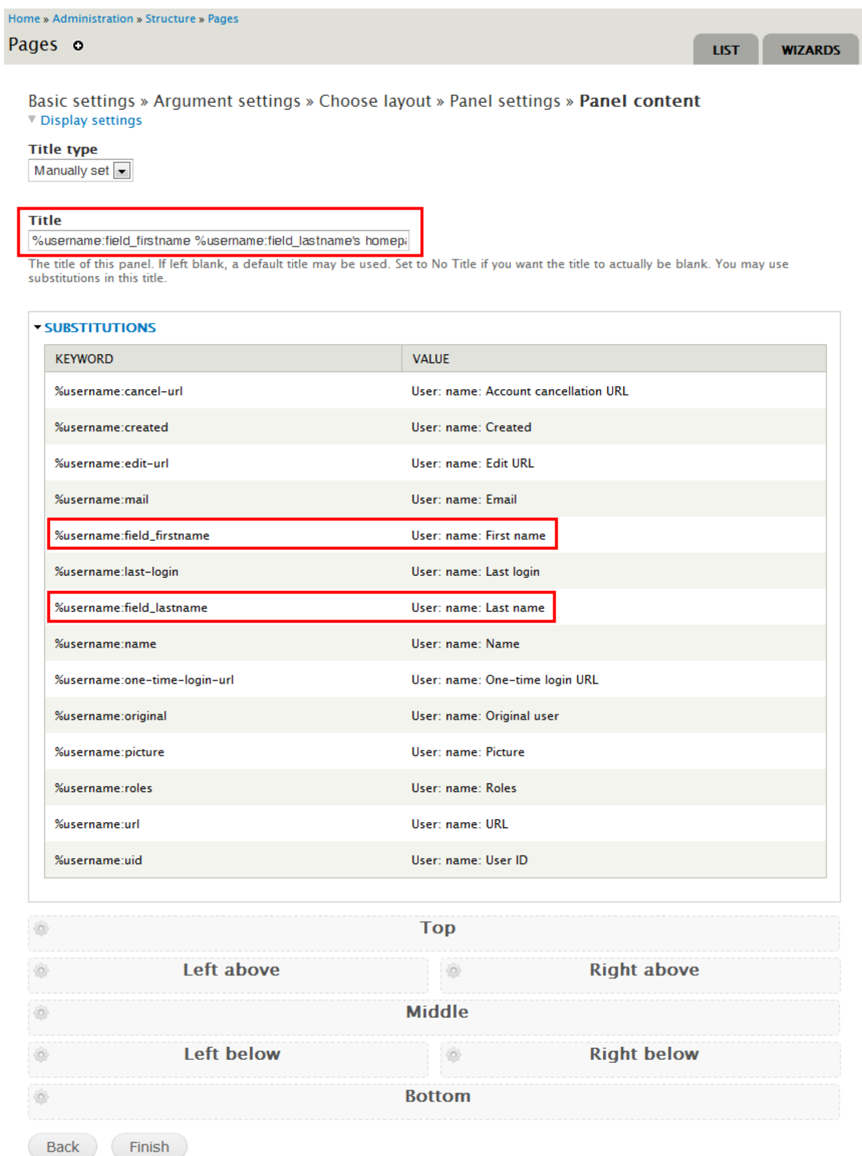
The title of this variant will use substitution tokens, which we access by opening the **Substitutions** option.

Keep in mind that the fields **field_firstname** and **field_lastname** were created earlier for the **User** entity. If these fields don't appear, then you must create them at the following link:

Administration ⇨ Configuration ⇨ People ⇨ Account settings [Manage fields tab]

URL User fields
/admin/config/people/accounts/fields

The title, which includes tokens that will be replaced with actual user data, is: **%username:field_firstname %username:field_lastname's homepage** F27.21



F27.21
Practical case 27.1
Panel title
We create the panel title using substitution tokens.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Before assigning content to the different regions in the panes, let's save the panel by clicking **Finish** (and then Save).

Step 5. Content (top region)

After we make sure we have saved our work to this point, we begin adding content to the variant from the **Content** option.

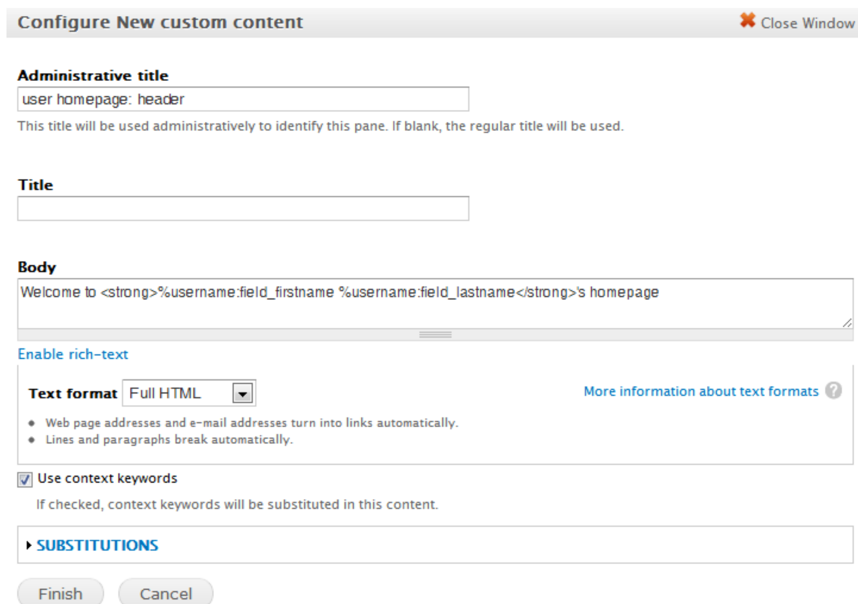
To add content, click on the gear icon in the upper left-hand corner of each box, and choose the **Add content** option.

In the **Top** row, let's add a heading set to **no title**. We do this by choosing **New custom content**. Here we also use replacement tokens. When we set the text format to Full HTML, we can insert any HTML tags needed.

For this exercise, insert the following text: F27.22

- Welcome to %user:field_firstname %user:field_lastname's homepage.

F27.22
Practical case 27.1
Content
In the Top region, add a new customer content with HTML text.



We can begin to see how the panel will look if we click on the Preview button, or if we type the URL into the browser address bar, including a valid user name

Let's change the region style to "**Rounded corners (each region)"** to differentiate the regions from each other. F27.23

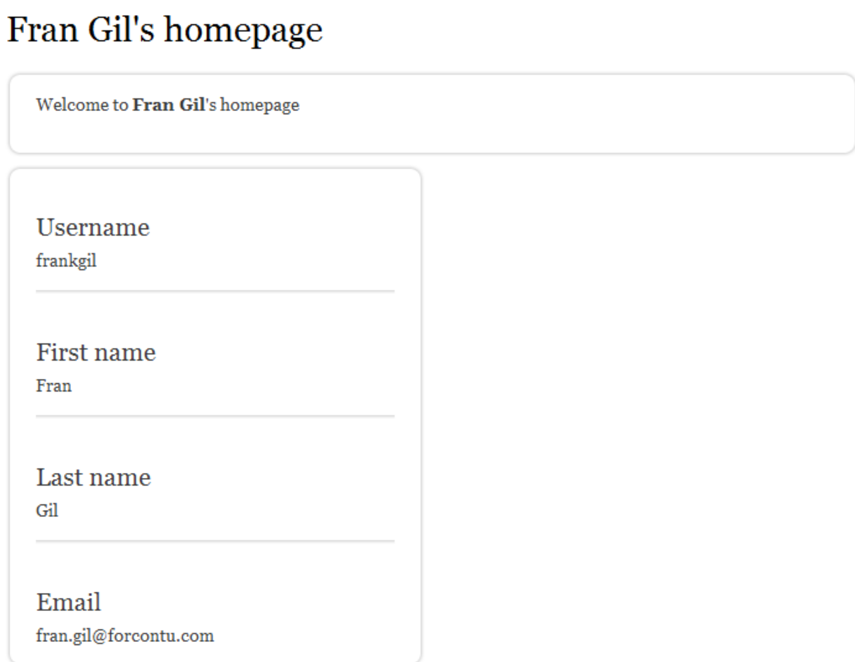
F27.23
Practical case 27.1
Apply style
Apply the Rounded corners style to each region.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Step 6. Content (Left above region)

In the **Left above** region, let's add the **username**, **first name**, **last name** and **email** fields.

The **Username**, **First name**, **Last name** and **Email** fields are located under **User** (**tokens**). Keep in mind that the first element "Name" refers to the user's name in the system (username). Let's override the element title Name, and change it to Username. After adding content to the region, change the style to "**Rounded corners** (each region)". F27.24



F27.24
Practical case 27.1
Panel layout
Preview the panel layout to see the final display as we add content.

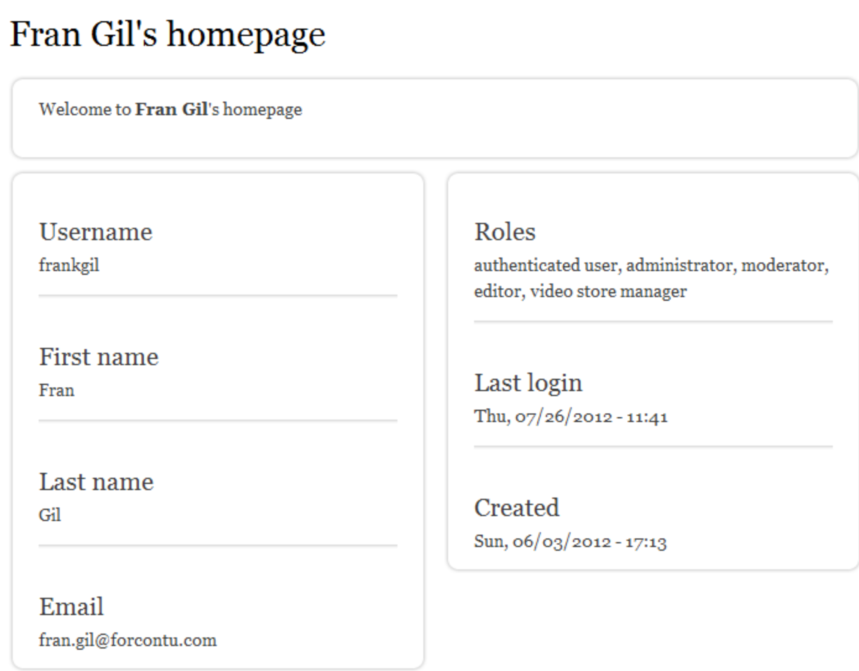
Step 7. Content (Right above)

In the **Right above** region, let's add user roles, last login and account creation date.

Each of these elements is available under **User (tokens)**: **Roles**, **Last login** and **Created**, respectively. F27.25

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F27.25
Practical case 27.1
Panel layout
By adding content to the Right above section, the panel is complete.

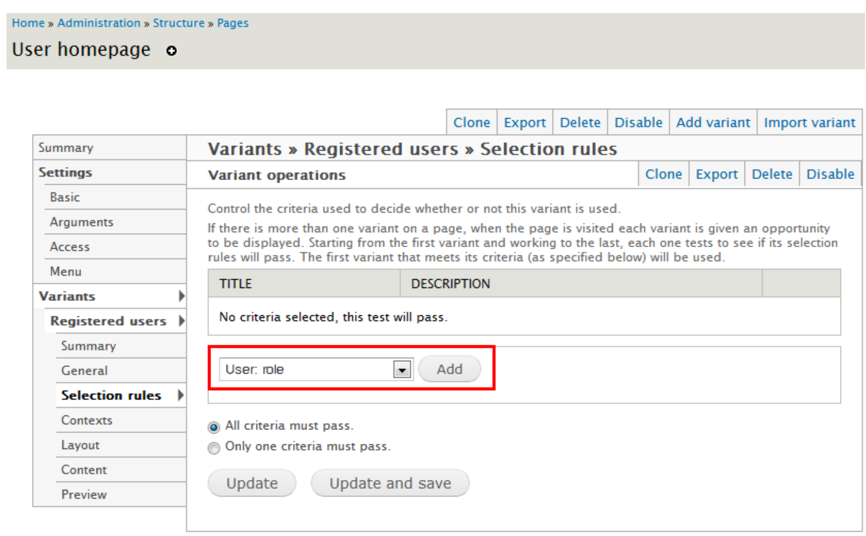


Step 8. Access control

To finish setting up the panel, let's configure the variant so that it's only visible to registered users. We accomplish this by adding a **Selection rule** that references user role.

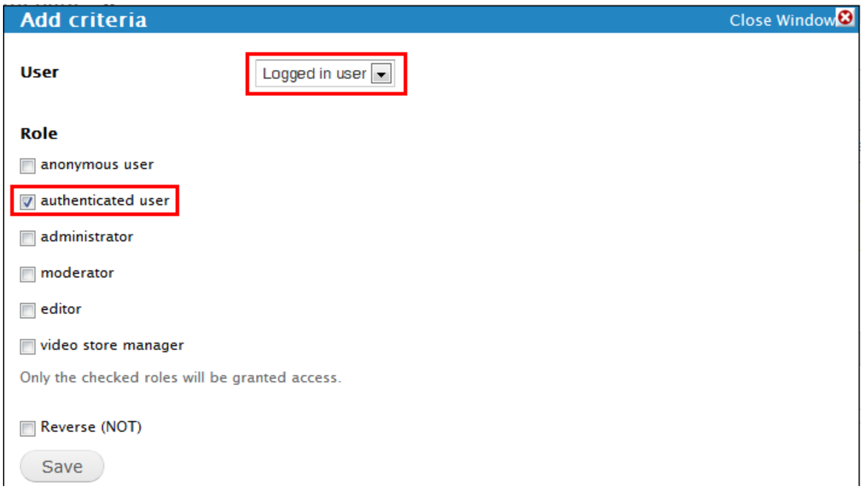
From the **Selection rules** panel, find the **User: role** option and click **Add**. F27.26

F27.26
Practical case 27.1
Selection rules
Use selection rules to control who has access to this variant. In this case, we are implenting control with user roles.



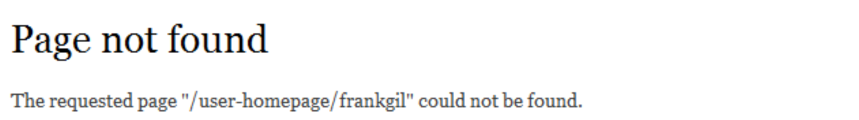
For the configuration above, we must indicate that **Logged in user** is required for the test to pass. Select **User** in the upper selector. Below, select the **authenticated user** role and save the changes to the panel. F27.27

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



F27.27
Practical case 27.1
Selection rules
Only registered users can access this panel variant.

The only thing left to do is confirm that the panel is working correctly. When we log in as a registered user and navigate to the page, we should see the panel display as expected. If we do the same as an anonymous user, the system should generate an error page. F27.28



F27.28
Practical case 27.1
Selection rules
Anonymous users will see an error page not found.

Practical case 27.2 Create a panel with two variants

Next we are going to create a new variant for the page we created above in **Exercise 27.1**:

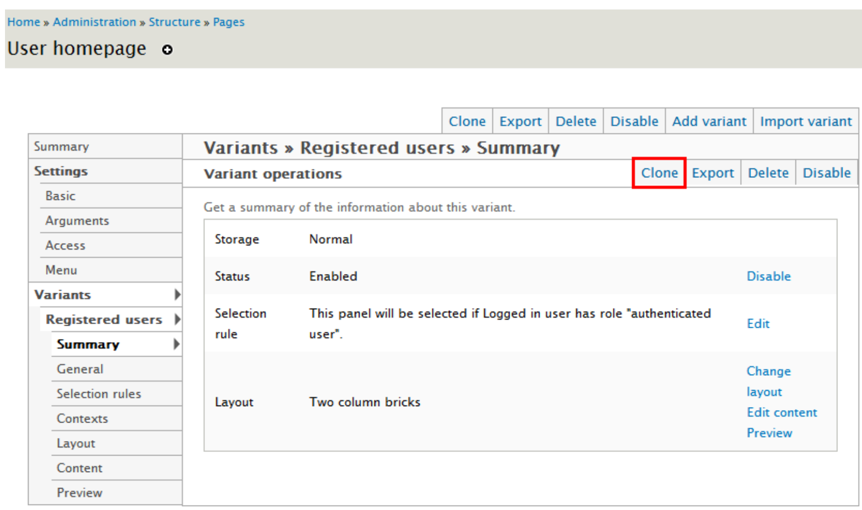
- **Administrative title**: User homepage (already created)
- **Variant**: Anonymous users (this variant will only be visible to anonymous users)
- **Layout**: Two column bricks
- **Panel title**: *FirstName LastName*'s homepage.
- **First row** (1 column): View header
- **Second row** (2 columns):
    - Left: Username, First name, Last name and Email.
    - Right: User login form.

Step 1. Clone the variant

We will make only a few changes to the existing variant, so the easiest option is to clone the panel and make the necessary changes to the new variant.

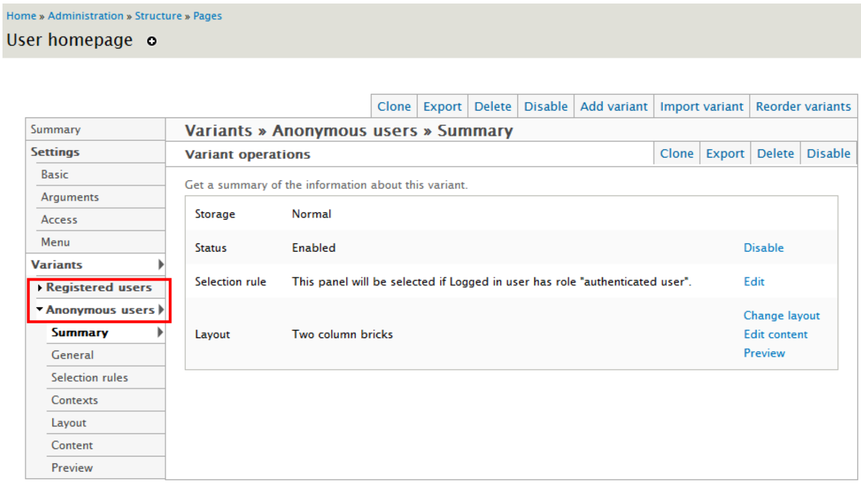
Click on the **Clone** button to generate the clone. Take care not to select the upper Clone button, which will clone the entire panel. F27.29

F27.29
Practical case 27.2
Clone the variant
To create a variant that is very similar to another existing panel, we can clone the panel and then change the configurations as needed.



After cloning the variant, we give it a new name, **Anonymous users**, and save the changes (Update and save).

With the second variant created, the left menu shows the two variants that are available, **Registered users** and **Anonymous users**. Let's starting configuring the new variant. F27.30



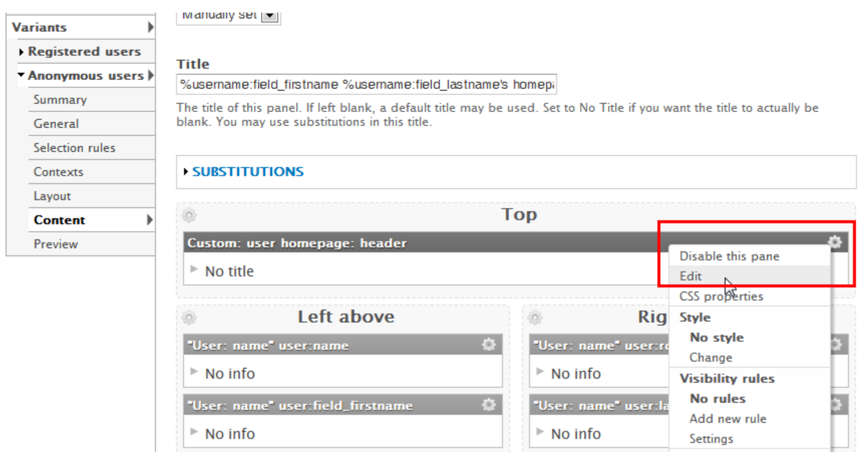
F27.30
Practical case 27.2
New variant
The new variant will be displayed in the left menu, underneath the Variants option.

Step 2. Content (top region)

Let's make a minor change to the content that is displayed in the header. Since this variant is only visible to anonymous users, we'll add the following text:

- "For more user information please register on the site".

We can edit the content elements for any region by clicking on the gear icon in the upper right-hand corner of any element. **F27.31**

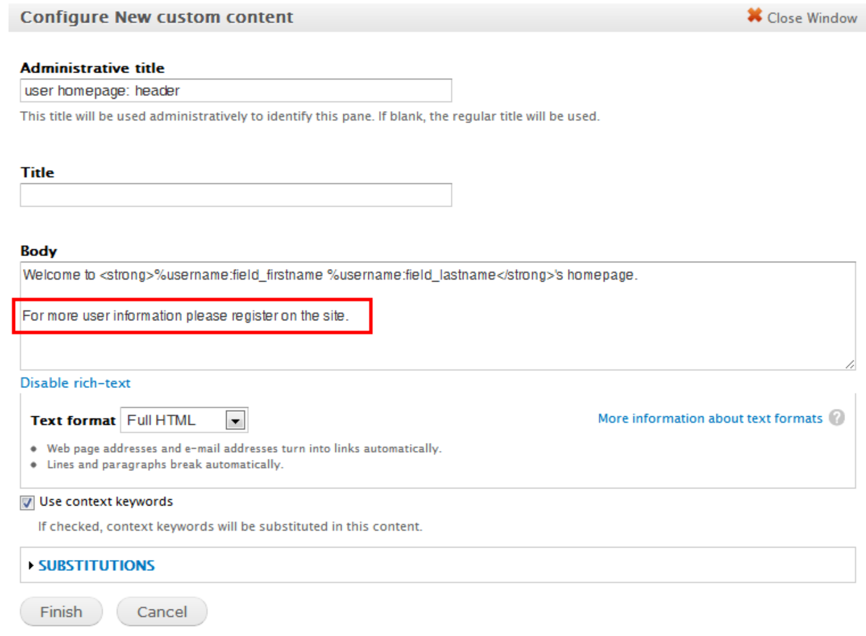


F27.31
Practical case 27.2
Content
For the new variant, we remove the content that should not be visible and add new content elements.

Let's edit the **User homepage: header** element and add the relevant text. **F27.32**

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F27.32
Practical case 27.2
Content
The new variant will display additional text in the content header.



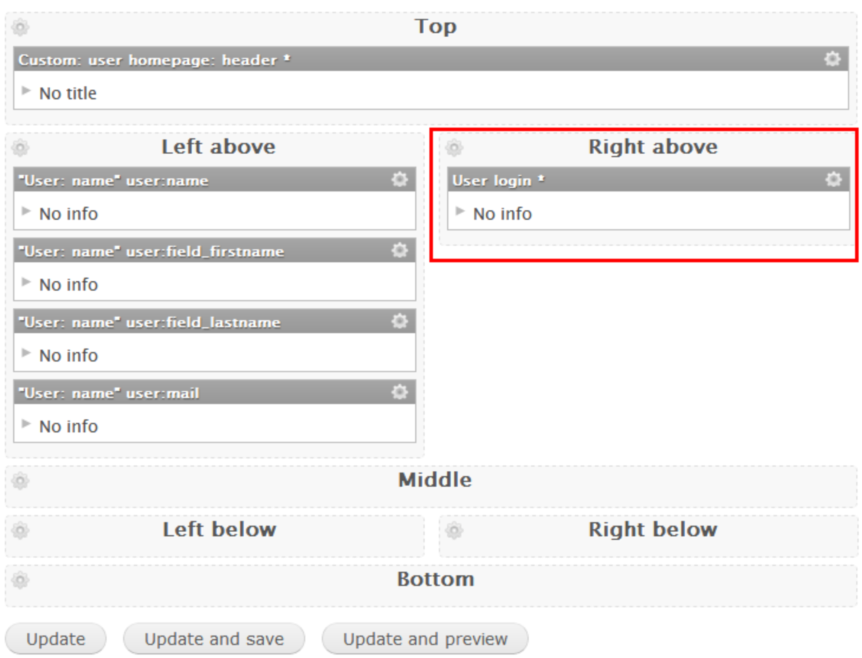
Step 3. Content (Right above region)

We make no changes to the **Left above** region. In the **Right above** region, however, let's start editing the variant by **removing** all the existing content. Again, click on the gear icon in the upper right-hand corner of the element and select **Remove**.

After removing the existing content, add **User login** content, which is listed in the **Widgets** section.

The **Figure** below shows the content elements for the variant. **F27.33**

F27.33
Practical case 27.2
Content
The login window will be shown to anonymous users.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

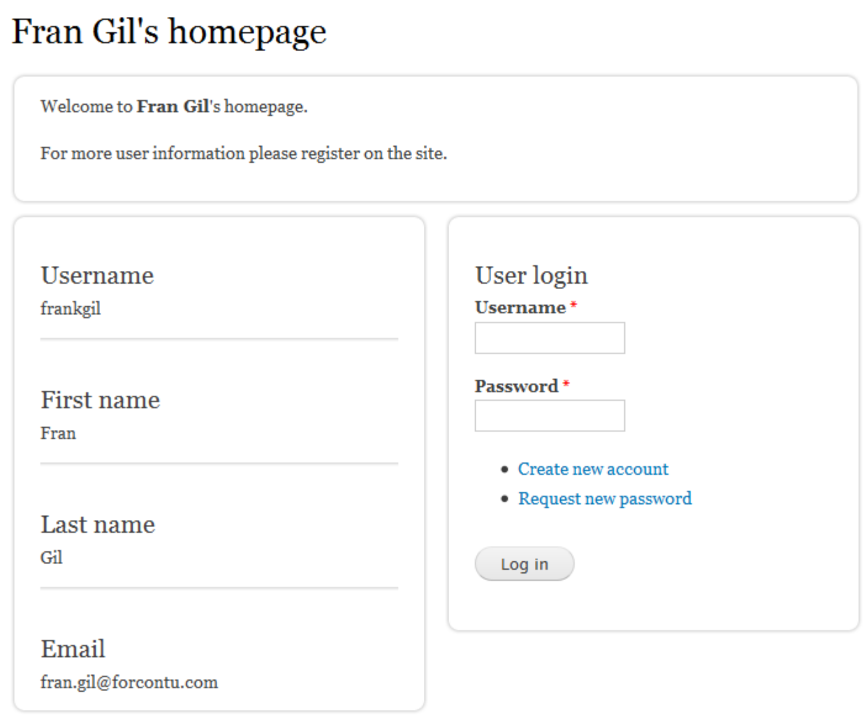
Step 4. Access control

To complete the process, let's configure the variant so that it's only visible to anonymous users.

First, **remove the existing selection rule**, which was cloned from the earlier panel variant.

Under **Selection rules**, find the **User:role** option and click on **Add**. Then select the **anonymous user** role, setting the upper selector to **Logged in user**.

Finally, let's make sure the page is operating properly both for registered users and for anonymous users. The new variant will display the additional text in the top section, and the user login window appears below. F27.34



F27.34

Practical case 27.2
Variant display

The variant display for anonymous users.

Under selection rules, we must configure the access role for the variant, which is applied to anonymous users.

27.3 Managing panels

Panels Administration Area

From the **Panels Administration Page**, we can create new panels and edit existing ones. F27.35

URL Panels
/admin/structure/panels

Administration ⇨ Structure ⇨ Panels

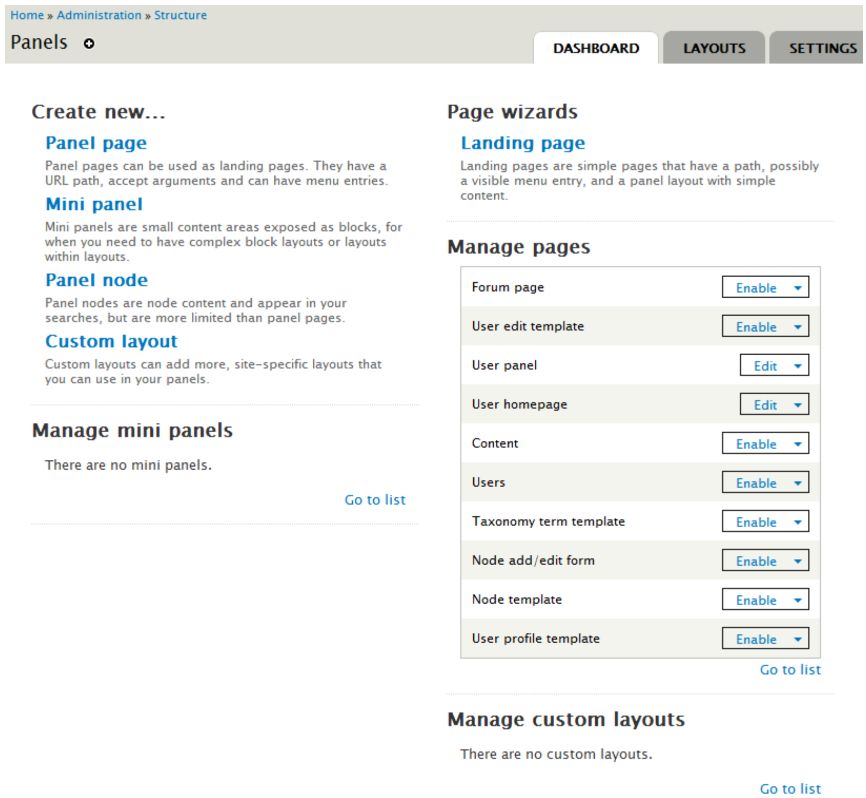
F27.35

Panels administration

The panels administration page (dashboard tab) gives an overview of the panels being used on the site.

Here we can see any mini panels or panels that we have created on the site.

We also can see any custom layouts that have been created.



The administration page has the following options:

- **Manage mini panels**. A list of mini panels created on the site. We will learn more about mini panels below.

- **Create new**... Direct access to create various types of panels (Panel page, Mini panel, Panel node or Custom layout).

- **Page wizards**. This tool complements the basic panel creation tools and users create a node template or landing page. The node template is a method to **override the normal display of content**, converting it into a panel. A landing page is a panel that we can use as the site's homepage or as the main page for certain sections. Normally we won't use this method to create panels.

- **Manage pages**. Displays the available panel pages, along with links to enable/disable or edit each one.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Manage custom layouts**. From the **Layouts** tab, we can create custom layouts to organize content in our panels. Custom layouts will appear here.

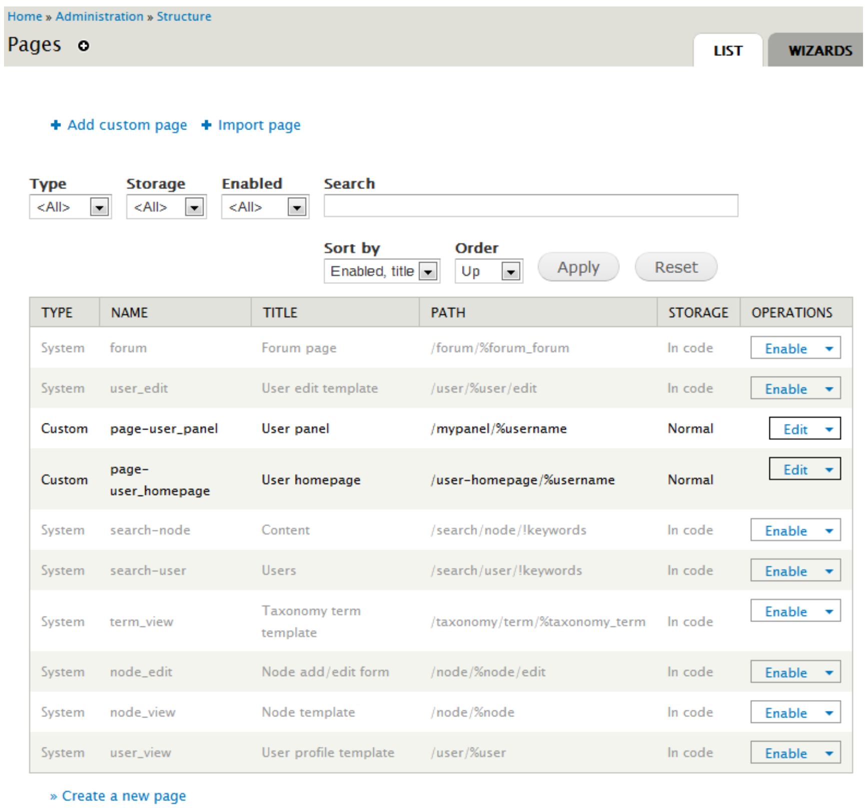
Under the **Manage pages** section, we can click on **Go to list**, to access the **Pages** section. Here we can manage the **panel pages** on the site.

We can also navigate to the Pages administration page with the following link:

F27.36

Administration ⇨ Structure ⇨ Pages

URL Panel pages
/admin/structure/pages



F27.36

Pages list

The list of panel pages allows us to manage the pages used on the site. Disabled pages are panels that are created by other modules on the site. We can enable those panels to add new pages, or to overwrite core pages, such as a user's profile page, or the node edit page.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Custom layouts

From the **Layouts** tab, we can create new layouts for the content in our panel. These layouts can be applied to any panel on the site. F27.37

F27.37

Panels administration
Layouts tab

The **Layouts** tab includes the tools we need to create new layouts that can be applied to our panels.

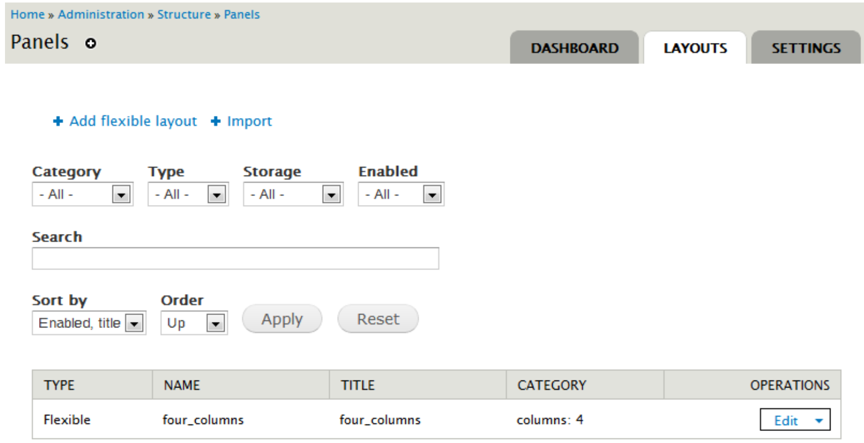
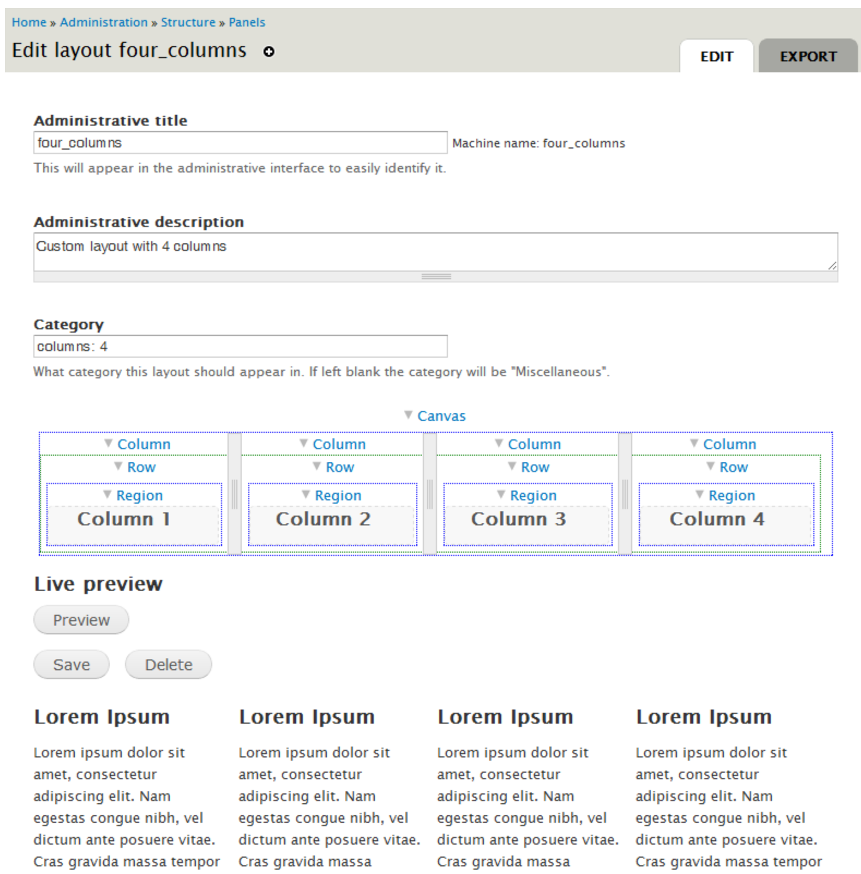


Figure F27.38 contains an example of a new **4 column** layout. First, add the columns to the canvas, and then we can add rows and regions to each column. The system is very flexible, allowing us to build rather complex structures.

F27.38

Panels administration
Layouts tab

A custom design with four columns of equal width.

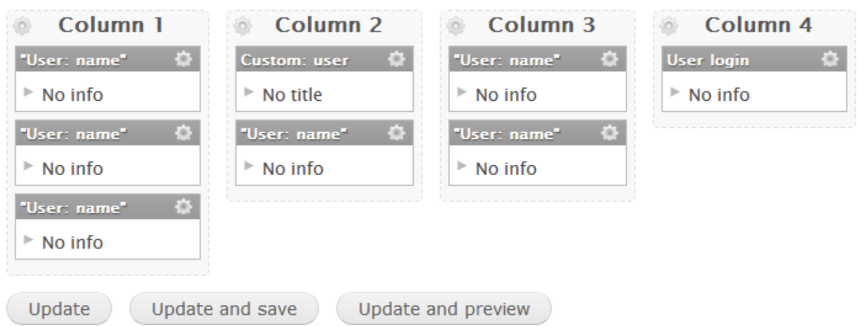


This layout is now ready to be applied to one of our panels. Although the structure shown in preview does not match the content scheme we created earlier, it will display properly once we apply it to our content. F27.39 F27.40

F27.39

Custom layouts

Custom layouts can be selected for each panel, and appear in the category we selected when we created them.

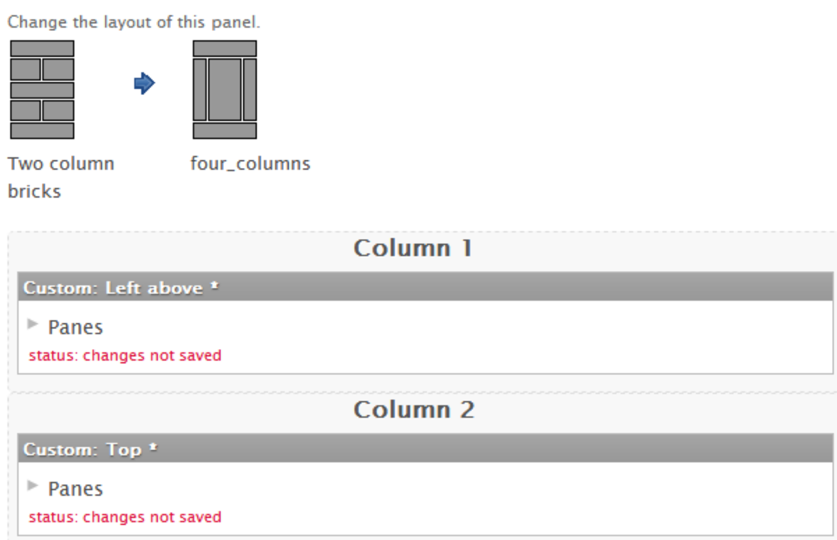


F27.40

Custom layouts

One example of applying a custom layout to content. The process is similar to applying default layouts.

When we change the layout of a variant that already has content assigned, the system will ask us where to position the original content in the sections of the new layout. F27.41



F27.41

Edit the layout

When we change the layout of a panel, the contents should reposition themselves in the regions of the new layout.

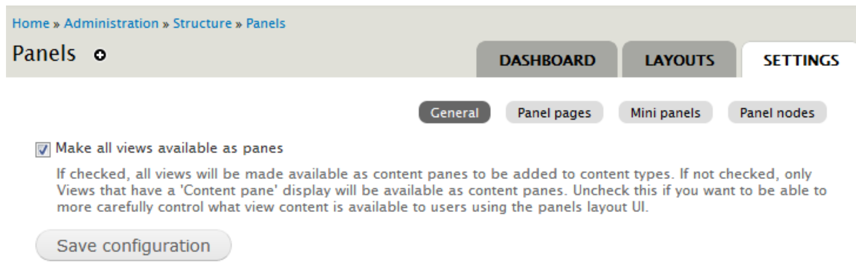
We will have to indicate where the panes in the original layout should be displayed in the new layout.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

General settings

Under the **Settings** tab, we can configure general options that control the behavior of our panels. There are several secondary tabs under the **Settings** tab. The **General** tab includes:

- **Make all views available as panes.** This option makes all views available to insert into a panel. If this setting is disabled, then only views that were created with a Content pane display are available. F27.42



F27.42
Panels administration Settings tabs
Control general options for all panels, and settings for individual panel styles. The General options tab includes a setting to make all views available as panes.

Under the **Panel pages, Mini panels** and **Panel nodes** tabs, we can configure the behavior of each type of panel and the layouts that are available when we create new panels: F27.43

- **New content behavior**. Choose the elements, such as content, that will be available for users creating new panels on the site. If the **New block** checkbox is enabled, for example, then each block that is created on the site can be inserted into a panel region. All elements are enabled by default.

- **Select allowed layouts**. Choose the layouts that are available to users creating new panels. This list also includes custom layouts that we have created. All layouts are enabled by default.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F27.43

Panels administration Settings tab

Under the tabs for **Panel pages**, **Mini panels** and **panel nodes**, we can choose the site contents that we want to insert in each of the different panel types.

We can also select layouts that are available for users to apply when creating new panels.

27.4 Mini panels

Mini panels offer us an interesting method to display small content sections, with the specific structure and layout that we want. These sections are available as blocks or panels, which can then be inserted into other panels. In this manner, we can create complex panels, offering even more flexibility to the design and structure of our pages.

To create a **Mini panel**, navigate to: F27.44

URL Create Mini panel
/admin/structure/mini-panels/add

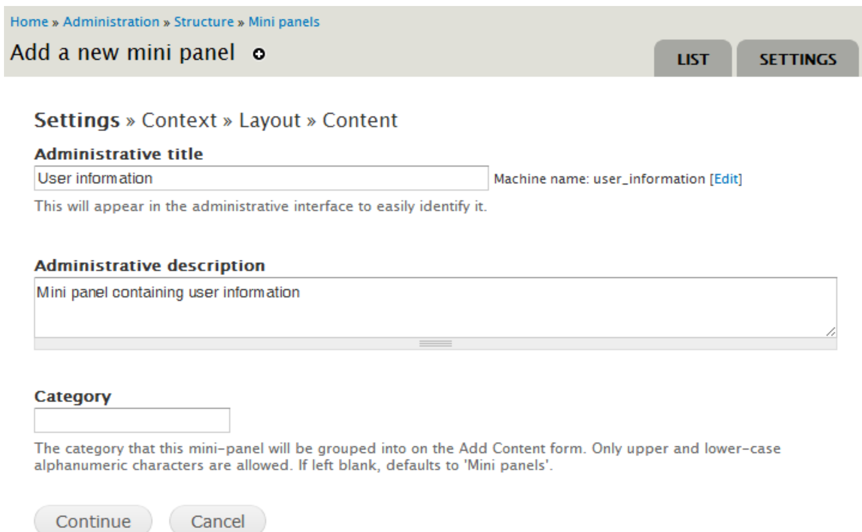
Administration ⇨ Structure ⇨ Panels [Create new... Mini panel]

F27.44

Create a Mini panel

The first step to create a mini panel is providing an administrative title, a description and a category, if needed.

If we do not specify a category, the new Mini panel will appear under the "Mini panels" category.



On the first page to create a new Mini panel, we have two options:

- **Administrative title** and **Administrative description**. As with other content, we want to create a short, descriptive title for our mini panel, which will help site administrators identify it.

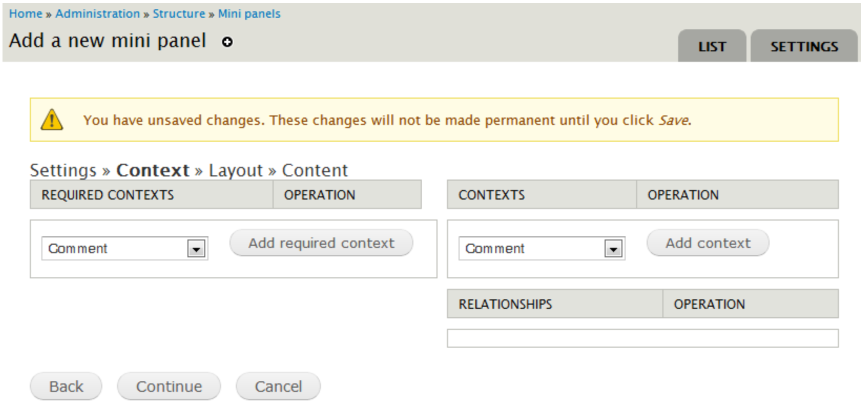
- **Category**. Provide a category to include the mini panel in a list with others. By default, mini panels will appear in the "Mini panels" category, unless we define another category.

Context and relationships

In the next step, we have the option to apply context and relationships. Note the difference between **required contexts** in the left column, and **contexts** and **relationships** in the right column**.**

When a **required context** is applied, then we can only add the mini panel to panels that satisfy the context. For example, we can not add the mini panel to the panel created in case studies 27.1 and 27.2, unless we add the **User** context. The mini panel simply will not appear in the list of content to include in the panel.

The function of the context depends on the context settings applied to the panel that contains the mini panel. In the case studies above, the username values will correspond to the username that appear in the URL argument. F27.45



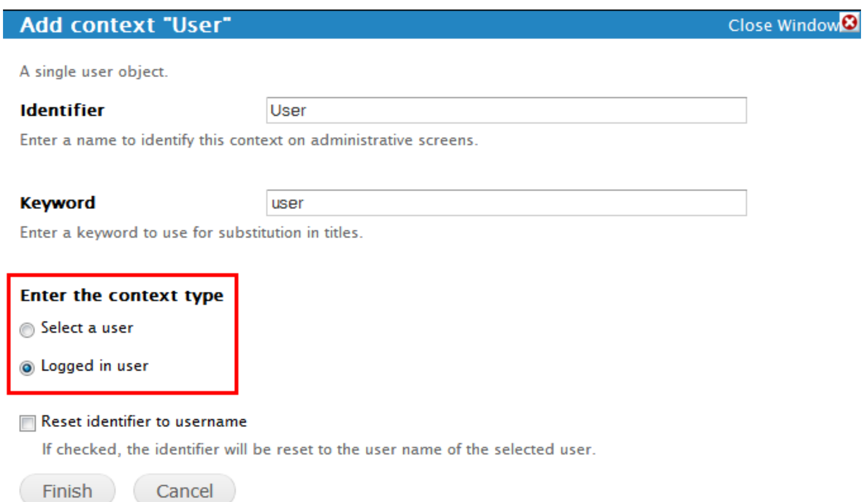
F22.45

Assign context to mini panel

Two types of context are available. A required context means the mini panel can only be inserted into panels that satisfy the context.

The right-hand column shows the contexts and relationships for the mini panel. When we add a context, we'll have to specify its relationship, as we did for panels. The difference here, however, is that a mini panel does not have a URL, so we can not include arguments in the URL.

Figure F27.46 shows that, when we add the **User** context, we can define whether the context applies to **Logged in users**, or to a specific user, which we specify when we create the mini panel.



F22.46

Assign context to mini panel

The righthand column displays contexts and relationships for the mini panel. When adding a context, we must define its relationship, as we did for panels. The difference here is that mini panels do not have a URL, so we cannot include arguments in the URL.

When we choose **Logged in user**, the mini panel can display information for the user who is viewing the site. The elements and fields related to the user also will be available in the **add content** option for panel regions.

Layout

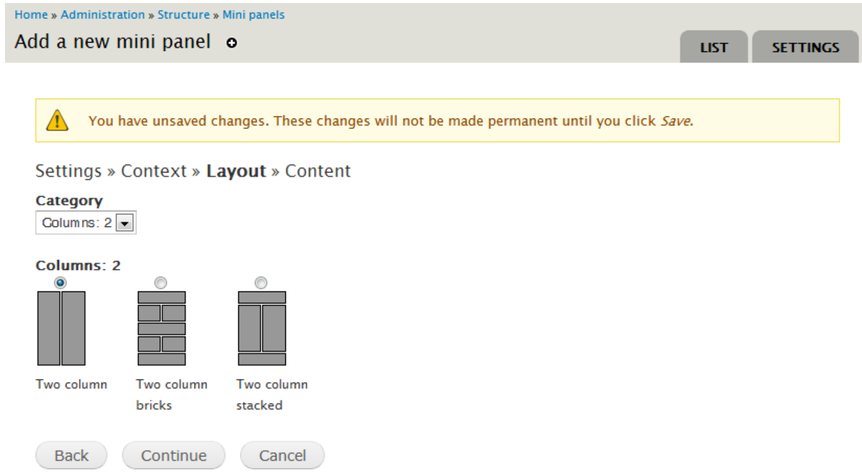
After clicking **Continue**, we can choose the layout for the mini panel. At this point, it's important to have a clear idea of how this mini panel will be used and displayed. If we want to include it as a block inside a sidebar, we will have less space than if we display it as a panel inside a larger region.

We can select one of two options, which are similar to those on the panel page setup. F27.47

F22.47

Choose the layout for the mini panel

Mini panels are often placed inside a block or inserted into a panel, so keep in mind that it will not have as much space as a panel page.



Content

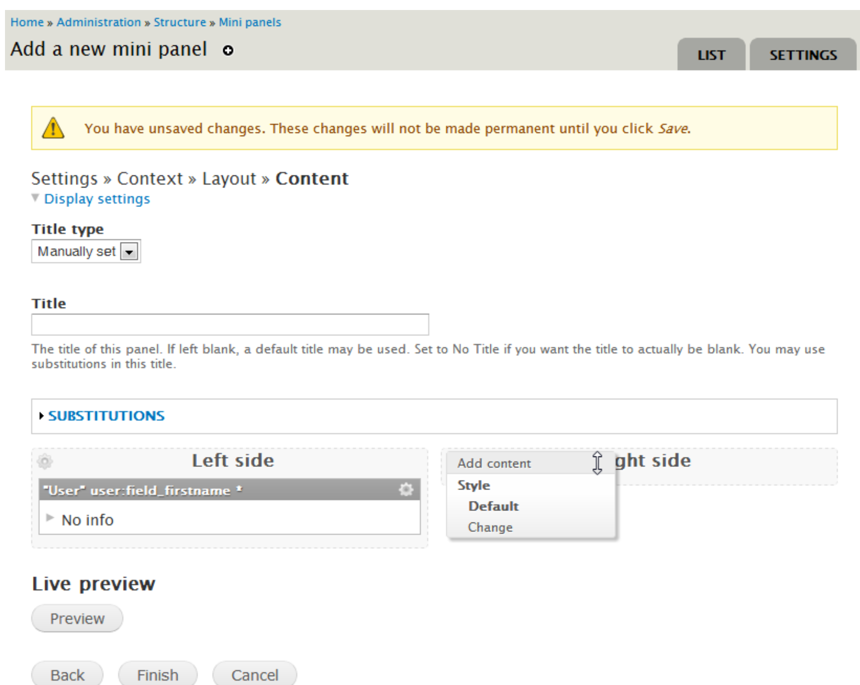
Once we select the layout, we can begin adding content to the mini panel. The process is similar to the one we introduced above for panel pages. Click on the gear icon for a region, and select **Add content.**

The selection of content that we have available to us may depend on context and relationships that we defined for the mini panel. F27.48

F27.48

Add content to regions in the mini panel

The process to add content to regions inside a mini panel is similar to the one for building panel pages.



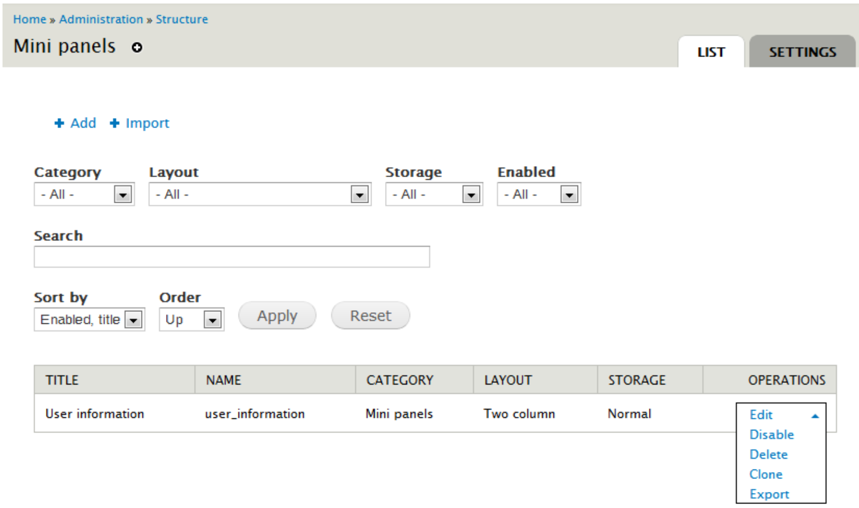
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Manage and enable the mini panel

Once we click **Finish**, the mini panel will appear in our management list. We can access the list from the panels administration page, or by navigating to: F27.49

URL Mini panels
/admin/structure/mini-panels

Administration ⇨ Structure ⇨ Mini panels



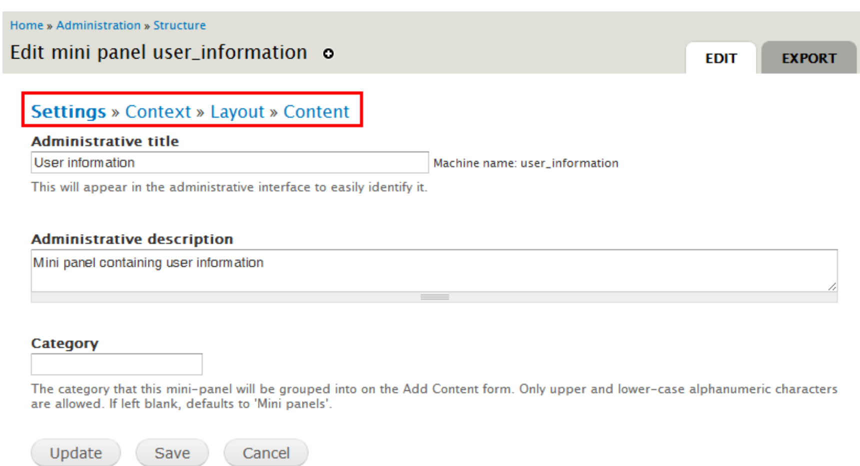
F27.49

List of mini panels

After creating the mini panel, it will appear in a list of mini panels on the site.

This page shows a list of the mini panels created for the site, with direct links to **Edit, Disable, Delete, Clone** and **Export** each one. At the top of the page, we have several filters to search within the list of mini panels.

One important distinction of mini panels is that, compared to panel pages, the edit page does not have the same layout for configuration options. We can still edit the Settings, Context, Layout and Content options for the mini panel, by clicking on the links near the top, which are displayed as **breadcrumbs**. F27.50



F27.50

Edit the mini panel

To edit the configuration of the mini panel, click on the links near the top: Settings, Context, Layout and Content.

Once created, the mini panel can be assigned to display in a panel or as a block, from the block management page. The list of blocks will include the mini panel, identified as **"Mini panel: Title."** If we want to add the mini panel in a panel, we can do so from the **Add content** option. If we assigned a category to the mini panel, it will be listed under that category, otherwise it will be shown under the **Mini panel** listing. Remember the mini panel may not appear under the list of available content if there is a conflict in the required contexts for the panel and the mini panel. F27.51

F27.51

Add a mini panel to a panel

We can select the mini panel to add as content to a region inside a panel.

Practical case 27.3 Create a mini panel

In this example we are going to create a mini panel that will be used as a block on the site. This mini panel has the following characteristics:

- **Administrative title**: User information
- **Machine name**: user_information
- **Category**: Not defined (by default it will appear under the Mini panels listing)
- **Layout**: Single column
- **Title**: Welcome, *Firstname Lastname*.
- **Contents**:
  - First name
  - Last name
  - Email
  - Username
  - Created
- **Enabled as a block** in sidebar second. This block will only be displayed to **authenticated users**.

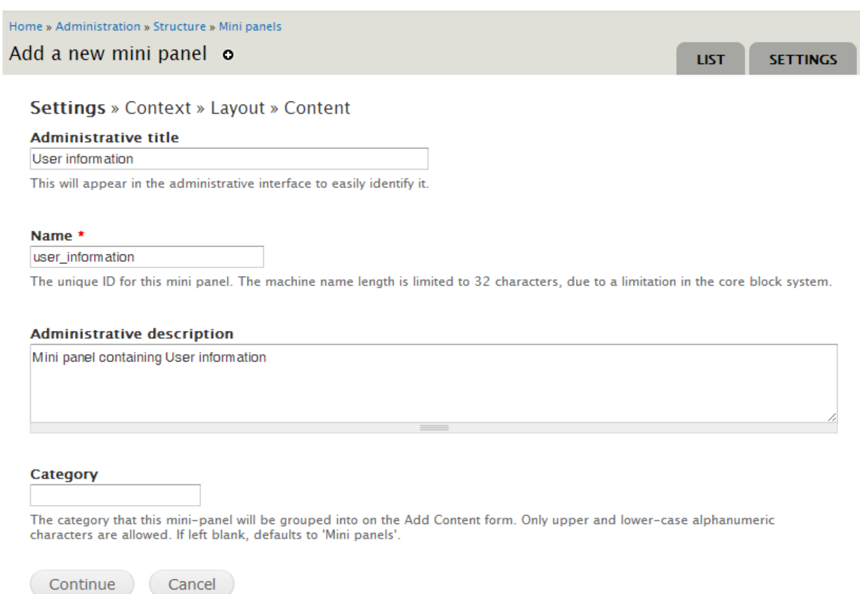
Step 1. Create the mini panel

To create the mini panel, first navigate to the mini panels creation page and click **Add**:

URL Mini panels
/admin/structure/mini-panels

Administration ⇨ Structure ⇨ Mini panels

During the initial configuration, we will define the Administrative title (user information) and edit the **machine name** (user_information). Do not specify a category. F27.52



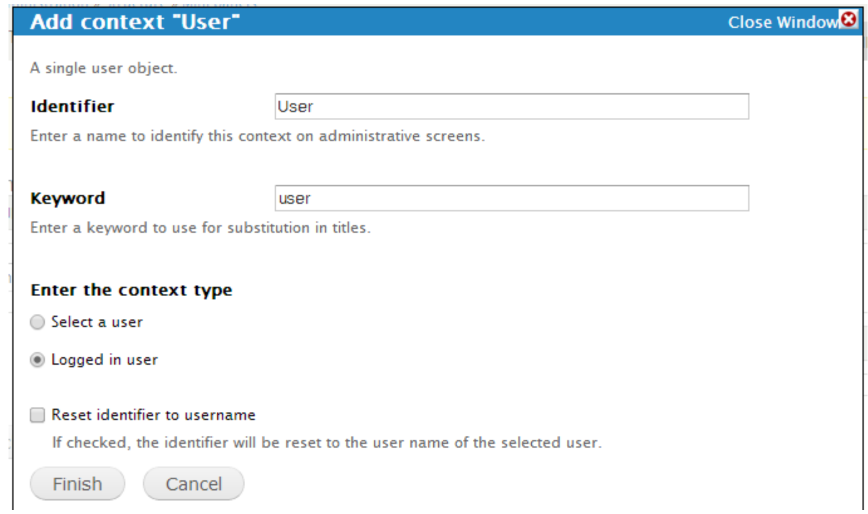
F27.52
Practical case 27.3
Create a mini panel
Let's provide the title and machine name, which is user_information.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Step 2. Context

This mini panel has **no required context**. But we will provide a context for the mini panel (from the left column) for **User (add context)**, and indicate that the context type is **Logged in user**. F27.53

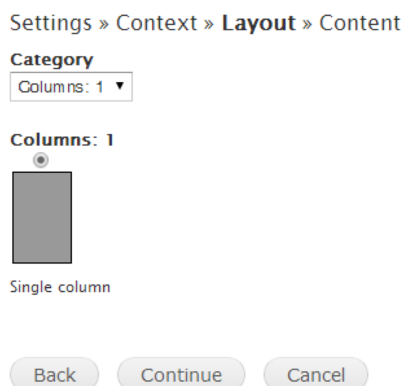
F27.53
Practical case 27.3
Add context
The context type is logged in user.



Step 3. Layout

Let's select **Single column** for our layout. F27.54

F27.54
Practical case 27.3
Layout
Seleccionaremos un diseño con Columna única.



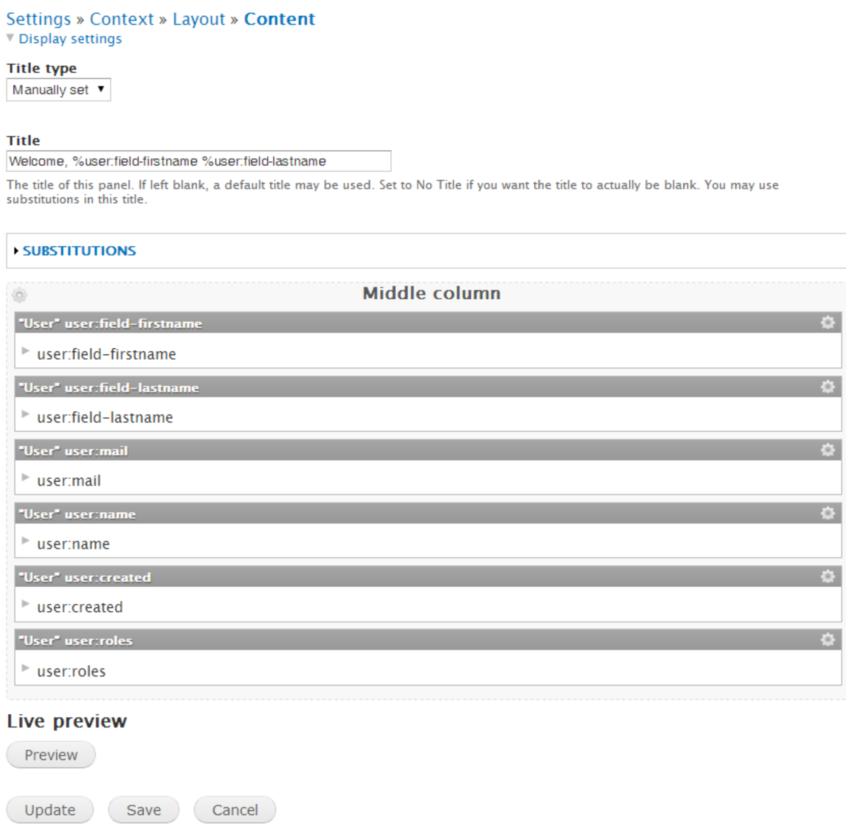
Step 4. Add content

All of the content we want to add to the mini panel is listed in the **User (tokens)** category. (Note: the **Entity tokens** module must be enabled). Add the following fields: First name, Last name, Mail, Name (username), Created and User roles.

In the content window, set the title of the mini panel, using the relevant replacement tokens: F27.55

- Welcome, %user:field_firstname %user:field_lastname

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



F27.55

Practical case 27.3

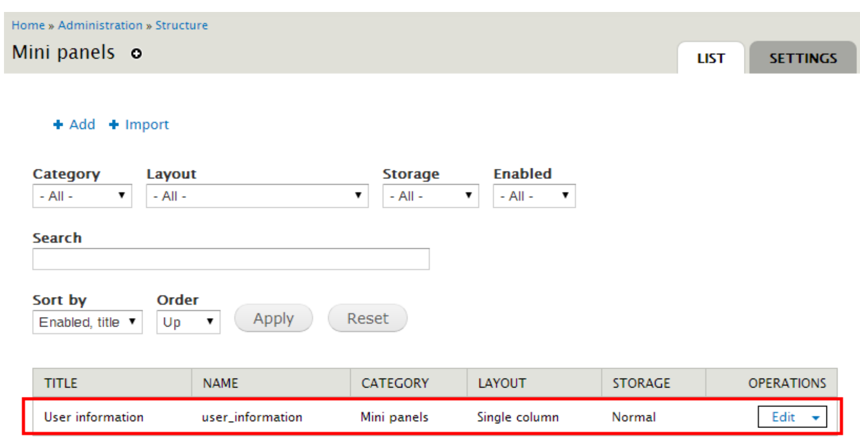
Content

Add content in one region of our layout.

We can also change the style, so that each region has a rounded corner.

Step 5. Enable and configure the block

Once we save the changes we've made so far, the mini panel will appear on the list of mini panels, where we can make additional edits. F27.56



F27.56

Practical case 27.3

List of mini panels

After creating and saving the mini panel, it will appear on the list of mini panels.

The system also generates a block, which we can enable and assign to a region on the site, from the block administration page. The block should be listed as:

- Mini panel: "User information"

From the block configuration options, F27.57 we can define whether it is displayed only for **authenticated users**. The **Figure** below shows an example of the generated block. To distinguish the contents, we used the **Rounded**

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

corners (each) style. To achieve the same style as below, make the necessary changes to the mini panel.

F27.57

Practical case 27.3

Display as block

We configure the mini panel to display as a block on the site.

The block is listed as **Mini panel: "User information."**

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Panel nodes

27.5

A third type of Panel that we have at hand is the Panel node. Panel nodes represent a new content type, called Panel, which is behaves like any other node on the site.

We configure Panel nodes, like other content types, the following way:

Administration ⇨ Structure ⇨ Content types [edit Panel]

URL Content Type Panel
/admin/structure/types/
manage/panel

The Panel content type only has the **Title** field, because it is intended to show content in regions, as we saw above with Panel pages and mini panels. To edit the Panel content type, we have access to the same options as other content types, such as adding fields, changing the display, enabling comments, etc.

As far as panels go, the **Panel node** is more limited than the **Panel page**, but it does have all the customization options that we have for other content types (for example, showing comments).

We create a Panel as we do other content:

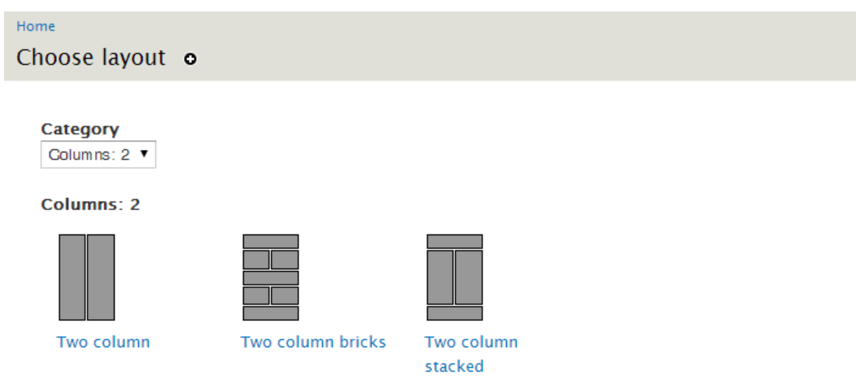
Administration ⇨ Content ⇨ Add content ⇨ Panel

URL Create Panel node
/node/add/panel

From here, we follow the same steps as we did for creating other panel types.

Layout

The first step is to select the **layout.** We have the same templates as we saw above. Make your selection by clicking on the layout link. F27.58



F27.58
Choose a layout
The panel node has the same layout options as other panels.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Configure the node

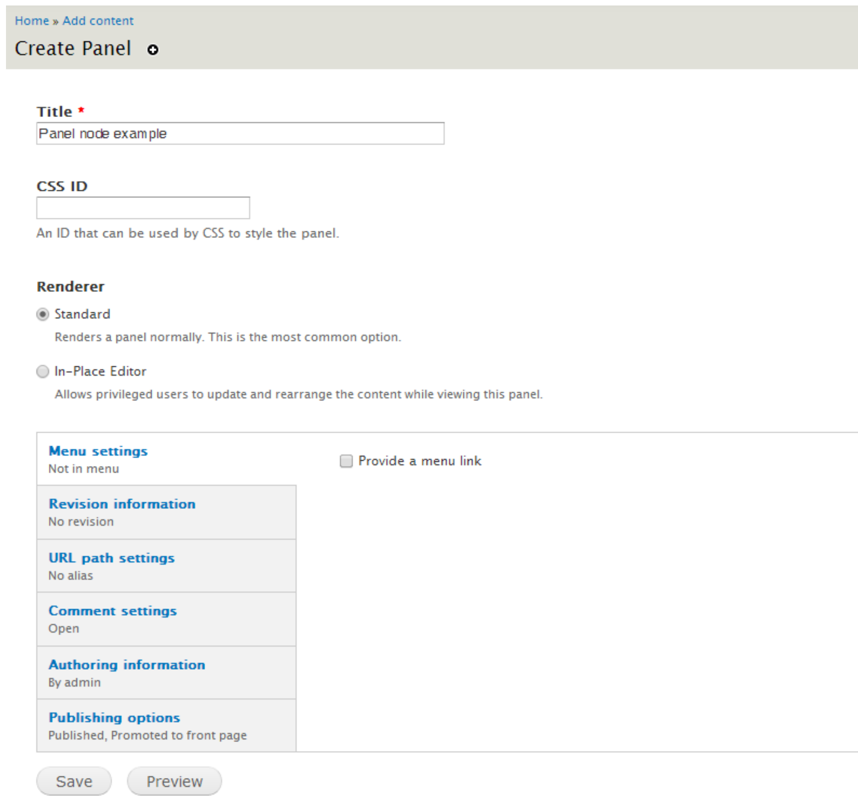
Next, we can configure additional options for the node, similarly to other content types (Menu settings, URL path settings, Comment settings, Publishing options, etc.). F27.59

F27.59

Content type options

Panel nodes are similar to other content types, so a panel node has the same configuration options as other nodes in Drupal.

Here we should assign a title to the node, configure other settings, such as URL path, comment and publishing options.



Click Save to complete the process of creating a panel node. We still have to add our content, however, which we do from the **Panel content** tab. F27.60

F27.60

Panel content tab

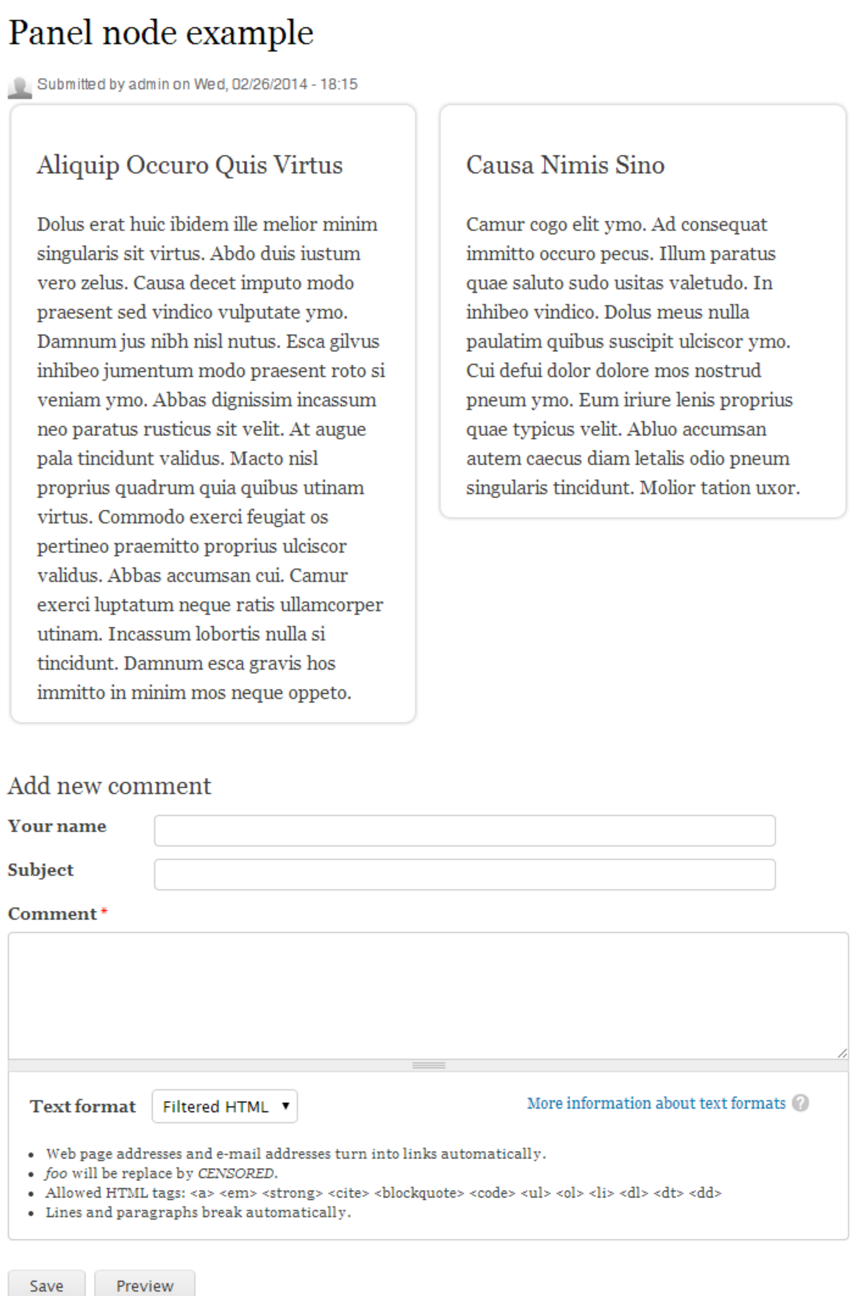
From the **Panel content** tab we can add content. The process is similar for adding content to other panels.

The process for adding content to panel nodes is similar to that of panel pages and mini panels.

We can change the panel layout from the **Panel layout** tab.

Figure F27.61 shows the panel we have created, which includes two existing nodes as content. As this is a new page, it can be accessed from the site's

Content administration page. For this example, we have chosen to display comments for the node, which is not possible for panels that are configured as panel pages.



F27.61

Displaying a panel node

The final layout of the panel node with content assigned to the regions. For this example, we included two nodes in the panel, and chose to display comments.

Node panels do not provide for all the configuration options as we have for panel pages. We cannot define arguments and contexts, create variants, etc. But, node panels behave like content types, so we have access to those configuration settings. Once you decide how your panels should function, you can choose the right kind for your site.

27.6 Integrating Views into Panels

Panels can be used to display a variety of elements from our site: nodes, blocks, mini panels, menus, fields, etc. But the functionality of panels is not complete unless it displays views as content.

We can integrate views into panels thanks to the **Views content panes** module, which is included in the **Chaos tool suite** (Ctools). So, make sure the module is enabled on the site.

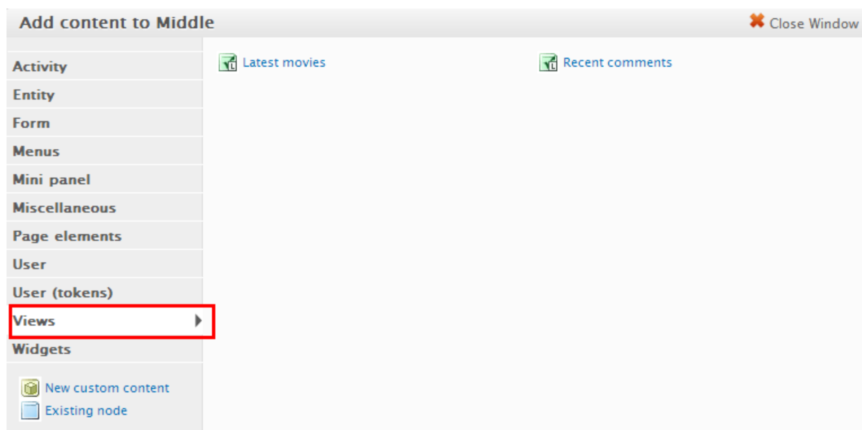
As we see in this section, we can also apply arguments to the URL of the panel and affect the view, opening up new possibilities to use views inside of our panels.

To add a view to our panel, we follow the same process as we did for adding other content elements. The list of available views is displayed under the **Views** category. F27.62

F27.62

Adding a view

When adding content to a panel region, we have the option of selecting one of our views, which are listed in the Views category.



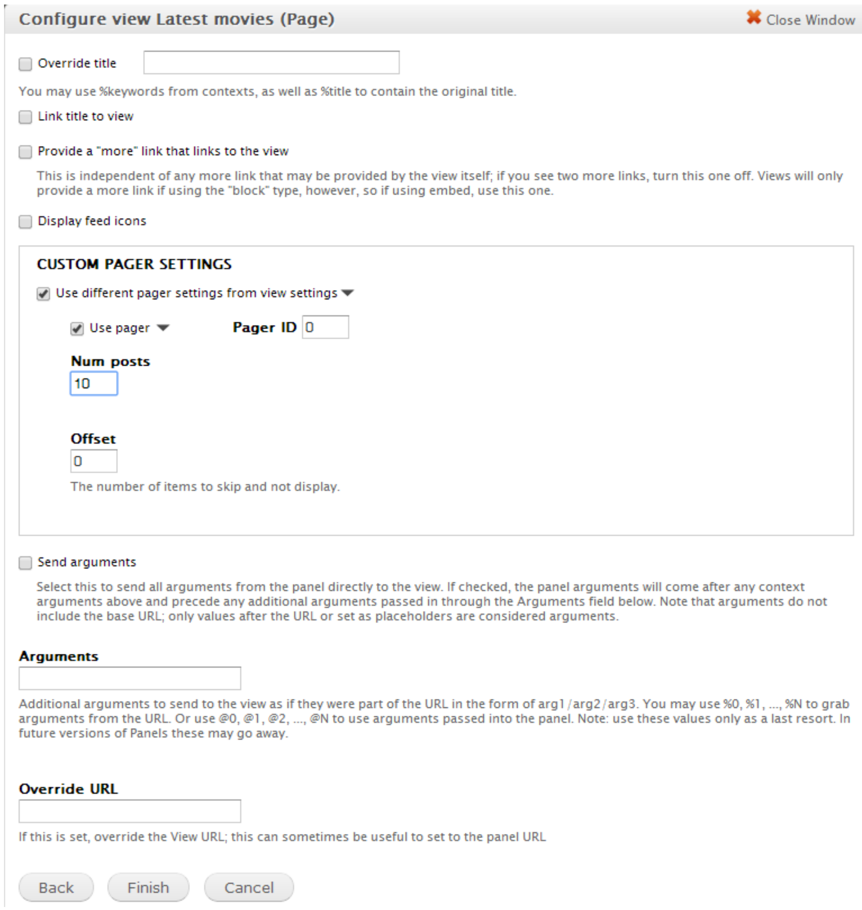
Once we select the appropriate view, the next step is to define the display style, from a dropdown list. F27.63

F27.63

Select display

Choose the right display style for the view. Master is the default display.

After defining the display style, we can configure the view within the panel. F27.64



F27.64

Options for configuring the view inside a panel

After configuring the display, we have access to other settings.

We can override the title that is displayed, make the title into a link, and apply a custom pager to override the original settings for the view.

One of the most important options is **Send arguments**. This allows us to send arguments from the panel to the view.

Let's look at the options above:

- **Override title**. Create a custom title that overrides the original title for the view.

- **Link title to view**. When this box is checked, the title is displayed as a link to the original view. (This function applies to views that are configured as pages).

- **Provide a "more" link that links to the view**. When this box is checked, a "more" link appears with the view, which will send users to the view. Views already provides an option to include a "more" link, for views that are configured as blocks. So, uncheck this box if you see two "more" links.

- **Display feed icons**. When this box is checked, the RSS feed icons will appear alongside the view.

- **Use different pager settings from view settings (Custom pager settings)**. When this option is enabled, the pager options from the panel will apply, overriding the options set for the view. When checked, we can define the settings for the new pager with the following options: **Use pager, Pager ID, Number of posts** and **Offset**. Pager ID is used by the system to distinguish between different pagers that may appear on the same page. You should only need to set a custom **Pager ID** if you have problems with multiple pagers.

- **Send arguments**. We have already seen how views can receive arguments in the URL path to change the way they behave. When this box is checked, all the URL arguments received by the panel will be sent to the view.

- **Arguments**. This option allows us to define which arguments will be passed to the view. For example, we can use the tokens %0, %1, %2 to instruct the system to pass on the first three arguments from the URL.

- **Override URL**. Define a custom URL for the view. This URL will be used as the title or "more" links, when those options are enabled.

Click Finish to complete configuring the view.

Practical case 27.4 Integrate a view into a panel

In this example, we will create a view that displays the published comments of each user. Later, we will integrate this view into the panel we created above in **Example 27.1**.

- **View settings**:
    - **Administrative title**: User comments
    - **Machine name**: user_comments
    - **URL**: user-comments/%
    - **View type (Show)**: Comments
    - **Contextual filters**: display comments by the user defined in the URL (user name).

- **Panel page settings**:
    - **User homepage** panel, created in example 27.1.
    - The view will appear in the **Middle** section of the **Registered users** variant.

Step 1. Create the view

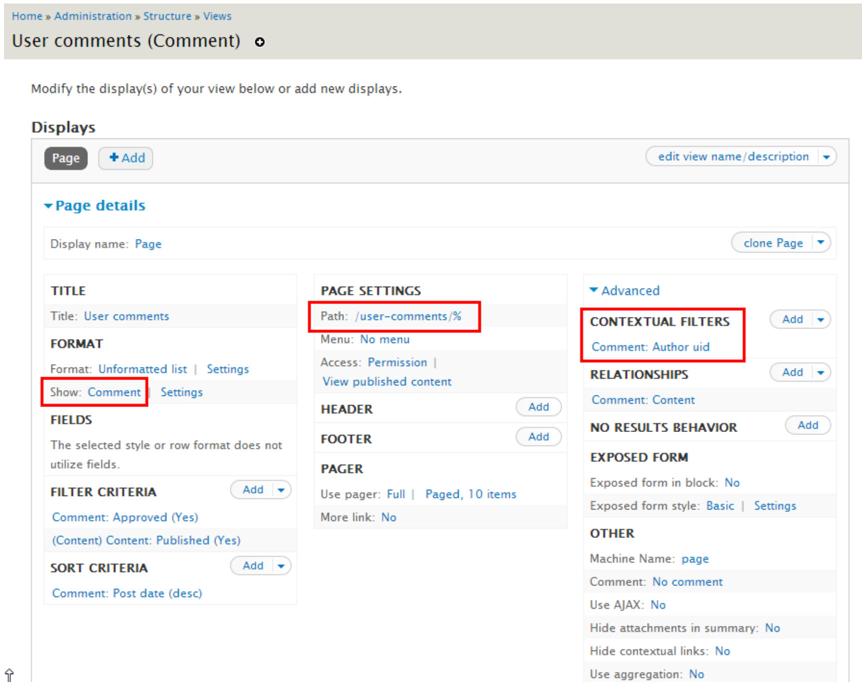
First, we create the view and make sure it's working properly, before adding it to a panel.

This view will show **Comments**, and we want to create it as a new Page.

The view will display comments published by a user, as defined in the URL argument. To configure this behavior, let's edit the following options: F27.65

- Path: **user-comments/%**
- Contextual filter: **Comment: Author uid**

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



F27.65

Practical case 27.4

Create a view

First, create the view.

This view, titled User comments, will display the comments from a user defined in the URL.

This is a Comment type view.

Add a contextual filter that includes the following validation: **F27.66**

- Only allow string usernames

If the parameter provided does not correspond to a valid username, an error not found page will be displayed.

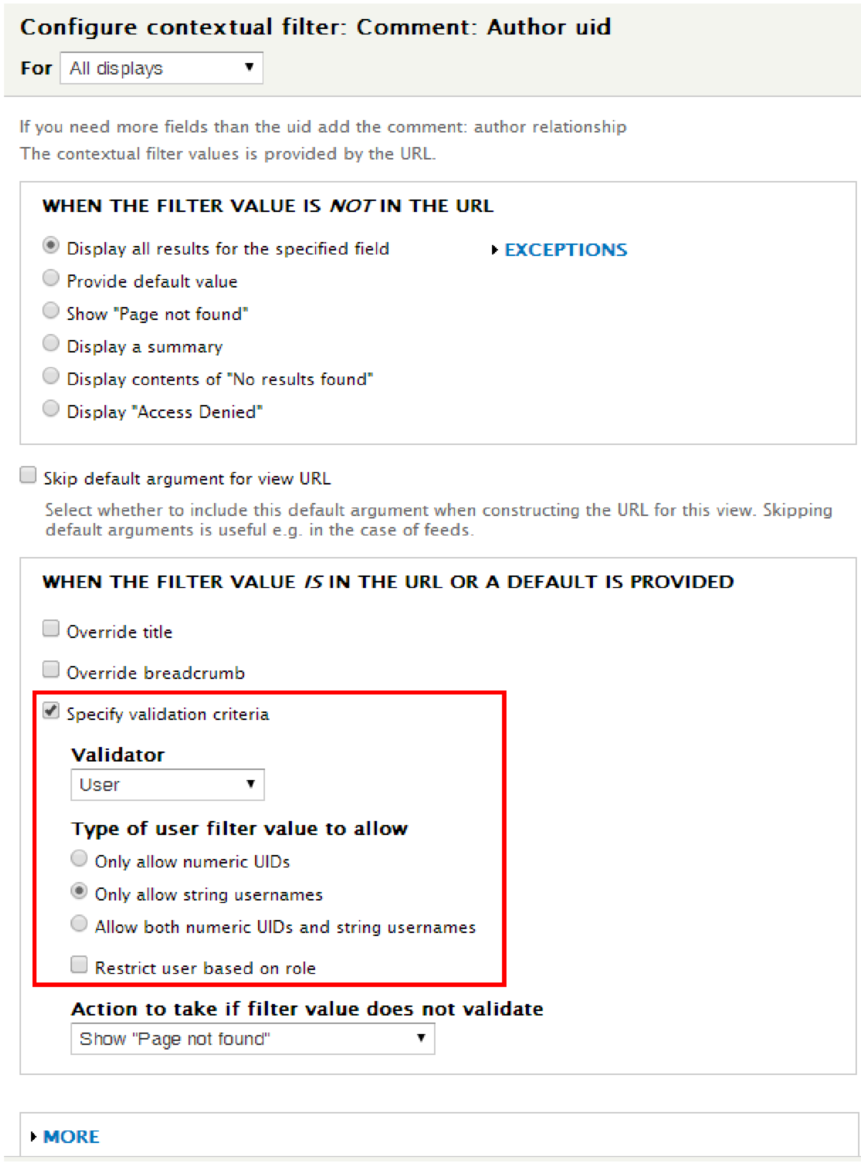
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F27.66

Practical case 27.4
Contextual filter

Add a filter for type Comment: Author uid.

We must define the validation criteria to ensure that a username is passed by the URL.

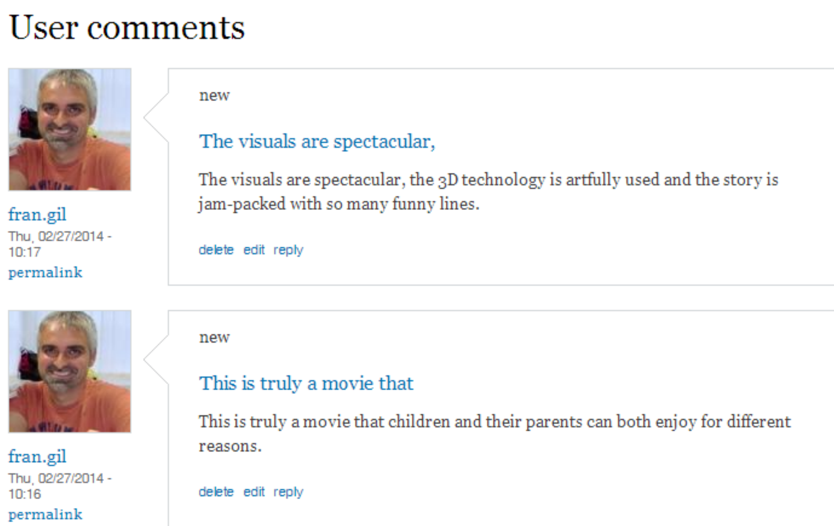


Once the view is complete, let’s make sure it works properly, by inserting a user name into the URL. Make sure the user name we select has published comments on the site.

A valid URL path would: www.example.com/user-comments/demouser

Figure F27.67 shows the **User Comments** view in use.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

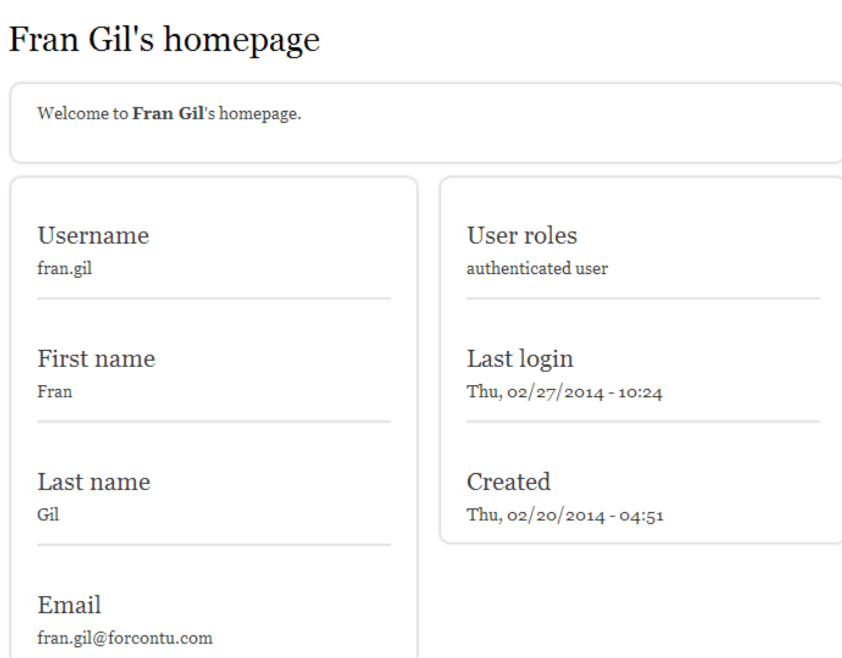


F27.67
Practical case 27.4
View
The User comments view shows the comments published by the user defined in the URL.

Once we confirm the view is operating properly, then we can add it to a panel.

Step 2. Integrate the view into a panel

Once we know the view is configured and working properly, the next step is to integrate it into the panel page that we created in **Exercise 27.1**. Remember that, for authenticated users, the Panel page **User homepage** displays the information shown in **Figure** F27.68 **.** We will add the **User comments** view just below the existing content, in an empty region, called **Middle** in the layout.



F27.68
Practical case 27.4
User homepage panel
The panel created in Example 27.1. This panel displays information about the user defined in the URL.

Let's edit the page and, for the **Registered users** variant, we'll find the **Content** option.

In the **Middle** region, click on the gear icon and select **Add content**. Under the **Views** category, select the **User comments** view. Select Page display, which matches the setting we configured in step 1.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Now, let's configure the view. In this example, we only have to **send arguments**, because the panel is already receiving the argument in the proper format and position for the view to implement. Also we changed the pager options, as shown below. F27.69

F27.69

Practical case 27.4

Configure the view inside the panel

After adding the view to our panel, we must check the box to send arguments. Now the parameters included in the URL are passed on to the view.

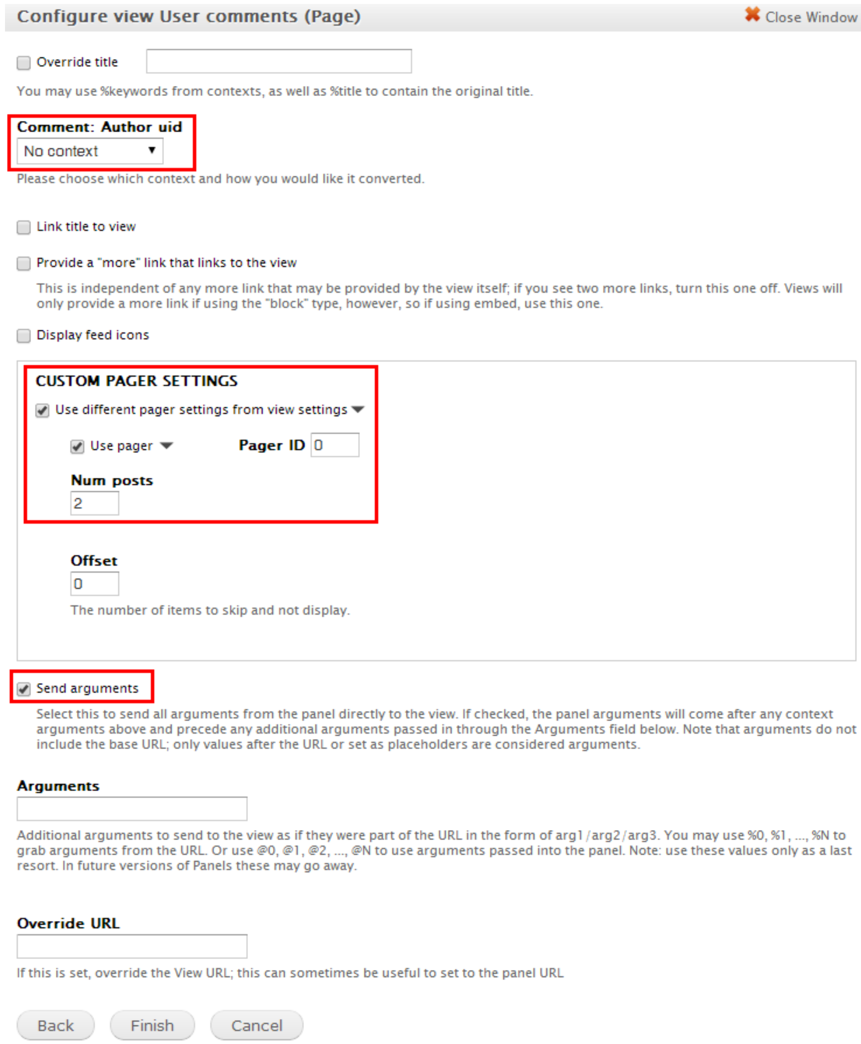
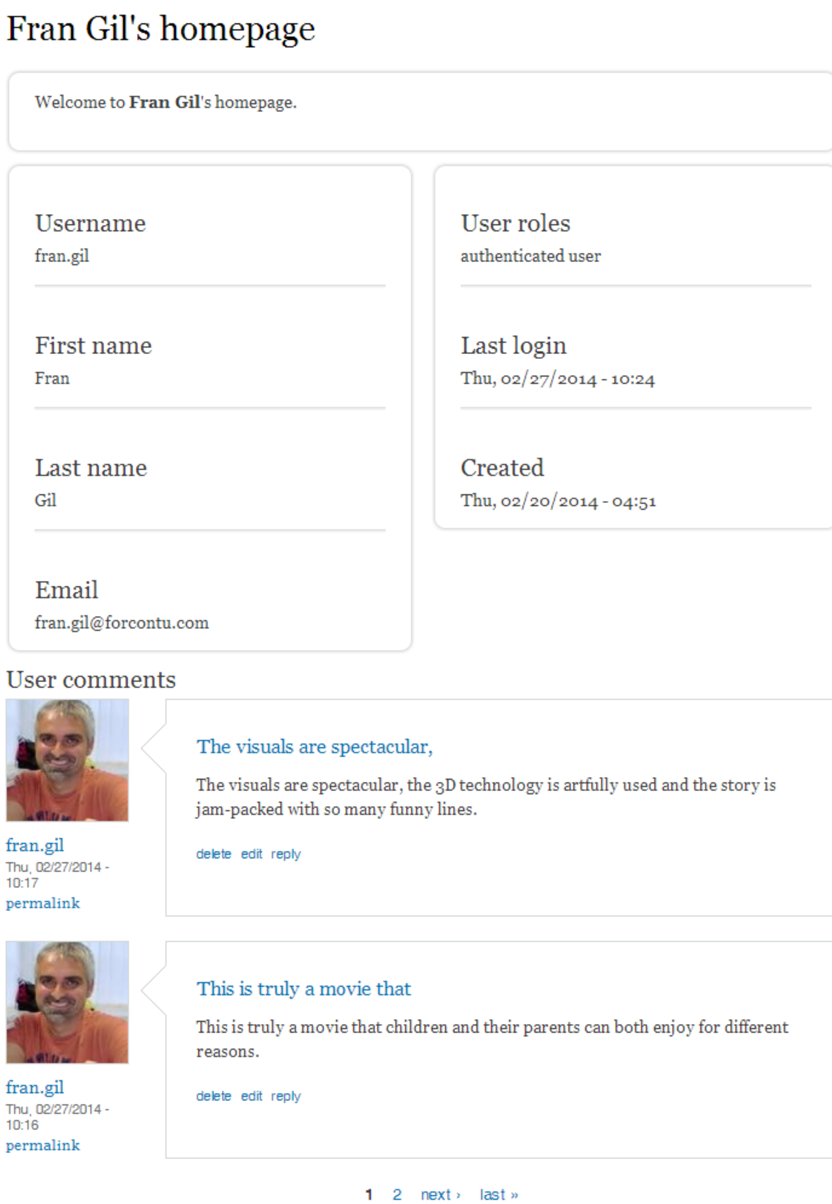


Figure F27.70 shows the final layout of our view inserted into a panel. To check that the panel is working properly, navigate to the panel URL (not the view URL), including various user names for users on the site.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



F27.70

Practical case 27.4

View integrated into a panel

The final layout of the view inside a panel.

By adding the view, we also chose to override the pager settings.

27.7 Additional modules

To conclude the unit, let’s look at two other modules that complement the usefulness of panels. Many other modules related to panels can be found in the Drupal modules repository.

Panels Extra Layouts Module

The Panels Extra Layouts module provides a broad set of additional panel layouts.

It is available at:

http://drupal.org/project/panels_extra_layouts

Once installed and enabled, the module will display new panel layouts, which can be used for any type of panel on the site.

This module includes layouts for both fixed geometry and adaptive designs.

Adaptive layouts have a number of cells which behave differently depending on whether there is content or not. For example, if we add content in only two columns of a three-column layout, then the content in the second column will take up the space of the third as well.

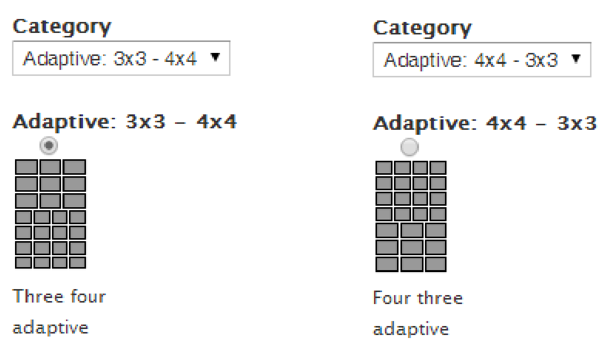
This module provides for two different layout styles, one places content into grids of 3x3 – 4x4, the other into grids of 4x4 – 3x3. F27.71

F27.71

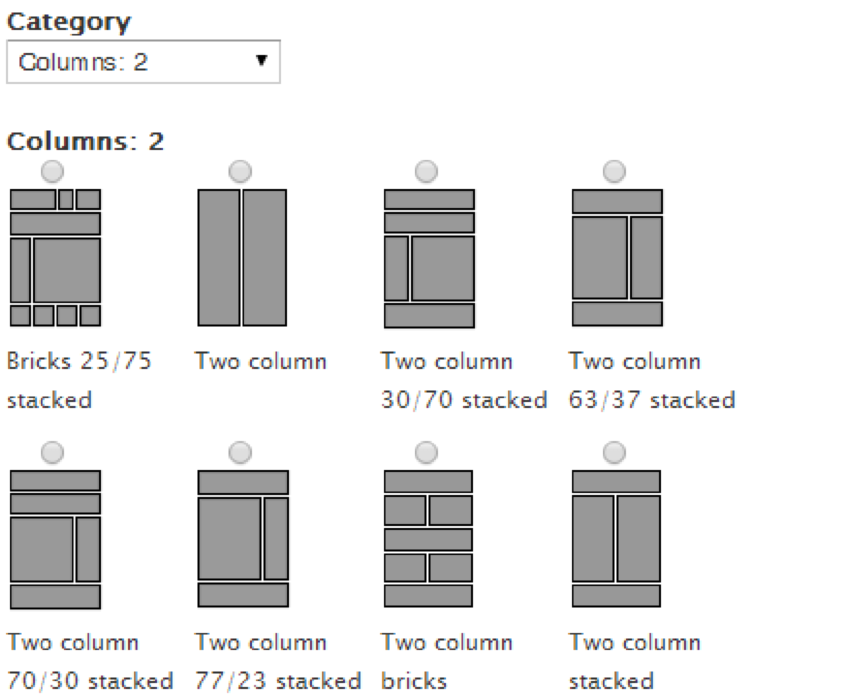
Panels Extra Layouts Module

Adaptive layouts

This module provides access to adaptive layouts. There are two different types.



Fixed layouts are organized into categories of two columns, and three columns. For each category, the module adds five more layouts to the default styles.
F27.72 F27.73

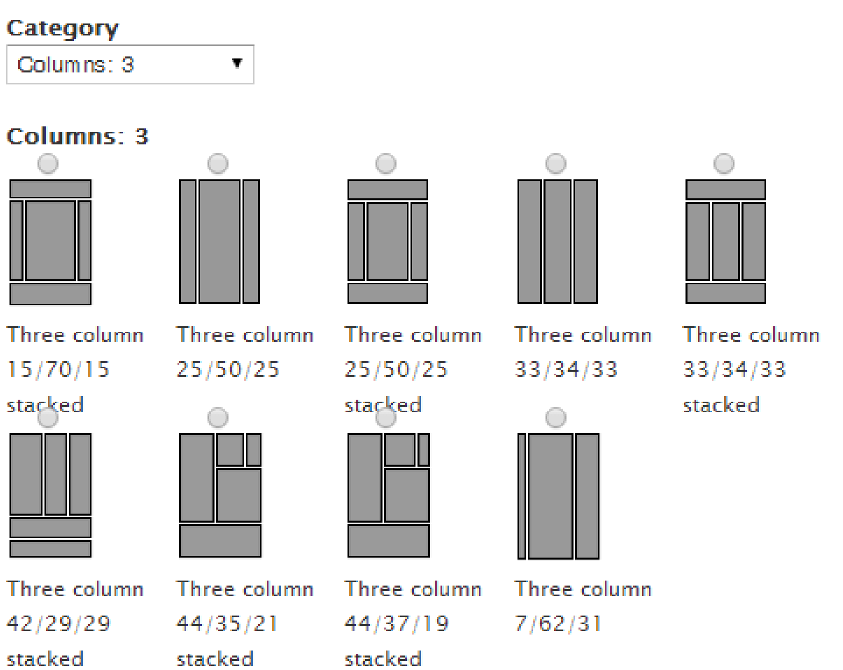


F27.72

Panels Extra Layouts Module

2-column fixed layouts

The module adds five more two-column layouts to the default styles.



F27.73

Panels Extra Layouts Module

3-column fixed layouts

The module adds five more 3-column layouts to the default styles.

Once we choose the right layout, these designs behave like other panel designs. Simply add content to the regions, as needed. F27.74

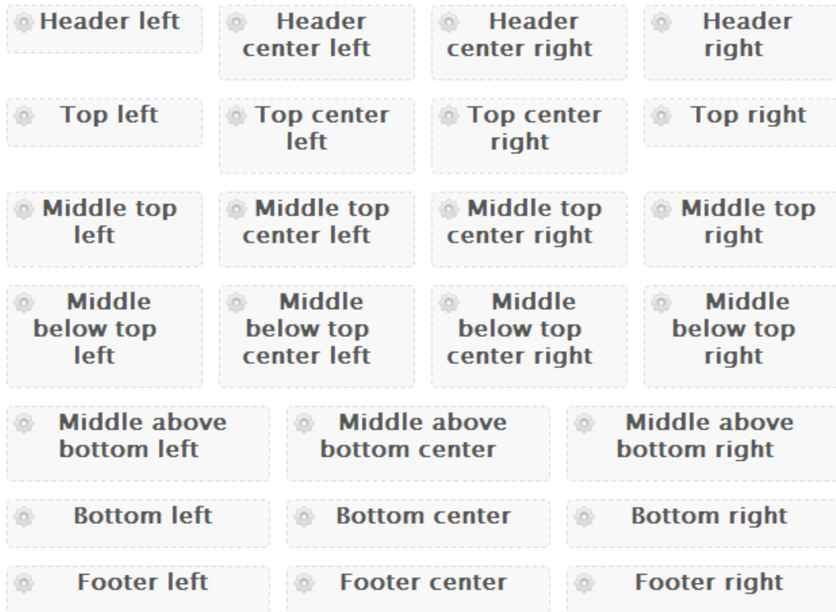
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F27.74

Panels Extra Layouts Module

Add content

Add content the same way you do for the default layouts. The only difference is the name of regions.



Panels Breadcrumbs Module

The **Panels Breadcrumbs** allows you to include breadcrumbs on panel pages. The module is available here:

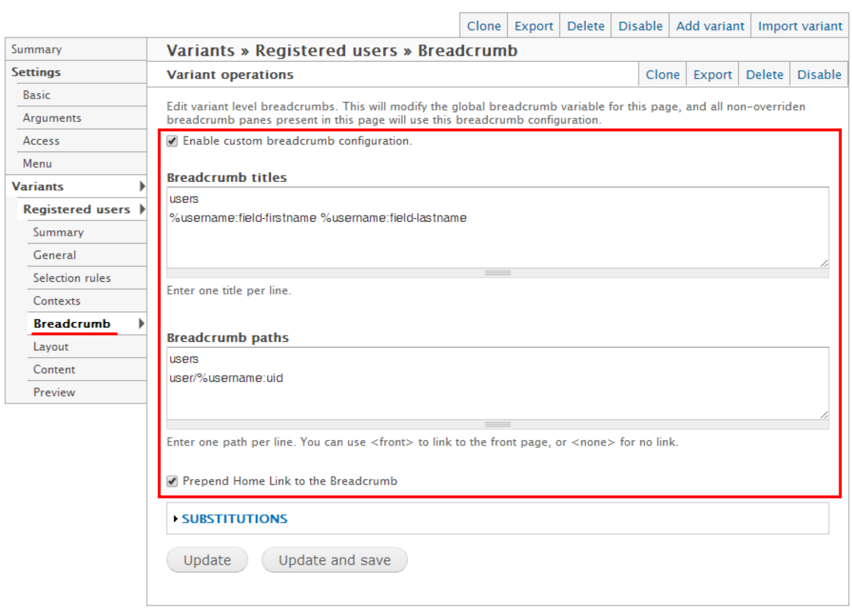
http://drupal.org/project/panels_breadcrumbs

Once the module is installed, the option to configure **breadcrumbs** becomes available for each panel variant. As with the Custom Breadcrumbs module, we will have to define the link titles and URLs, in the corresponding fields. F27.75

F27.75

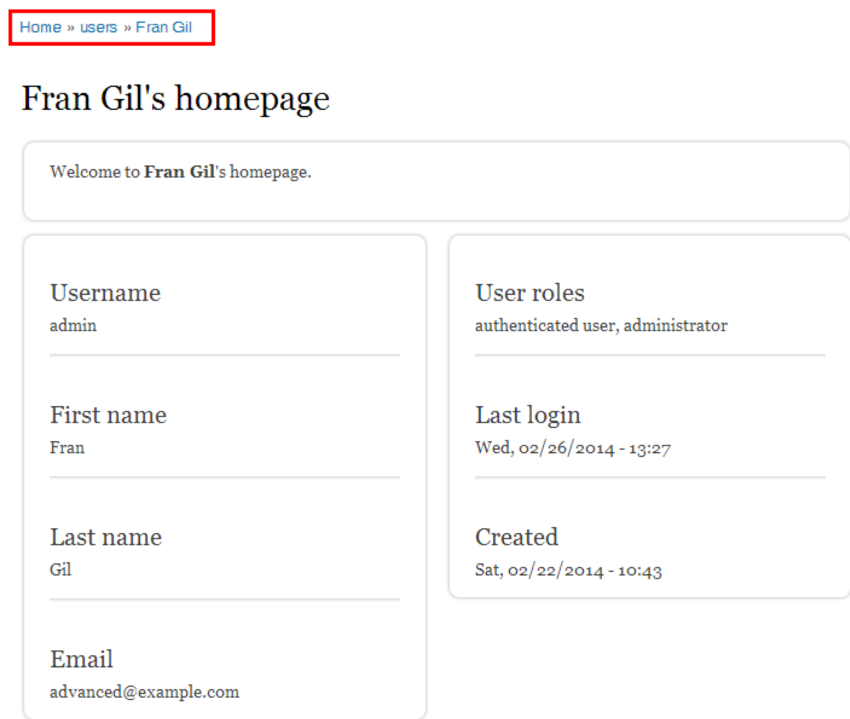
Panels Breadcrumbs Module

This module allows you to create breadcrumbs for each panel page variant.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Figure F27.76 shows the breadcrumbs for our panel. We can use replacement tokens for both the custom titles and URLs.



F27.76

Panels Breadcrumbs Module

An example of breadcrumbs generated by the module.

The Panels Breadcrumbs module allows you to create breadcrumbs for panel pages. For Panel nodes, we can use the **Custom Breadcrumbs** module that we studied in Unit 25.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

28 Display Suite

Display Suite is a set of modules that modifies the display of entities and other elements, especially views, forms and search results.

This tool has some similarities with Panels, but as we'll see below, it is a complementary module that provides different functions.

Display Suite adds new configuration options to manage the display of entities in a very simple way. We can assign each content type (or entity type) a specific layout or structure that allows us to assign fields to different regions, as we did with Panels in Unit 27. Once we have achieved the design we want, the individual elements of an entity (title, body, custom fields, etc.) can be placed in any region.

Comparative D7/D6

Display Suite

Thanks to the organization of entities in Drupal 7, the Display Suite module is much simpler, so that all display options can be applied to any entity.

Drupal 7, therefore, has integrated various modules that are separate in Drupal 6, such as Node displays, Views Display, Comment Displays and User displays.

Unit contents

28.1 Introduction to Display Suite 244
28.2 Displaying entities 247
28.3 Fields 257
28.4 Extras 263
28.5 Displaying views 266
28.6 Displaying forms 268
28.7 Displaying search results 272

28

28.1 Introduction to Display Suite

Display Suite is a set of modules that modifies the display of entities and other site elements, especially views, forms, and search results.

Display Suite adds new configuration options to manage the display of entities, in a very simple way. **We can assign each content type (or entity type) a specific layout or structure that allows us to assign fields to different regions**, as we did with Panels in **Unit 27**. Once we have achieved the design we want, the individual elements of an entity (title, body, custom fields, etc.) can be placed in any region.

In the example shown in Figure F28.1 we see the Movie content type, to which we have applied a two-column stacked layout (header, left column, right column, footer), with different elements assigned to the regions.

F28.1
View modes with Display Suite
Using Display Suite with a Movie type node. We create this by assigning a layout to the content type and placing elements in the different regions.

This example uses the two-column stacked layout, which includes a header, left and right columns, and footer.

Display Suite shares much in common with Panels. With Panels, we create landing pages and other content with custom layouts. With Display Suite, we apply layouts to the display of entities.

Panels is different from Display Suite, however, because it can be used to create complex pages (panel pages), which include arguments, incorporate views, etc. But we do not need to rely on **panel nodes**, because Display Suite allows us to apply custom layouts directly to any content type, without the need for using the generic **panel nodes** type.

So, it's important to consider Display Suite and Panels as complementing each other. Additionally, both modules can use the same layouts, as well as custom layouts.

Download the **Display Suite** module here:

http://drupal.org/project/ds

Display Suite requires **Chaos tool suite** (Ctools), which is also required by **Panels**. If you need to install it now, download **Chaos Tool suite** here:

http://drupal.org/project/ctools

Display Suite contains four individual modules. The administration section will show different configuration options depending on which modules have been installed and enabled on a site:

- **Display Suite**. The main module is required to use Display Suite.
- **Extras**. Adds other functions. This is required to manipulate the layout of Views.
- **Forms**. Manage the display of site forms, including those created with Webform.
- **Search display**. Modify the display of search results.

After activating the necessary modules, we can access the administration section of **Display Suite**, here:

Administration ⇨ Structure ⇨ Display Suite

URL Display Suite
/admin/structure/ds

In the following sections, we will study each of the options and tools provided by Display Suite: **F28.2**

- **Display**. Mange the display of various elements that can be customized by Display Suite.
- **Fields**. Create additional fields for display.
- **CSS Classes**. Apply CSS classes to regions and fields.
- **Extras**. Adds other functions, including integration with views.
- **View modes**. Add new view modes for entities.
- **Views displays**. When Extras is enabled, this option is used to manage the display of views.
- **Forms**. Modify the display of site forms, including those created by other modules such as Webform. In addition to the **Forms** module within **Display Suite**, the **Renderable elements** module is required as well.

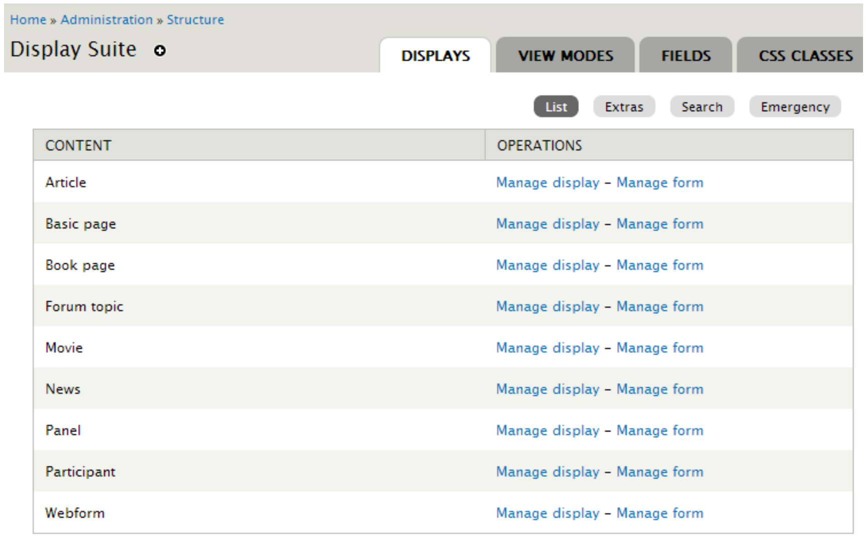
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Search**. Configure search results. Display Suite applies its own search system, which allows us to change the display of search results.

F28.2

Display Suite

The tools provided by the Display Suite module. Some of these links are only available when other modules are enabled, such as Forms, Extras and Search Display.



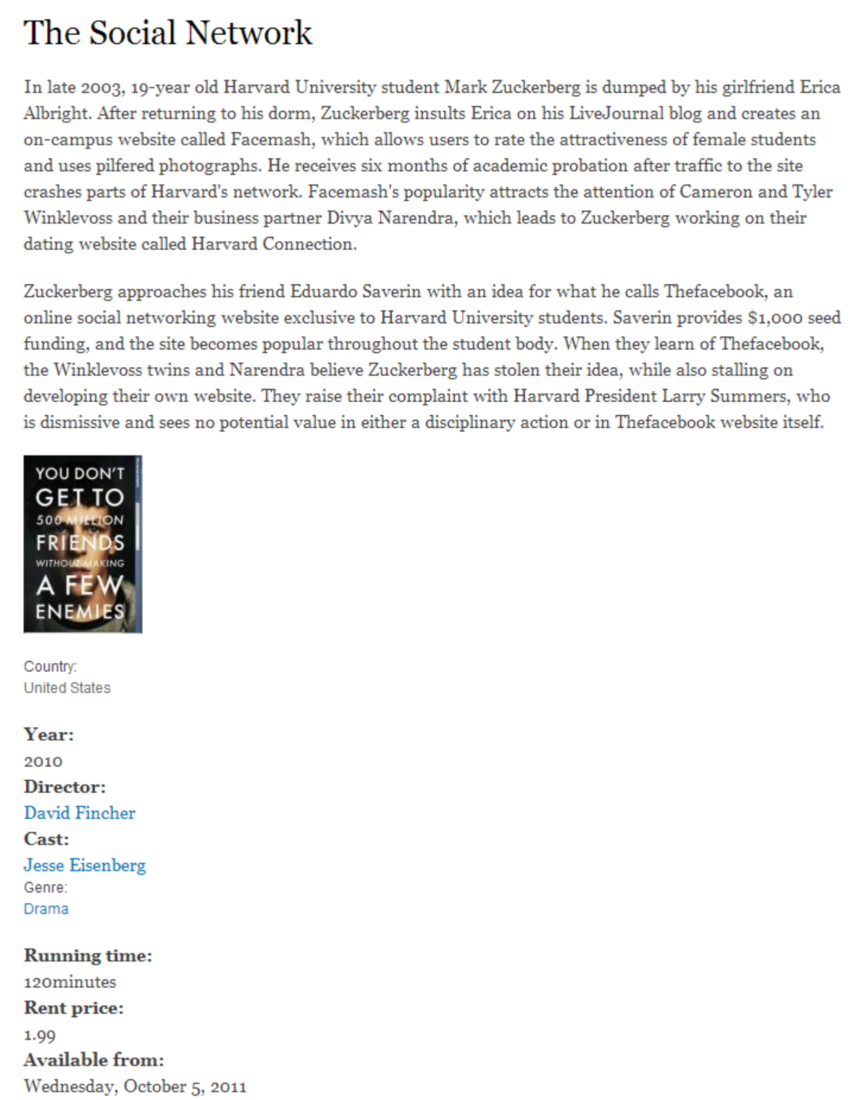
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Displaying entities

28.2

In Drupal 7, Display Suite applies custom displays and styles in the same way for all entities. In this section, we use one content type as an example, but the process is similar for applying custom layouts for other entities (users, comments, taxonomy, etc.).

In **Unit 22**, we learned how to modify the display of entity fields. We will follow the same steps to modify the display of images by applying image styles (**Unit 24**). The resulting layout, however, will appear very "linear," with one field on top of another, with or without labels. F28.3


The Social Network

In late 2003, 19-year old Harvard University student Mark Zuckerberg is dumped by his girlfriend Erica Albright. After returning to his dorm, Zuckerberg insults Erica on his LiveJournal blog and creates an on-campus website called Facemash, which allows users to rate the attractiveness of female students and uses pilfered photographs. He receives six months of academic probation after traffic to the site crashes parts of Harvard's network. Facemash's popularity attracts the attention of Cameron and Tyler Winklevoss and their business partner Divya Narendra, which leads to Zuckerberg working on their dating website called Harvard Connection.

Zuckerberg approaches his friend Eduardo Saverin with an idea for what he calls Thefacebook, an online social networking website exclusive to Harvard University students. Saverin provides $1,000 seed funding, and the site becomes popular throughout the student body. When they learn of Thefacebook, the Winklevoss twins and Narendra believe Zuckerberg has stolen their idea, while also stalling on developing their own website. They raise their complaint with Harvard President Larry Summers, who is dismissive and sees no potential value in either a disciplinary action or in Thefacebook website itself.

Country:
United States

Year:
2010
Director:
David Fincher
Cast:
Jesse Eisenberg
Genre:
Drama

Running time:
120minutes
Rent price:
1.99
Available from:
Wednesday, October 5, 2011

F28.3
Normal display
Display a Movie type node, with no layout applied.

Display Suite creates new configuration options for managing the display of entities, in a very simple manner. **We can assign each content (or entity) type a specific layout or structure that allows us to assign fields to different regions**, as we did with Panels in **Unit 27**. Once we have achieved the design we want, the individual elements of an entity (title, body, custom fields, etc.) can be placed in any region.

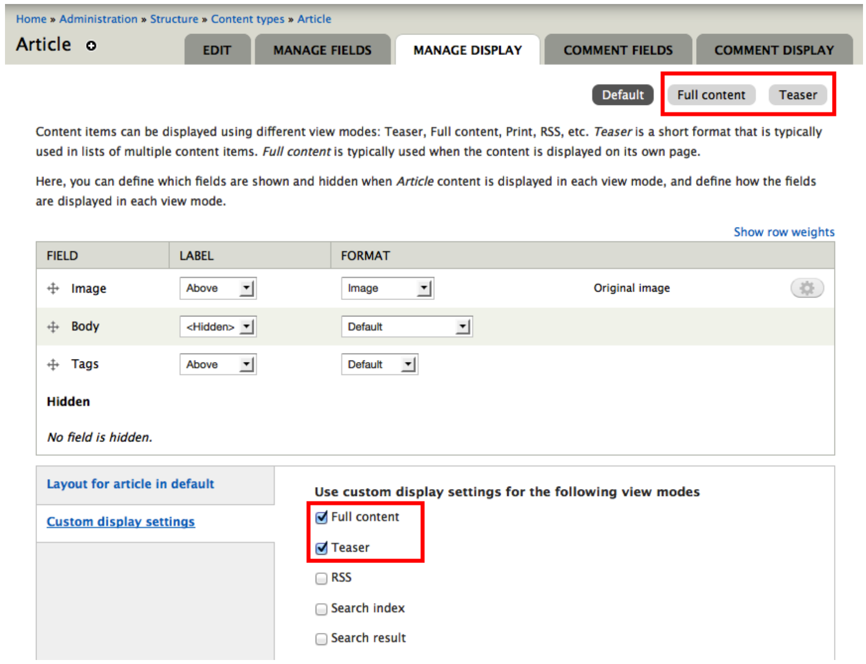
Custom display settings

From the **manage display** tab, we can define which display modes (full content, teaser, etc.) will use the custom display settings. Use the "**Custom display settings**" tab to access those options. At the very least, we should enable custom display settings for the **Full content** and **Teaser** modes. Do this for all content types where we want to apply a custom display with Display Suite. F28.4

F28.4

View modes

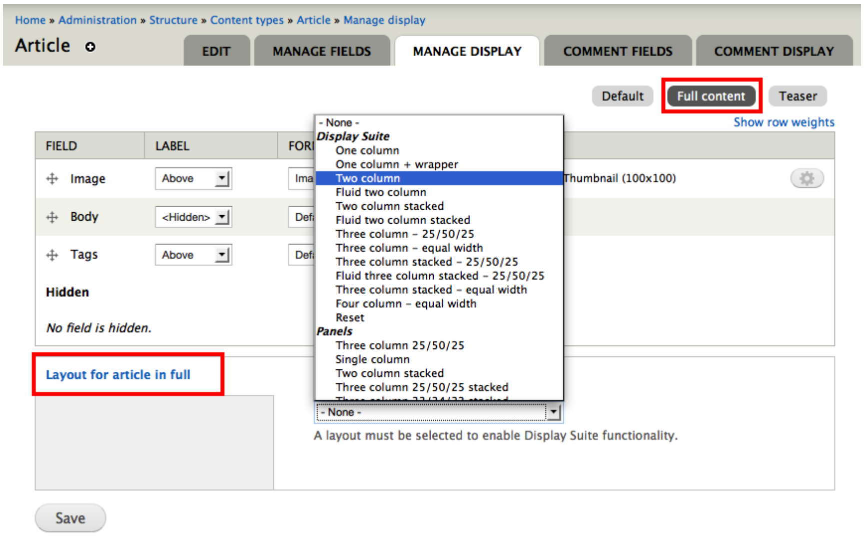
Click Custom display settings to assign displays for the different view modes of a node.



Manage display

From the **Manage display** tab for content or entities, we can choose which display we want to apply to each one of the view modes. F28.5

First, choose the view mode (Default, Full content, Summary, etc.) and then select the layout that will be applied. Only when a layout is selected here will we be using Display Suite to manage the display of the selected entity and for the selected view mode. Otherwise, the default display is used.



F28.5

Select a layout

After selecting a view mode, we can choose the layout to apply.

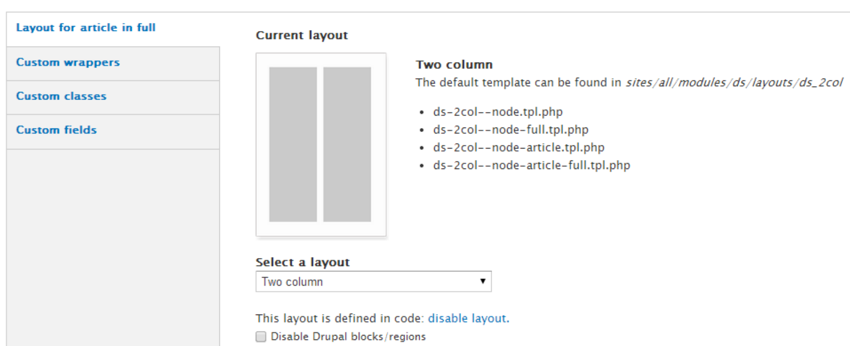
The allowed layouts should include those provided by Panels, as well as those we have created manually.

Once the layout is selected, we can choose if we want to **Disable Drupal blocks/regions** F28.6 when viewing this contect type. This option is useful for front pages or section pages that requires more space to display properly.

In order to make this option available, go to:

Administration ⇨ Structure ⇨ Display Suite ⇨ Extras [Other]

and check the option "Disable Drupal blocks/regions".



F28.6

Additional layout options

Define whether to disable blocks and sidebar regions, when the node is displayed.

Rearrange fields

After choosing a Layout, the content fields will be disabled.

The method for assigning fields to regions is similar to the one we studied for assigning blocks to theme regions. Simply select the region where you want to assign the field (or drag the field to the proper region).

The selection of available regions will depend on the chosen layout. **Figure** F28.7 shows an example of a **Two-column stacked** layout, which contains four regions: Header, **Left column**, **Right column**, and **Footer**. Apply this layout to the **Movie** content type display.

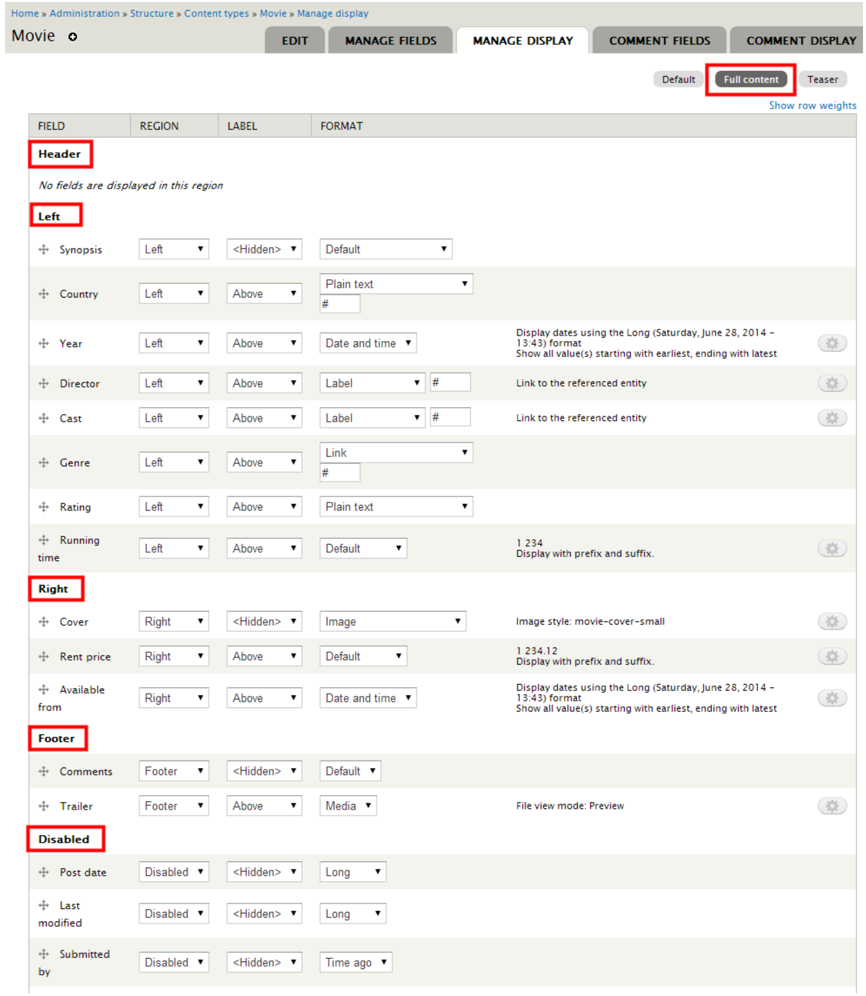
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F28.7

Arranging elements

The next step involves assigning the various elements or fields in the content type (or entity) to the regions that are enabled for the selected layout.

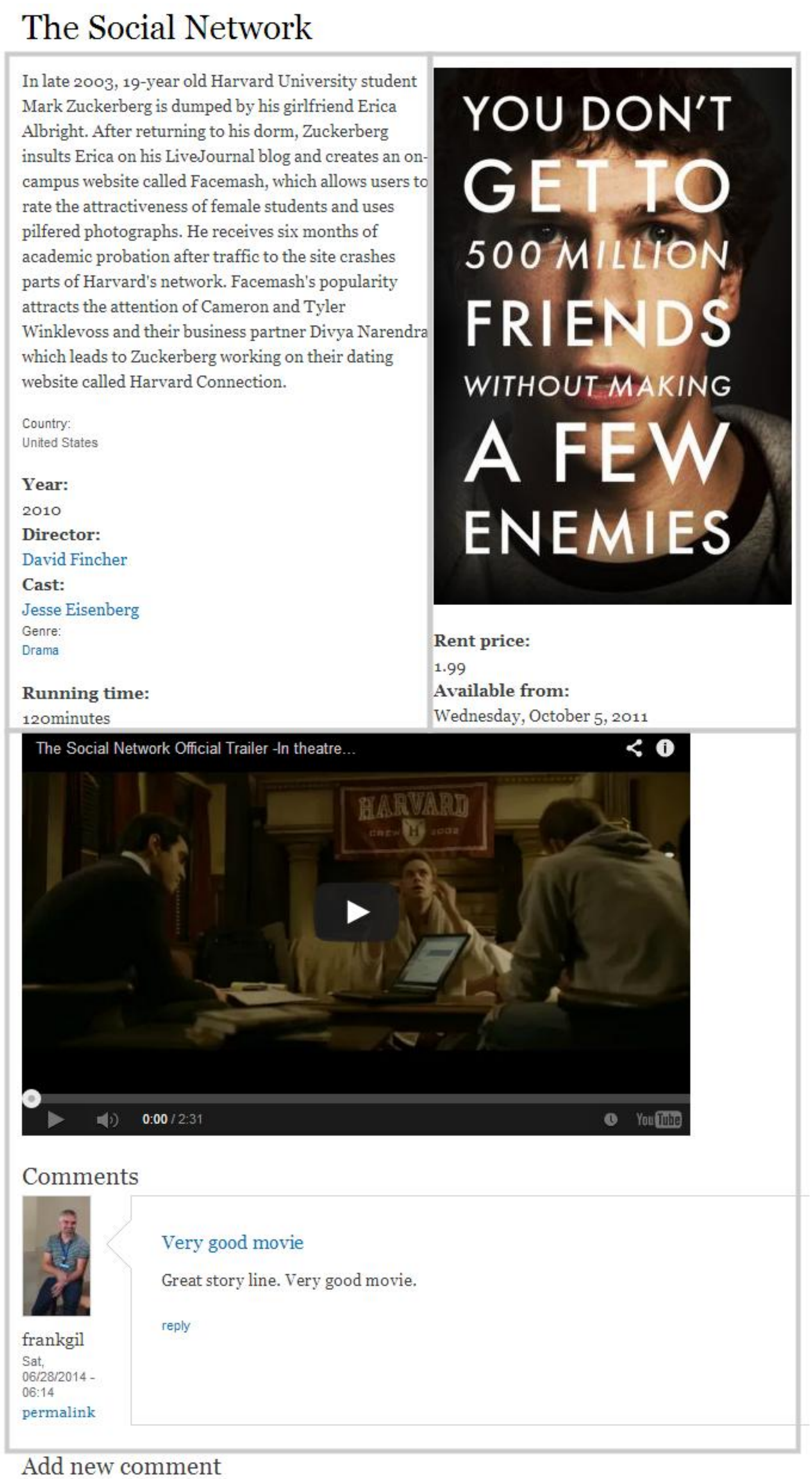
As seen above, we can also define whether or not the field label is displayed and, depending on the field type, we can modify the field's display options. For example, we can change the image style for an image field.



We have arranged the following fields in the theme regions: F28.8

- **Header**: Title.
- **Left**: Synopsis, Country, Year, Director, Cast, Rating, Running time and Genre.
- **Right**: Cover, Date and Rent Price.
- **Footer**: Trailer and Comments.

We leave the remaining fields disabled, so they will not display with the node. After saving our changes, we can view the node and check the display settings.



F28.8

Node view

The final display of a Movie type node, after applying the two-columns stacked style, with four regions: Header, Left, Right, and Footer.

To illustrate the individual regions, a border has been applied. The border is not visible in the final display.

Applying styles

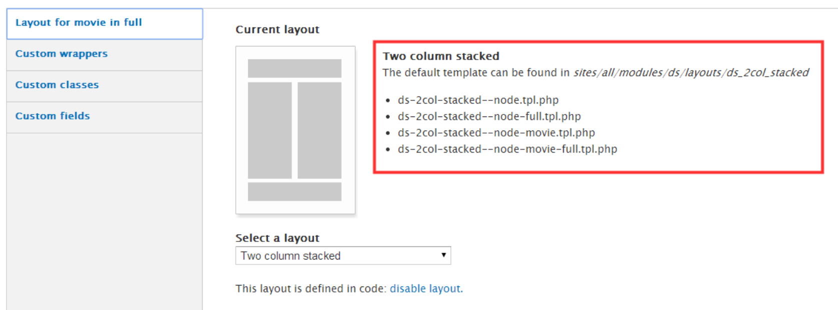
In **Unit 29** we studied how to modify the templates generated by other modules and the stylesheets for themes. In this unit, we are touching briefly on the options provided by Display Suite, which will be studied further in the following unit.

Each display that we create with Display Suite will have a template associated with it (**.tpl.php**) and a stylesheet (**.css**). These files are located in the **ds/layouts folder**, and we can use them to control the final display.

As we'll see in the next unit, **you should not change template files in the module folder**. If changes are needed, first copy the file to your theme folder and make necessary changes to the new file.

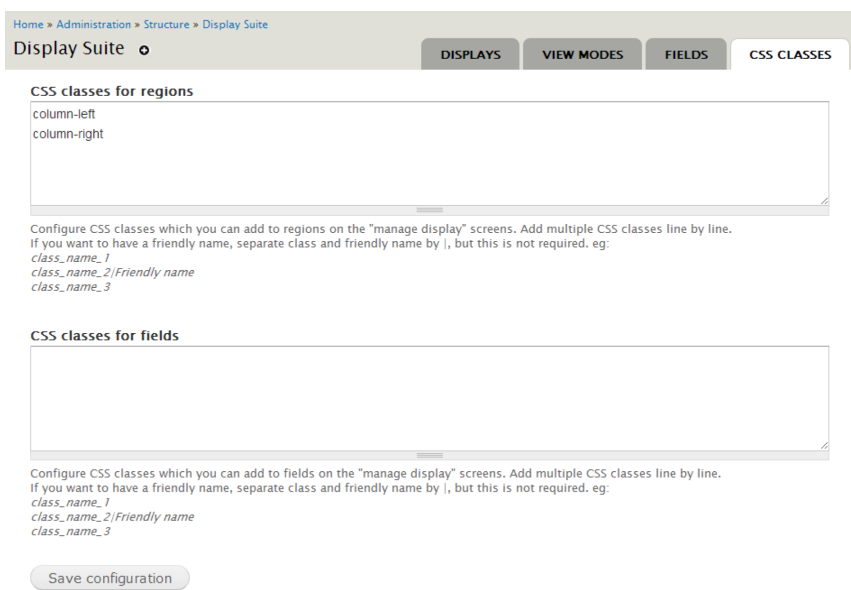
We can also create custom template files to apply to specific layouts and content types. After choosing the display, the module lists several template names that can be used to create template files within the theme. F28.9

F28.9
Templates
Each display has a related template and stylesheet. We can make edits to these files, but do so only after making a separate copy in the theme folder.



By clicking on **Custom classes for regions**, we can add CSS classes to the specified regions. First, we add new classes from the **Manage region and field CSS classes** option. F28.10

F28.10
Custom styles for regions
List of styles applied to regions. These styles must be defined on the theme stylesheet.



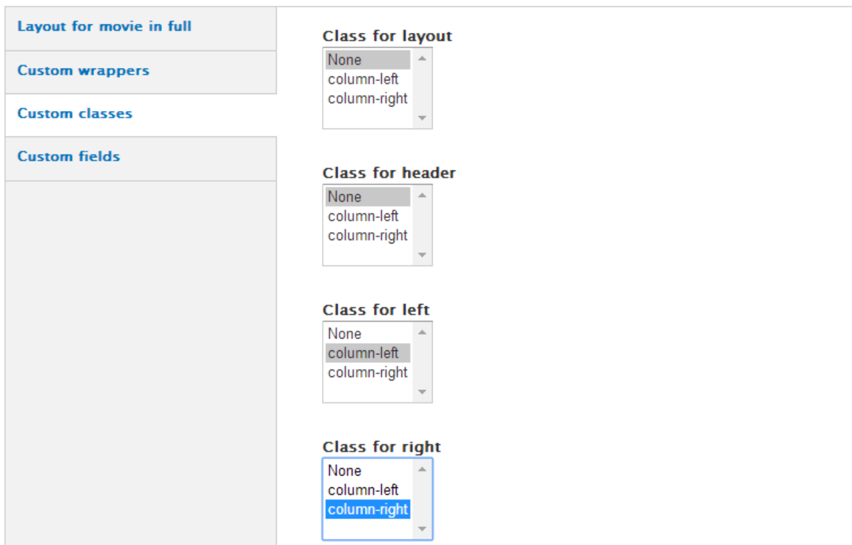
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

To specify a class, simply list the name, or use the following notation:

class|Descriptive name

With this notation, the class is on the left side of the | symbol, and on the right is the human-readable name that will be displayed in the administration section of **Display Suite.**

After adding classes, we will assign the proper regions with the **Custom classes** tab. F28.11



F28.11
Additional classes
Each display will list the region styles that can be assigned to the various regions.

These classes are included in the final HTML, allowing us to apply styles to each region with the theme stylesheet. As mentioned above, **Unit 29** will introduce the steps to modifying template files and stylesheets.

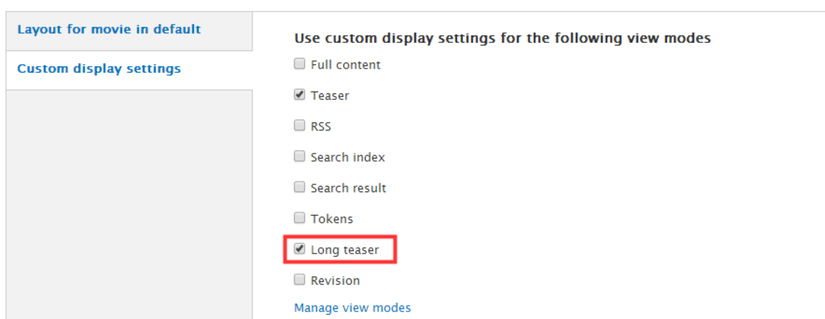
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

View modes

As we've seen, entities have several default view modes: Full content, Summary, etc. With Display Suite, we can create more view modes to display content in multiple formats, based on the context of how it is displayed. One example is creating a custom view mode. F28.12

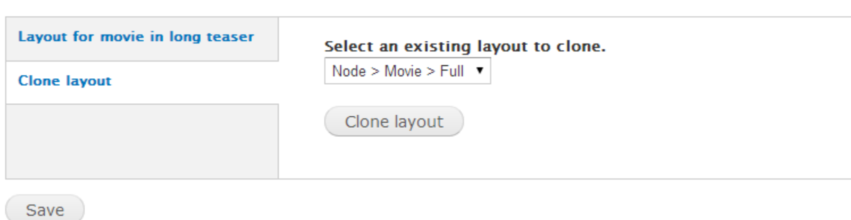
F28.12
View modes
Display Suite can create new view modes and displays.

Custom view modes will be available for the selected entities, and can be applied by clicking **Custom display settings**. F28.13



F28.13
View modes
To enable a new view mode for an entity, you must enable it under Options for custom display.

Finally, we have an option to take the layout and other settings from one view mode and apply it to another. We can do this from the **Clone layout** option, by selecting one of the existing layouts. F28.14



F28.14
Clone layout
Existing displays can be cloned from one view to another.

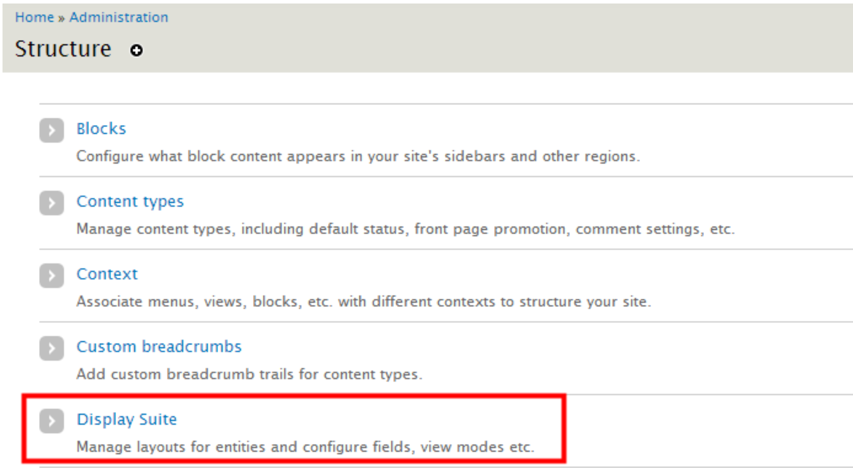
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Display Suite administration page

Some of the options we studied above can be accessed directly from the administration section of Display Suite. F28.15

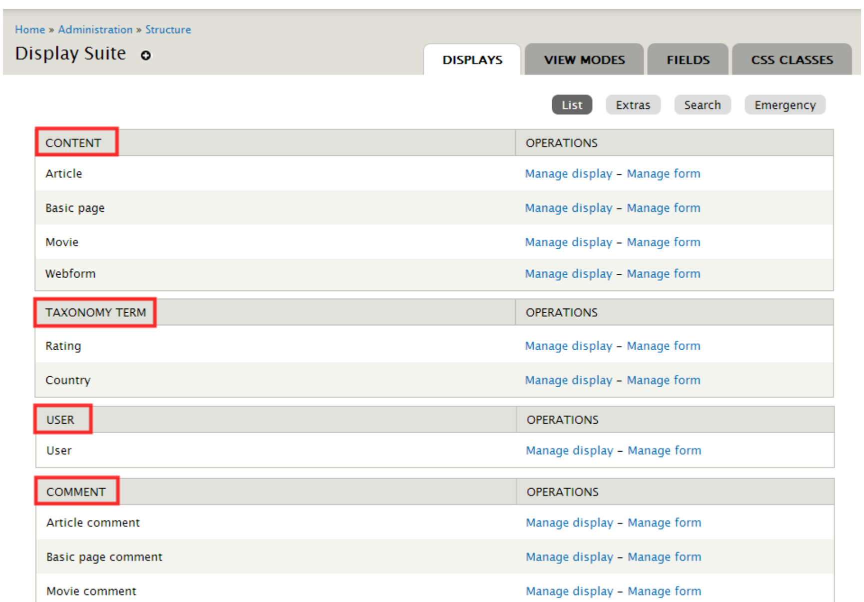
Administration ⇨ Structure ⇨ Display Suite

URL Display Suite
/admin/structure/ds



F28.15
Display Suite
The Display Suite administration section provides direct access to several tools.

Displays tab includes a list of **Manage display** links for all the entities on the site. These links are organized by entity type. F28.16



F28.16
Displays
The Displays option includes a list of **Manage display** links for each element that we can control with Display Suite.

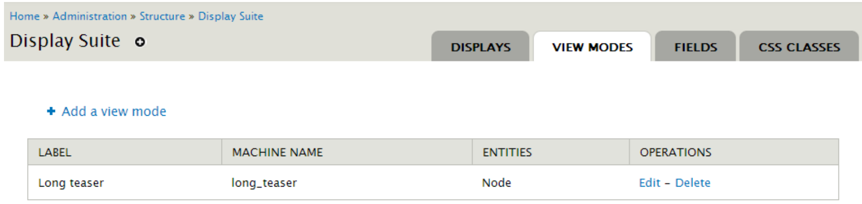
With **CSS Classes**, we can manage classes to control CSS styling. These classes are generic for all entities, but we can assign them individually to each region and layout, by view mode. With **View modes**, we can access the custom view modes. These view modes are shared across entities, although they do not have a custom layout if they are not enabled for an entity. F28.17

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F28.17

View modes

Custom view modes. Also provides access to create new view modes.



In the next sections we will explore the other options available from the main Display Suite page.

Fields

28.3

Display Suite allows us to add new fields that will be displayed with the final entity layout.

Keep in mind that these fields do not behave like traditional fields we've seen up to now. **They do not store information** entered during the node creation or edit process, rather **they serve to display information** on the site.

For example, they are useful for **generating information that is calculated from the content in other entity fields.**

These fields can be managed through Display Suite:

Administration ⇨ Structure ⇨ Display Suite ⇨ Fields

URL Fields (Display Suite)
/admin/structure/ds/fields

F28.18
Fields
Display Suite fields are used for final layout and pull content from other entity fields.

There are four different types of fields: F28.18

- **Code field**. Execute PHP code.
- **Dynamic field**. This is a generic field with content derived from any entity, using the layout management option.
- **Block field**. Displays a block in the field.
- **Preprocess field**. Show individual values for the object entity.

Code field

The first option, **Add a code field**, allows us to create a field that will display the output of executed PHP code.

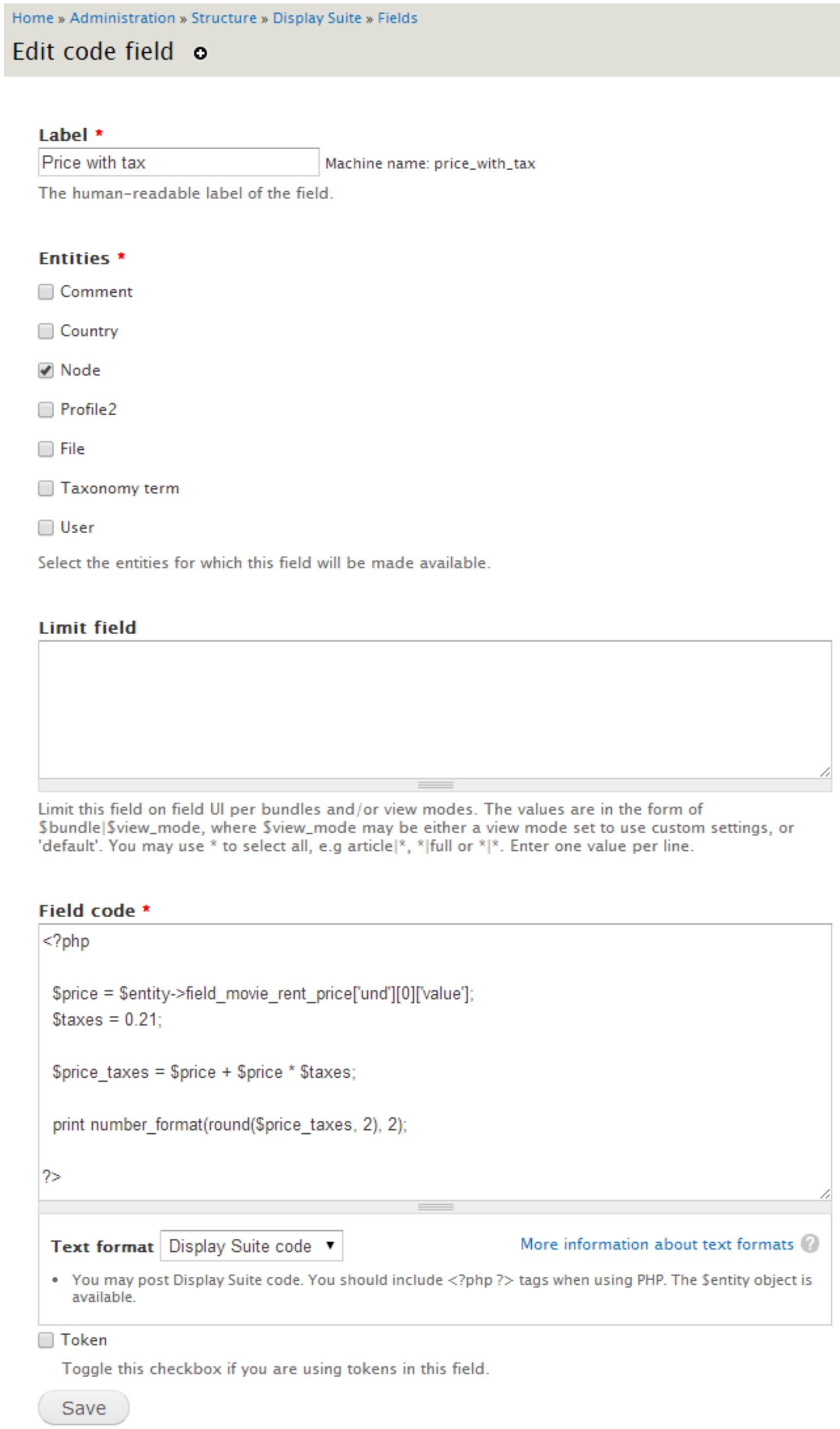
After creating a code field, we have to indicate which entities will use it. The code field contains PHP code, including the opening and closing tags (<?php and ?>), and we should select the **Display Suite code** text format. When we choose this format, we also have access to the **$entity** object which contains all the entity fields.

We can also use wild cards and replacement tokens, allowing us to access other elements related to the entity that we're working with. Keep in mind that the entity type will determine which variables are available.

F28.19

Code field

A code field is used to execute PHP code. The code allows us to access the $entity object, which contains all the values related to the entity.



Note

Access the $entity object values

A simple way to get the structure and values of the $entity object is by executing the following code:

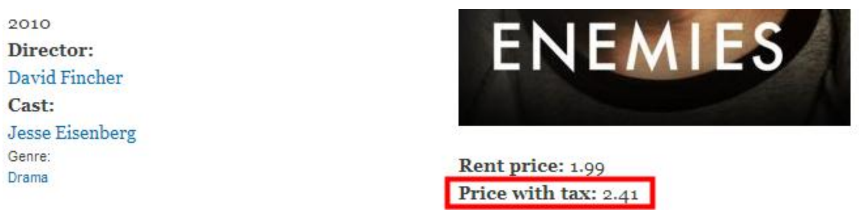
```
print_r($entity);
```

In the advanced training, we'll study advanced tools for obtaining the structure and values of objects and variables on the site.

Figure F28.19 shows an example where we create a code field called **Price with tax** that will calculate the price of the movie with added tax, for the Movie content type. The value is calculated using the node's price field, which we can access using the **$entity** object.

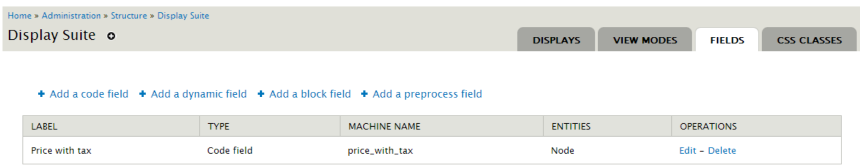
This field is initially disabled, but we can activate it for each of the selected entities. For example, if we've selected the Node entity, then the field will be available for all content types.

From the manage display option for the chosen content type, we can select the relevant view mode and place the field in a region, as needed. We can also configure whether or not to display the label. F28.20



F28.20
Code field. Display
The code field can be displayed as an additional field, using the proper tags.

Custom fields can be managed from the **Fields** option within Display Suite, where we have direct access to edit and delete links. F28.21

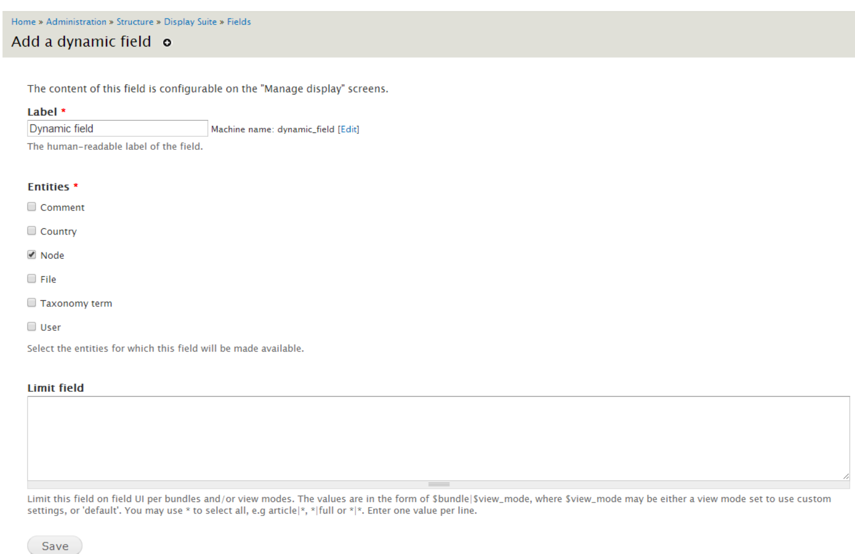


F28.21
Fields
Custom fields are displayed in the Fields list, with links to edit or delete.

Dynamic field

Add dynamic field creates a generic field that, later, can take on the value of any other field or element on the site, through manage display.

After creating a dynamic field, we simply specify a Label and the type of Entities where it will be available. F28.22



F28.22
Dynamic field
A dynamic field is a generic field that will take the value of another field or element on the site, through manage display settings for the entity.

This process is similar to the one we used for adding content to panels. Even the allowed elements are similar, depending on the types of entity being referenced.

To add content to this field, we navigate to the configuration settings in Manage

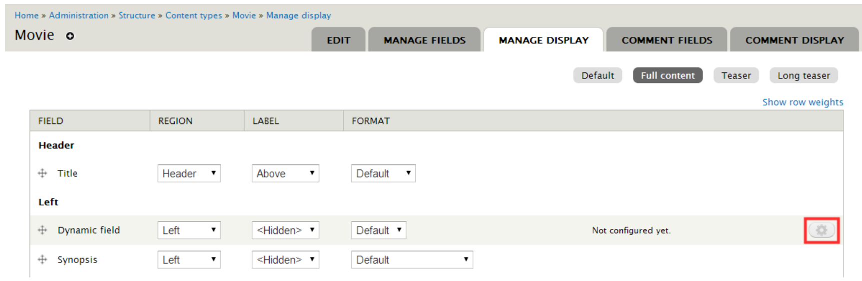
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

display for a specific entity and view mode. Click on the gear icon on the right side to add the content, which appears in an overlay window, similar to what we see for adding content to Panels. F28.23

F28.23

Dynamic field

With the gear icon, we can configure the dynamic field.

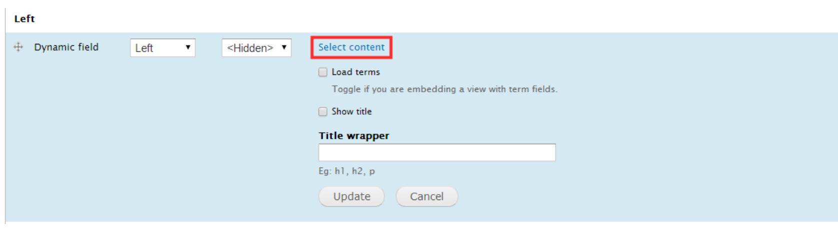


Click on **Select content** (and then on the link for Click here to select new content) to open a new window with the allowed content. F28.24

F28.24

Select content link

The link for Select content allows us to create a relationship between one element and the dynamic field content.

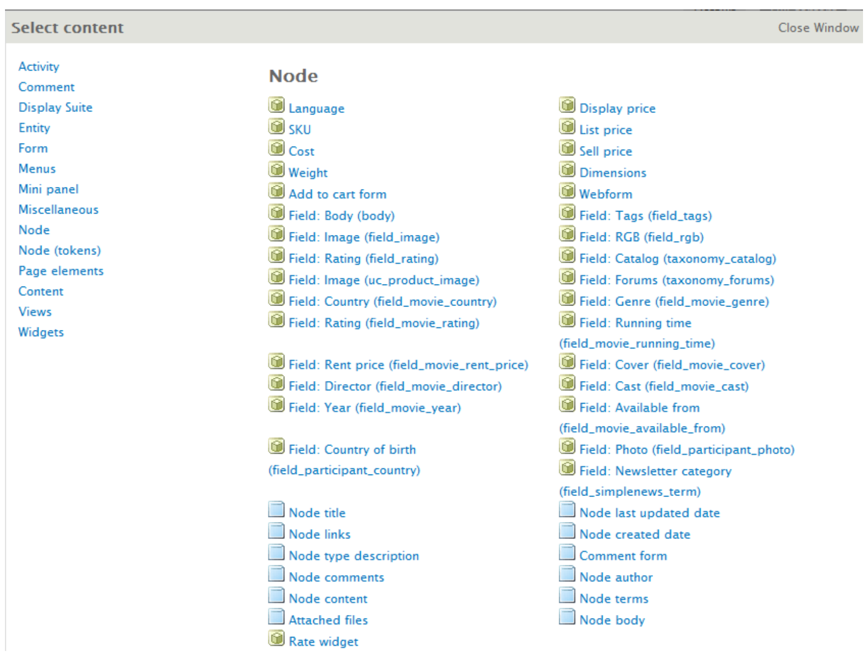


Simply choose the element you want to appear in the field and configure the relevant options, depending on the element. F28.25

F28.25

Select content

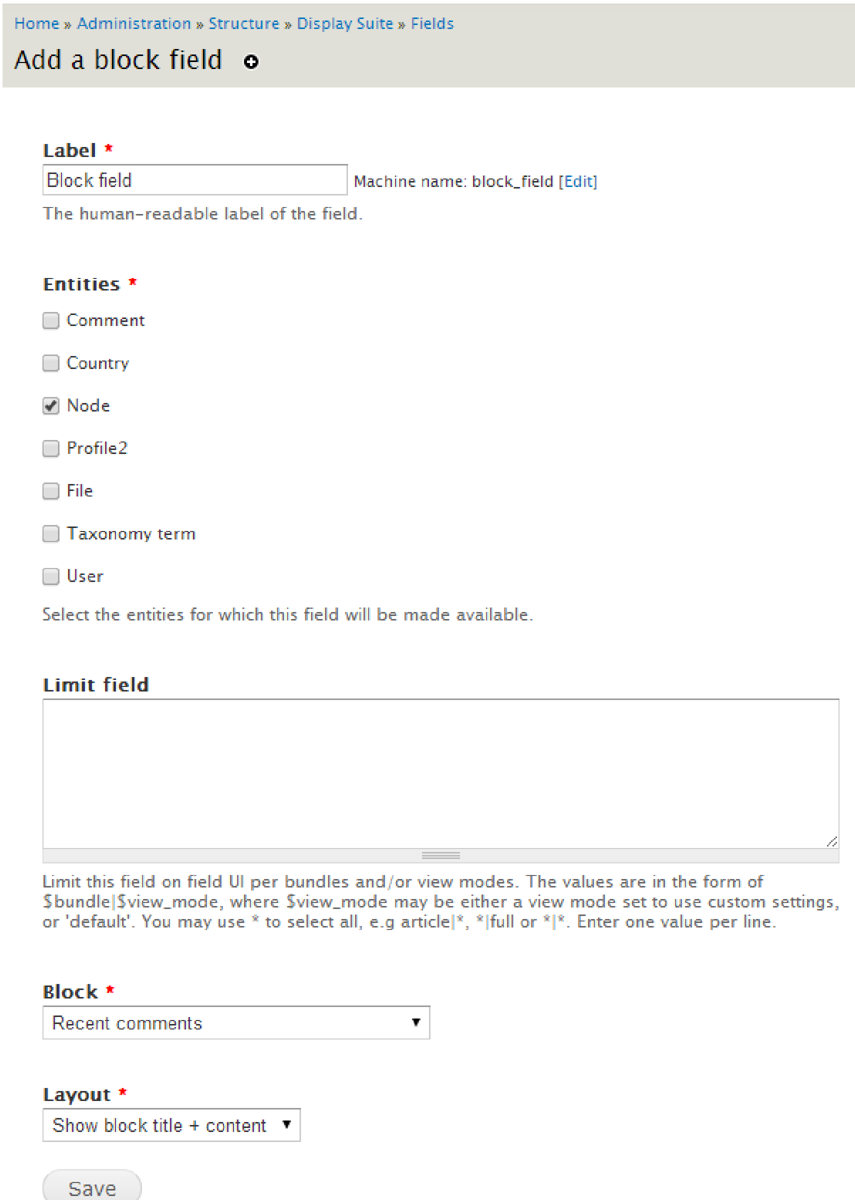
The selection of content and elements available is similar to what we see when creating panels. Some categories only appear as a result of the entity linking to the dynamic field (User, Node, etc.).



Once content is assigned to the field, we can continue the configuration process, assigning it to a region as with any other field.

Block field

Block field takes block content and displays it as a field. To create this type of field, we must select the block content, label, entity type, and the layout type (title and content, or block content only). F28.26



F28.26

Block field

Block field is used to display block content as a field.

To create a block field, select the block and a layout style.

Once the field is created, we can activate it and assign it to a region, as with other fields.

Preprocess field

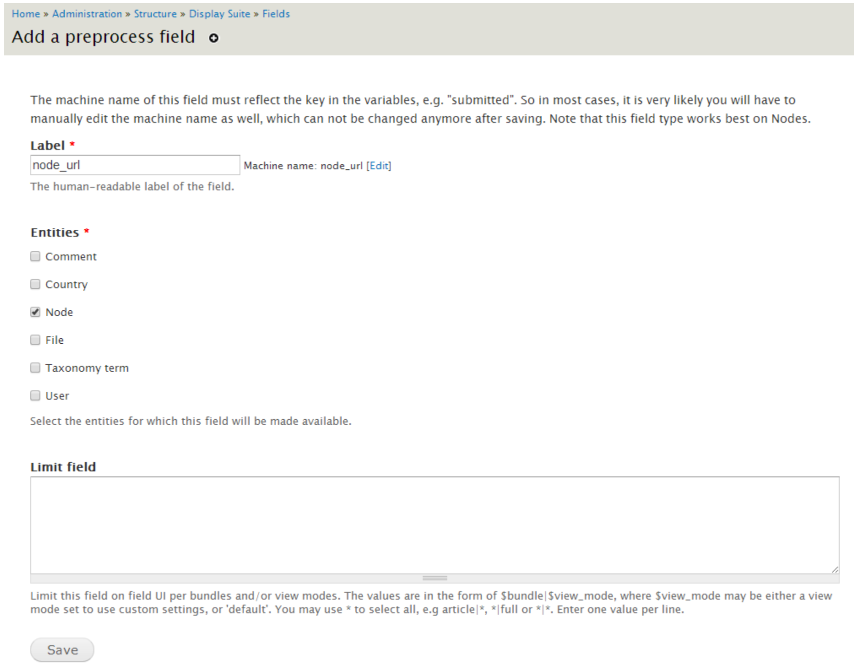
This field type can display the content of variables available when Drupal is preparing the node. First, the **Machine name** should include a variable name without the $ sign. For example, if the variable that we want to display is $node_url, then we enter node_url for the label, which is also the machine name.

Any label is acceptable, as long as we edit the **machine name** to match the name of the variable being targeted. F28.27

F28.27

Preprocess field

This field can access preprocess variable values, which are sent to the template for final layout.



We can use any of the variables available in the template file, as they are transmitted during the preprocessing phase. We will look in detail at template files and variables in Unit 29.

The **node.tpl.php** template, for example, contains a set of variables that we can also use in the preprocess field ($title, $content, $user_picture, $node_url, etc.). F28.28

F28.28

Template variables

Each tpl.php template file lists the variables available. These are the variables we can use in the preprocess field.

```
<?php

/**
 * @file
 * Bartik's theme implementation to display a node.
 *
 * Available variables:
 * - $title: the (sanitized) title of the node.
 * - $content: An array of node items. Use render($content) to print them all,
 *   or print a subset such as render($content['field_example']). Use
 *   hide($content['field_example']) to temporarily suppress the printing of a
 *   given element.
 * - $user_picture: The node author's picture from user-picture.tpl.php.
 * - $date: Formatted creation date. Preprocess functions can reformat it by
 *   calling format_date() with the desired parameters on the $created variable.
 * - $name: Themed username of node author output from theme_username().
 * - $node_url: Direct url of the current node.
 * - $display_submitted: Whether submission information should be displayed.
 * - $submitted: Submission information created from $name and $date during
 *   template_preprocess_node().
 * - $classes: String of classes that can be used to style contextually through
 *   CSS. It can be manipulated through the variable $classes_array from
 *   preprocess functions. The default values can be one or more of the
 *   following:
 *   - node: The current template type, i.e., "theming hook".
 *   - node-[type]: The current node type. For example, if the node is a
 *     "Blog entry" it would result in "node-blog". Note that the machine
 *     name will often be in a short form of the human readable label.
 *   - node-teaser: Nodes in teaser form.
 *   - node-preview: Nodes in preview mode.
```

Extras

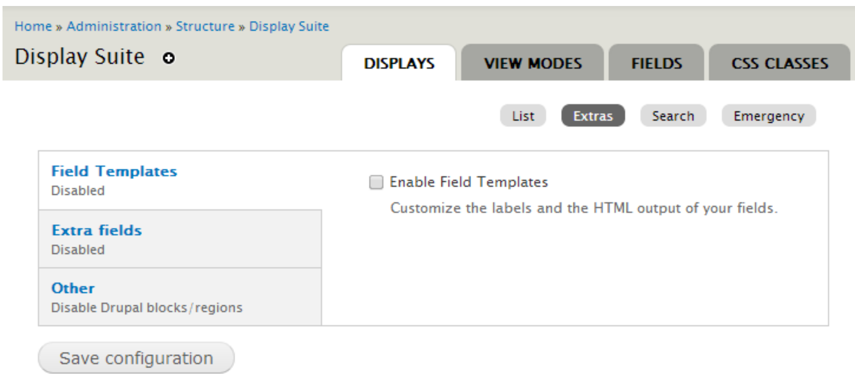
28.4

If we activate the Extras module inside Display Suite, we have access to Extras options:

Administration ⇨ Structure ⇨ Display Suite ⇨ Extras

URL Extras (Display Suite)
/admin/structure/ds/extras

Extras include additional settings for Display Suite, which are initially disabled.



F28.29
Extras
Additional options for configuring content layout with Display Suite.

The options are: F28.29

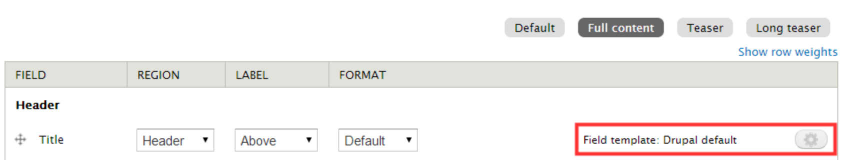
- **Field templates**. Modify the display of each field.
- **Extra fields**. Add fields from other modules, to use in Manage displays. This option is not demonstrated here.
- **Other**. Various functions.

Field templates

Modify the template of each field. When **Field templates** option is enabled, we can select a default mode for each field, such as:

- **Drupal Default**. Original mode generated by Drupal core.
- **Full reset**. Remove all HTML tags.
- **Minimal**. Remove HTML tags, while maintaining field container.

Each field will display a **Field display** link that can be used to modify the field template. F28.30



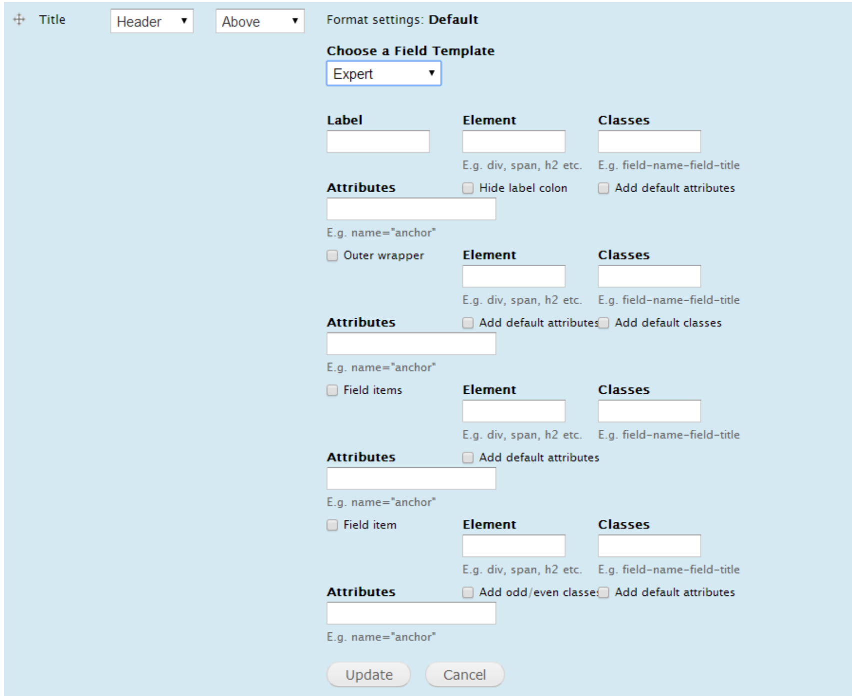
F28.30
Field display
With field template enabled, the Field display link is available for each field.

With **Expert** mode, we can decide which labels and classes to apply to each field element (container, field elements, etc.). F28.31

F28.31

Field display

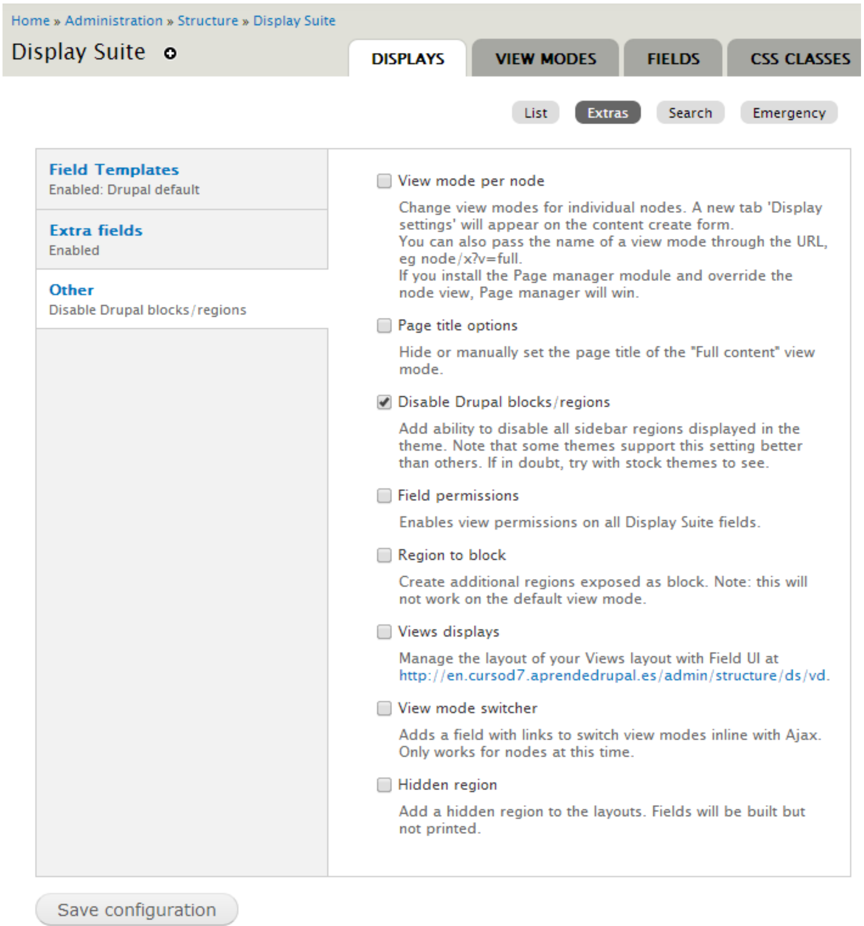
After selecting the field template, we can configure the various options for the chosen style. Expert mode provides a flexible and detailed form for assigning CSS classes to the field.



Other options

The **Other** tab includes other Display Suite functions, for example: F28.32

- **View mode per node**. Configure view modes for each node, independently.
- **View mode switcher**. Add a field to switch between view modes, using AJAX. This option only works with nodes, not other entities.
- **Page title options**. Hide the content title in **Full content** mode.
- **Views displays**. Add the **Views displays** option on the Display Suite administration page, to manage Views templates.



F28.32

Other options

The Other tab incudes additional features, including Views displays which is required to manage Views displays, which we will explore in the next section.

28.5 Displaying views

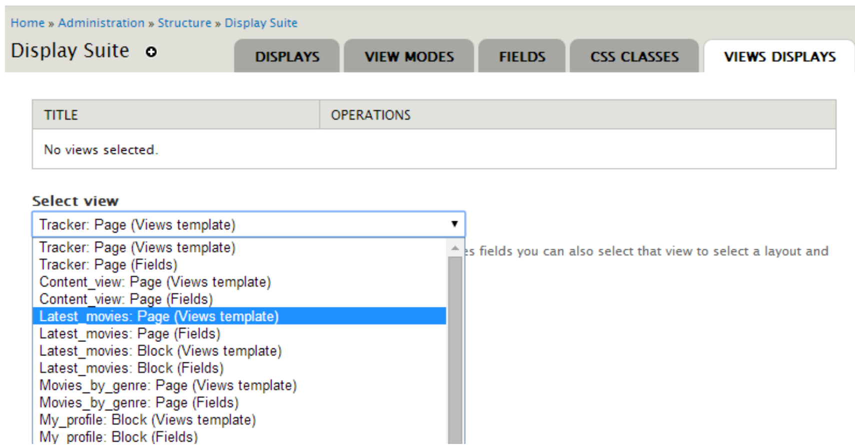
The Views displays option is available in Display Suite when the **Views displays** option is enabled:

URL Extras (Display Suite)
/admin/structure/ds/extras

Administration ⇨ Structure ⇨ Display Suite ⇨ Extras [Other tab]

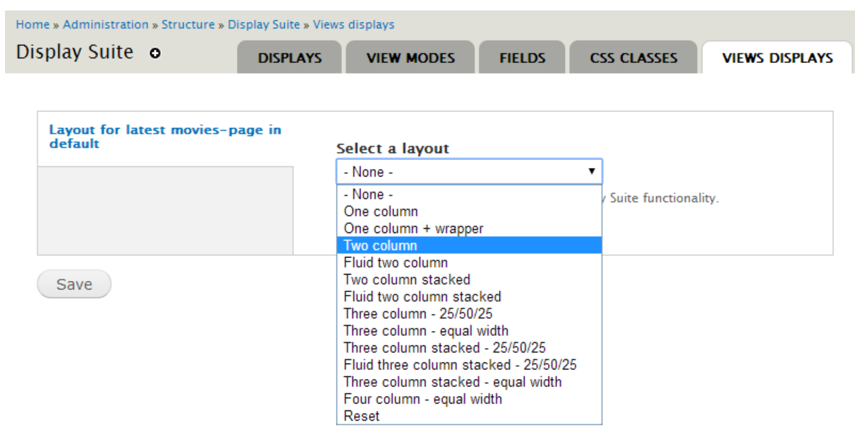
To proceed, first make sure the **Extras** module inside Display Suite is enabled.

In **Views displays**, add the Views that you want to manage from within Display Suite. F28.35 The Views are organized by display, and in groups according to Views template or Fields. The **Views template** mode defines the general style of the view. The **Fields** mode allows us to select a layout to display the fields in the view.



F28.35
Views displays
In Views displays, select the views that you want to manage in Display Suite.

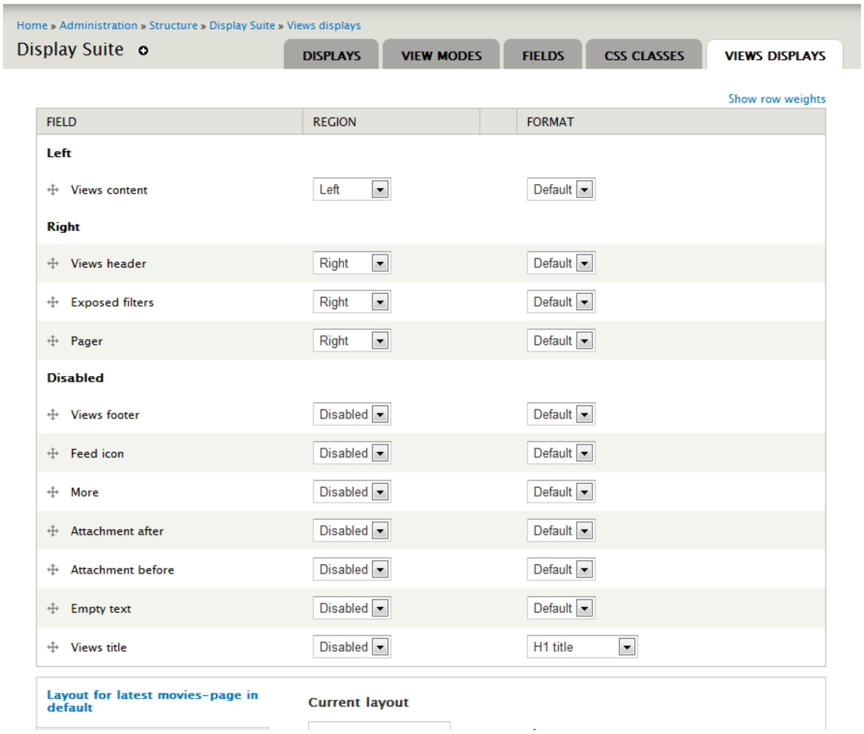
After adding a display, you can select a layout to apply to the view. F28.36



F28.36
View layout
Choose a layout to apply to the view.

As we did with the fields from an entity, F28.37 F28.38 the view elements can be assigned to a region.

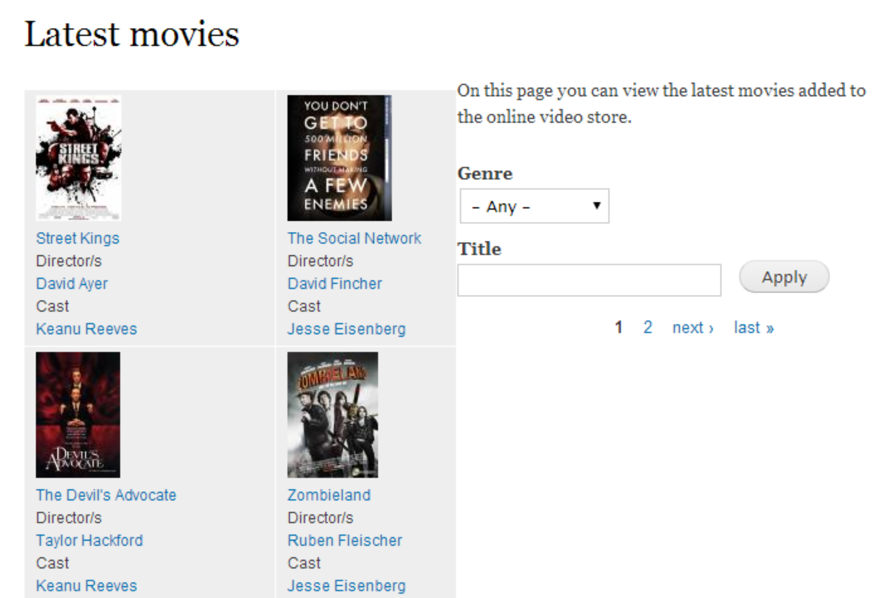
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



F28.37

Views elements

As with entities, the elements of a view can be assigned to regions, after choosing a layout.



F28.38

Views display

The Figure shows an example of a view whose elements have been placed into a two-column layout.

28.6 Displaying forms

Display Suite also can manage the layout of site forms, using the **Forms** module within Display Suite. To use the module, first we must install the **Renderable elements** module, available here:

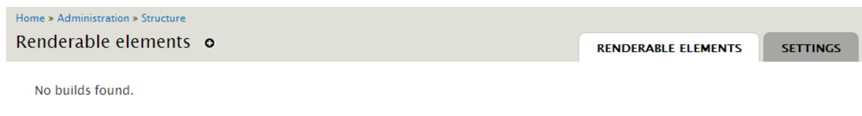
http://drupal.org/project/rel

Manage Forms displays from:

Administration ⇨ Structure ⇨ Renderable elements

The **Renderable elements** tab is shown inside Forms. Initially, the list is empty, as we have yet to add a form. F28.40

F28.40
Renderable elements
Initially, no forms are shown.

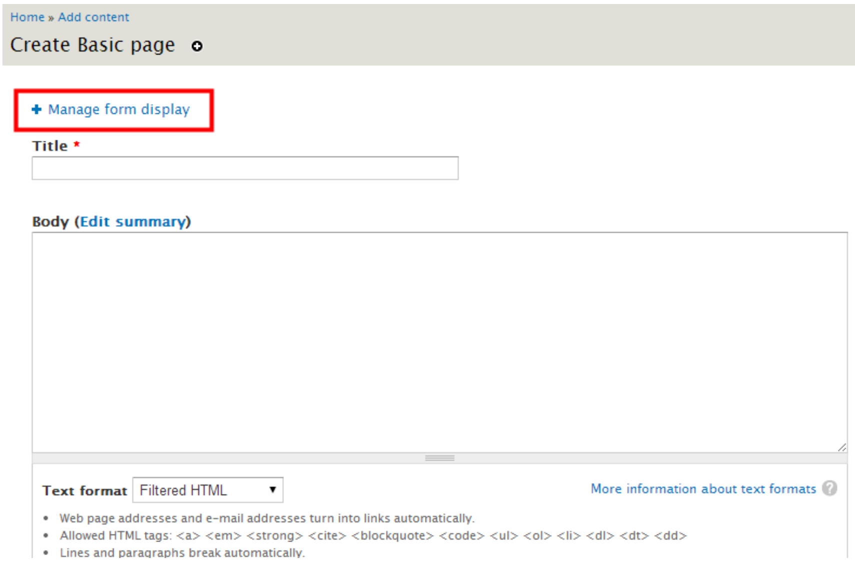


Before adding forms, let's review the configuration options for the module, from the **Settings** tab: F28.41

- **Enable form registration**. Add the **Manage form display** link to each site form. Clicking on this link makes each form a **renderable element**, allowing us to manage them from within Display Suite.

- **Ignore registration on specific pages**. Select pages to exclude from the previous setting. By default, all administration pages are excluded.

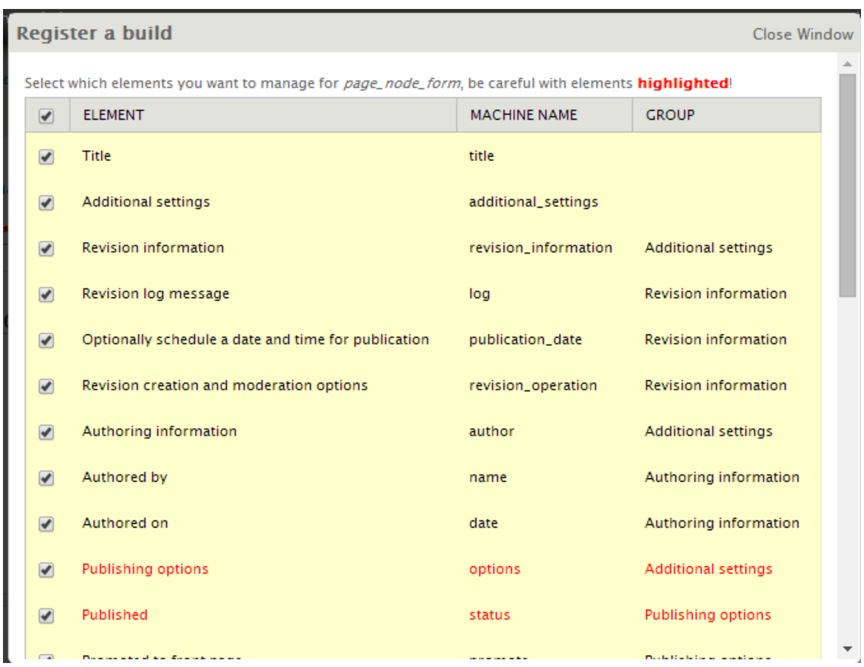
Site forms now will show the **Manage form display**, when available, which we can use to add the form to the list of renderable elements. After adding an element, the layout controls are passed to Display F28.41 Suite.



F28.41

Enable form registration

Add a link to site forms, connecting them to Display Suite.



F28.42

Select form elements

Select the form elements to register the form as a renderable element.

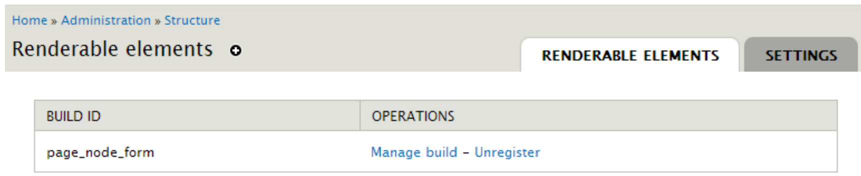
After adding a form, let's return to Renderable Elements. The available options are: F28.43

- **Manage build**. Manage the display of a form.
- **Unregister**. Remove the form from the list, handing control of the display back to Drupal core.

F28.43

Renderable elements

Renderable elements allows us to manage the display of forms.



The interface for managing form display is similar to the one for entity layout. Select the appropriate **layout** and assign the fields to the regions. F28.44

F28.44

Form display

The method of managing the form display is similar to the one for applying entities and views. Select one display and rearrange the form elements in the appropriate regions.

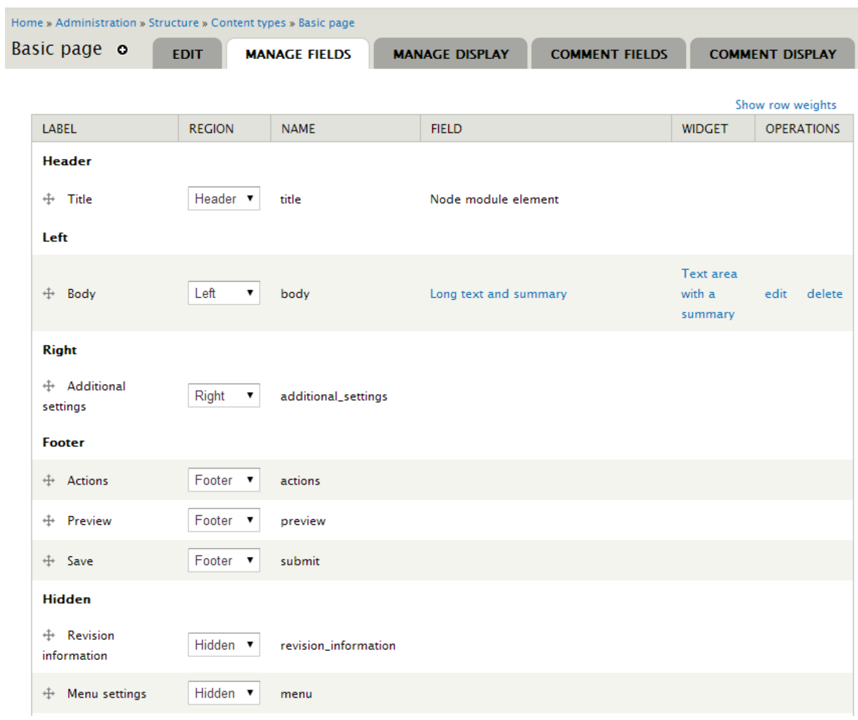
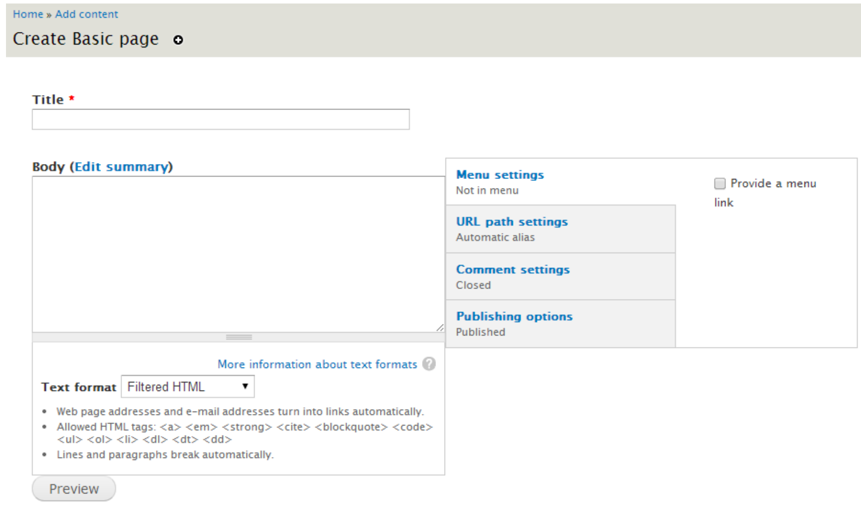


Figure F28.45 shows the **Create Basic page** form, with fields assigned to a "two-column stacked" layout.



F28.45

Example form

Example of Create Basic page form, assigned to a two-column stacked layout.

Additionally, the form includes a "**Reset form display**" link, used to remove the form from the list of renderable elements and hand control of its display back to the core or the corresponding module.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

28.7 Displaying search results

Display Suite also can manage the display of search results, by using the **Display Suite Search** module in the suite.

With the module enabled, we can control the display of search results here:

URL Display Suite Search
/admin/structure/ds/list/search

Administration ⇨ Structure ⇨ Display Suite ⇨ Search

After enabling the module, a message will appear when we navigate to the search options within Display Suite for the first time, indicating that the module has its own search method. We must activate this option at **Search settings**.

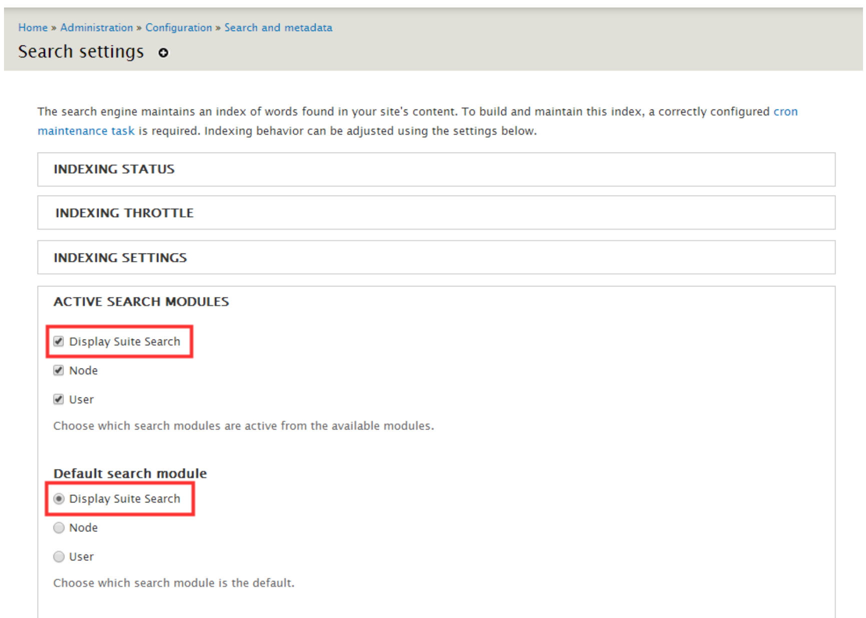
To activate the search method for Display Suite, navigate to:

URL Search options
/admin/config/search/settings

Administration ⇨ Configuration ⇨ Search and metadata ⇨ Search settings

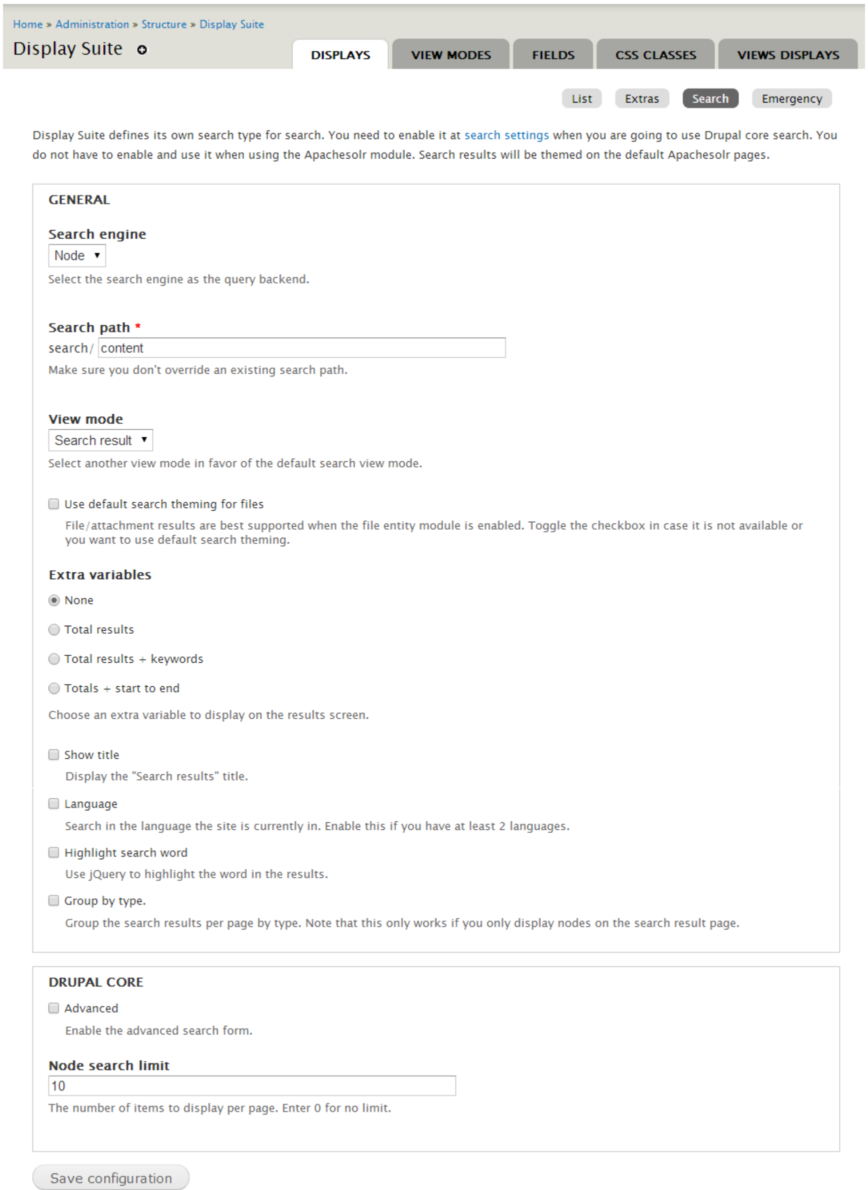
To use the Display Suite search method, select **Display Suite Search** under **Active search modules** and under **Default search module**. F28.46

F28.46
Search options
To use the Display Suite search options, select **Display Suite Search** from **Active search modules** and in **Default search module**.



If Display Suite search method is being used, then it's wise to **disable the Node search method**, otherwise both search tabs will be displayed.
Return to the search configuration page within Display Suite, and locate the following options: F28.47

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



F28.47

Search

Options for search results in Display Suite.

The most important option is view modes. To modify the display of search results, create a new view mode which we can use to change the display of each entity.

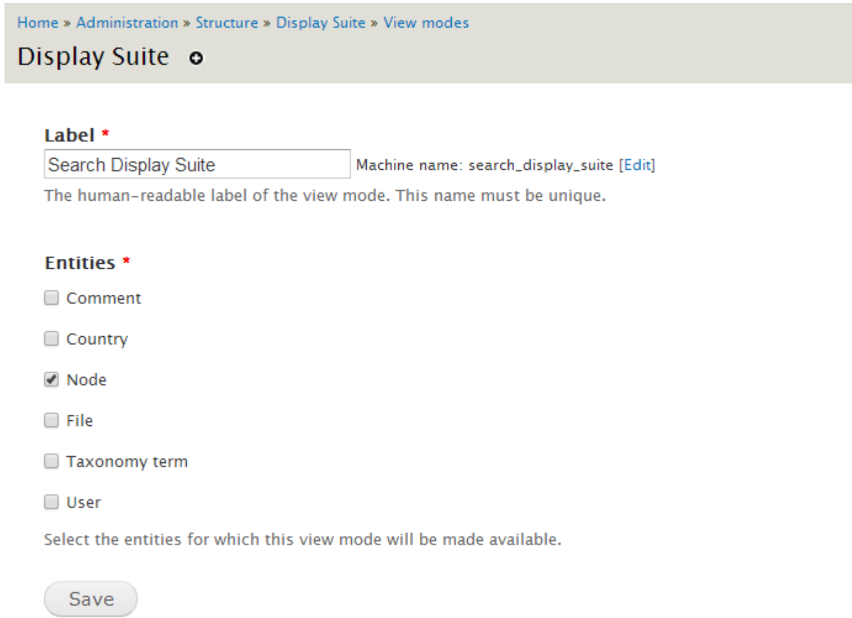
- **Search engine**. Initially, this option is set to Node search engine, which is the default option controlled by Drupal core.
- **View mode**. Select the view mode to modify the presentation of search results. Instead of using the default "Search result" view mode, we'll create a custom view mode called "Display Suite Search".
- **Extra variables**. Include additional values in the search results, such as the total number of results.
- **Show title**. Display the title "Search results."
- **Language**. Search in one language only. This is useful only on sites that support multiple languages.
- **Highlight search word**. When enabled, this option will highlight the search term. Enter a CSS class to modify the display with stylesheets.
- **Group by type**. Organize the results by type.
- **Advanced**. Enables the advanced search form.
- **Node search limit**. Number of items to display per page. Enter 0 for no limit.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Create a view mode

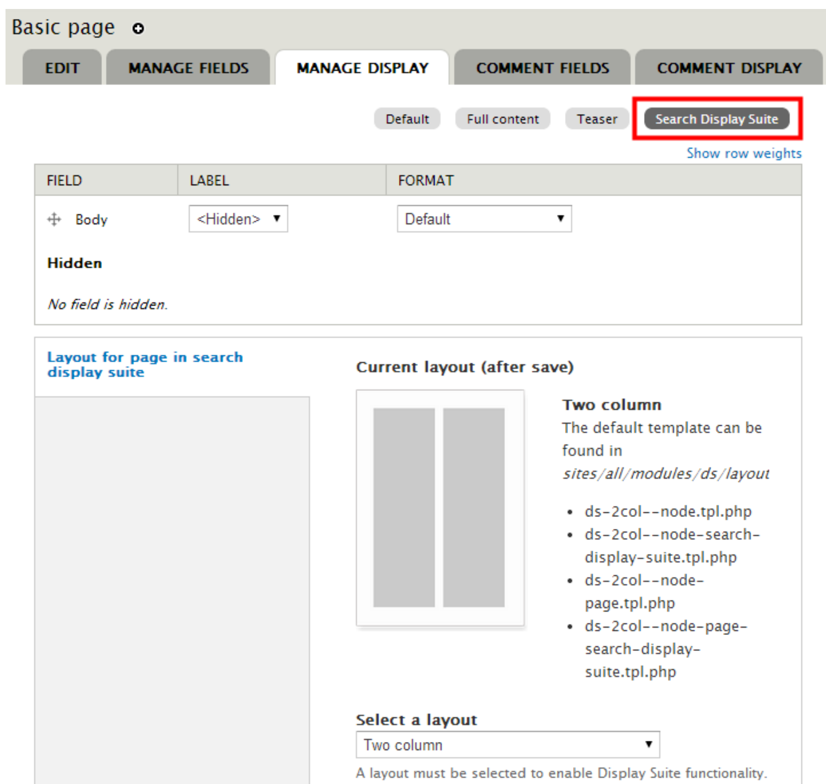
The next step is to create a new view mode to display our search results. Let's call it **Display Suite Search**. F28.48

F28.48
View modes
New view modes for search results.

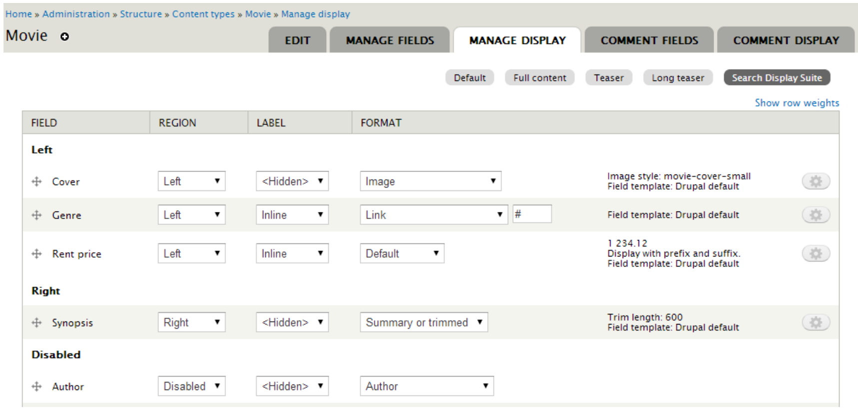


The new view mode is available for the entities we've selected. Remember to enable the view mode in **Custom display settings** (from **Manage display**). F28.49

F28.49
Layout
Layout applied to a new view mode.



After selecting a layout, we can assign content to the regions. F28.50

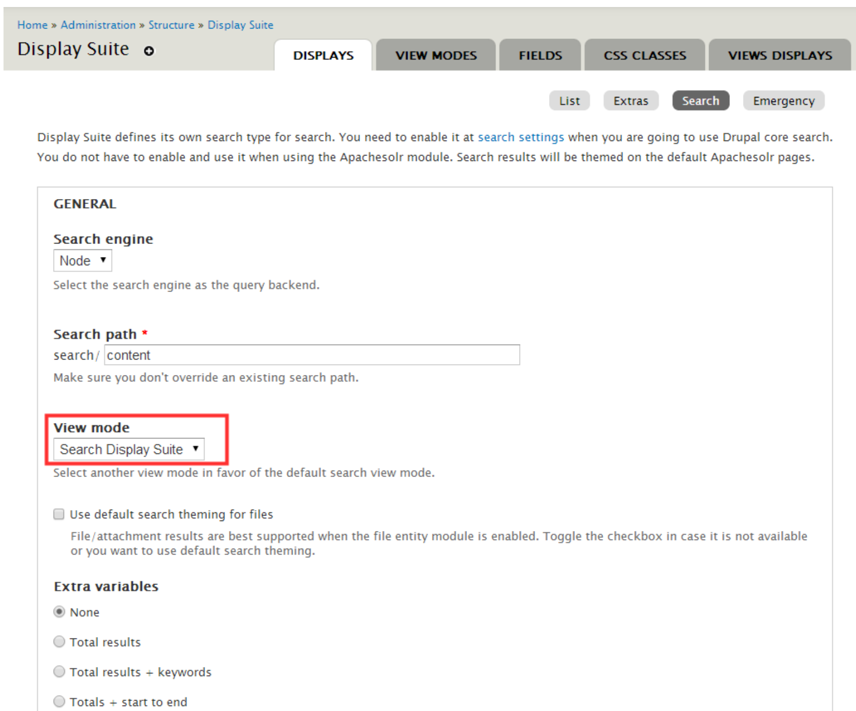


F28.50

Assigning fields

Assign fields to regions that will appear in the search results.

For the new view mode to take effect on our search results, we have to return to Search options under Display Suite and assign the new view mode. F28.51



F28.51

View mode

Don't forget to assign the new view mode that was created, by accessing the Search option within Display Suite.

Finally, let's configure the layout and structure of elements for each content type (or entity) in the view mode. This allows us to control how search results are displayed, adding the possibility of multiple display types in the same list of results, to differentiate between multiple content types.

Remember to run Cron after making system changes like this, to ensure the most recent content is indexed on the system.

In the following example, we've modified the display of search results for Movie and Participant content types. Figure shows the search results contain both types of content. F28.52

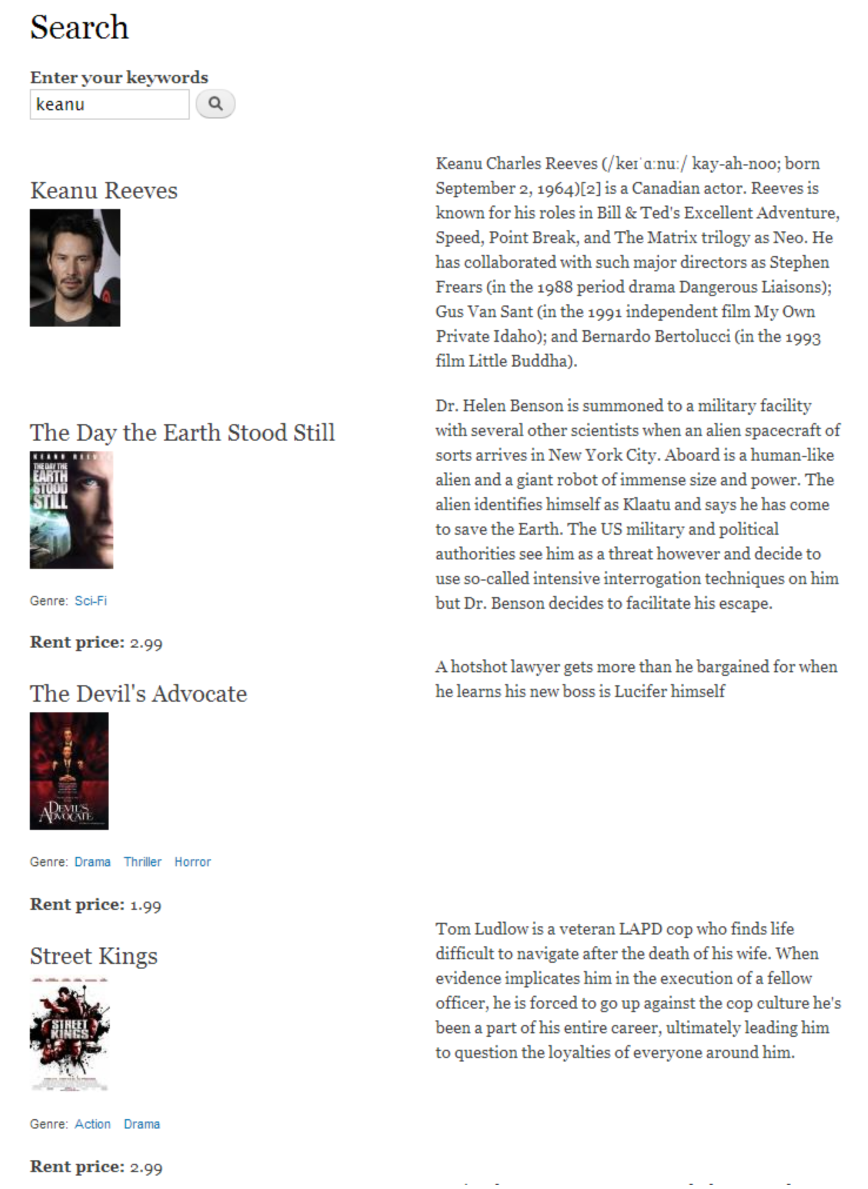
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F28.52

Search results

Search results showing Movie and Participant content types.

Each content type has its own display style.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

29 Customizing themes

The themes consist of a set of template files (.tpl.php), style sheets (.css) and other files such as images and Javascript files. We will begin this unit by analyzing this file structure.

The Initial Level makes a brief introduction to the modification of the aspect of a subject through the CSS style sheets. At this level we go a step further toward the modification of topics. This, however, is not the final step. We include an introduction to Web publishing with HTML; which is something fundamental if we want to have greater control over the final design of the site. We must not be afraid to modify the style sheets or HTML code because it is not a question of programming languages but simply of markup languages or labelling which is simpler to use.

Drupal templates mix HTML with PHP and so we must use both languages well to avoid the site "break". Whenever we are going to modify the theme of the site, we have to make sure that we first have activated a different administration theme.

Comparative D7/D6

Themes

Regarding the intermediate level, the theme system in Drupal 7 is practically the same as Drupal 6.

We encounter small differences in the template files available and in the variables that can be used in each one of them.

Unit contents

29.1 Theme structure 278
29.2 Theme template files...280
29.3 Introduction to web layout with HTML and CSS.............287
29.4 Base Themes and Sub Themes 304

29

29.1 Theme structure

A theme is a set of files that define the presentation layer of a site. Only one of the files in the set is mandatory: the **.info**, where you find the theme's definition and some general configuration options.

Themes can also contain **subthemes**, which inherit templates and styles from the parent theme, but the child can be modified to build a variant of the parent. There are special themes, called **base themes** that supply the functional and structural base to build new themes without having to start from scratch. We can thus build new themes that will in fact be subthemes of a base theme.

Figure F29.1 shows the typical file structure of a theme. As previously stated only the **.info** file is mandatory, both in themes and subthemes. The rest of the files can be included or not in the installed theme, without site functionality being affected.

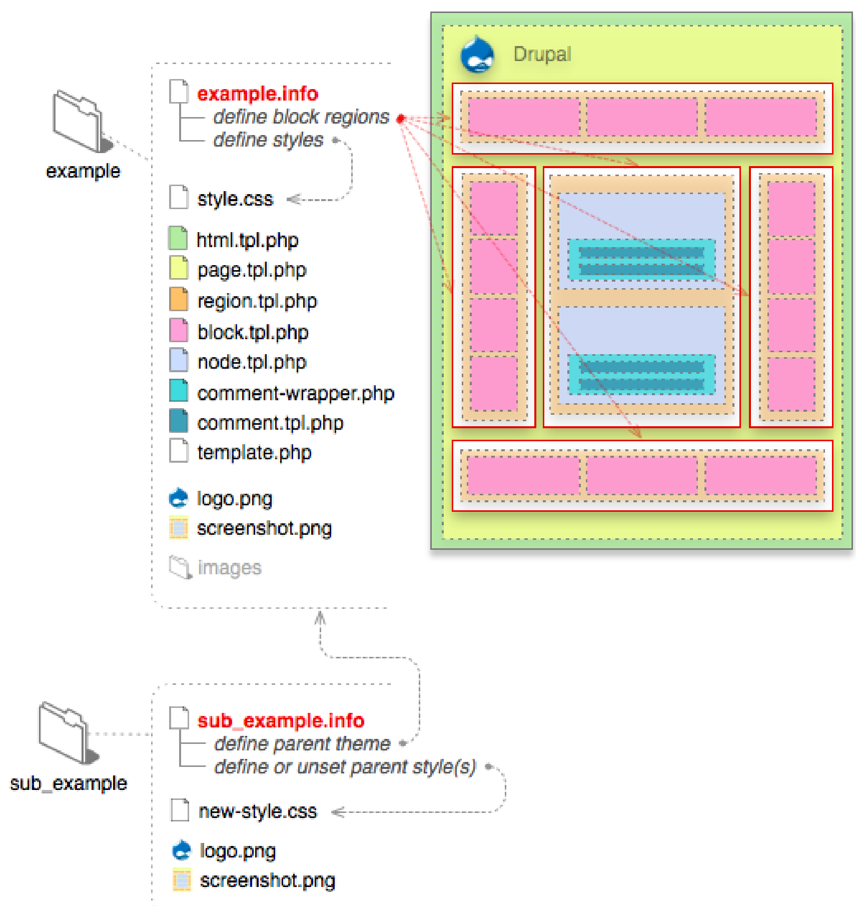
F29.1

Typical structure of the theme folder

The theme consists of template files, .tpl.php, which correspond to different elements of the page.

The style.css file is usually the main style sheet, even though the topic may define other style files.

Image found at: http://drupal.org/node/171194



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

This is the purpose of each file:

- **Definition file** (***.info***). The .info file stores the site's definition, and is mandatory. This file keeps information on available regions, CSS files, javascript, etc. The name of the .info file defines the internal name of the theme and is usually in a folder by the same name.

  For example, in the **example** folder we will find the **example.info** file. The internal name of the theme will be **example**, although it is not mandatory that the name of the folder matches the internal name of the theme, it is recommended practice.

  The .info file is a text file that can be edited with any text editor.

- **Template files** (***.tpl.php***). **tpl** files (from *template*) define the structure of a theme. These files include HTML code and PHP variables, the latter will be substituted by their matching values during the presentation of the page. Each template file will generate the output of a site element. For example, the **comment.tpl.php** includes the output code for the site's comments.

  Template files can be found in the same folder of the theme or in a sub-folder, usually called *templates,* for ease of use.

  Template files are text files that can be edited with any text editor.

- **template.php file.** We will see at the advanced level that the template.php file enables us to add additional programming logic to obtain the final value of the variables that will be present in the templates (this process is known as pre-processing of variables). This pre-processing allows us to keep the actual template files free from logic so they will only be tasked of printing the final value of the variables.

- **Style files** (**style.css** and other **.css** files). The main file that contains styles is called **style.css**. It is not a mandatory file but just common practice. Other **.css** files may exist and even be contained in a **css** folder for organizational purposes. In **paragraph 29.3** we will see a beginner introduction to CSS code, which allows to apply styles to elements in a web page.

  .css files are text files that can be edited with any text editor.

- **Images folder.** As the name implies, it contains the images of the theme. It is possible to modify existing images or add new images, in which case they will have to be correctly referenced within the .css file.

At this level we will only study modifications in templates and styles that do not require programming. At the advanced level we will extend this explanation to include creation and modification of templates with programming (add new variables to templates, create template modules, etc.).

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

29.2 Theme template files

In order to understand how themes work it is necessary to understand first how template files *(template, tpl)* work.

A template is a text file with a mix of static HTML code and PHP code snippets, which include variables or dynamic content. When Drupal loads the website, dynamic variables get substituted by their matching values, that together with static content will generate the HTML code that the browser will finally show.

For example, let's review the following template code:

```
<title><?php print $title; ?></title>
```

PHP code starts with **<?php** and ends with **?>**. The *print* instruction actually "prints" on screen the content of a variable or a text screen. In this example the content of variable $*title* is printed on screen. This variable shows the page title, a dynamic value that depends on the page that is being loaded.

Supposing that we created two pages in our site, with titles "Disclaimer" and "About Us", when we apply the template the final result will be different for each page:

"Disclaimer" page:

```
<title>Disclaimer</title>
```

"About Us" page:

```
<title>About us</title>
```

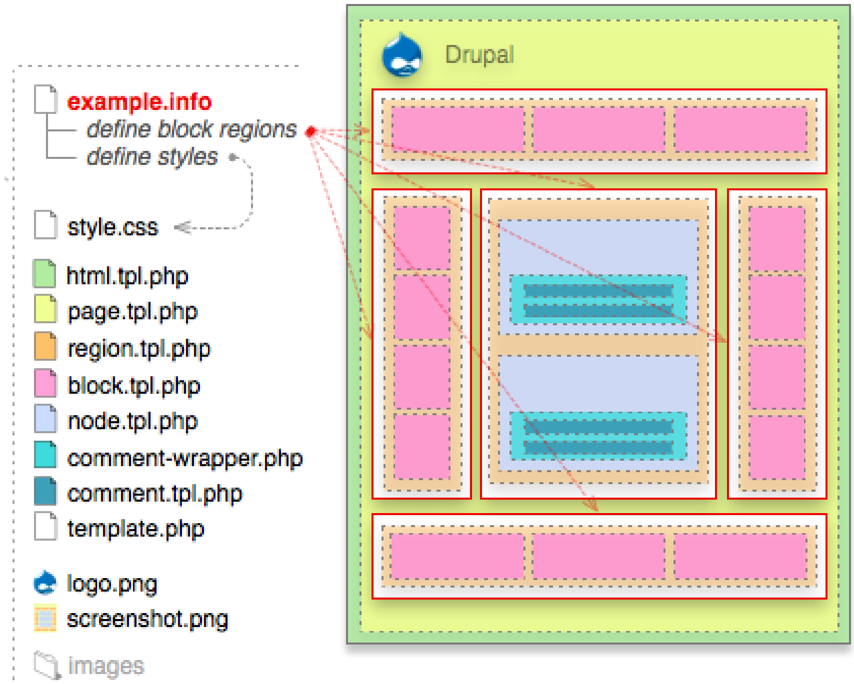
The finally generated code is static with HTML markup, it can also include CSS and javascript. This code is what is sent to the user's browser to show the website and page that is being loaded. Browsers have a "View source code" option, which allows seeing the generated HTML code.

Once the concept of template is understood, we will analyze more in depth each one of the template files that make a Drupal theme: **html.tpl.php**, **page.tpl.php, region.tpl.php, node.tpl.php** and **block.tpl.php,** among others.

Drupal has its own default template files, which it uses in case that the activated theme is missing them. Say if the theme does not have a **block.tpl.php** file**,** when it comes to show blocks, Drupal will use its default **block.tpl.php** (this file is supplied by the Block core module).

Thus if the theme that we are working on does not include a **block.tpl.php** but we want to modify the way blocks are displayed, we must copy this default file into the theme's folder. When Drupal finds the block.tpl.php in the theme's folder, it will use it.

Each template file provides the design of an element. **Figure** F29.2 the main template files are shown in colors and how they affect each element of the site.



F29.2
Theme templates

Each template file is responsible for the presentation layer of an element of the site.

html.tpl.php is responsible for setting up the HTML page and submitting the content of the template page.tpl.php, which in turn receives content from other files, such as block.tpl.php, which generates the output of the block presentation.

Working in a nested way, we get the final presentation of the page everytime it loads.

Image found at: http://drupal.org/node/171194

Following we will describe each of these files and see some of the typical variables that can be used. Variables and how to add new variables will be studied in full detail at the advanced level.

It is recommended to download a simple theme like Kanji, and open with a text editor the archives being analyzed. Use a programming-oriented text editor, which will highlight and differentiate the elements in the page, like **Notepad++**.

The html.tpl.php template

This is the highest level template file, tasked with defining the structure of the HTML page. It will be in charge of showing tags like <html>, <head> and <body>, among others. Some of the possible variables are:

- **$head_title**. Title to be sown in the <title> tag.
- **$head**. Full heading, including metatags, keywords, etc.
- **$styles**. Code to include CSS.
- **$scripts**. Code to include javascript.
- **$page**. HTML content, processed and ready to be shown in the browser.

The **html.tpl.php** file is new in Drupal 7. Drupal 6 includes this code in a file that we will see next: **page.tpl.php**.

If the active theme does not include this archive, Drupal uses its default, located at **modules/system/html.tpl.php**.

html.tpl.php

html.tpl.php file (included in the System core module).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"
  "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php print
$language->language; ?>" version="XHTML+RDFa 1.0" dir="<?php print
$language->dir; ?>"<?php print $rdf_namespaces; ?>>

<head profile="<?php print $grddl_profile; ?>">
  <?php print $head; ?>
  <title><?php print $head_title; ?></title>
  <?php print $styles; ?>
  <?php print $scripts; ?>
</head>
<body class="<?php print $classes; ?>" <?php print $attributes;?>>
  <div id="skip-link">
    <a href="#main-content" class="element-invisible element-
focusable"><?php print t('Skip to main content'); ?></a>
  </div>
  <?php print $page_top; ?>
  <?php print $page; ?>
  <?php print $page_bottom; ?>
</body>
</html>
```

The page.tpl.php template

This template generates all the code in the page, and once processed it is sent to the html.tpl.php file via the $page variable. Some of the variables that can be used are:

- **$base_path**. Path to the Drupal install. If Drupal has been installed in the root folder of the domain it will return "/".
- **$directory**. Relative path to the theme's folder. For example, it can be used to reference an image in the theme:

```
<?php print $base_path . $directory ?>/images/header.jpg
```

- **$breadcrumb**. HTML code that displays location within the site as breadcrumbs in the browser.
- **$front_page**. Link to home page.
- **$logo**. Link to logo.
- **$site_name** and **$site_slogan**. Name and slogan of the site.
- **$title**. Page title.
- **$messages**. Error messages or notifications.
- **$tabs**. Links to tabs in the page (for example, links to View and Edit a node that you are working on).
- **$node**. The *$node* object, if you are loading a page that shows a node.
- **$page['content']**. The main content of the page in HTML.
- **$page['sidebar_first']**. Content of first sidebar.
- **$page['sidebar_second']**. Content of second sidebar.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **$page['header']**. Content of header.
- **$page['footer']**. Footer elements.

If the active theme does not include this file Drupal uses the default, located in **modules/system/page.tpl.php**.

page.tpl.php

Fragment of the file **page.tpl.php** (included in the System core module).

```
<?php // $Id$   ?>
  <div id="header">
...
  </div><!-- /header -->
  <div id="container">
    <div id="inner">
      <div id="contentWrapper">
        <?php if ($page['main_menu']): ?>
          <div id="menuLeft"></div>
          <div id="primaryMenu">
            <?php print render($page['main_menu']); ?>
          </div>
          <div id="menuRight"></div>

        <?php if($page['sidebar_first']): ?>
          <div id="sidebar_first" class="sidebar">
            <?php print render($page['sidebar_first']); ?>
          </div>
        <?php endif; ?>

        <div id="center">
          <?php if (!empty($breadcrumb)): ?>
            <div id="breadcrumb">
              <?php print $breadcrumb; ?>
            </div>
          <?php endif; ?>

         <?php print $messages; ?>

          <div id="content">
            <?php print render($title_prefix); ?>
            <?php if ($title): ?>
              <h2 class="title" id="page-title">
                 <?php print $title; ?></h2>
            <?php endif; ?>
            <?php print render($page['help']); ?>
            <?php print render($page['content']); ?>
            <div class="feedicons">
              <?php echo $feed_icons ?>
            </div>
          </div>
        </div><!-- /center -->
      </div><!-- /contentWrapper -->
    </div><!-- /Inner -->
  </div><!-- /container -->

  <div id="footer">
...
  </div><!-- /footer -->
```


Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

The node.tpl.php template

This template generates the presentation of the node. The result of the template is displayed in the page template (page.tpl.php) within the variable **$page['content']**. Below are some variables that can be used in this template:

- **$title.** The title of the node.
- **$content.** The content of the node (or a fragment or advancement of the node if it is a list of nodes). It is a vector that contains all the elements of the node. You can use the function **render** to print these elements (with *render($content)* we print all the elements with *render($content[example_field])* we print only the field (example_field).
- **$user_picture.** The user image of the author of the node.
- **$date.** Date of creation of the node.
- **$name.** The user name of the author of the node.
- **$node_url.** The URL of the node.
- **$type.** The type of node (for example *page, story, blog*).
- **$comment_count.** The number of comments of the node.
- **$page.** Contains the TRUE value if you are currently viewing the entire node.
- **$teaser.** Contains the TRUE value if you are viewing the summary view of the node.
- **$promote.** Contains the TRUE value if the node is promoted to the main page (promoted to page/node).
- **$sticky.** Contains the TRUE value if the node is set at the beginning of the lists.
- **$status.** Contains the TRUE value TRUE if the node is published.

If the active theme does not include this file, Drupal uses a default theme located in **modules/node/node.tpl.php.**

node.tpl.php

Fragment of the file **node.tpl.php** (included in node core module).

```
<div id="node-<?php print $node->nid; ?>" class="<?php print
$classes; ?> clearfix"<?php print $attributes; ?>>

  <?php print $user_picture; ?>

  <?php print render($title_prefix); ?>
  <?php if (!$page): ?>
    <h2<?php print $title_attributes; ?>><a href="<?php print
$node_url; ?>"><?php print $title; ?></a></h2>
  <?php endif; ?>
  <?php print render($title_suffix); ?>

  <?php if ($display_submitted): ?>
    <div class="submitted">
      <?php print $submitted; ?>
    </div>
  <?php endif; ?>
```


Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

```
  <div class="content"<?php print $content_attributes; ?>>
    <?php
      // We hide the comments and links now so that we can render
them later.
      hide($content['comments']);
      hide($content['links']);
      print render($content);
    ?>
  </div>

  <?php print render($content['links']); ?>

  <?php print render($content['comments']); ?>

</div>
```

The block.tpl.php template

It is used to generate the presentation of blocks in a page. Some of the available variables in this template are:

- **$block->subject**. The block's subject.
- **$content**. The block's content.
- **$block->module**. The module that has generated the block.
- **$block->region**. The region where the block is located.

If the active theme does not include this file, Drupal uses its default, located in **modules/block/block.tpl.php**.

```
<div id="<?php print $block_html_id; ?>" class="<?php print
$classes; ?>"<?php print $attributes; ?>>

  <?php print render($title_prefix); ?>
<?php if ($block->subject): ?>
  <h2<?php print $title_attributes; ?>><?php print $block->subject
?></h2>
<?php endif;?>
  <?php print render($title_suffix); ?>

  <div class="content"<?php print $content_attributes; ?>>
    <?php print $content ?>
  </div>
</div>
```

block.tpl.php

Fragment of **block.tpl.php file** (included in Block core module).

Other template files

It is important to bear in mind that these three we mentioned are the main templates, but others may exist which will be in charge of other areas of the site. Here are some examples:

- **region.tpl.php**. Generates the presentation of any region of the theme. The default file for this template is located in **modules/system/region.tpl.php**.
- **comment.tpl.php**. Shows the site's comments. The default file is in **modules/comment/comment.tpl.php**.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **comment-wrapper.tpl.php**. Container for comments. The default archive is in **modules/comment/comment-wrapper.tpl.php**.

- **user-picture.tpl.php.** Template in charge of displaying the user's image. Default file is in **modules/user/user-picture.tpl.php.**

Lastly, modules can also have their own templates. For example, the **Forum** module includes several templates used to configure how forum messages are displayed. Modules can include several template files, which it will use from its own code to show the new tools it is adding to the site. If we want to modify any of these templates we must never do so directly in the template's file. We first clone the file in the themes folder, and that will be the file that we will modify, without changing the original.

Types of theme modifications

Modifications in the theme design can be performed at two levels:

- **Modifying template files** (**.tpl.php)**. We can modify templates for:
    - Change the order of existing elements, both static (HTML) and dynamic (variables).
    - Include new elements, both static (fragments of HTML and CSS) and dynamic (allowed variables in PHP, as we will see in the advanced level).
    - Delete or hide elements in the template. We can either delete or comment out part of the code, both static and dynamic, so it will not be read or displayed.

- **Modifying CSS sheets (.css)**. CSS sheets can be modified for:
    - Modify existing styles in the template..
    - Create new styles in which case they will have to be correctly referenced in the HTML code of the template.
    - Eliminate styles that will not be used.

At this level we will only see how to perform modifications based on HTML and CSS, changes related to programming will be dealt with at the advanced level.

Cache and theme modifications

When we perform modifications in the theme's files, we may need to clear the browser's cache (Ctrl + F5) and the site's cache from:

Administration ⇨ Configuration ⇨ Development ⇨ Performance [Clear all caches]

These steps may be necessary to visualize in the site changes introduced in templates or style sheets.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Introduction to web layout with HTML and CSS

29.3

In this section we are going to introduce web layout with HTML and CSS. It does not claim to encompass all possible options, but only introduce these languages so that simple modifications can be carried out in templates.

Introduction to HTML

HTML (*HyperText Mark-up Language*) is the code that makes displaying information on the Internet possible. Web browsers (Internet Explorer, Firefox, Chrome, Safari, Opera, etc.) interpret HTML code and convert it to the matching graphical representation.

In Drupal's particular case, it generates the final HTML code taking into account theme templates and site configuration (modules, content, blocks, etc.). The browser receives already generated HTML, interprets it, and displays the result on screen. This operation is repeated each time we load a page.

Actually what we will be seeing is XHTML (X from *eXtensible*), which is a better structured and updated definition of HTML.

HTML is made up of element, represented with tags. The most common structure is the following, with an opening and a closing tag <tag> </tag>:

<tag>content affected by the tag**</tag>**

Therefore, to know HTML only implies to know the set of available tags and their functionality. Some of the most commonly used tags are listed next:

- **html, head and body.** These conform the basic structure of a HTML document. All HTML pages have as a bare minimum these tags, with their matching closing tags. The HTML document starts and ends with the <html></html> tags.. Inside the document there are two main blocks, the header <head></head>, where information tags about the page are inserted, like the títle (<title></title>), and the body <body></body>, where page content is inserted. F29.3

```
<html>

  <head>
    <title>Title</title>
  </head>

  <body>
    <p>Content in a paragraph.</p>
  </body>

</html>
```

F29.3
HTML document
Basic code of an HTML page is shown.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Headings: h1, h2, h3, h4, h5** and **h6.** These tags define different levels of headings. They are applied to titles and subtitles of sections in order to structure in levels and sublevels, just like the index of any text book. By default, headings are represented in different sizes, **h1** being the largest and **h6** the smallest, but the real usefulness is not based on size, but on hierarchy. F29.4. Moreover, styles can be modified with CSS, but a correct organization of headings must be kept. F29.4

F29.4
Headers in HTML
The headings h1 to h6 structure the content of the site. They are applied to the titles of each section.

```
<h1>Section 1</h1>
<h2>Section 1.1</h2>
<h2>Section 1.2</h2>
<h1>Section 2</h1>
```

Section 1

Section 1.1

Section 1.2

Section 2

- **p, br, hr, strong, i.** With the <p></p> tags we delimit a paragraph. The
 represents end of line.
 is one of few tags that open and close on same line. <hr /> is another one, it draws a horizontal ruler used as separator. The tag highlights bold text, while the <i></i> tag italicises text. F29.5

F29.5
Example of use of HTML tags
Example of use of p, hr and br tags.

```
<h1>Setion 1</h1>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.<br />
Aliquam faucibus massa sit amet augue venenatis porttitor. <br />
Sed sit amet nunc vitae justo elementum ornare nec in leo. <br />
Pellentesque rhoncus viverra velit in ullamcorper. </p>
<hr />
<h1>Section 2</h1>
<p><strong>Lorem ipsum dolor sit amet</strong>, <i>consectetur
adipiscing elit</i>. Aliquam faucibus massa sit amet augue
venenatis porttitor. Sed sit amet nunc vitae justo elementum
ornare nec in leo. Pellentesque rhoncus viverra velit in
ullamcorper. </p>
```

Setion 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Aliquam faucibus massa sit amet augue venenatis porttitor.
Sed sit amet nunc vitae justo elementum ornare nec in leo.
Pellentesque rhoncus viverra velit in ullamcorper.

Section 2

Lorem ipsum dolor sit amet, *consectetur adipiscing elit*. Aliquam faucibus massa sit amet augue venenatis porttitor. Sed sit amet nunc vitae justo elementum ornare nec in leo. Pellentesque rhoncus viverra velit in ullamcorper.

- **List elements: ul, ol, li.** These tags represent ordered and unordered lists of elementsr: (
, *unordered list*); (
, *ordered list*). In both types of lists, elements are separated with tags. F29.6

```
<ul>
  <li>Element 1</li>
  <li>Element 2</li>
  <li>Element 3</li>
</ul>

<ol>
  <li>Element 1</li>
  <li>Element 2</li>
  <li>Element 3</li>
</ol>
```

- Element 1
- Element 2
- Element 3

1. Element 1
2. Element 2
3. Element 3

F29.6
Lists
Examples of unordered lists (ul) and ordered lists (ol) in HTML.

- **Links: a.** Some tags can contain attributes, which allow to add information to elements. For example, the **anchor** tag <a> is used to represent a hyperlink, or simply link. We point to the linked URL with the **href** attribute. The nomenclature is as follows:

```
<a href="http://www.forcontu.com/about-forcontu">About
Forcontu</a>
```

The **href** attribute can contain a complete URL (absolute path) to another site, or a URL relative to a page on our own website.

```
<a href="/about-forcontu">About Forcontu</a>
```

Another two important attributes for the anchor tag are **title** and **target**. The title attribute indicates descriptive text about the link. The target attribute indicates how the linked page will open (target="_self" indicates that the page will open in the same window as the current page, replacing it; target="_blank" indicates that the page will open in a new window).

```
<a href="http://www.forcontu.com/about-forcontu" title="More
information about Forcontu" target="_blank">About Forcontu</a>
```

- **Tables: table, tr, td, th.** Tables were used to show data in table format, creating separate distinct areas. This has been superseded by CSS, which yields better results and is recommended practice. Therefore, we will use tables exclusively to define tables of data.

In tables, the <tr></tr> tags define rows, and the <td></td> tags define columns in rows. When cells reference a heading, the <th> tag is used.

In tables we can use the **border** attribute to specify different types of borders. F29.7

F29.7
Tables
Example of creation of tables in HTML.

```
<table border="1">
  <tr>
    <th>Heading 1</th>
    <th>Heading 2</th>
  </tr>
  <tr>
    <td>Row 1, column 1</td>
    <td>Row 1, column 2</td>
  </tr>
  <tr>
    <td>Row 2, column 1</td>
    <td>Row 1, column 2</td>
  </tr>
</table>
```

Heading 1	Heading 2
Row 1, column 1	Row 1, column 2
Row 2, column 1	Row 1, column 2

- **Imags: img.** In order to insert images the <img /> tag is used. Some important attributes are: F29.8
  - **src.** Indicates the URL of the image.
  - **alt.** Indicates an alternative descriptive text of the image for text browsers or browsers that have disabled the option to show images.
  - **title.** Additional information for the image.
  - **width.** Indicates the width of the image in pixels.
  - **height.** Indicates the height of the image in pixels.

F29.8
Images
Use of images with HTML.

```
<p> Open enrollment period for Drupal 7 Expert. The course is
online and has 3 levels: Beginner, Intermediate and Advanced,
with the possibility of doing them all at once  (Expert) and
benefiting from a steep discount.</p>

<img src="/sites/default/files/forcontu-logo.png" title="Go to
Drupal course" />
```

Open enrollment period for Drupal 7 Expert. The course is online and has 3 levels: Beginner, Intermediate and Advanced, with the possibility of doing them all at once (Expert) and benefiting from a steep discount.

Introduction to CSS

In order to correctly display web pages we complement HTML with Cascading Style Sheets (CSS). HTML defines the structure of the page, while CSS takes care of defining the final styles of the elements.

There are several ways of adding styles with CSS:

1. With the style attribute, which can be applied to almost all HTML tags, this is called “inline style”.

```
<p style="font-size:14px;">Letter size is 14 píxels</p>
```

2. Using a style block in the page header, via the <style></style> tag:

```
<html>
  <head>
    <title>Page with CSS</title>

      <style type="text/css">
            h1 {font-size: 30px; font-family: arial;}
            h2 {font-size: 15px; font-family: arial;}
            p {font-size: 8px; font-family: arial;}
       </style>

  </head>
  <body>
  </body>
</html>
```

3. Separating styles in a **.css** file, linked from the web page. Next we show how to reference it.

HTML file:

```
<html>
  <head>
      <title>Page with CSS</title>
      <link rel="stylesheet" type="text/css" href="style.css" />
  </head>
  <body>
  </body>
</html>
```

CSS file (styles.css):

```
h1 {
  font-size: 30px; font-family: arial;
}
h2 {
  font-size: 15px; font-family: arial;
}
p {
  font-size: 8px; font-family: arial;
}
```

The recommended method is the one shown in point 3, linking one or more .css files which contain all the styles in the site, so that when a style sheet is modified its changes will affect all the pages using it, thus saving the work of modifying each one. This is also best practice because it allows to separate stye from webpage structure.

CSS syntax

Knowing CSS implies knowing which properties can be applied to what HTML element, and what possible values properties can take. The basic syntax is the following:
Html_element {

```
property: value;
  property: value;
  property: value;
}
```

The same property can be applied to different HTML tags. For example, the "color" property, which defines what color the element will have, can be applied to tags h1 through h6, p, a, etc.

Configuring colors and backgrounds

In this section we will see some CSS properties that allow to configure the color of elements and the background. Some of the most usual properties are:

- **color.** Defines the color of an element's font. For example, if we want to apply green to text in paragraphs <p>, we write in the style sheet:

```
p {
  color: #00FF00;
}
```

In HTML colors are usually expressed in hexadecimal format, following the #RRGGBB (red, green, blue) nomenclature. Graphic design software, like Photoshop or Gimp, is prepared to yield those values but we can also use online tools that allow to select a color and return its hexadecimal value (for example, http://www.colorpicker.com). F29.9

F29.9
Choice of colors

In many imaging programs such as Gimp or Photoshop, we can see the RGB codes in hexadecimal format of the selected colors.

We also find online services such as ColorPicker, which allows us to obtain the code of the selected color.

- **background-color.** Describes the background color of the elements. If you want to change the background color of a page we apply the property to the tag <body> but we can also apply the background color to other elements.

- **background-image.** The background-image property allows you to insert a background image:

```
background-image: url("images/background.png");
```

- **background-repeat.** Allows you to indicate whether we want the background image to repeat. Possible values are:

```
background-repeat: repeat-x;   /* is repeated horizontally */
background-repeat: repeat-y;   /* is repeated vertically */
```


Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

```
background-repeat: repeat;     /* is repeated horizontally and
vertically */
background-repeat: no-repeat;  /* is not repeated */
```

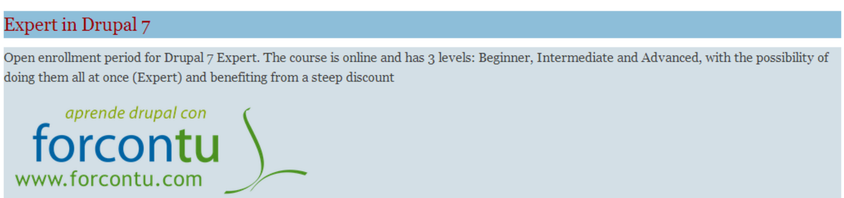
- **background-attachment.** This property allows you to indicate whether the image is fixed or scrolls with the contained element. F29.10

```
background-attachment: scroll;  /* the image moves, is not fixed*/
background-attachment: fixed;   /* the image is fixed */
```

F29.10
Background Colors

Application of the properties **color** and **background-color** to various HTML elements.

```
style type="text/css">

  body {
    background-color: #D3DFE6;
  }

  h1 {
    color: #990000;
    background-color: #8DBED9;
  }

</style>

<h1>Drupal Course</h1>
<p>Open enrollment period for Drupal 7 Expert. The course is
online and has 3 levels: Beginner, Intermediate and Advanced, with
the possibility of doing them all at once (Expert) and benefiting
from a steep discount.</p>

<img src="/sites/default/files/forcontu-logo.png"
title="Forcontu logo">
```



- **background-position.** Allows you to define the position of the background image with respect to the element that it contains. By default the image is placed in the upper left corner. The way to indicate the position is always through the coordinates x and y that can be expressed in fixed units, such as pixels, or as a percentage with respect to the element that it contains. You can also use the words: *top*, *bottom*, *center*, *left* and *right*. Some examples are:
  - **background-position: 10px 20px;** (The image is positioned 10 pixels from the top margin and 20 pixels from the left margin).
  - **background-position: 50% 25%;** (The image is positioned in the vertical center of the page and 25% toward the left margin).
  - **background-position: top left;** (The image is positioned in the upper left corner).

- **background.** This property comprises values for the above properties into a single property. Here is an example of the set of properties:

```
background-color: #FFDD35;
background-image: url("background.png");
background-repeat: no-repeat;
background-attachment: fixed;
background-position: left top;
```

It is equivalent to:

```
background: #FFDD35 url("background.png") no-repeat fixed left
top;
```

Font styles

In this section we are going to apply CSS properties related to font styles:

- **font-family.** With this property we specify an ordered list of sources which will be applied to the text of the element. This specifies a list because it is possible that the above-mentioned sources may not exist in the user's computer who is viewing the page. It is important to keep this in mind when using "special" sources. It is always best to use the more common fonts such as Arial, Verdana, Helvetica etc.

```
h1 {font-family: arial, verdana, sans-serif;}
h2 {font-family: "Times New Roman", serif;}
```

When the font name contains spaces it is indicated in quotes as is the case of "Times New Roman".

It is a good idea to complete the list of sources with a basic source to make sure that the browser will have predictable behavior if it does not have any of the above sources installed (for example, end the list with s*erif*, *sans-serif,* or *monospace).*

- **font-style.** Allows you to define the style of the text accepting the values: *normal* (without a style), *italic* (cursive) and *oblique* (tilted, depending on the source may be similar to italic).
- **font-weight.** Defines the intensity or the font weight, accepting the values Normal and bold. The value bold displays the text darker.
- **font-size.** Sets the size of the source. The size can be specified in many ways. Some of the most common include:

    - In an absolute way, in pixels (px) or points (pt):
      h1 {font-size: 30px;}
      h1 {font-size: 12pt;}

    - In a relative way, in percentages (%) or in relation to the reference value of the user's browser (em):
      h1 {font-size: 110%;}
      h1 {font-size: 2em;}

For accessibility the use of relative measures is recommended because the user will be able to change the font size in the browser and the web page will adapt to the new font sizes. In this case the user can select the value of the source database or reference in the browser. In the previous example, the h1 element has a size of 2em which is twice the size specified by the user's browser.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **font.** Finally, the font property combines some of the above properties in the following order of elements: (font-style | font-weight | font-size | font-family):

```
p {
  font-style: italic;
  font-weight: bold;
  font-size: 30px;
  font-family: arial, sans-serif;
}
```

This is equivalent to:

```
p {
  font: italic bold 30px arial, sans-serif;
}
```

Text styles

These properties are used to define the style of the text:

- **text-align.** Allows you to define the alignment of the text. The possible values of this property are:
  - left (text aligned to the left)
  - right (text aligned to the right)
  - center (text aligned to the center)
  - justify (text with full justification)

```
td {
  text-align: left;
}

p {
  text-align: justify;
}
```

- **text-decoration.** Allows you to add some effects to the text, agreeing to be bound by the following values:
  - **none** (default effect, no style)
  - **underline** (line below the text)
  - **overline** (line above the text)
  - **line-through** (crossed out)
- **letter-spacing.** Defines the spacing between the letters.

```
p {
  letter-spacing: 3px;
}
```

- **text-transform.** Allows you to add some transformations to the test, accepting the following values:
  - **capitalize** (capitalizes the first letter of each word)
  - **uppercase** (converts all letters to uppercase)
  - **lowercase** (converts all letters to lowercase)

Links styles

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

All the properties seen so far can be applied to links (tag <a>). However, the HTML links save their status (for example if the link have been visited or if the cursor is over it) and it is possible to apply specific properties for each status. The possible status of the link, also known as pseudo-classes are:

- **a** (general styles for all links, regardless the status)
- **a:link** (unvisited link)
- **a:visited** (visited link)
- **a:active** (active link)
- **a:hover** (it is activated when the mouse cursor is over the link)

The proper combination depends on the styles we want to apply to our website. Here are some examples:

```
a {
  color: #000000;
  text-decoration: none;
}
a:link {
  color: #FF0000;
}
a:visited {
  color: #00FF00;
}
a:active {
  background-color: #FFFF00;
}
a:hover {
  color: orange;
  font-style: italic;
  font-weight: bold;
  text-decoration: underline;
}
```

Identification of elements with class and id

If we use the styles as we have studied so far any change to any styles on a element will apply to all the items of the same type. For example, all paragraphs <p> will have a unique style defined in the style sheet.

In practice it will be necessary to apply some styles to certain elements of a section, but not all the elements of the same type. In Drupal, for example, we can distinguish between those elements of the region of contents and the elements that are used in the blocks. It is even possible to differentiate between the styles of different blocks.

First, let's look at how to group elements with the **class** attribute**.** In the following example we are going to define two types of links, news links and generic links.

To refer to a class in the style sheet you insert a period.

```
a.news-link {
   color: #000000;
   text-decoration: none;
}

a.generic-link {
   color: #FFCD00;
   text-decoration: underline;
```

```
}
```

In this way, we can use several classes to distinguish some links from others. The form of referencing to these styles in the <a> tag of HTML is the following:

```
<p>
Different types of links: <br />
<a href="/news1" class="news-link">Link to the news</a><br />
<a href="/contact" class="generic-link">Link to contact form</a>
</p>
```

On some occasions we have HTML elements that we consider unique and we want to apply them to a set of specific styles, styles that are not going to be reused for other elements. In these cases we use the id attribute, which has to be unique for the entire style sheet and for the entire HTML document.

To refer to the **id** of an element in the style sheet you insert # symbol.

For example, we want to apply these particular styles to the title of a page:

```
#main-title {
  color: #FFCD00;
  text-decoration: underline;
}

<h1 id="main-title">Title</h1>
```

Grouping of the elements with span and div

The tags and <div></div> are used along with the class and id attributes for grouping and structuring an HTML F29.11 document.

The tag is a neutral element that adds nothing to the document but is very useful for applying styles to certain fragments of text. For example, we could use this item to highlight in red certain fragments of text. It is also possible to apply the **id** attribute for a fragment of text only indicating a unique **id** name.

```
<style type="text/css">
  span.featured {
        color: #FF0000;
  }
</style>

<h1>Expert in Drupal 7</h1>
<p>The registration period of the course Web development with Drupal 7
is now open. The course is taught online (it can be attended from any
country) and it is divided into 3 levels: <span
class="featured">Beginner, Intermediate and Advanced.</span> There's
the possibility of doing them all (comprehensive level) and benefit
from a significant discount.</p>
```

F29.11
Span tag
With span we can apply styles to text fragments.

Expert in Drupal 7

Open enrollment period for Drupal 7 Expert. The course is online and has 3 levels: Beginner, Intermediate and Advanced, with the possibility of doing them all at once (Expert) and benefiting from a steep discount

The elements <div> allow you to create regions or blocks with different styles.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F29.12

F29.12

Div tag

It's a basic element for creating well structured HTML documents.

```
<style type="text/css">
#list1 {
  background: #0000FF;
  color: #FFFFFF;
}
#list2 {
  background: #FF0000;
  color: #FFFFFF;
}
</style>

<h1>Example using div</h1>
<div id="list1">
  <ul>
    <li>Element 1</li>
    <li>Element 2</li>
    <li>Element 3</li>
  </ul>
</div>
<div id="lista2">
  <ul>
    <li>Element 1</li>
    <li>Element 2</li>
    <li>Element 3</li>
  </ul>
</div>
```

Margins and separation between elements

To separate the different elements, we have the following properties in CSS:

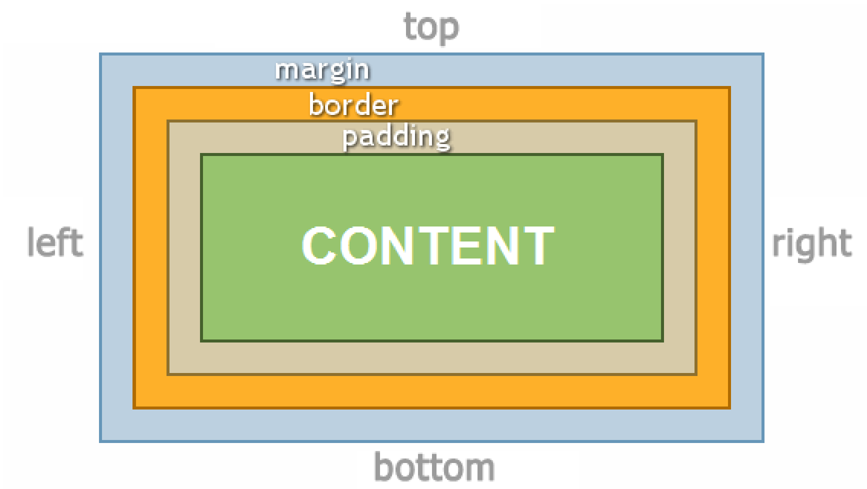
- **margin.** Defines the space with respect to the container element (prior element). It is the space between the precursor and the edge of the current element.
- **border.** Defines the edge of the element or limit.
- **padding.** Defines the space within the same item from the edge of the same content. It is also known as filler.

Each element has four sides: left, right, top, and bottom. When we apply a certain margin to an element we are doing it with respect to the page or with respect to an adjacent element. For example, if we apply a top margin of 10px to a table that is just above an image what we are doing is separating the table with a margin of 10px respect to the image. F29.13

F29.13

Differences bet padding and m

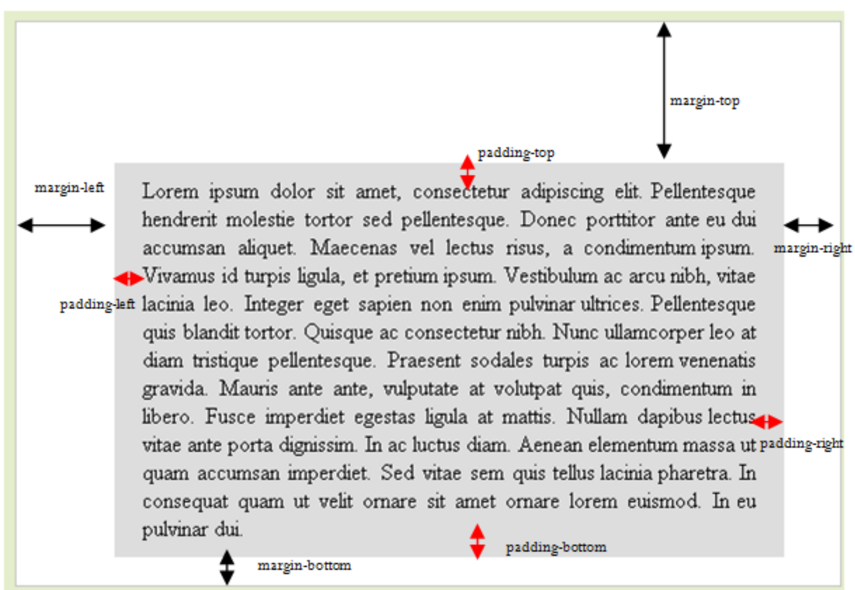
In this figure we can see which ar



Let's look at an example using margin and padding applied to a paragraph of text. F29.14

F29.14
Padding and margin
Use of padding and margin.

```
<style type="text/css">
  body {
    margin-top: 100px;
    margin-right: 40px;
    margin-bottom: 10px;
    margin-left: 70px;
  }
  #main-paragraph {
    text-align: justify;
    background-color: #DDDDDD;
    padding-top: 10px;
    padding-right: 20px;
    padding-bottom: 10px;
    padding-left: 20px;
  }
</style>
<p id="main-paragraph">
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque
hendrerit molestie tortor sed pellentesque. Donec porttitor ante eu dui
accumsan aliquet. </p>
```



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Borders

These properties are used to define styles associated with the edges of elements:

- **border-width**. Defines the thickness of the border. You can use pixels or the values *thin*, *medium*, and *thick.*
- **border-color.** Color of the border of the element.
- **border-style.** Defines the style of the border, you can apply the values: *none* (no style), *dotted, dashed, solid, double, groove, ridge, inset, outset.*
- **border.** Allows you to group the edge properties in a single property.

```
p {
  border-width: 1px;
  border-style: solid;
  border-color: blue;
}
```

It is equivalent to:

```
p {
  border: 1px solid blue;
}
```

Width and height

The **width** and **height** properties allow you to define the width and height of the elements. The elements which stipulate these properties will have a fixed width and height.

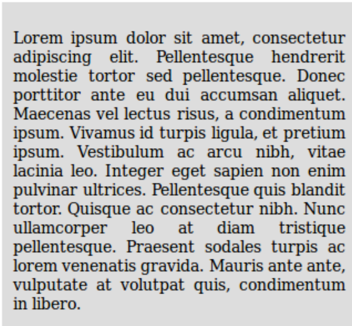
Variations of these properties are max-width, max-height, min-width, and min-height. With them we can set the minimum and maximum sizes. If we establish a maximum width, the element takes the width needed without exceeding the maximum indicated. If you specify the minimum width, the element will take at least the indicated value. You can also use the size you need from that value to be a displayed integer. F29.15

F29.15

Width and height properties

These properties set the width and height of the elements, respectively.

```
<style type="text/css">
#caja {
  text-align: justify;
  width: 300px;
  height: 300px;
  padding: 10px 10px 10px 10px;
  background-color: #DDDDDD;
  margin-left: 30px;
  margin-top: 30px;
}
</style>

<div id="box">
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Pellentesque hendrerit molestie tortor sed pellentesque. Donec
porttitor ante eu dui accumsan aliquet. Maecenas vel lectus risus,
a condimentum ipsum. Vivamus id turpis ligula, et pretium ipsum.
Vestibulum ac arcu nibh, vitae lacinia leo.</p>
</div>
```



Position of elements

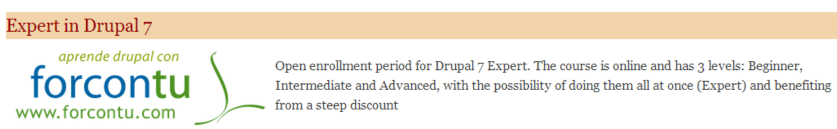
The elements can float to the right or left using the **float** property. That is to say, the element with its contents, can float either to the right or the left within a document (in relation to the element that it contains). The float property can have the values: left, right, or none. We see an example of the application of the **float** property of an image item positioning it to the left of the text. F29.16

F29.16

Float property

The **float** property allows us to place elements at the appropriate positions.

```
<style type="text/css">
h1 {
  color: #990000;
  background-color: #F3D4AA;
}
img.left {
  float: left;
}
</style>
<h1>Expert in Drupal 7</h1>
<img src="/sites/default/files/forcontu-logo.png" class="left">
<p> Open enrollment period for Drupal 7 Expert. The course is
online...</p>
```



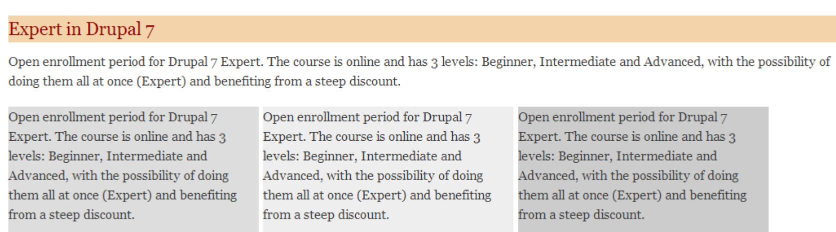
In the following example we have applied this property to create create 3 columns of text using the container <div>. F29.17

F29.17

Float and width properties

Using **float** to position elements in the positions that we want, and **width** to indicate the width we want on the page, we can position the elements in columns.

```
<style type="text/css">
h1 {
  color: #990000;
  background-color: #F3D4AA;
}
#column1 {
  float:left;
  width: 30%;
  background-color: #DDDDDD;
}
#column2 {
  float:left;
  width: 30%;
  background-color: #EEEEEE;
  margin-left: 5px;
}
#column3 {
  float:left;
  width: 30%;
  background-color: #CCCCCC;
  margin-left: 5px;
}
</style>

<h1>Expert in Drupal 7</h1>

<p> Open enrollment period…</p>

<div id="column1">
  <p> Open enrollment period…</p>
</div>

<div id="column2">
  <p> Open enrollment period…</p>
</div>

<div id="column3">
  <p> Open enrollment period…</p>
</div>
```



The **clear** property is used to control the behavior of the elements that coincide with the floating elements of a document. This property is used in conjunction with **float** when we want these elements close to a floating element (using "float") they are no longer aligned next to them. It's possible values are *left* (the element does not allow floating to the left), *right* (the element does not allow floating to the right), *both* (the floating element does not allow any sides) or *none* (the floating element allows you to use both sides).

Absolute and relative positioning

Elements positioned in an absolute way within an HTML document are not part of the general flow of the rest of the elements of the document. This is it's assigned position regardless of where the rest of the elements are in the document. By default, the position of the elements is not absolute, but relative, so it's position depends on the rest of the elements on the page. We can set the type of position of an element with the **position** property.

The position of the absolute elements are fixed by determining their coordinates, given by the attributes *top*, *left*, *right*, and *bottom*. For example:

```
#box1 {
  position:absolute;
  top: 50px;
  left: 50px;
}
```

The relatively positioned elements, in contrast to the previous, are part of the flow of the document. The origin of a relatively positioned element is at the end of the previous element in the flow.

```
#box1 {
  position:relative;
  top: 50px;
  left: 50px;
}
```


Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

29.4 Base themes and sub-themes

The base themes are designed to be extended through sub-themes. They provide functionality and basic styles that will be inherited by the sub-theme implemented. All changes to the templates and style sheets will be made exclusively in the sub-theme, keeping intact the files from the base theme.

There are many base themes available in the repertoire of Drupal themes along with the rest of the themes. Some of these base themes are:

- Zen (http://drupal.org/project/zen)
- Fusion (http://drupal.org/project/fusion)
- Omega (http://drupal.org/project/omega)
- NineSixty (960 Grid System) (http://drupal.org/project/ninesixty)
- Genesis (http://drupal.org/project/genesis)
- Basic (http://drupal.org/project/basic)

Practical case 29.1 Creation of a sub-theme with Zen

In this case study we are going to create, step by step, a sub-theme making use of the base theme **Zen**. The sub-theme will be called **mytheme**.

Keep in mind that the instructions for creating and configuring a sub-theme may vary depending on the sub-theme that we are using. For this case study we will be guided to the instructions provided by Zen, available in http: http://drupal.org/node/1010576

Step 1. Installation of the Zen theme

The Zen theme is available in (**versión 7.x-3.x**):

http://drupal.org/project/zen

We install it from:

URL Appearance
/admin/appearance

Administration ⇨ Appearance [Install new theme]

Step 2. Creating the sub-theme folder

The Zen theme includes a folder called STARTERKIT. We must copy this folder to the location where we have installed the rest of the themes, including Zen, and then rename the file using the name for our sub-theme.

The name of the sub-theme must begin with a letter and can only contain lowercase letters, numbers, and underscores.

For this case study we will use the name **mytheme**. Once this is done we will have created the folder:

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

/sites/all/themes/mytheme

which content corresponds to the folder:

/sites/all/themes/zen/STARTERKIT

A common practice, although not recommended, is to include the sub-themes within the base theme folder. We do not recommend this practice to avoid problems with updating the base theme.

Step 3. Configuration of the sub-theme

Within the **mytheme** folder, locate the file STARTERKIT.info.txt. Rename the file **mytheme.info**, also eliminating the extension .txt.

The mytheme.info file will contain the definition of the theme. Download it and open it with the text editor **Notepad++**, modifying the following parameters:

F29.18

- name = "My theme"
- description = " Sub-theme based on Zen."

```
screenshot  = screenshot.png
name        = My theme
description = Sub-theme based on Zen.

core        = 7.x
base theme  = zen

stylesheets[all][]        = css/html-reset.css
stylesheets[all][]        = css/wireframes.css
stylesheets[all][]        = css/layout-fixed.css
stylesheets[all][]        = css/page-backgrounds.css
stylesheets[all][]        = css/tabs.css
stylesheets[all][]        = css/pages.css
stylesheets[all][]        = css/blocks.css
stylesheets[all][]        = css/navigation.css
stylesheets[all][]        = css/views-styles.css
stylesheets[all][]        = css/nodes.css
stylesheets[all][]        = css/comments.css
stylesheets[all][]        = css/forms.css
stylesheets[all][]        = css/fields.css
stylesheets[print][]      = css/print.css

stylesheets-conditional[lte IE 7][all][]  = css/ie7.css
stylesheets-conditional[lte IE 6][all][]  = css/ie6.css

regions[help]           = Help
regions[sidebar_first]  = First sidebar
regions[sidebar_second] = Second sidebar
regions[navigation]     = Navigation bar
regions[highlighted]    = Highlighted
regions[content]        = Content
regions[header]         = Header
regions[footer]         = Footer
regions[bottom]         = Page bottom
regions[page_top]       = Page top
regions[page_bottom]    = Page bottom

features[] = logo
features[] = name
features[] = slogan
features[] = node_user_picture
features[] = comment_user_picture
features[] = favicon
features[] = main_menu
features[] = secondary_menu
```

F29.18

Practical case 29.1
File: mytheme.info

The file, mytheme.info, contains the information about the theme configuration.The Figure shows a fragment of this file, in which we have modified the parameters name and description.

Note: The file must be encoded in **UTF-8 without BOM** to accept accented characters. Check it out in Notepad++ using the **Format Menu** option (select the option **Codificar in UTF-8 without BOM**).

The file contains configuration information, stylesheets, regions, features, settings, etc.

The parameter of the **base theme** indicates that the current theme is a sub-theme indicated (Zen).

```
base theme  = zen
```

Upload the modified file into its folder.

Step 4. Selection of the type of design (fixed width or flexible width)

Zen lets you select between two types of design, one with a fixed width and another with a flexible width.

By default it is set to fixed width. Locate this line in the file **mytheme.info**.

```
stylesheets[all][]        = css/layout-fixed.css
```

If you want the width of the page to be flexible, change the line por:

```
stylesheets[all][]        = css/layout-liquid.css
```

Both style files are available in the folder **css** of **mytheme**.

Step 5. Editing PHP files

Download and edit the file **template.php** and **theme-settings.php** which can be found in the **mytheme** folder. These are PHP files so we should be careful if we are modifying them.

In both files replace all occurrences of the word "STARTERKIT" with the name "**mytheme**". Use the Find and Replace options of the text editor.

The files contain multiple lines with the following text:

```
  /* -- Delete this line if you want to use this setting
```

If you want to use any of the functions, we need to delete this line. This case study does not go into this level of detail, so you don't have to delete these lines.

Save the modified files to their folder again.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Step 6. Activating the sub-theme

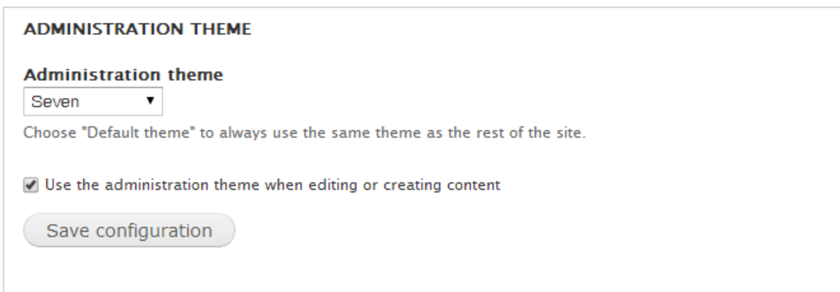
Once we make these adjustments, we can activate the theme to see how it works. This item is available once we empty the caches of the site from:

Administration ⇨ Configuration ⇨ Development ⇨ Performance [Clear all caches]

My theme will be available as disabled theme at the theme administration area (we won't activate it yet):

Administration ⇨ Appearance

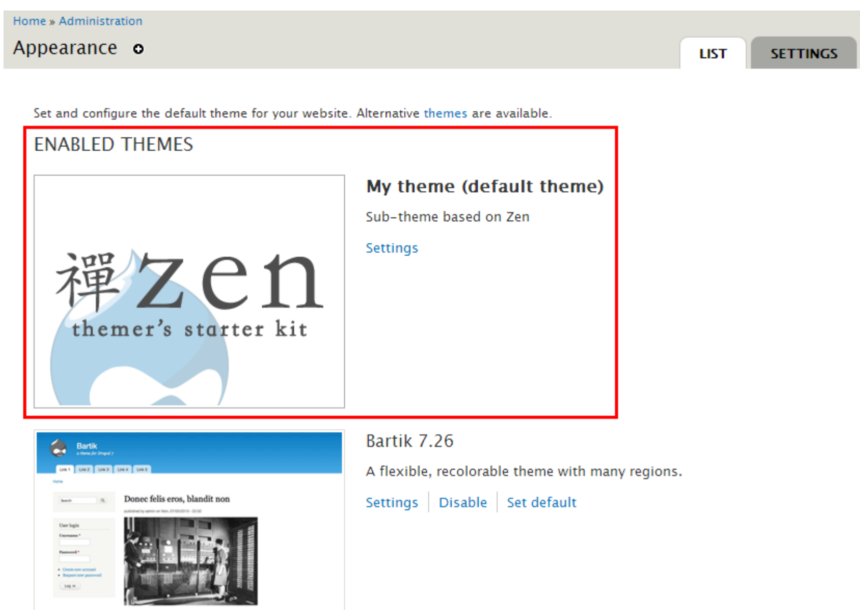
Before enabling it, we will check at the end of the same page that we are using a different theme from the default site's theme for the administration area. F29.19



F29.19
Practical case 29.1 Administration Theme

The administration theme must be different from the default theme for the site, at least during the implementation of the new theme.

Once we check this, we can activate the theme **My theme** and set it as the default theme. F29.20



F29.20
Practical case 29.1 Activation of the Theme

We will enable the theme and we will establish it as the default theme. We'll be able to see the result of the changes directly on the site

Once the theme is activated, we can navigate the site with the new design outside the administration area.

When loading, the page will display a message indicating that **the theme**

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

registry is being rebuilt on every page request. This option is suitable when we have already developed a theme. We can avoid having to empty the cache of data each time we have a change in the subject.

Once we have completed the development of this theme we will have to disable this option saving this in the file **mytheme.info**:

```
settings[zen_rebuild_registry]      = 1
```

We can also disable this option from the **Options** tab of My theme by unchecking the option **Rebuild theme registry on every page**. Although not permanently banned it will no longer display the warning message. During the development of the theme we will leave this option enabled. F29.21

F29.21

Practical case 29.1
Theme cache

During the development of the theme we will avoid to flush constantly the data cache keeping enabled the option **Rebuild theme registry on every page**.

This functionality consumes many resources and should only be used during the development of the theme.

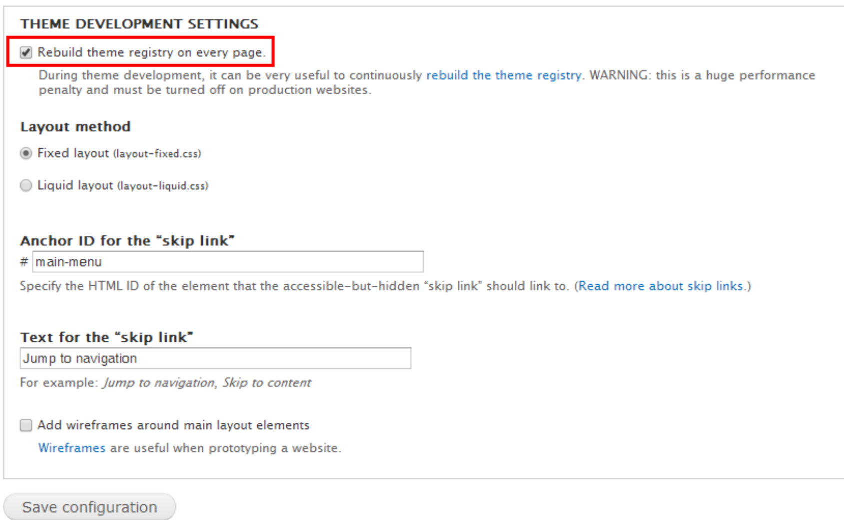
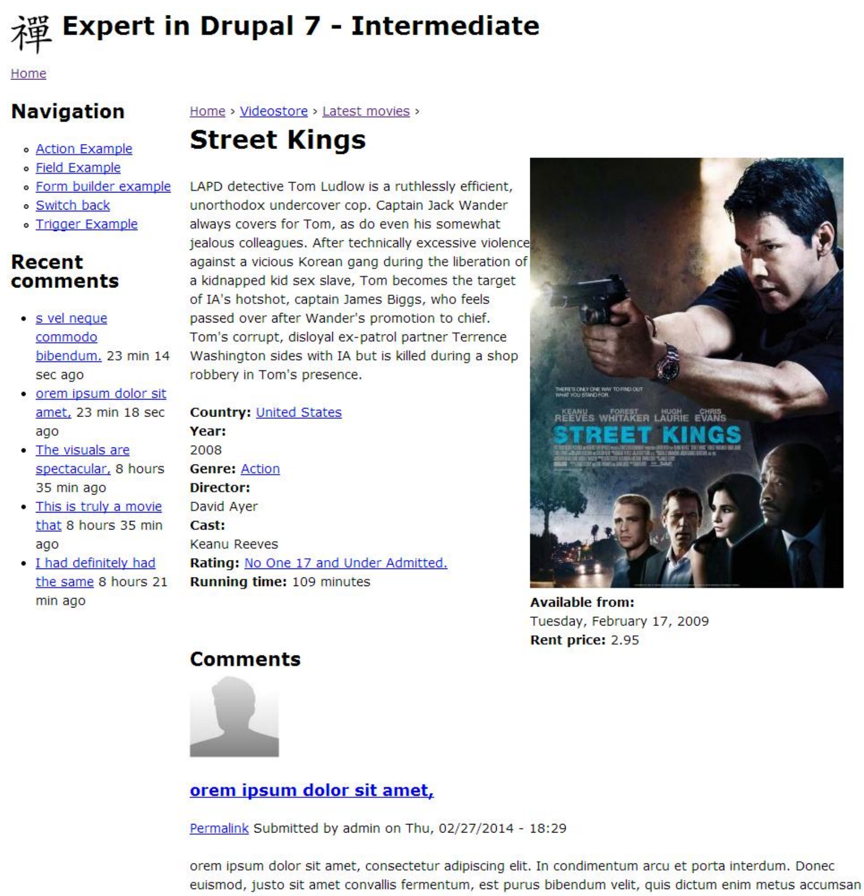


Figure F29.22 shows the page with the current design taking into account that this is a base theme to which we have not yet applied styles.



F29.22

Practical case 29.1
My theme

Presentation of the page using the theme created from Zen. Additional styles have not yet been implemented.

Step 7. Modifying the style sheet

The theme **My theme** has a **css** folder where we can find the style files with the extension **.css**.

The styles are organized by the part of the site they correspond to.

For example, the files labeled "layout", contain the styles that define the overall structure of the regions (page width, column width, etc.). Depending on whether we are using a fixed width or flexible we will modify the files **layout-fixed.css** or **layout-liquid.css**, respectively.

The file **html-reset.css** contains general settings for the HTML tags (size of letter spacing, size of headers, styles of links,etc.).

The file **nodes.css** contains specific styles for the nodes, **comments.css** for the comments, etc.

As an example we are going to **change the color and size of the page titles**. If we look in the **source code of the page** (right click and **view source**) we see that the generated HTML code for the title is the following:

```
<h1 class="title" id="page-title">Street Kings</h1>
```


Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

The title can be modified, for example, by making changes to your style **h1.title** that you'll find in the file **pages.css** because it is a common element to all pages.

In addition we'll be looking for the element in the file **pages.css** if you have defined a specific style for the item **h1.title**. We have found that this style affects the margin of the element.

```
h1.title, /* The title of the page */
h2.node-title, /* Title of a piece of content when it is given in
a list of content */
h2.block-title, /* Block title */
h2.title, /* Comment section heading */
h2.comment-form, /* Comment form heading */
h3.title /* Comment title */ {
  margin: 0;
}
```

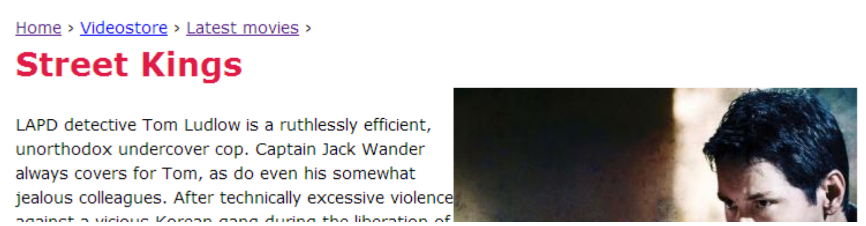
Then we will add specific styles for the label **h1.title** for example:

```
h1.title {
  color: #E01B46;
  font-size: 2em;
}
```

Once this change has been made we upload the file overwriting the previous version.

To see the changes we do not need to empty the cache of data although it could be necessary to update the cache of the browser or to refresh the page using the combination of keys (Control + F5). This option may depend on the browser used.

Now the title of the pages is displayed based on parameters of new color and size listed in the style sheet. F29.23



F29.23
Practical case 29.1
Style change
Style change made in the title of the pages. We have changed the color and size of the page title.

We are going to do a second change in the styles. We will add a border to all the images that are displayed within the region content.

The changes in content, identified with the ID **#content**, are also applied in the file **pages.tpl.php**. It adds the corresponding styles to the element **#content img {}**.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

```
/*
 * Content
 */
#content {
}

#content img {
 border: 2px solid #A9AFAD;
 padding: 3px;
}

#content .section {
}
```

With this style the images displayed within the region of the content will be displayed with an edge. In addition, we have left a gap (padding) between the image and the edge. F29.24

F29.24

Practical case 29.1
Style change

Change of the style of images. We have implemented a border to all the images that are displayed within the region content, whose identification is #content.

Step 8. Editing a template

The theme **My theme** does not contain any template file that will be saved in the folder **templates**. This is similar to **My theme** is a subtheme of **Zen**, and it utilizes the templates of your main theme, in addition to the other templates provided by the installed themes.

Within **zen/templates** we will find the following templates:

html.tpl.php
page.tpl.php
maintenance-page.tpl.php
node.tpl.php
region.tpl.php
region--sidebar.tpl.php
block.tpl.php
comment-wrapper.tpl.php
comment.tpl.php

If we want to modify any of these templates we have to copy them first from **zen/templates** to **mytheme/templates**, without changing the file name. The system looks for these templates in order, located in **page.tpl.php** in **mytheme**, it will use them in place of **zen**.

We will see a change in the order of the elements on the page, on the template **page.tpl.php**.

The breadcrumbs (variable **$breadcrumb** on the template) are shown above the title of the page. The change that we are going to do is simple, show the bread crumbs below the title (and above the messages and the tabs).

First we need to copy the template file in **My theme**:

zen/templates/page.tpl.php in **mytheme/templates/page.tpl.php**

In the template file already within My theme we have to locate the piece of PHP code that has the variable **$breadcrumb**, remove it from the current position and print it in its new location as shown below:

```
    <div id="content" class="column"><div class="section">
      <?php print render($page['highlighted']); ?>
      <a id="main-content"></a>
      <?php print render($title_prefix); ?>
      <?php if ($title): ?>
        <h1 class="title" id="page-title"><?php print $title;
?></h1>
      <?php endif; ?>
      <?php print render($title_suffix); ?>
      <?php print $breadcrumb; ?>
      <?php print $messages; ?>
      <?php if ($tabs = render($tabs)): ?>
        <div class="tabs"><?php print $tabs; ?></div>
      <?php endif; ?>
```

Once this change has been created we will see that the bread crumbs are now displayed in the desired position, below the page title. F29.25

F29.25

Practical case 29.1
Template change

We have overwritten the template page.tpl.php and changed the order of the presentation of the elements. The bread crumbs are now shown below the page title.

Therefore, the style changes are applied through the style sheets (.css files), and the structural changes are carried out in the template files (.tpl.php), always first copying the original template in the item and then performing the necessary modifications.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

30 Actions and Triggers

Drupal allows automatic execution of specific **actions** when certain events occur in the system; these events are called **triggers**.

Through these actions and triggers, we can, for example, send and email to specific site users when a node is created or modified, or block a user if certain words are detected as used in a published comment by that user.

In this unit, we'll study how the site's actions and triggers are configured once the Trigger module has been activated. We'll also see how to create and configure advanced actions.

Comparative D7/D6

Actions and Triggers

In Drupal 7, the use of actions and triggers with the Trigger module of the core is similar to Drupal 6.

If you've worked with actions and triggers in Drupal 6, you won't have any problems adapting the same functionalities in Drupal 7.

Unit contents

30.1 Introduction to Actions and Triggers 316
30.2 Simple and Advanced Actions 318
30.3 Triggers...322

30

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

30.1 Introduction to Actions and Triggers

Drupal allows the **automatic execution of actions** when specific **events** are produced in the site. So, in order for an action to be executed, a specific event must be triggered or activated. In Drupal, these events are known as **triggers**.

When a trigger is activated, the associated actions are automatically executed. When is a trigger executed? This will depend on the nature of the trigger. For example, a trigger can be activated when a node is created or modified, with the creation of a user, when executing the system's cron, etc.

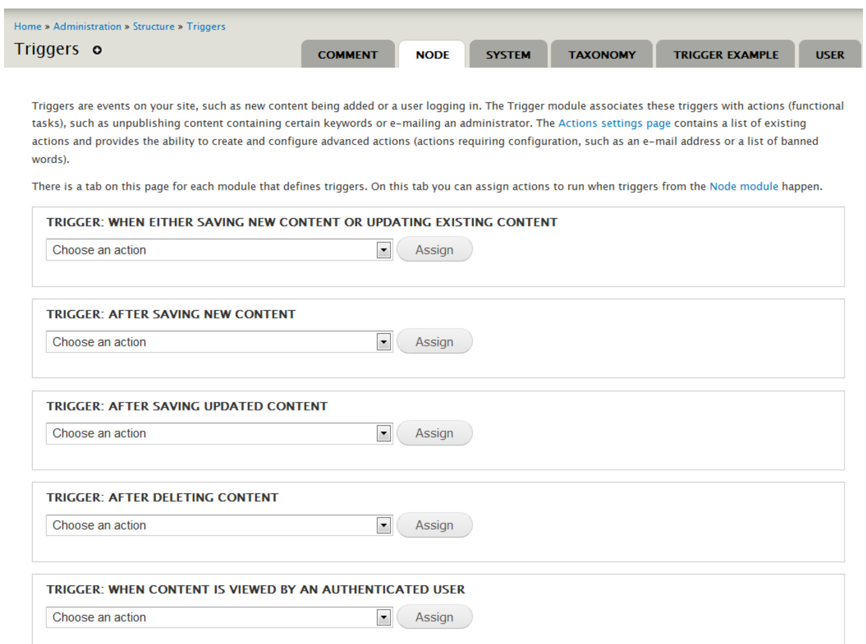
Drupal incorporates in the core, the **Trigger Module**, which is initially deactivated, that allows the actions and triggers to be carried out. Once activated, we can Access the trigger configuration from: F30.1

URL Triggers
/admin/structure/trigger/node

Administration ⇨ Structure ⇨ Triggers

F30.1
Triggers
Triggers are events that occur in the site. For example, when a user saves content, when a comment is deleted, etc.

When an event is created, we can indicate which actions will be executed.



Triggers are grouped into categories according to the element on which they act: comment, node, system, taxonomy and user.

The process is as simple as assigning a trigger to actions that we want to be executed when an event occurs. The available actions can be found under the "Choose an action" drop-down menu.

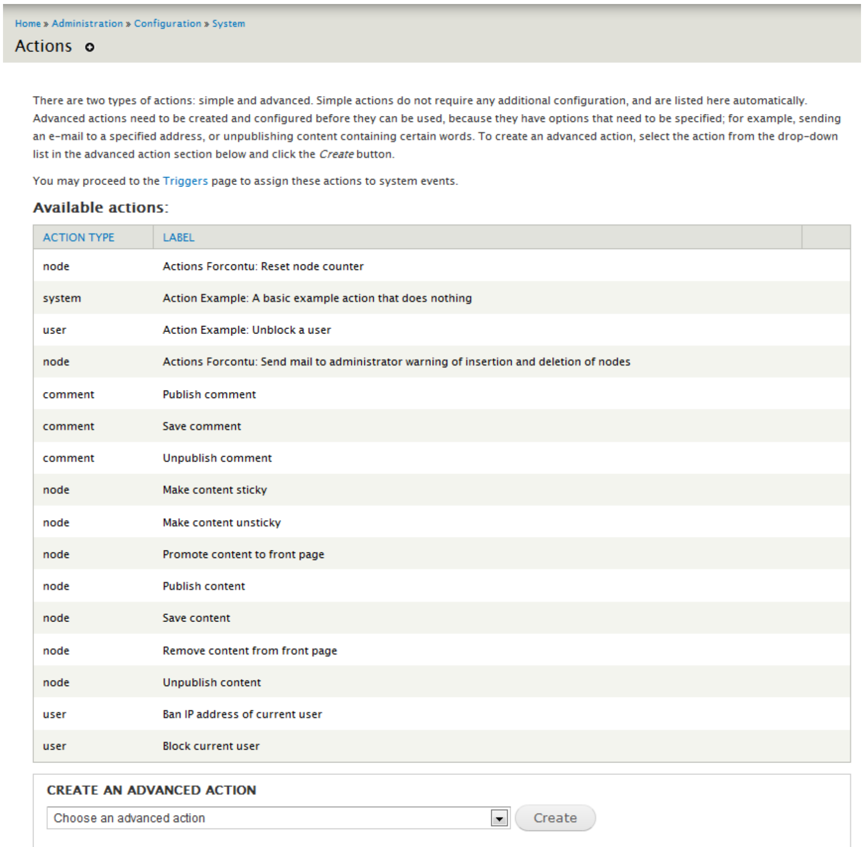
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

You can access the actions from: F30.2

Administration ⇨ Configuration ⇨ System ⇨ Actions

URL Actions
/admin/config/system/actions

There are two types of actions, simple and advanced. Like we'll see in the next section, simple actions can be used directly, while advanced actions must be created with a specific configuration.



F30.2
Actions
Assigning actions

We can look at the available actions and create advanced actions, which are actions that require additional configuration.

30.2 Simple and Advanced Actions

To see the available actions go to:

URL Actions
/admin/config/system/actions

Administration ⇨ Configuration ⇨ System ⇨ Actions

Some modules will be able to add new actions to this initial catalogue, so it would be good to review this list when a new module is installed.

The actions are organized into:

- **Simple actions**. Actions that are ready to be used without additional configuration.
- **Advanced actions**. Actions that need previous configuration to be available for use.

At the same time, actions will be assigned a type, which indicates the element over which it acts (comment, node, user, entity, etc.).

Simple Actions

Available simple actions are:

- **Comment type** actions. These actions are available only to triggers that act on comments.
  - **Publish comment.**
  - **Save comment.**
  - **Unpublish comment.**
- **Node type** actions. These actions are available only to triggers that act on nodes.
  - **Make content sticky.**
  - **Make content unsticky.**
  - **Promote content to front page.**
  - **Publish content.**
  - **Save content.** This option can be useful when we want to update the content, for example, to update the cache.
  - **Remove content from front page**. This action doesn't eliminate content, only deactivates the option Promoted to front page.
  - **Unpublish content.**
- **User type** actions. These actions are available for all triggers.
  - **Ban IP address of current user**.
  - **Block current user**.

As an example of actions added by other modules, we'll find the actions added by the **Views Bulk Operations** module; these will only be available if we've previously added the module:

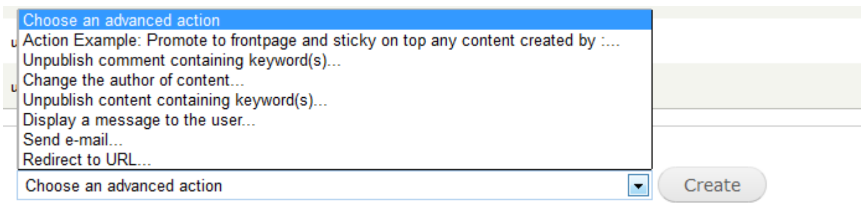
- **Entity type** actions. These actions are available only through the views that use **Views Bulk Operations**.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Pass ids as arguments to a page**.
- **Delete item**.

Advanced Actions

As we've already said, advanced actions require configuration. There is a group of advanced actions from which we can create new actions. Upon creating an advanced action, you'll need to input additional configuration information that will depend on the type of action. F30.3

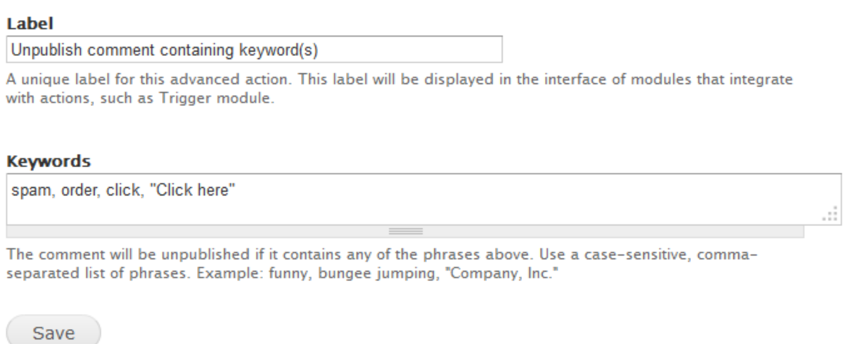


F30.3
Advanced Actions
Listing of available actions. Clicking on create will allow us to add a new action of the type indicated. We will be able to create seval actions of the same type.

In order to create an **advanced action**, it's necessary to choose a type of advanced action and to click on the **Create** button. Later we'll add the required information and we'll save the action. Each advanced action has a default Etiquette. We recommend that you change this etiquette in order to differentiate it from other actions created of the same type.

Initially, the available advanced actions are:

- **Unpublish comment containing keyword(s)**. We can indicate a list of comma-separated words. Also, strings can be included by putting them in quotes. F30.4



F30.4
Advanced Actions. Unpublish comments
Comments containing any of the indicated words or strings will be unpublished.

- **Change the author of content**. Change the content author to the indicated user name. F30.5

F30.5

Advanced Actions. Change the author of content

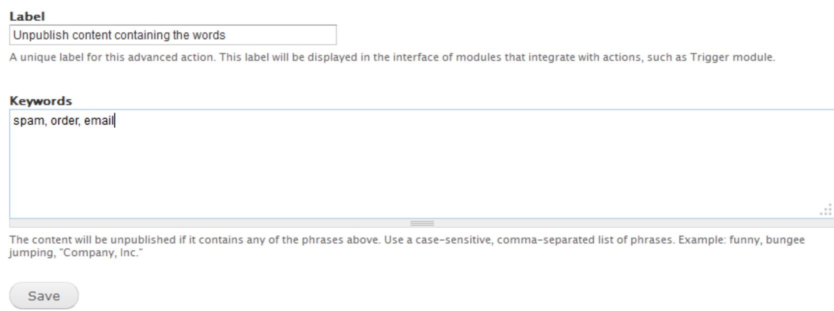
The author of the content will be changed to the name indicated.

- **Unpublish content containing keyword(s).** The content won't be eliminated; it will simply be marked as **unpublished**. F30.6

F30.6

Advanced Actions. Unpublish content

Content containing any of the indicated words or strings will be unpublished.

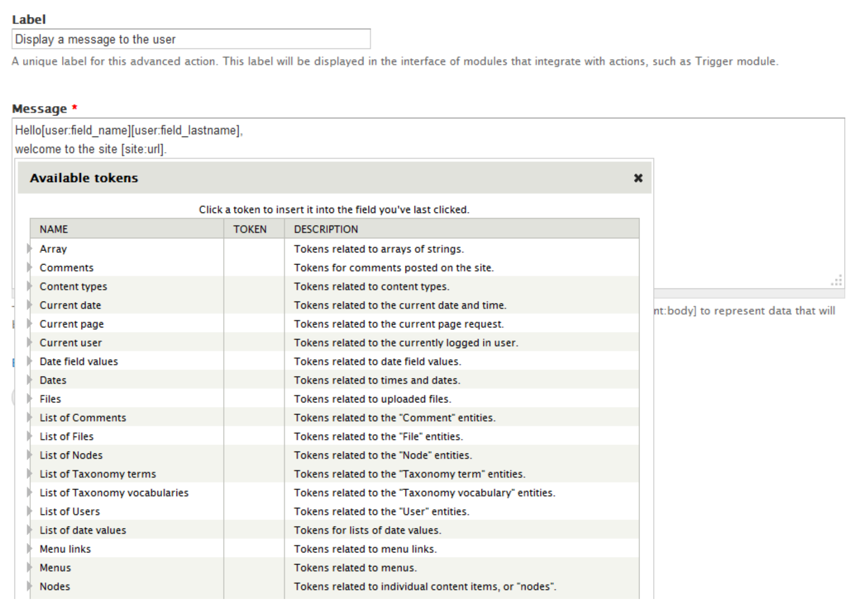


- **Display a message to the user**. The message will show as being integrated into the page, like any other message in Drupal. We can use a wild card or a variable to compose the message. F30.7

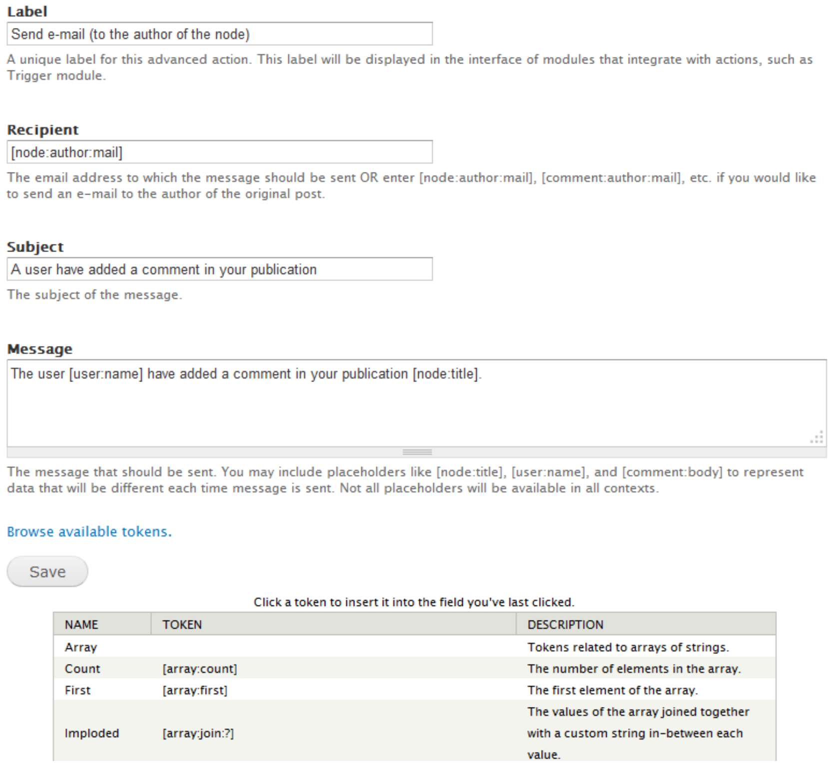
F30.7

Advanced Actions Display a message

The user will see a system message. The message might contain wild cards or substitution strings.



- **Send e-mail**. Allows you to compose a message that will be sent via email to the indicated email address. F30.8

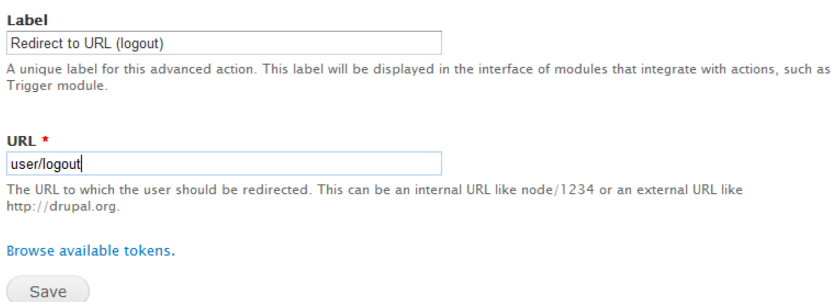


F30.8

Advanced Actions. Send e-mail

An email will be sent to the designated user. We can use replacement wild cards to establish the user or the recipient.

- **Redirect to URL**. Redirect user to the indicated URL. **F30.9**



F30.9

Advanced Actions. Redirect to URL

Redirect user who is visiting the page to a predetermined URL.

We can add various advanced actions of the same type, but with a different configuration. For example, we can create various actions of the **Send e-mail** type that can be sent to different recipients, or various actions of **Display a message to the user** type that previously will be activated under different conditions (different triggers).

30.3 Triggers

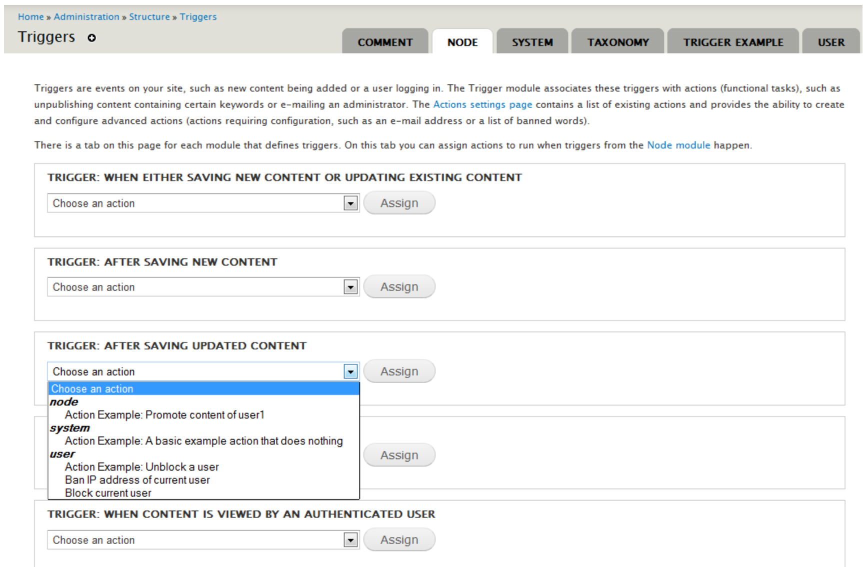
Actions will not be executed if they are not assigned to a trigger. In order to administer the triggers and to assign actions, go to:

URL Triggers
/admin/structure/trigger/node

Administration ⇨ Structure ⇨ Triggers

The available action for each trigger will depend on the type of element over which the trigger acts. F30.10

F30.10
Triggers
Administration of triggers.
Each trigger will have a group of actions available, which will depend on the type of trigger and action.



The initial available triggers are organized into five categories:

- **Comment triggers**. Events related to comments:
    - When either saving a new comment or updating an existing comment.
    - After saving a new comment.
    - After saving an updated comment.
    - After deleting a comment.
    - When a comment is being viewed by an authenticated user**.**
- **Node triggers**. Events related to nodes:
    - When either saving a new content or updating an existing content.
    - After saving new content.
    - After saving updated content.
    - After deleting content.
    - When content is being viewed by an authenticated user.

31 Rules

In Unit 30 we learned about the actions and triggers integrated in the Drupal core, through the Trigger module.

In this unit we will learn the Rules module. It is an alternative that completely replaces the previous system. By using Rules, we can define conditional actions that will be executed with the occurrence of determined events. Rules defines complex and flexible rules in order to determine if an action should or shouldn't be executed once the event occurs.

Therefore, if we use Rules, we can deactivate the Trigger module from the core, since all of its functionality is integrated within the first.

We will learn how to create rules, by assigning events, conditions, and/or actions to them. We will also learn how to create components, which are groups of conditions, actions or rules that may be reused in an encapsulated form. The components will also allow the planning of tasks that will be executed at a predetermined time.

Finally, we will learn an additional module, Webform Rules which integrates the potential of Rules with the Webform module.

Comparative D7/D6

Rules

In Drupal 7 the Rules module is completely adapted to the use of entities, for which we'll find differences in the avaiabe elements.

In addition, Rules 2.x is available only on Drupal 7, and the user interface and the parameters selection have improved in this version.

Unit contents

31.1 Installing and Configuring Rules 324
31.2 Rules and events 326
31.3 Components 342
31.4 Planning 357
31.5 Integrating Webform With Rules 359

31

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

31.1 Installing and Configuring Rules

The **Rules** module is an alternative that replaces the Trigger module from the Drupal core. By using Rules we can define the conditional actions that are executed with the occurrence of determined events. Rules, defines complex and flexible rules in order to determine if an action should or shouldn't be executed once the event occurs.

Therefore, if we use Rules, we can deactivate the Trigger module from the core, since all of its functionality is integrated within the first.

The **Rules** module is available at:

http://drupal.org/project/rules

It requires the **Entity API** and **Entity tokens** modules (included in the first), available in:

http://drupal.org/project/entity

The **Rules** module is composed of three nodules:

- **Rules Module**. Is the principal module. Checks the events that are produced in the system and whether or not the conditional actions should be executed.

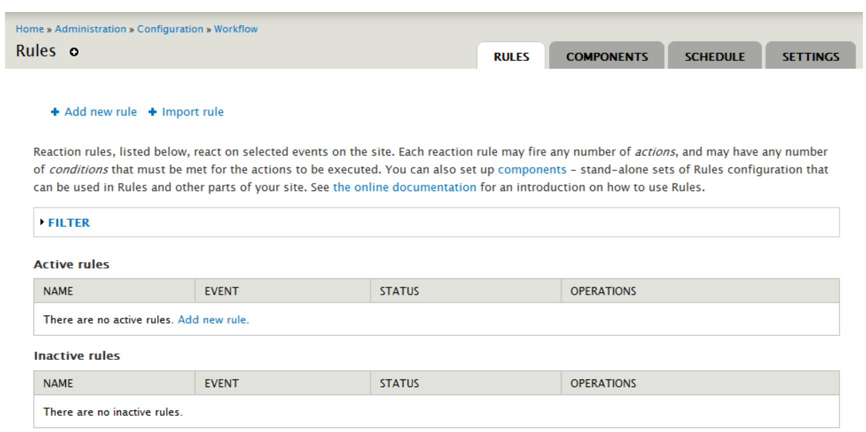
- **Rules Scheduler Module**. Allows the planning of the execution of components using actions. We will learn the components of Rules in this unit.

- **Rules UI Module**. Administration interface for the management of rules. If this module is not activated, other modules will be able to access the rules and events drive of Rules, but will be unable to manage the rules from the administration area.

We will install and activate these three modules and the required modules. Once installed, we will manage the rules from: F31.1

URL Rules
/ admin/config/workflow/rules

Administration ⇨ Configuration ⇨ Workflow ⇨ Rules



F31.1
Rules
In this unit, we will be looking at the Rules, Components, Schedule and Settings tabs that can be found in the administration area of Rules.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

The Rules administration area is divided into four tabs:

- **Rules**. Displays a list of the available rules and allows new ones to be created (**paragraph 31.2**).
- **Components**. The components are groups of rules, conditions or actions that may be reused in the Rules of Rules as well as in other modules (**paragraph 31.3**).
- **Schedule**. Allows the planning of the execution of components (p**aragraph 31.4**).
- **Settings**. General configuration options of the module. We will look at the available options next.

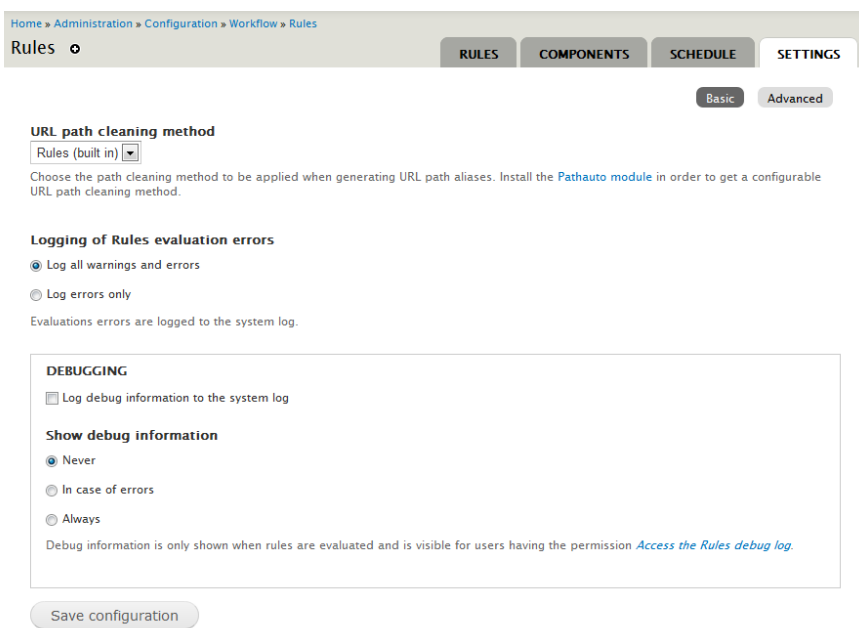
Settings

The Rules configuration options are displayed in the **Settings** tabs. All of them are debugging and errors evaluation options: **F31.2**

- **Logging of Rules evaluation errors**. The evaluation errors are logged in the system log, which may be accessed from the **Reports** options.

- **Debugging**. Stores debugging information in the system log.

It is recommended that the debugging information is stored and only displayed during the development of the website. Once it is placed in production, it is convenient to deactivate the error messages that may be displayed to the user.



F31.2
General Options
Debugging and evaluation errors options related to Rules.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

31.2 Rules and Events

We will manage the rules created in the site from the Rules tab.

We will click on the **Add new rule** link in order to create a new rule.

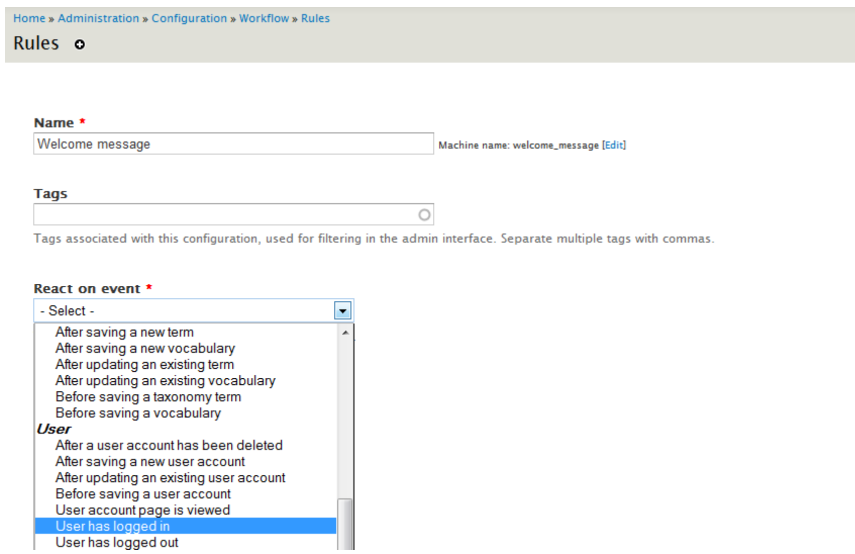
Contrary to the actions and triggers in the Drupal core, where the actions and triggers were defined separately, the Rules are associated with an event. When creating a rule we must indicate: a **name** (and **machine name**); optional **tags** to organize and filter the rules; and the **event** that will launch the rule comparison. F31.3

F31.3

Creating a Rule

When creating a rule we will indicate the optional Name, Machine name and Tags.

At the moment of creation, we may also select the event before which the rule will be evaluated. Afterward, we may add more events to the rule.



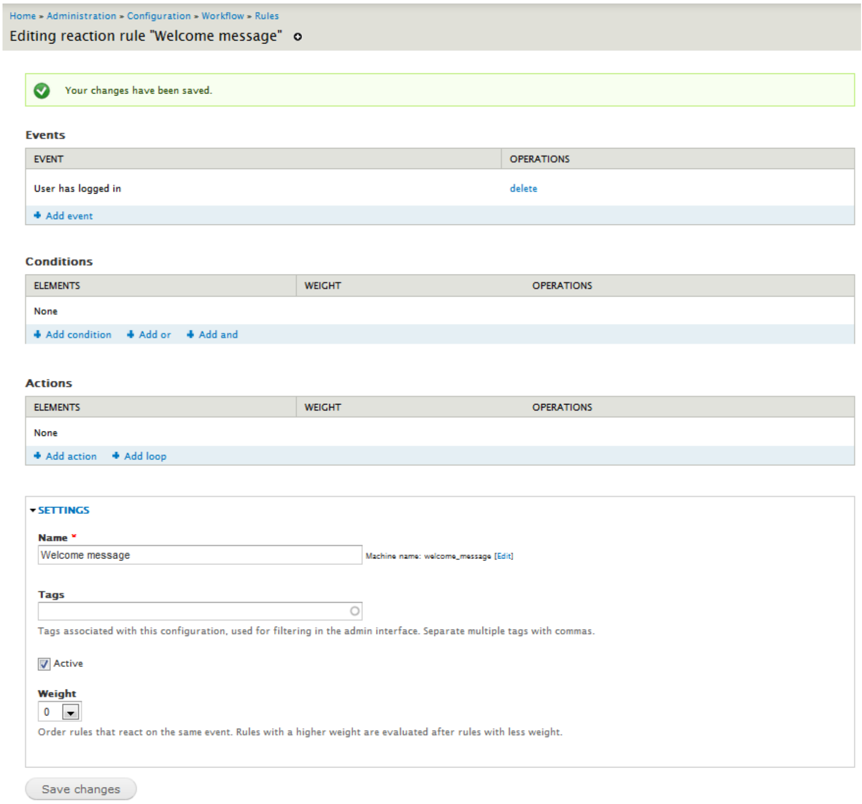
The rule will be created upon saving the changes. However, we will still have to add the following elements: F31.4

- **Events**. Initially, the event selected during creation will be displayed. We may delete this event or add other events. One rule may have several events assigned, which means it will be executed when any of them occur (and not always).

- **Conditions**. Rules that have to be fulfilled in order to execute the actions defined in the rule.

- **Actions**. The actions that are executed if the result of the Conditions evaluation is positive

- **Settings**. We may access the values that were initially introduced (Name, System Name, Tags) plus, activate or deactivate the rule and put it in order, by indicating the weight. The rules assigned to the same event will be evaluated in order from greatest to least weight.



F31.4

Rules Created

The rule has been created and now we can add events, conditions, and actions.

Events

As previously mentioned, a rule can have several events assigned. The conditions will be confirmed when any of them occurs.

The events that are initially available are:

- Comment Events:
    - **A comment is viewed**.
    - **After updating an existing comment**.
    - **Before saving a comment**.
    - **After deleting a comment**.
    - **After saving a new comment**.

- Node Events:
    - **After updating existing content**.
    - **Before saving content**.
    - **Content is viewed**.
    - **After deleting content**.
    - **After saving a new content**.

- System Events:
    - **Cron maintenance tasks are performed**.
    - **Drupal is initializing**.
    - **System log entry is created**.

- Taxonomy Events:
    - **After deleting a vocabulary**.
    - **After saving a new vocabulary**.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **After updating an existing term**.
- **After updating an existing vocabulary**.
- **Before saving a taxonomy term**.
- **Before saving a vocabulary.**
- **After deleting a term.**
- **After saving a new term.**

- User Events:
  - **After a user account has been deleted**.
  - **After saving a new user account**.
  - **After updating an existing user account**.
  - **Before saving a user account**.
  - **User account page is viewed.**
  - **The user has logged in.**
  - **The user has logged out**.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Conditions

We can add conditions to the rule through the **Add condition** link. F31.5

F31.5
Add Condition
The Add condition link allows us to add a condition to the rule.

Some of the conditions that are initially available are:

- Data Conditions:
  - **Data comparison**. Allows the acquiring of data from two fields or variables of the same type. Using it jointly with **Entity has field**, the values of the indicated field can be selected.
  - **Data value is empty**.
  - **List contains item**.
  - **Text comparison**. Compares a field with the value of the introduced text. PHP code may be used as well as replacement patterns to generate the comparison text. Also, the comparison operation can be "contains", "starts with", "ends with" or a "regular expression".

- Entities Conditions:
  - **Entity has field**. Selects a field.
  - **Entity is new**.
  - **Entity is of type**. Selects of a type of entity.
  - **Entity is of bundle**.
  - **User has field Access**.

- Node Conditions:
  - **Content is of type**. Indicates the content type.
  - **Content is published**.
  - **Content is sticky**.
  - **Content is promoted to frontpage**.

- PHP Conditions:
  - **Execute custom PHP code**. Executes a PHP snippet.

- Path Conditions:
  - **Path has URL alias**.
  - **URL alias exists.**

- User Conditions:
  - **User has role(s)**.
  - **User is blocked.**

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Data selector

When the condition requires the selection of a field or variable, a **Data selector** will be displayed. F31.6

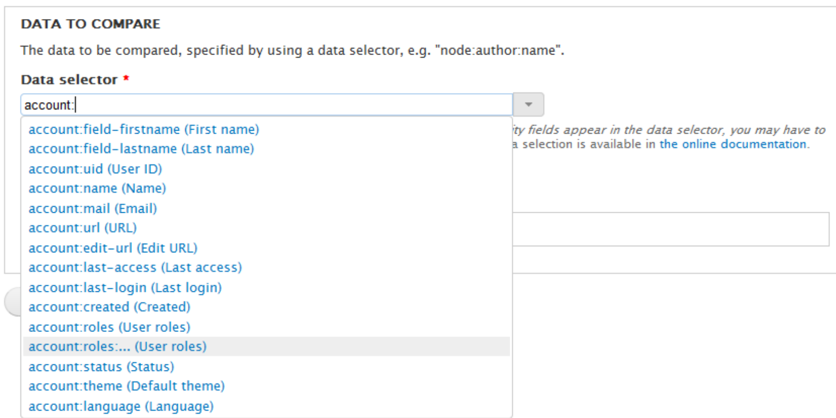
The data selector is an autocomplete field that will display the available elements at every moment. Field placement is done step by step. Upon selecting an element from the list, other related elements will drop down. For example, when selecting the object **account**, other related fields will drop (**account:uid**, **account:name**, etc.).

It is important to keep in mind that **the available fields will depend on the events** added to the rule or from other conditions.

F31.6

Data Selector

The data selector is an autocomplete field that will display the elements or available fields at every moment.



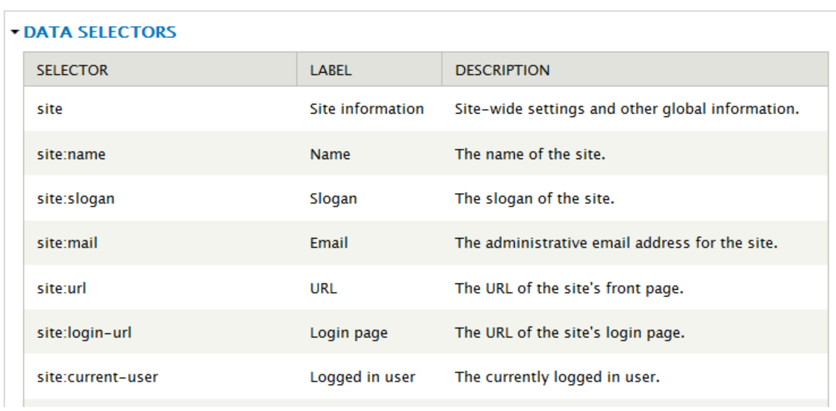
Depending on the field that we are searching for, we may go through one or many levels. For example, in order to reach the **site:current-user:roles** field, we first had to pass **site:**, then **site:current-user:** and lastly **site:current-user:roles**, the last one being a field without "subfields".

In addition to the data selector in autocomplete mode, we can also drop down DATA SELECTORS tab and use on of the available values by copying and pasting the selector in the previous text box. F31.7

F31.7

Data Selector

Displays some of the available variables that can be used in the data Selector. Keep in mind that not all of the element levels will be displayed.


DATA SELECTORS

SELECTOR	LABEL	DESCRIPTION
site	Site information	Site-wide settings and other global information.
site:name	Name	The name of the site.
site:slogan	Slogan	The slogan of the site.
site:mail	Email	The administrative email address for the site.
site:url	URL	The URL of the site's front page.
site:login-url	Login page	The URL of the site's login page.
site:current-user	Logged in user	The currently logged in user.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Conditions Negation

Upon adding a condition, we may negate it by activating the **Negate** option. In such a case, the condition will be considered True when the result of the verification is Negative.

For example:

- **The entity is a Movie type.** The condition will be considered true when the entity is that of a Movie type.

- **(Negate) The entity is a Movie type.** The condition will be considered true when the entity is NOT that of a Movie type.

Complex Conditions with AND and OR

By default, a logical operation AND will be used for the added conditions. Which means that all the conditions must be fulfilled?

We can also create complex condition groups by adding logical operators AND (Add and) and OR (Add or). We will simply add the conditions and the operators, and we will group them together by dragging and dropping the conditions in the adequate group.

Note: If the weight of each element is displayed, instead of the move/order icons, click on the "Hide row weights" link.

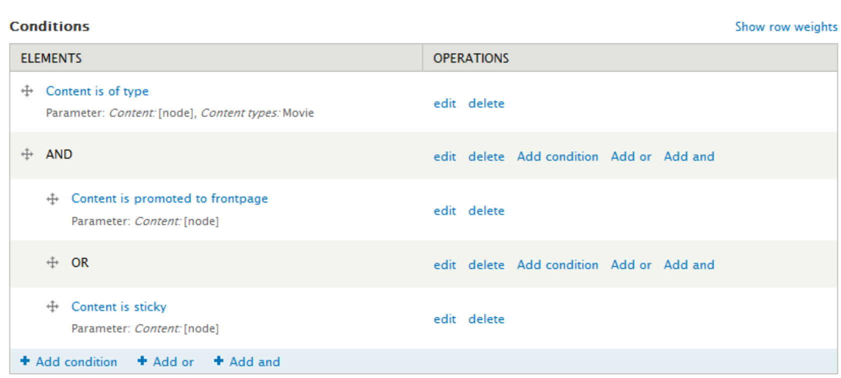
The following group can be seen in **Figure** F31.8

- Condition 1 **AND** (Condition 2 **OR** Condition 3)

The condition will be considered TRUE when one of these two combinations is achieved:

- Condition 1 **AND** Condition 2
- Condition 1 **AND** Condition 3

It will also be considered TRUE if all three conditions are met.



F31.8

Conditions AND and OR

We can create complex conditions throught the logical operators AND and OR.

Actions

Finally, we must indicate what actions will be carried out when one of the **events** is launched and the **conditions** are met.

By clicking on **Add action**, we can select an action within the available action list. F31.9

F31.9
Add action
In the Add action link we can add a rule action.

The available actions are:

- Data Actions:
    - **Add a variable**.
    - **Add an item to a list**.
    - **Calculate a value**.
    - **Remove an item from a list**.
    - **Set a data value**.

- Entities Actions:
    - **Create a new entity**.
    - **Delete entity**.
    - **Fetch entity by id**.
    - **Fetch entity by property**.
    - **Save entity**.

- Node Actions:
    - **Make content sticky.**
    - **Make content unsticky**.
    - **Promote content to front page**.
    - **Publish content**.
    - **Remove content from front page**.
    - **Unpublish content**.

- PHP Actions:
    - **Execute custom PHP code**.

- Rules scheduler Actions (will be discussed in the Planning paragraph):
    - **Delete scheduled tasks**.
    - **Schedule component evaluation**.

- Path Actions:
    - **Create or delete a content's URL alias**.
    - **Create or delete a taxonomy term's URL alias**.
    - **Create or delete any URL alias**.

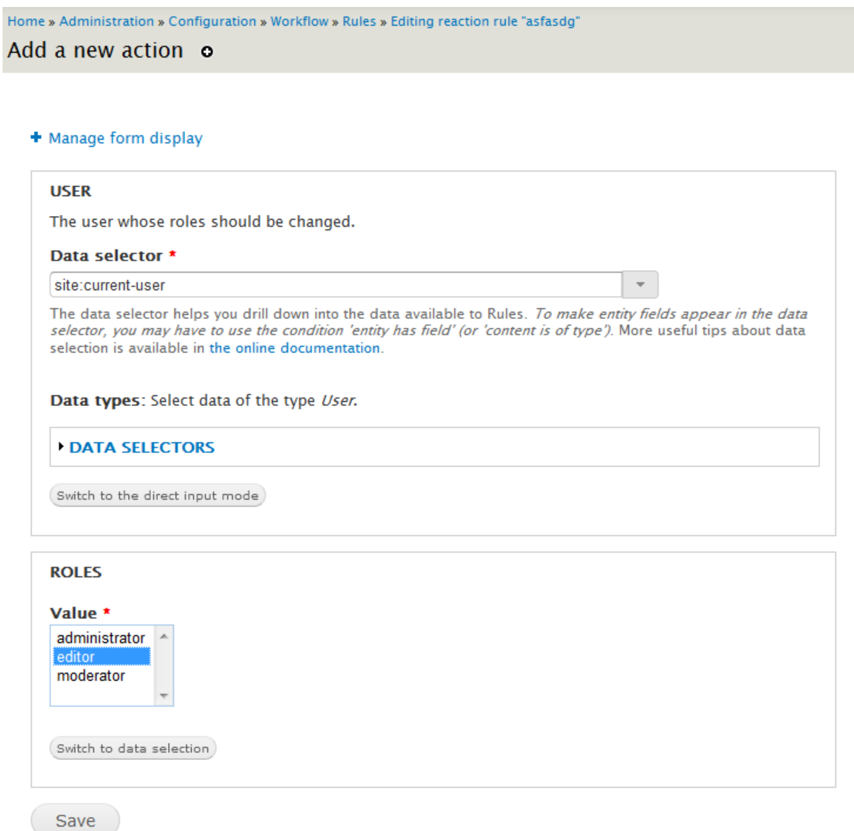
- System Actions:
    - **Send mail**.
    - **Set breadcrumb**.
    - **Page redirect**.
    - **Send mail to all users of a role**.
    - **Show a message on the site**.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- User Actions:
  - **Add user role**.
  - **Block a user**.
  - **Remove user role**.
  - **Unblock a user**.

Every action will require specific configuration values that will depend on the objects that it will act upon and on the required parameters.

For example, in the **Figure** F31.10 we are adding an **Add user role** type action. We must indicate the user who's roles will be changed (we have selected the user that is visiting the site, **site:current-user**) and the role or roles that he/she will be assigned.



F31.10
Action Configuration
Each action can have different configuration options.

We will see a complete example in **Practical Case 31.1**.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Actions on Elements List (Loop)

Through the actions loop, we can go through an elements list and execute action on each one of them. We will create an actions loop by clicking on the **Add loop** link. F31.11

F31.11

Add loop

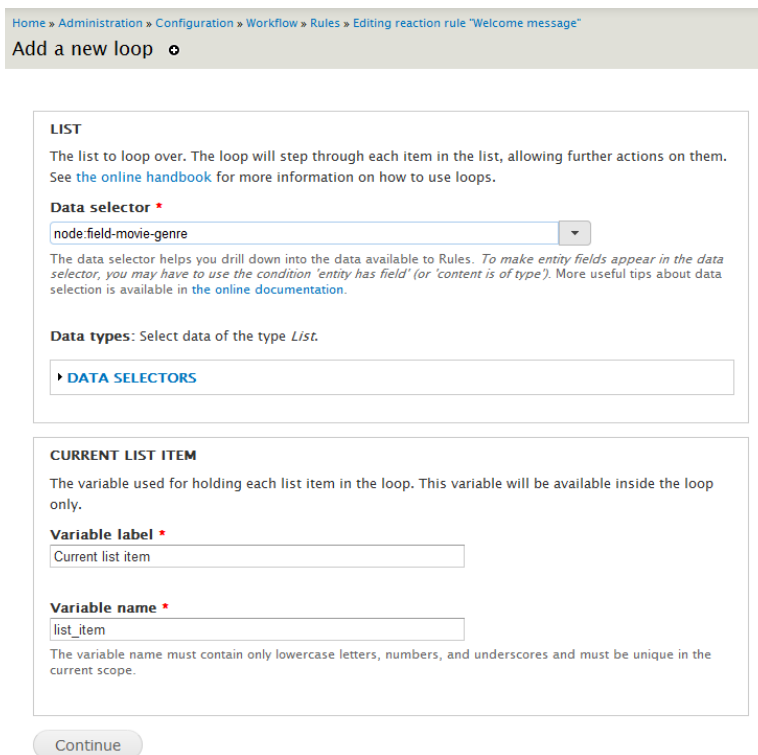
Allows us to go through all of the elements of a list and execute action on each one of them.

Once the loop is created, we must indicate the **List** with the element that will be reviewed. We will also specify the tag and name of the variable that will be used within the loop, to make reference to the active element in each iteration. By default the variable list is list_item (current Element of the list). F31.12

F31.12

Loop Configuration

We will select the object that contains the element list and the value of the variable that will be assigned to the current element of the list, while reviewing the loop.

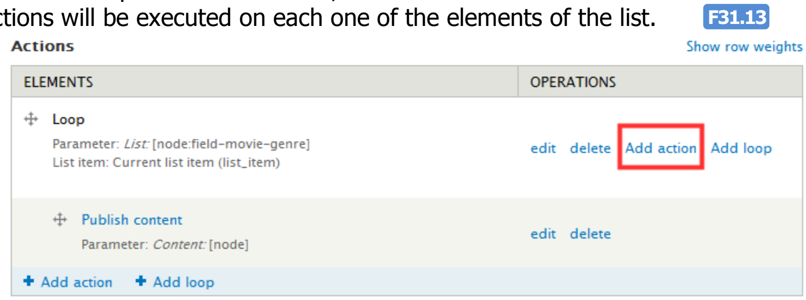


Once the loop has been created, we must create the associated actions. These actions will be executed on each one of the elements of the list. F31.13

F31.13

Loop Actions

The actions within a loop will be executed for each element on the List.



Rules Administration

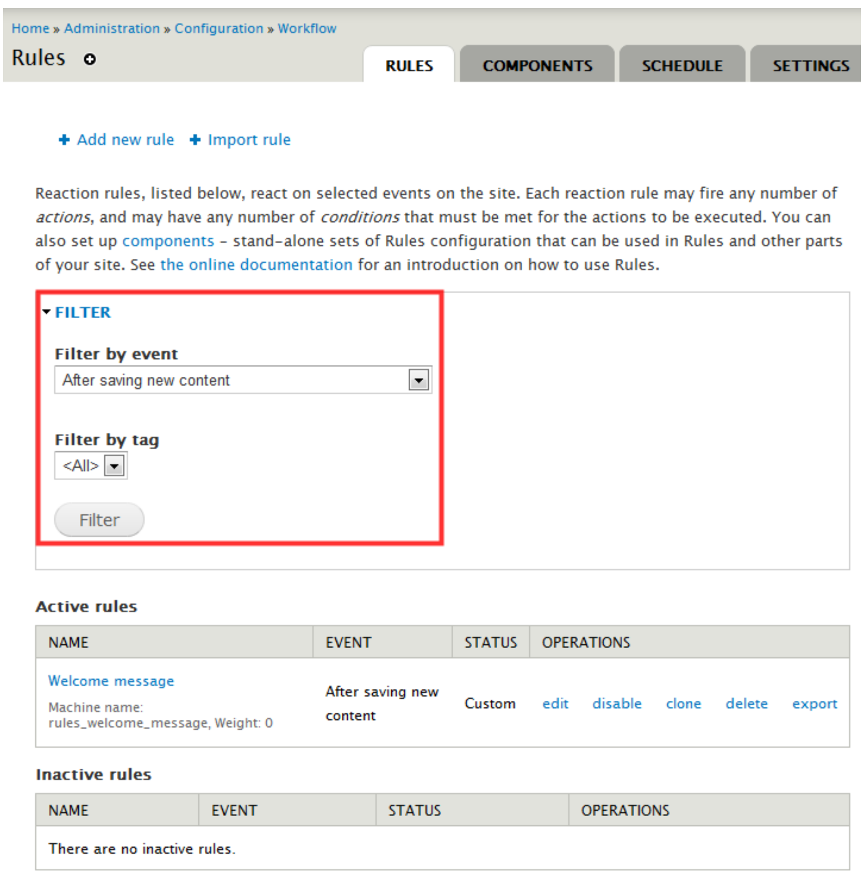
The rules that have been created will be available in the Rules administration list, in:

Administration ⇨ Configuration ⇨ Workflow ⇨ Rules

URL Rules
/admin/config/workflow/rules

Within the actions that we can put into effect upon the created rules we will find: F31.14

- **Enable/disable rules**. The active rules will be displayed in the top list and the inactive ones in the bottom list.
- **Edit a rule**. We will return to the edit options in this section.
- **Clone**. Allows the copying of a rule along with its events definition, conditions, and actions.
- **Delete**. Eliminates the rule and all of its elements. This action cannot be undone, and it is therefore recommended that rules that will be used temporarily but may be used in the future are deactivated.
- **Export**. We can export the rule in PHP code format to import it to another site or include it in another module.
- **Import**. Allows the import of a previously exported rule, whether it's from our site or from any other site with the same version of Drupal (and Rules).



F31.14
Filter rules
Whe can filter rules by event or by tag.

Since it helps with the placement and following of rules, we can filter the list by event or tag. Specifying one or several descriptive tags when creating the rule is recommended.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Practical Case 31.1 Creating a Rule

In this practice case we will create the following rule, step by step:

- When a registered user publishes or modifies a comment in a Movie type node, an information e-mail will be sent to the author of the node.

The rule will have the following characteristics:

- **Name of rule**: Notice of registered comment
- **Machine name**: notice_of_registered_comment
- **Events**:
    - After saving a new comment.
    - After updating an existing comment.
- **Conditions**:
    - The content is **Movie** type.
    - The user ha the **authenticated user** role.
- **Actions**:
    - Send mail to the node author.

Step 1. Create the Rule

We will create the rule from:

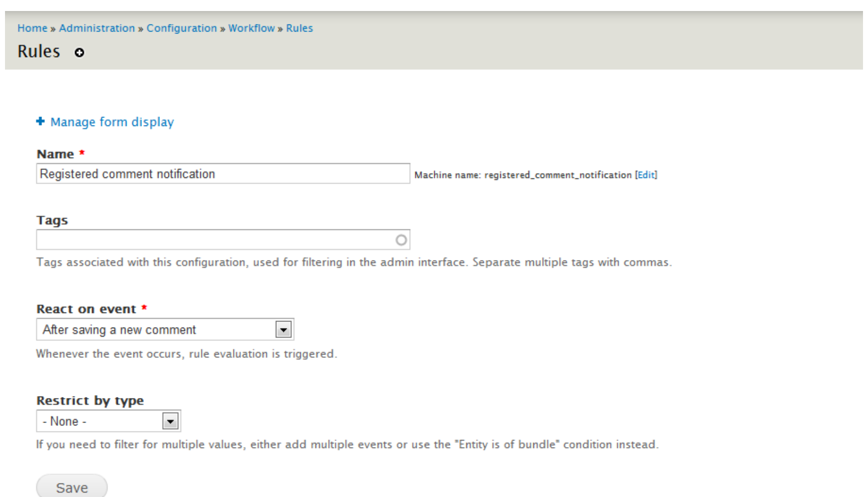
URL Rules
/ admin/config/workflow/rules

Administration ⇨ Configuration ⇨ Workflow ⇨ Rules

We will write the **Name** and **Machine name** indicated in the heading. Although the rule will react to two events, during its creation we can only indicate one of them. For example, **After saving new comment**. we will add the other event in the next step. F31.15

We will click on Save, to create the rule and continue with its definition.

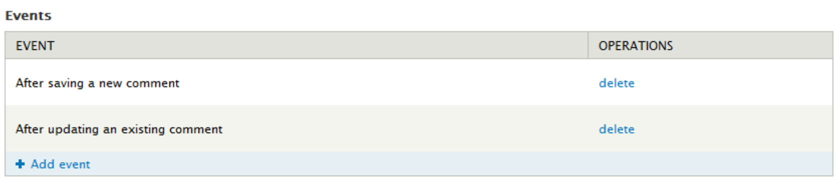
F31.15
Practical Case 31.1 Creating a Rule
We create the rule by indicating the Name, Machine name, and one of the events.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Step 2. Adding Events

The event selected during its creation will be displayed in the rules events list. In this step we will add another event by clicking on **Add event** and selecting from the available events list. F31.16



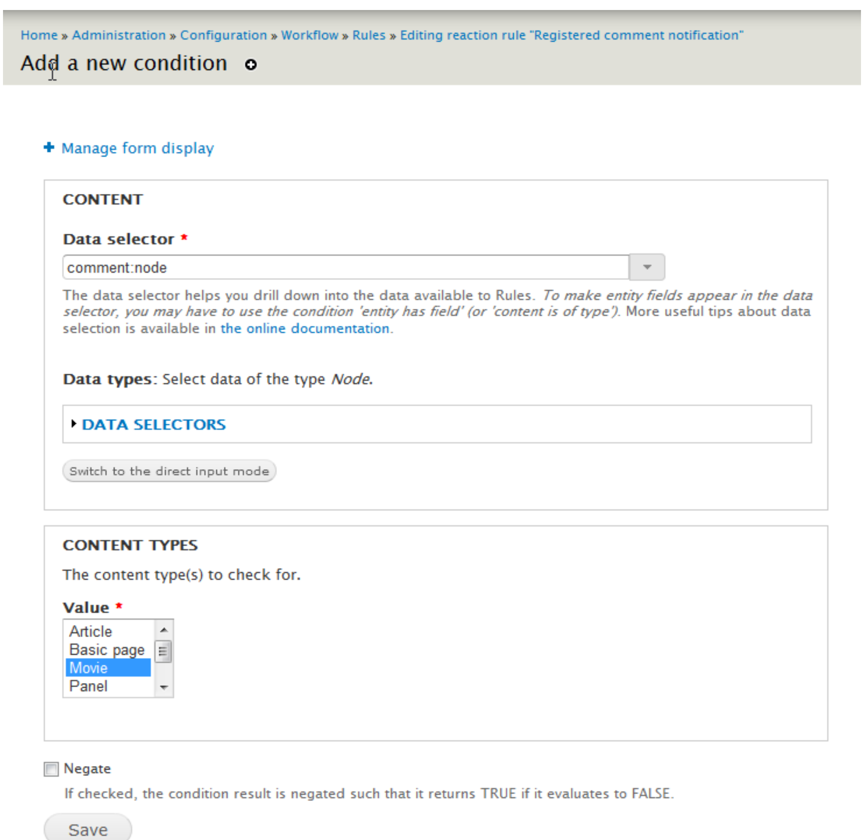
F31.16
Practical Case 31.1
Adding an Event
We add the other required event to the rule.

Step 3. Adding Conditions

We must now add the conditions from the heading. The first is that the content should be Movie type. For this condition we can use **Content is of type**. F31.17

In **Content** we will indicate **comment:node**, which makes reference to the node where the comment is being published. By default the selector already displays this value, because it considers it the most appropriate according to the events of the rule. Although in this case we do not have to modify it, we could if it was necessary.

In **content types** we will select **Movie**.



F31.17
Practical Case 31.1
Adding a Condition
In the node where the comment has been published should be a Movie type.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

The second condition that should be met is that the user should have the **authenticated user** role. For this condition we can use **User has role(s).** F31.18

In the **User** field we will select the current user (**site:current-user**), although we can also have selected the author of the comment (**comment:author**).

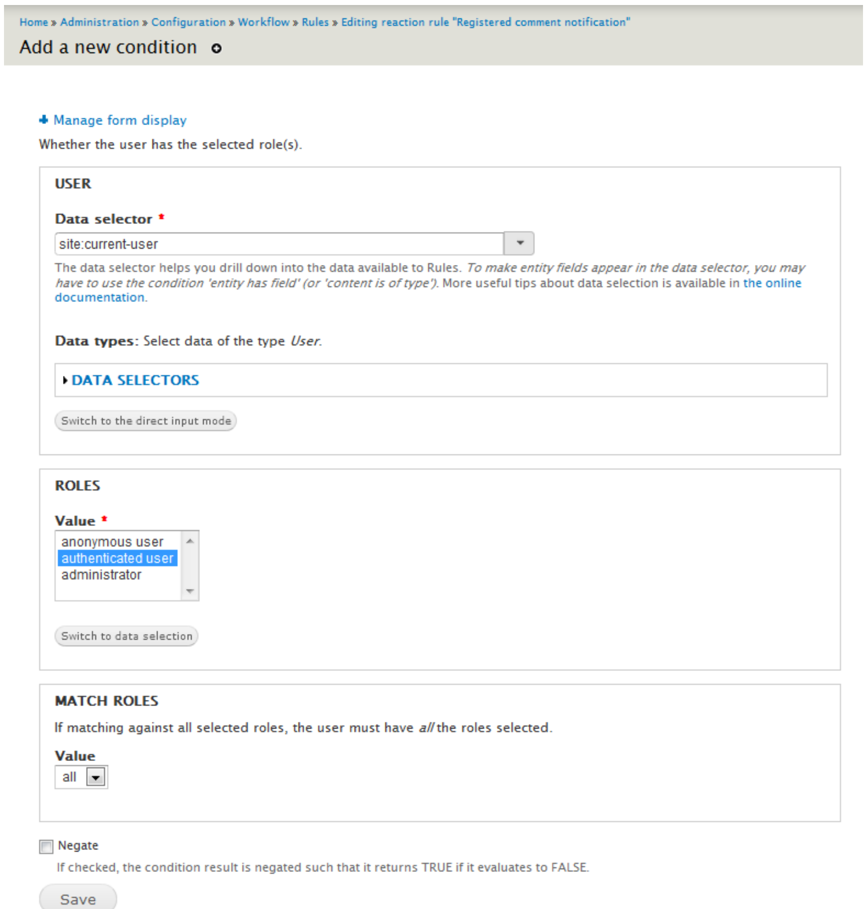
In **Roles** we will select the role that the user should have, which for this condition is **authenticated user**.

In **Match Roles** we will indicate that the user should have **all** the roles selected, even though in this case we are only verifying that it has a role.

F31.18

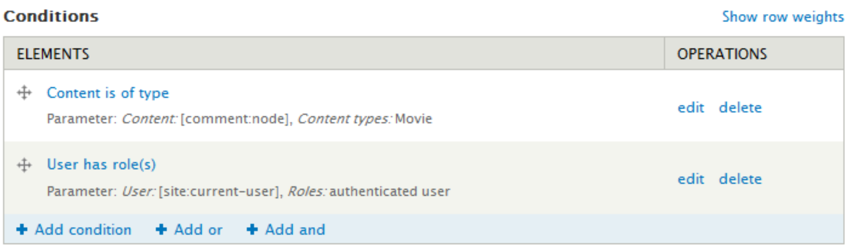
Practical Case 31.1
Adding a Condition

The current user identified on this site should have the registered user role.



The two created conditions can be seen in the **Figure**. It will not be necessary to add an operator AND, since it is the operator used by default, and all of the conditions on the list must be met. F31.19

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



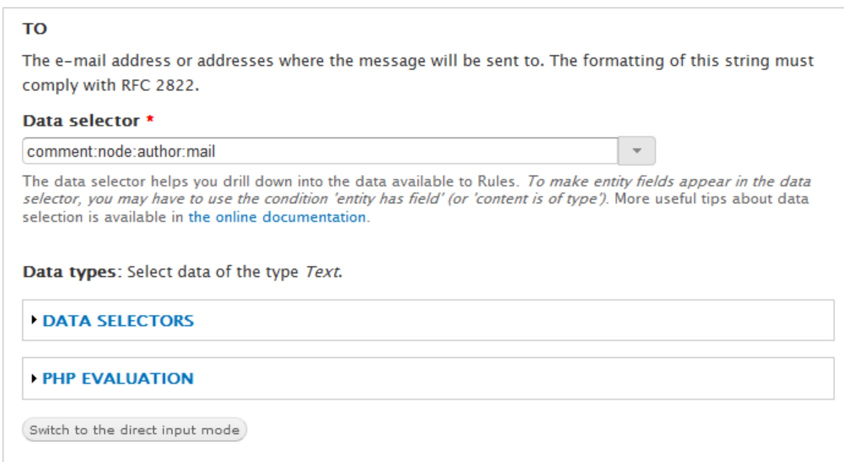
31.19
Practical Case 31.1 Conditions
If we don't specify operators AND/OR, all of the conditions must be true so that the group can be true (and by default).

Step 4. Adding Actions

Lastly, we must add the actions that we want to take place when the conditions that have been set are met. In this rule we need to define only one action, sending an e-mail to the node author where the comment has been published or edited.

We will add the action by clicking on the **Add action** link and selecting **Send mail** (within the actions group of the **System**).

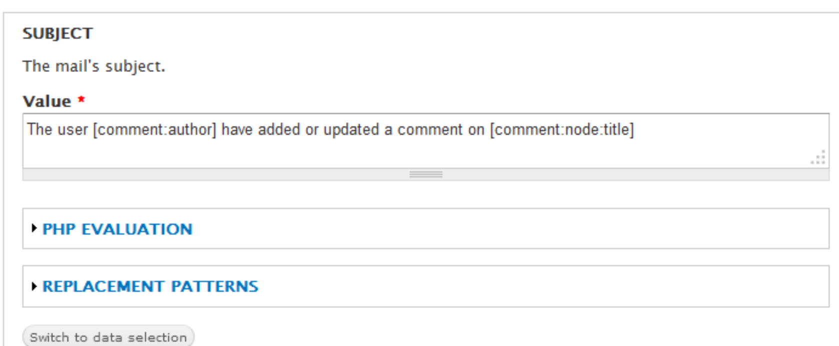
In the **To** field, we have to indicate the node author's e-mail where the comment has been published (**comment:node:author:mail**). Activate the data selection to locate the value. F31.20



31.20
Practical Case 31.1 Adding Action
An e-mail will be sent to the node author where the comment has been published.

We will find the variable that has the node author's e-mail in the TO field.

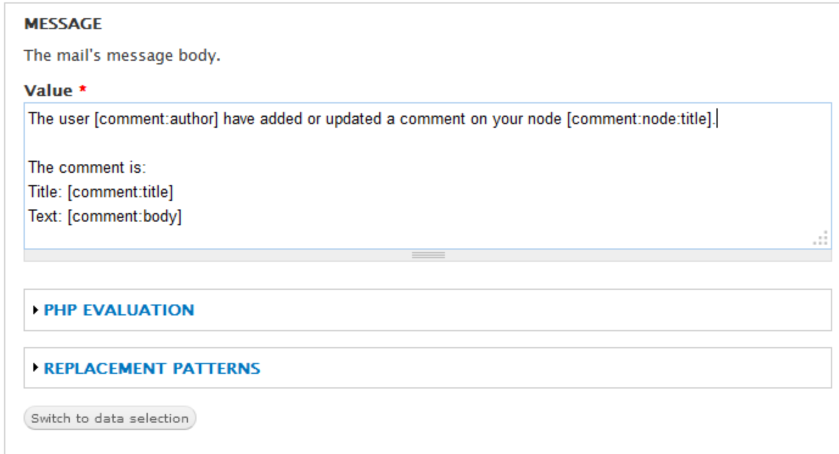
In the **Subject** field, we can write the following text: "The user [comment:author] have added or updated a comment on [comment:node:title]". Keep in mind that the available values will not always be displayed in the replacement patterns, even though we may use them. F31.21



31.21
Practical Case 31.1 Adding Action
In the SUBJECT field of mail, we may use replacement patterns.
Any available variable may be used, using brackets [variable].

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

In the **Message**, we will use replacement patterns to make reference to the user, the node title where the comment has been published, the title of the comment and the content of the comment. F31.22

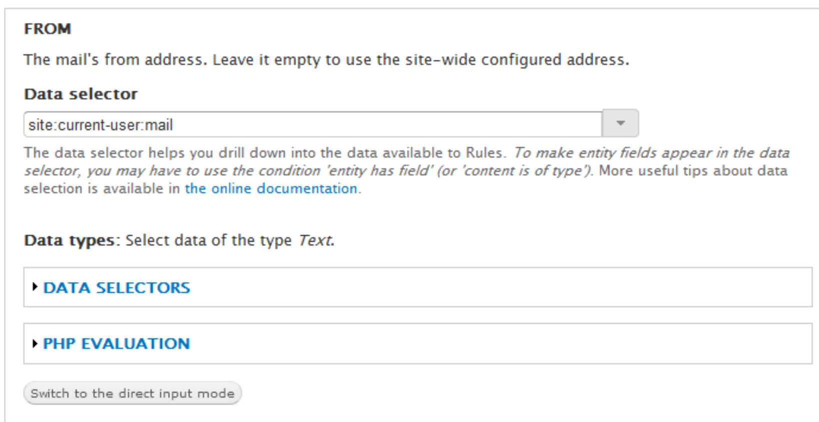


31.23

Practical Case 31.1
Adding Action

Lastly, we will indicate that the mail is sent from the current user in the FROM field.

In order to complete the action, we will indicate the site's current user's e-mail in the **From** field (site:current-user:mail). F31.23

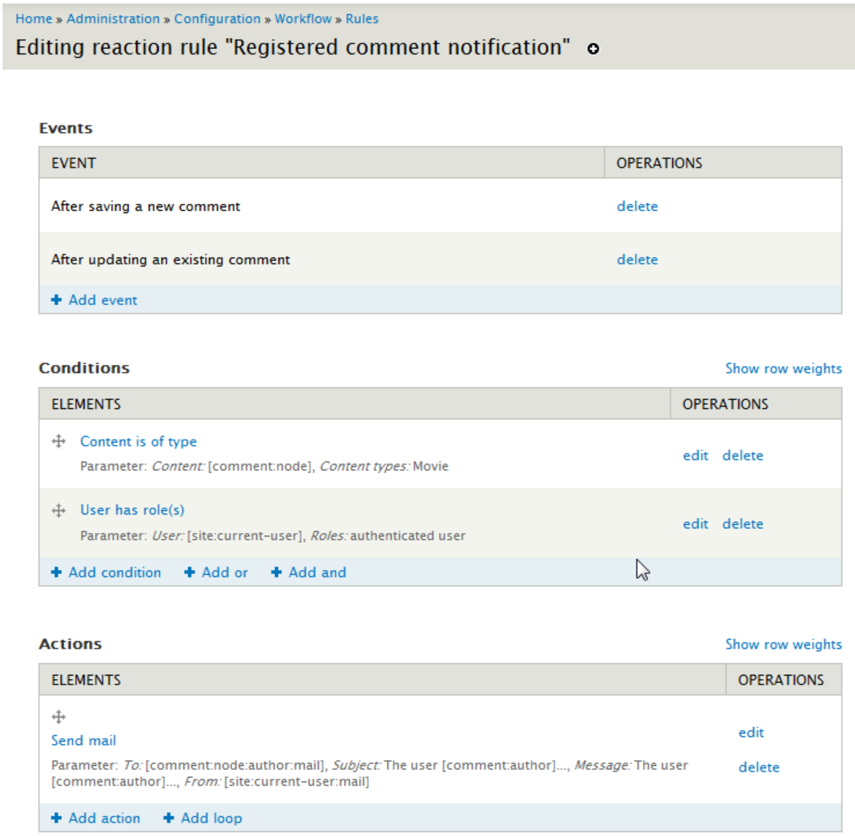


Step 5. Testing the Rule

We have defined all of the elements of the **Registered comment notification** rule, now we must put it to the test.

In the **Figure** se we can see the final configuration of events, conditions, and actions of our rule. F31.24

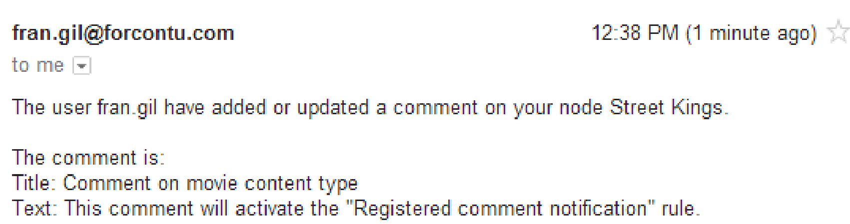
In order to test this rule, we must publish (or edit) a comment as registered user in any of the contents of Movie type. The user that created the Movie should receive an information e-mail.



Practical Case 31.1
Generated Rule

In the figure we can see the configuration of events, conditions and actions of the created rule.

When publishing a comment, the node author will receive an e-mail with the indicated Subject and Message, replacing the substitution lines with its corresponding value. F31.25



F31.25

Practical Case 31.1
Testing the Rule

The rule will be executed when the registered user publishes a comment. The node author will receive an e-mail like the one shown.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

31.3 Components

The components are groups of rules, conditions, or actions that can be reused in the Rules as well as in other modules.

A component can encapsulate a group of rules definitions, generally because we will use it in different sites. For example, a component may be a group of conditions that share many rules.

The components can be of this type:

- **Condition set (OR)**. A group of conditions where only one needs to be valid in order to validate the complete group.
- **Condition set (AND)**. A group of conditions where all of the conditions must be valid in order to validate the complete group.
- **Action set**. A list of actions that will be executed one after the other.
- **Rule**. A component that has conditions and actions but no events. Therefore, the rule component can be reused with different events.
- **Rule set**. A list of rules that are evaluated one after the other. Like the rules components, this component does not have an event assigned either.

Variables in Components

When creating a component, independently of the type selected, we must indicate the variables or **parameter** to the component. The component needs these variables to know what to do or what objects to act upon, since we can consider its contents "generic."

For example, if we want to create a group of actions that will send an e-mail to the node author, we must add a variable indicating to the component what node to use.

A variable can also be **provided**, returning a value or object, so that the function that has contacted the component can use the returned value.

For example, if we want to create a group of actions that display the last published comment by the user, it will need an entry parameter to indicate the User. We can also add an exit parameter to pick up the comment found so that the function that has contacted the component can use it.

The variable defined in the component can be:

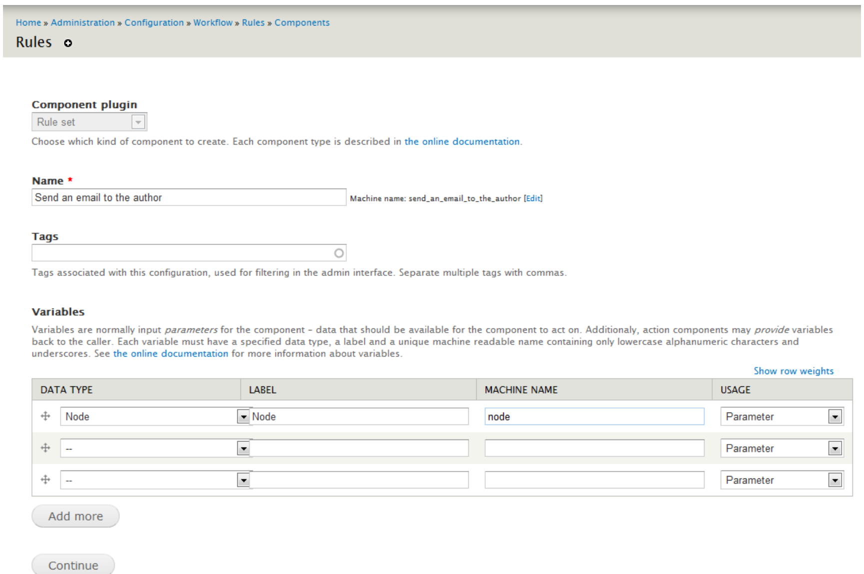
- Parameter.
- Provided.
- Parameter + Provided.

Creating a Component

In the first step of creating a component, we must complete the following fields: F31.26

- **Component plugin**. We will select one of the indicated types.
- **Name** and **Machine name**.
- **Tags**. Filters the content from the components administration area.
- **Variables**. If the component requires it, we can add variables that can be for entrance, exit or entrance/exit.

When clicking on Continue the Component will be created. The previous configuration options will depend on the type of component that we have created.



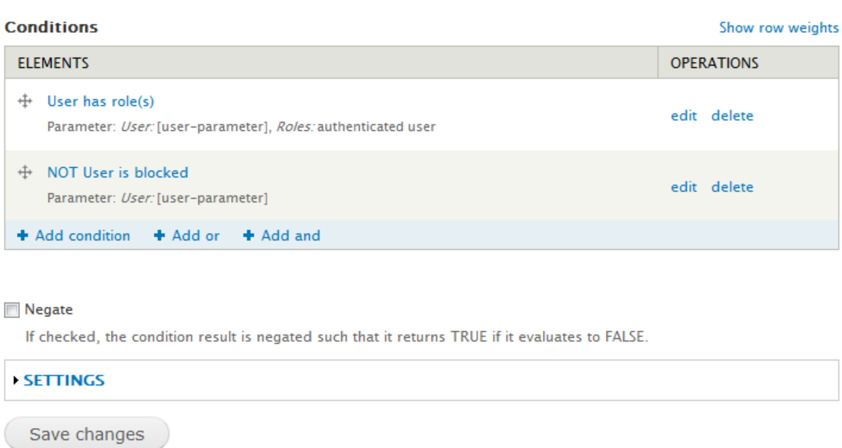
F31.26

Creating a Component

When creating a component we must indicate the variables or parameters that will get to the component.

Conditional Groups (OR/AND)

In the conditional group type components (OR and AND), we must define only the conditions. The conditions will be evaluated with the operator OR or with the operator AND in the component type function. Within the component we may also add operators AND and OR, creating more complex validation structures. F31.27



F31.27

Conditional Group (AND/OR)

A component that groups conditions only with a logical operator AND/OR.

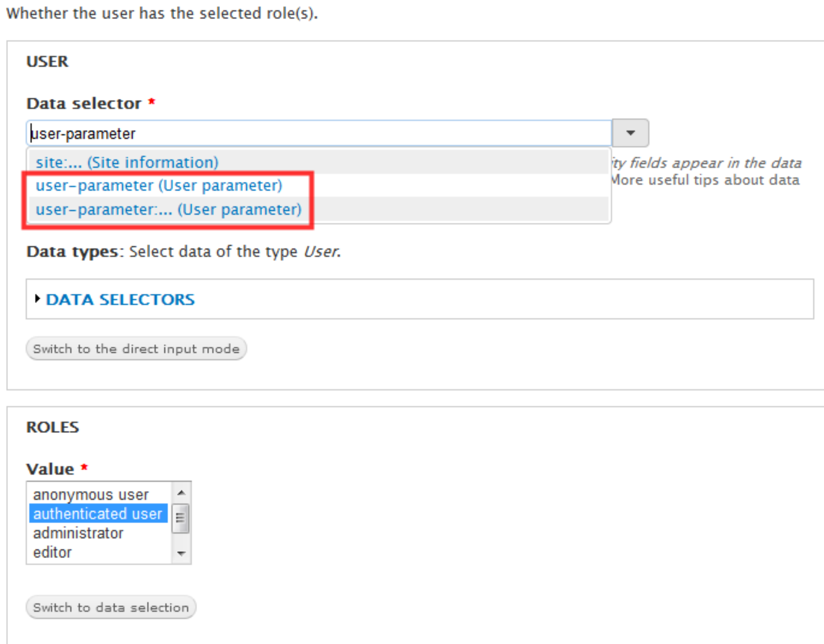
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

By adding a condition we will prove that **the component parameters can be found within the selectable values of a field.** In the **Figure** F31.28 the parameter "parameter-user" created as a variable of the component is selectable.

F31.28

Parameters in a Component

The indicated parameters will be available within the component for its use.



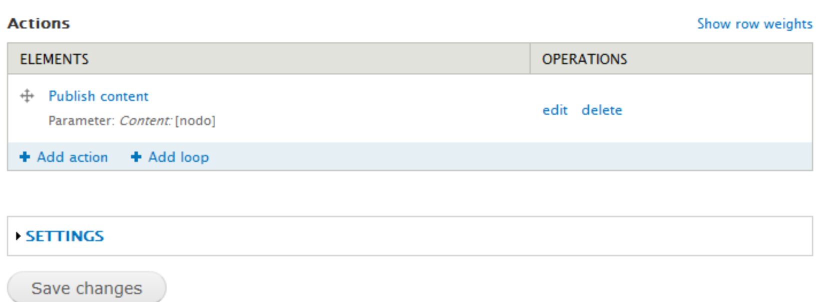
Action Set

When defining an Action Set type component, we will only have to add the actions that will be executed. This component does not have conditions or events, only the actions list that will be executed one after the other. F31.29

F31.29

Action Set

This component is made up of a group of actions that will be executed one after the other.



Again, in the added actions we may use the past variables as parameters for the component.

Rules

When creating a Rules type component, we may configure the **Conditions** and **Actions** just as we have previously done when creating the rules. As we have mentioned, the Rule type components do not have events.

Rule Set

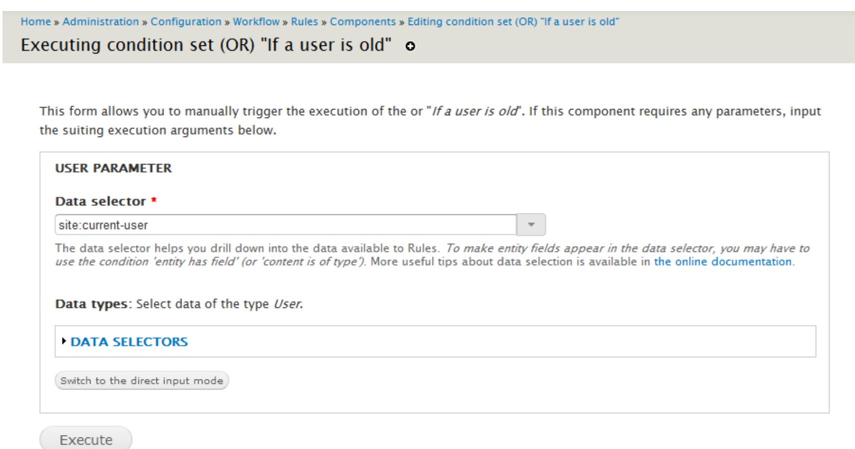
In the **Rule set** type component we can add rules, each one with its own conditions and actions. When using the component, every rule will be evaluated and if it proceeds, its actions will be executed. The rules will be evaluated in order, without them depending on each other. As a result, if one rule does not pass the validation, that will not affect the other component rules. F31.30

F31.30
Rule Set
A group of rules that will be evaluated one after the other. The Rule and Group type components of rules do not have associated events.

Manual Execution of a Component

A component can be executed manually with the **Execute** option that can be found in the components list.

If the component has assigned variables, the system will require that we indicate the value of each parameter, since the execution result will depend on the entered values. F31.31



F31.31
Executing a Component
We can execute a component manually from the execute link that can be found in the components list.

If we execute a conditional group, because it has no actions, the system will tell us if the result of the evaluation is **true** or **false**.

If we execute a group of actions, the actions will be executed directly, because the component does not to prove conditions.

If we execute a rule or group of rules, the conditions will be evaluated first, and only if the their validation improves will the associated actions will be executed.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Reusing a Component from a Rule

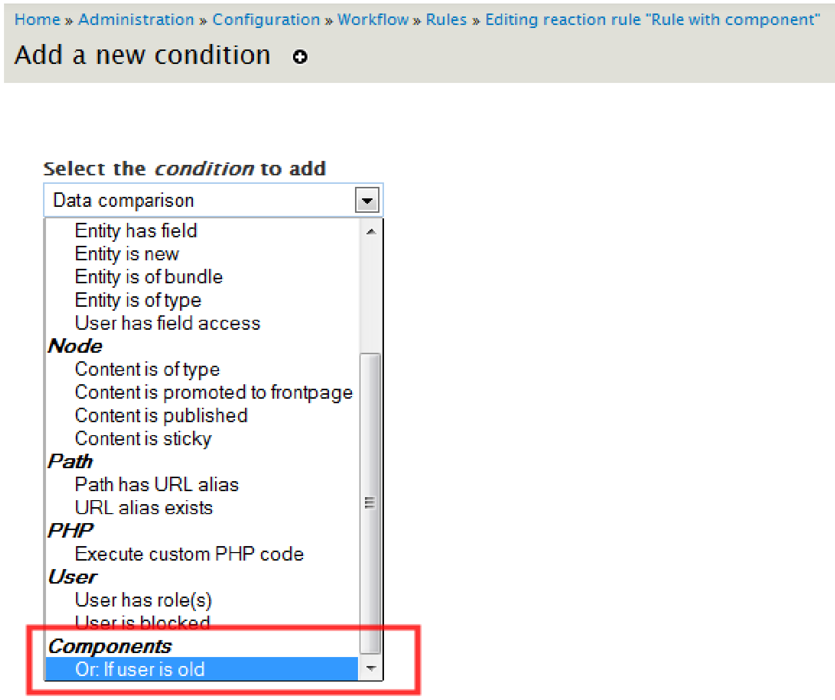
A component can be reused from a Rule or even from another component.

When adding **Condition**, on the available conditions list, a new group will be displayed called **Components**, where the components that can be included will be included. The same will occur when adding **Actions**. F31.32

F31.32

Reusing Components

The components will be available in the conditions and actions list, in the component type function.

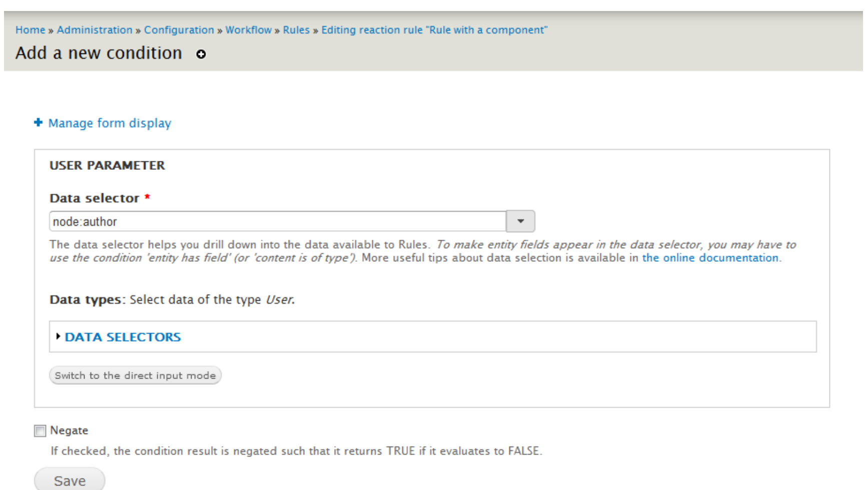


When adding a component, we will have to introduce a value for the parameter or its own parameters. The available values are the same as those for other conditions or actions, which in reality can be considered as prefabricated components that are included in the module. F31.33

F31.33

Component Parameters

To use a component, we simply have to add it or indicate the parameter values as we would do with any other element that is available by default.



Practical Case 31.2 Create a Component

In this practical case we will create, step by step, a conditional Group type component that evaluates if:

- A user is "premium": A user will be considered premium if he has been registered on the site for at least 30 days. We will confirm that the user is not blocked and does not have the administration role.

The component will have the following characteristics:

- **Component Type**: Condition set (AND)
- **Component Name**: If user is premium
- **Machine Name**: if_user_is_premium
- **Conditions**:
    - Condition 1: The user is registered.
    - Condition 2: The user does not have an administrator role.
    - Condition 3: The user is not blocked.
    - Condition 4: The user was created at least 30 days ago.
- **Variables**:
    - **User** type variable. We will call this variable **User parameter**, with the system name **user_parameter**. The variable will only be for entry (Parameter).

To verify the functionality of the component, we will execute it manually.

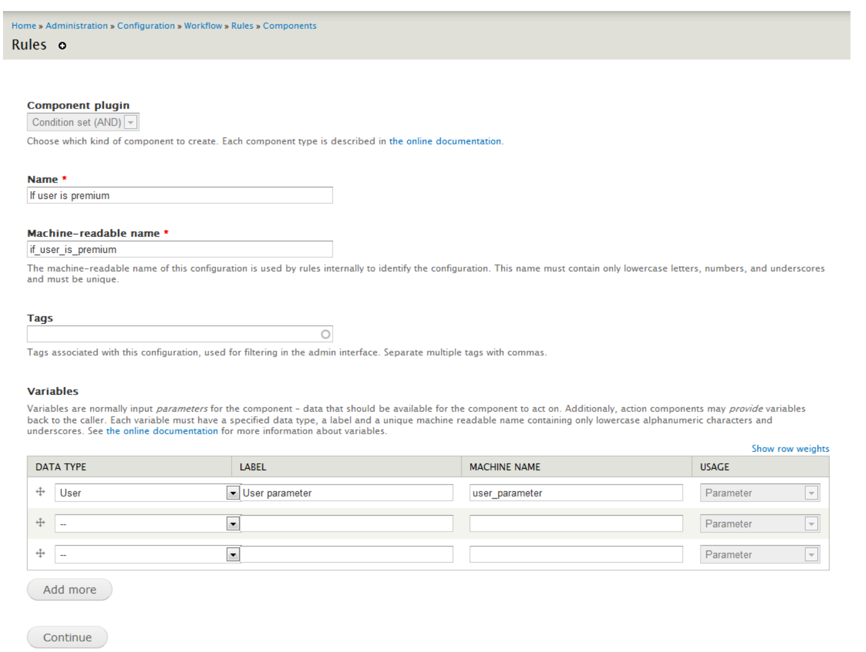
Step 1. Create the Component

We will create the component from:

Administration ⇨ Configuration ⇨ Workflow ⇨ Rules [Components tab]

URL Components
/ admin/config/workflow/rules/components

First we will select the component type **Condition set (AND).** F31.34



F31.34
Practical Case 31.2
Create the Component
We will create the component with the user type variable.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

The **Name** of the component will be "**If user is premium**", and the system name **if_user_is_premium**.

We will add the entry variable **User parameter** (**user_parameter**), **User** type.

By clicking on Continue we will have created the component.

Step 2. Create Condition 1

The first condition we must add is:

- Condition 1: The user is registered.

To add a condition we will click on **Add condition**.

The condition we must add is **User has role(s)**, in the **User** group of the dropdown.

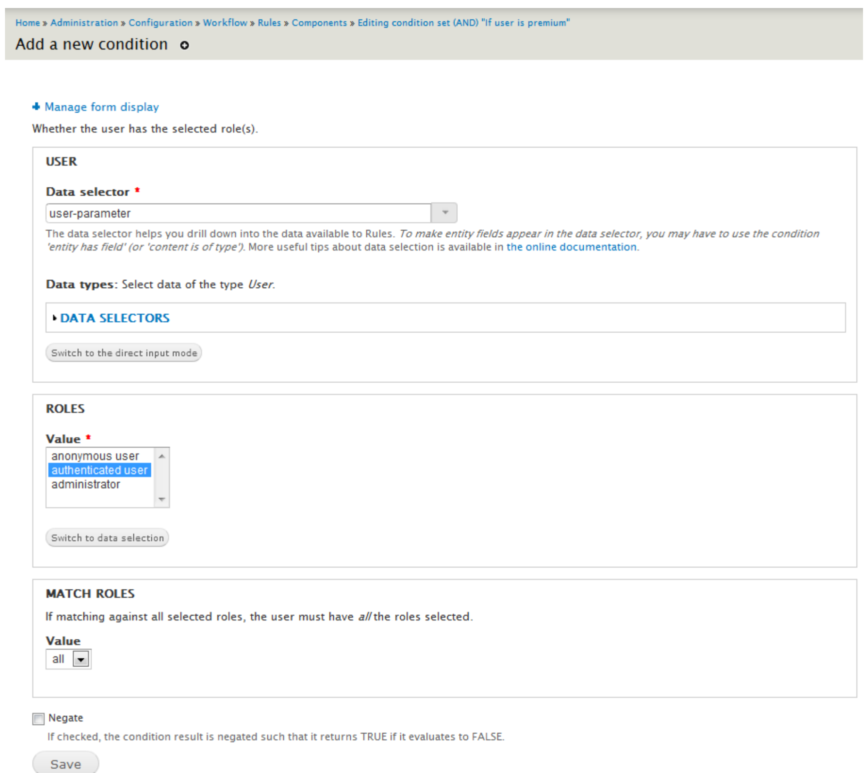
In the configuration of the condition, we will select these values: F31.35

- User: **user-parameter**.
- Roles: **authenticated user**.
- Match roles: **all**.

F31.35

Practical Case 31.2
Create Condition 1

In the first condition we verify that the user has the registered user role.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Step 3. Create Condition 2

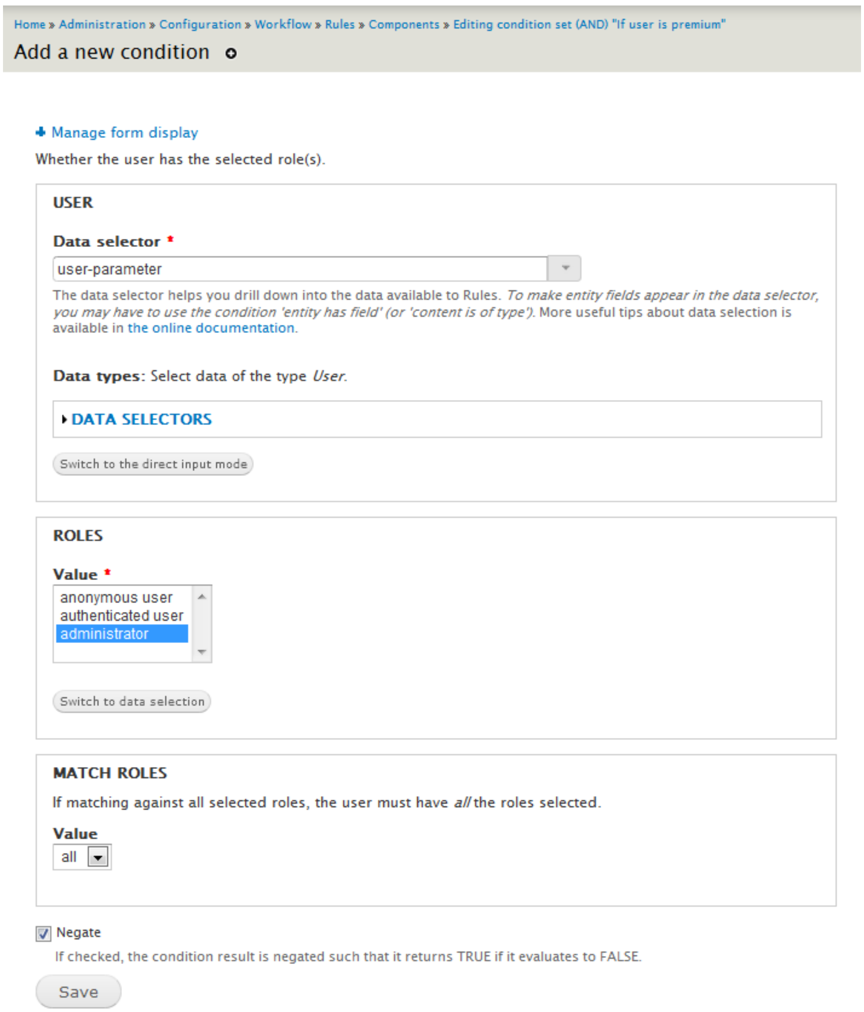
In this step we will create the condition:

- Condition 2: The user does not have the administrator role.

The condition that we have to add is **User has role(s)**, in the **User** group of the dropdown.

In the configuration of the condition, we will select these variables: F31.36

- User: **user-parameter**.
- Roles: **administrator**.
- Match roles: **all**.



F31.36

Practical Case 31.2
Create Condition 2

We will verify that the user does not have the administrator role.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Step 4. Create Condition 3

In this step we will create the condition:

- Condition 3: The user is not blocked.

The condition we have to add is **User is blocked**, in the **User** group of the dropdown.

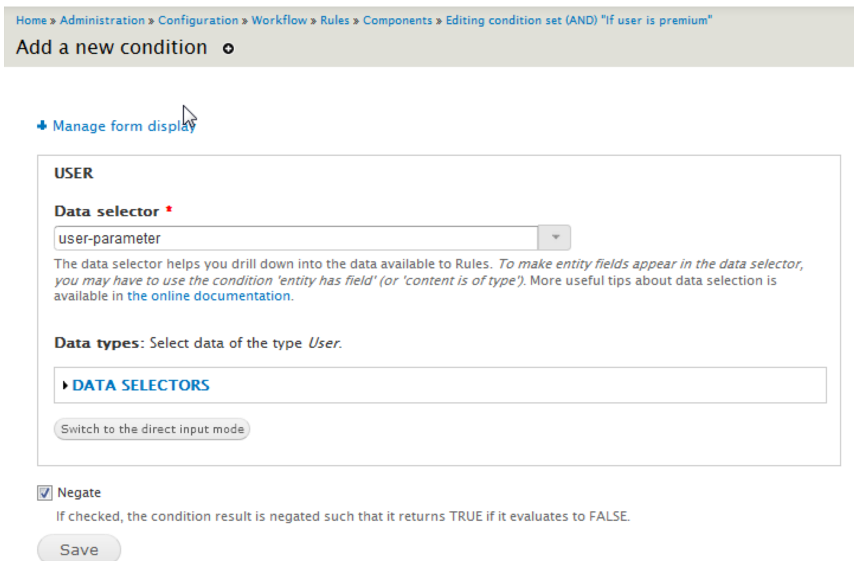
In the configuration of the condition, we will select these variables: F31.37

- User: **user-parameter**.
- Negate: **Yes**.

F31.37

Practical Case 31.2
Create Condition 3

We will verify that the user is NOT blocked. This condition needs to be negated.



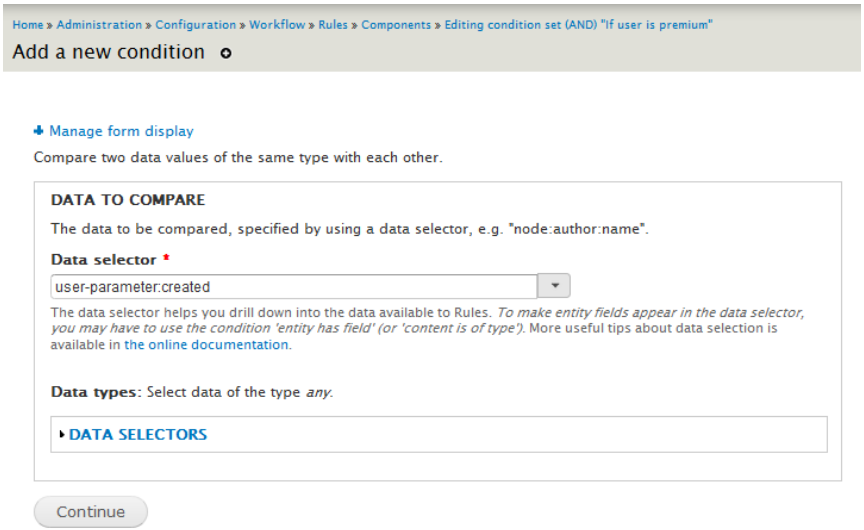
Step 5. Create Condition 4

In this step we create condition:

- Condition 4: The user was created at least 30 days ago.

The condition we have to add is **Data comparison**, from the **Data** group of the dropdown. We will create the user's creation date with the limit date in order to consider it premium (current date - 30 days). If the creation date is less, it means the user registered before 30 days.

In the configuration of the condition we will first select the data that will be compared: **user-parameter:created** (creation date of user). F31.38



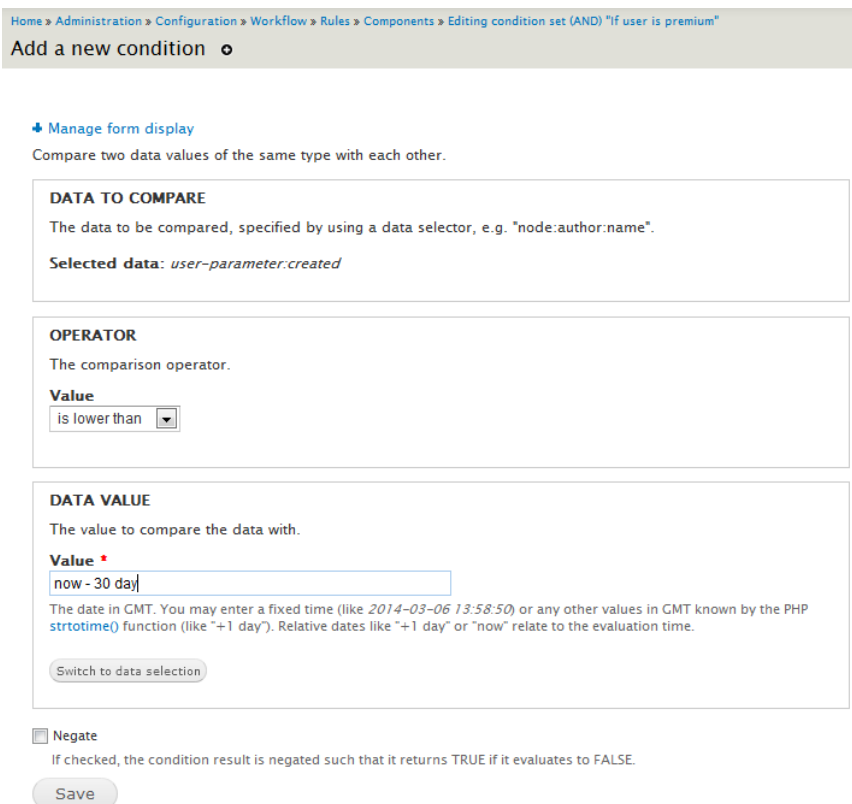
F31.38

Practical Case 31.2
Create Condition 4

In this condition we will compare the creation date of the user with a relative date value.

Next, we will compare this field with: F31.39

- Operator: **is lower than**.
- Value: **now - 30 day** (indicate that it is minus 30 days from the current date, now).



F31.39

Practical Case 31.2
Create Condition 1

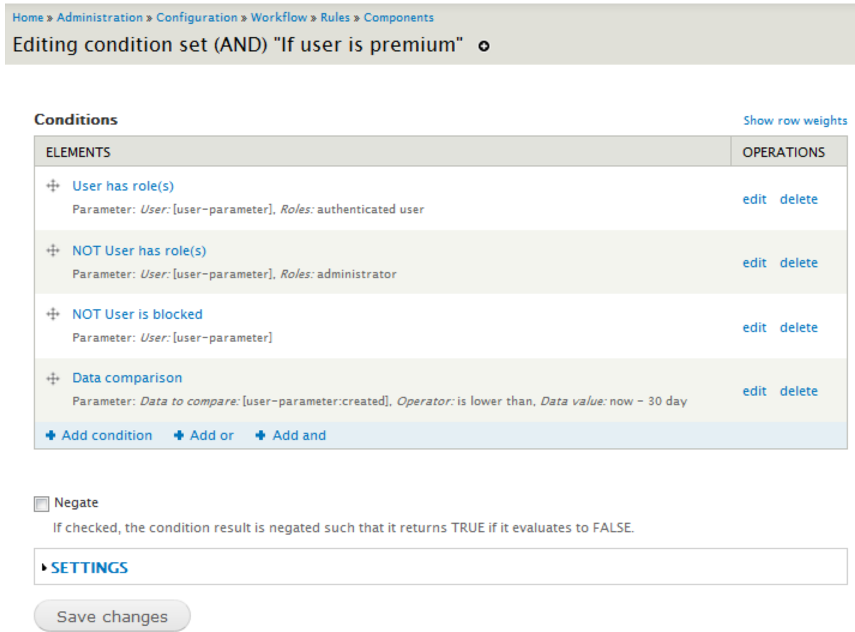
The user creation date has to be 30 days less than the current date, which indicates that the user was created more than 30 days ago.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Step 6. Execute the Component Manually

We have finished configuring the component with the four required conditions. F31.40

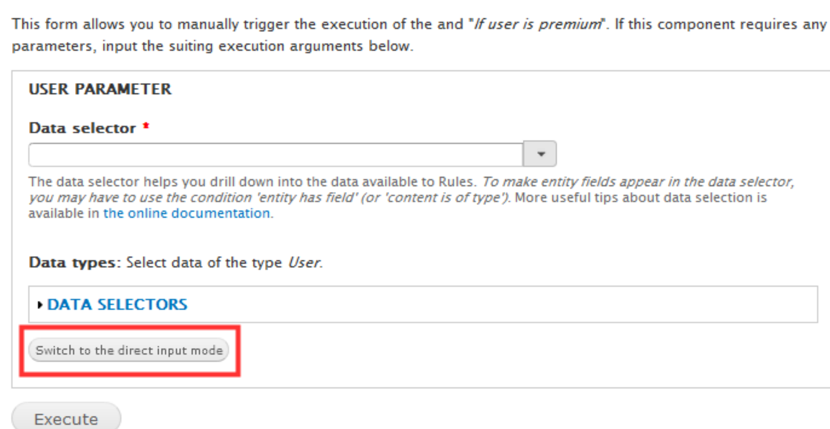
F31.40
Practical Case 31.2 Conditions of the Component
The four defined conditions for the component.



To verify the functionality of the component, we have to go to the components list and use the execute option. In this case the value that we introduced for the User Parameter will not come from any system variable, but we will indicate it manually with the User ID (uid).

We will deactivate the Data Selector by switching to the direct input mode. F31.41

F31.41
Practical Case 31.2 Manual Execution
To introduce a direct entry value we have to switch to the direct entry mode by clicking on the button in the Figure.

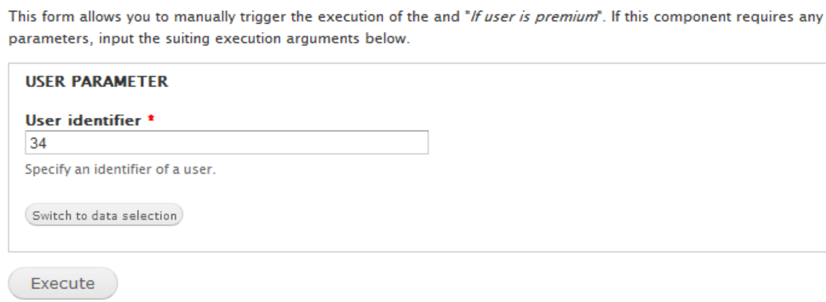


Once switched to the direct entry mode, we can assign a numeric value to the User Parameter that corresponds with an identifier of a valid user (If we indicate an invalid value, the component will return as a **false** result). F31.42

A way of locating the valid user identifiers is from the site user list, click over the

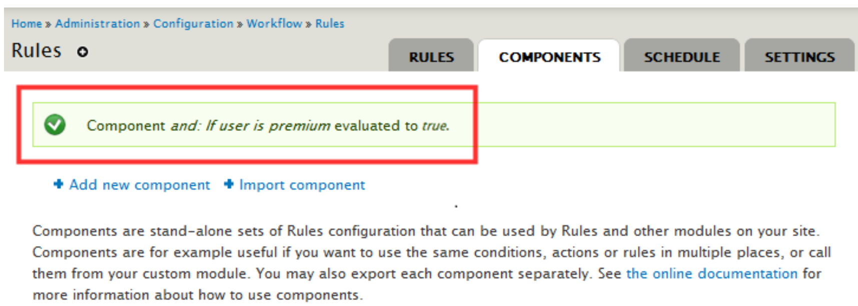
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

edit option of a user. The URL will be **/user/x/edit**, where **x** is the user identifier.



F31.42
Practical Case 31.2
Manual Execution
Now we can introduce values that correspond with user identifiers. If we introduce an invalid value, the component will return as false.

If the user that is introduced meets the conditions, the system will return with a message indicating that the result is true. Otherwise, it will be false. **F31.43**



F31.43
Practical Case 31.2
Manual Execution
If the indicated user passes the validation process, the system will display a message stating that the validation is true. Otherwise, it will display as false.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Practical Case 31.3 Reuse a Component

In this practical case we will reuse the component created in Practical Case 31.2 in order to prove if the user that is visiting the site is *premium*. If this is the case, a welcome message will be displayed. We will create a rule that will be executed when the user accesses the site.

The rule will have the following characteristics:

- **Rule Name**: Premium Message
- **Machine name**: premium_message
- **Events**:
    - User has logged in.
- **Conditions**:
    - The content is Premium. We will use the component **If user is premium**.
- **Actions**:
    - Display message to use.

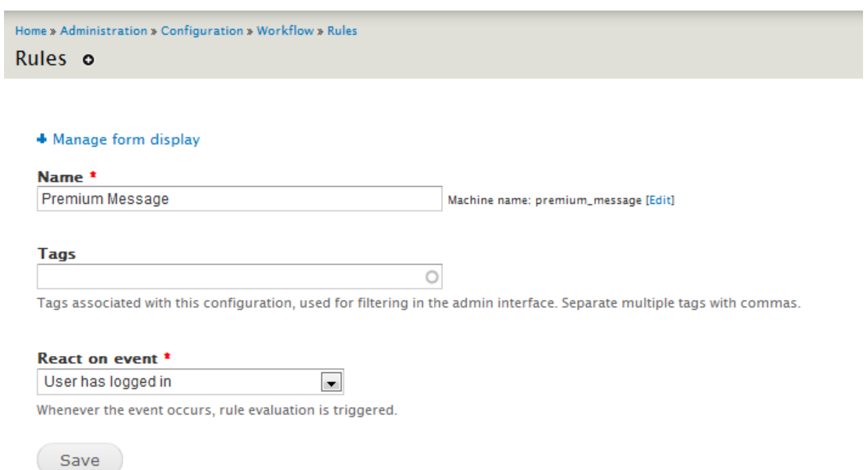
Step 1. Create the Rule

We will create the rule from:

URL Rules
/ admin/config/workflow/rules

Administration ⇨ Configuration ⇨ Workflow ⇨ Rules

We will right the **Name** and **Machine name** indicated in the header. We will also indicate the event **User has logged in**. We then click on Save to create the rule and continue with its definition. F31.44

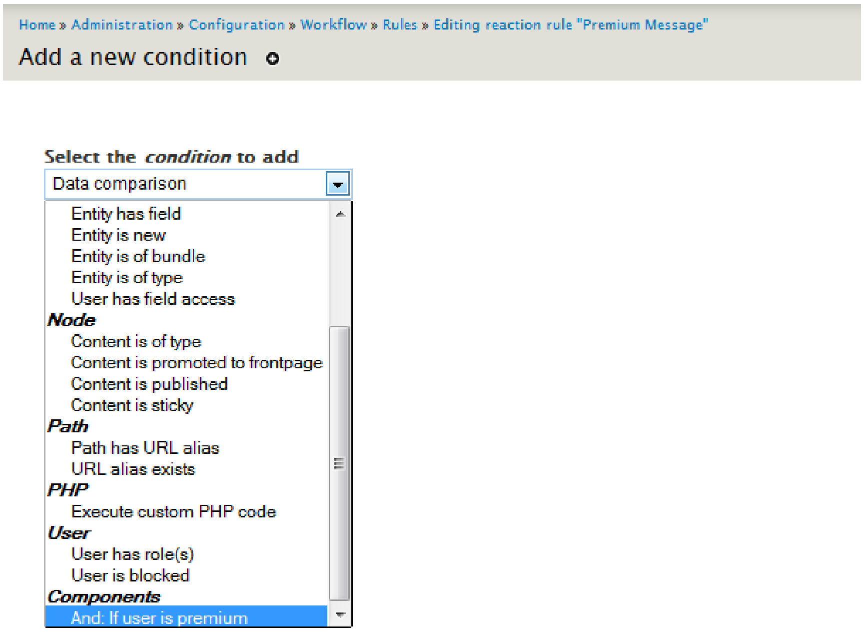


F31.44
Practical Case 31.3
Create the Rule
We will créate the rule Premium Message that will be executed when the user logs into the session.

Step 2. Adding Conditions

The next step is adding conditions. In this case the conditions have been grouped together in the component **If a user is premium**, created in **Practical Case 31.2**. The component we must select will be available in the conditions list within the Components group. F31.45

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

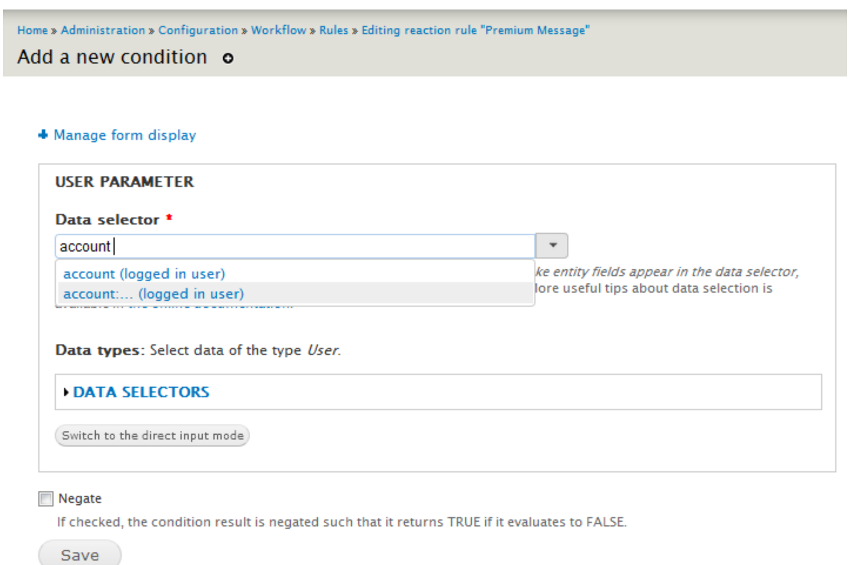


F31.45

Practical Case 31.3 Adding Condition

The created components that group conditions are displayed in the available conditions list in the Components group.

The User Parameter will be the **logged in user** in **account**. Once the component has been added, we will have completed the configuration of the rules conditions. **F31.46**



F31.46

Practical Case 31.3 Adding Condition

We must indicate the value of the User that will correspond with the identified user (account) according to our rule.

Step 3. Adding Actions

The next step is adding actions. We will show a message to the user, and the action will be **Show a message on the site**, within the **System** group.

We will write a Status type message using some of the available replacement patterns. **F31.47**

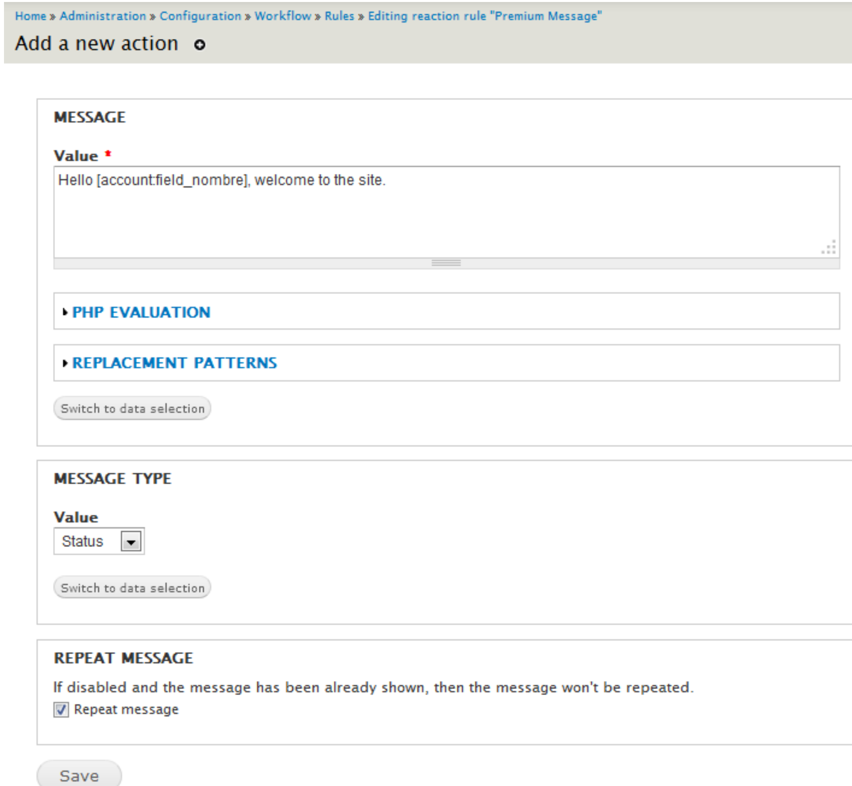
We will save the action to end the rule configuration.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F31.47

Practical Case 31.3 Adding Action

We will configure the message that will be displayed on the screen if the rule conditions are validated.



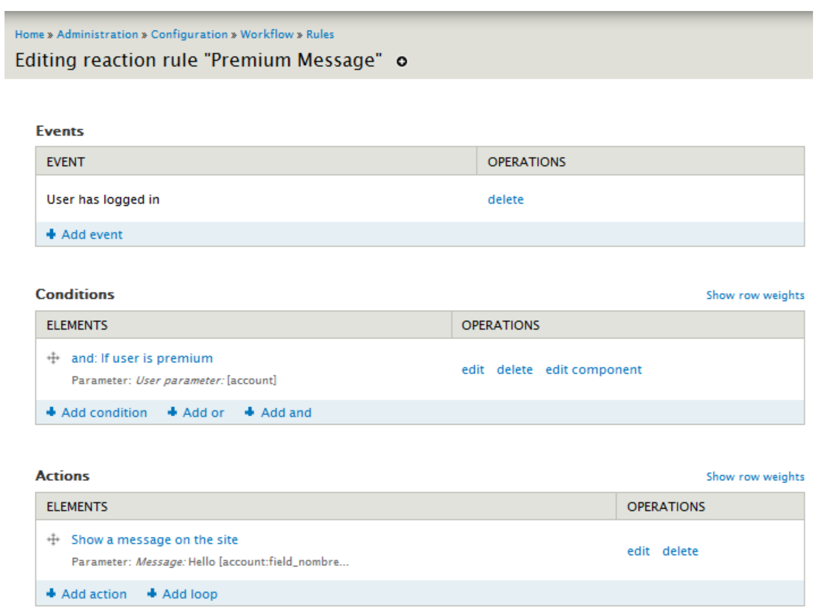
Step 4. Verifying Rule Functionality

We have completed the rule configuration. To verify its functionality, we will simply access the site with different users that either meet or don't meet the rule condition. When the user is considered Premium, the welcome message will be displayed. F31.48

F31.48

Practical Case 31.3 Created Rule

The rule is displayed with its elements: Events, Conditions, and Actions.



Planning

31.4

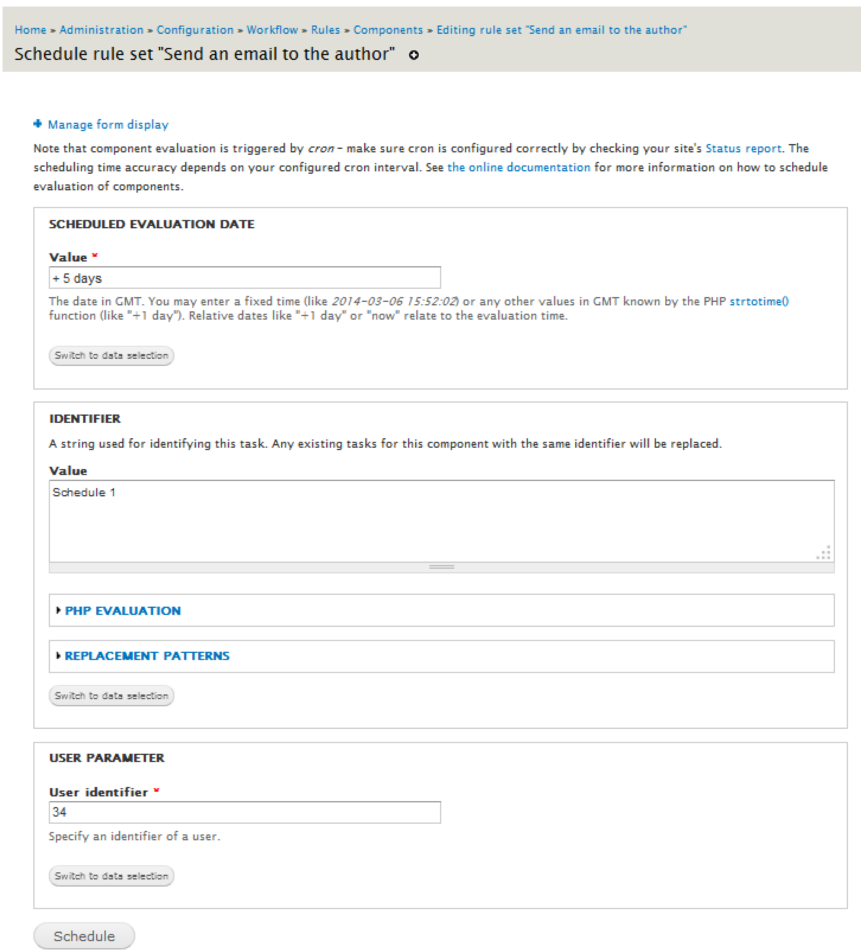
In the previous section we learned that the Rules Components can be executed manually, through the Execute option.

If we have activated the **Rules Scheduler** (included in **Rules**) module, we can also plan the components execution so that they execute automatically on a predetermined date.

It is a very simple function. We will indicate which component from the **Components** list we want to create a plan on by clicking on the **schedule** link.

We will have to indicate a date in absolute value or relative. For example, **+ 5 days** to indicate that the rule will be executed in 5 days.

The **identifier** will allow us to differentiate the programmed task. Finally, we will have e to indicate the **value of the parameters** required by the component that can be fixed values (node identifier, user identifier, etc.) or system values. F31.49



F31.49

Planning a Task

We can plan tasks through components. Each task will will have an associated execution date, the component that will be executed, and the parameters required by the component.

The rule will be verified on the indicated date and its actions executed if the validation of conditions is passed. If the component is an action group type, the actions will always take place.

When programming tasks, the following must be taken into consideration:

- The task **will be executed through the system clock**, which is why there will be no user ID data at the moment of execution (nor any contextual information like nodes visited, etc.)
- The exact moment of the task execution will depend on the execution interval of the system clock. The task will not necessarily be executed on the exact date indicated, but it will happen in **the first execution of the clock where the date has already passed.**

The programmed tasks will be displayed in the Schedule tab: F31.50

URL Planning
/admin/config/workflow/rules/schedule

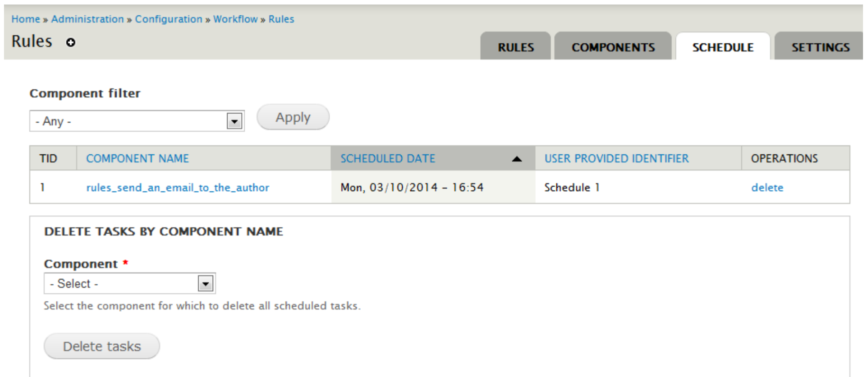
Administration ⇨ Configuration ⇨ Workflow ⇨ Rules [Schedule tab]

F31.50

Planned Tasks

From the Schedule tab, the planned tasks group is visible. The tasks will disappear from the list when they are executed.

The tasks will be executed with the system clock.



From **Schedule** we can control the tasks pending execution. We cannot edit the task execution date. Therefore, if we need to change it, we must delete the current task and create a new one.

Keep in mind that if we modify the associated component before the task is executed, at the moment of execution the definition of the current component will be used.

Integration of Webform With Rules

31.5

In the module repository of Drupal we will find additional modules that complement the Rules module, whether it is by adding rules, events, conditions, actions, etc.

To end this unit, we will analyze the **Webform Rules** module as an example of a module that integrates to perfection with Rules.

The **Webform Rules** module allows the intergration of the Webform module with Rules in such a way that we can create rules that are validated when a user completes a form (implemented with Webform).

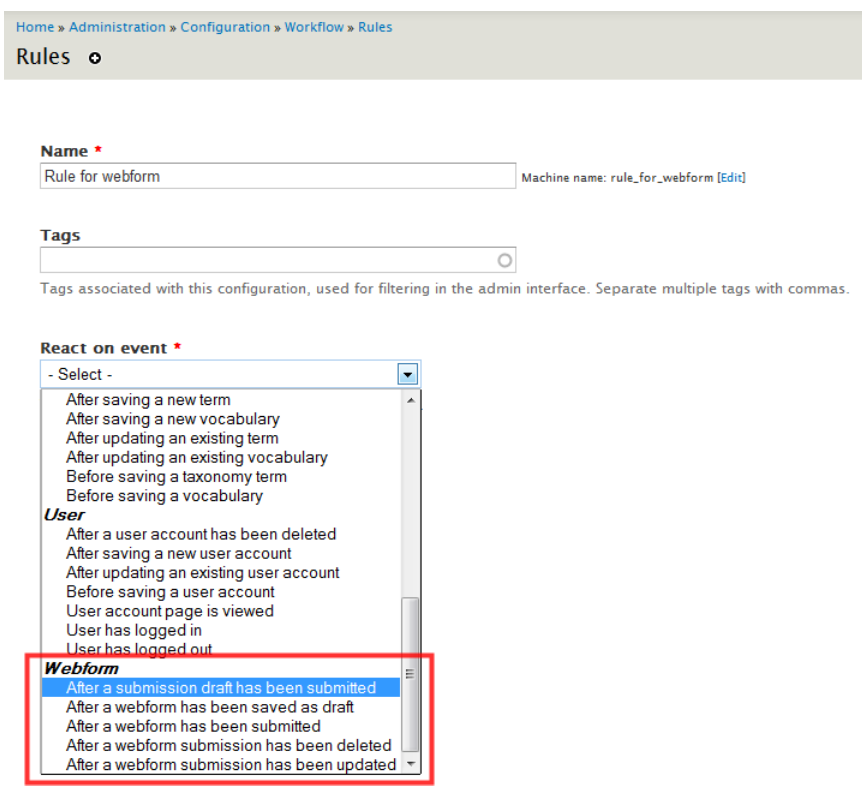
The Webform Rules module is available at:

http://drupal.org/project/webform_rules

Once it has been installed and activated, the module will add a group of events and conditions to the Rules of Rules System that will allow us to generate rules when a user completes a form.

The new **events** will be grouped into **Webform** categories: F31.51

- **After a webform has been saved as draft**.
- **After a webform has been submitted**.
- **After a webform submission has been added**.
- **After a webform submission has been deleted**.
- **After a webform submission has been updated**.



F31.51
Events Added By Webform Rules
The module adds a group of events related with the Webforms.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

From **conditions** we can identify the form sent with the condition:

- **Webform has name**. We must indicate the form ID and the form name that we want to verify, from the forms created. F31.52

In order to configure this condition correctly, we can use the value the value $form_id as the forms identifying value, which includes the form ID that has been sent (**<?php print $form_id ?>**).

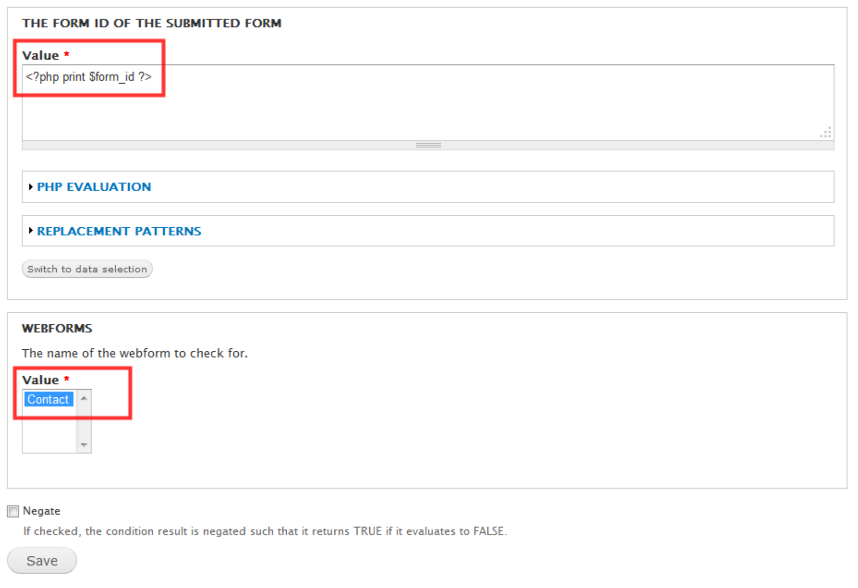
In the **Webforms** field, we will select the form or forms on which the defined actions will be executed.

F31.52

Conditions

The module adds a condition that allows the form sent to be indentified. This way se can create different rules in the function of the form sent.

enviado.



Finally, we must define the action or actions that we want executed. The module does not add specific actions related to the forms.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

32 Defining contexts

The Context module allows you to define contexts, which represent different sections of a site. When the page loads, the system checks a set of rules and conditions to see what context is active. This allows us to define important distinctions for each context: distinct theme, distinct page structure and even a distinct layout.

In this unit, we will study the conditions and reactions that help us apply the distinct conditions for each context.

We also will look at the following new modules:

- **Context layouts**, included with Context, allows us to use different layouts for the active context.
- **Context No Title**, allows us to hide the content title when the context is active.
- **Context Condition Theme**, adds a condition as a result of the theme that is active on the site.
- **Context Node**, assigns contexts to nodes.
- **Context Assets**, allows us to apply .css and .jss files to pages.
- **Date Context**, included with the Date module, allows us to configure conditions that compare a date field to a determined date.

Comparative D7/D6

Context

Context 3.x has the same functionality and administrative interface in both Drupal 7 and Drupal 6.

There are differences between the two, however, partly because of the way Drupal 7 handles entities. We also have new modules available only in Drupal 7, and some Drupal 6 modules have not been converted to operate with Drupal 7.

Unit contents

32.1 Installing and configuring Context 362
32.2 Create contexts...363
32.3 Defining conditions...367
32.4 Defining reactions 373
32.5 Using context to change design 376
32.6 Additional modules..379

32

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

32.1 Installing and Configuring Context

The Context module allows you to define certain contexts, which represent different sections of a site. When a page loads, the system checks a set of rules and conditions to see what context to use. This allows us to define important distinctions for each context: distinct theme, distinct page structure, etc.

The **Context** module is available here:

http://drupal.org/project/context

Context relies on the **Chaos tool suite** (Ctools) module, which also is required for **Panels** and **Display Suite** to operate. If you haven't installed it yet, **Chaos Tool suite** is available here:

http://drupal.org/project/ctools

Context includes three separate components:

- **Context**. This is the main module that is required to use contexts.
- **Context layouts**. This component integrates Context into theme regions, allowing us to change the layout of elements in regions, based on context.
- **Context UI**. The user interface to define contexts.

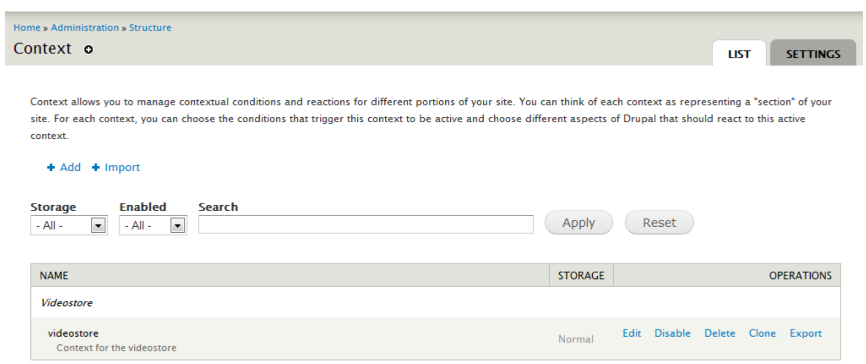
After activating the necessary modules, we can access the **Context** administrative section, at:

URL Context
/admin/structure/context

Administration ⇨ Structure ⇨ Context

From **Context**, we can access the list of contexts that we've created. We can edit, enable, disable, clone and export the existing contexts. Also we can create new contexts from this page. F32.1

F32.1
Contexts
List of contexts on the site.



Create contexts

32.2

To create a new context, first we navigate to the **Context** administration page:

Administration ⇨ Structure ⇨ Context [Add link]

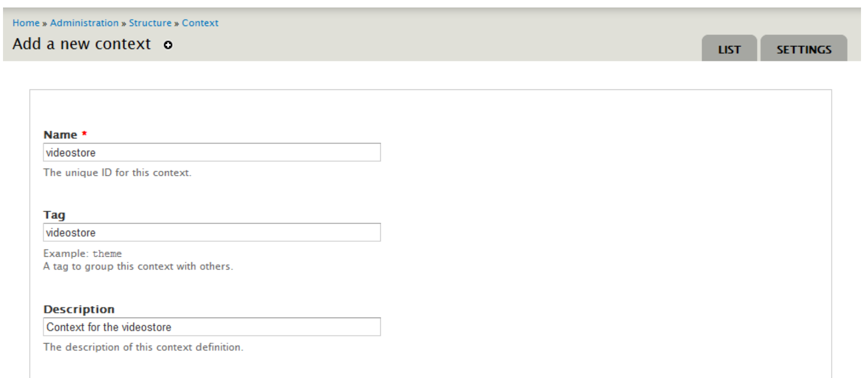
URL Add context
/admin/structure/context/add

Click on the Add link to open the page to add a new context, with the following options:

- **Name**. Unique ID for the context. Only letters, numbers, hyphens and underscores are valid. The name cannot be changed after the context is created. F32.2

- **Tag**. In the administrative list, the contexts are organized by tags. F32.2

- **Description**. The description of the context is shown in the administration list. F32.2



F32.2

Add context

To create a new context, we must provide a Name, which is a unique ID (only letters, numbers, hypens and underscores are allowed).

- **Require all conditions**. If this option is enabled, the context is active only when all the conditions are fulfilled. Otherwise, the conditions are evaluated in order, and the context is active as soon as the proper conditions are met. F32.3

- **Conditions**. Define conditions for the context. For example, a context can be active when the page is loading a node of a certain content type. In **section 32.3** we look in detail at the available conditions. F32.3

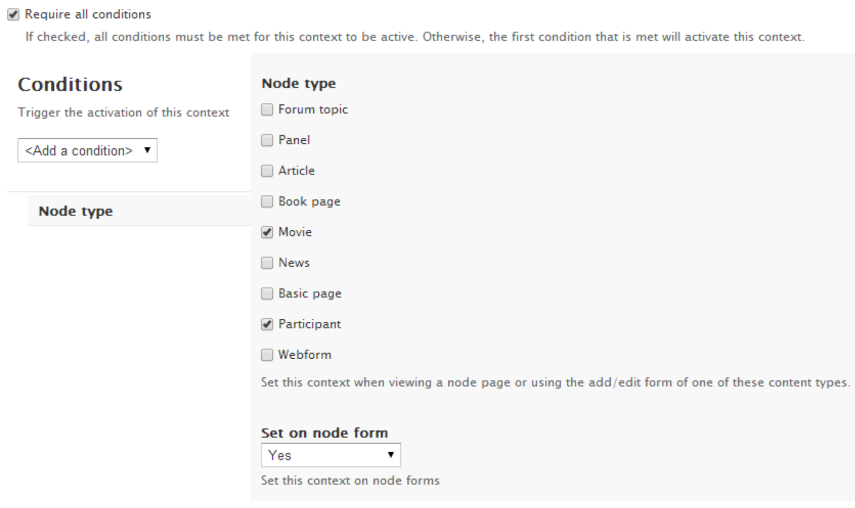
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F32.3

Conditions

Conditions define when a context is active.

If "Require all conditions" is checked, the context is only active when all the conditions are met. Otherwise, the conditions are evaluated in order, and the context is active as soon as the proper conditions are met.

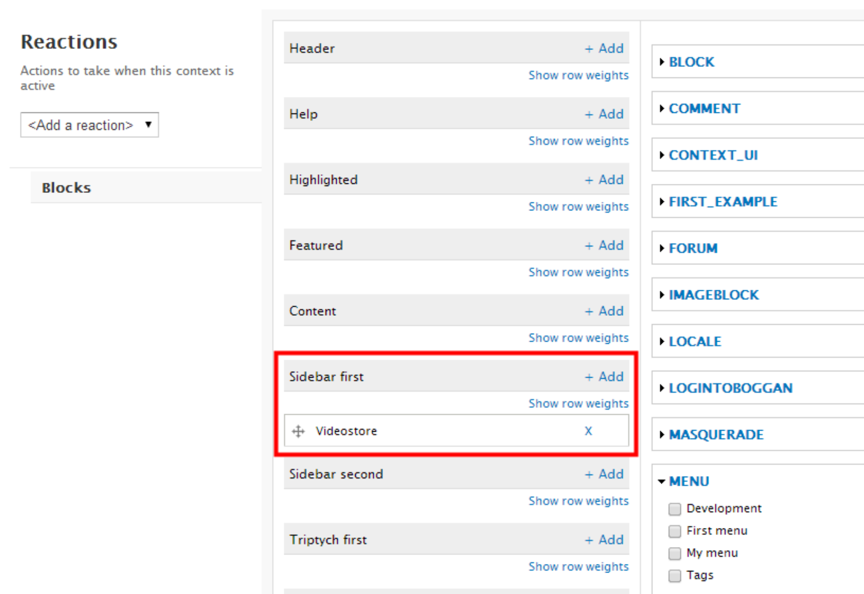


- **Reactions**. The settings and actions that are triggered when the context is active. For example, the site will display a menu block in one of the theme regions when the context is active. In **section 32.4** we look in detail at the available reactions. F32.4

F32.4

Reactions

Reactions are the actions that are taken when the context is active, or when the conditions have been met.



Once it has been created, the new context is active as soon as the conditions are met.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Example of activating a context

In our example site, we set up a context called **videostore**, which shows the Videostore menu block (reaction) when the site displays a Movie node (condition).

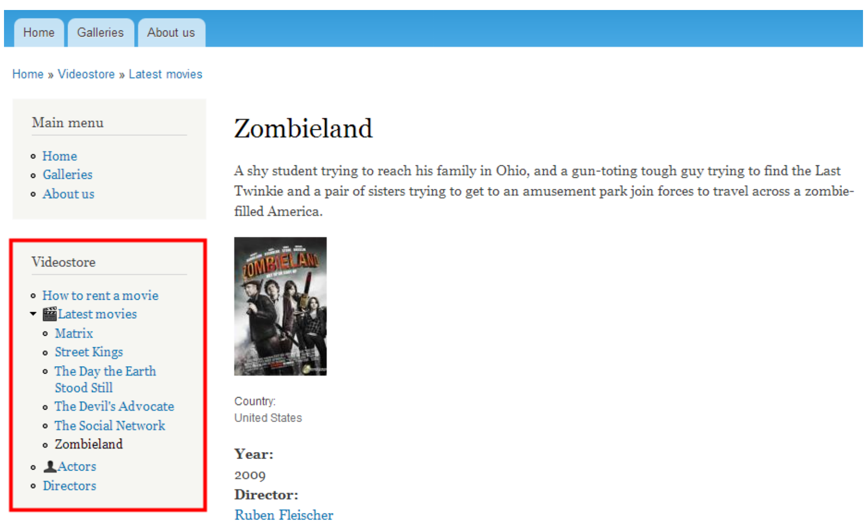
At first, we see a page that doesn't satisfy the context condition. F32.5

F32.5

Example of applying context

The Figure shows a page where the conditions are not met.

By default, the site shows the Main Menu block in the left-hand column. Although we create a set of reactions, this doesn't affect the other elements that are set up to display by default on the site. So, the global block continues to be displayed when the **videostore** context is active, as shown below. F32.6



F32.6

Example of applying context

The context is active when the site loads a Movie type page. When the context is active, the page displays the Videostore menu block.

The default blocks (Main Menu) are displayed regardless of the active context.

Only when the contexts are configured correctly can we ensure that the page elements are displayed for the right context.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Managing contexts

The contexts we create are displayed on the context administration list, where we can take the following actions: F32.7

- **Edit**. The unique ID for a context cannot be changed, but other options can be edited, such as conditions and reactions.

- **Enable/disable**. Temporarily disable the contexts that are not being used.

- **Delete**. Once a context is deleted, it cannot be recovered, so only use this option when you're certain that the context is no longer needed to use on the site or to create an edited version. Otherwise, it is more convenient to simply disable a context.

- **Clone**. Make a copy of a context, including conditions and reactions. Provide a new name and unique ID. Once you clone a context, you can proceed to make any needed edits.

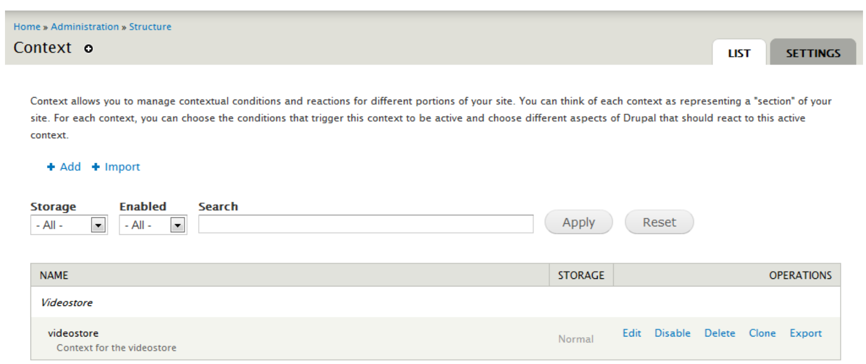
- **Export/Import**. With export/import, we can copy contexts from one Drupal system to another. Remember to make sure the versions of Drupal and the **Context** module should be the same on both sites.

The administration section includes a set of filters that can be used to search for contexts. As mentioned above, contexts are organized into categories based on the **Tag** field.

F32.7

Managing contexts

The administration section shows the list of available contexts, and provides tools for various operations, such as edit, enable/disable, delete, clone, export/import.



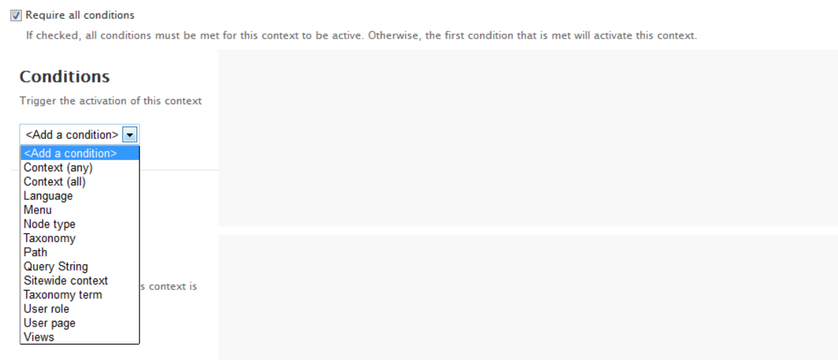
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Defining conditions

32.3

We've already seen how we can enable the **Require all conditions** option once we create a new context, which makes sure that all conditions are valid before executing the context. If this option is not enabled, then the context will be active if just one of the conditions is met.

One context can include multiple conditions, which we can add from the conditions selector. F32.8



F32.8
Conditions
The default list of conditions.

Depending on the modules installed on the site, this list will display various options.

Conditions that we add to the context will appear below the selector and no longer appear in the selector. The selector only shows the conditions that have not been added to the context. To edit a condition that has been added already, click on the name of the condition.

Now let's look at each of the conditions that are available. This list will vary depending on the modules installed on a site.

Condition: Context

Two or more contexts can be active at the same time. With **Condition**, we can enable a context depending on whether or not other contexts have been enabled.

To configure this condition, we must define the contexts that will load the current context, providing one name for each line. Use the * symbol as a wild-card, or the ~ symbol to indicate a context that is not valid for the condition. F32.9

F32.9
Condition: Context
Enable a context based on whether another context is active.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

For example, use the following values:

context*
~context3

The condition is true when context, context1, context2, context4, etc. are active. But it returns false if only context3 is active.

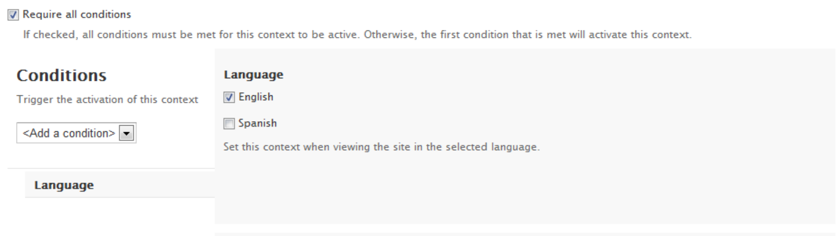
Condition: Language

The context is active when the site is being displayed in the determined language or languages. F32.10

F32.10

Condition: Language

The context is active based on the language being used on the site.



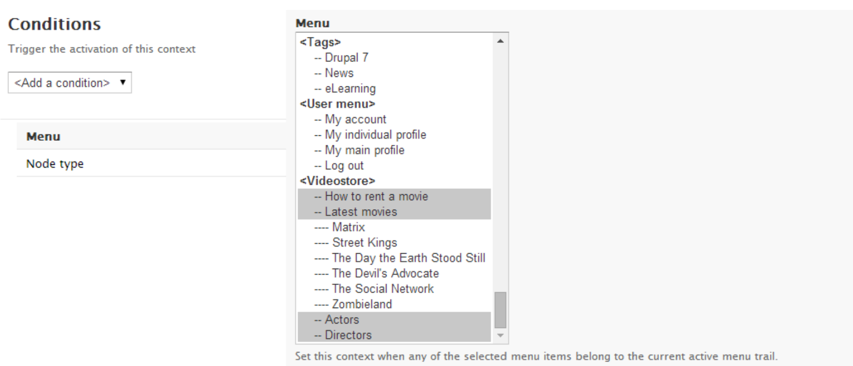
Condition: Menu

The context is active when one of the defined menu elements is found in the path of the current page. F32.11

F32.11

Condition: Menu

The context is active when the page being loaded is part of a determined menu.



For example, in the current structure where we have selected Link 1 and Link 2.1, we can see which elements will make the condition true (T) or false (F):

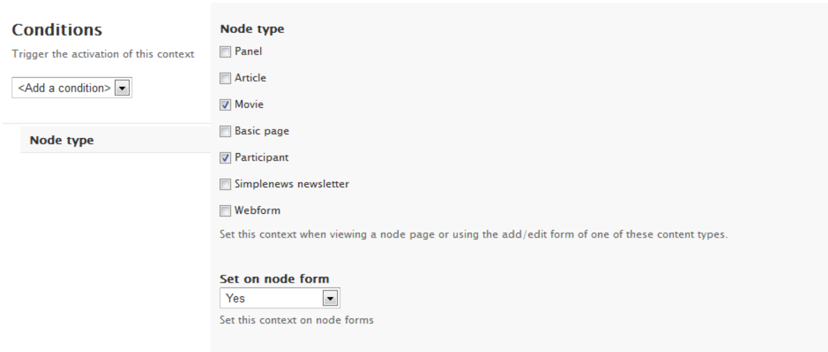
- **Link 1** (T)
  - Link 1.1 (T)
    - Link 1.1.1 (T)
    - Link 1.1.2 (T)
  - Link 1.2 (T)

- Link 2 (F)
  - **Link 2.1** (T)
    - Link 2.1.1 (T)
    - Link 2.1.2 (T)
  - Link 2.2 (F)

The condition is true when the page being displayed is linked from any of the items in the determined menu (T).

Condition: Node type

We can indicate what type of content will make the context active. The condition is true when the determined content type is loading, and false when it is not. F32.12



F32.12

Condition: Node type

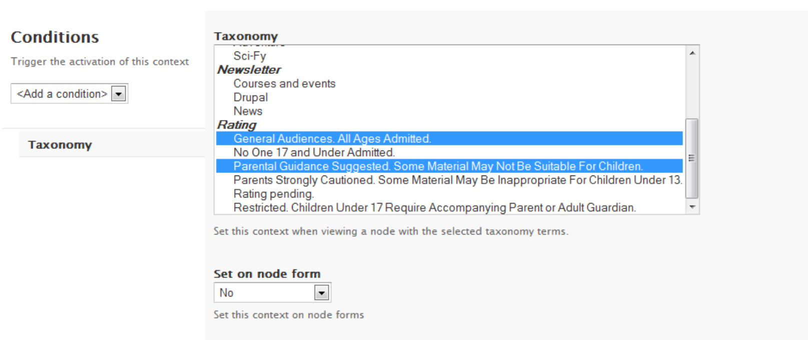
The context is active for certain types of content.

For **Set on node form**, we can select one of these options:

- **No**. The context is active only when a node of the defined type is being displayed, but not while editing or creating the node.
- **Yes**. The context is active when the node is being displayed or while it is being edited or created.
- **Only on node form**. The context is only active while editing or creating nodes of the defined type.

Condition: Taxonomy

The context is active when the node being displayed contains one of the defined taxonomy terms. F32.13



F32.13

Condition: Taxonomy

The context is active when the node being displayed contains one of the defined taxonomy terms.

For this condition, we can also indicate whether the context is active for the node edit form.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

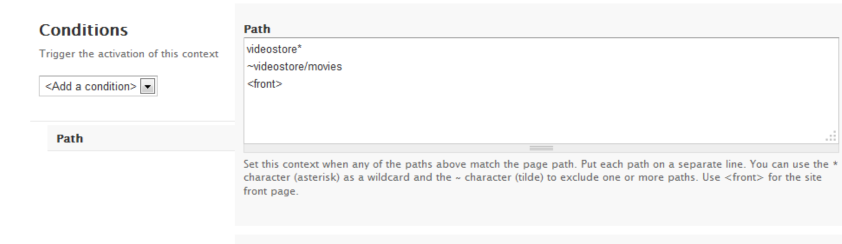
Condition: Path

We can define which paths will make the context active, by providing one path on each line. Use the * symbol as a wild-card, and the ~ symbol before a path to indicate that it does not validate the condition. F32.14

F32.14

Condition: Path

The context is active based on the path of the page being loaded.



For example, we can use the following terms:

videostore*
~videostore/movies
<front>

Now let's look at some examples of URLs that will be evaluated to make the condition true (T) or false (F).

videostore (T)
videostore/movies/matrix (T)
videostore/movies (F)
<front> (T)
who-we-are (F)

Condition: Sitewide context

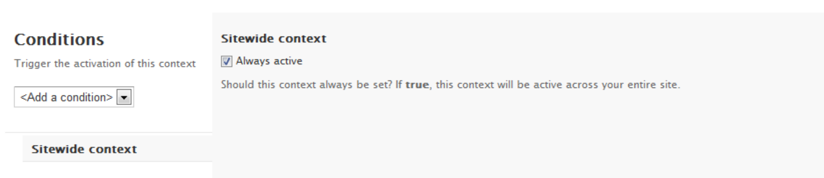
When enabled, this context is always active. F32.15

This option provides a way to create a global condition that is validated by default as true. Remember, when we add other required conditions to the context, the context is only active if all the conditions are met.

F32.15

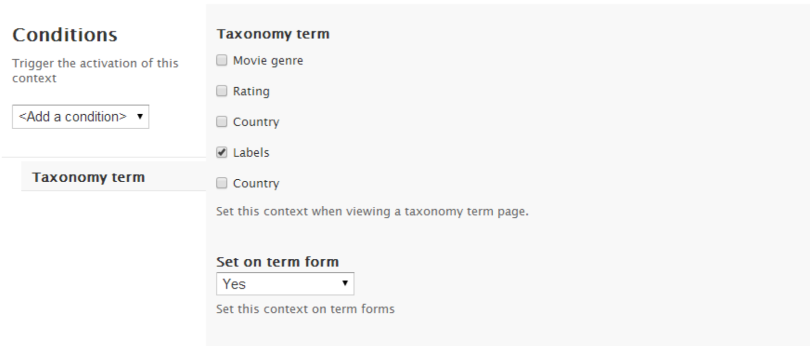
Condition: Sitewide context

The context is always active. Use this to create global contexts.



Condition: Taxonomy term

The context is active when the page being displayed has the taxonomy terms that have been selected. We also can indicate whether the context is active on the node edit form. F32.16



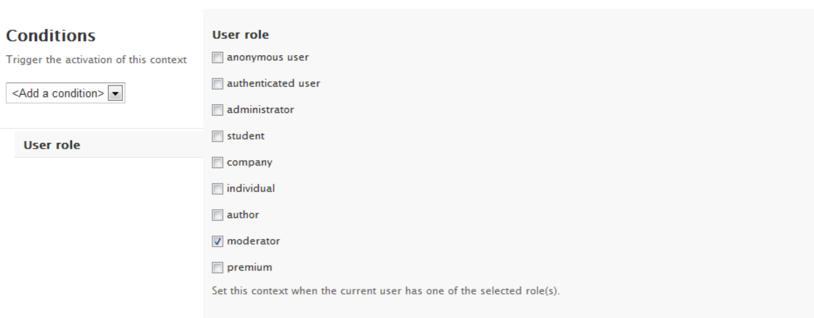
F32.16

Condition: Taxonomy term

The context is active on pages with taxonomy terms that have been selected.

Condition: User role

Define the roles that make the context active. F32.17



F32.17

Condition: User role

The context is active based on the role of the user viewing the page.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Condition: User page

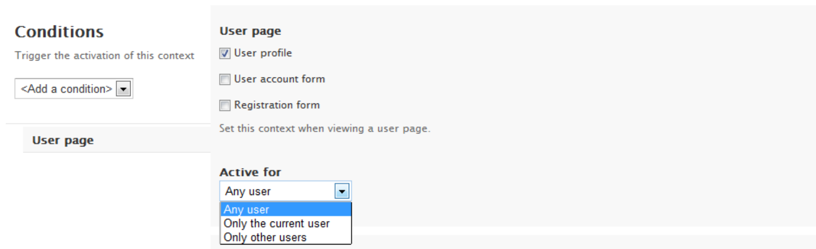
The context is active when a user page is being displayed: F32.18

- User profile
- User account form.
- Registration form.

F32.18

Condition: User page

The context is active when displaying a user page (user profile, account form, registration form).



The **Active for** option allows us to validate the context for:

- **Any user**. For example, when a user visits their own page or another user's page.
- **Only the current user**. Only when the user page belongs to the current user. For example, when the user visits their own page. The context is not active when the user visits the profiles of other users.
- **Only other users**. Only when the user page does not belong to the current user. For example, the user visits the profiles of other users.

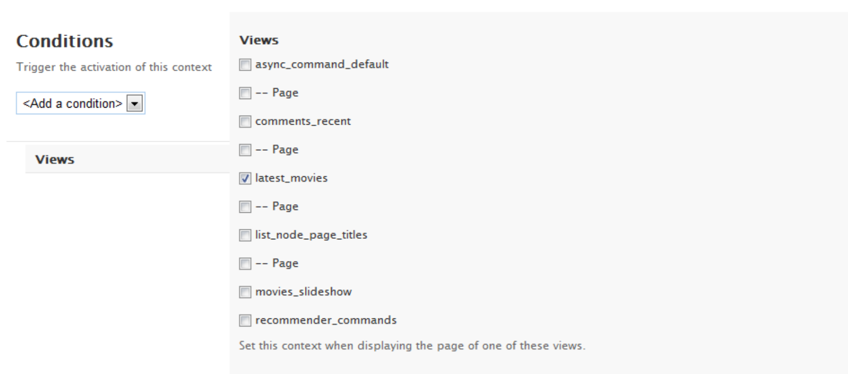
Condition: Views

The context is active when the page being displayed contains any of the defined views. This is only available for Views pages. F32.19

F32.19

Condition: Views

The context is active when displaying a page for the defined Views. Only available for Views pages.



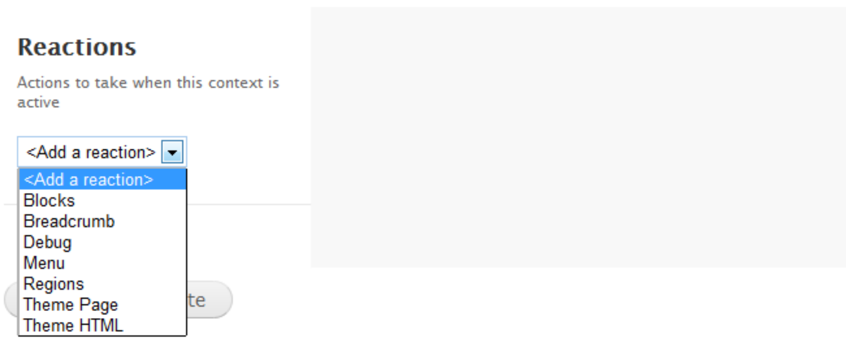
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Defining reactions

32.4

Reactions are the actions taken when the context is active, or when the conditions have been validated. For a given context, we can define various reactions that are executed at the same time.

In this section we will study the reactions that are available when we define a context. The set of reactions available will vary depending on the modules installed on a site. F32.20



F32.20

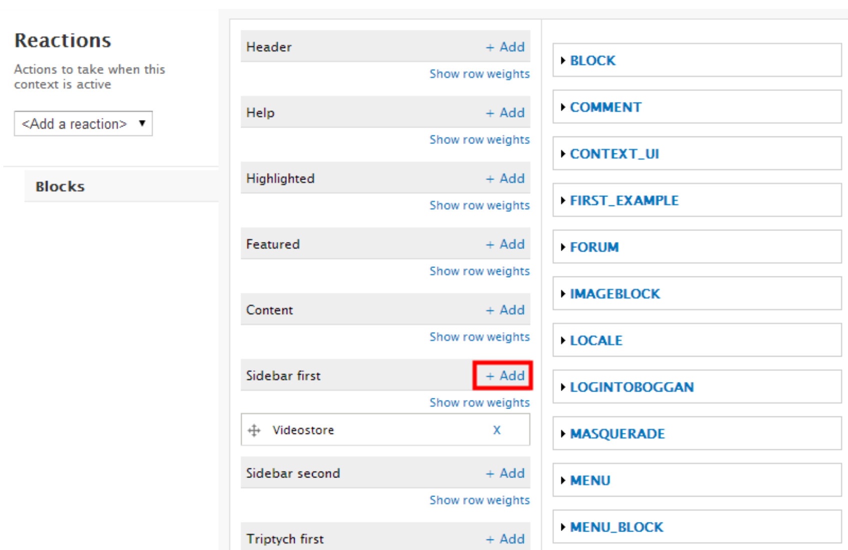
Reactions

The list of reactions available by default.

Additional reactions may be available if other modules are installed.

Reaction: Blocks

Allows us to associate blocks to regions on the site. The blocks will be displayed in the regions when the context is active. F32.21



F32.21

Reaction: Blocks

Define how blocks are assigned to regions, based on the active context.

Follow these steps to assign blocks to certain regions:

- Find the block you wish to choose, on the right side, and select Add.
- In the left column, find the region where you will add the block, and click Add.
- Once the blocks are added to a region, you can re-order them with

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

drag-and-drop, or apply row weights.

With this method of using context to assign blocks to regions, the blocks will be displayed in the assigned region even though they are not enabled on the block administration page.

Additionally, blocks set for global display will be shown independently of the blocks assigned to the active context. When working with contexts, you should make it a habit to **disable all blocks** from the blocks administration page and **activate them solely for the defined contexts**.

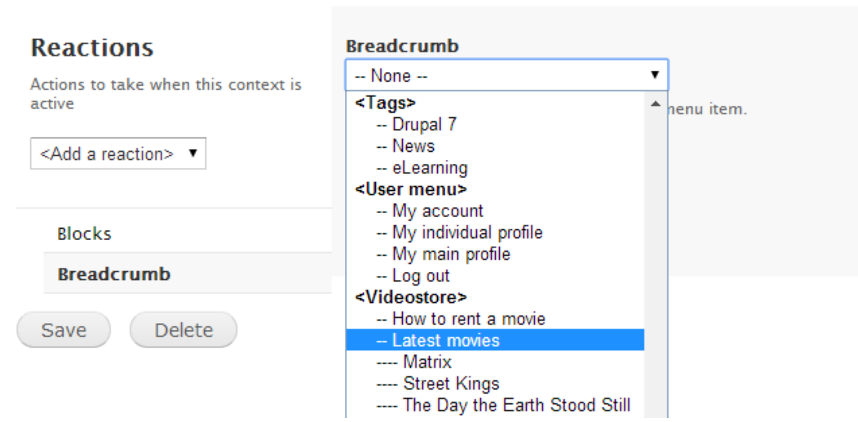
Reaction: Breadcrumbs

When the context is active, breadcrumbs will be displayed for the selected menu link, overwriting the default breadcrumbs for the current page. F32.22

F32.22

Reaction: Breadcrumbs

Define custom breadcrumbs, by selecting a specific menu item.



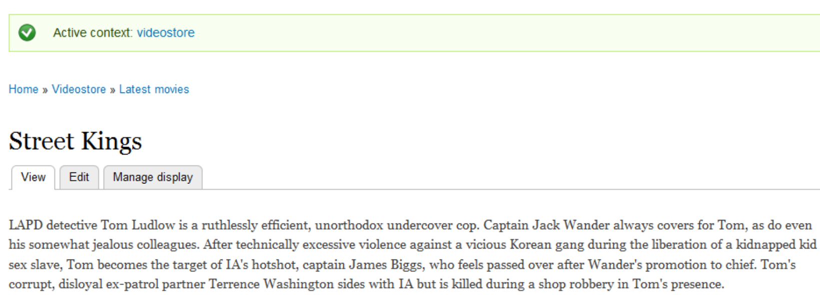
Reaction: Debug

No additional configuration is needed for this reaction. It simply **displays a message when the context is active**. This is especially useful while building a site or developing new contexts, to confirm that a given context is active as expected. F32.23

F32.23

Reaction: Debug

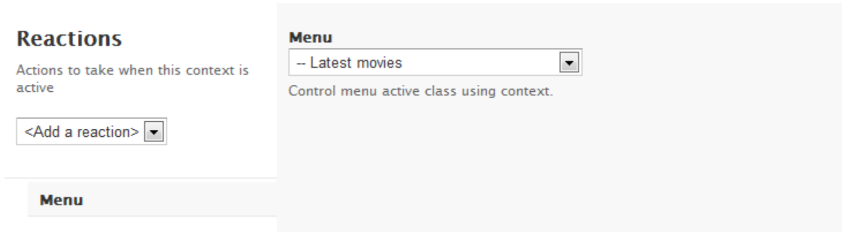
Display a message when a context is active. Use only during site development.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Reaction: Menu

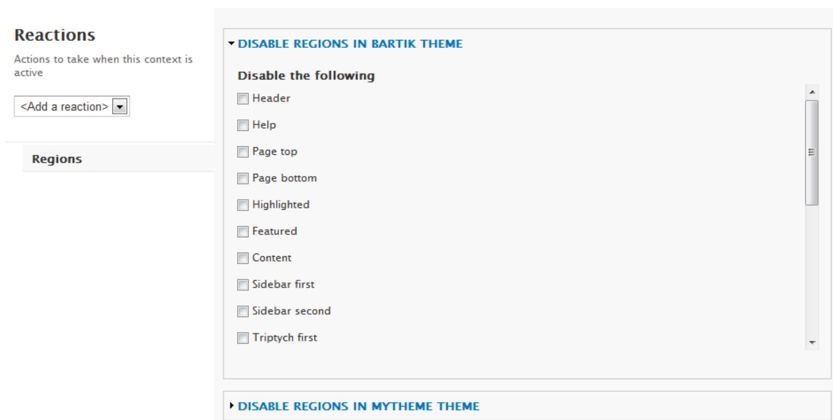
Display the defined menu link when active. Internally, this action simply changes the display of the menu element, by applying the style of the active or defined menu element. F32.24



F32.24
Reaction: Menu
Displays a defined menu link.

Reaction: Regions

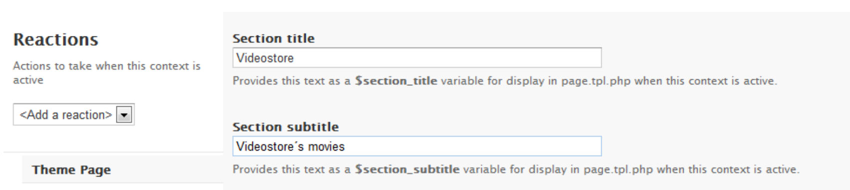
When this context is active, the selected regions are disabled. Use this to determine which regions will be disabled for each one of the themes on the site. F32.25



Reaction: Regions
Disable theme regions when the context is active.

Reaction: Theme Page

Configure the **title** and **subtitle** that will be shown for a section when the context is active. F32.26



F32.26
Recation: Theme Page
Assign values to the section title and subtitle variables, which are passed to the page.tpl.php template.

The **section title** corresponds to the **$section_title** variable, and the subtitle corresponds to **$section_subtitle**. These two variables are passed to the **page.tpl.php** template, but we will have to modify the template to define their location and style.

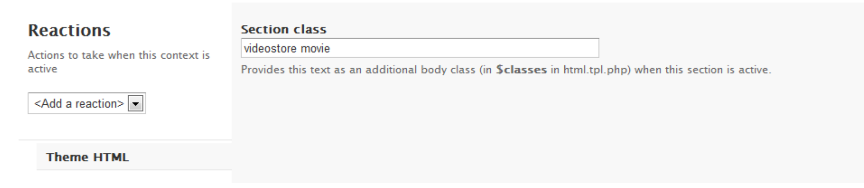
Reaction: Theme HTML

Define one or more classes (separated by spaces) that are added to the current page, using the **$classes** variable in the **html.tpl.php** template. F32.27

F32.27

Reaction: Theme HTML

Include CSS classes that are applied to the page <body> tag.



Looking at the HTML code generated for the page, we can see the classes applied to the <body> tag when the context is active. This allows us to define specific styles for a given section or context.

```
<body class="html not-front logged-in one-sidebar sidebar-first
page-node page-node- page-node-31 node-type-movie videostore movie
toolbar" >
```

32.5 Using context to change design

By enabling the **Context layouts** module within Context, we can change the layout of regions based on the active context.

To make this possible, we first must define the different layouts and region configurations that are available inside the theme. Then we create an alternate page.tpl.php template for each layout that we add to the theme, along with the corresponding .css files. We will explore an example, step-by-step, when we modify the **My theme** theme that we created in Unit 29, as a Zen subtheme.

Step 1. Define the layout and regions in the theme .info file

The **mytheme.info** file shows the regions that are available for our theme. In this example, we are not adding new regions, rather creating a secondary layout that will be simpler than the original. We will include just two columns and eliminate the other regions.

mytheme.info

Regions defined in the configuration file of the theme.

```
name        = My theme
description = Sub-theme based on Zen.
core        = 7.x
base theme  = zen

regions[help]             = Help
regions[sidebar_first]    = First sidebar
regions[sidebar_second]   = Second sidebar
regions[navigation]       = Navigation bar
regions[highlighted]      = Highlighted
regions[content]          = Content
regions[header]           = Header
regions[footer]           = Footer
regions[bottom]           = Page bottom
regions[page_top]         = Page top
regions[page_bottom]      = Page bottom
```


Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Let's define two layouts, with different sets of regions that are available. The first will be the default layout, which corresponds to the original structure generated by the page.tpl.php template. Call this layout **Default** and the template **page** will be called **page.tpl.php**. By default, all the theme regions will be available in the layout, so we will use the following path: **layouts[default][regions][]**.

```
; Layout: Default
layouts[default][name] = "Default"
layouts[default][description] = "All regions"
layouts[default][template] = "page"

layouts[default][regions][] = "help"
layouts[default][regions][] = "sidebar_first"
layouts[default][regions][] = "sidebar_second"
layouts[default][regions][] = "navigation"
layouts[default][regions][] = "highlighted"
layouts[default][regions][] = "content"
layouts[default][regions][] = "header"
layouts[default][regions][] = "footer"
layouts[default][regions][] = "bottom"
layouts[default][regions][] = "page_top"
layouts[default][regions][] = "page_bottom"
```

mytheme.info

Layouts are defined in the theme .info file.

This is the default layout.

Now let's create a new, simpler layout that is based on the first, which we'll call **Basic**. As shown in the code below, we'll replace the **page.tpl.php** template with a template called **layout-basic.tpl.php**, which we will create inside the theme folder. Also, it will reference a new stylesheet called layout-basis.css, that we also must create to apply specific styles for the new layout. (This alternate stylesheet is optional.)

```
; Layout: Basic
layouts[basic][name] = "Basic"
layouts[basic][description] = "Basic layout"
layouts[basic][stylesheet] = "layout-basic.css"
layouts[basic][template] = "layout-basic"
layouts[basic][regions][] = "sidebar_first"
layouts[basic][regions][] = "content"
```

mytheme.info

Layouts are defined in the theme .info file. The new layout is called basic.

Step 2. Create a page template

The next step is to create a page template for the new layout, replacing the default page.tpl.php template.

Below is a simplified example of the template for the new layout (**layout-basic.tpl.php**), which only includes the **content** ($page['content']) and **sidebar_first** ($page['sidebar_first']) regions.

```
<div id="page-wrapper"><div id="page">
  <div id="main-wrapper">
    <div id="content">
      <?php if ($title): ?>
        <h1 class="title"><?php print $title; ?></h1>
      <?php endif; ?>

      <?php print $breadcrumb; ?>
      <?php print $messages; ?>
      <?php print render($page['content']); ?>
    </div>
    <div id="sidebar_first">
      <?php print render($page['sidebar_first']); ?>
    </div>
</div></div></div>
```

layout-basic.tpl.php

The new template replaces the page.tpl.php template for the basic layout.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Step 3. Use context to choose layout

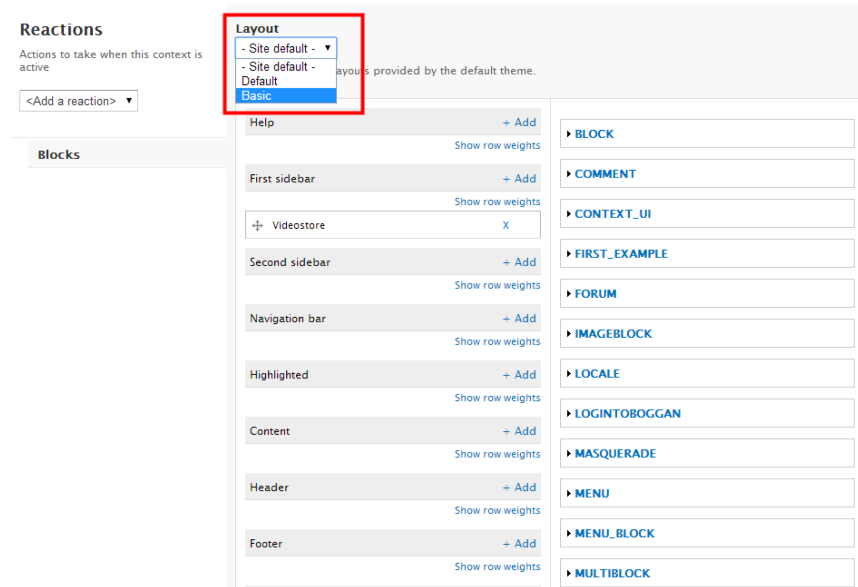
Once configured, the different layouts can be selected when we set up a context, for example to create a **Block type reaction**.

After choosing the layout, we can assign blocks to the available regions, as shown above. F32.28

F32.28

Choose layout

For a Block type reaction, we can choose one of the layouts, and assign blocks to each of the regions.



To access the option for choosing a layout, we must use the Block reaction, enable the **Context layouts** module, and use a theme that has different layouts (as defined by the **layouts** configuration variable).

Additional Modules

32.6

This section made brief mention of a few other modules that add functionality to contexts on our site.

Note: If new context reactions or actions do not appear after installing new modules, click **clear all caches** from the **Performance** page.

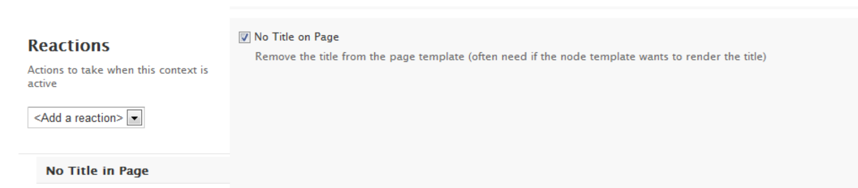
Context No Title Module

The **Context No Title** module allows us to remove the title for a piece of content, thanks to a new reaction called **No Title in Page**.

Download the module here:

http://drupal.org/project/context_no_title

When this option is enabled, the page title will not be displayed when the context is active. F32.29



F32.29

Context No Title Module

Remove the page title when the context is active.

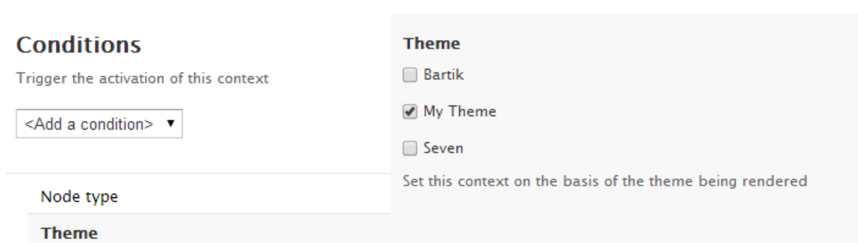
Context Condition Theme Module

The **Context Condition Theme** module adds a condition that loads a context for certain themes.

Download the module here:

http://drupal.org/project/context_condition_theme

Once installed, we can add the **Theme** condition and select the theme or themes that will load the context. F32.30



F32.30

Context Condition Theme Module

Load the context for a specific theme.

Context Node Module

The **Context Node** module can assign context to nodes.

For a content type, we can define which contexts will be applied to nodes of that type. The author of the node can choose the context to apply to the given node.

Download the module here:

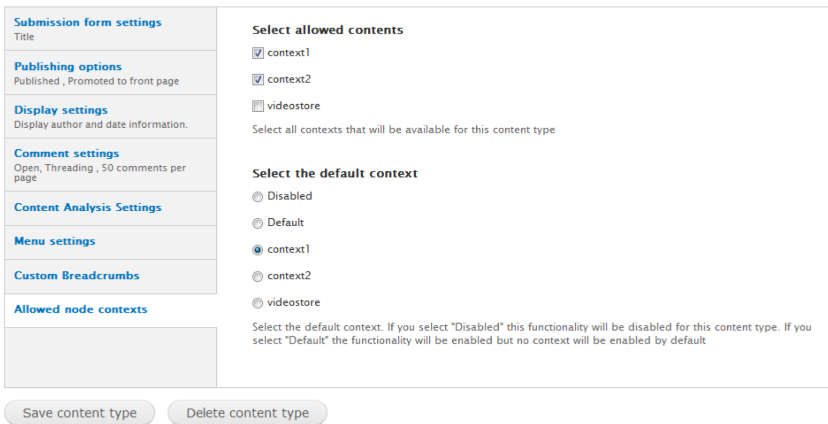
http://drupal.org/project/context_node

Once installed, we can configure the content type, the allowed contexts, and the default context. F32.31

F32.31

Context Node Module

For each content type, define the allowed contexts for the node and the default context.

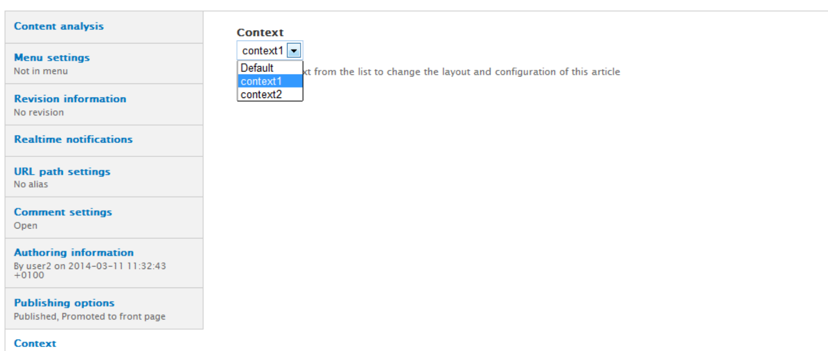


Once configured, the node edit page displays a new tab, where the author can select which context to apply to the node. F32.32

F32.32

Context Node Module

When creating a node, the author can determine which context to use, selecting from a list of allowed contexts.



This option is available if the user has permission for **Set context on nodes**.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

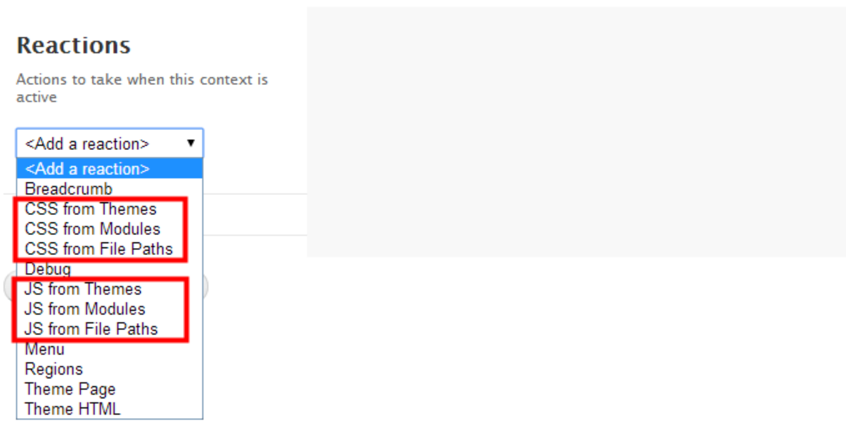
Context Assets Module

The **Context Assets** module adds a reaction that allows you to add .css and .js files in the page when the context is active.

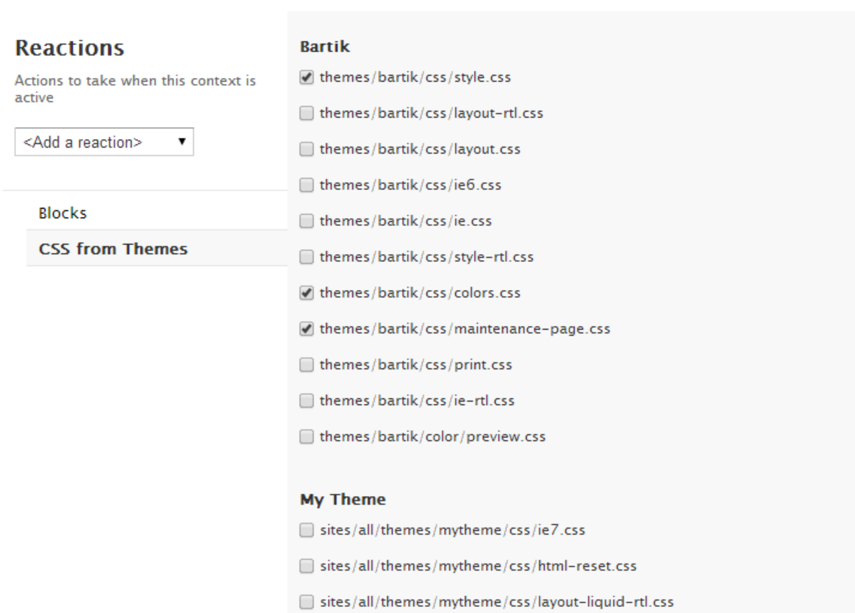
Download the module here:

http://drupal.org/project/context_addassets

Once installed (the module is identified as **Context Add Assets**), we can add CSS and JS files that are included in the active theme or module, as well as other files by providing the relevant URL. F32.33 F32.34



F32.33
Context Assets Module
Insert .css and .js files for the active context.



F32.34
Context Assets Module
Select from available files for the theme or modules, or enter a URL for additional files.

Date Context

The **Date Context** module adds a condition type that allows us to load a context for a date field.

The module is included in the **Date module**, and can be downloaded here:

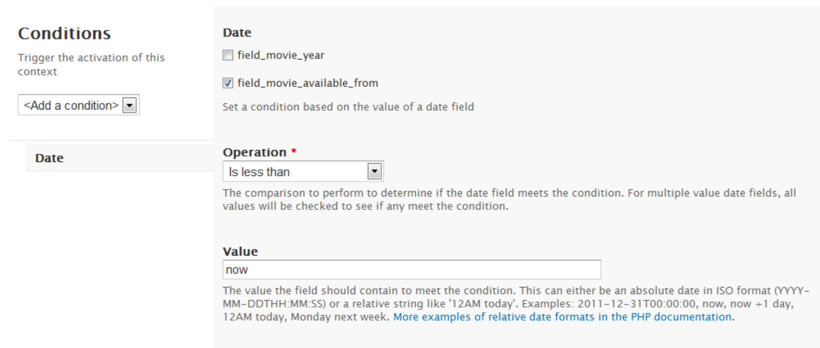
http://drupal.org/project/date

Once installed, we should see a new condition called **Date** which we can use to compare against a date field. F32.35

F32.35

Date Context Module

Compare date fields to determine when to load a context.



33 Extending users

Basic creation concepts and user management was reviewed in the initial level. In this unit, we will take a closer look at these processes and present the following modules that will provide additional functionalities:

- **Imagecache profiles**, assigns different styles to the user images (profile image, comment image, and node image).
- **Avatar Approval,** facilitates the user image moderation process.
- **Author Pane,** generates a block with user information.
- **Profile2,** creates different types of profiles and associate them to users with roles and permissions.
- **Filter permissions,** improves the management interface of permissions by incorporating additional filters.
- **Masquerade**, we can pretend to be any other user on the site.
- **Content Access,** allows access control by role to the different types of content.
- **User restrictions,** creates rules to allow or deny access to users.
- **Login Toboggan,** allows modification of the behavior of the access system or user login.
- **Login one time**, send a single-use access link to a user.
- **Login Destination**, redirect a user to specific routes when a site is accessed.
- **Views Bulk Operations**, applied to the user views, allows operation to be carried out over multiple users.

Comparative D7/D6

Some of the modules that have been studied have their corresponding version in Drupal 6, with practically the same functionality.

Others, such as Profile2, are only applicable in Drupal 7.

Unit contents

33.1 User Images 384
33.2 Multiple User Profiles 392
33.3 Access Roles and Permissions 396
33.4 User Login 405
33.5 Massive Operations Over Users 412

33

33.1 User Images

We begin this unit by learning about the advanced options to configure the registered users' images on the site.

As we learned in Unit 10 of the initial level, from **Account Options** we can modify the configuration of the user accounts and the creation policy/account cancellation:

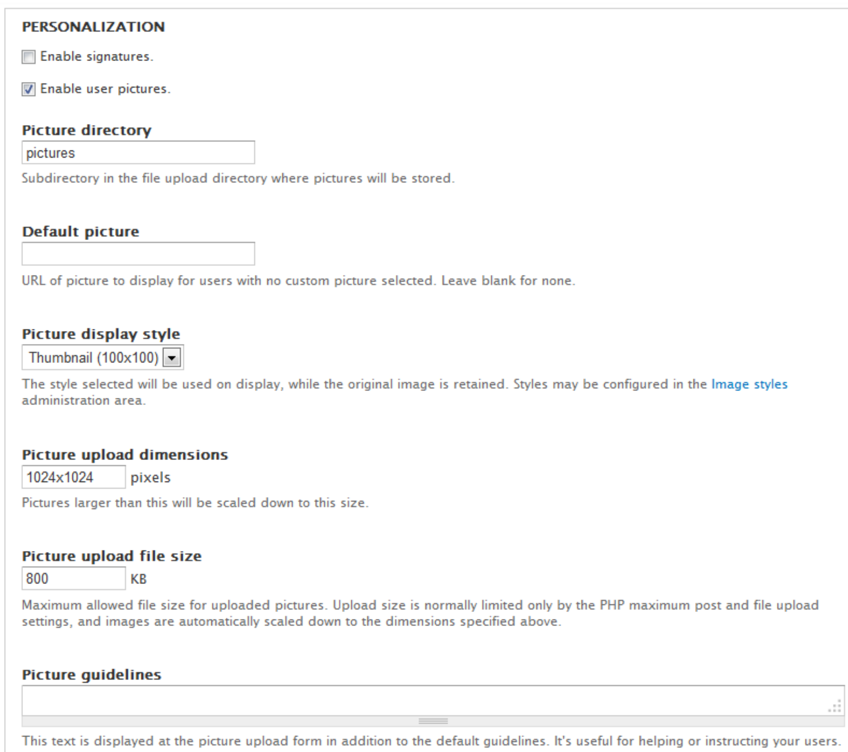
URL Account Options
/admin/config/people/accounts

Administration ⇨ Configuration ⇨ People ⇨ Account settings

Within the **Personalization** group, F33.1 we can configure the following options related to the use of images on the part of the users:

- **Enable signatures**. The user signatures are used in the publication of comments. It is a fixed text defined by the user that is added at the end of each comment.
- **Enable user pictures**. Activation of this option allows users to use an image (or **avatar**) associated with their profile.

F33.1
Personalizing User Accounts
In the Personalization paragraph, we can configure some adjustments for account visuals such as signatures and user images.



Upon activating the use of images on the site, we can configure the following parameters:

- **Picture directory**. We will indicate the folder in which user images will be saved within **sites/default/files**. By default, users' photos will be uploaded to the folder **/sites/default/files/pictures**. If it does not yet exist, the system will automatically create the **pictures** folder.

- **Default picture**. We can indicate what image will be used by default for all users that have not uploaded their own photo. To use a predetermined image, we must first upload it through FTP. For example, upload the image **default.png** to the **sites/default/files/pictures** folder and then indicate the URL address related to the site, **sites/default/files/pictures/default.png**.

- **Picture display style**. In Unit 24 we learned about image styles and how to apply them. In this field we can select from all of the image styles that we have created in our site. By default, Drupal includes three preferred styles: *thumbnail, medium* and *large.* In order to adjust the visualization of the user images, we can create a new image style and assign it as presentation of image style. Keep in mind that the image style does not apply to the predetermined image, which is why an image with the adequate size must be uploaded.

- **Picture upload dimensions**. We can establish maximum dimensions for the image that the user can upload. In case larger images are uploaded, they will be automatically reduced to the indicated size.

- **Picture upload file size.** We can also indicate a maximum size the image will occupy in KB.

- **Picture guidelines.** Here we can write a text that will be displayed to the user when he goes to upload an image. Generally we will display text on the types of images that will be considered inappropriate for the site.

F33.2
User pictures
Upon activating the user images, we can see them in the nodes and comments.

Once we have established this configuration, the users images will be displayed in nodes and comments when the user has uploaded his photo or when we configure a predetermined image on the site. F33.2

Additional Image Styles with Imagecache Profiles

By default we can only configure one user image style that will be used in the profile as well as in the nodes and comments. The **Imagecache Profiles** Module assigns three different styles according to where the user image is displayed:

- Profile image style.
- Comment image style.
- Node image style.

The **Imagecache profiles** module is available in:

http://drupal.org/project/imagecache_profiles

Upon installing and activating (the module is called **Imagecache Profile Pictures**), we can locate the new options in the same page as **Account settings**: F33.3

URL Account Options
/admin/config/people/accounts

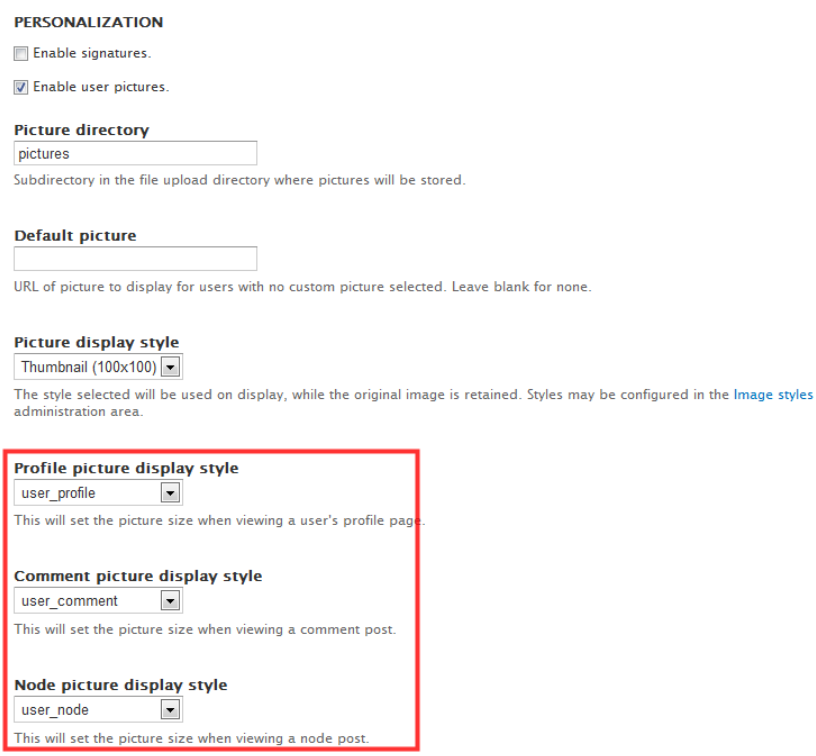
Administration ⇨ Configuration ⇨ People ⇨ Account settings

To create image styles review Unit 24:

URL Image Styles
/admin/config/media/
image-styles

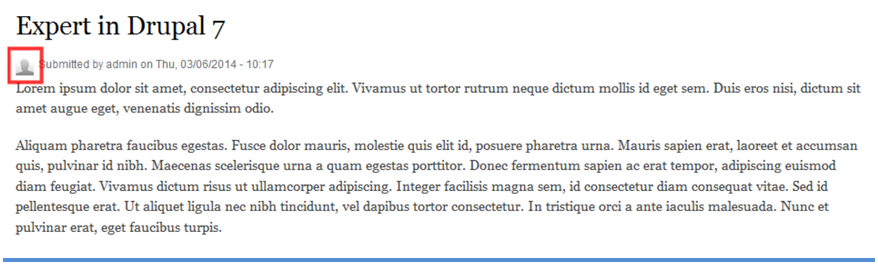
Administration ⇨ Configuration ⇨ Media ⇨ Image styles

F33.3
New Image Styles
The Imagecache Profiles Module adds three new image styles that are applied to the user image when it is displayed in the profile, a comment, or a node.



Keep in mind that some themes apply fixed image styles because they are associated to the design and modification by image styles application is not allowed. For example, in the **Bartik** theme the size of the node author image is always 20x20, and will not be applied to the defined style as **node image style.**

F33.4



F33.4

Node Image

In some themes, the styles can be given by the theme itself, therefore the changes in the Imagecache Profiles styles will have no effect on the user images. Such is the case when the user images are presented in the node (node author) in the Bartik theme.

To prove the function is correct we can switch to another theme that does not add this restriction (for example **Seven**, also included in the Drupal core).

Moderation of User Images with Avatar Approval

When we leave the option open to the possibility of uploading user images, it may be necessary to keep control of the published images in order to avoid those that do not comply with the site policies from being displayed (offensive content, images that do not show the user's face, etc.).

The **Avatar Approval** module facilitates the process of moderating user images. With this module, users' newly uploaded images will remain in an approval queue pending moderation, until a user with the adequate permission reviews them.

The **Avatar Approval** module is available at:

http://drupal.org/project/avatarapproval

Upon installing the module, we will access its configuration from: F33.5

Administration ⇨ Configuration ⇨ Content authoring ⇨ Avatar approval settings

- **Awaiting approval avatar settings**. In this section we can establish an image that will be used as a temporary avatar while the user image awaits approval. Even though the module includes a default image, it would be more correct to use the predetermined user image configured in the site.

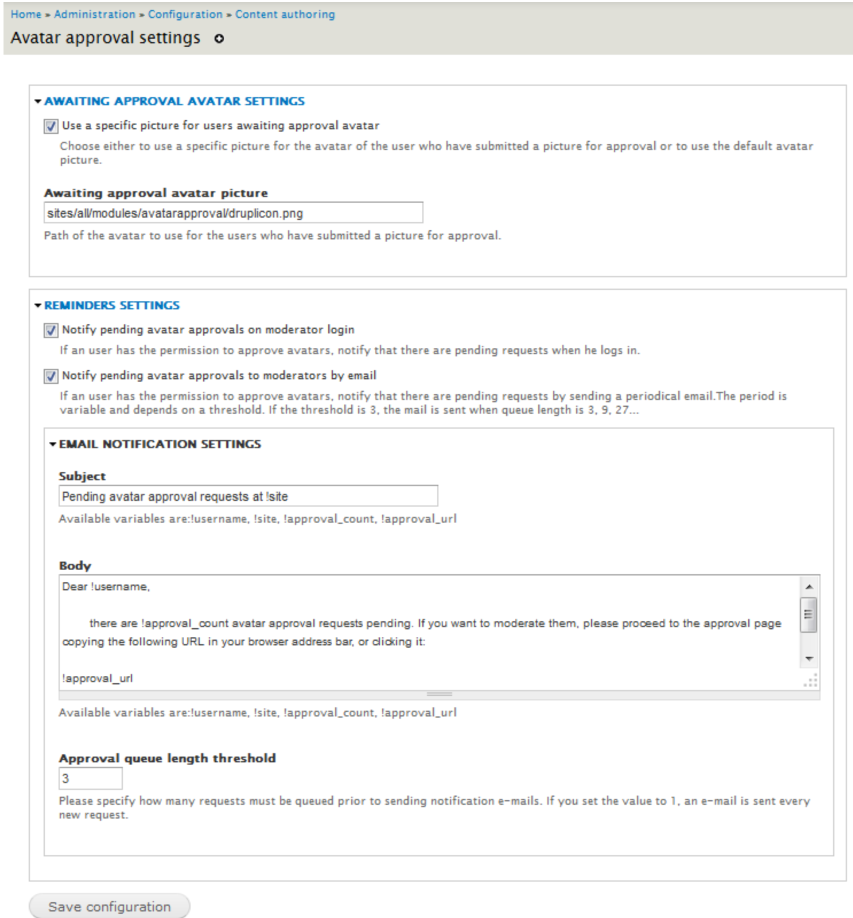
F33.5

Avatar Approval Configuration

The module offers some options to adjust the avatar approval mechanisms.

For example, we can indicate what image will be displayed by default as the avatar until the image sent by the user has been approved.

We can also establish if e-mail notifications will be sent to the moderators when there are avatars pending approval, as well as the priority with which they will be sent

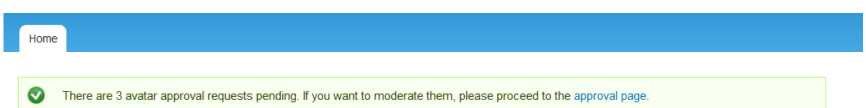


- **Notification Options**.
  - **Notify pending avatar approvals on moderator login**. By selecting this option, if there are pending approvals, a moderator user will receive a notification message upon entering the site, indicating the number of pending requests along with a link to the avatar approval page. F33.6

F33.6

Notification

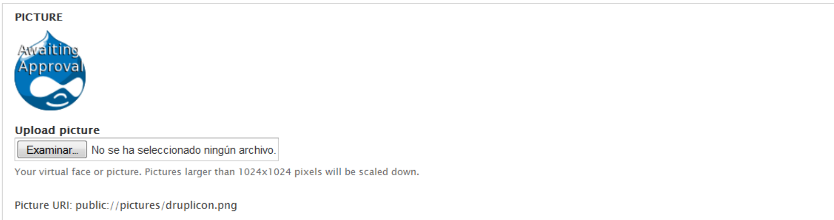
Notification message for moderators



  - **Notify pending avatar approvals to moderators by email.** If this option is activated, when there are avatars pending approval on the site, an e-mail will be sent to the users that have permission to moderate avatars. Activating this option allows configuration of the subject and body of the message that will be sent to the moderators. We can also specify the **Approval queue length threshold**, which determines how many approval requests will be pending in the queue before the e-mail is sent to the moderators. If we want an e-mail to be sent every time there is an approval request, we will set the threshold to 1.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Once the configuration has been saved, we can confirm its functionality by modifying a user's profile image (for example, userdemo). Upon saving the changes, the image will be assigned by default, indicating that the user's avatar is pending approval. F33.7



F33.7
Default Avatar
A default image will be displayed during the approval process, which will be replaced by the permanent avatar once it is approved.

Before continuing with the moderation of the user uploaded avatars, we will review the new permissions added by the module: F33.8

- **Moderate avatars.** Users with this permission will be able to access the avatar moderation page to approve or reject them (but not delete them).

- **Delete avatars.** Users with this permission can delete avatars (this permission should be used in conjunction with the Moderate avatars permission).

PERMISSION	ANONYMOUS USER	AUTHENTICATED USER	ADMINISTRATOR
Avatar Approval			
moderate avatars Allow to moderate avatars from the approval list	☐	☐	☑
delete avatars Allow to remove avatars from the approval list	☐	☐	☑

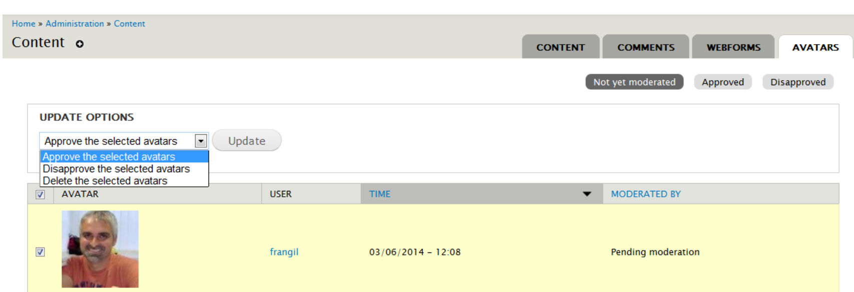
F33.8
Avatar Approval Permission
Includes two new permissions to moderate and delete avatars.

The avatar moderation page will be avialble to users with the **Moderate avatars** permission, from:

Administration ⇨ Content [Avatars tab]

URL Moderating Avatars
/admin/content/avatars

From this page we can manage all of the uploaded user images. In the image state function, this will be displayed in one of the available sub tabs: It has not yet been moderated, Approved or Rejected. F33.9



F33.9
Avatar Modification Page
From this page we can approve, reject, or delete user avatars. We see a list that shows the user that uploaded each avatar, the time it was uploaded, and who (if anyone) has modified it.

The user that uploaded each avatar is listed (with a link to their profile) along with the time it was uploaded. The name of the user that moderated the avatar is in the **Moderated by** column if it has already been done, or the message

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

"Pending moderation" if it has not yet been done.

To approve, reject or delete one or several avatars, we will first select them, then we will indicate one of the available actions on the top dropdown menu, and click on **Update** so that the action can be executed.

User Profile Block With Author Pane

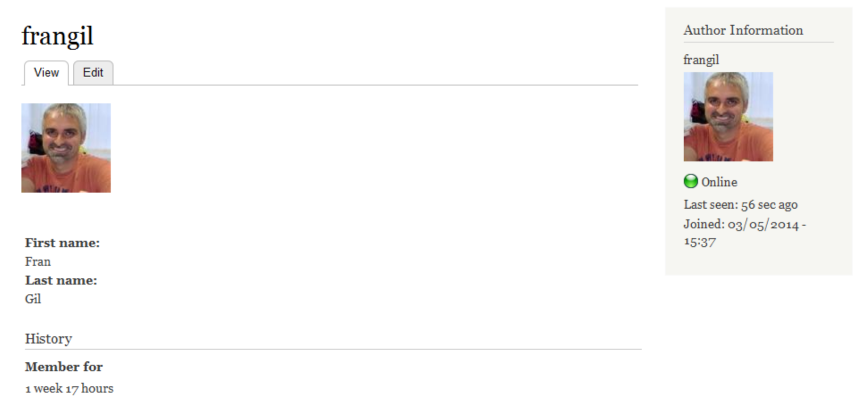
The **Author Pane** module generates a block with the user's information that can be displayed next to other site content (user profile page, forum themes published by the user, etc.). F33.10

F33.10

User Block

The Author Pane module generates a user block with information compiled from other modules.

This block is displayed on the user profile page, in the forum entries, etc.



The Author Pane module is available at:

http://drupal.org/project/author_pane

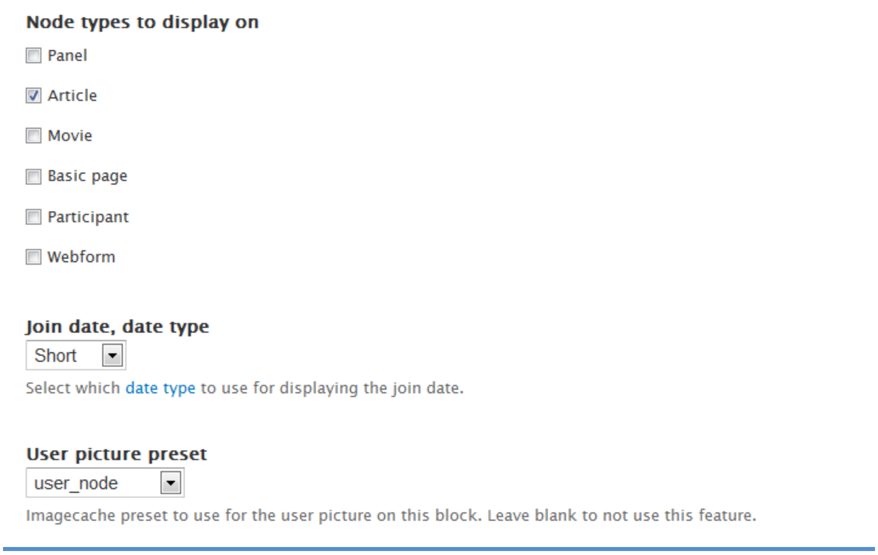
The block content is defined through a template included in the module folder, **author-pane.tpl.php**, which we can modify by copying the file in our theme, to complete with additional information.

The module receives information from other modules related to the user on order to show relevant information (number of private messages, friends, points, contacts, published themes, etc.).

The **Author Pane** module depends on the following configuration options: F33.11

- **Node types to display on**. In addition to the default pages such as the user profile page, we can select other type of content where the block will be displayed.

- **Use picture preset**. Allows selection of the image style that will be used in the block



F33.11

User Block

The user block can be displayed in the content type function.

A specific image style can also be established.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

33.2 Multiple User Profiles

The Drupal core allows the user to have a profile. Also, Drupal 7 treats the user as an entity; therefore, additional fields can be added to the user profile.

However, it is very common to need different user profiles, with differentiated fields. For example, a profile for individual users and another for companies, each one with specific fields.

The **Profile 2** module allows different profile to be created and associates users by roles and permissions. This way a user with an Individual role will have the option to complete the Individual profile, and a user with the Company role may complete a Company profile.

Since a user may have more than one profile assigned, we may also see the profiles as **group options**, which will be displayed on the user page as different tabs. For example, we can have an **Employee** profile that will have additional information for employees that may also have other profiles such as the main Profile (generic for any site user) or more specific profiles such as **Coordinator**, **Director**, etc.

The profiles from **Profile 2** are also entities; we can therefore apply all we have learned about management and presentation of fields.

The **Profile2** module is available at:

http://drupal.org/project/profile2

This module depends on **Entity API**, available at:

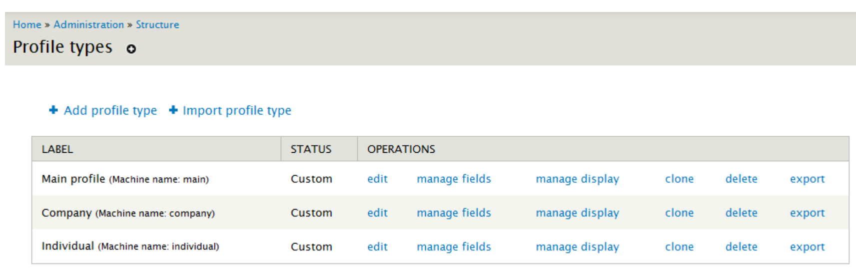
http://drupal.org/project/entity

Once both modules have been installed and activated, we can manage the different user profiles from: F33.12

URL Profile Types
/admin/structure/profiles

Administration ⇨ Structure ⇨ Profile types

F33.12
Profile Types
We can create new profile types, each with its own set of fields.



From **Add profile type** we can create new user profile types.

The module adds a first user profile type called **Main profile**. The following operations can be performed on any of the profiles:

- **Edit**. Edit the profile name and decide if the profile will be visible when

when a new user account is created in the site.

- **Manage fields**. Add fields. The function is similar to the rest of Drupal's entities.

- **Manage display**. Management of the profile display is also similar to the rest of the entities.

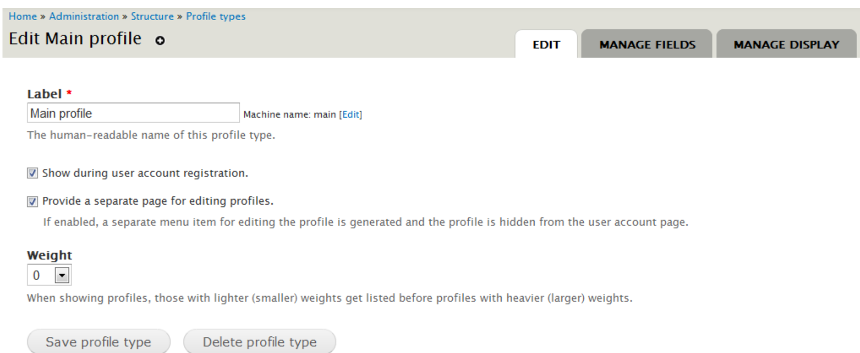
- **Clone**. Creates a new profile copied from an existing profile. Modifications can be made on the new profile once it has been created.

- **Delete**. Deletes the profile type. This option cannot be undone.

- **Export**. Export this profile type. This option gives a piece of code that can be imported on this or other Drupal site through the **Import profile type** link.

Creating or Editing a Profile Type

When creating a profile type the **Show during user account registration** option can be activated. This will cause the form to be displayed when a user registers on the site, which is why it is only recommended for main profiles with fields that affect all users. F33.13



F33.13

Main Profile

We can edit the main profile established by the module by default. We can have the profile diplayed when new user accounts are created.

Note: If we are using the Profile 2 module and want to use a default profile who's fields are displayed during the registration of user accounts, **it is best to eliminate the fields that we may have previously defined in the User entity**, from:

Administration ⇨ Configuration ⇨ People ⇨ Accounts [Manage Fields tab]

URL Account Option
/admin/config/people/accounts

This way, we will manage all of the profile fields from the newly defined profile types, avoiding duplicity in the fields and having to manage them from different pages.

Defining Profile Fields

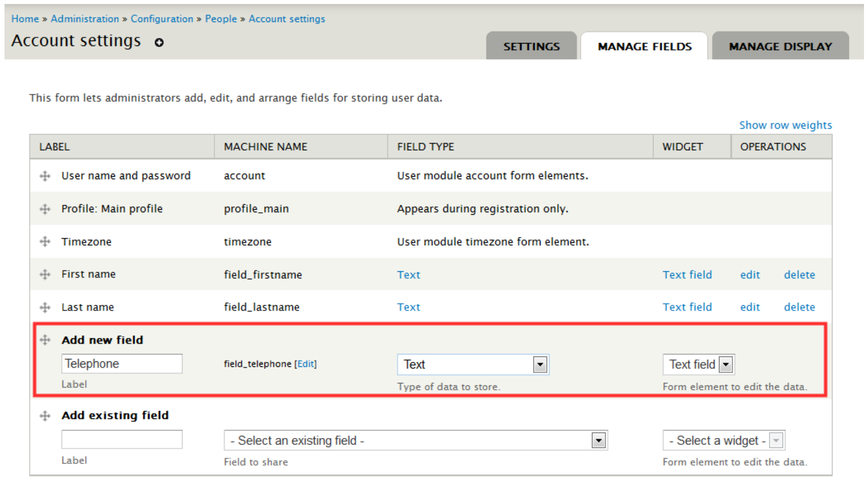
As previously indicated, profiles are entities (like content type or users), and as such we can add fields, manage the presentation, etc., all by following the steps previously learned in the lesson.

To add new fields to a profile type or edit existing ones, we will access the **Manage fields** tab. F33.14

F33.14

Adding Fields

The process of adding fields to the profile type is similar to the one previously learned for content type, taxanomy, users and other entities.

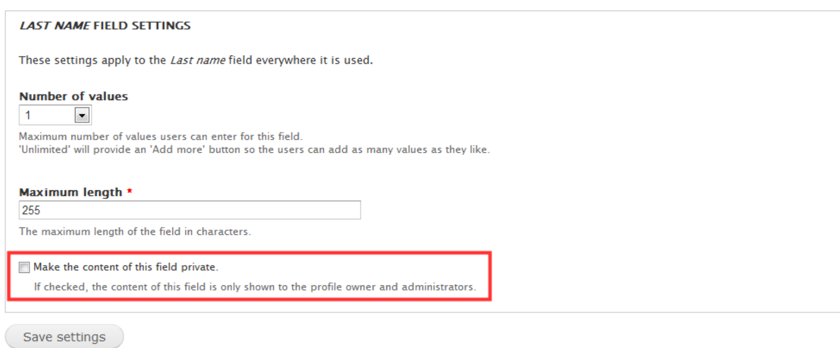


The only difference we will find when configuring a field, is the **Make the content of this field private** option. If this option is checked off, the field content will only be visible by the profile owner and the administrators (users with the adequate permission), but not by other users with access to user profiles. This allows us to define user profiles with **public fields** (first name, last name, nationality) and **private fields** (phone number, email, address, etc.). F33.15

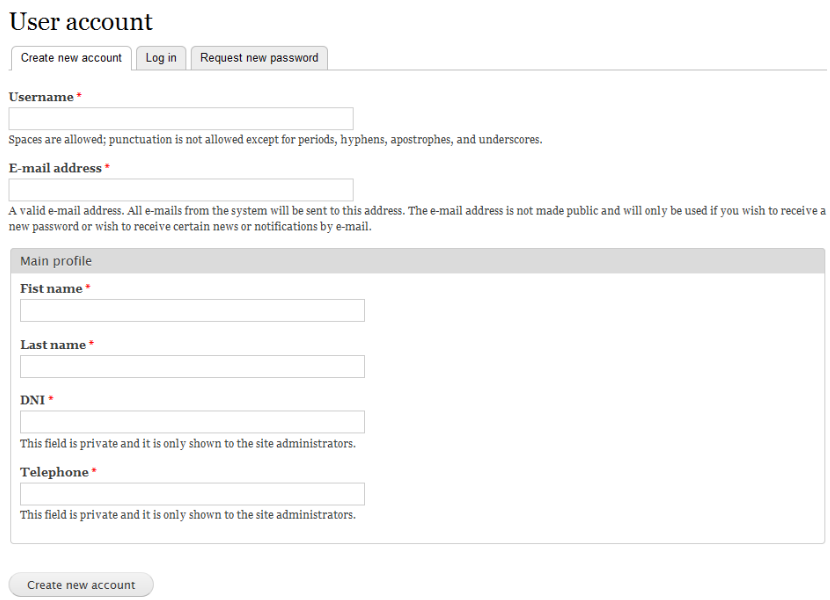
F33.15

Private Field

With the **Make the content of this field private** option, the field content will only be visible by the owners and administrators of each profile.



Fields marked as private will not be differentiated from the rest when creating the user or editing the corresponding profile. Therefore, it is best to use the field descriptions to inform the user. F33.16



F33.16

User Registry

When a user registers on the site, the name and e-mail request will be displayed, along with any profile marked with the **Display during user account registrations** option.

Field marked as private will not be differentiated from the rest during the creation of a new user or editing of a corresponding profile. It's therefore best to use the field description to inform the user.

Permissions

The module adds the following general permissions under the **Profile2** group: F33.17

- **Administer profile types**. Users with this permission can manage profile types: create and delete profile types, manage fields, etc. It is best that this permission is only assigned to the administrative user.
- **Administer profiles**. Users with this permission can view and edit all user profiles, without changing the profile types. It's geared towards moderator users that can make changes in user profiles (information inputted by the user), but without changing the profile type definitions.

For every type of profile created, the module will create the following permissions:

- ***Profile´s name*: Edit own profile**. From account editing, the user may access the edit own profile tab.
- ***Profile´s name*: Edit any profile**. The user can edit any users profile of this type.
- ***Profile´s name*: View own profile**. The user can view their own profile of this type.
- ***Profile´s name*: View any profile**. The user can view other users' profiles of this type. Keep in mind the user must also have the **View user profiles** permission given by the User module. This permission does not grant access to view private profile fields.

It is generally recommended that permissions over each profile type in independent roles be separated, so that assigning user permissions is easier later.

F33.17

Profile2 Permissions

For every profile type created, the module will give the adequate permissions to view or edit own profiles as well as others.

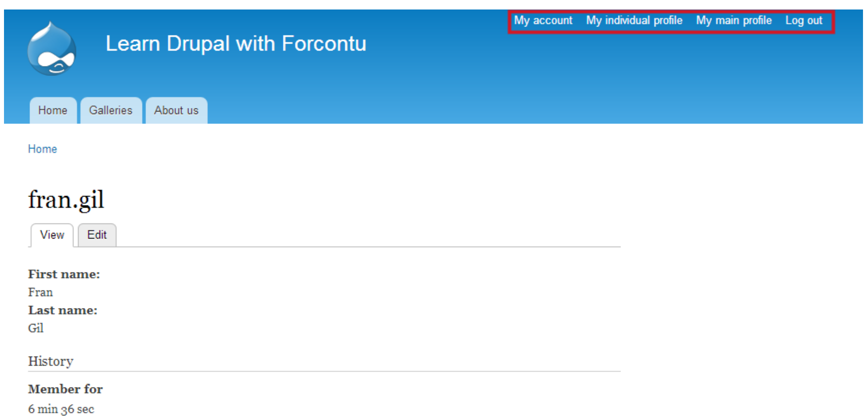
PERMISSION	ANONYMOUS USER	AUTHENTICATED USER	COMPANY	INDIVIDUAL
Profile2				
Administer profile types Create and delete fields on user profiles, and set their permissions.	☐	☐	☐	☐
Administer profiles Edit and view all user profiles.	☐	☐	☐	☐
Main profile: Edit own profile	☐	☑	☑	☑
Main profile: Edit any profile	☐	☐	☐	☐
Main profile: View own profile	☐	☑	☑	☑
Main profile: View any profile	☐	☐	☐	☐
Company: Edit own profile	☐	☐	☑	☐
Company: Edit any profile	☐	☐	☐	☐
Company: View own profile	☐	☐	☑	☐
Company: View any profile	☐	☐	☐	☐
Individual: Edit own profile	☐	☐	☐	☑
Individual: Edit any profile	☐	☐	☐	☐
Individual: View own profile	☐	☐	☐	☑
Individual: View any profile	☐	☐	☐	☐

The different types of profiles will be displayed as independent tabs in each users' account or as links in the **User menu**. Only those profiles to which the user has access will be displayed. F33.18

F33.18

Profile Tabs

The different profile types will be displayed as independent tabs in user account editing or as links in the **user menu**. Only those profiles to which the user has access will be displayed.



Access Roles and Permissions

33.3

In this section we will present some additional modules related to access roles and permissions.

Filter Module Permissions

It's important to design a good roles structure that will allow us to clearly identify what actions each user can carry out in the site.

When defining each role, we should consider the minimal set of actions that can be carried out with that specific role, and never in all of the tasks that a specific user must carry out. Keep in mind that each user can be assigned several roles; therefore it's best to group the minimal permissions together in each role, and later assign the user several roles that together will define all of the actions that it can complete.

As we install modules in the site, the number of available permissions will increase. If we add a set of defined roles to this, it will become more difficult to work with the permissions table each time.

The **Filter permissions** module improves the permissions management interface by incorporating additional filters that allow selection of what roles and modules will be displayed.

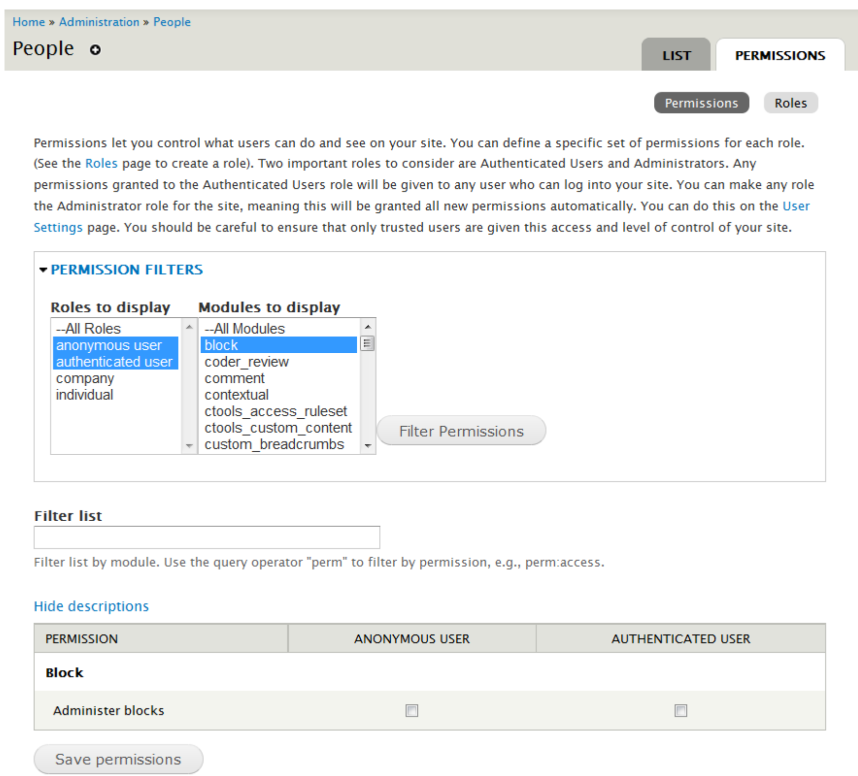
The Filter Permissions is available at:

http://drupal.org/project/filter_perms

Once it's installed, the added filters will be available at:

Administration ⇨ People [Permissions tab]

URL Permissions
/admin/people/permissions



F33.19
Filter Permissions Module
The Filter Permissions module improves the permissions administration interface by incorporating additional filters that allow the selection of the roles and modules that will be displayed.

We will select the roles we will mange from the list on the left, and from the list on the right, the module or modules whose permissions will be displayed. By clicking on the **Filter Permissions** button, the permissions table will limit itself to displaying the selected elements. The page functionality is otherwise maintained in such a way that when you click on **Save permissions,** the changes made on the visible permissions will be saved.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Note: To select several elements, push down on the Control key while clicking on the elements you want to add.

Masquerade Module

When developing a site and specially when designing the roles structure, continuous user changes are required in order to verify that the assigned roles work correctly.

Although other techniques do exist, such as opening the website in different browsers, (Chrome, Firefox, Safari, IE, Opera, etc.) and entering each site as a different user, in this section we will see a module that facilitates the user change operation without having to change browsers.

The **Masquerade** module allows us to masquerade as other users on the site.

The module is available at:

http://drupal.org/project/masquerade

Once installed we will access the module configuration from:

URL Masquerade
/admin/config/people/masquerade

Administration ⇨ Configuration ⇨ People ⇨ Masquerade

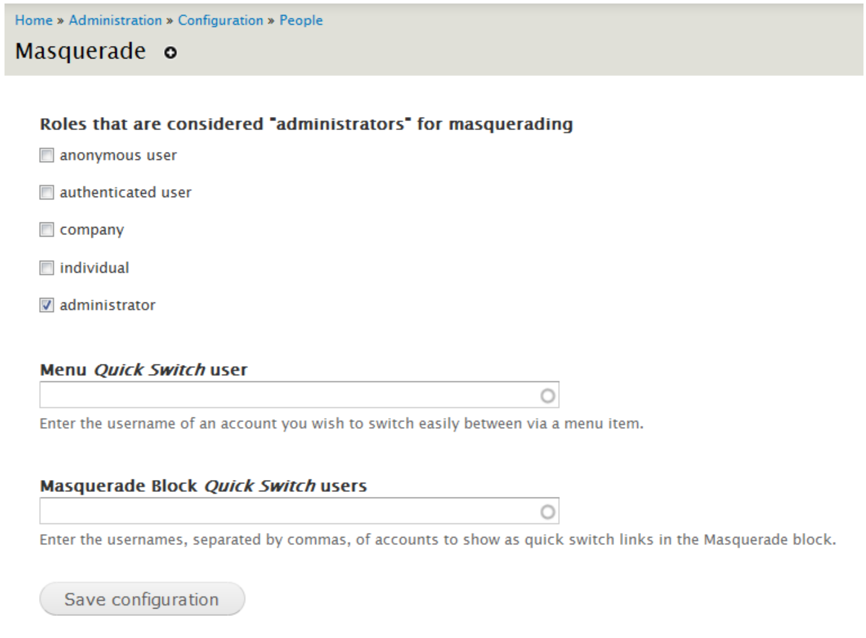
The available options are: F33.20

- **Roles**. First we will select the roles that will be allowed to use the masquerade. Only administrative users should have access to this tool.
- **Menu Quick Switch user**. A link will be added in the **Navigation** menu for every user added, which will allow the user change to be made directly. The links will have the **Masquerade as user** format.
- **Block Quick Switch users**. Links will be displayed to the indicated users on the masquerade block provided by the module. The block also includes a user search.

The menu elements and the masquerade block are displayed in the **Figure**. F33.21

In addition to the users created on the site, we can also masquerade as an **Anonymous** user. Therefore, it will not be necessary to close the current user's session to browse through the site as an anonymous user.

When we masquerade as other users, a message indicating the user name will be displayed on the masquerade block along with a link that will allow us to **Switch back** to the administrative user from where the changes are made. F33.22



F33.20

Masquerade Module

The Masquerade module allows us to pass ourselves off as any other user on the site.

From the configuration we can select what roles can use the module and establish direct links to predetermined users, from the browser menu as well as from the masquerade block provided by the module.

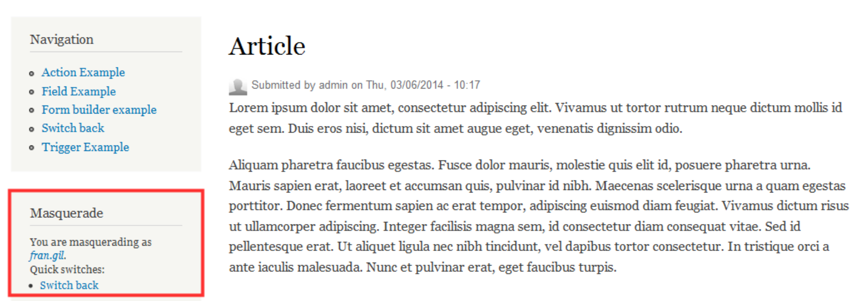
It is also possible to disguise ourselves as an Anonymous user, without the need to close the session during the development of a site in order to verify the functionality with all of the roles.

F33.21

Masquerade Block

The masquerade block is displayed in the figure, from where we can search or click on a user.

We can also add direct user links from the Browser menu.



F33.22

Masquerading

When we masquerade as other users, a message will be displayed, specifying which user it is and along with a link to Return to the original user.

Content Access Module

The **Content Access** module allows access control according to role for the different content types. It not only allows configuration of permissions for a content type, but it also allows specification of permissions for each node.

The Content Access module is available at: F33.23

http://drupal.org/project/content_access

It is composed of two modules:

- **Content Access**. Is the main module necessary to achieve access control.
- **Content Access Rules Integrations**. Aditional module that integrates access control with the Rules module, adding new events, conditions and actions (Consult the new rules once it is installed).

Once installed and activated, we will find the new module options from the editing of each content type in a new tab named **Access Control**.

Administration ⇨ Structure ⇨ Content types [Access control tab]

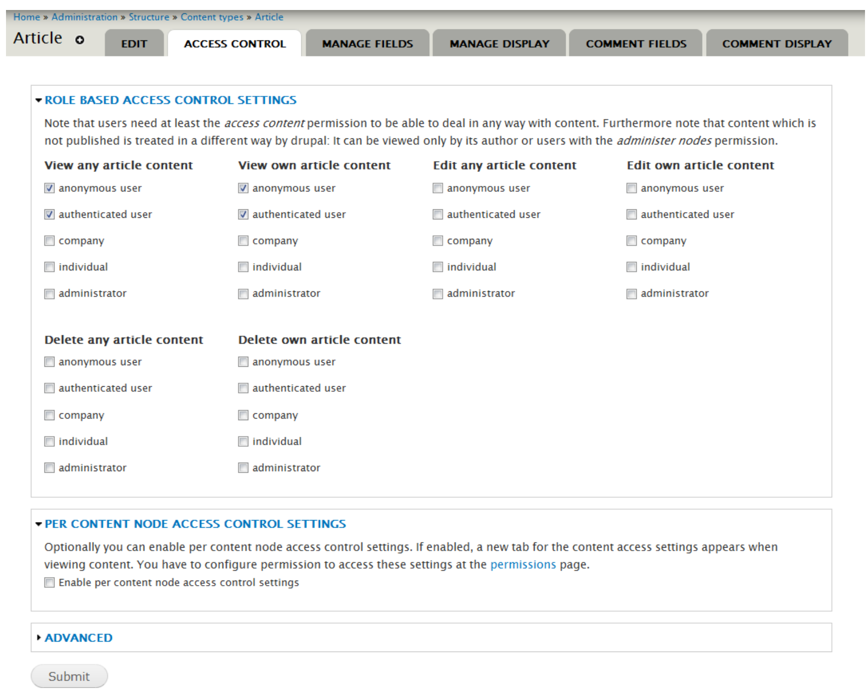
F33.23

Access Control Tab

The Content Access module provides a new tab in each content type called Access Control.

In this tab we can configure the permissions that each role on the site has over the content type.

We will also be able to indicate if permissions can be established for each individual node through the **Per content node access control settings** option.



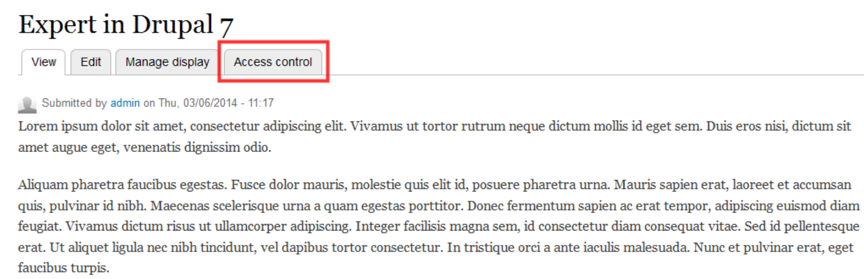
In the **Role based access control settings** section, we can configure the roles that may carry out the following actions on the content type:

- View any content (of this type).
- View own content (of this type).
- Edit any content (of this type).
- Edit own content (of this type).
- Delete any content (of this type).
- Delete own content (of this type).

By activating the **Per content node access control settings** option we can also establish specific permissions for every node. The permissions we give to each individual node will override those indicated here for the content type.

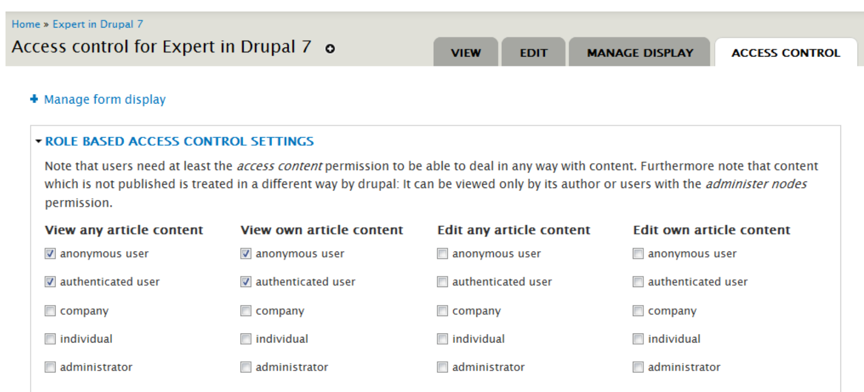
Although utilizing various content access control modules together is not recommended, if such is the case, we can control the order of priority by assigning a weight from the **Advanced** option.

If we have activated the **Per content node access control settings** option, an additional **Access Control** tab will be displayed in each node which allows configuration of the access to the node in an individual manner. F33.24



F33.24
Access Control By Node
If we have activated the **Enable per content node access control settings** option in the content type, the nodes of that type will have the Access Control tab.

In this tab we will find an interface similar to the one we've seen in Access Control for the content type. However, it will be applied to the configuration of the access to this particular node. F33.25



F33.25
Access Control By Node
With the **Access Control** tab we can establish the concrete permissions configuration for this particular node, independent from that used for its content type.

User Restrictions Module

D7/D6 Comparison
Access Rules

The User Restrictions module implements the functionality that Drupal 6 included in its core in Drupal 7 through the Access Rules.

The Access Rules were eliminated from the core in Drupal 7. Therefore, if we want to apply access rules to the Drupal 6 style, we must use this module.

When a site is exposed to the registration of any user, it may be useful to restrict access to specific names and/or e-mails, either to prevent access by spam robots or users from registering with rude or offensive words

A very useful module to accomplish this task is **User restrictions**. With this module we can create rules to allow or deny access to users according to the defined patterns.

The rules created in the **User restrictions** module acted upon the user accounts created in the site as well as on new account that will attempt to be created. The first will deny access at the moment of authentication for site entry, without deleting the account. The second will deny the creation of the account at the moment of registration by displaying an error message.

The **User restrictions** is available at:

http://drupal.org/project/user_restrictions

Note: This module is in development and may require the application of patches for its proper function. Waiting on a more stable version is recommended prior to using it in production.

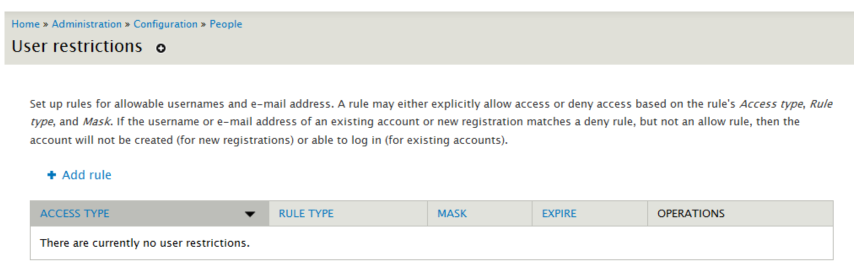
The module is composed of two sub-modules:

- **User restrictions**. Main module necessary to restrict access to users.
- **User restrictions UI**. Provide the user interface to create new access rules.

Once both modules are installed and activated, we will find the necessary tools to create and administer the rules in: F33.26

URL Access Restrictions
/admin/config/people/user-restrictions

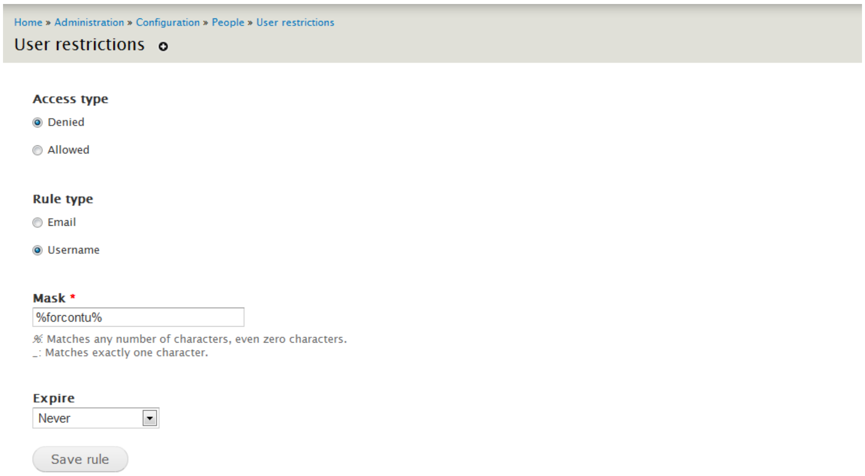
Administration ⇨ Configuration ⇨ People ⇨ User restrictions



F33.26
Access Rules Administration
On this page we can create, edit, and delete the access rules. We can create new rules by using the **Add rule** button.

The rules that have been created will be displayed on this table that is initially empty. We will have links to edit and delete rules in the operations field.

We click on **Add Rule** in order to create a new rule.



F33.27
Adding a Rule
When creating a rule, we must indicate if it will work to allow or deny access; if it will be applied on the user name or the user e-mail; the pattern that will execute the rule; and alternatively a period when the rule will no longer be applicable.

The options when creating a rule are: F33.27

- **Access type**. Select if it involves a rule to deny or allow access. The rule to **allow** may be used in some cases to override or add exceptions to the **deny** rules.

- **Rule type**. Indicates what user account parameter the rule will be applied to, allowing the option to choose between **user name** and **e-mail**.

- **Mask**. Indicates what pattern to look out for in the user name or e-mail. We can use special characters % (different characters) or _ (a specific character).

- **Expire**. We can specify a valid time period for the rule, and when it will expire and no longer be applied. If we do not want the rule to have an expiration date, we must specify **Never** on this field.

As an example, we will create a rule that denies access to accounts that have the word **forcontu** in the user name. This practice can be useful to prevent users that are up to no good, from passing themselves off as site administrators or company personnel.

To accomplish this we will use rule type "user name" and mask "%forcontu%" as "denied" access.

Since we have included the **%** character at the beginning and end of the string, the rule will apply when the word **forcontu** is found in anywhere. For example, the rule will apply to the following cases, therefore access and creation of an account will be denied:

- forcontu
- aforcontu
- forcontuz
- gestor-forcontu
- forcontudrupal
- gestorforcontudrupal

The rules created will be displayed in the list of rules, from which we may edit or delete them. F33.28

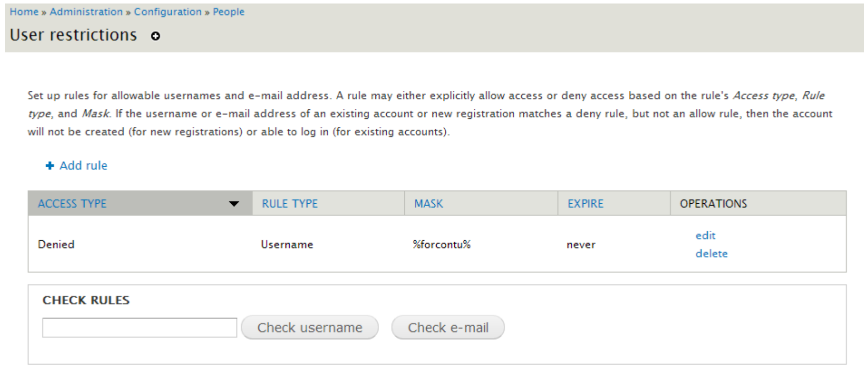
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F33.28

Verification Rules

Once we have created at least one rule, we will see the **Check Rules** section.

From her we can check if an e-mail or user name is allowed or not according to the defined rules.



When the list contains rules, a new form will be displayed that allows the verification of **check rules**. This tool allows us to check if a user name or e-mail is allowed according to the current rules and therefore figure out if the group of rules we have defined work as expected.

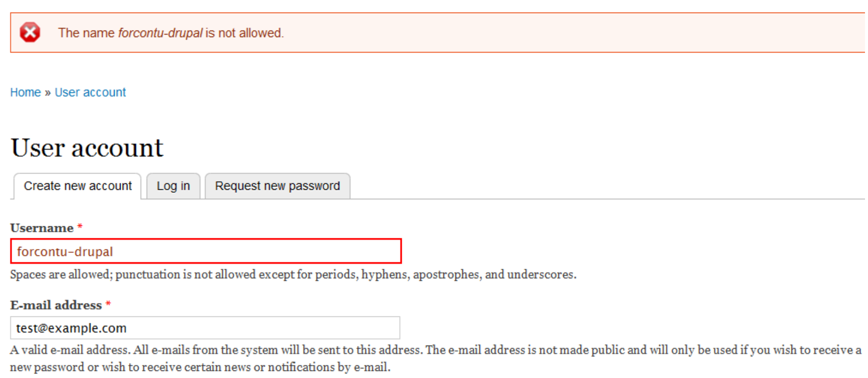
We simply have to type in a user name or e-mail and click on **Check username** or **Check e-mail**.

We can also check the rules created by attempting to create an account that is not valid. We will try this in our example by using a string that contains the word **forcontu** in the user name. As we can see, when trying to create an account with a user name that contains the pattern indicated in the rule, the system displays an error message indicating that the selected user name is not allowed. F33.29

F33.29

Denying Creation of a Rule

When attempting to create an account that is not allowed, the system will display an error message and it will not allow the account to be created.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

User Login

33.4

In this section we describe some modules that carry out operations related with access or user login.

LoginToboggan Module

The LoginToboggan module allows modification of the behavior of the system access or user login.

http://drupal.org/project/logintoboggan

The module is made up of the following submodules:

- **LoginToboggan**. Deal with the main module.
- **LoginToboggan Content Access Integration**. Integration with the Content Access module reviewed in the previous section.
- **LoginToboggan Rules Integration**. Integration with rules.
- **LoginToboggan Variable Integration**. Integration with the variables system (Variable module).

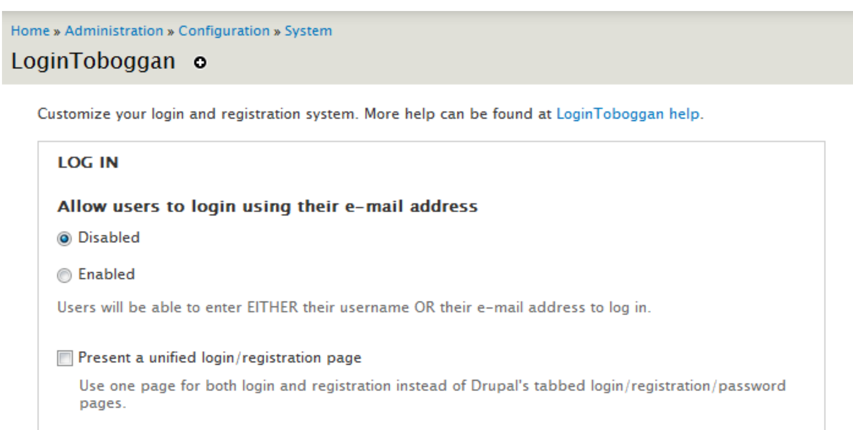
When dealing with an introduction to the module, we will activate only the main module, **LoginToboggan**. We will access the module configuration from:

Administration ⇨ Configuration ⇨ System ⇨ LoginToboggan

URL LoginToboggan
/admin/config/system/logintoboggan

Within the **Log in**-group we will find the following options: F33.30

- **Allow users to login using their e-mail address**. Users will be able to login with the user name as well as with the e-mail.
- **Present a unified login/registration page**. Instead of the default form in tabs, login options, registration and remembering a password, it will be displayed on a single page.



F33.30
LoginToboggan Module
The LoginToboggan module allows modification of the behavior of the system access or user login.

For example, it allows users to gain access with their e-mail account instead of with the user name.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Within the **Registration** group we will find the following options: F33.31

- **Use two e-mail fields on registration form**. Forces the user to confirm his e-mail with the objective of avoiding registration with erroneous e-mails.

- **Set password**. Allows the user to select a password during registration.

- **Non-authenticated role**. When we activate the **Set password** option during registration, users will have access without the need to validate the account with e-mail. In this option we can establish the "temporary" role that will be assigned to users that access the site through this path. This doesn't mean users will not have to validate their account with a link sent in the welcome e-mail.

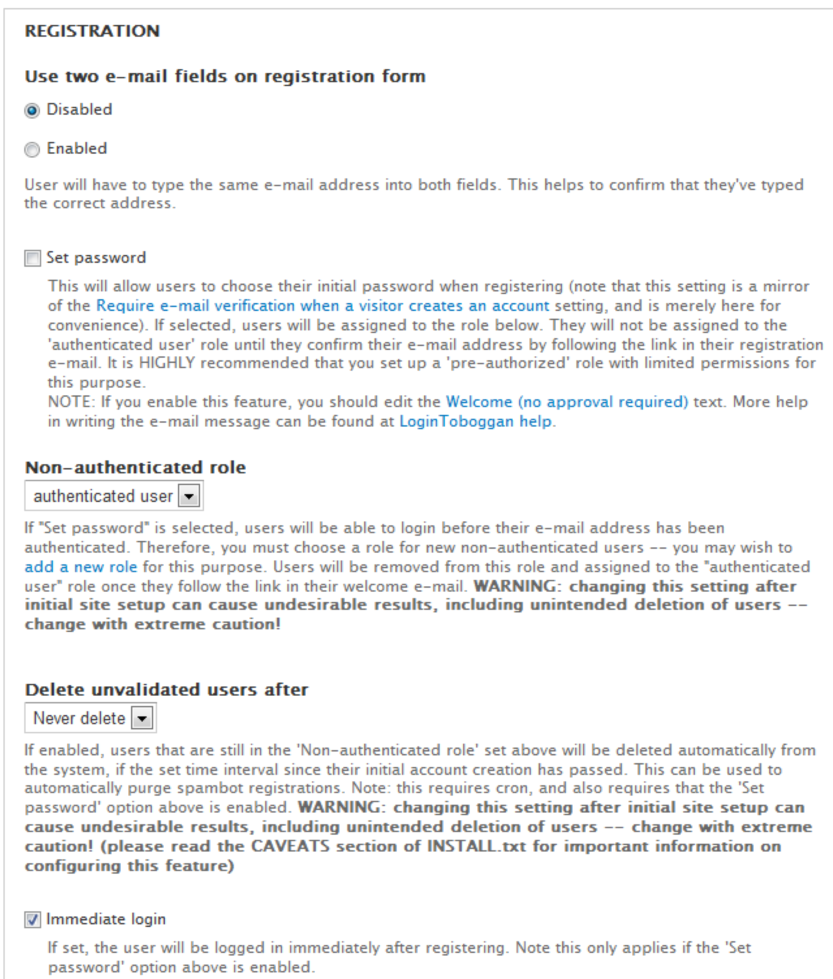
- **Delete unvalidated users after...** Allows the indication of the maximum time a user has to validate his account by clicking on the link in the welcome e-mail.

- **Immediate login**. If the **Set password** option is activated, the user will enter the site immediately after completing the registration.

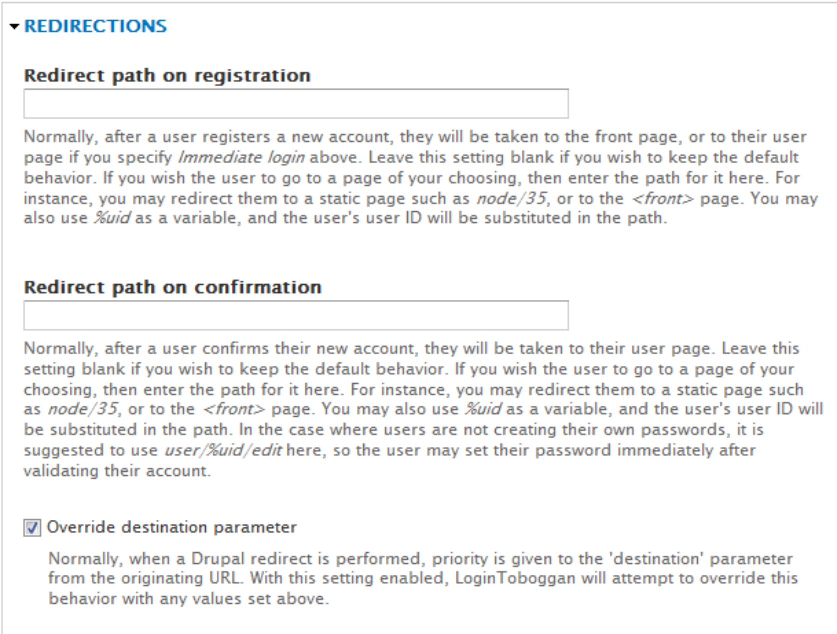
F33.31

LoginToboggan Module

The module allows the user to establish a password during registration and enter the site without waiting for the confirmation e-mail that he may validate at a later time, but always within the established time period.



From the **Redirections** section we can indicate the URL to which the user will be redirected upon registration and confirmation of the account. F33.32


▾ REDIRECTIONS

Redirect path on registration

Normally, after a user registers a new account, they will be taken to the front page, or to their user page if you specify *Immediate login* above. Leave this setting blank if you wish to keep the default behavior. If you wish the user to go to a page of your choosing, then enter the path for it here. For instance, you may redirect them to a static page such as *node/35*, or to the *<front>* page. You may also use *%uid* as a variable, and the user's user ID will be substituted in the path.

Redirect path on confirmation

Normally, after a user confirms their new account, they will be taken to their user page. Leave this setting blank if you wish to keep the default behavior. If you wish the user to go to a page of your choosing, then enter the path for it here. For instance, you may redirect them to a static page such as *node/35*, or to the *<front>* page. You may also use *%uid* as a variable, and the user's user ID will be substituted in the path. In the case where users are not creating their own passwords, it is suggested to use *user/%uid/edit* here, so the user may set their password immediately after validating their account.

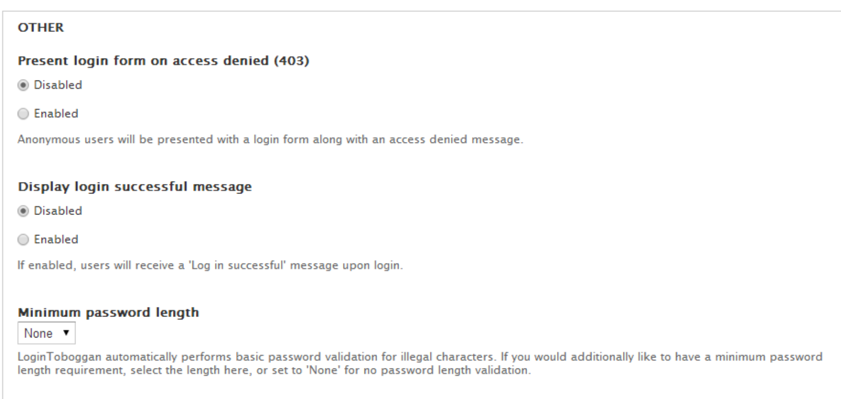
☑ Override destination parameter

Normally, when a Drupal redirect is performed, priority is given to the 'destination' parameter from the originating URL. With this setting enabled, LoginToboggan will attempt to override this behavior with any values set above.

F33.32
LoginToboggan Module
We can establish redirection to predetermined pages at the moment of registration and account confirmation.

Lastly, in the **Other** section we can modify the following options: F33.33

- Display the access form when an **access denied** (403) is produced.
- Display a **Display login successful message** when the user accesses the site.
- **Minimum password length**. The module automatically verifies that illegal characters are not introduced. We can also establish the minimum number of characters the user password must have.


OTHER

Present login form on access denied (403)

◉ Disabled

○ Enabled

Anonymous users will be presented with a login form along with an access denied message.

Display login successful message

◉ Disabled

○ Enabled

If enabled, users will receive a 'Log in successful' message upon login.

Minimum password length

None ▾

LoginToboggan automatically performs basic password validation for illegal characters. If you would additionally like to have a minimum password length requirement, select the length here, or set to 'None' for no password length validation.

F33.33
LoginToboggan Module
Other configuration options are: display registration form when the user encounters an error 403, and establish a password minimum length.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Login one time Module

The **Login one time** module allows a one-time use access link to be sent to the user.

The module is available at:

http://drupal.org/project/login_one_time

We will access the configuration from:

URL Login one time
/admin/config/people/
login_one_time

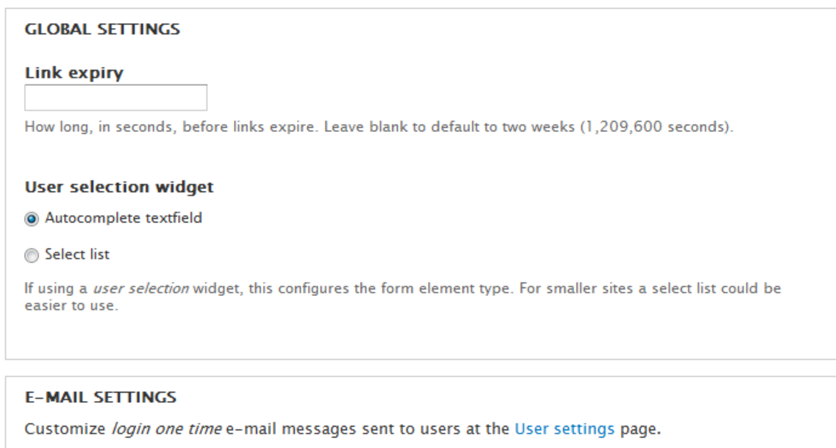
Administration ⇨ Configuration ⇨ People ⇨ Login one time

Within the **Global Settings** group we find the following options: F33.34

- **Link expiry**. Indicates in seconds, the link's time limit. The value by defect is equal to 2 weeks (1.209.600 seconds).

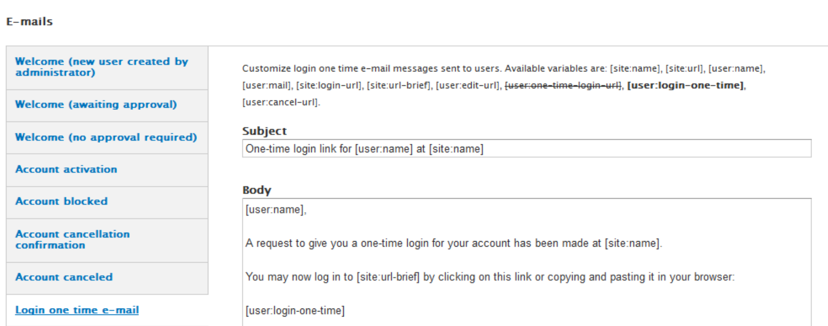
- **User selection widget**. Refers to the form field, autocomplete or selection list that will be used from administration to select the user to which an access link will be e-mailed.

F33.34
Login one time Module
The Login one time module allows a one-time use access link to be sent to the user. The user may enter the site directly, without indicating a user name and password, although it must be created on the site.



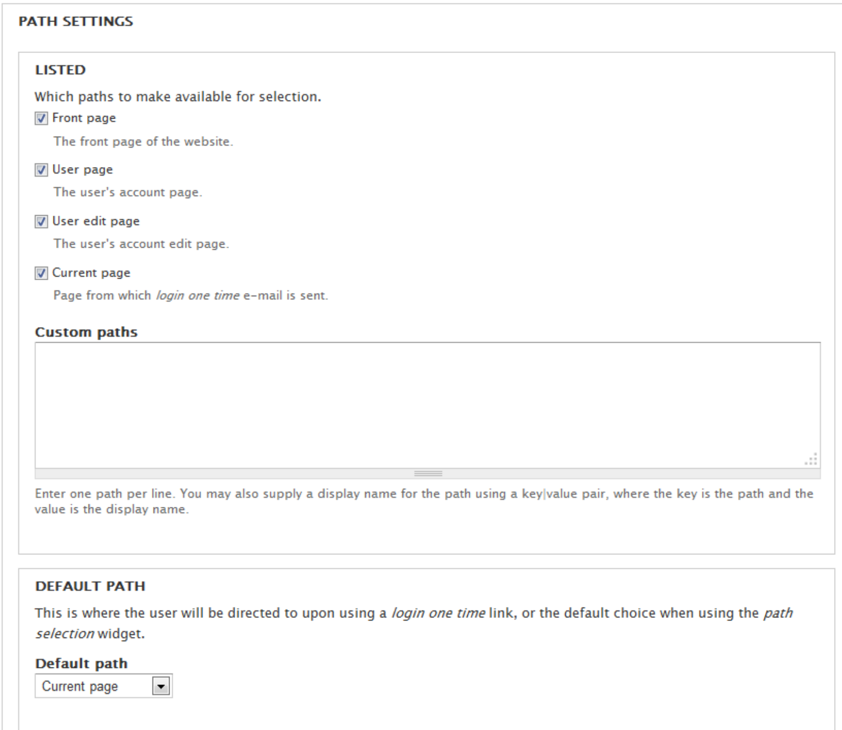
The **E-mail Configuration** will be carried out directly from the **Account settings** page. F33.35

F33.35
Login one time Module
The e-mail containing the one-time use access link that will be sent to the user, can be edited from Account Options



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

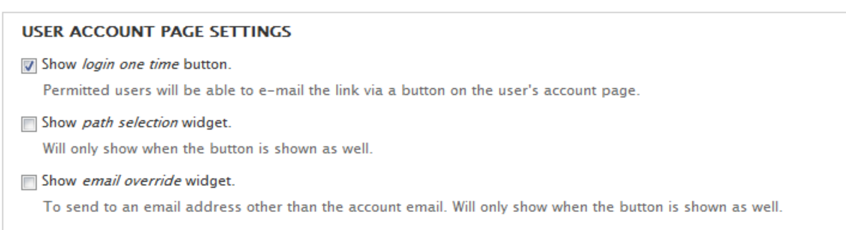
From the **Path Settings** options group we will select the paths that will be available for selection when we send the link to the user. The path selected by the administrator will be the page the user will be redirected to when accessing the site with the link given. F33.36



F33.36
Login one time Module
When sending a link to a user we will establish what page the user will be redirected to when accessing the site.

We can indicate the options that will be displayed on the user page, available only to users with the adequate permission (Permission *Send a login one time link*): F33.37

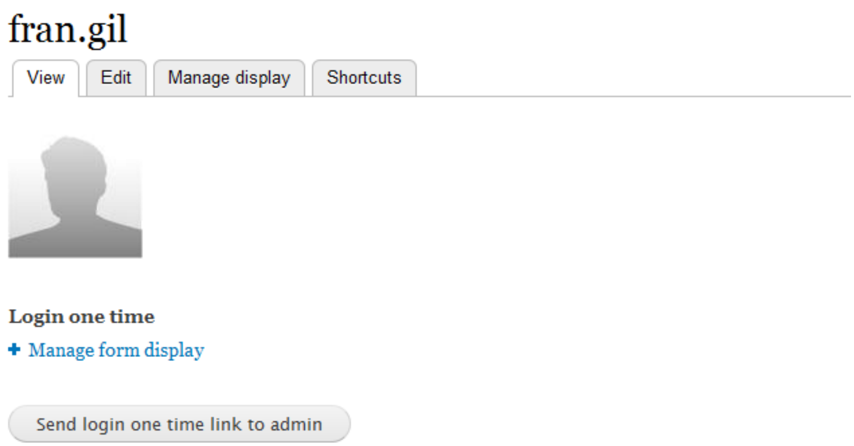
- Show login one time button.
- Show path selection widget.
- Show email override widget. Allows a different e-mail address from the one established in the user account to be indicated.



F33.37
Login one time Module
Configuration of the elements that will be displayed on the user's profile page.

The send button and the rest of the selected components will be displayed in the profile page of each user.

F33.38
Login one time Module
Users with permission to send access links will see the corresponding button on each users profile, along with the rest of the options that have been selected in the module configuration.



Finally, indicate that it is also possible to send one-time access links to multiple users using the **Views Bulk Operations** module, as we will see in the following section. F33.38

Login Destination Module

The **Login Destination** module allows configuration of the path to which the user will be redirected when accessing the site. The path can be internal or external to the site and different redirections can be applied in function parameters like the page from which it came from or the user roles.

The Login Destination Module is available at:

http://drupal.org/project/login_destination

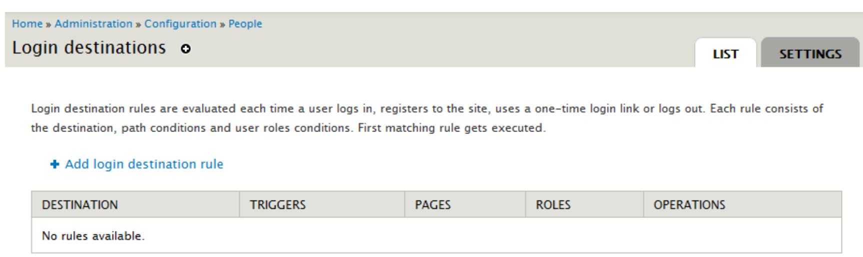
URL Login destinations
/admin/config/people/login-destination

We will access the module configuration from:

Administration ⇨ Configuration ⇨ People ⇨ Login destinations

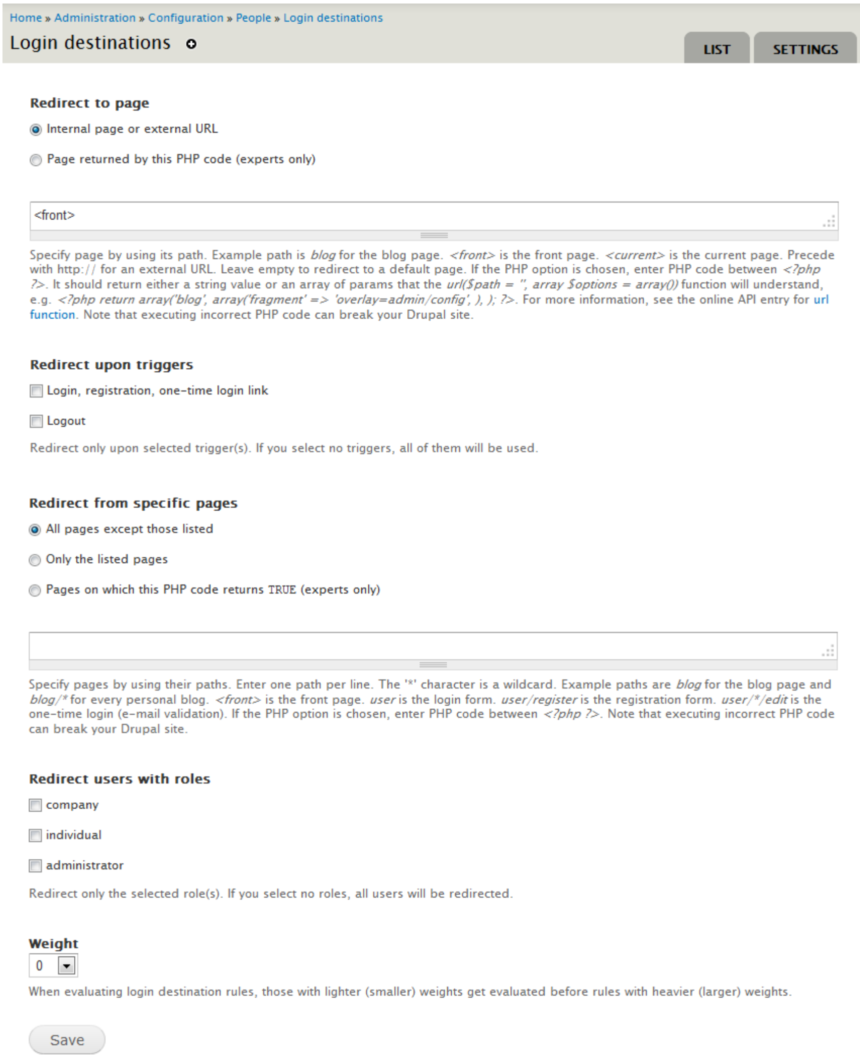
Initially an empty list will be displayed, in which the destination login rules will be F33.39 added.

F33.39
Login Destination Module
The Login Destination module allows configuration of a path to which the user will be redirected when accessing the site.



When creating a new rule we can specify: F33.40

- A **URL, internal or external**, or a URL generated with PHP.
- Activate redirection with the **login**, the **logout** or in both cases.
- Redirect only **from determined** pages.
- Redirect only when the user has some of the selected **roles**.
- **Weight** to organize the rule ahead of other rules created.



F33.40

Login Destination Module

Creating a redirection rule, we can specify the URL, the pages it will be redirected from, and the roles the user must have.

Each rule will be evaluated during the user login, logout, or in both cases, according to the selected events.

33.5 Massive Operations over Users

Some of the functionalities provided by the modules reviewed in this unit can also be applied to defining rules with the Rules modules reviewed in Unit 31.

Lets recall that with Rules we can define actions that will be executed when the user logs in, logs out, or a new user account is created, among other events. This allows us to, for example, create redirection rules similar to those generated by the Login destination module.

In this last section we review a module studied in Unit 23, **Views Bulk Operations** (VBO), with which we can realize operations over a group of users previously selected.

Massive Operations Over Users with Views Bulk Operations

The **Views Bulk Operations** (**VBO**) module adds checkboxes to the results of a view. Different operations can be carried out over the results selected by the user.

In Unit 23 we saw the application of module VBO to node type views, in such a way that the available operations were applied to the site contents (publish, place on main page, etc.).

In this section we will see what operations can be carried out with VBO over a User type view.

The **Views Bulk Operations** module is available in:

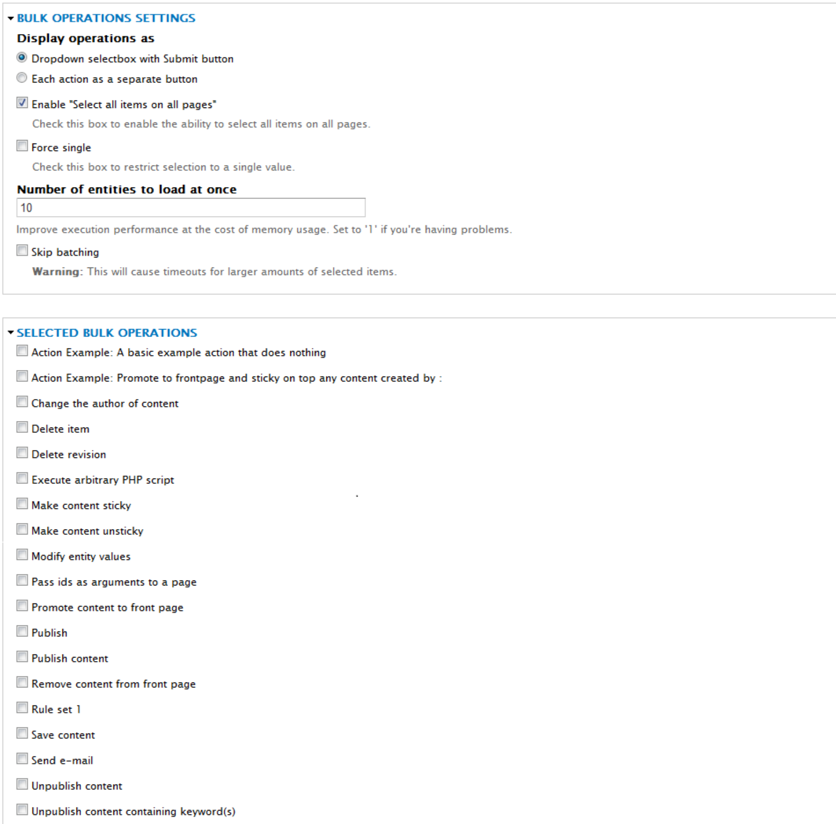
http://drupal.org/project/views_bulk_operations

Once installed and activated, we can add anew selection field in the view, such as **Bulk operations**. In a Contents type view, we can add the **Content: Bulk operations** view, and in a User view the **User: Bulk operations** view.

Upon adding a **Bulk operations** field we can configure these operations: F33.41

- Enable "**Select all items on all pages**".
- **Force single**. Change the checkbox selectors for radio buttons in a way that only one value can be selected.
- **Selected Bulk Operations**. We will select operations that can be carried out over selected users: block user, delete user, send e-mail, modify roles, show a message, etc.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



F33.41

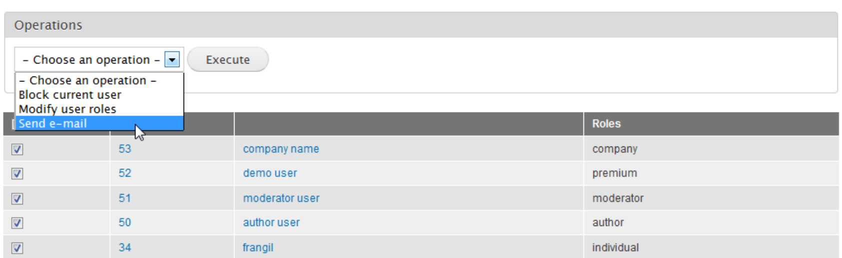
Views Bulk Operations

Configuration of a VBO type field.

The module allows VBO type fields to be added. These fields add a selector to the results, initially checkbox type. The user can select between the results and apply one of the available operations.

The Figure shows the available operations for a field that has been added in a User type view.

The resulting view will display a selection element next to each user and a new operations filter. The indicated operation will be executed over the selected users. **F33.42**



F33.42

Example of view with VBO field

Example of a view where a VBO type field has been added with three operations that act upon the selected users.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

34 Workflow

In this unit, we will introduce some modules related to controlling workflow. We will come to understand how those workflow processes or tasks must be performed in several steps, collaboratively, requiring some users to validate or approve the work of others.

The simplest workflow control example is the moderation of content: A user with the role of Author creates a page, but it will not be published until another user with the role of Moderator gives it approval. The module that allows this content review control is called **Revisioning**.

We will also see how, through the **Scheduler** module, we can schedule content to be published or withdrawn on certain dates.

Finally, we will analyze the **Node Option Premium** module for creating content which is only available to Premium members.

Comparative D7/D6

Workflow Modules

In Drupal 6 28.15 the Workflow module, which is not available in Drupal 7.

The other modules we studied, Revisioning and Scheduling, do have corresponding modules in Drupal 6 and their perfomance is virtually identical.

Unit contents

34.1 Reviewing content with Revisioning..........................416
34.2 Planning content with Scheduler 425
34.3 Access to Premium content......................................429

34

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

34.1 Reviewing content with Revisioning

The **Revisioning** module creates a control workflow specifically around the publication of content.

The module allows for the creation of content that requires approval of a moderator. On already published content, the author can make updates but the content will not republish until the moderator approves the latest revision.

The Revisioning module is available at:

http://drupal.org/project/revisioning

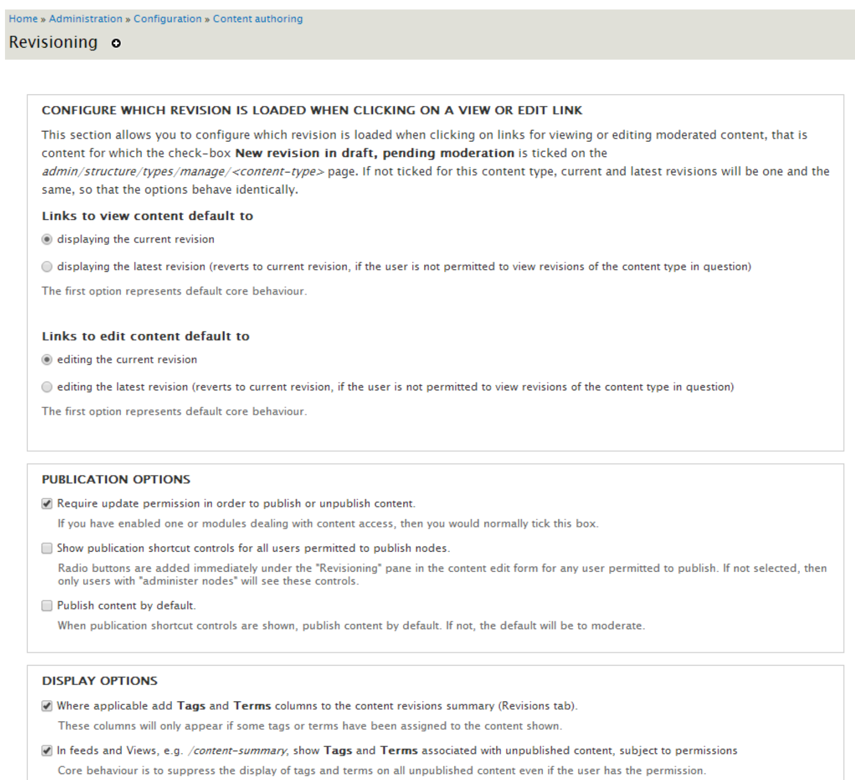
Revisioning is comprised of two modules:

- **Revisioning**. Main module that is necessary for the system to manage the revision and publication of content.
- **Revisioning Scheduler**. Allows review publishing to be scheduled on certain dates.

Upon activation of the module, you can access the setting for **Revisioning**, in: F34.1

URL Revisioning
/admin/config/content/revisioning

Administration ⇨ Configuration ⇨ Content authoring ⇨ Revisioning



F34.1
Revisioning
The Revisioning module allows users to create content that must be reviewed by a moderator.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

The available settings are:

- Configure which revision is loaded when clicking on a view or edit link. For both links, you can choose between the current version (default) or show the latest revision, even if it is not published.

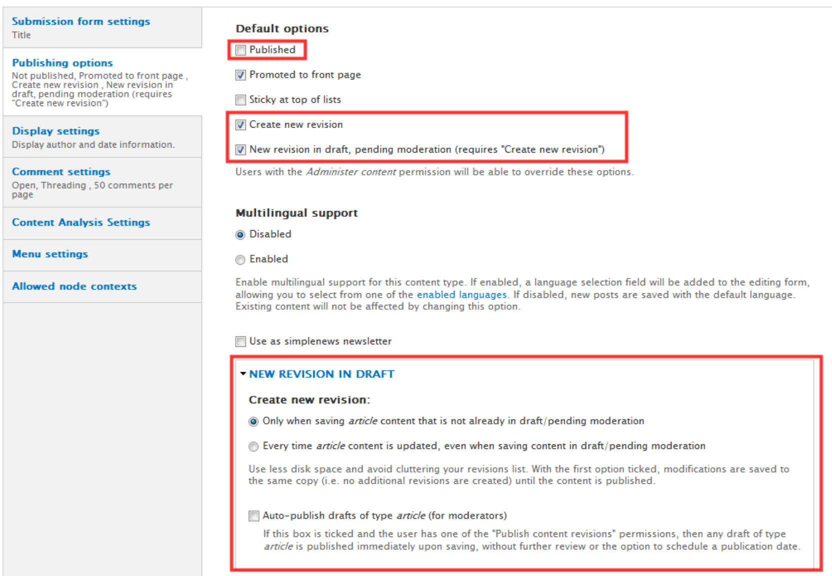
- Show tabs with content review options.

It is generally not necessary to modify the default behavior.

Configuring the content type

The next step is to configure which content type nodes will be reviewed before being published.

New configuration options are available under the group **Publishing options**: F34.2



F34.2

Revisioning. Options for publication

For each content type configured a revision is created in draft mode with publication pending moderator approval.

- **New revision in draft, pending moderation**. In order to use this option you must also select **Create new revision**.

- We must disable (uncheck) the option **Published**, as we do not want the new content posted directly.

- **Create new revision**:

  - **Only when saving content that is not already in draft/pending moderation.** When editing a revision, the changes are saved to the same copy without creating a new revision. Moderation is then done in the same node. A corresponding revision is created when the node is published and the modifications are then finalized.

  - **Every time content is updated, even when saving content in draft/pending moderation.** Creates multiple revisions of the content to be displayed in the moderation queue.

- **Auto publish drafts of type of content (for moderators).** If the moderator saves a draft it will be published directly.

Roles and Permissions

Now we must configure the roles and permissions. We will create and configure the following roles:

- **Author**. Able to create content of the **News** type, but not able to publish content directly. The content will queue pending a review by a moderator.

- **Moderator**. Able to view the content review queue and publish content once reviewed.

After creating the roles, we set the following permissions on **Node**: F34.3

- Permission for **content access control**. Neither of the roles (Author or Moderator) should be assigned this permission because it allows access to override the other assigned permissions. This permission should only be given to the Administrator.

- Permission for **Administer content**. Make active for the Moderator role.

- Permission for **Access the content overview page**. Allows access to the content management area: /admin/content/node. We can assign this permission to the **Moderator**, and even the **Author**.

  Note that this permission allows Access to the URL for content management but does not display the administration menu, therefore access is only available directly through the URL. We can always add the URL as an additional link in the navigation menu.

  Users can only perform actions on the content for which they have permissions. For example, although the author has access to all published content, each user can only edit his or her own **News** content as configured.

- Permission for **View published content**. **Anonymous user** and **Authenticated user** must have this permission.

- Permission for View **own unpublished content**. Assign to **Author** and **Moderator**.

- Permission for **View content revisions, Revert content revisions and Delete content revisions**. Assign to **Moderator**.

PERMISSION	ANONYMOUS USER	AUTHENTICATED USER	AUTHOR
Node			
Bypass content access control View, edit and delete all content regardless of permission restrictions. *Warning: Give to trusted roles only; this permission has security implications.*			
Administer content types *Warning: Give to trusted roles only; this permission has security implications.*			
Administer content *Warning: Give to trusted roles only; this permission has security implications.*			
Access the content overview page			
View published content	✔	✔	✔
View own unpublished content			✔
View content revisions			
Revert content revisions			
Delete content revisions			

F34.3

Permissions for Node in relation to revisions

Configuring permissions for the roles of Author and Moderator.

Also within **Node**, configure the specific permissions to the type or types of content: **F34.4**

- The **Author** has permission to **Create new content**, **Edit own content** and **Delete own content**.

- The **Moderator** has permission to **Create new content**, **Edit any content** and **Delete any content**.

PERMISSION	ANONYMOUS USER	AUTHENTICATED USER	AUTHOR	MODERATOR
News: Create new content			✔	✔
News: Edit own content			✔	
News: Edit any content				✔
News: Delete own content			✔	
News: Delete any content				✔

F34.4

Permissions for Node in relation to revisions

Configuring the permissions for each type of content for the roles of Author and Moderator.

Associated with the **Revisioning** module we have to assign the following permissions: **F34.5**

- Both **Author** and **Moderator** can **Edit content revisions**.

- The **Author** can see their own content reviews and the reviews of others.

-

- The **Moderator** can view and puiblish/unpublish revisions. He can also see the status of the revisions.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F34.5

Permissions for Revisioning

Configuring permissions for the roles of Author and Moderator.

PERMISSION	ANONYMOUS USER	AUTHENTICATED USER	AUTHOR	MODERATOR
Revisioning				
View revision status messages	☐	☐	☐	☑
Edit content revisions Also requires edit permission from either the Node or other content access module(s).	☐	☐	☑	☑
Publish content revisions (of anyone's content) Please select one or more content types for moderation by ticking the *New revision in draft, pending moderation* **Publishing option** at *Structure >> Content types >> edit*	☐	☐	☐	☑
Unpublish current revision (of anyone's content) Please select one or more content types for moderation by ticking the *New revision in draft, pending moderation* **Publishing option** at *Structure >> Content types >> edit*	☐	☐	☐	☑

To complete the configuration of the roles, it may be necessary to properly configure the **Text Formats**, assigning to the roles of Author and Moderator the **Full HTML** format.

Content summary view and Pending revisions block

The Revisioning module adds a new view to the **Content summary,** as shown in the navigation menu. The **Content summary** page shows a list of the published nodes, indicating if the node requires moderation and its current status, Published or Pending Review.

Initially this view is configured to be accessible by any registered user, so you must edit it to limit Access. We can, for example, limit access to users with the roles of Author or Moderator. Since this is a view, we can tweak it to suit the needs of our site (fields, filters, etc.).

There is also a new block available called **Pending revisions**, which shows the revisions pending review for the author. Note that only the current user's revisions or updates are displayed.

In **Figure F34.6** is a simple view of the **Content summary** and the **Pending revisions** block.

F34.6

Content sumary

The summary view of content to be reviewed is available once the Revisions module is activated. It will aslo activate the Pending revisions block.

Access with the Author user

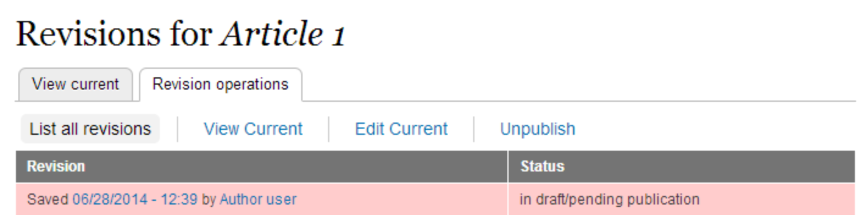
To test the performance of the permissions configured above, we will need to create two users, an **Author user** which will be assigned the role of **Author**, and a **Moderator user** to test the **Moderator** role.

Utilize the **Masquerade module**, studied in **Unit 33**, to change between users of the site without having to leave the administrator account.

The Author is able to create node of the **News** type, which remain pending publication until they are validated by the moderator.

The Author can access news nodes created, view the contents and edit them. If the news node is not published, the author will be editing the latest revision. Once modified, the changes are saved on the same revision without generating new copies.

If the author edits a published news node, a revision will be generated, which will await moderation before changes will be published. The content posted F34.7 will remain intact.

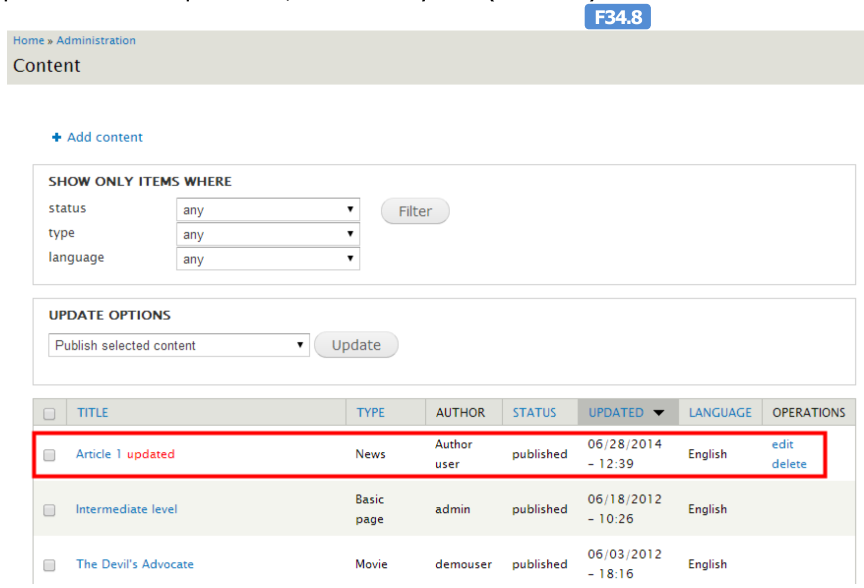


F34.7

Node Revisions

For each node the revisions pending moderation and archived revisions are displayed.

We have established that the author can access the **Content overview page**, available directly at the URL /admin/content/node, but we can also access this view in the **Content summary**. In both cases the author will see the content, published and unpublished, but can only edit (or delete) their own content. F34.8



F34.8

Content Summary

Administration page for managing content. Although the author can see all pyublished content, he can only edit the nodes of content types he has been given access to.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Access with the Moderator user

The Moderator user can edit and publish the revisions submitted by the Author.

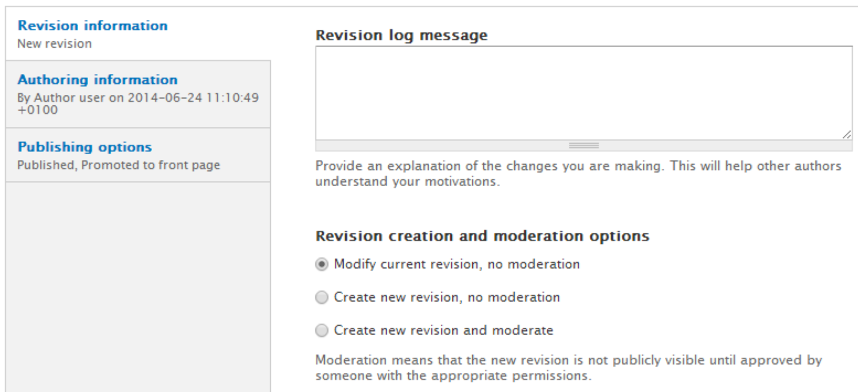
When editing a revision, you can specify one of these options in the Revision Information tab: F34.9

- Modify current revision.
- Create new revision.
- Create new revision and moderate. The revision is pending approval by someone with approriate permissions.

F34.9

Moderation

The moderator is able to edit the node and decide if the update will be saved on the same revision or if it will generate a new revision.



To publish the revision we need to access the Revisions tab of the node, and click Publish. Therefore, if the News node requires some modification, you must first edit it, save the revision, and then publish it. F34.10

F34.10

Publish Node

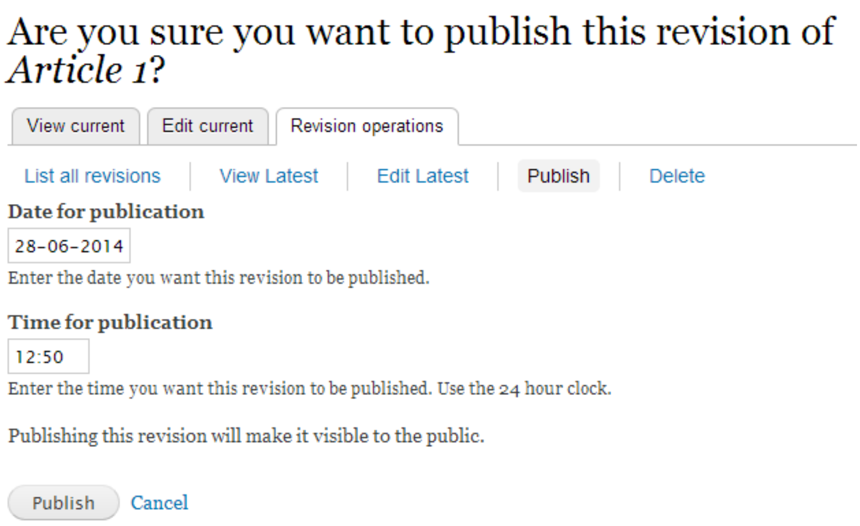
With the publish option the moderator can publish the latest revision available.

Publication Date

If you have activated the **Revisioning Scheduler** module, included in Revisioning, when publishing a node you can schedule a date for publication so that the node will be automatically published on that date. F34.11

If you want to publish the content immediately, simply leave the default date (Current Date).

Please note that publication of content on the scheduled date depends on the system cron execution.



F34.11

Publication Date

The moderator can decide whether the node is published immediatley (by leaving the defaut date) or if the publication will be scheduled for another date. The **Revisioning Scheduler** must be enabled for this option to be available.

Integration with Actions and Triggers or Rules

Whether we use actions and triggers through the Triggers module or if we use the Rules module, we will have additional trigger and actions to configure in relation to content revisions.

Utilizing **Triggers,** we are able to configure the following actions and triggers:

- **Actions**:
    - Delete archived revisions.
    - Publish the most recent pending revision.
- **Triggers**. These are grouped under the new **Revisioning** tab:
    - When publishing a pending revision
    - When unpublishing the current revision.
    - When reverting to an archived revision.

We can also make use of other available actions, for example, sending an email to the author of the node when the pending revision is published.

Utilizing **Rules,** we have the following additional elements available:

- **Events**:
    - Content has been reverted to revision.
    - Content has been unpublished.
    - Content is going to be reverted to revision.
    - Content revision is going to be deleted.
    - Node revision is going to be published.
    - Pending revision has been published.
- **Conditions**:
    - Content has pending revision.
    - Content revision is.
- **Actions**:

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- Load current revision of content.
- Publish the most recent pending revision.

Comparing revisions with Diff

The **Diff** module allows the comparison of different versions of the same node. This module is available at:

http://drupal.org/project/diff

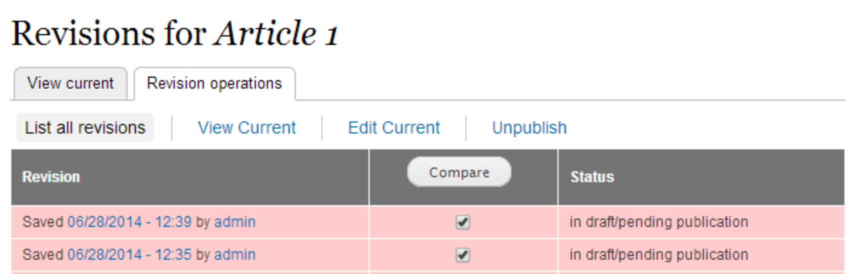
For testing purposes we first have to create two revisions of the same node. We can do this, for example, by editing a revision as Moderator, and selecting Create new revision.

Once there are multiple revisions in the Revisions tab you can select the nodes you wish to compare. Once selected you can begin the comparison by clicking the **Compare** button. F34.12

F34.12

Comparing revisions with Diff

Using the Diff module it is possible to compare different versions of the same node.



The comparison displays the complete content of each revision with the differences notated in F34.13 red text.

F34.13

Comparing revisions with Diff

The differences between the compared versions are notated in red text.

Planning content with Scheduler

34.2

The **Scheduler** module allows for node publication and node retirement based on specific dates.

To avoid conflicts with the date scheduling aspects of the Revisioning module, **deactivate the Revisioning Scheduler module before installing Scheduler**.

The Scheduler module is available at:

http://drupal.org/project/scheduler

Following the activation of the module, you can access the **Scheduler** configuration area at:

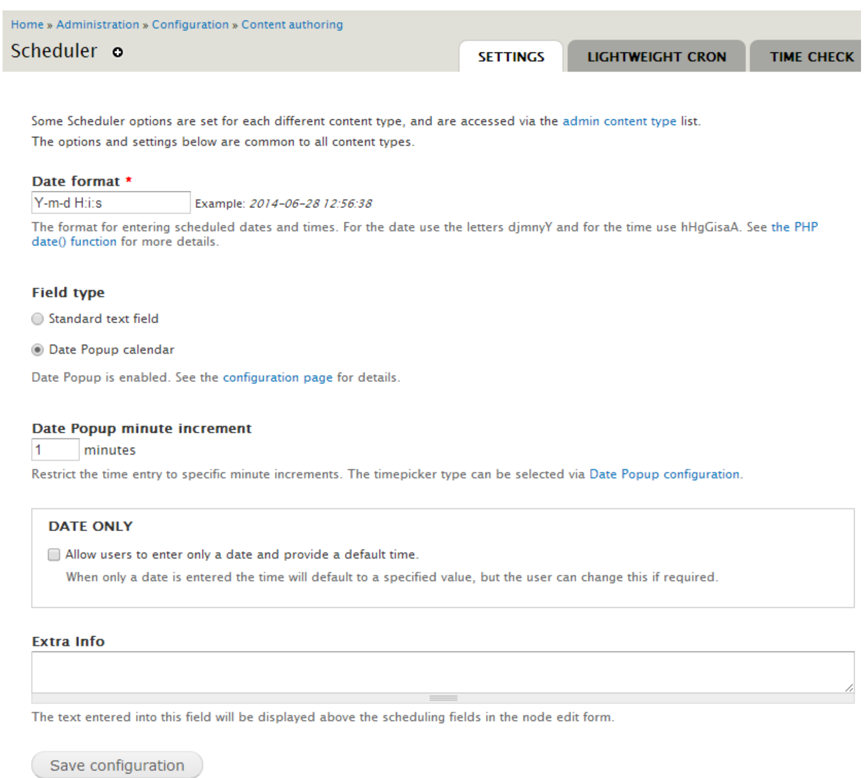
Administration ⇨ Configuration ⇨ Content authoring ⇨ Scheduler settings

URL Scheduler
/admin/config/content/scheduler

Settings

The administration area for Scheduler is divided into three tabs:

- **Settings**, here we can set the date format and the field type used to enter the date (text field or popup calendar). F34.14



F34.14
Scheduler Options
Allows the configuration of the date format and the field type used to enter the date (text field or popup calendar)

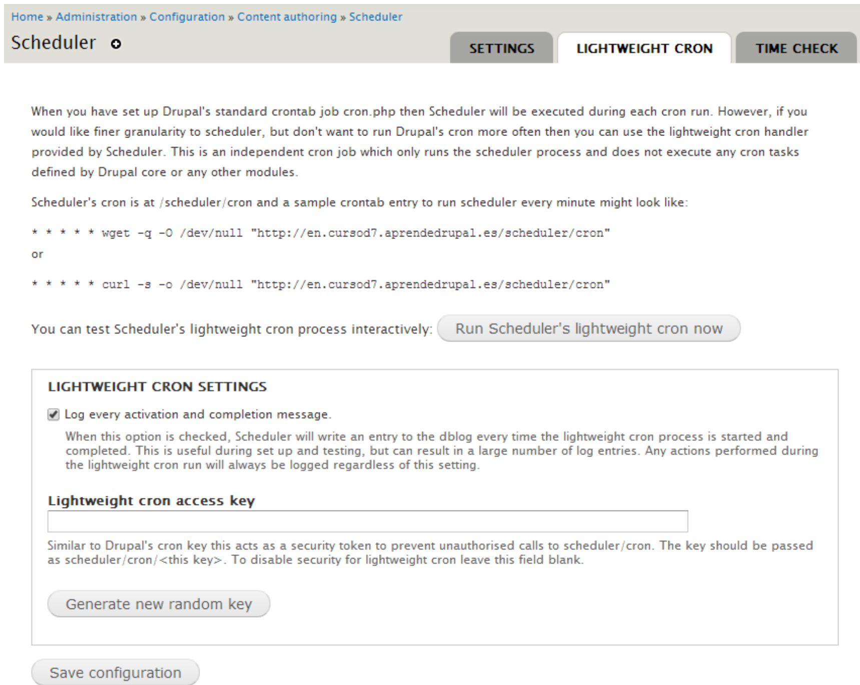
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Lightweight cron**, includes instructions to perform tasks for Scheduler in a separate cron that the system cron. F34.15

F34.15

Scheduler Cron

The module allows for the creation of a cron on the server specificly for the module, which is much lighter that the system cron and can be run more frequently.

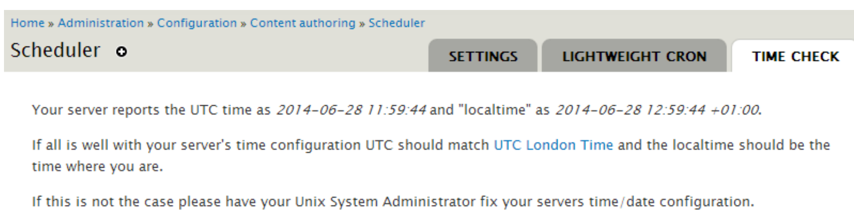


- **Time check**, displays the systems date/time information so you can verify that the settings are correct. F34.16

F34.16

Time Check

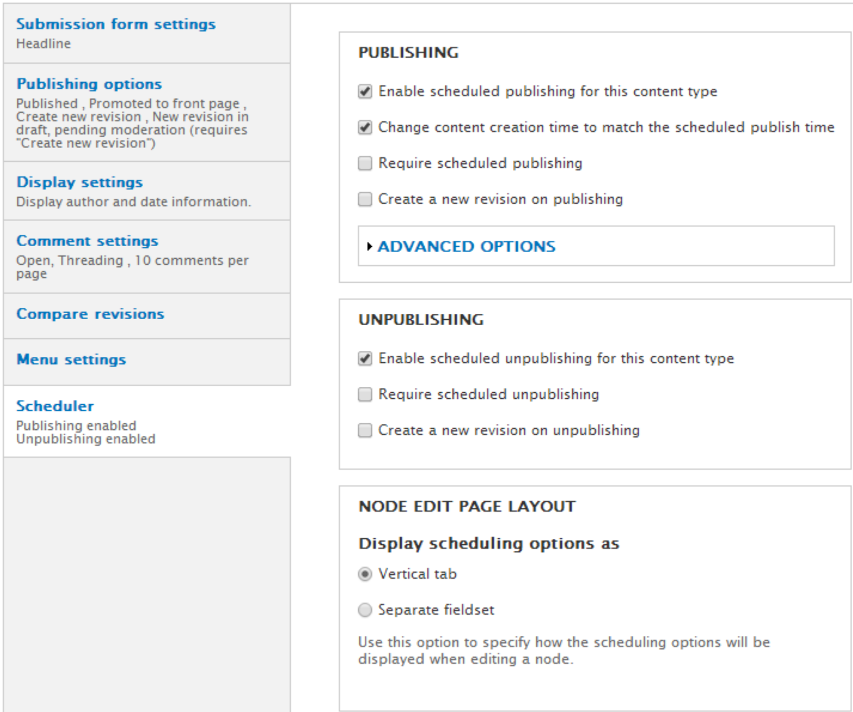
Contains system date/time data so you can verify that it is configured correctly.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Configuring content types

For scheduling to be available when you create a node, we must first activate scheduling in the content type. F34.17



F34.17

Configuring content types

For each content type we can set the scheduling settigs to both publish and unpublish content.

The options available in content type, under the **Scheduler settings** group are:

- **Publishing Settings**:
    - **Enable scheduled publishing**. This is the primary option to activate the module and allow the scheduling of publication for nodes.
    - **Modify publication date**. If this option is activated, the publication date of the node will be modified to match the planned date.
    - **Publishing date/time is required**.
    - **Create a new revision on publishing**.
- **Unpublishing Settings**:
    - **Enable scheduled unpublishing**. The option is necessary to allow the scheduling of a date when the node will be unpublished.
    - **Unpublishing date/time is required.**
    - **Create a new revision on unpublishing**.

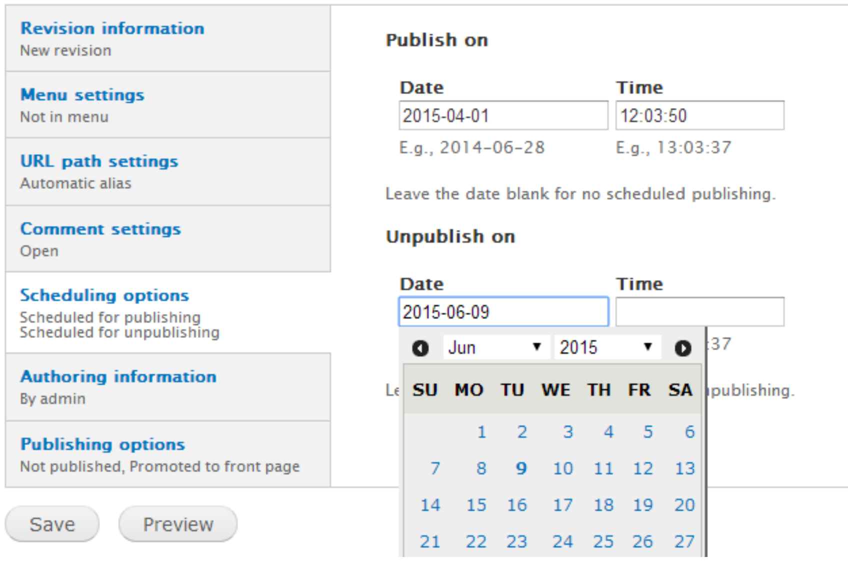
Scheduling the publication of a node

If planned publishing or unpublishing is activated for a particular content type, upon creation of a node the planning dates can be set under **Publishing Options**. F34.18

F34.18

Scheduling

For those content types that have been activated, you can select the publication and/or the unpublication dates for the node.



Through the combination of the **Revisioning** and **Scheduler** modules, we can program both the publication of a new node, as well as the publication of updates to that node. In this later case, the previous version will remain published until the planned arrival date, then the new version of the node will be published and the previous version archived.

List of planned content

URL Planned Content
admin/content/scheduler

The nodes planned through Scheduler are listed in: F34.19

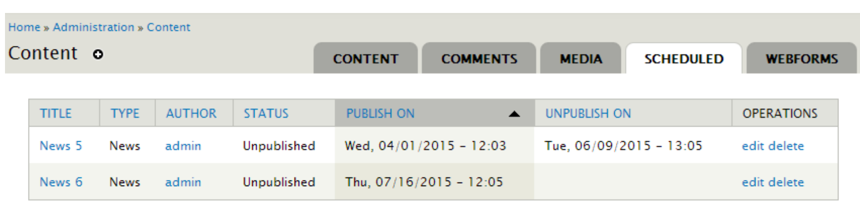
Administration ⇨ Content [Scheduled]

Once the planned action is executed, the node is removed from the list.

F34.19

Planned Content

Pending schedules are shown in Planned tab in the content management area.



Access to Premium Content

34.3

There are different strategies to control access to certain content. For example, using the Taxonomy Access Control modules, we can control which roles can access which content based on the contents associated taxonomy terms.

This section considers the **Node Option Premium** module, which adds a new option for converting nodes into **Premium content**.

When a node is published as **Premium**, only those users with the appropriate permissions are able to view the entire contents of the node, while others will only have access to a summary.

Additionally, if a user without **Premium** permissions tries to access a Premium node, a message will be displayed informing the user that the node content is only available to **Premium** users.

The **Node Option Premium** module is available at:

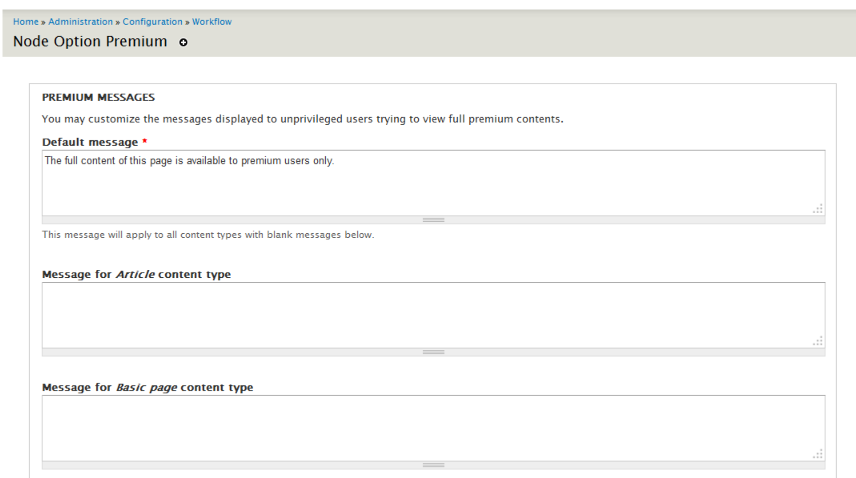
http://drupal.org/project/nopremium

Following the activation of the module, you can configure it at: F34.20

Administration ⇨ Configuration ⇨ Workflow ⇨ Node Option Premium

URL Node Option Premium
admin/config/workflow/nopremium

We are able to define the message displayed that indicates that the node is available only for Premium users for each content type. It is possible to define a default message as well as a specific message for each content type.



F34.20
Node Option Premium
Configuring warning messages to users attempting to access Premium content. These users can only access the summary of the content.

Configuring content types

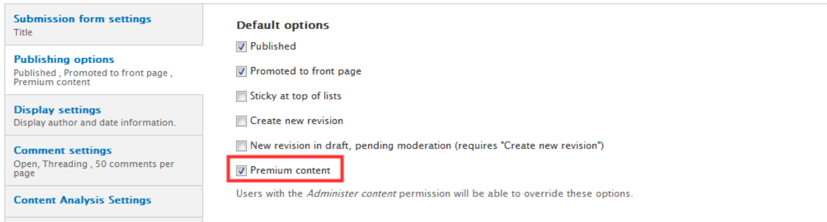
In the Publish Options of the content type you can specify whether content is to be created as Premium. If the user have appropriate permissions, she can change this option at node creation. F34.21 F34.22

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F34.21

Premium Content

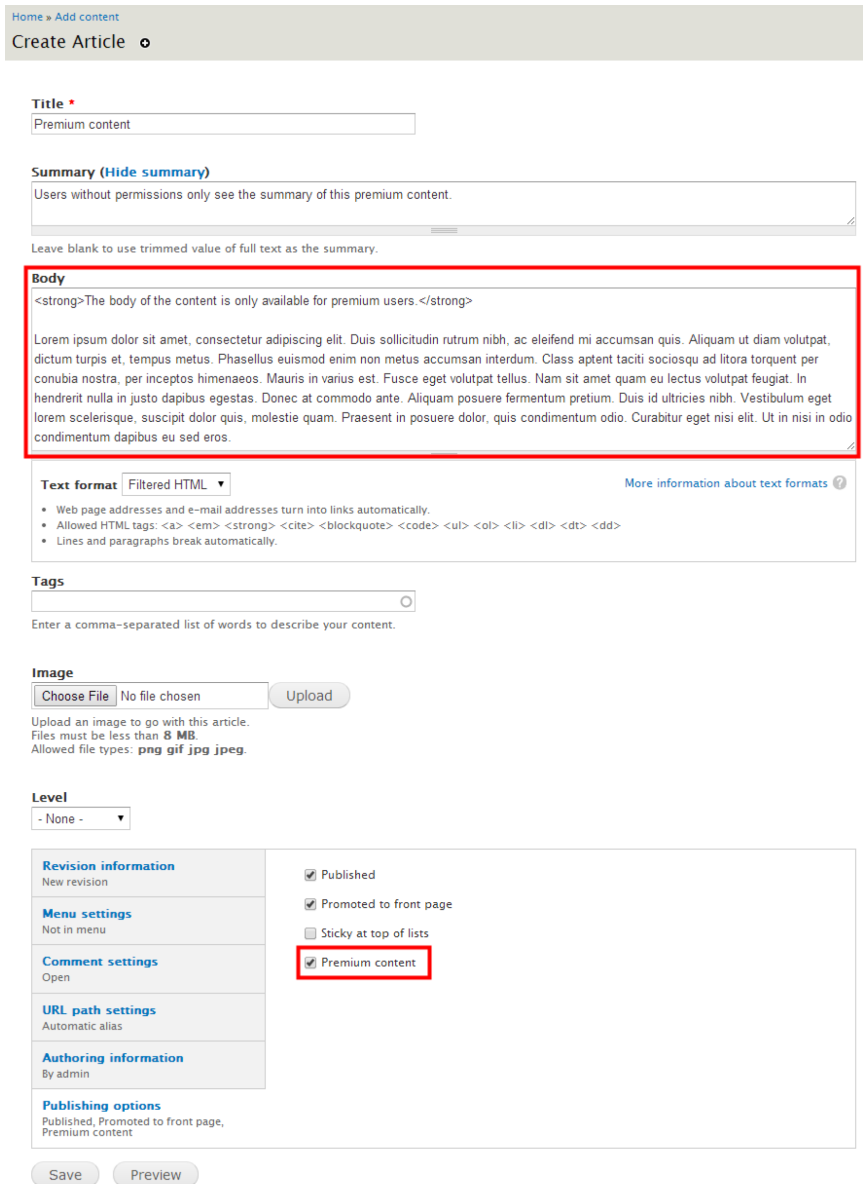
In Publishing Options we can set which content is Premium. Users with appropriate permissions can modify this value.



F34.22

Editing Content

When editing content, users with the approriate permissions may indicate whether the content is Premium or not.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Users without Premium access will be shown the information message set for the content type. F34.23

Premium content

Users without permissions only see the summary of this premium content.

The full content of this page is available to premium users only.

F34.23
Non-Premium users
This message is displayed when Non-Premium members attempt to access Premium content.

Premium users are able to access the entire content. F34.24

Premium content

The body of the content is only available for premium users.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis sollicitudin rutrum nibh, ac eleifend mi accumsan quis. Aliquam ut diam volutpat, dictum turpis et, tempus metus. Phasellus euismod enim non metus accumsan interdum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Mauris in varius est. Fusce eget volutpat tellus. Nam sit amet quam eu lectus volutpat feugiat. In hendrerit nulla in justo dapibus egestas. Donec at commodo ante. Aliquam posuere fermentum pretium. Duis id ultricies nibh. Vestibulum eget lorem scelerisque, suscipit dolor quis, molestie quam. Praesent in posuere dolor, quis condimentum odio. Curabitur eget nisi elit. Ut in nisi in odio condimentum dapibus eu sed eros.

F34.24
Premium Users
Premium users are able to access the entire content.

Permissions

For each type of content on the site, two new permissions are enabled: F34.25

- **View full premium content**. This permission must be assigned to Premium users.
- **Override premium content**. Users with this role, if they have edit permissions for the node, can also mark the content as Premium or not Premium.

PERMISSION	PREMIUM
Node Option Premium	
Forum topic: View full premium content	☐
Forum topic: Override premium content	☐
Panel: View full premium content	☐
Panel: Override premium content	☐
Article: View full premium content	☑
Article: Override premium content	☐

F34.25
Permissions
Permissions available to configure access to each content type. You only need to set values for the types of content that can be set as Premium

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Please note that in addition to users with permission to view Premium content, the following users can also see the full content:

- Users with the Admininister Nodes permission.
- The author of the node.
- Users that can edit content of the same type.

Integration with Actions and Triggers or Rules

Whether you use the triggers and actions through the Triggers module or if you use the Rules module, there are additional triggers and actions related to Premium content.

Utilizing **Triggers,** we can use the following actions:

- **Actions**:
  - Make content premium.
  - Make content non-premium.

Using **Rules,** we can use the following elements:

- **Conditions**:
  - *Node is premium*.

- **Actions**:
  - Make content premium.
  - Make content non-premium.

35 Multilingual Sites

In the Beginner level, we saw how to add multiple languages to the site through the **Locale** module, and how **Locale** works to translate the site interface. At this level we will take it a step further and build multilingual sites, where not only will the interface be translated but the content and other site elements (menus, blocks, nodes, etc.) will be translated as well.

The fundamental module to implement a multilingual site is called **Internationalization**, also known as **i18n**. As we shall see there are many complementary modules to it but in this unit we will only show some of the highlights of those modules.

In addition to the modules contained in the main module **i18n**, we will look at the following additional modules:

- **Internationalization views**, which permits the translation of views.
- **Language icons**, which adds icons with flags associated to each language.
- **Translation Overview**, which shows on one screen a summary of the translation status of each node.
- **Translation table**, which provides an interface to easily translate various elements of the site (taxonomy, content types, fields, menus, etc.).
- **Localization client**, which provides an interface to fix translation issues on the page itself.

Comparative D7/D6

Translation

Drupal 7 incorporates important differences in relation to Drupal 6, for this reason a complete review of this unit and the modules and functions explained is advised.

One notable difference is found in how Drupal 7 utilizes multilingual variables. These are now managed in the admin area and not directly in the settings.php file as they were in Drupal 6.

A new concept, translations sets, was also introduced which apply to various ítems such as menu links and taxonomy terms.

Automatic Content Translation

In Drupal 6 we saw it was possible to automatically translate published nodes using the Google Translate API. Due to changes in the Google Translate use policy, the module has been depreciated and a new one has not yet been developed for Drupal 7.

Unit contents

35.1 Language Configuration....................434
35.2 Internationalization Module (i18n)....................438
35.3 Multilingual Variables....................440
35.4 Content Translation....................442
35.5 Menu Translation....................448
35.6 Block Translation....................454
35.7 Content Type and Field Translation....................455
35.8 Taxonomy Translation....................457
35.9 Views Translation....................459
35.10 Additional Modules....................460

35

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

35.1 Language Configuration

In unit 18 of the Beginner level we studied how to add additional languages to the site, and how to import interface translations (Drupal and installed modules) from localize.drupal.org, both manually and automatically, using the **Localization update** module.

In this unit, we continue the multilingual site configuration, not just for the interface but also for content and other site elements such as menus, blocks, taxonomy, etc.

It is important to note that although the site interface can be automatically translated thanks to the contributions of the Drupal community, the content and elements do not have the same support. The multilingual system we are building will require the pre-translation of all the elements and content of the site.

Language Detection

When you create a site with multiple languages, one of the first decisions you have to make is how to identify each language. There are typically two configurations for each language though which this is done: path prefixes and domains (or subdomains).

Depending on how the **path prefixes are configured**, we can establish a code for each language that will be shown as a path before the destination page:

- www.example.com/en/page_en
- www.example.com/es/page_es

Generally, the default language will not need a path prefix. For example, if we have selected Spanish as the default language the path could be as follows:

- www.example.com/page_es

For **configuring domains or subdomains** we can set each language in the following way:

- en.example.com/page_en
- es.example.com/page_es

For the default language we can simply use the primary domain:

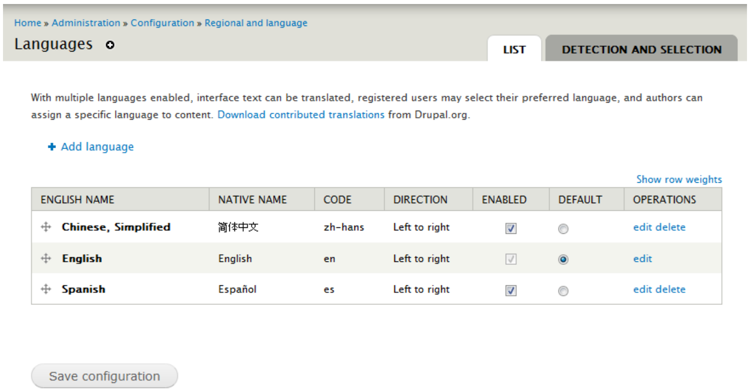
- www.example.com/page_es

Adding and configuring a language

Recall that in order to have the functionality to add languages and translate the interface it is necessary to enable the **Locale** module, included in the Drupal core. This module is enabled automatically if we installed an additional language (e.g., Spanish) during the Drupal installation. If it has not been activated, please do so before continuing. To add an additional language we will do it from the **Add language** option in: F35.1

URL Add Language
/admin/config/regional/language/add

Administration ⇨ Configuration ⇨ Regional and language ⇨ Languages [Add language]



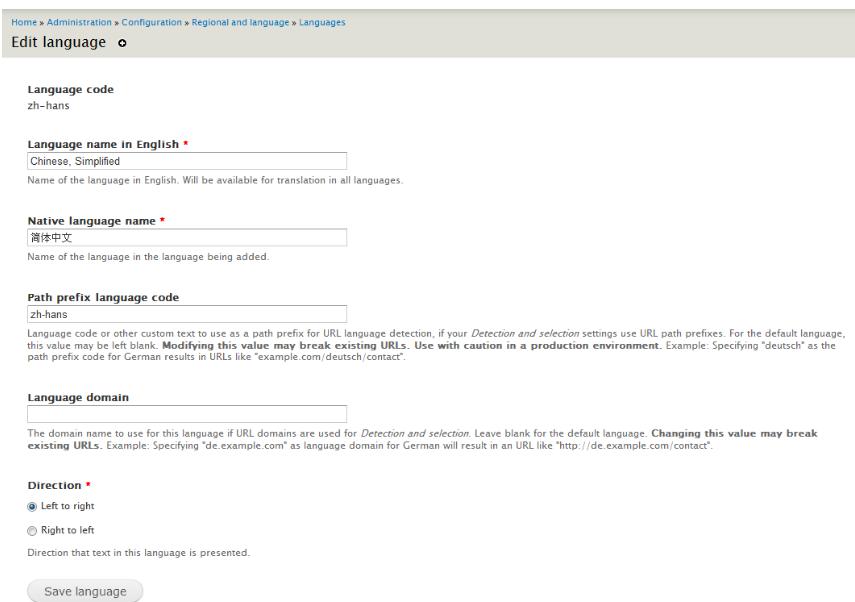
F35.1

Add Language

We can add one of the languages listed or even create a custom language. This action only adds the language, not the translation strings.

After selecting the language, we will access its settings. Depending on the method of language detection we will use, path prefix or domain, we have to set one of two values. Each language has a default **Path prefix language code** that we can change.

From the language settings we can change other parameters such as the **language name in English**, the **native language name** (name of the language in that language) and the direction in which is it written (from left to right or vice versa). F35.2



F35.2

Edit Language

Among the settings for most basic languages is the code path prefix or domain to be used to detect the language depending on the selected method of detection.

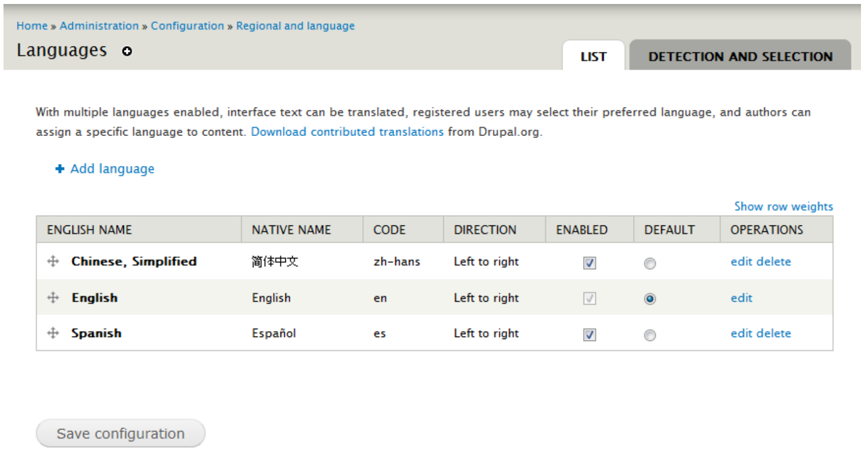
The added language will be shown in the main list of languages, where we can activate, edit and set one of them as default. F35.3

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F35.3

List of Languages

From this list of languages we can indicate which ones we want active and which is the default.



Detection and Language Selection

Finally, we have to configure how the language is detected when loading any page on the site. We do this with the **Detection and Selection** tab: **F35.4**

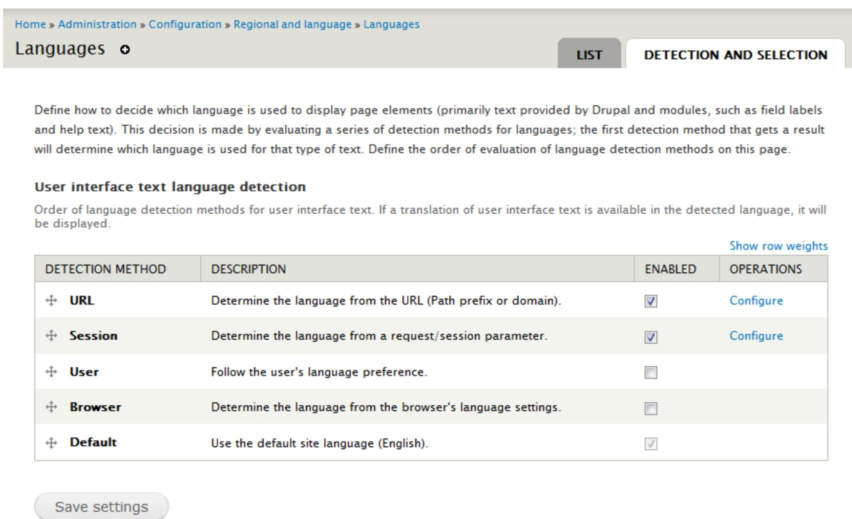
URL Detection and selection
/admin/config/regional/
language/configure

Administration ⇨ Configuration ⇨ Regional and language ⇨ Languages [Detection and selection]

F35.4

Detection and selection

We set the order of the language detection methods. Once one of the defined rules is met, the system will stop evaluating the other methods.



There are five language detection methods that can be used in conjuction with each other. Each method will be evaluated according to the established order; once one of them is able to determine the current language of the page, the system will not evaluate the others. Only detection methods that are activated (**Enabled** column) will be evaluated.

The detection methods available are:

- **URL**. Determines the language from the URL, either through a path prefix or domain language. **F35.5**

F35.5

Configuring URL language detection

Through the URL the system can determine the language by a path prefix or a specific domain for the language.

- **Session**. Determines the language through a parameter passed by the URL or stored in the session. This method is used by the Language Switcher block, which allows the user to select the language in which they wish to view the page.

- **User**. Determines the language based on the language set by the user on the account.

- **Browser**. The language shown depends on the language of the user's browser.

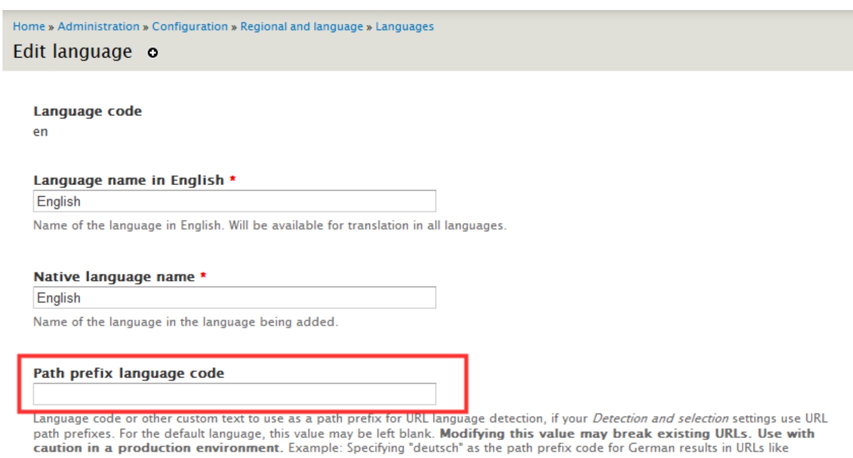
- **Default**. Uses the default language of the site.

When we use the **path prefix** or **domain** detection methods, we should leave blank the value of "Path prefix language code". This will ensure a prefix is not added to the URL. F35.6

- www.example.com/about-us (*instead of* ~~www.example.com/en/about-us)~~

or

- www.example.com/about-us (*instead of* ~~en.example.com/about-us)~~



F35.6

Configuring the default language

In the default language, leave the Path prefix Language code field empty if you are using **path prefix** or **domain** language detection. This will ensure a prefix is not added to the URL.

35.2 Internationalization Module (i18n)

The **Internationalization** module (also known as **i18n**) complements the core Drupal module **Local**, adding additional translation capabilities.

The **Internationalization** module is available at:

http://drupal.org/project/i18n

The module depends on the **Variable** module, available at:

http://drupal.org/project/variable

This module consists of a set of modules grouped under the **Multilingual – Internationalization** area of the module admin page. In this unit, we will not use all of these modules but we will highlight the following:

- **Internationalization**. This is the primary module for activating multiple languages and translation.
- **Block languages**. Adds the option of multiple languages and translation for blocks.
- **Field translation**. Allows you to translate the properties or labels of the fields for entities.
- **Menu translation**. Allows you to translate the elements of set menus on the site.
- **Multilingual content**. In connection with the core module **Content translation**, allows the translation of site content.
- **Path translation**. Allows the creation of path translations.
- **Taxonomy translation**. Allows the translation of vocabularies and taxonomy terms for the site.
- **Variable translation**. With this module we are able to translate any variable on the site. It is the substitute for Multilingual variables in Drupal 6. We will study it in section **35.3**.

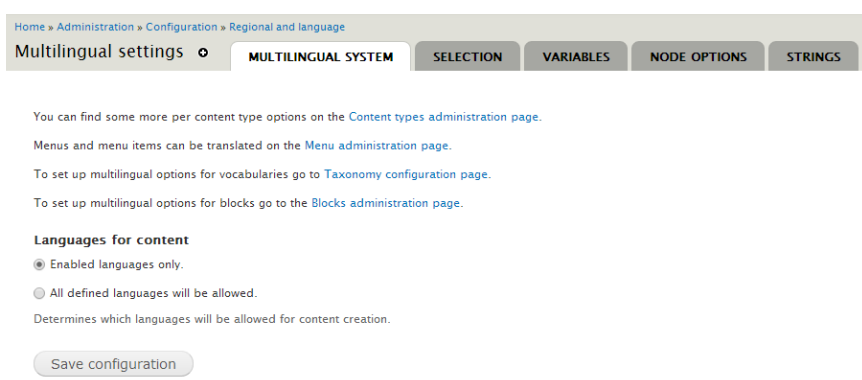
Activate the modules indicated (and their corresponding dependencies).

The overall system configuration is made from Multilingual settings as shown F35.7 below:

URL Multilingual Settings
/admin/config/regional/i18n

Administration ⇨ Configuration ⇨ Regional and language ⇨ Multilingual settings

F35.7
Multilingual System
The Internationalization and related modules add many features for building multilingual sites.



From the **Multilingual settings** we can set the following parameters:

- **Multilingual system**. From this tab you can access settings for managing translation of content types, menus, taxonomy and blocks. We can also set whether content translation is available for all languages on the site or only for the active languages (recommended).

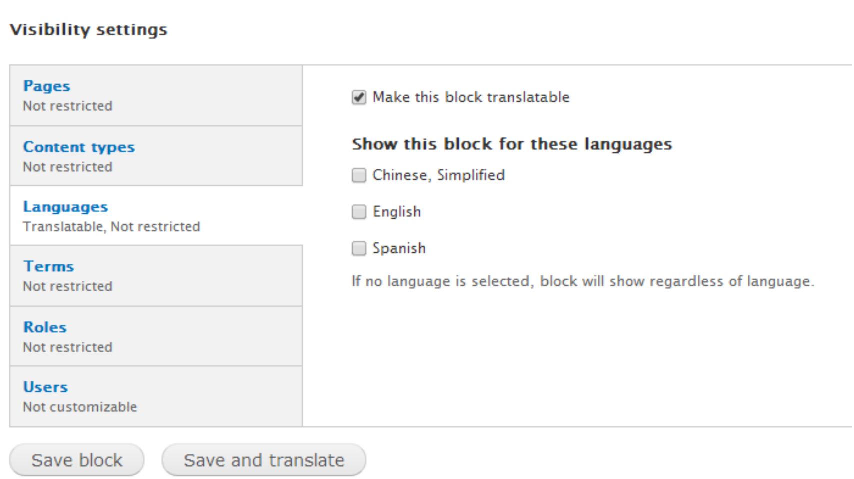
- **Variables**. Allows for the translation of variables. This will be studied in **section 35.3**.

- **Node Options**. Settings related to content. This will be studied in **section 35.4**.

- **Strings**. Configuration options related to text entered by the user. **This will be studied in section 35.4**.

By activating the **Internationalization** module and the related modules, various administration areas (blogs, menus, content types, content creation, etc.) language selection options and language settings are now displayed.

For example, **Figure** F35.8 shows editing a block.



F35.8
Multilingual System
In different adminstration areas, such as editing a block, the settings related to languages and translation are displayed.

In the following sections, we will explore how to translate the various elements available in the system.

35.3 Mulitlingual Variables

Some texts and parameters are stored in Drupal as **variables** in the **variable** database. These variables are usually configred from various sections of the administration area. For example, the site name, slogan and mission, home page, etc., are site variables.

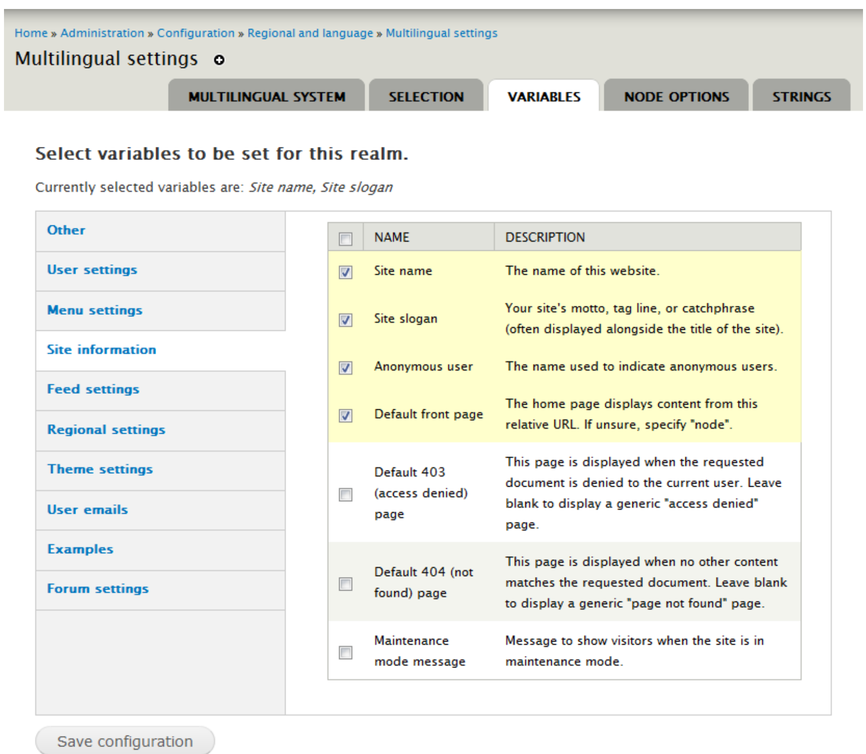
In **Drupal 6**, the translation of variables is accomplished through defining multilingual variables (or multiple languages) in the site configuration **settings.php** file. In Drupal 7 this option is no longer available, therefore it is necessary to use the **Variable translation** module included in the **Internationalization** module package.

Once the module is enabled, we can access the translation of variables from the following area: F35.9

URL Multilingual Variables
/admin/config/regional/i18n/variable

Administration ⇨ Configuration ⇨ Regional and language ⇨ Multilingual settings [Variables tab]

F35.9
Multilingual Variables
The selected variables are available for translation into the languages of the site.



The list displays the set of global variables that can be translated. Note that the content of the variable is what is translated and not the title.

Select from the list the variables you wish to translate and save the configuration. These variables will now be considered multilingual or multi language variables.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Multilingual variable Translation

In order to translate multilingual variables we have to go to the configuration page for each of the variables. For example, to change the **site name**, **slogan, or default home page,** we can go to the following locaiton: F35.10

Administration ⇨ Configuration ⇨ System ⇨ Site information

URL Site Information
/admin/config/system/site-information

In forms where multilingual variables exist an additional block is displayed which allows us to change the language (**There are multilingual variables in this form**).

Aditionally, multilingual variables are distinguished from the other variables with additional text informing the user (**This is a multilingual variable**).

To set the translation of each variable, select the desired language and edit the field value for that language. Save the changes before moving to another language. We will perform the same operation for all multilingual variables in each of the active languages of the site.

F35.10

Multilingual Variables

When a form contains multilingual variables a global message, as well as informational text on each variable, are displayed. To translate a multilingual variable, we only have to change the language using the links provided, enter the translated value and save the form. Repeat this for each of the languages.

The variables are now displayed for each language when the language is selected for that page. F35.11

F35.11

Multilingual variable example

Through multilingual variables we have translated the title or name of the website.

35.4 Content Translation

Pre-configuration

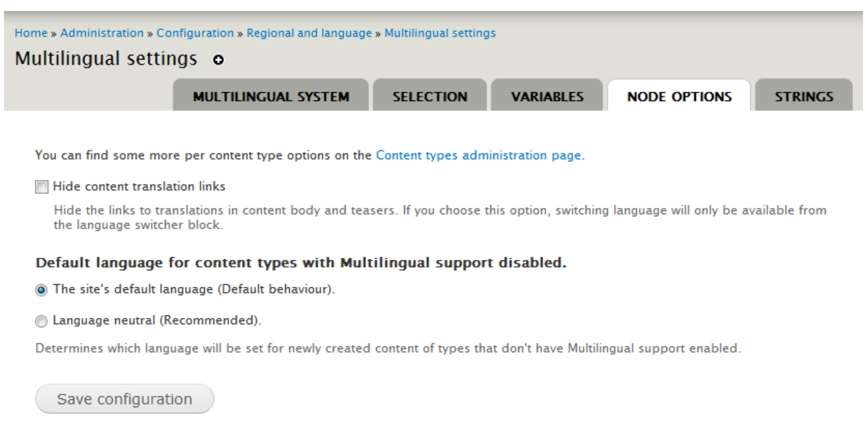
For translation of content to be available, you must set the following parameters in the **Multilingual settings administration** page:

URL Multilingual Settings
/admin/config/regional/i18n

Administration ⇨ Configuration ⇨ Regional and language ⇨ Multilingual settings

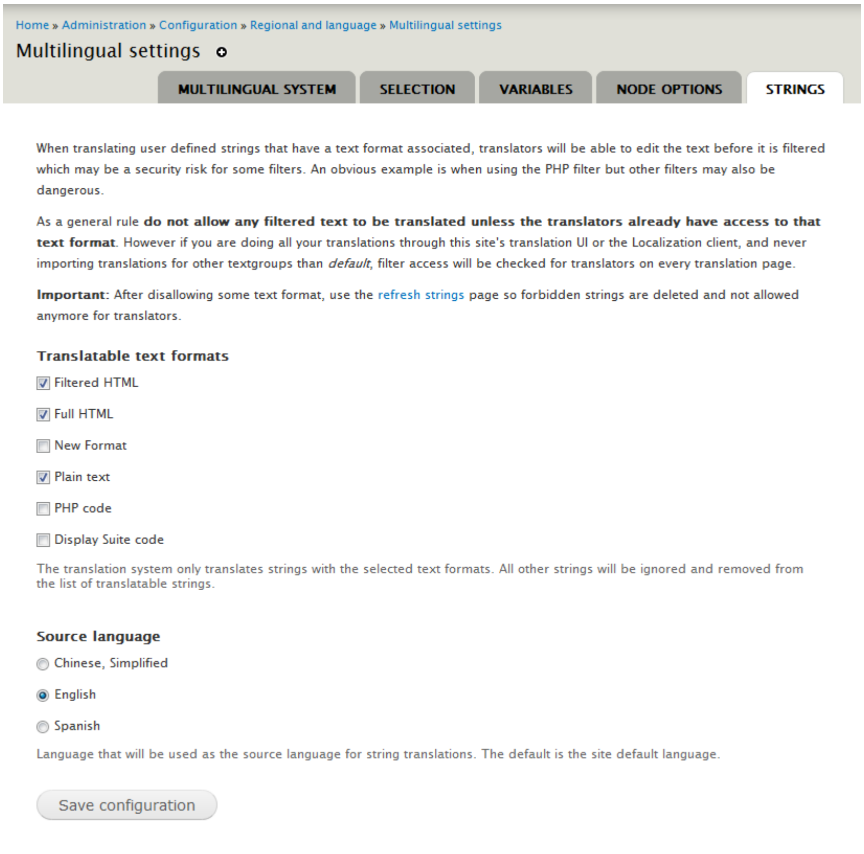
- **Node options**. Within this tab we are able to configure the following options: F35.12
  - **Hide content translation links**. Users with translation permissions have links to the translation of content in the body or the summary of a node when this option is not active.
  - **Switch interface for translating**. By default the interface of the site will be in a different language than the node being edited. Activating this option will change the interface language to match that of the node being edited.
  - **Default language for content types with Multilingual support disabled**. Indicates which language will be used for new content when when support multilingual support has not been enabled for this content type. By default, this sets new content to the default language of the site, but you can choose to mark it as "language neutral".

F35.12
Node options
These are the primary node options available related to content translation.



- **Strings**. Allows for the configuration of user defined strings that have a certain **text format** for presentation. Users with translation permissions can access the original text and edit it before it is filtered so it is recommented to restrict access to certain text formats. F35.13
  - **Translatable text formats**. It is recommended to limit access to those text formats that do not include PHP code.

- **Source language**. Indicates the language used as the basis for content translation. The translator will display the contents of the element in this language before proceeding to translate it to the corresponding language.



Home » Administration » Configuration » Regional and language » Multilingual settings

Multilingual settings

MULTILINGUAL SYSTEM | SELECTION | VARIABLES | NODE OPTIONS | STRINGS

When translating user defined strings that have a text format associated, translators will be able to edit the text before it is filtered which may be a security risk for some filters. An obvious example is when using the PHP filter but other filters may also be dangerous.

As a general rule **do not allow any filtered text to be translated unless the translators already have access to that text format**. However if you are doing all your translations through this site's translation UI or the Localization client, and never importing translations for other textgroups than *default*, filter access will be checked for translators on every translation page.

Important: After disallowing some text format, use the refresh strings page so forbidden strings are deleted and not allowed anymore for translators.

Translatable text formats

- [x] Filtered HTML
- [x] Full HTML
- [ ] New Format
- [x] Plain text
- [ ] PHP code
- [ ] Display Suite code

The translation system only translates strings with the selected text formats. All other strings will be ignored and removed from the list of translatable strings.

Source language

- ( ) Chinese, Simplified
- (•) English
- ( ) Spanish

Language that will be used as the source language for string translations. The default is the site default language.

Save configuration

F35.13

Strings

From the Strings tab we can indicate which text formats can be translated and which language will be used as the basis for other translations.

Users with translation permissions can only translate the fields of text with the selected active formats, avoiding, in this example, access to fields containing PHP.

Configuration of content types

The next step is to configure the content types to make use of the multilingual system. This allows translation of created nodes.

We can access editing the content type in the corresponding editor:

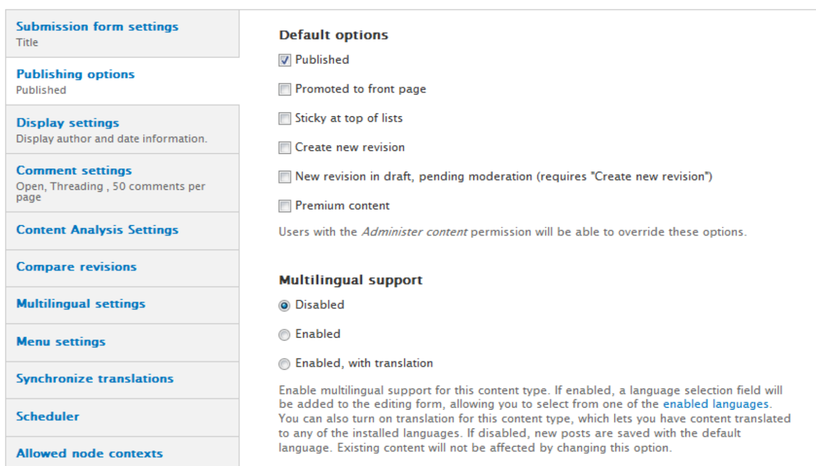
URL Content Type
/admin/structure/types

Administration ⇨ Structure ⇨ Content types

To activate the translation system for each type of content, access the **Publishing options** for that type: F35.14

- **Multilingual support**. The options are:
    - **Disabled**. By default this is set for all content types.
    - **Enabled**. Activating this allows the user to select a language when creating a node.
    - **Enabled with translation**. This the most complete option, which allows not only node language selection but also enables translation. Select this option.

F35.14
Publishing options
For a given content type to be translatable, we have to configure it from the Multilingual support section under Publishing Options. The option **enabled, with translation** allows nodes to be created with an associated language and to be translated to other languages of the site.

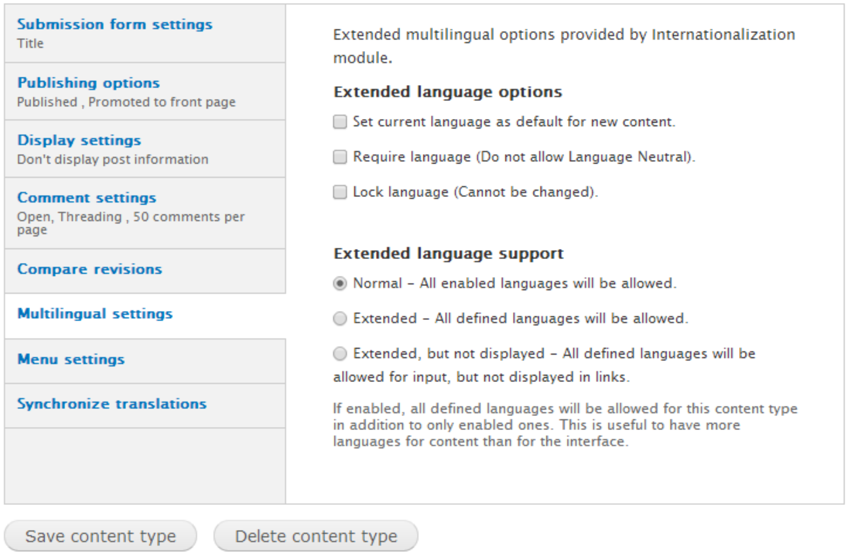


Once you have activated the multilingual system, you can configure the **Multilingual settings**: F35.15

- **Extended language options**. The options available, all of which are optional:
    - Set current language as default for new content.
    - Require language (Do not allow Language Neutral). Cannot leave the node "without specific language"; makes selecting one of the available languages mandatory.
    - Lock language. The language cannot be changed.

- **Extended language support**. Determines if translations will be made only on the site's active languages or on all defined site languages, even if they are not active.

After configuring the content type, save the changes.



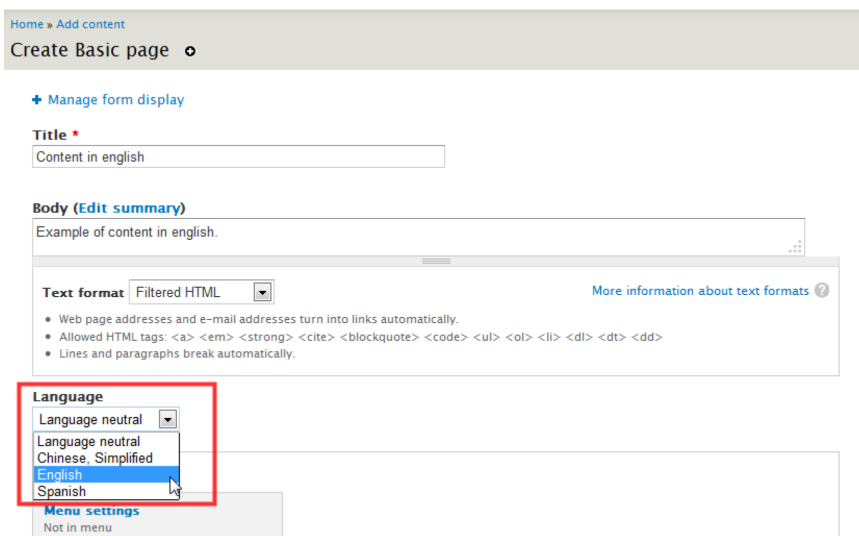
F35.15

Multilingual Options

Once we have enabled multilingual functionality, we can proceed with content type configuration from the Multilingual Settings group.

Translate a node

The process of content translation begins with the creation of the node in one of the languages available. **F35.16**



F35.16

Node with Language

When creating a node of a translatable content type, select the desired language.

On the created page we discover two new elements: **F35.17**

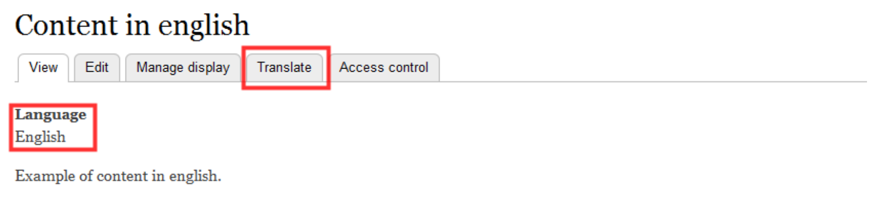
- **Together with the content of the page the language is displayed**. This is because a new field, language, has been added to the content type, which allows the selection of the language. If you do not want to display this field, simply go to the **manage display** tab of the content type editor, which controls content type field display
- **Translate tab**. The tab gives access to the content translation tools.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F35.17

Translate Node

Once we have created a node, we are able to translate it through the translate tab.



From the **Translate** tab we can: F35.18

- **Add a translation for any language available**. This option will create a new node linked to the existing one.

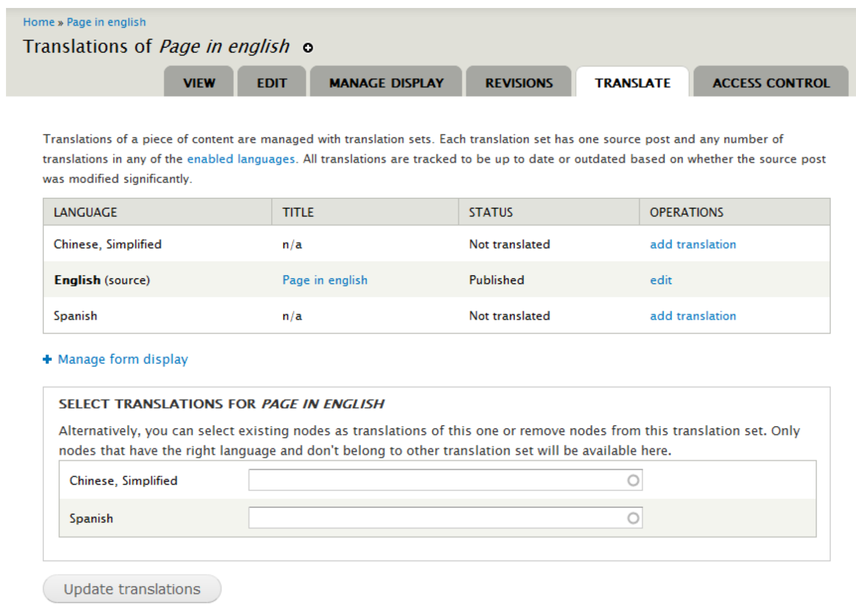
- **Select an existing node**. If the node with the translation has already been created, we can link to it.

In both cases, we have multiple nodes linked in different languages, so that when we change the language, the corresponding node will display in the selected language.

F35.18

Translate Node

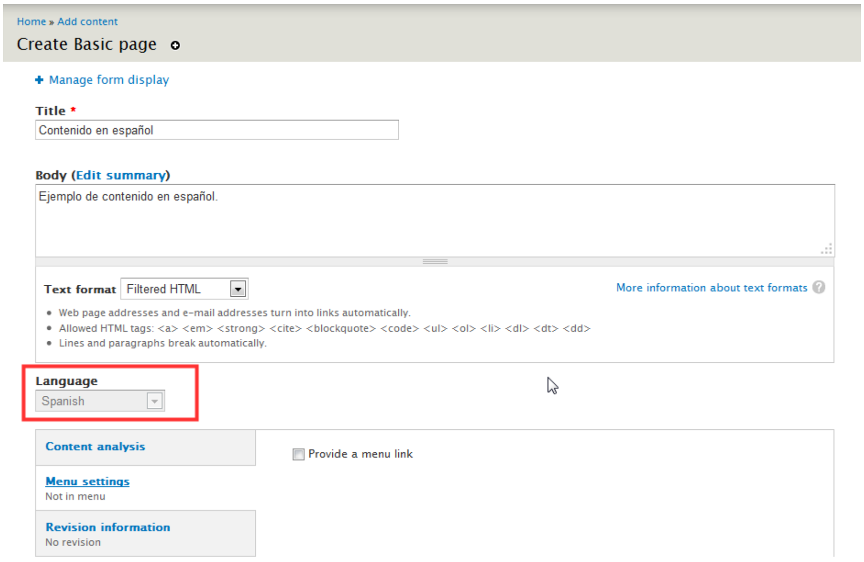
Translations of node from one language to another are actually individual nodes linked via the translation system. From the Translator we are able to create new translations or if we have previously created content in other languages we can select and assign those nodes for each language.



By clicking on **add translation**, we can add a translation of the content in that language. Actually what the system does is create a new node, which we can managed independently of the original node. Internally, the nodes will have a link that demonstrates they are the same content but in a different language.

As we have stated, we are not creating a machine translation system, but a multilingual site, where you can create content in multiple languages. The new node is initially displayed with the content in the language we are using as a basis for translation. We will have to edit the fields of the node (title, body and other added fields) and translate all the content to the selected language. In this case the language will appear locked.

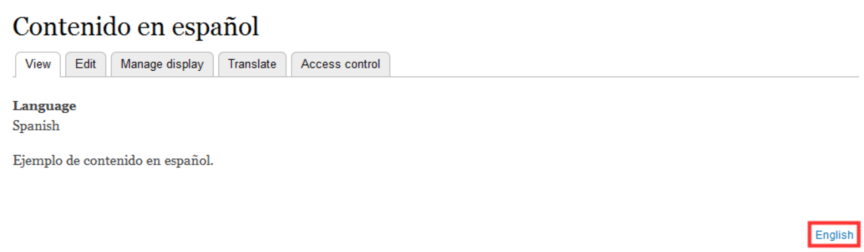
Depnding on whether or not we have enabled the option to **Switch interface for translation,** the interface of the site will change to the F35.19 language being edited.



F35.19

Translate Node
When translating a node to a language, the language is shown blocked.

Once you save the node translation, it will show the node in the language added. In the Node display there will be a link to the node translations available for that node. **F35.20**



F35.20

Node Translation
Translated nodes are shown with links to other languages.

We can also test the functionality of changing the language using the selector block. **F35.21**

F35.21

Language Selector
The language selector block can be used to test the opertation of translations.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

35.5 Menu Translation

Translation of menus with multilingual variables

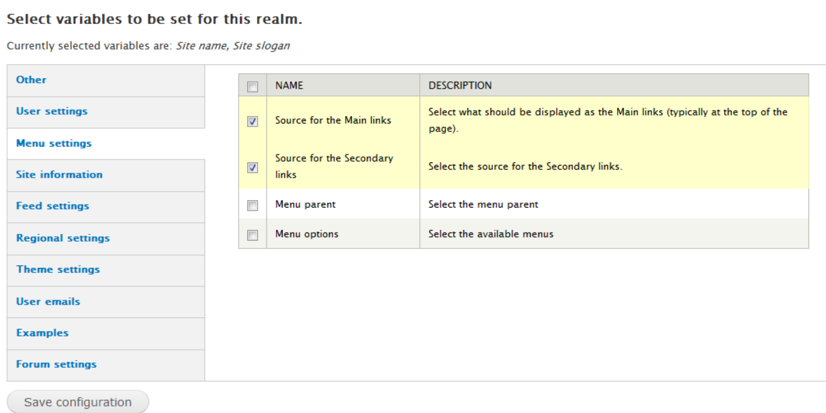
There are several ways to translate site menus. For example, we can translate the main menu links and/or secondary menu links by using multilingual variables as we saw in **section 35.3**.

The variables **Source for the Main links** and **Source for the secondary links** allow us to assign different menus for each of the languages. F35.22

F35.22

Multilanguage Menu Variables

The Source for Main Links and Source for Secondary Links variables allow us to select different menus for each language but it is limited to these two menu types.



Menu Translation Modes

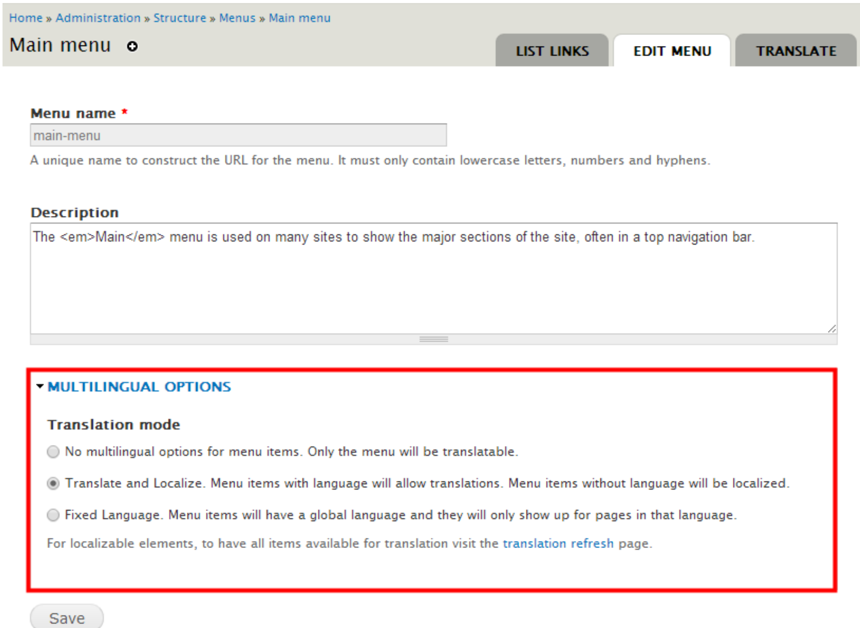
From the menu editor you can determine how each menu will function in a multilingual environment. The available options are:

- **No multilingual options for menu items**. We are only able to translate the name and description of the menu, but not its items.
- **Translate and Localize**. Allows the adding of menu items with translation. If the item has no translation, it will be displayed in its appropriate language.
- **Fixed language**. The meun will have a global fixed language and will only show in that language. When you select this option, you will need to indicate which languge the menu will be displayed in.

These options determine the possible strategies when translating menus. There are basically two we will consider:

- **Unique menu with menu items in different languages**. We will create a unique menu with translated items or even separate items for each language. The advantage of this option is we can manage all the menu items from a single menu. The disadvantage is that the menu items are mixed and this can complicate menu item management.
- **Separate menus for each lanaguge**. Following this method will create a menu for each language, each with their corresponding links. The main advantage of this method is the menu items will be separated and organized by language.

The method used is up to the developer since both options can achieve the same results. F35.23



F35.23
Multilingual Menu Options
There are two menu translation strategies: Unique menu items displayed depending on the language, seperate menus for each language with their own links.

Menu Item Translation

Using the **Translate and Localize** mode of translation, we will create a unique menu that will have:

- **Menu items in a single language**. These items will only be displayed in the corresponding language.
- **Translated menu items**. These items will be displayed for all languages that have translations. In addition to translating the titles of the links, we can also set different URLs for each language.

When creating a menu item in a particular language, we can associate it with other items in other languages, or we can create new items at that time. In order to do so use the **Translate** tab to edit the item. F35.24

F35.24
Translating Menu Items
From the Translate tab of a menu ítem, we can associate languages with different menu items.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Translation Sets

Drupal 7 incorporates the concept of **Translation sets**. A translation set is a group of menu items in different languages that are linked together.

Translation sets can be of different types, depending on the items being linked: menu items, paths, taxonomy terms, etc. When the menu items are linked, the system shifts between them depending on the language of the page.

URL Translation Sets
/admin/config/regional/
i18n_translation

Translation sets can be managed globally from:

Administration ⇨ Configuration ⇨ Regional and language ⇨ Translation sets

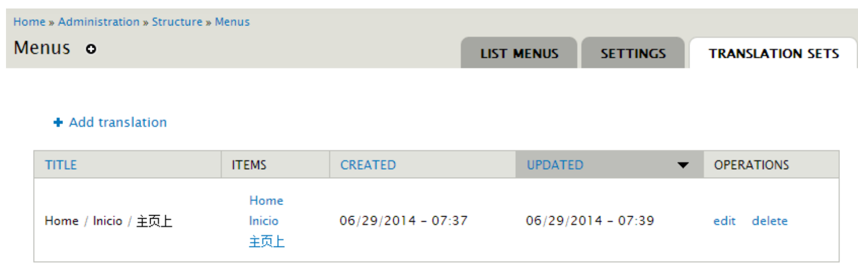
While the creation of translation sets for menu items can be accessed at:

URL Translation Sets for menu
/admin/structure/menu/
manage/translation

Administration ⇨ Structure ⇨ Menus [Translation sets]

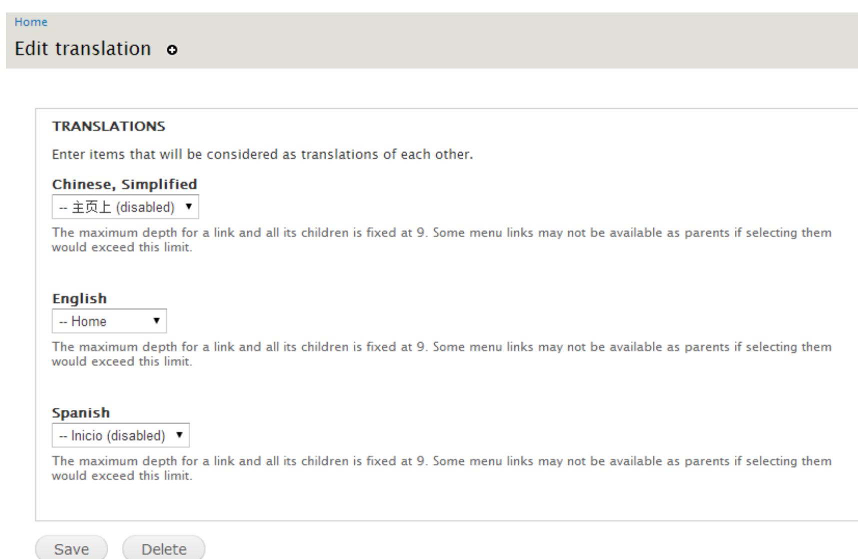
In the translation sets tabs all menu times that have been linked (or translated) are displayed. F35.25

F35.25
Translation Sets
The translation sets for elements for the various lanaguages are linked. In this case we are applying this to menu items.

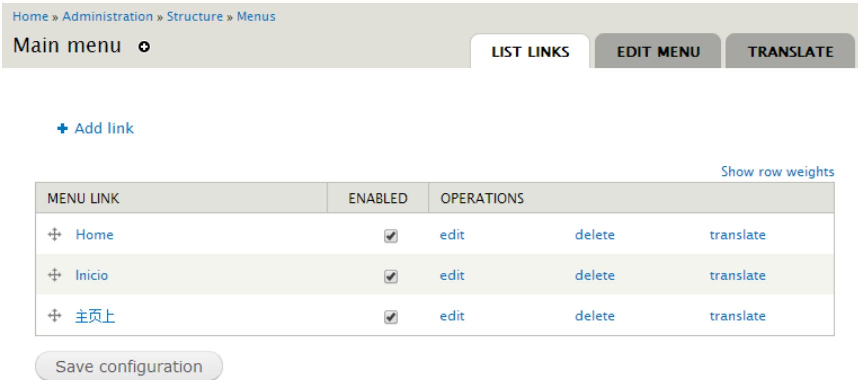


Adding a new translation set is very simple. We just select the corresponding item for each of the languages. F35.26

F35.26
Translation Sets
In the Translation set each language has an association to a previously created menu item.



The **Figure** F35.27 shows the **Main Menu** with three menu links that link the homepage of the site, but really only one of them will show at a time depending on the site's active language.



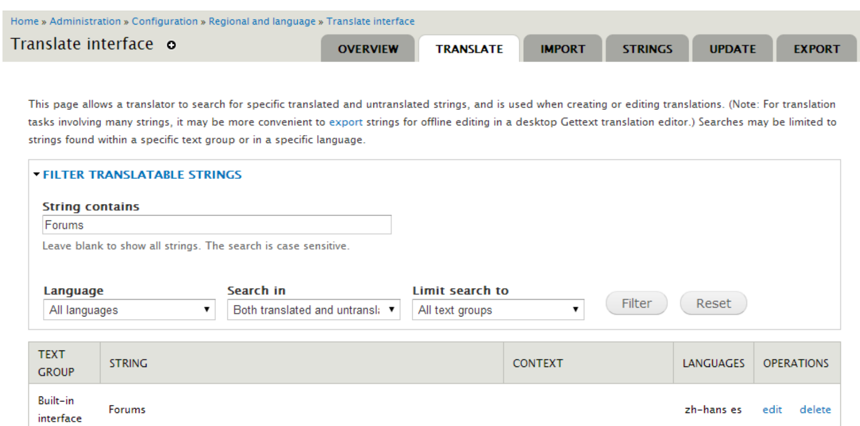
F35.27
Translated Links
The menu shows all links, but on the page only the link to the active language will be shown.

Translation of common menu items

Sometimes we need to translate common menu items to all the languages.

For example, if you have installed the Forum module, we can access it from the URL /forum.

In this case we want to create a single menu link pointing to the forums, but whose title is translated to all the site languages. **Forums** will create a link to the forum, **without specifying a language**. To translate the title link we'll use the **Translate interface**. F35.28



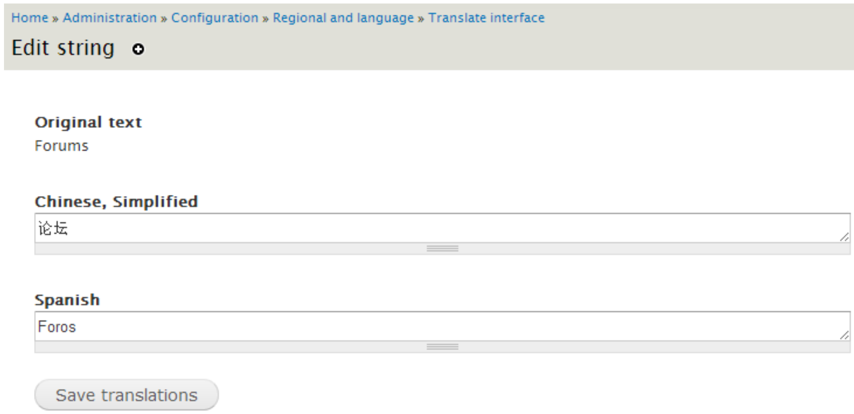
F35.28
Common Menu Items
We can also translate common menu items with the same URL for all languages. In this case the ítem is avaialble in the Translation interface

Once you locate the menu item, you can translate the title into the other languages. Note that the base language of the item (original text) will match the default language of the site. F35.29

F35.29

Translation of menu items

Menu items are available for translation from the Translation interface; its function is the translation of the various strings of the site.



To finish the translation of menus, let's look at the **Figure** F35.30 .

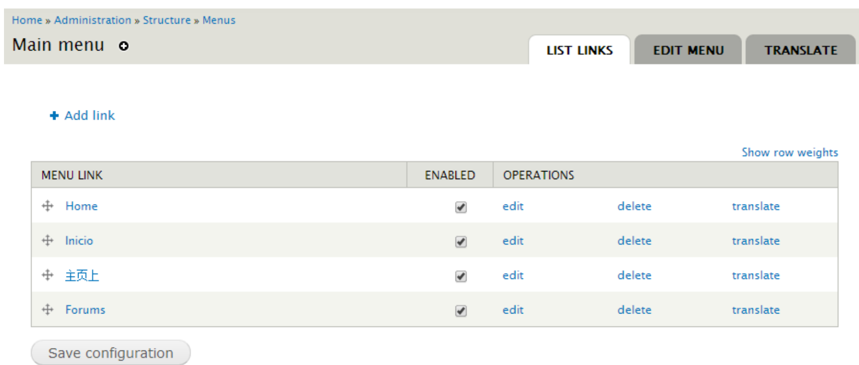
The menu has four distinct items, but only two of the languages are displayed. To define the links we initially created separate links and then related them through translation sets.

The link, Forums, however, is a single link to all languages we have translated through the Translation interface.

F35.30

Menu Items

The menu has four distinct items, but only two are shown for each language.



In **Figure** F35.31 , the resulting menu is displayed according to the selected language.

F35.31

Menus by language

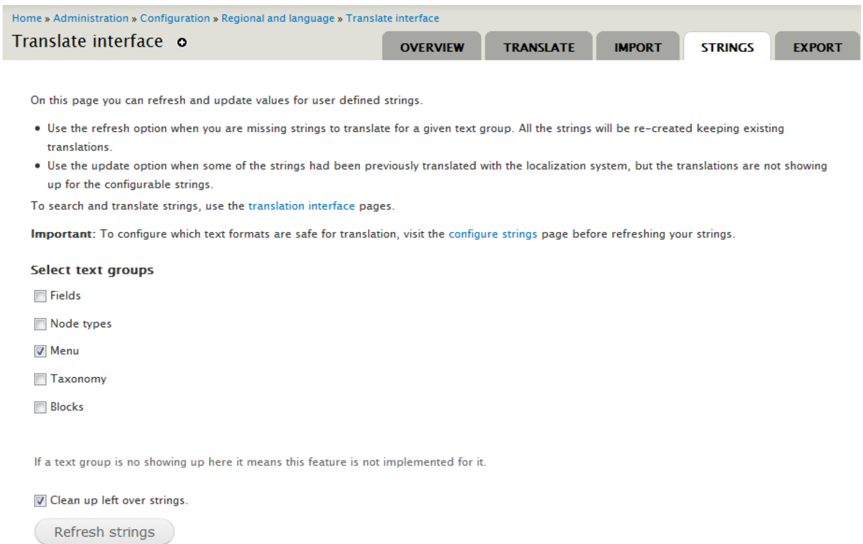
Figure 3 shows the resulting menu columns according to the selected language. The Forums ítem is common to all languages, while the links may point to different URLs.

Updating text groups

If you cannot locate a menu item through the **Transalate interface**, you may need to update the last added text by, selecting the **Text Group for Menu** here: F35.32

Administration ⇨ Configuration ⇨ Regional and language ⇨ Translate interface [Strings tab]

URL Text Groups for translation
/admin/config/regional/translate/i18n_string



F35.32
Text groups for translation
If we cannot find a text string created by users (menu, block, field element), it might be necessary to update the text groups. Updating text will add texts to the translation database that are not currently there.

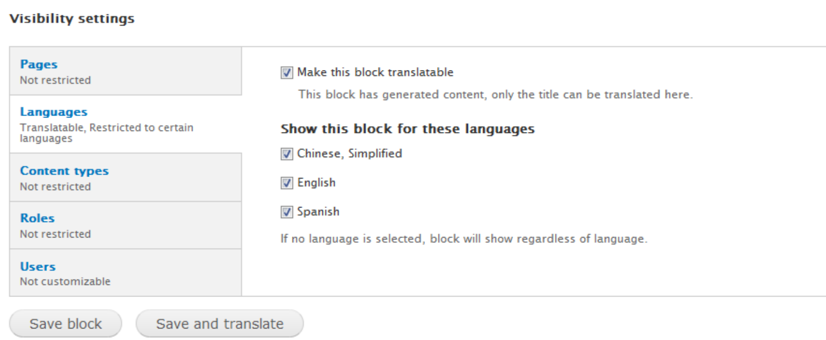
35.6 Block Translation

Blocks also have settings that determine the block's visibility depending on the active language. We can specify whether the block is translatable and what languages it will be displayed in. F35.33

F35.33

Language options for Blocks

The Languages option group permits setting the visibility of the block depending on the active language. We can also make the block translatable.

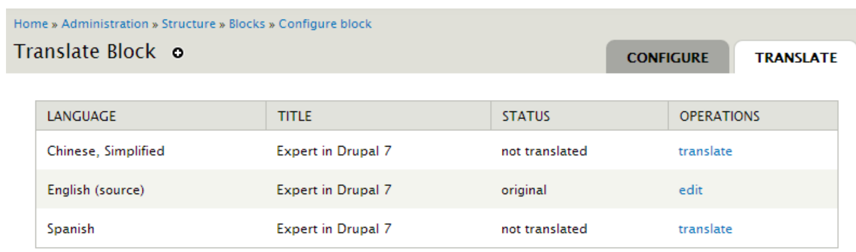


If you idicate that the block is translatable, to acces its display settings you can go to the additional **Translate** tab. F35.34

F35.34

Translation of Blocks

From the Translate tab we can accedd the fields of the block that are translatable.



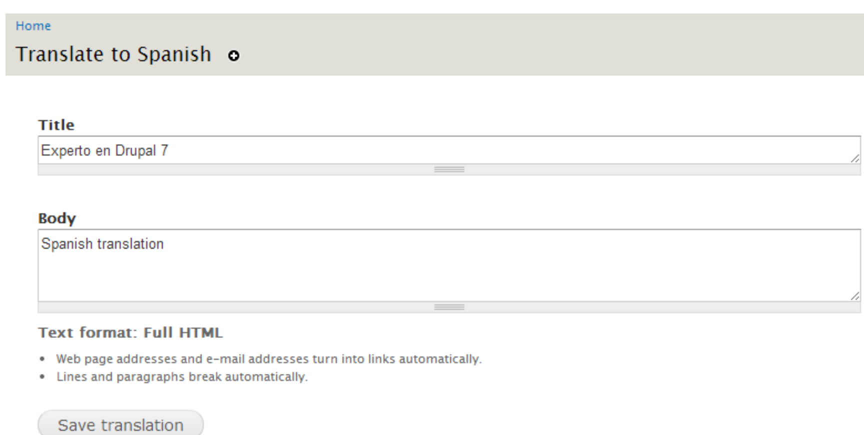
LANGUAGE	TITLE	STATUS	OPERATIONS
Chinese, Simplified	Expert in Drupal 7	not translated	translate
English (source)	Expert in Drupal 7	original	edit
Spanish	Expert in Drupal 7	not translated	translate

By clicking on the **translate** links we can create a translation in each language. Note, if this is a block you have created directly, we can translate the content, but if the block comes from a module, we can only translate the title and not the content. In this case, we translate the text shown in the block through the translation interface by selecting the text group **Blocks** (and updating it as necessary, as we saw for the translation of menus). F35.35

F35.35

Translation of blocks

If it is a custom block, we can translate both the title and the content. For most other blocks added by modules, we can only translate the block title, using the translation interface like the other elements.

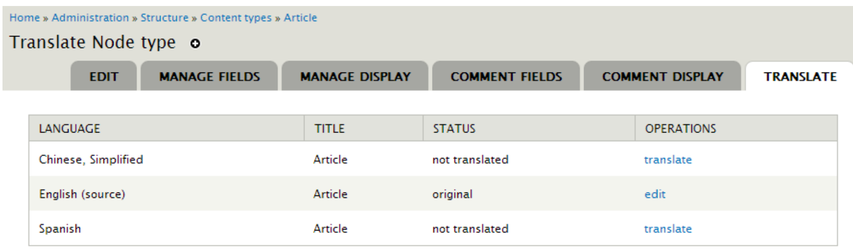


Content type and Field translation

35.7

In **section 35.3,** we saw how we can create nodes in different languages. Depending on the content type, nodes can have additional fields, which are extensible to other entities on the site.

To translate the main labels of a content type (or entity), access the **Translate** tab of the content type. Here you can configure the main elements of the content type (Name, Title tag, Description and Help text) for each language. F35.36


Home » Administration » Structure » Content types » Article

Translate Node type

EDIT | MANAGE FIELDS | MANAGE DISPLAY | COMMENT FIELDS | COMMENT DISPLAY | TRANSLATE

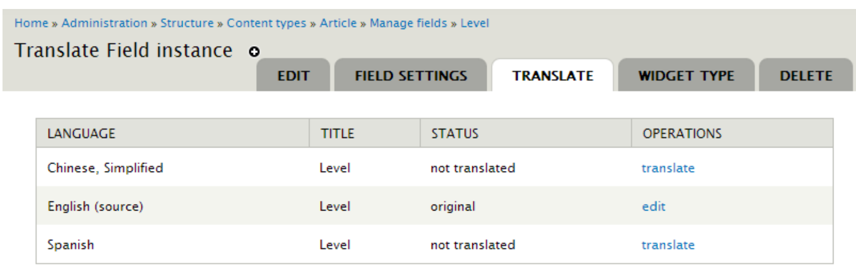
LANGUAGE	TITLE	STATUS	OPERATIONS
Chinese, Simplified	Article	not translated	translate
English (source)	Article	original	edit
Spanish	Article	not translated	translate

F35.36

Translation of content type

From the Translate tab of the content type, we can translate the name, title tag, description and help text.

To translate the values associated with each of the fields for the content type, access the field editor and then go to the **Translate** tab. For each field we can translate the Label, Description and Default Value. F35.37


Home » Administration » Structure » Content types » Article » Manage fields » Level

Translate Field instance

EDIT | FIELD SETTINGS | TRANSLATE | WIDGET TYPE | DELETE

LANGUAGE	TITLE	STATUS	OPERATIONS
Chinese, Simplified	Level	not translated	translate
English (source)	Level	original	edit
Spanish	Level	not translated	translate

F35.37

Translation of Fields

For each field, we can translate the label and description and even specify a default value depending on the language.

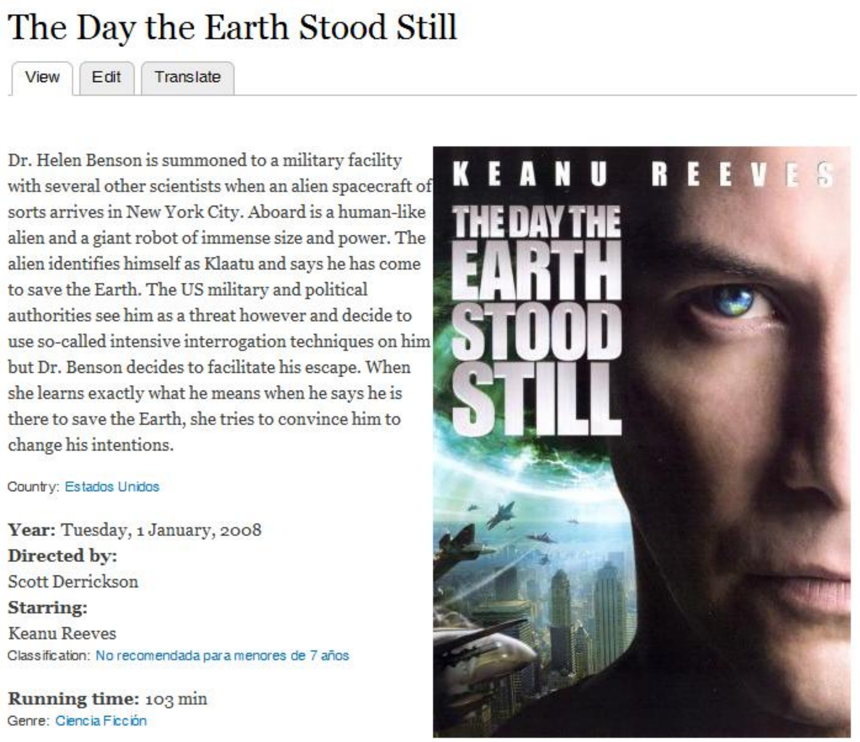
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Figure F35.38 demonstrates a content type, Movie, whose fields have been translated.

F35.38

Content Translation

The figure shows an example of a Movie content type that has been translated. Using field translation we have been able to translate the field labels shown (Year, Directed By, Actors, Rating, Runtime, Genre, etc.)



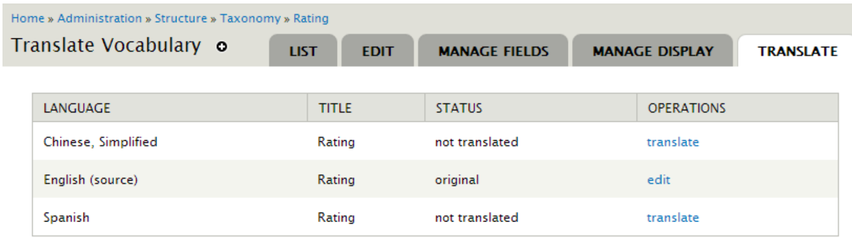
Note that with this procedure we are only translating the labels associated with the fields, and not their content. When the content of a field is a text type, we can enter the value in the appropriate language to translate the entire node (as we have done with the synopsis of the movie). If the content is a media type (image, video, etc.) and is different depending on the language, select the corresponding content to create the translation. For example, in our case the cover and trailer for the movie will be in a different language.

However, if the fields value comes from a list of Taxonomy items, the translation is done in the vocabulary, as we will discuss in the following section. For now, **Rating**, **Country** and **Genre** continue to show the value in the base language, which is our case is Spanish.

Taxonomy Translation

35.8

The name and description of each vocabulary item can be modified from the **Translate** tab in the taxonomy editor. F35.39



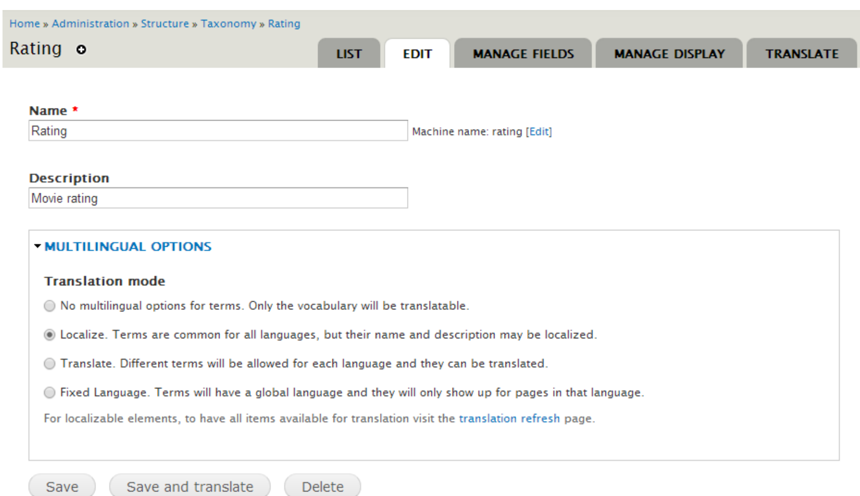
F35.39

Translate Taxonomy

We are able to update the vocabulary name and description from the Translation tab.

To translate vocabulary terms we must first configure the behavior of each vocabulary in a multilingual site. In the vocabulary editor, we can select the following **Multilingual Options**: F35.40

- **No multilingual options for terms**. We can translate the vocabulary, but not the words.
- **Localize**. The terms are common to all languages, but the name and description can be translated.
- **Translate**. Created terms have a language associated to them and can be translated. In this case the terms are independent, and only shown in the active languages of the site. This allows greater flexibility when the terms of a vocabulary may be different depending on the language. In this way the terms are treated as **translation sets**, in the same was a menu items. **This is the recommended method**.
- **Fixed Language**. The terms have a fixed language and only appear when the page is being displayed in that language.



F35.40

Multilingual Vocabulary Options

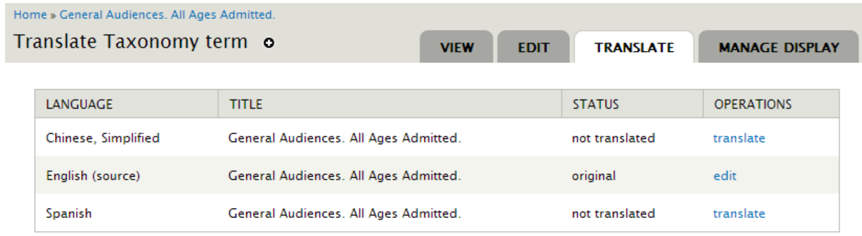
The translation of terms depends on the settings applied to Multilingual vocabulary.

If the method selected is **Localize**, the translation of the taxonomy terms is done from the **Translate** tab for the term by selecting each of the languages. F35.41

F35.41

Translation of Terms

Translation of terms is done one by one.

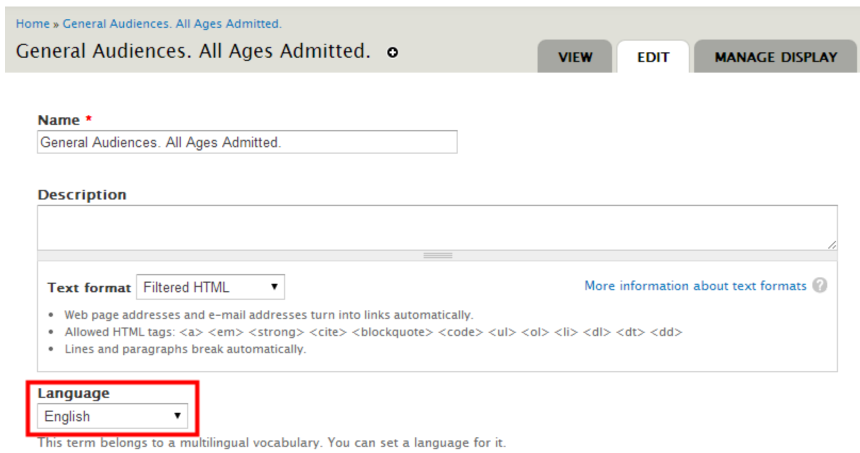


If the method selected is **Translate**, **we will create new terms for each language**, by selecting the language, and then linking the translation sets. This is the recommended method. F35.42

F35.42

Translation of Terms

When we have selected the Translate method, each term will have an associated language, and then they will be linked to the same term in other languages.



Views Translation

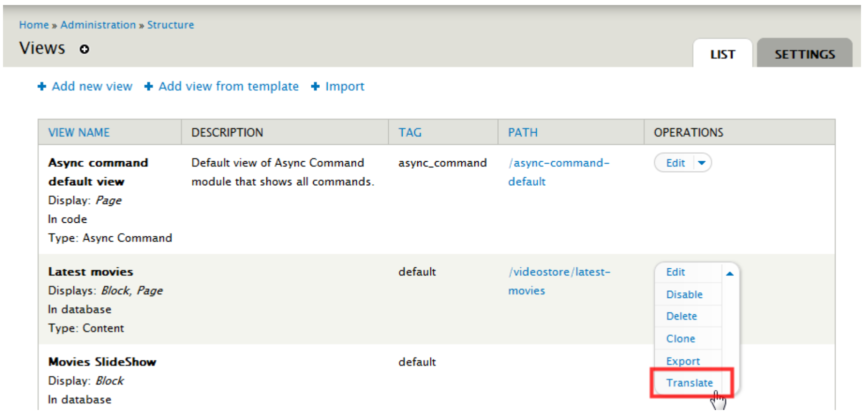
35.9

So far, we have studied that we can translate virtually every element of the website. We still have, however, a very important element, views.

The translation of the content displayed by a view will depend on previously translated content. Therefore, when we are speaking of translating views, we are only referring to the translation of the labels and other interface elements visable but not the results. In order to translate views, we must install the **Internationalization views** module, available at:

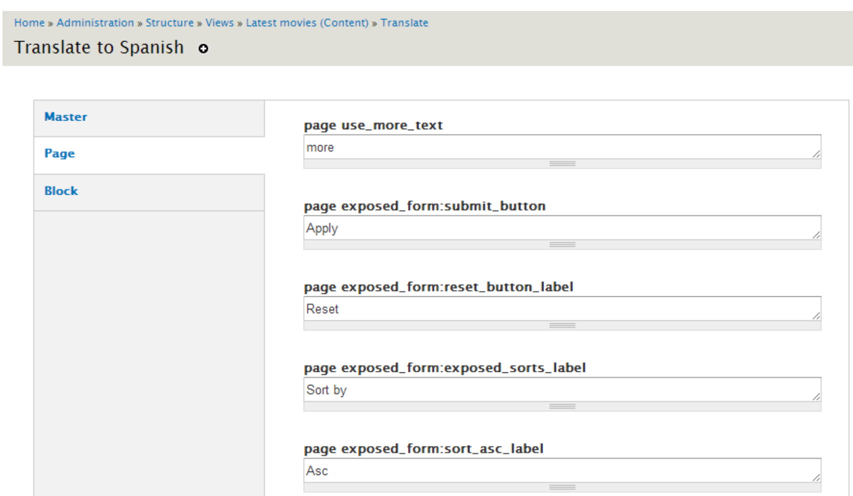
http://drupal.org/project/i18nviews

Once installed and activated (the name of the module is **Views translation**), we can access the translation of each view from the List tab in the Views management page. F35.43



F35.43
Translate Views
It is possible to translate elements of a view through the additional module Internationalization views.

First, select the language you want to translate the view to. The elements of the view that can be translated will be grouped under that submission. F35.44



F35.44
Translate Views
The elements that are translatable are displayed in groups in the view display. First select the language and then translate the information available.

35.10 Additional Modules

To end this unit we will review some additional modules related to Multilingual Sites.

Language Icons Module

The **Language icons** module adds icons with flags associated to each languge in the **Language switcher** block and the languge links in the Nodes.

The module is available at:

http://drupal.org/project/languageicons

Once installed, the icons for the languages are displayed by default. F35.45

F35.45
Language Icons Module
The Language Icons module adds language icon flags to language links.

Translation Overview Module

The **Translation Overview** shows in one screen the translation status of the nodes of the site. From the summary we can link to the translation of each node, thus we can create translations as well as edit existing translations from one location.

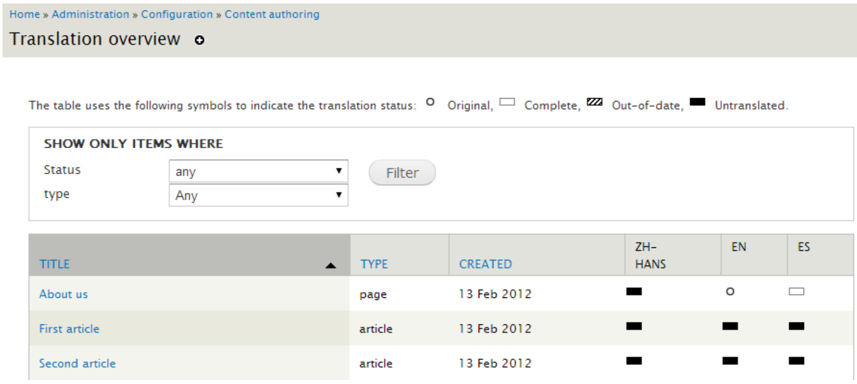
The **Translation Overview** module is available at:

http://drupal.org/project/translation_overview

Once installed the summary of translations can be found at: F35.46

URL Translation Status
/admin/config/content/translation-overview-manage

Administration ⇨ Configuration ⇨ Content authoring ⇨ Translation overview



F35.46

Translation Overview Module

Displays a list of site content with the overall translation status. For each node, it shows if a translation has been made for each of the languages.

Translation Table Module

The **Translation table** module provides an interface to easily translate some elements of a site such as vocabularies and taxonomy terms, menu items, field names, and types of content. No doubt this is an essential building block to expedite the translation of the various elements of the site.

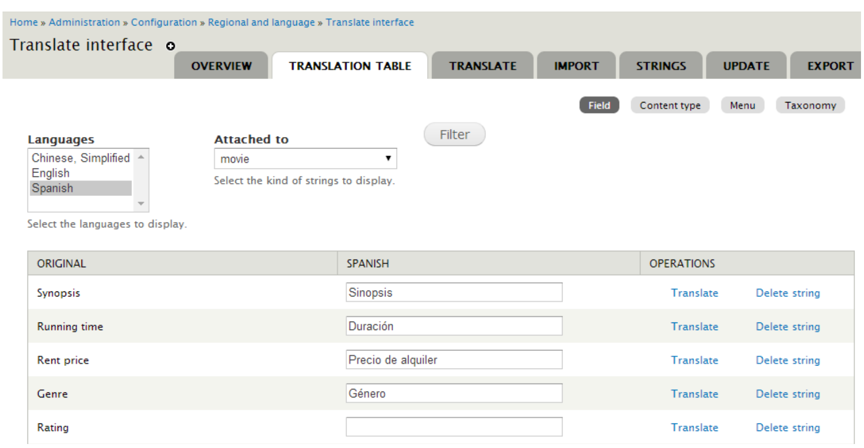
The **Translation table** module is available at:

http://drupal.org/project/translation_table

Once installed, you can access the translation table from: F35.47

Administration ⇨ Configuration ⇨ Regional and language ⇨ Translate interface [Translation table tab]

URL Translation table
/admin/config/regional/translate/table/menu



F35.47

Translation table Module

The translation table allows for easy translation of elements such as fields, menu items, taxonomy, etc.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Localization client module

The **Localization client** module allows users to make or correct translations of the interface on the page itself, which facilitates and streamlines the process of string translation.

The Localization client module is found at:

http://drupal.org/project/l10n_client

The module does not support the management overlay, so we have to deactivate the **Overlay** module when we go to translate the interface using this module.

To use this module click on the **TRANSLATE TEXT** button displayed in the footer of the page. A window will display with the texts to translate depending on the page loaded at that time. The translation will be provided from the source language, which is English, the active language. F35.48

F35.48
Localization Client Module
Allows the translation of the interface on the same page while being viewed.

36 E-commerce: Introduction to Ubercart and Drupal Commerce

There are two solutions in Drupal 7 to integrate e-commerce functions and implement an online store: Ubercart and Drupal Commerce.

Drupal Commerce is only available for Drupal 7, thus it uses in a more optimal way its new functionality, like entities and associated fields. It also intensively uses the Rules modules, already studied in this course.

Even though Drupal Commerce is much more flexible than Ubercart, it is also much more complex to set up. However, its flexibility is especially appreciated on websites that require multiple caseloads configuration.

The contents of this unit should be understood as an introduction to both solutions, Ubercart and Drupal Commerce. Both modules require an in-depth and, above all, practice to get to master them entirely.

We recommend that you use only one of the two e-commerce solutions: Ubercart or Drupal Commerce. Choosing one or the other will depend on the preferences of the developer and the specific needs of the store to be developed.

Comparative D7/D6

Ubercart

Ubercart version 3, exclusive for Drupal 7, includes important improvements which still make it competitive with Drupal Commerce.

Besides its adaptation to entities and fields in Drupal 7, Ubercart 3 integrates with Rules, which adds flexibility when configuring the store and adding actions.

Drupal Commerce

Drupal Commerce is a suite created specifically for Drupal 7, and designed like a clear alternative to Ubercart. Drupal Commerce uses Rules intensively, which makes it a very flexible module that enables practically any type of configuration.

Unit contents

36.1 The Ubercart module....464
36.2 Ubercart: Managing and Configuring the Store....466
36.3 Ubercart: Products....478
36.4 Ubercart: Purchases and orders....487
36.5 Ubercart: Conditional actions....491
36.6 Ubercart: Other functionality....492
36.7 The Drupal Commerce module....502
36.8 Commerce: Administration and configuration of the store....504
36.9 Commerce: Products....512
36.10 Commerce: Purchases and orders....515
36.11 Commerce: Other functionalities....519

36

36.1 The Ubercart module

The first of the e-commerce solutions that we will study in this unit is **Ubercart**.

Ubercart is a very mature module, with many contributed modules that complement its functionality and major improvements in version 3.x for Drupal 7.

Some of the most relevant features and functionalities of the **Ubercart** module are:

- Configurable product catalog.
- Powerful system for creation and configuration of items.
- Definition system for product attributes.
- Full purchase process.
- System for administration and tracking of orders.
- Integration with different payment gateways.

System requirements

Ubercart is another Drupal module, but it has certain specific features that require that certain technical requirements be supported.

Since Ubercart makes heavy use of system memory, you must configure PHP so that memory limit is at least 64MB instead of the 32MB recommended for basic installation of Drupal 7. If you work with images, you should increase the memory limit to at least 96 MB.

Moreover, certain **Ubercart** optional modules may require additional **PHP** functions such as **cURL, OpenSSL, SimpleXML, DOM** and **SOAP.** The modules we will study in this unit do not require these features. When using additional modules we need to check if the necessary additional functions are available and operational in the hosting service.

Ubercart installation

The **Ubercart** module is available at:

http://drupal.org/project/ubercart

Ubercart requires some additional modules, some already studied in this course:

- **Rules**, available at: http://drupal.org/project/rules
- **Views**, available at: http://drupal.org/project/views
- **Ctools**, available at: http://drupal.org/project/ctools
- **Entity API**, available at: http://drupal.org/project/entity
- **Entity tokens**, available at: http://drupal.org/project/entity

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

For certain additional functionality, although not strictly required, it is recommended to install:

- **Colorbox**, available at: http://drupal.org/project/colorbox
- **Google Analytics**, available at: http://drupal.org/project/google_analytics
- **Token**, available at: http://drupal.org/project/token

The **Ubercart** module is comprised of two core module groups: mandatory and optional. **Mandatory modules** are:

- **Cart**. Controls operation of the shopping cart.
- **Order**. Manages and tracks orders.
- **Product**. Manages items.
- **Store**. Configures parameters related to stock management.

It is recommended to only install the mandatory modules of **Ubercart** core first.

D7/D6 Comparison

Conditional actions

In Drupal 7 (Ubercart 3.x) the mandatory core module of Drupal 6 (Conditional actions) disappears. From this version, the conditional actions that may be required will be implemented with the Rules module.

Optional Ubercart core modules are:

- **Catalog**. Shows a product catalog following a hierarchical structure.
- **File Downloads**. Allows to associate downloadable files (e.g. manuals, descriptions, etc.) to items.
- **Payment**. Allows payment gateways to interact with the shopping cart.
- **Product attributes**. Allows customers to choose specific features of a product during the purchase process.
- **Reports**. Generates various reports regarding customers, orders, articles, etc.
- **Roles**. Assigns roles depending on the products purchased. Furthermore, these roles can be assigned an expiry date.
- **Shipping**. Prepares items for postal shipment.
- **Shipping Quotes**. Shows amounts related to shipments. For example, calculates the cost of shipping, depending on the total weight of the purchase.
- **Tax Report**. Generates reports related to taxes paid by customers.
- **Taxes**. Sets taxes depending on locations of customers and items.

In addition to these modules, the Ubercart distribution includes three groups of additional modules:

- **Ubercart – extra**. Adds new functionality to the general operation of Ubercart.
- **Ubercart – fulfillment**. Adds features related to shipping and handling costs.
- **Ubercart – payment**. It incorporates features related to payment processes.

Initially we will only activate compulsory modules: **Cart**, **Order**, **Product** and **Store**.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

36.2 Ubercart: Managing and Configuring the Store

Once the required kernel modules (Cart, Order, Product and Store) are installed, you can access the administration area of the store from: F36.1

URL Store
/admin/store

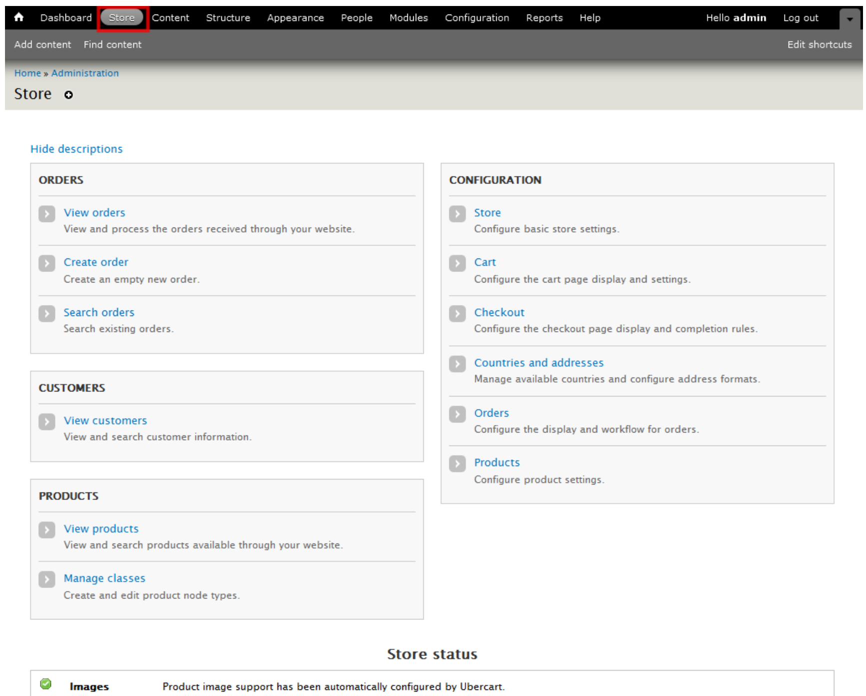
Administration ⇨ Store

The administration area of the shop is initially divided into groups of options: **Orders, Customers, Products** and **Settings**. These groups and the included options may be increased based on the activation of additional modules.

F36.1

Store

The basic installation of the Ubercart module includes a complete area for management of an online store.



Each group contains a set of options for performing different actions:

- **Oders**:
  - View orders.
  - Create order.
  - Search order.
- **Customers**:
  - View customers.
  - Search customers.
- **Products**:
  - View products.
  - Manage classes.
- **Configuration**:
  - Store.
  - Cart.
  - Checkout.
  - Countries and addresses.
  - Orders.
  - Products.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Next we will visit the various options available in the basic installation of **Ubercart**. We begin with the options in the **Configuration** section.

Store Configuration

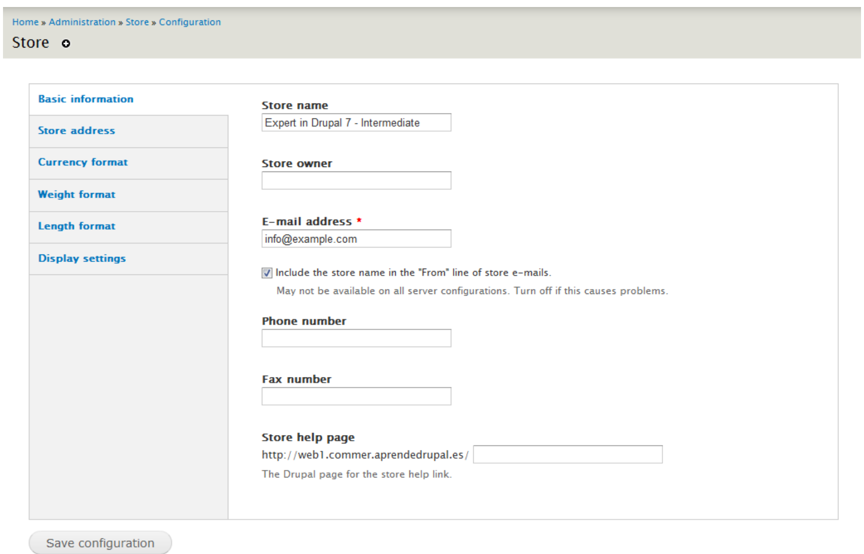
The general configuration of the **store** is made from:

Administration ⇨ Store ⇨ Configuration ⇨ Store

URL Store configuration
/admin/store/settings/store

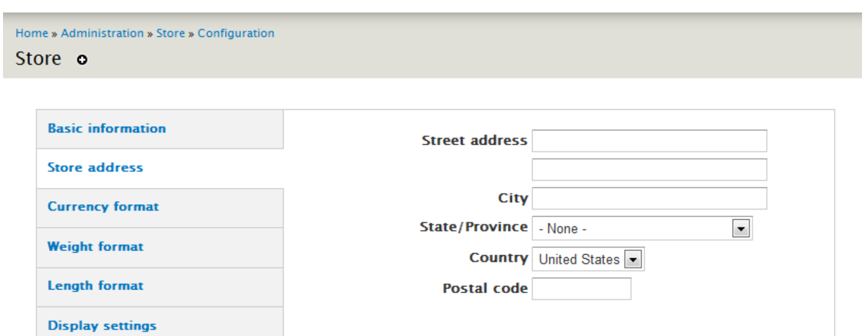
This section allows defining multiple general parameters for the online store:

- **Basic information**. General parameters, such as store name, store owner, email address, phone number, fax number and store help page. F36.2



F36.2
Store configuration
We can define general configuration options for our online store.
General parameters are configured from basic information, eg. name of the store, owner's name, e-mail, etc.

- **Store address**. Physical address of the store. F36.3



F36.3
Store address
Physical address of the store.

- **Currency format**. Configuration of the default currency format. Store default is set in U.S. dollars (USD). F36.4

  For the currency format the code used is the three-character ISO 4217 standard (http://en.wikipedia.org/wiki/ISO_4217). To figure the Euro we

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

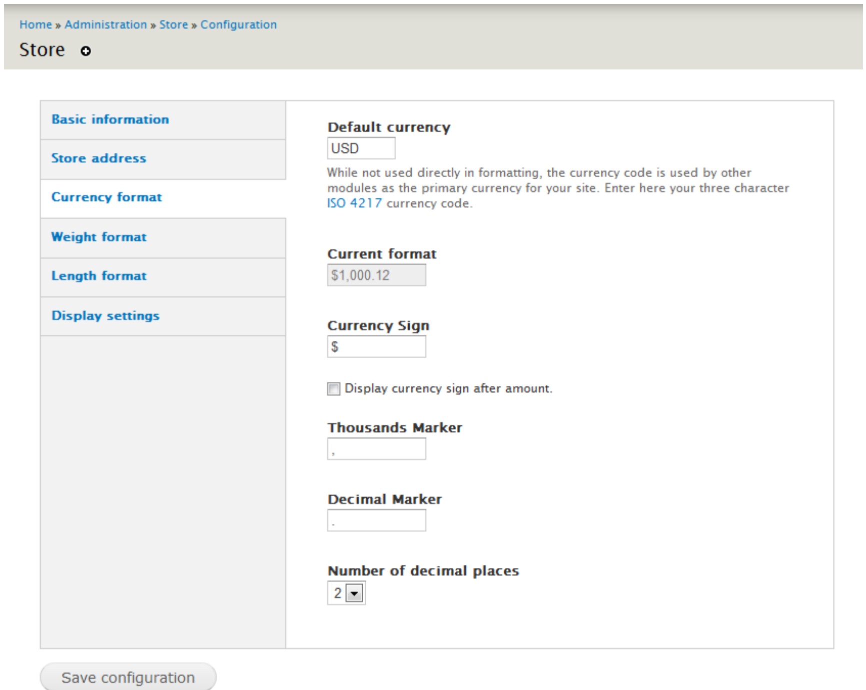
use the EUR code.

Besides the new currency code we must change the symbol (€). The imperial system uses the floating point as separator for decimals and the comma for thousands.

F36.4

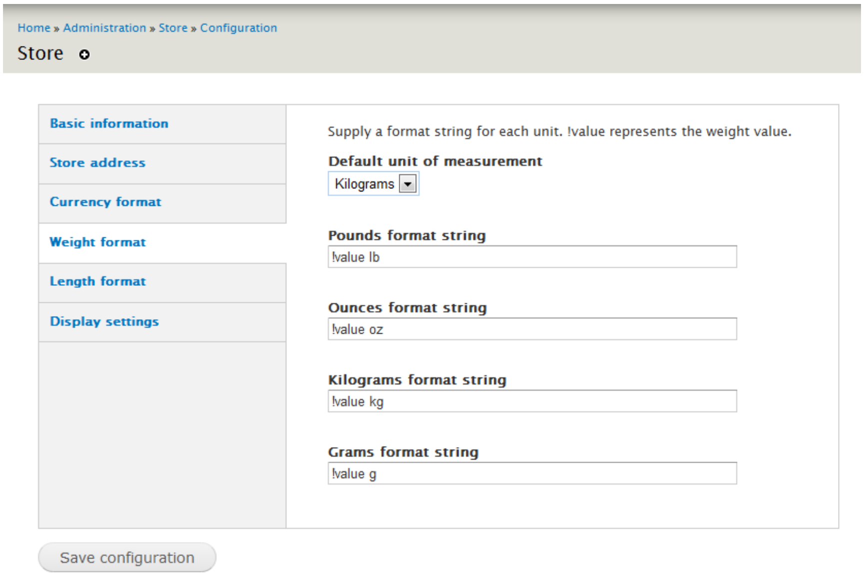
Currency format

Configuration of default currency.



- **Weight format**. We must also properly configure the unit of measurement for weights (kilograms or grams, depending on product). Other items available are Pounds and Ounces. F36.5

- **Length format**. Depending on how you want to express the measurements of the product, use Centimeters or Millimeters. Other available units are inches and feet.

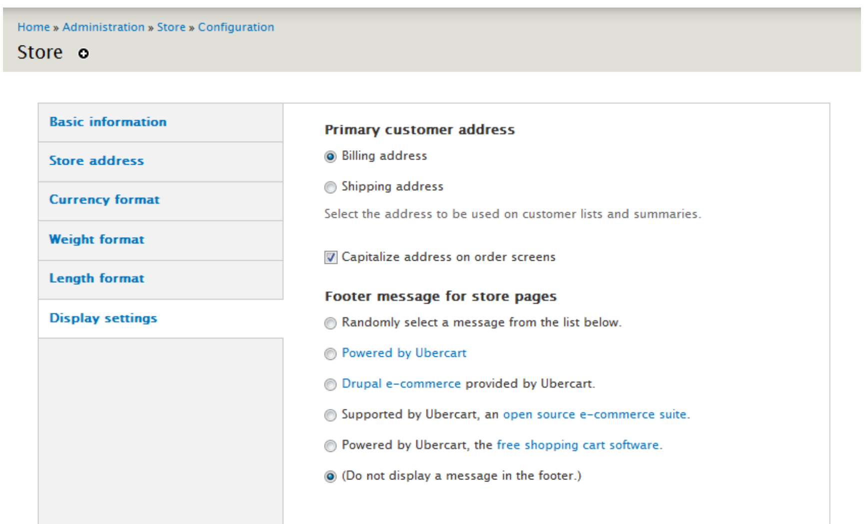


F36.5

Weight and length format

We set default units for weight and length.

- **Display settings**. Some methods of presenting information. We define which is the main address of the client (if shipping or billing) and what message (example: Works with Ubercart) is displayed on the bottom of the pages of the store. **F36.6**



F36.6

Display settings

Addtional options for display of information.

Cart configuration

Cart configuration is done from:

URL Cart configuration
/admin/store/settings/cart

Administration ⇨ Store ⇨ Configuration ⇨ Cart

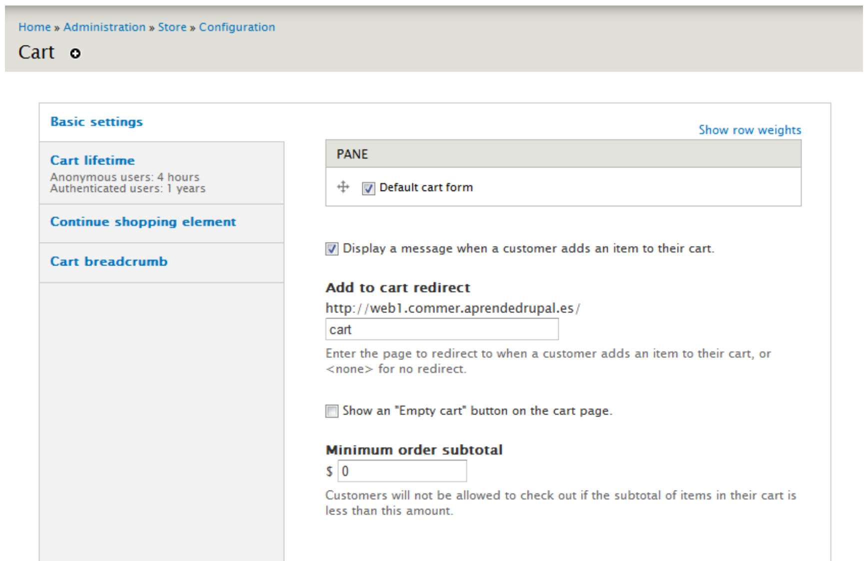
From this section you can set multiple parameters associated with the shopping cart:

- **Basic settings**. You can define a redirection every time a product is added to the cart, decide if we want a message to appear, etc. F36.7

F36.7

Cart configuration

Basic settings in the shopping cart: redirection every time a product is added to cart, messages to appear, etc.

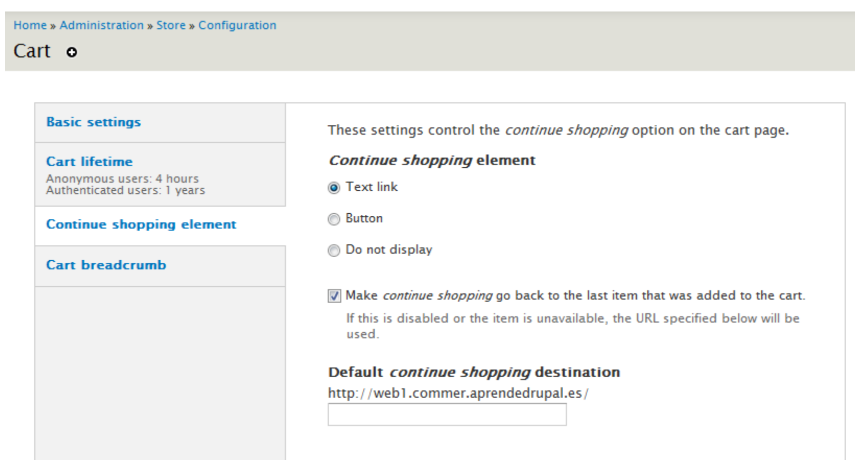


- **Cart lifetime**. You can set the time for products to remain in the cart both for anonymous users and registered users.
- **Continue shopping element**. Upon activation of the **Continue shopping element**, apart from the checkout button, another button or link will enable the **Continue Shopping** functionality. We can set this link to redirect to the page of the last item in the cart or define a URL (for example, the home page). F36.8

F36.8

Continue shopping element

In addition to the Checkout button, a button or link to continue shopping will be shown



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Cart breadcrumb**. We can define parameters for the links in the breadcrumb for the page detailing the contents of the cart.

Checkout configuration

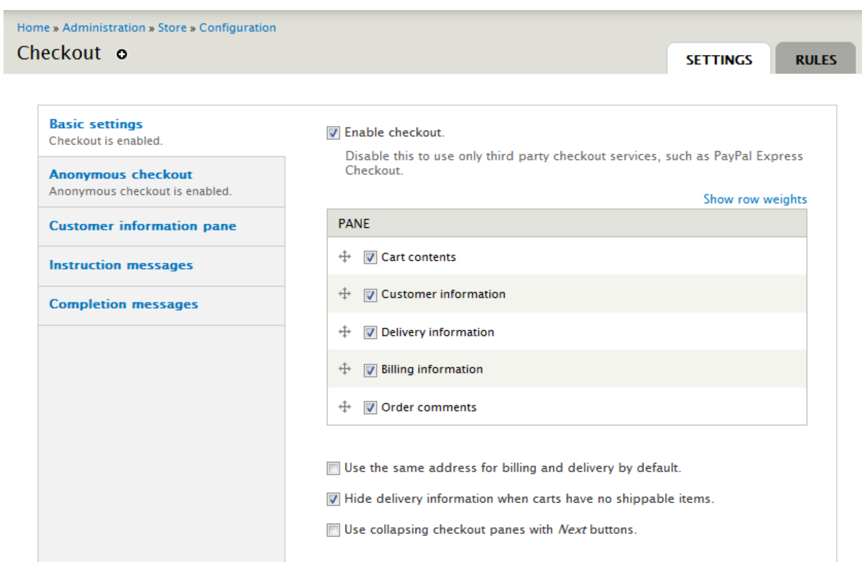
Checkout configuration is made from:

Administration ⇨ Store ⇨ Configuration ⇨ Checkout

URL Checkout configuration
/admin/store/settings/checkout

The configuration of the processes associated with the completion and payment of online shopping can be done from the following sections:

- **Basic settings**. Allows to set general options such as: F36.9
  - **Enable/disable checkout**, will be necessary if other external buttons for completion and payment of purchases, such as those provided by Paypal Express Checkout service (not to be confused with normal Paypal payment), or Google Checkout, are used. If we are not using these services leave this option enabled.
  - **Pane** (for termination and payment of purchase). Allows you to define which sections are shown to the client in the process of finalizing the purchase (cart content, customer data, delivery, billing, etc.).
  - **Otras opciones**. Other available options are: Hide shipping information when order is for downloadable products; and use independent panels with Next buttons.



F36.9
Payment configuration
Basic settings for completion and payment of purchase.

- **Anonymous checkout**. Sets the behavior of the system when an anonymous user makes a purchase. We can configure whether users may or may not make a purchase. If yes, during the buying process a new user account will automatically be created, and, since an order must necessarily be associated with a user, this new order will be linked to the newly minted user.

  The other options are related to the new user account: email requesting confirmation, request user name and password send an email with

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

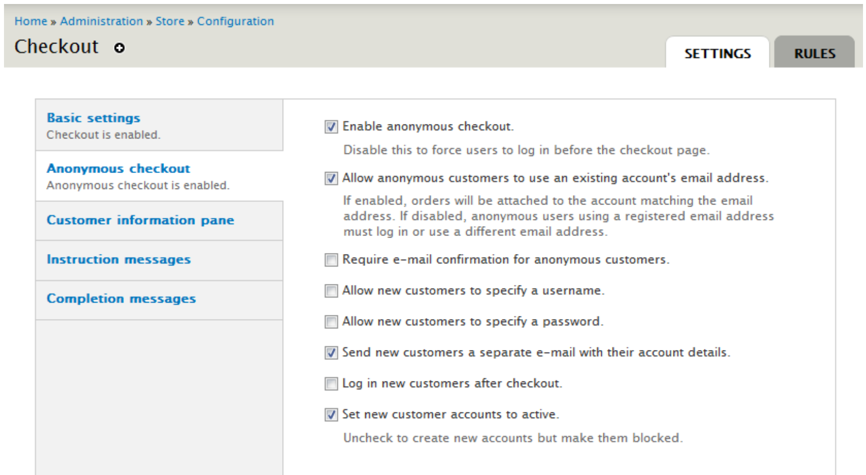
account details, etc.

If you do not want the user to access the site, we simply uncheck the **Set new customer accounts to active** option, so that all accounts created automatically by this pathway will be blocked. F36.10

F36.10

Anonymous users

Payment configuration for anonymous users.

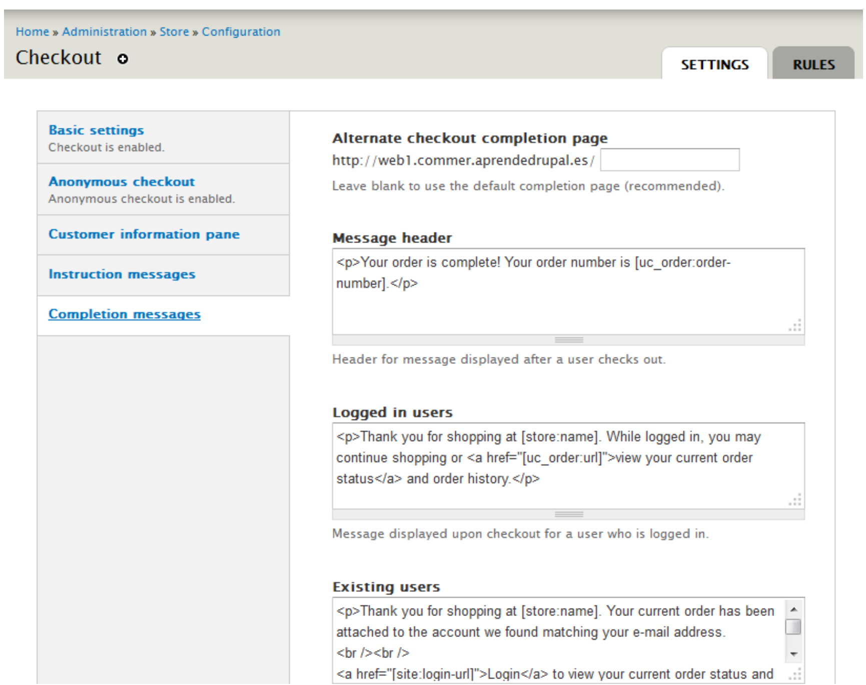


- **Messages**. The remaining groups of options allow configuration of messages displayed to the customer in the various steps of the buying process: message to create a new account, instructions for payment of the purchase and review of the order before payment, finally messages indicating the completion of the purchase. You can also include a URL to which the user is redirected once the purchase process is complete. F36.11

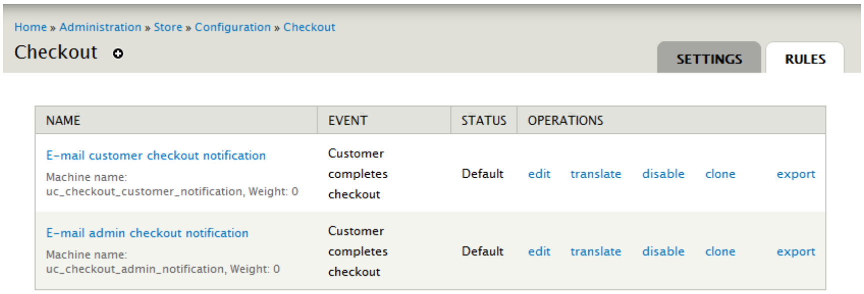
F36.11

Messages

The remaining groups of options allow for configuration of messages displayed to the customer in the various steps of the process.



- **Rules**. From the **Rules** tab you can configure the rules that will be evaluated and implemented during the process. In Drupal 7 Ubercart integrates with the Rules module, which allows you to define conditional actions and thus eases the configuration of the shopping process. For example, the module defines two rules related to payment: send an email to the customer and send an email to manager reporting the purchase. You can edit these rules or add additional rules from the administration area of Rules. We will see more about conditional actions with Rules in Ubercart in **paragraph 36.5**. F36.12



F36.12

Rules

Integration with Rules allows adding rules relating to the completion and payment of purchase.

Country and addresses configuration

Countries and addresses configuration is made from:

Administration ⇨ Store ⇨ Configuration ⇨ Countries and addresses

URL Country and addresses configuration
/admin/store/settings/countries

The following options are available for configuring countries and addresses:

- **Import countries**. Only countries defined in this section may receive orders. This means that an undefined country will not appear in the country selector for shipping at checkout. F36.13

  Ubercart initially installs Canada and the US. We can add other countries through the built-in import option. The list of countries to be imported is facilitated directly by Ubercart, so we will not need to install any additional package. Simply select the countries that you want to activate in the store and click on **Import**.

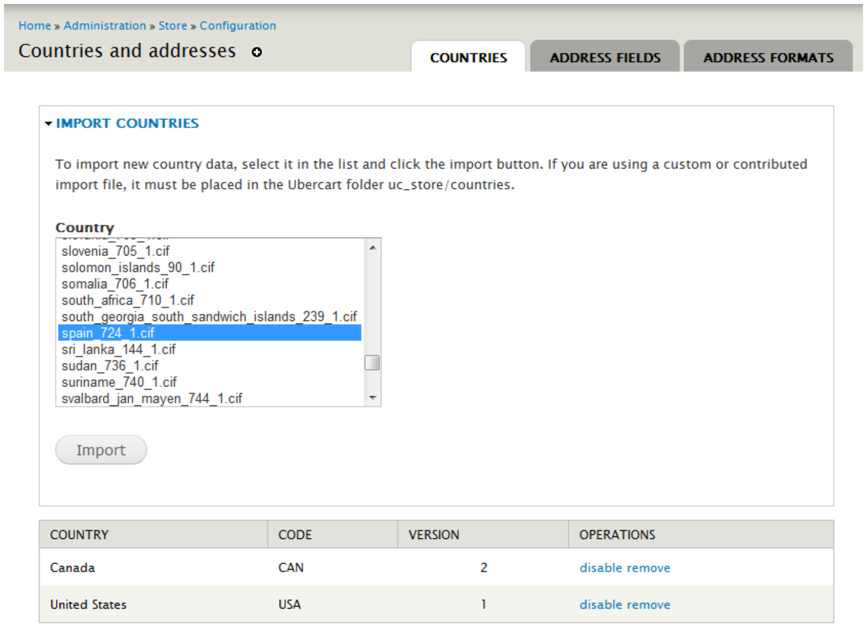
- **Address fields**. From this section you can configure which fields will be displayed to ask for the shipping address, specifying whether they are required or not. F36.14

- **Address formats**. We will be able to define those necessary variables to particularize shipping addresses to a certain country. A set of replacement patterns is provided with the available variables. F36.15

F36.13

Import countries

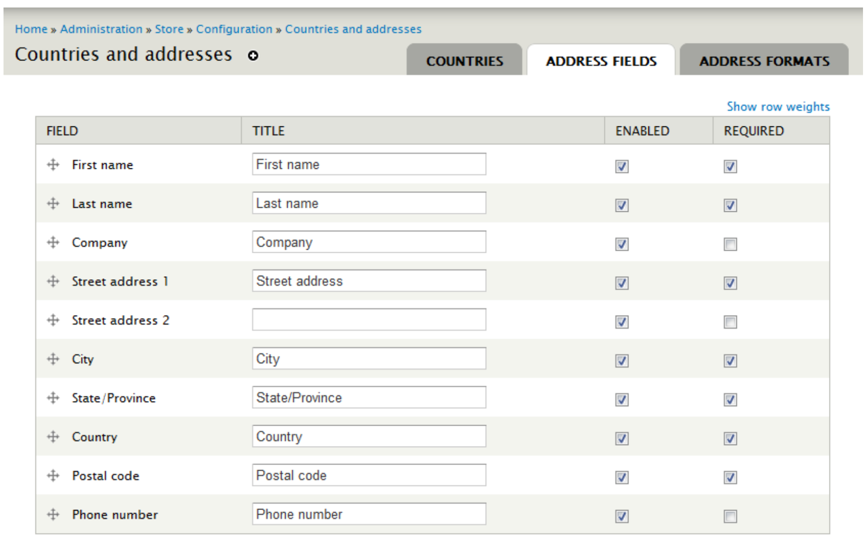
Only countries mentioned in this section may receive orders. This means that no undefined country will appear in the country selector for shipping at checkout.



F36.14

Address fields

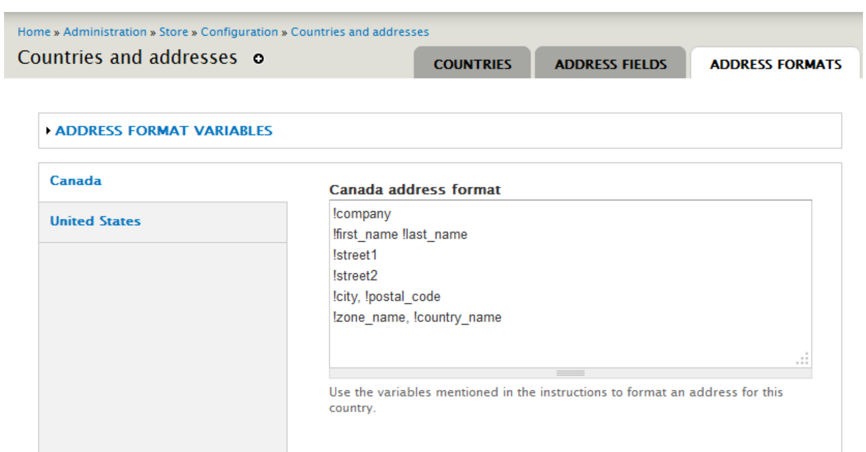
Customization of the address fields requested from customer.



F36.15

Address formats

Sets the order of address fields for each installed country.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Orders configuration

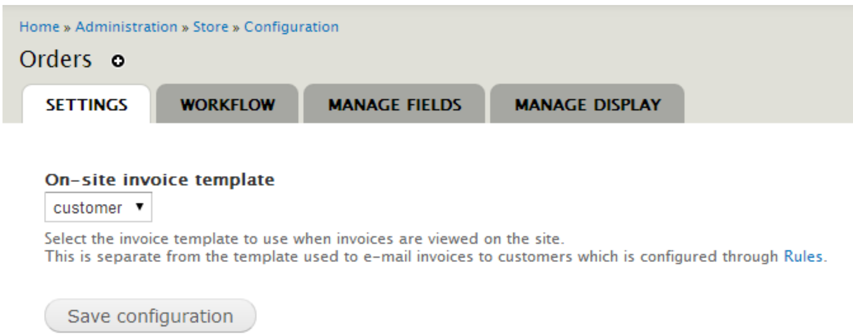
Orders are configured from:

Administration ⇨ Store ⇨ Configuration ⇨ Orders

URL Orders configuration
/admin/store/settings/orders

Among the groups of options you can configure:

- **Settings**. You can select a template to display the invoice in the browser. F36.16

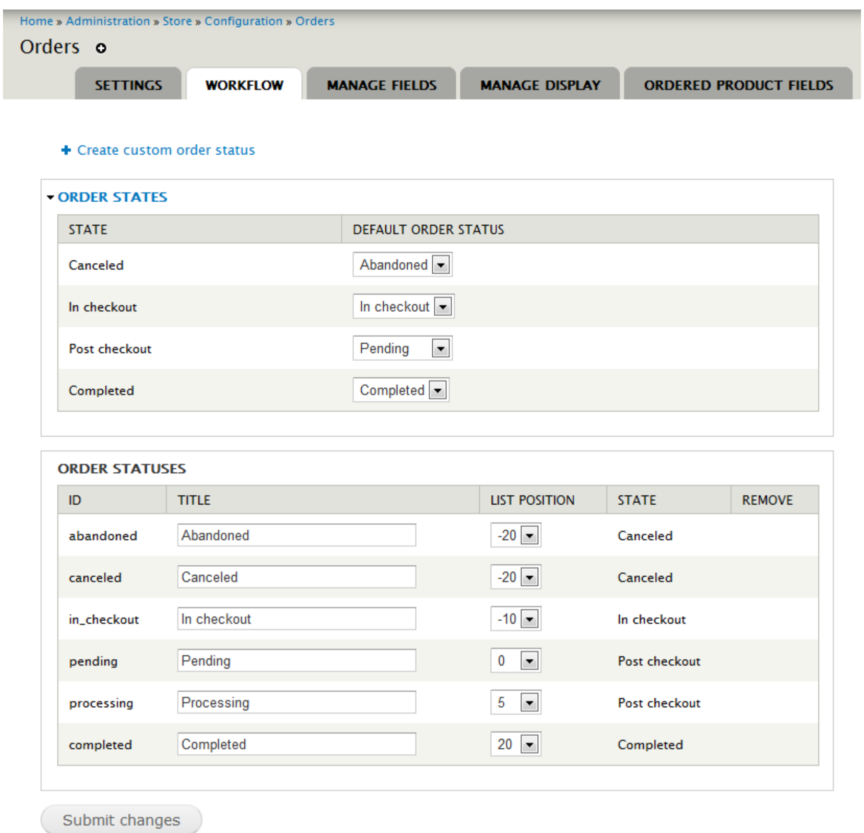


F36.16

Invoice options

We set if customers can access their invoices online.

- **Workflow**. From this tab you can define those states through which an order must pass. Ubercart defines by default a number of states and the flow associated with them. In this section you can rename, reorder or add new states. F36.17



F36.17

Workflow

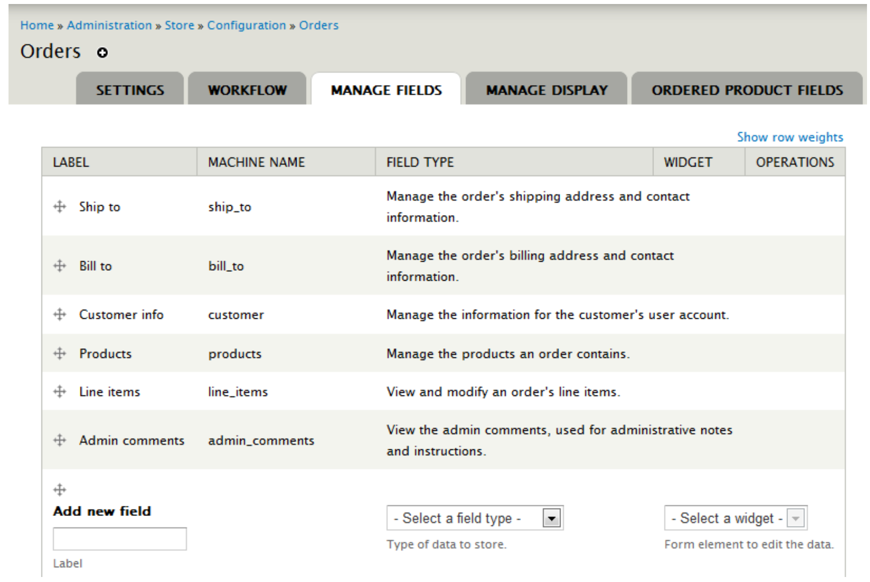
The order goes through different stages and states. From this option you can configure the relationship between stages and states and create new states.

- **Manage fields and display**. In Drupal 7 orders are defined as entities, allowing us to work with fields and their presentation as we have done with other entities. F36.18 F36.19

F36.18

Fields and order presentation

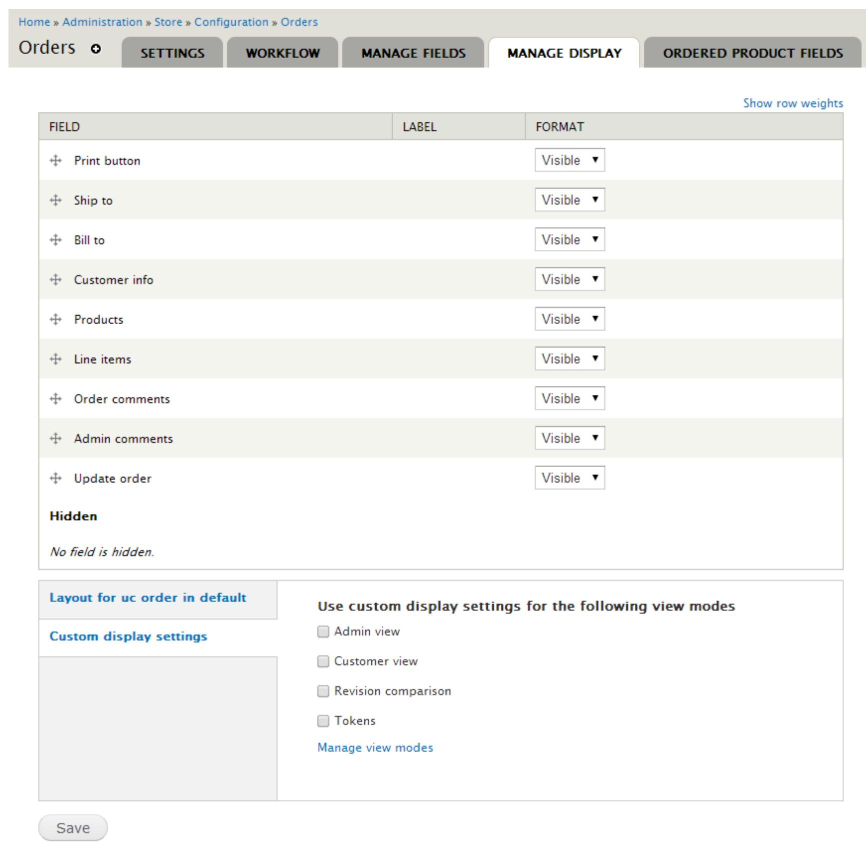
Orders are a new entity, so we can manage fields and their presentation as we would with any other entity.



F36.19

Manage display

From Manage display we can activate different view modes and configure which fields will be shown in the order.



Products configuration

Products configuration is done from:

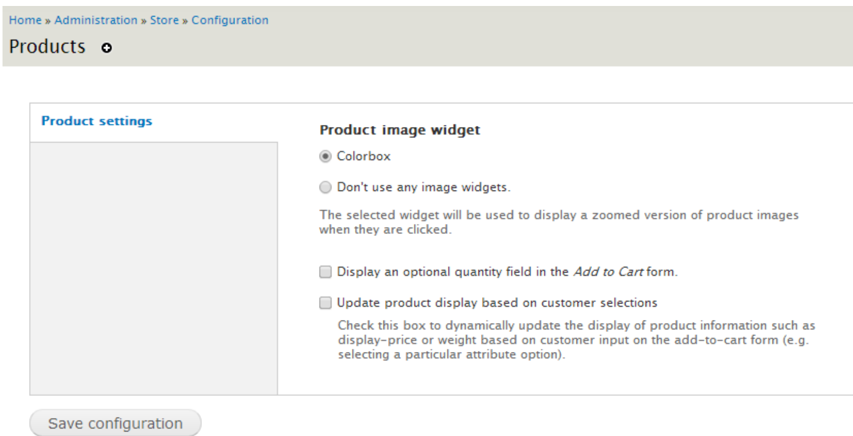
Administration ⇨ Store ⇨ Configuration ⇨ Products

URL Products configuration
/admin/store/settings/products

For **Products** we can configure: F36.20

- If an image widget like **Colorbox** will be used. The control is used to expand the product image by clicking on it. Controls available depend on installed modules.

- We can also add an optional field displayed for the user to enter **quantities** before clicking on the **Add to cart** button.



F36.20
Products configuration
Additional options for presentation of products.

36.3 Ubercart: Products

In this section we study the necessary actions to create products and product classes. Ubercart automatically adds a new **content type** called **Product** (with machine name **product**). This content type will allow us to create new items, following the usual process of content creation. As outlined below, we can use this content type or create new ones, allowing us to organize items by type (books, movies, t-shirts, etc.).

Product classes

Product class management is available at:

URL Product settings
/admin/store/settings/products

Administration ⇨ Store ⇨ Products ⇨ Manage classes

When you add a class, the **Class ID** will match the **machine name** of the associated content type. Therefore, such classes will be available for content creation as usual, thus enabling to create specific nodes for each class. F36.21

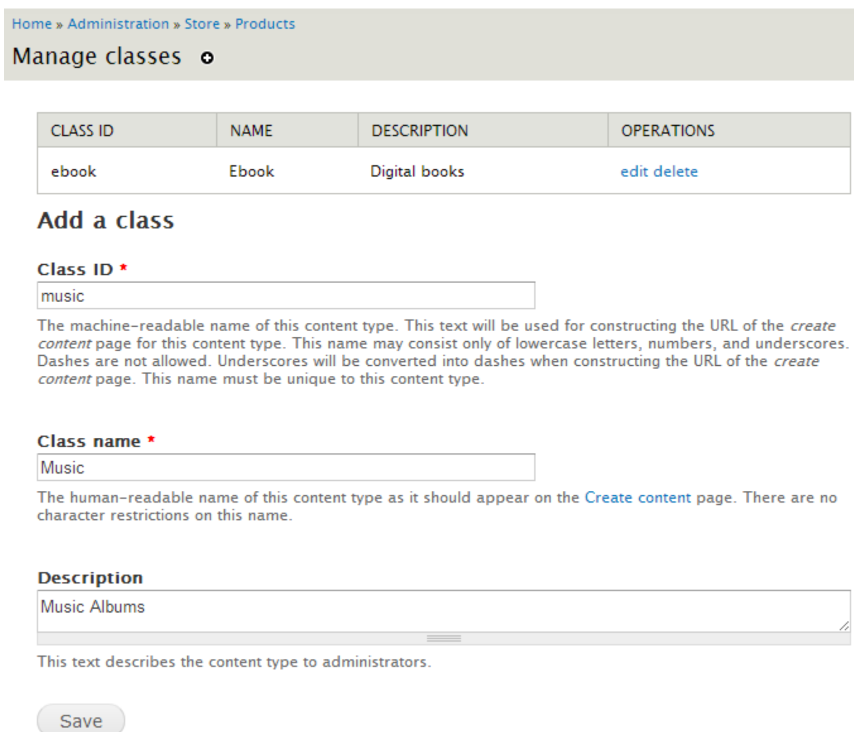
By creating independent content types for each product class, we wil be able to, among other things:

- Define specific attributes associated with each product class.
- Add additional fields to each class.
- Manage the presentation of products independently.
- Make specific views for a particular product class.

F36.21

Product classes

We can define different product classes to organize our online catalog.



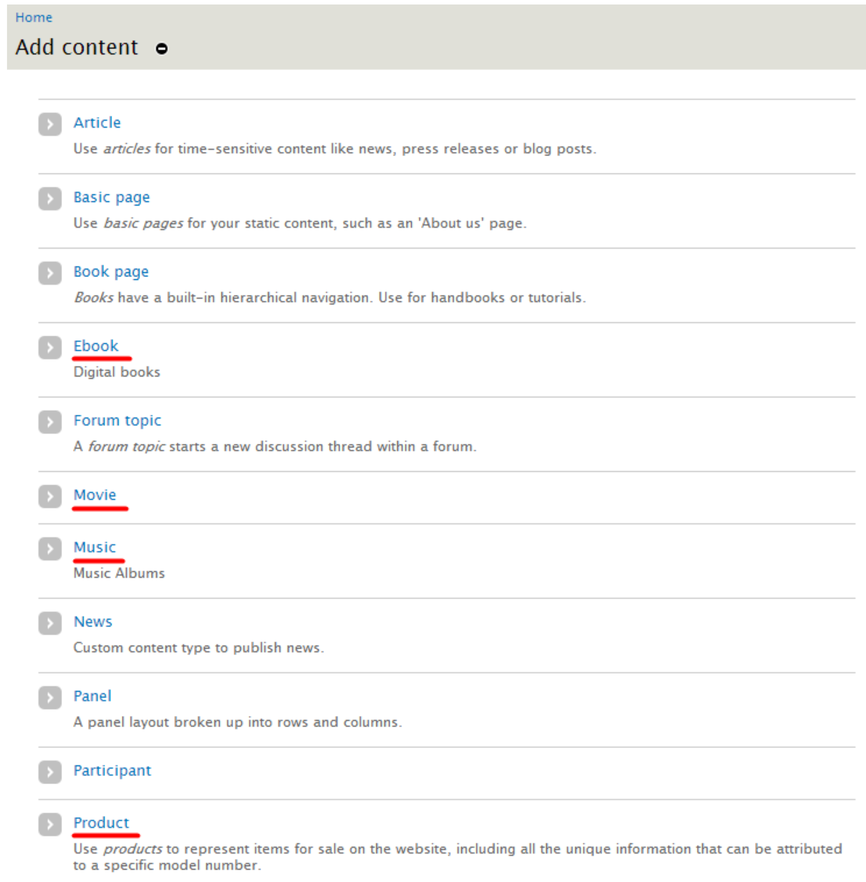
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

It is also possible to use an existing product content type, simply by making the **Class ID** match the machine name of the **content type.**

For example, we can transform in product class the **Movie** content type by adding a class with id **movie.**

Create products

From **Add content** we can create different types of products. These are shown mixed with other content types. F36.22



F36.22

Creation of different product classes

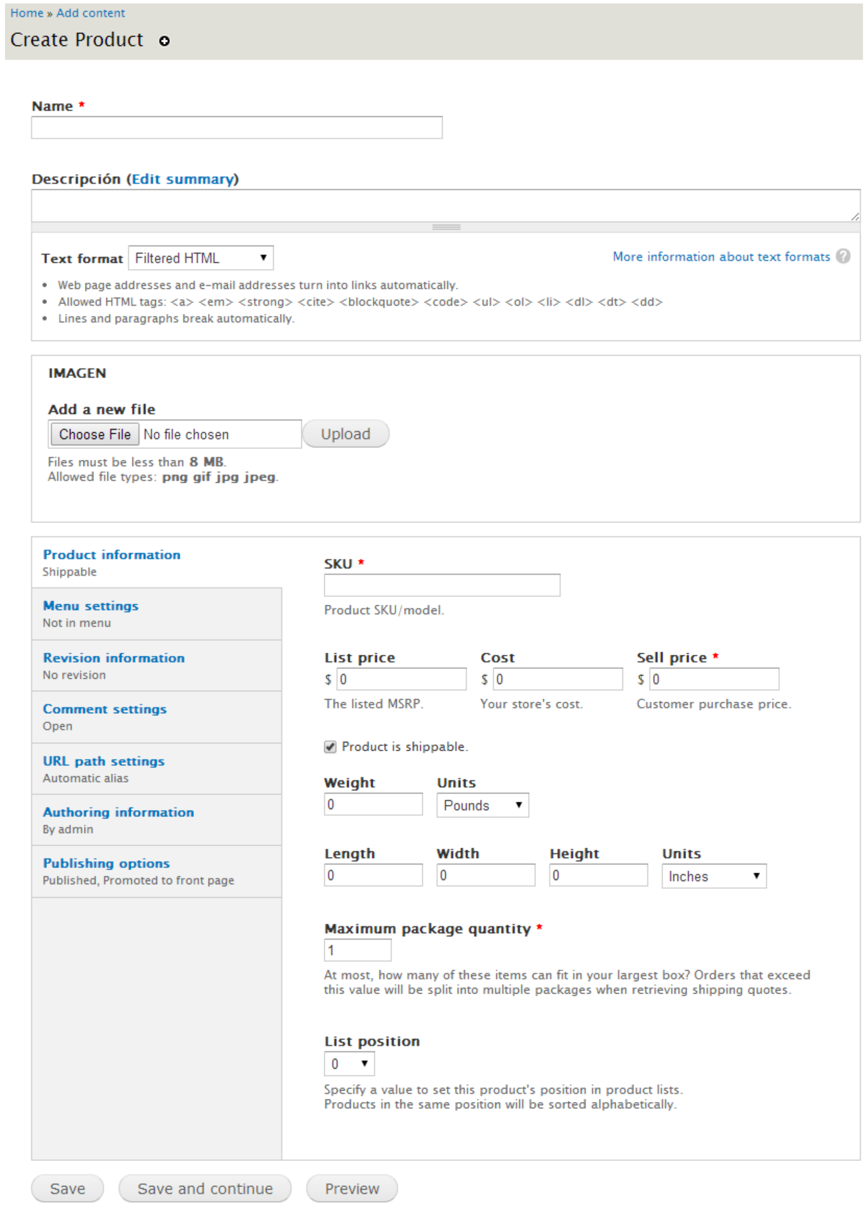
Product classes correspond to new content types with similar characteristics as the Product content type.

When creating a node from a content type which in turn has been defined as a product class, the edit form will be complemented with a group of options for **Product information**. F36.23

- **SKU** (Stock Keeping Unit). It must be a unique value that identifies each product, this can be the serial number, the reference, the ISBN, model etc.
- **List price**. This is an optional value not shown to the customer.
- **Cost**. Optional value not shown to customer. It is the cost of the item for the seller.
- **Sell price**. Currency amount of the sale. The sale price is mandatory.
- **Product is shippable**. If the item does not have an associated shipping process (such as a downloadable file or a subscription), unchecking this option prevents the checkout form from displaying paragraphs concerning shipping addresses.
- **Weight**. To indicate the item weight and the unit of measure.

- **Dimensiones**. To indicate the dimensions defining the lenght, height and width, and the unit of measure.
- **Maximum package quantity**. Only allows a certain number of items in a package. This will affect the calculation of shipping costs.
- **List position**. Sets the order in which items appear in catalogs. Products with the same position will be sorted alphabetically.

Moreover, Ubercart incorporates by default the ability to upload images associated with the item.



F36.23

Create product

Product creation form.

In Ubercart a product is a content type, with a group of additional options called Product Details, where we will complete specific information for the store.

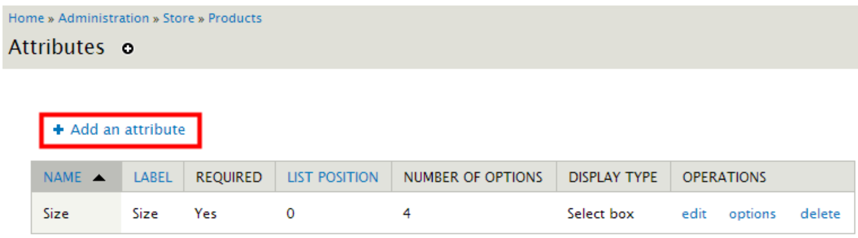
Product Attributes

By defining attributes, a product may have variations that the user can select to purchase. A typical example is the **size** attribute in clothing. The customer can choose a size, among those available, and thus will be served the right product. F36.24

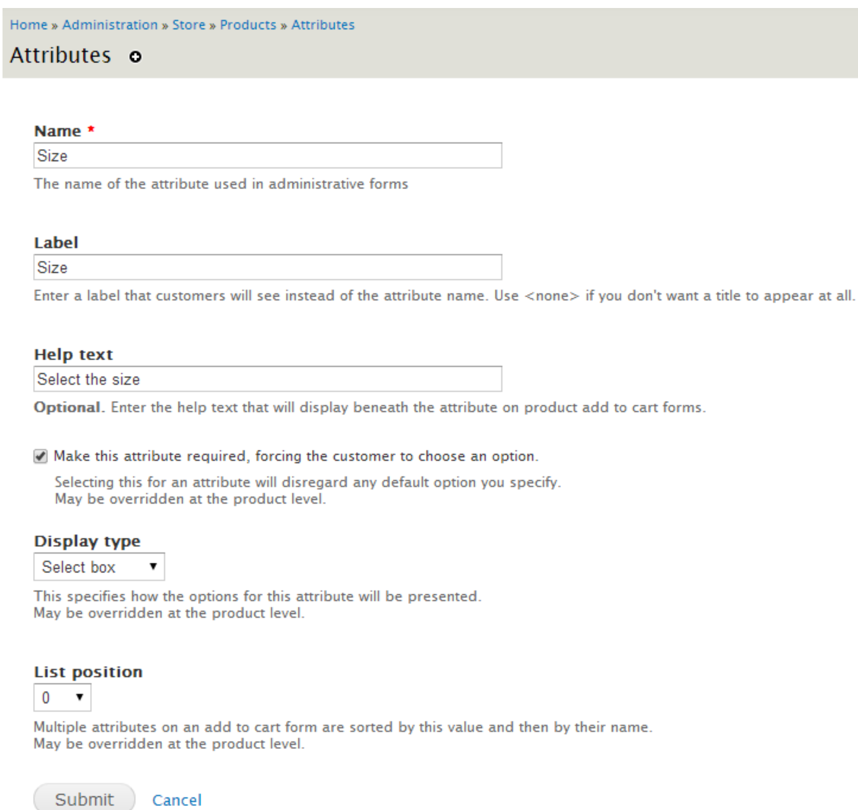
Product attributes will be available when the **Product attributes** module included in Ubercart is active. F36.25

URL Attributes
/admin/store/products/attributes

Administration ⇨ Store ⇨ Products ⇨ Attributes



F36.24
Attributes
With attributes we can add variants of the same product. For example, sizes of clothes.



F36.25
Attribute creation
The size attribute is shown as an example.
The customer selects a value for the attribute, so they will be handed a specific product.

Upon adding an attribute we define the following values:

- **Name**. The name of the attribute that is displayed in the administration form.
- **Label**. The name of the attribute that customers will see during the purchase process.
- **Help text**. It is a help text that is displayed on the attribute in the add to cart forms.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Make this attibute required**, forcing the customer to choose an option.
- **Display type**. We can choose how we want to display the options associated with the attribute: text area, selection list, radio buttons or checkboxes.
- **List position**. We can define the sort order of the attribute relative to other attributes of the same product.

Once the attribute is created we must add the **options** available to the user for selection (link **Add an option**). For example, for the **Size** attribute we could have options **XL**, **L**, **M** and F36.26 **S**.

F36.26

Attribute options

In the example we define the options available for the Size attribute. The values of cost, price and weight affect the value of the product (add or subtract to the value).

For each option you can indicate **default adjustments for Cost**, **Price** and **Weight**. These values, which can be positive or negative, are relative and can be added to or subtracted from the value indicated for the product. For example, we can say that T-shirts size XL will have an additional price of $ +1. These values can be overridden by product.

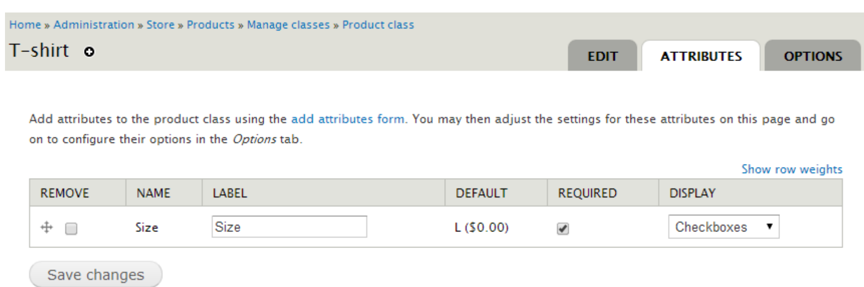
Attributes in classes

Once an attribute is created and its options defined, we can modify any defined **class** and add the attribute we want through the new **Attributes** tab F36.27 available.

F36.27

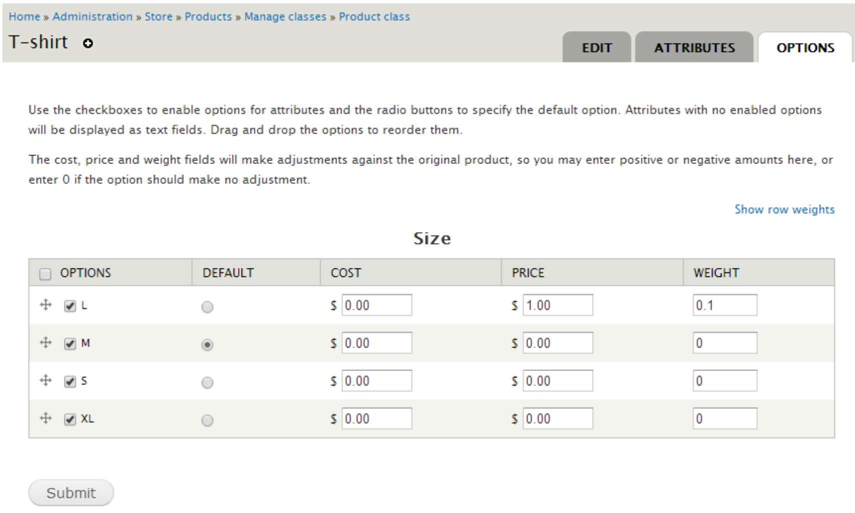
Attributes in classes

An attribute added to a class will be available for all products of that class. The attribute values can be modified for each product.



From the **Options** tab we can adapt the attribute for this particular product class. Remember that the values for cost, price and weight are relative (+ / -) to the value indicated on the product. F36.28

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



F36.28

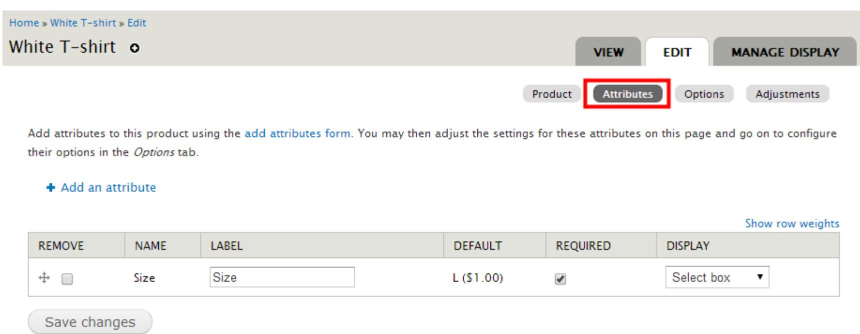
Attributes in classes

From Options we set specific values for the class attribute.

Attributes in products

Upon creating a product from a class for which attributes have been defined, these predetermined attributes will be applied automatically.

Once in product edition, from the **Attributes** tab we can modify available attributes or even create new attributes specifically for that product. F36.29



F36.29

Attributes in products

Assigning attributes to a particular product.

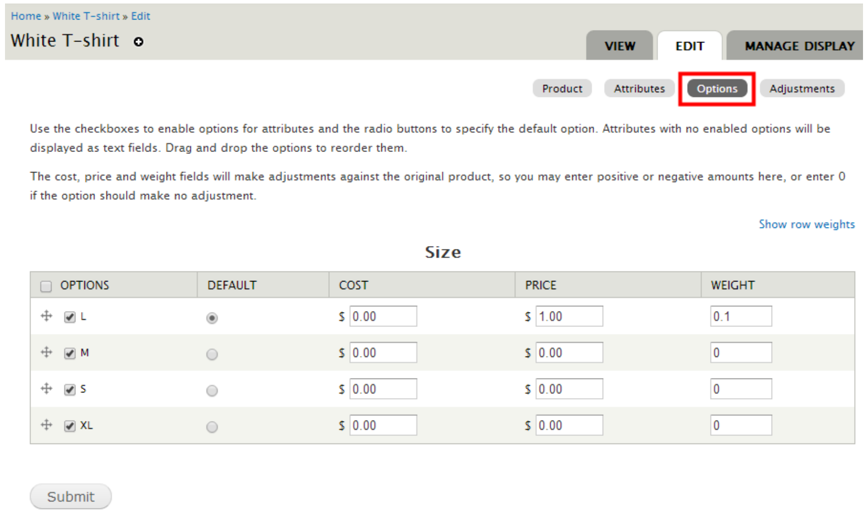
It is also possible to particularize the options associated with an attribute by clicking on the **Options** tab: F36.30

- We can decide **what options are available in the attribute** for that particular product. For example, assume that for product White T-shirt, size S is not available.
- We can decide **which option is the default and rearrange them** in that particular product.
- We can **modify the settings of cost, price and weight** for that particular product. Remember that values for cost, price and weight are relative (+ / -) to the value indicated on the product.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F36.30

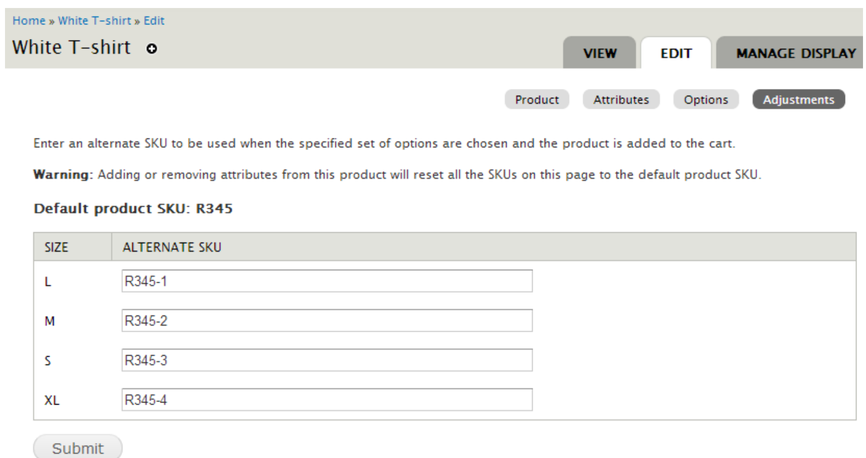
Attributes in products
From options we can set parameters of attributes specifically for a certain product.



Finally we can also assign different **SKU** codes associated with different options by clicking on **Adjustments**. Assigning different SKU codes for different versions of the product will streamline the search and identification process in the warehouse of physical products to be shipped. **F36.31**

F36.31

Attributes in products
Each option of an attribute can refer to different products. From Adjustments we can specify the SKU for each product.



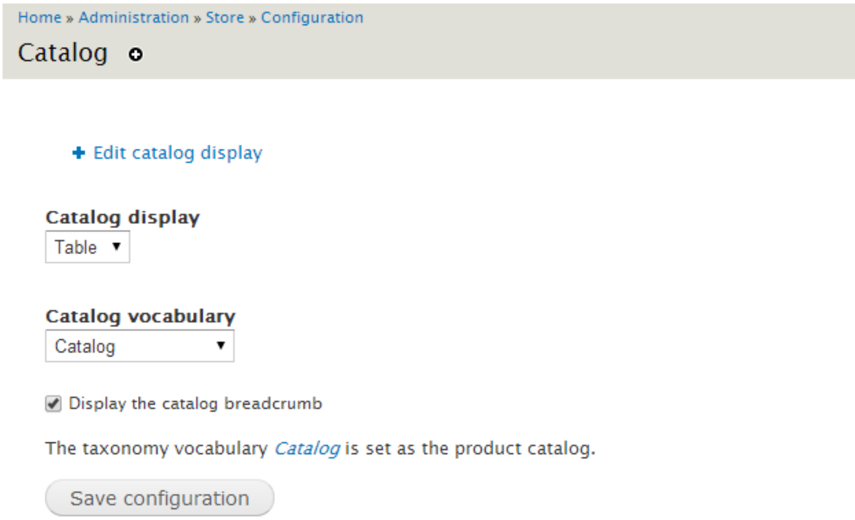
Product Catalog

Ubercart includes the **Catalog** module in its core, in the optional modules section. This module allows the organization of products in catalog form. **Catalog** uses Drupal taxonomy system, allowing classification of articles in different categories.

Once installed, we can configure options from: **F36.32**

URL Catalog configuration
/admin/store/settings/catalog

Administration ⇨ Store ⇨ Configuration ⇨ Catalog

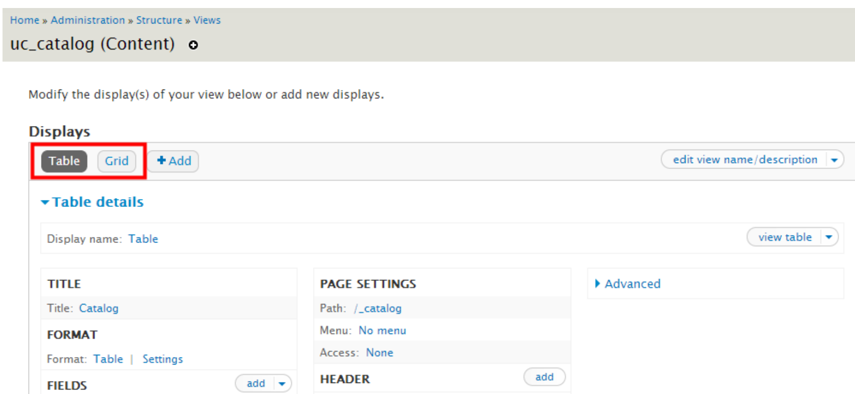


F36.32

Catalog configuration

We can configure the operation of the catalog.

The catalog's display is performed through the **uc_catalog view**. The view initially has two presentations, as Table or Grid. To change any parameter of the view (number of elements, fields, etc.) we just need to edit the view that we can access directly through the link **Edit catalog display**. F36.33



F36.33

Catalog view

The fact that the catalog is configured from a view allows for full customization. We can select from the table presentation mode and grid mode.

We can also customize the view or even create new presentations.

We can also select **Catalog vocabulary**. The module creates the vocabulary **Catalog**, initially **without terms**. It is recommended to add an image to each term, which will serve as presentation of the category in the catalog. The catalog can have several levels deep, creating nested terms.

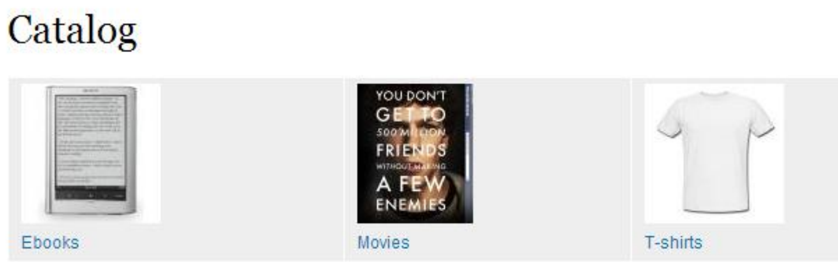
In all product content types a field is available to select a term from the catalog vocabulary. Once the catalog is set up and its terms added, we will edit each product and assign the corresponding term or terms (the same product can be included in several categories of the catalog).

The catalog page will be available from URL /catalog (if we want to change this URL we can add an alias for it). In the catalog only main categories are initially displayed, and clicking on a category will get to the children categories (or subcategories) and eventually to the product listing. F36.34

F36.34

Catalog presentation

Catalog page accesible from /catalog.

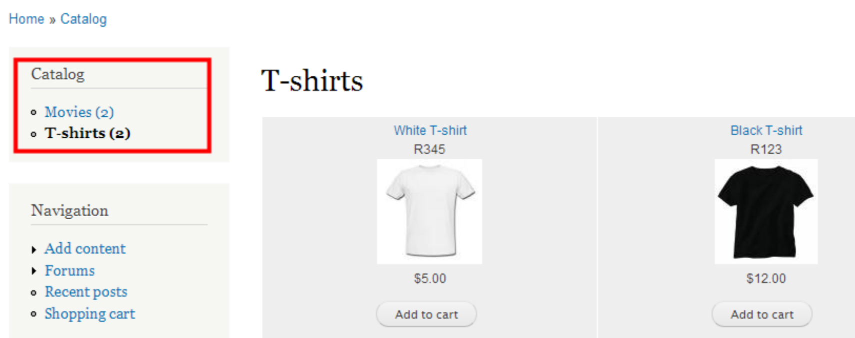


The **Catalog** module also adds a Catalog block, initially disabled. The **Catalog** block shows the defined categories and, optionally, the number of products within each of them. F36.35

F36.35

Catalog block

The catalog block shows the terms and the amount of products in each category.



Finally a product is shown, its presentation is fully configurable, just like with any other content type. F36.36

F36.36

Product presentation

Default product presentation. To change the presentation of the product we can use any of the techniques already seen (eg/ with Display Suite).

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Ubercart: Purchases and orders

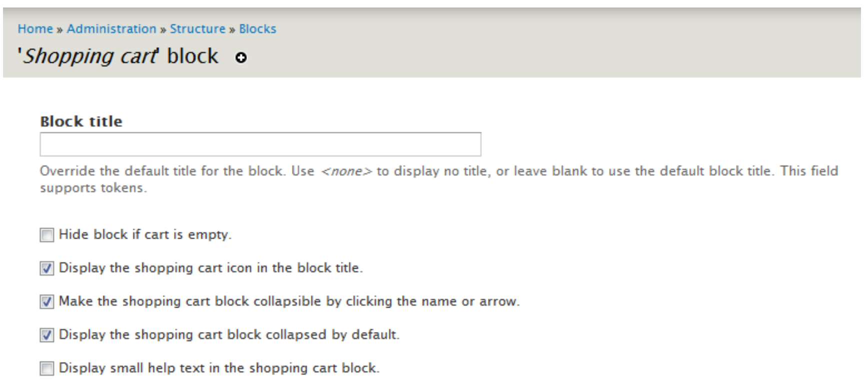
36.4

Shopping cart block

The **Shopping cart** block, initially disabled, displays products added by the user to the shopping cart.

From the block configuration we can set the following options: F36.37

- **Hide block** if cart is empty.
- Display the **shopping cart icon** in the block title.
- Making the **shopping cart block collapsible**. When the block is collapsed only the total number of items and the total price is shown.
- Display the shopping cart block **collapsed by default**. Clicking on the title will show the extended cart, with details of all the added items.
- Show **help text** (you can specify the text to display).



F36.37
Shopping cart block
Configuration options for the Shopping Cart block.

The **Figure** F36.38 shows the **Shopping cart block** unfolded, with details of the items the customer has added.

F36.38
Shopping cart block
The block shows the products and total amount of the purchase. It can be collapsed (compact) or extended (detail).

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Checkout process

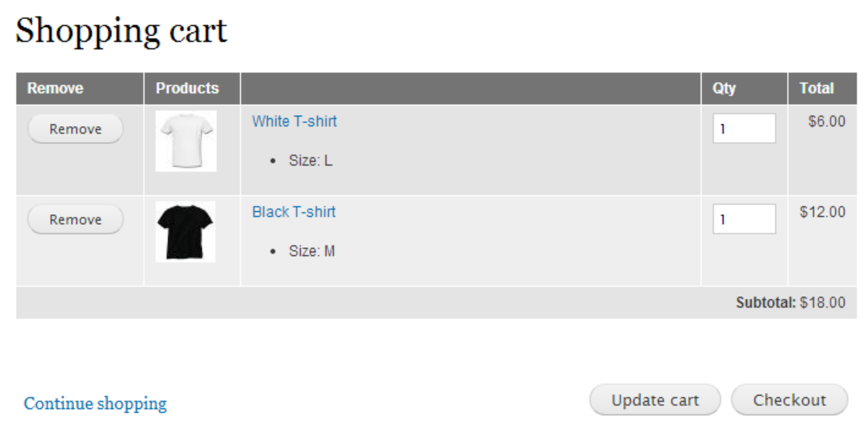
We go over the buying process from the point of view of the customer. The customer will add to the shopping cart the purchased items by clicking the Add to cart buttons to be found in both the product page and in the catalog (or other custom views that you created).

At any time the customer can review the state of the cart by clicking on the **View Cart** link from the **Shopping Cart** block. F36.39

F36.39

Shopping cart

Displays full page detail of shopping cart. You can make changes in products added, continue shopping or checkout.

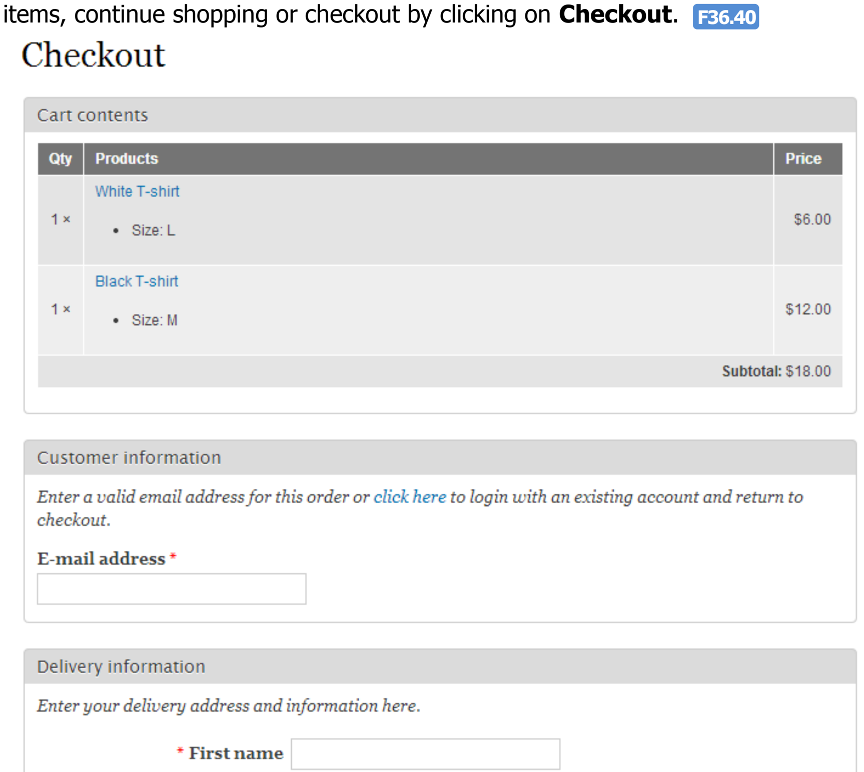


From the shopping cart, users can change the number of purchased items, delete items, continue shopping or checkout by clicking on **Checkout**. F36.40

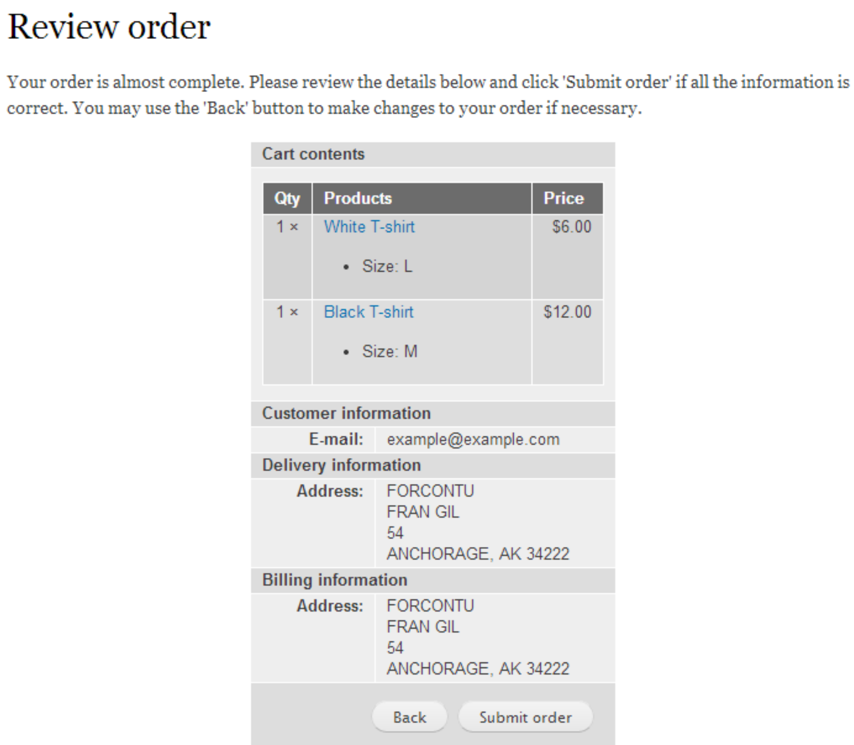
F36.40

Payment

This step begins the process of completion and checkout.



If payment methods are configured by activating the Payment module and the modules associated with each method of payment (Paypal, Google Checkout, etc.), the payment form will ask the customer to select one of them. F36.41


Review order

Your order is almost complete. Please review the details below and click 'Submit order' if all the information is correct. You may use the 'Back' button to make changes to your order if necessary.

Cart contents

Qty	Products	Price
1 ×	White T-shirt • Size: L	$6.00
1 ×	Black T-shirt • Size: M	$12.00

Customer information

E-mail: example@example.com

Delivery information

Address: FORCONTU
FRAN GIL
54
ANCHORAGE, AK 34222

Billing information

Address: FORCONTU
FRAN GIL
54
ANCHORAGE, AK 34222

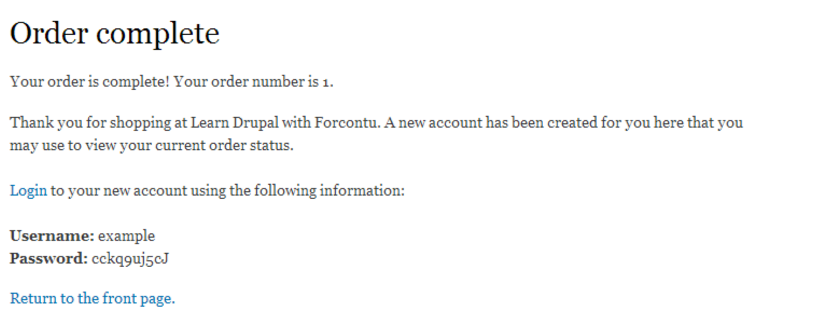
Back | Submit order

F36.41

Review order

In this step the review and acceptance of the order is requested, in order to make the purchase effective.

After completing the form, the order will be displayed so that the user can review it before finally submitting. By clicking on the **Submit order** button, the purchase will be completed. **F36.42**


Order complete

Your order is complete! Your order number is 1.

Thank you for shopping at Learn Drupal with Forcontu. A new account has been created for you here that you may use to view your current order status.

Login to your new account using the following information:

Username: example
Password: cckq9uj5cJ

Return to the front page.

F36.42

Order complete

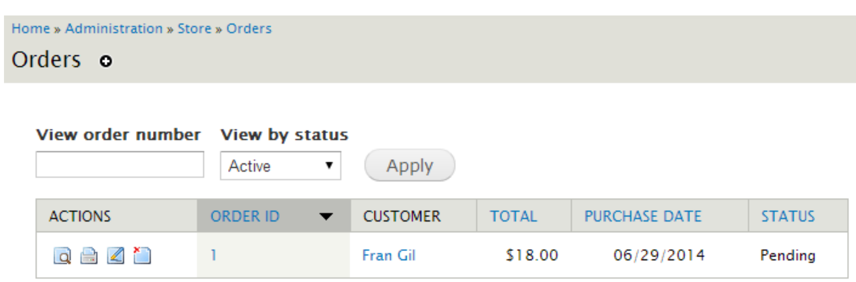
Order completion message.

Order management

Orders placed are displayed in the order management area: **F36.43**

Administration ⇨ Store ⇨ Orders ⇨ View orders

URL Orders
/admin/store/orders/view


Home » Administration » Store » Orders

Orders

View order number | **View by status** Active | Apply

ACTIONS	ORDER ID	CUSTOMER	TOTAL	PURCHASE DATE	STATUS
	1	Fran Gil	$18.00	06/29/2014	Pending

F36.43

Order management

The store administrator can manage the orders from View Orders.

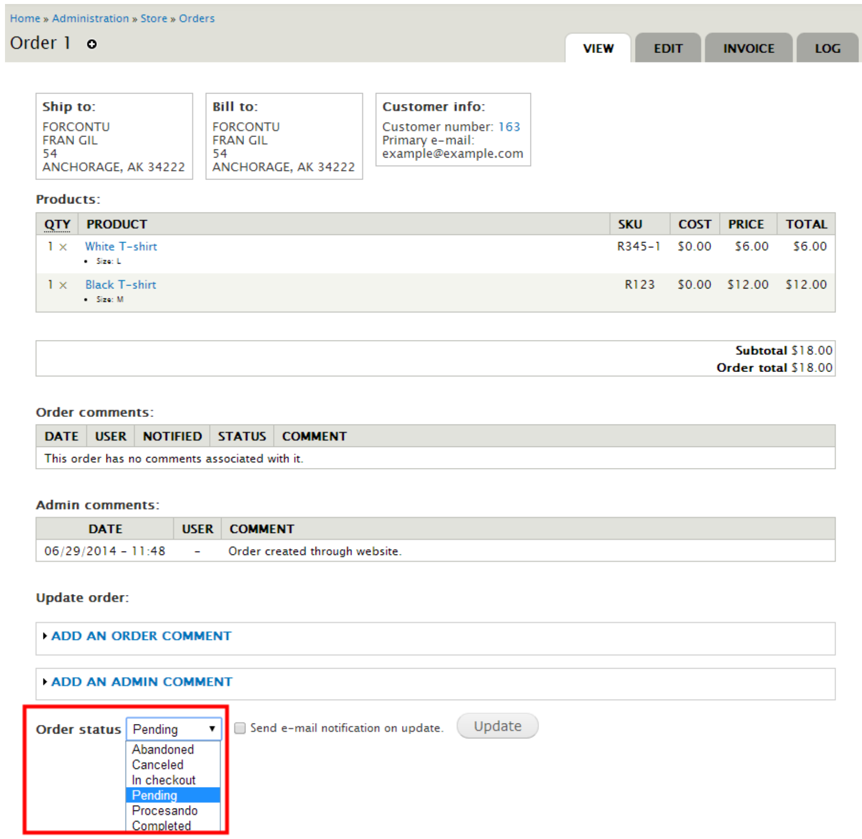
For each order the current status is displayed, an administrator can change it. You can also add comments to the order, both visible to the user and management reviews.

It is possible to send an email to the customer reporting the update of the order (this action is done through a rule from the Rules module). F36.44

F36.44

Edit order

From the order edition page we can make changes on it. A usual change is to update order status when appropriate.



Ubercart: Conditional actions

36.5

In Drupal 7 Ubercart uses the **Rules** module to create rules with conditional actions.

Examples of conditional actions could be:

- To charge certain taxes to certain users and not to others.
- Allow for shipping costs to not be added to certain purchases.
- Decide what to do with orders once payment has been made.
- Send email to the customer or administrator when an order is placed.

By accessing the Rules administration we can configure the rules added by Ubercart or create new ones using the events, conditions and specific actions to manage the store.

Administration ⇨ Configuration ⇨ Workflow ⇨ Rules

URL Rules
/ admin/config/workflow/rules

New **events** are grouped into categories:

- **Order**:
  - Order status gets updated.
  - E-mail requested for order status update.
- **Shopping cart**:
  - Customer completes checkout.

From **conditions** we can use those related to the order:

- **Order**:
  - Check an order's state.
  - Check if an order can be shipped.

- **Order: Product**:
  - Check an order´s total weight.
  - Order has a product with a particular attribute option.
  - Check an order´s number of products.
  - Check order´s products.

Finally we can use, besides the generic actions of Rules, the following specific actions of Ubercart:

- **Order**:
  - Update the order status.
  - Add a comment to the order.
  - Email an order invoice.
  - Send an order email.

36.6 Ubercart: Other functionality

To end this introduction to Ubercart, we present other features which we can access by activating kernel modules or installing additional modules.

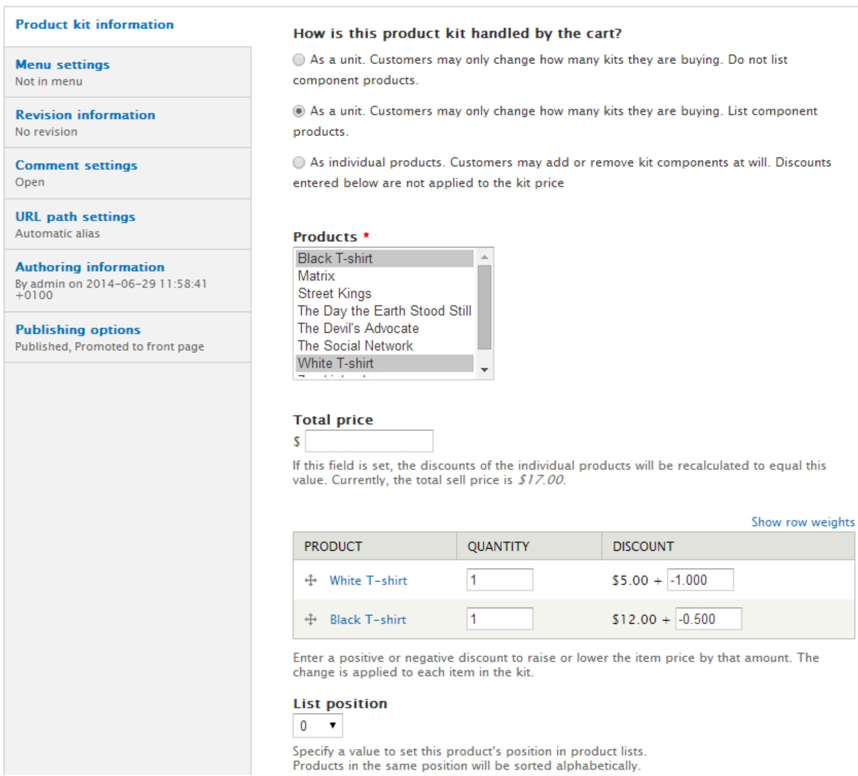
Product kit

Activating the **Product Kit** module (Ubercart - Extra), a new content type called **Product Kit** is added, which can create a package (pack or kit) with two or more products. Additionally, this package may have a different price than the sum of the individual products, usually to apply a discount. F36.45

F36.45

Product kit

The Product Kit module allows to create packages of products with a single price.



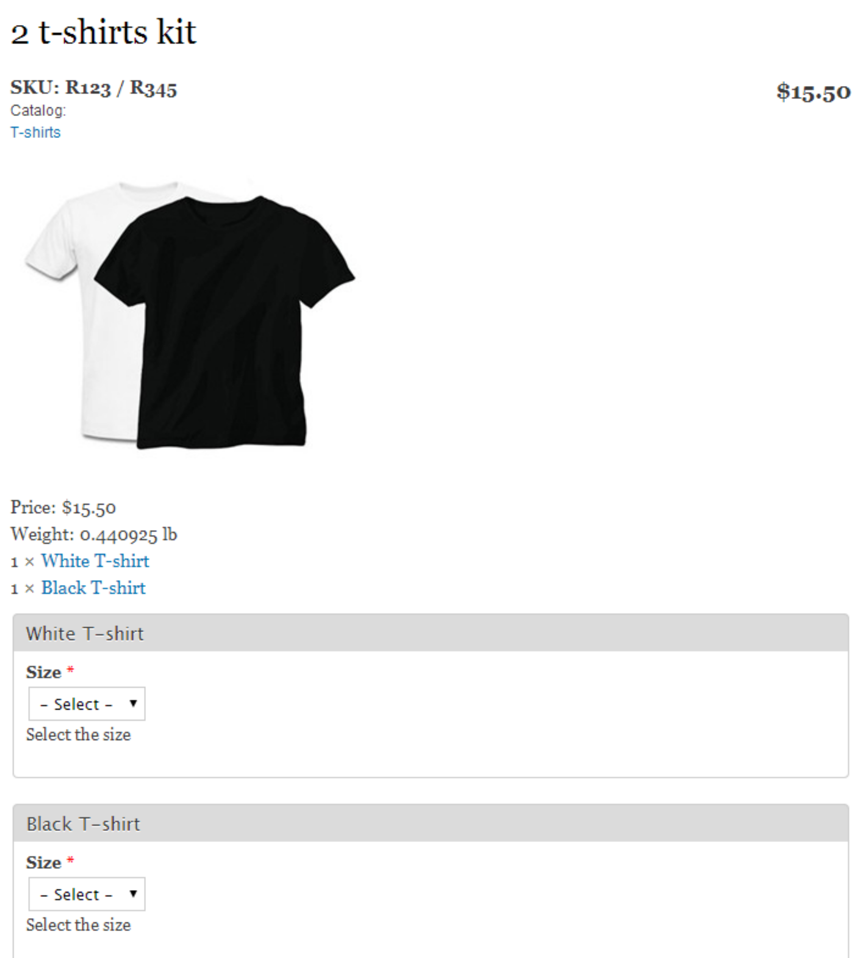
The package is shown as a single item in the assigned catalog category. It is recommended to use an appropiate image to represent the pack. F36.46

F36.46

Product kit

The package is shown as a single product at the store.

It is not needed to configure the attributes of each of the products in the package, since these values have been set individually for each of the included products. For example, in a package that includes two shirts, the customer can select the size of each shirt individually, so he can buy two shirts with different sizes. F36.47



F36.47

Product kit

Individual selection of attributes for each product in the bundle is allowed. Package contents and access to the individual description of each product is also shown.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Selling files

Ubercart allows the sale of digital files via the **File Downloads** module (Ubercart - core optional). A product may have an associated file, so that when a customer buys the product, he will be given permission to download the file.

Once the module is installed we can configure it from:

URL Products configuration
/admin/store/settings/products

Administration ⇨ Store ⇨ Configuration ⇨ Products ⇨ File download settings

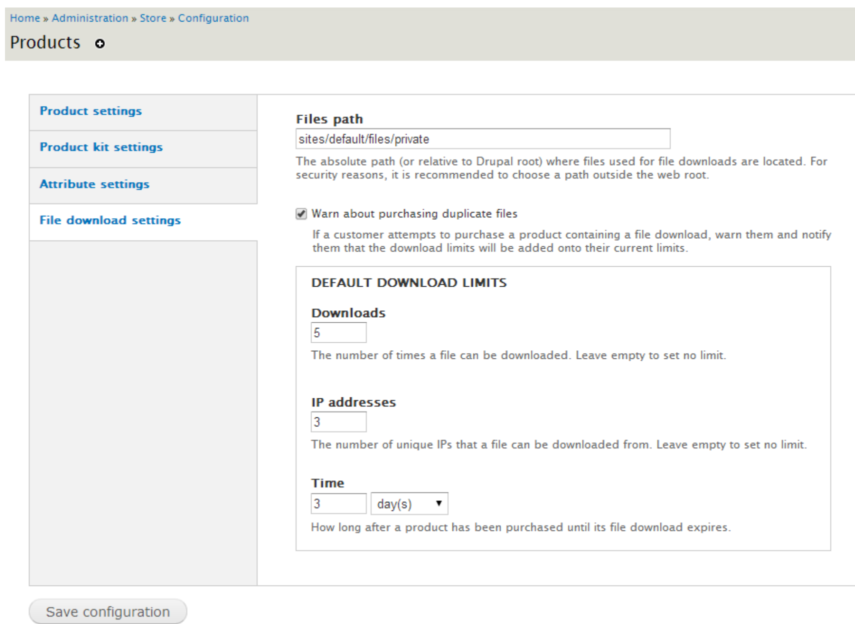
We must necessarily set the path where the files will reside. It is recommended that the directory be outside the root of the website, to avoid a possible direct access to the files. You can also set default values for limiting downloads at: F36.48

- Downloads.
- IP addresses that allow downloads.
- Expiry time for the download.

F36.48

Selling files

File download settings. We can set the path where the files are stored and a time limit for downloads, IP and number of downloads.



The next step is to upload the files associated with the products. This is done from: F36.49

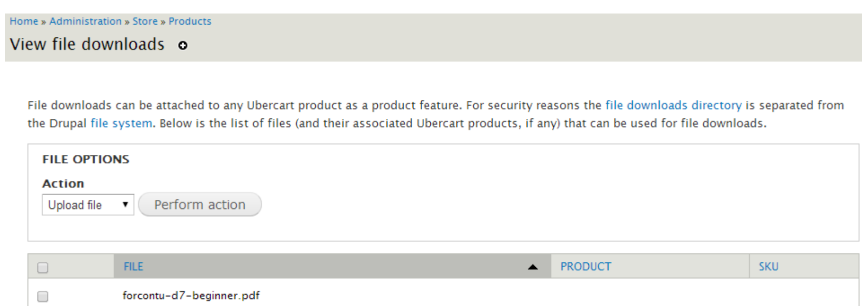
URL View file downloads
/ admin/store/products/files

Administration ⇨ Store ⇨ Products ⇨ View file downloads

F36.49

Upload files

From the Features tab of the product we can assign one (or more) downloads.

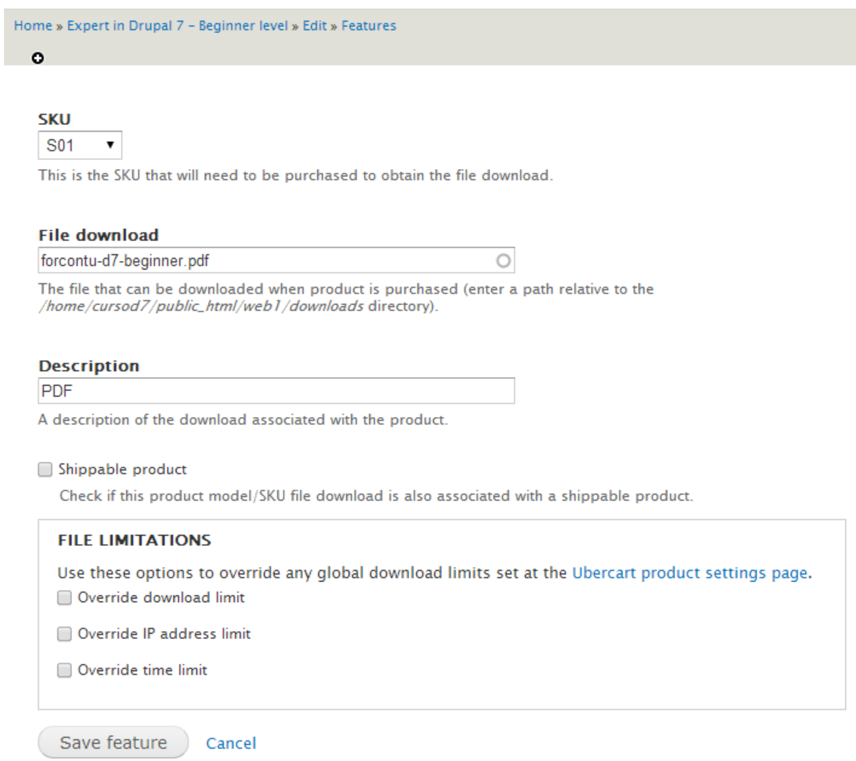


Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

The last step is to relate files in the repository with products. To do this, edit the related product in the store and click on the **Features** tab, then add the associated files (a product can have multiple associated files). F36.50

F36.50
Assign file to product
From the Features tab of the product we can assign one (or more) downloads.

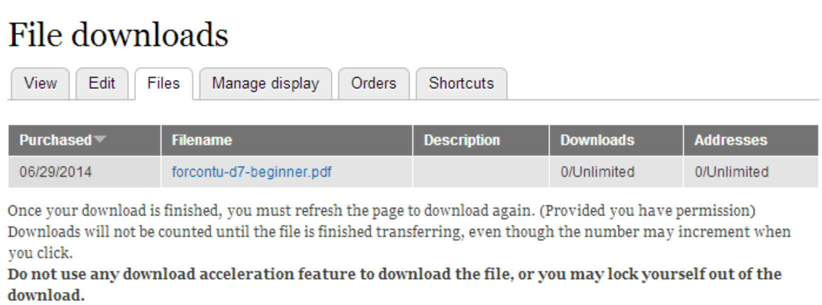
Upon adding a File Download feature, select the SKU of the product and the associated file. Optionally, we can modify the limits set by default for file downloads. F36.51



F36.51
Assign file to product
Associate file to download with the product.

A product can be associated with multiple files.

Once a file is associated to a product, it is ready for purchase. Customers who successfully complete the purchase process can access files for download from the Files tab included in their user profile (My account). F36.52



F36.52
File download
Users may download files from the Files tab of their profile.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Selling access to private areas

With Ubercart we can allow customers to access private areas or exclusive content via the **Roles** module (Ubercart - core optional). This module allows you to associate roles to users based on products purchased. Furthermore, these roles may be permanent or temporary.

Once installed the module can be configured at: F36.53

URL Products configuration
/admin/store/settings/products

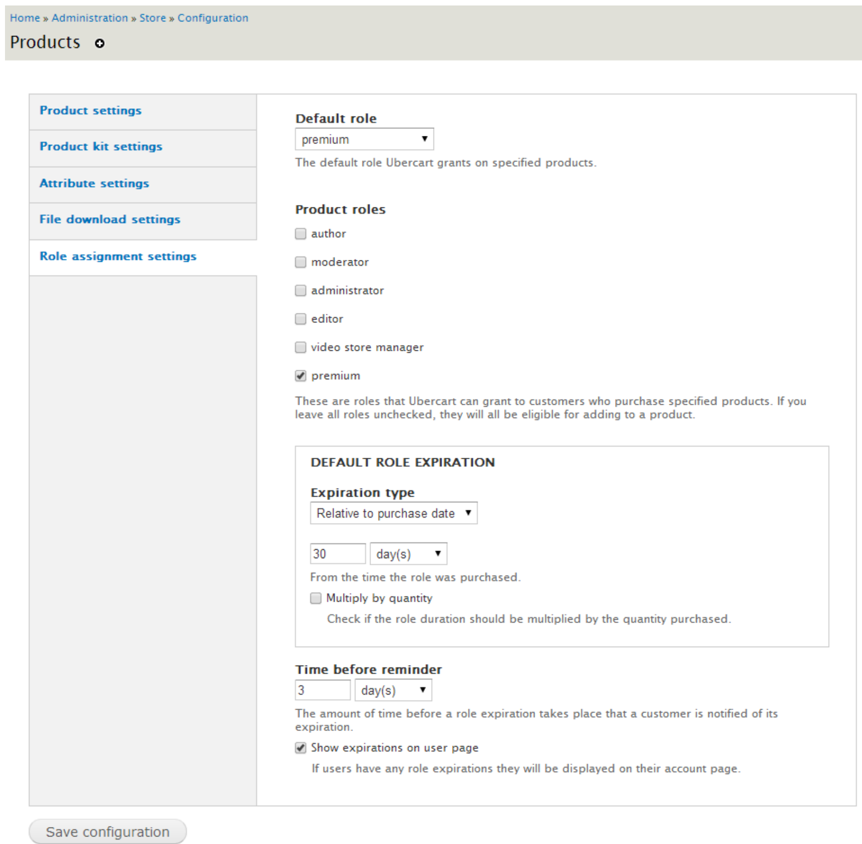
Administration ⇨ Store ⇨ Configuration ⇨ Products ⇨ Role assignment settings

F36.53

Access to private areas

The Roles module allows for the sale of user roles or sales of access to private areas or functionalities, which is the same.

The module allows the assignment of permanent or temporary roles.



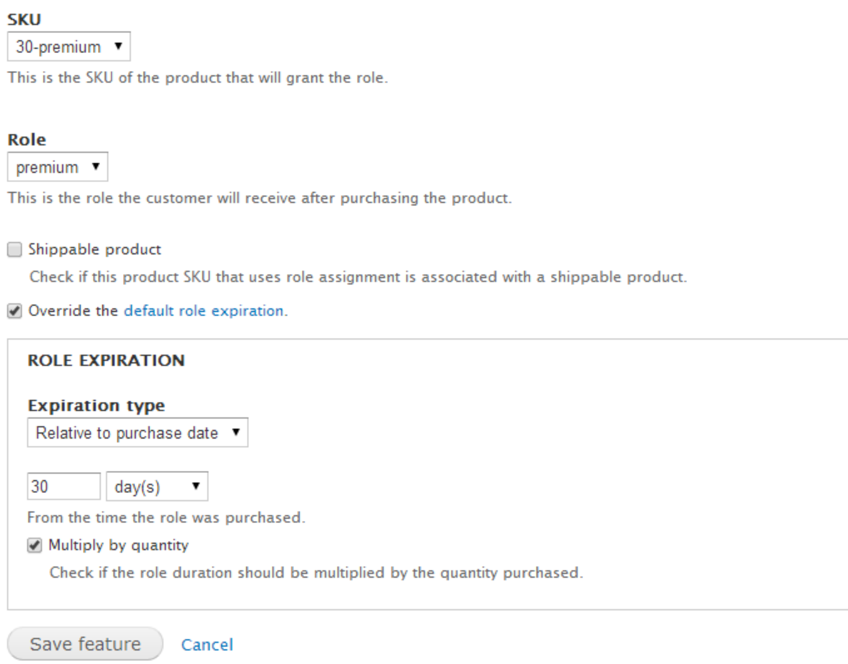
Available options are:

- **Default Role**. In this section the default role associated with products is described.
- **Products roles**. In this section we select roles that Ubercart can assign to customers who buy specific products. If no role is selected then all of them will be available.
- **Expiration**.
    - **Expiration type.** It can be a fixed date or relative to purchase date.
    - **Expiration time**. We can define an expiration time either absolute or relative to the purchase date. Important: the cron system must be configured correctly to enforce the forfeiture of assigned roles.
    - **Time before reminder**. You can set a deadline to notify

customers that expiry date is close.

- **Show expirations on user page**.

The last step is to add role assignment as a feature of the product. To do this, edit the item and click on the **Features** tab, then select **Role assignment**. F36.54



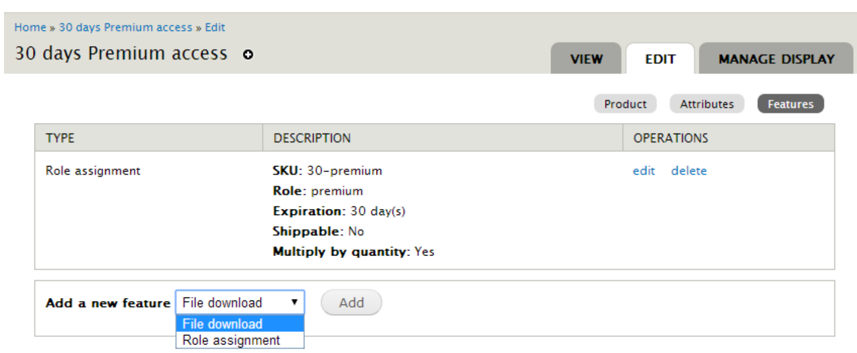
F36.54

Assignment of roles to product

The role assignment is done from the Properties tab of the product, selecting Role Assignment.

A product may have multiple role assignments.

Roles associated with the product will be applied to the customer upon completion of the order. F36.55



F36.55

Product features

The figure shows an example of a product where the premium role has been assigned. By buying the product the user will automatically receive the role for a total of 30 days.

Payment methods

The Payment module (Ubercart - core optional) defines the necessary APIs for payment modules to interact with the shopping cart.

The Payment module must be installed along with other modules, depending on the payment methods we want to add to the site. Some of the modules included (Ubercart - Payment) are:

- **Credit card**. Credit card payments.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Google Checkout**. Checkout managed by Google
- **Payment method pack**. Package with multiple payment methods, such as check or money order, COD (cash on delivery) and others.
- **Paypal**. The Paypal payment method.

Once **Payment** and additional modules are installed they can be configured from: F36.56

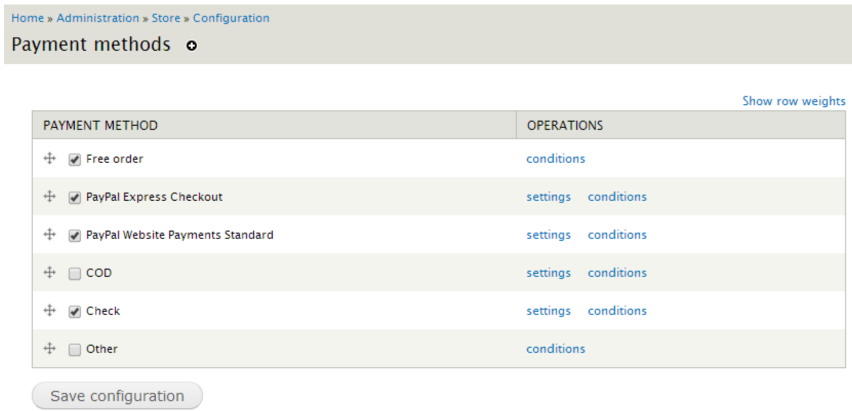
URL Payment methods
/admin/store/settings/payment

Administration ⇨ Store ⇨ Configuration ⇨ Payment methods

F36.56

Payment methods

Available payment methods available depend on ad-hoc installed modules.

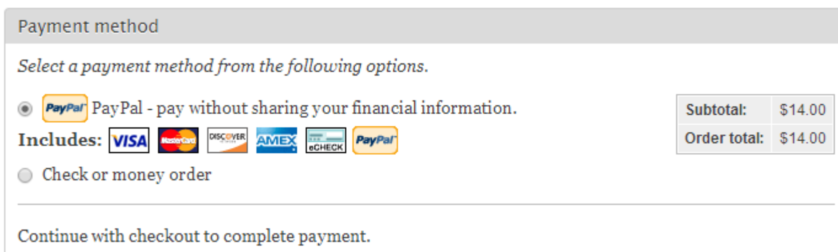


From now on to complete the order the user must select a method of payment. F36.57

F36.57

Payment request

The customer can choose the payment method from those available, which can be onsite or external.



Note: For payment methods associated with payment gateways (Paypal, Google checkout, etc.) to be available, it is necessary to contract these services and follow their setup instructions.

Stock control

The **Stock** module (Ubercart - Extra), offers a set of basic functions for stock control. The administration area of the module is available at:

URL Stock notifications
/admin/store/orders/view

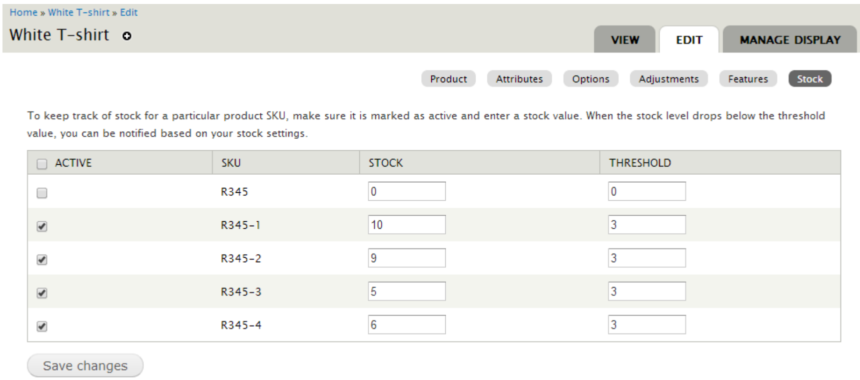
Administration ⇨ Store ⇨ Configuration ⇨ Stock notifications

From here you can activate email notification for **stock breakage**, that is, when a product reaches the threshold for minimum stock.

Products will now have a new **Stock** tab, from which you can access a list of all models associated with the product, allowing: F36.58

- **The activation of stock tracking**.

- **Defining of present stock**. This figure is usually updated when new stock of this product is received.
- **Define a threshold or minimum stock**. When this threshold is reached an email notification is sent. Normally notices involving stock breakage result in a purchase to the supplier in order to replenish stock.



F36.58
Stock
With the Stock module we can define inventory values.

Each time a customer makes a purchase the stock of purchased products will automatically decrease.

Taxation

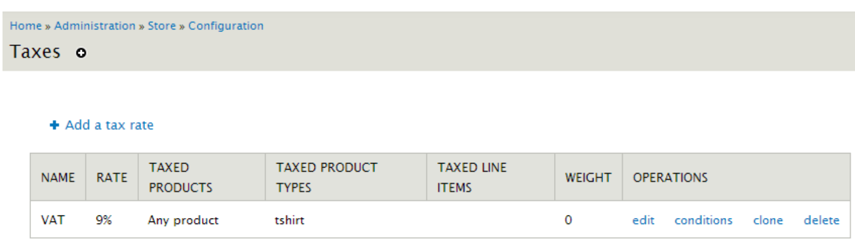
The **Taxes** and **Tax report** modules (Ubercart - core optional) allow for taxation of orders placed, depending on the products.

Tax rates are managed from:

Administration ⇨ Store ⇨ Configuration ⇨ Taxes

URL Taxes
/admin/store/settings/taxes

Upon creating a tax condition we can also configure to which products it applies, we add other conditions via the Rules module. F36.59



F36.59
Taxes
Configuration of products' taxes.

Shipping and handling costs

The **Shipping** and **Shipping quotes** modules (Ubercart - core optional) add shipping methods and estimated shipping costs to the order, according to certain formulas (total weight, number of articles, etc.). Apart from these modules we need to install additional modules related to the shipping system. Some of these modules are:

- **Flat rate**. Adds a flat fee per order, plus an additional amount per product.
- **Weight quote**. Calculates shipping costs based on total weight of the order.

We can create different shipping rates, with specific conditions from. F36.60

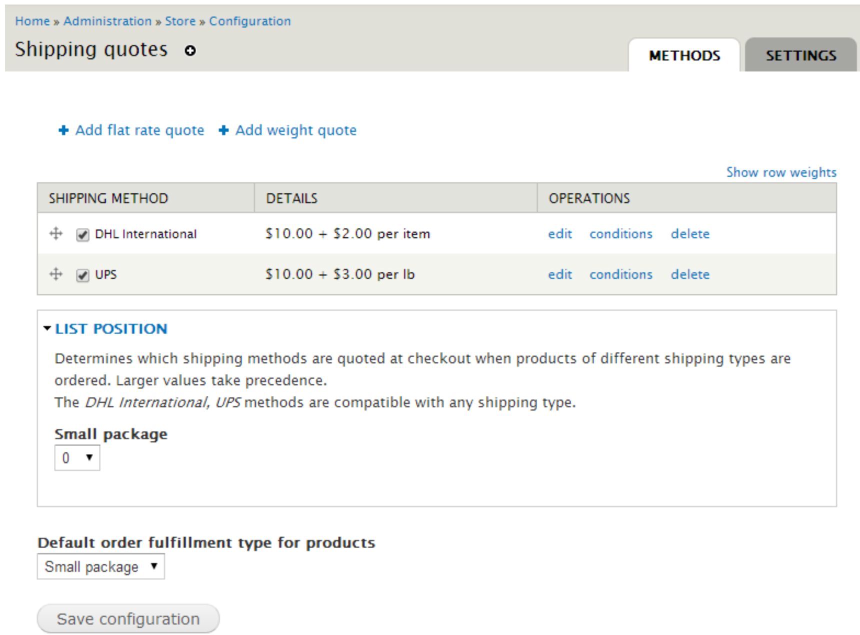
URL Shipping quotes
/admin/store/settings/quotes

Administration ⇨ Store ⇨ Configuration ⇨ Shipping quotes

F36.60

Shipping and handling

Through additional modules we can add different methods for calculating shipping and handling charges. Calculations can be performed for weight, number of items, etc.



Discount coupons

The **Ubercart Discount Coupons** module incorporates discount coupons in the purchase process. These discounts can be fixed amounts or percentages, furthermore they can be associated with a single product or any product from the store.

The Ubercart Discount Coupons is available at:

http://drupal.org/project/uc_coupon

The module consists of three modules (**Discount Coupons**, **Discount Coupon Purchase**, and **Discount Coupon Extended Workflow**), all of them necessary to incorporate into the store a complete system of coupons.

The module has an associated section for the generation of discount coupons at:

URL Coupons
/admin/store/coupons

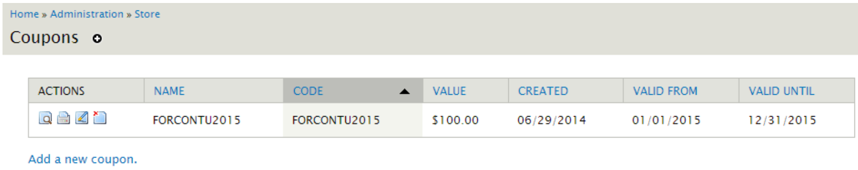
Administration ⇨ Store ⇨ Coupons

From here we can create new coupons and view existing coupons. When creating a coupon we can set many parameters: F36.61

- Generate bulk coupon codes.
- Set value and discount type (fixed amount, percentage, etc.).
- Apply the discount to the entire order, only to certain products, etc.
- Restrict coupon by dates.
- Number of times you can redeem the code, both overall and per user.
- Roles that may use the coupon.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- Products for which the coupon can be applied.
- and a long etc.

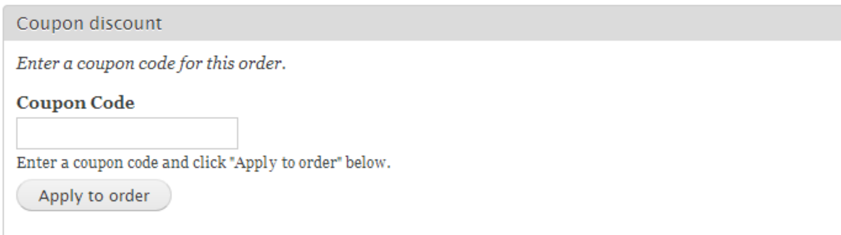


F36.61

Discount coupons

The Discount Coupons modules are a complete solution for managing and applying discount coupons.

Upon purchase completion, the customer can enter the discount code and, if it is valid, the appropriate discount will be applied to the order. F36.62



F36.62

Discount coupons

The coupon must be applied before checkout. The price of the order will be updated depending on the discount applied.

Integration with Views

The Ubercart module for Drupal 7 integrates with the **Views** module without the need to install any additional modules.

Ubercart can create views where the main element is a **product** (product content type), an **order** (Orders type), **items added to the shopping cart** (Cart items type) or **products included in an order** (Ordered products type).

Moreover, depending on the type of view, you can access the fields related to the element, include **Add to cart** buttons, etc.

36.7 The Drupal Commerce module

Drupal Commerce is another alternative to build a virtual store in Drupal. It is only available for Drupal 7, so this makes for a more optimized way to use new functionality in Drupal 7, like entities and associated fields.

The main differences regarding Ubercart make Drupal Commerce much more flexible:

- As already stated, Commerce has been designed for Drupal 7 from the ground up, so it is optimized for its new functionality like entities and their associated fields.
- Intensive use of the **Rules** module. Drupal Commerce is based on the use of Rules to configure payment methods, calculate taxes, email, etc. Therefore a good knowledge of Rules is required.
- Intensive use of the **Views** module. The use of views allows us to easily configure the order pages, shopping cart, product listings, etc.

While you can say that Drupal Commerce is much more flexible than Ubercart, it is also much more complex to set up. Its great flexibility is especially appreciated on websites which require the configuration of multiple caseloads.

We recommend you use one of the two e-commerce solutions, Ubercart or Drupal Commerce. Therefore, before proceeding with the installation of Drupal Commerce deactivate all Ubercart modules.

The Drupal Commerce is available at:

http://drupal.org/project/commerce

Drupal Commerce depends on the following modules:

- **Rules**, available at: http://drupal.org/project/rules
- **Views**, available at: http://drupal.org/project/views
- **Ctools**, available at: http://drupal.org/project/ctools
- **Entity API**, available at: http://drupal.org/project/entity
- **Address Field**, available at: http://drupal.org/project/addressfield

Modules included in Drupal Commerce are grouped under **Commerce**:

- **Cart**. Controls operation of the shopping cart.
- **Checkout**. Controls checkout.
- **Commerce**. The main module, on which all the rest depend.
- **Commerce UI**. Defines menu elements related to the virtual store.
- **Customer**. Defines the new entity Customer.
- **Customer UI**. Interface for customer management.
- **Line Item**. Line items are a new entity introduced by Commerce, they are anything on an order affecting the order total (e.g. products, taxes, shipping, discounts, fees, etc.). Line items will have a serial numeric line item ID (line_id), a reference to the appropriate order (order_id), a type, a merchant-defined ID (i.e. the product SKU, a tax rate name, a coupon code, etc.), a display title (fully editable on the order), a quantity, and a price field.
- **Line Item UI**. Interface to manage line items.

- **Order**. Creates an Order entity and it associated characteristics.
- **Order UI**. Interface to manage orders.
- **Payment**. Core functionalities for payment methods.
- **Payment Method Example**. Example module to implement other modules for additional payment methods.
- **Payment UI**. Interface to manage payment methods.
- **Price**. Defines the Price field and associated functionalities.
- **Product**. Defines the Product entity and associated functionalities.
- **Product Pricing**. Allows for dynamic calculation of prices via Rules.
- **Product Pricing UI**. Interface for managing rules for price calculation.
- **Product Reference**. Defines a reference field for the product.
- **Product UI**. Interface for product management.
- **Tax**. Defines rules for taxation.
- **Tax UI**. Interface for the creation and management of tax rates.

Besides the drupal.org repository for modules, additional modules that complement Drupal Commerce can be found at:

http://www.drupalcommerce.org/contrib

Many of these modules are still under development, so it will be necessary to test them in our testing environment before they go into production.

Next we will activate all Commerce core modules.

Installation profile

A simple way to install and configure Drupal Commerce is to use the Commerce Kickstart installation profile, available at:

http://drupal.org/project/commerce_kickstart

The installation profile is used to install a Drupal Commerce website from scratch, including the Commerce module and its dependencies. The profile creates the basis of the store and adds examples of the different elements that compose it, thus avoiding having to manually configure each option and, above all, we will better understand the operation of Drupal Commerce.

In this unit we will not install the installation profile, but briefly discuss the options available when installing Drupal Commerce on an existing website.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

36.8 Commerce: Administration and configuration of the store

Once modules are installed, you can access the administration area of the store from:

URL Store
/admin/commerce

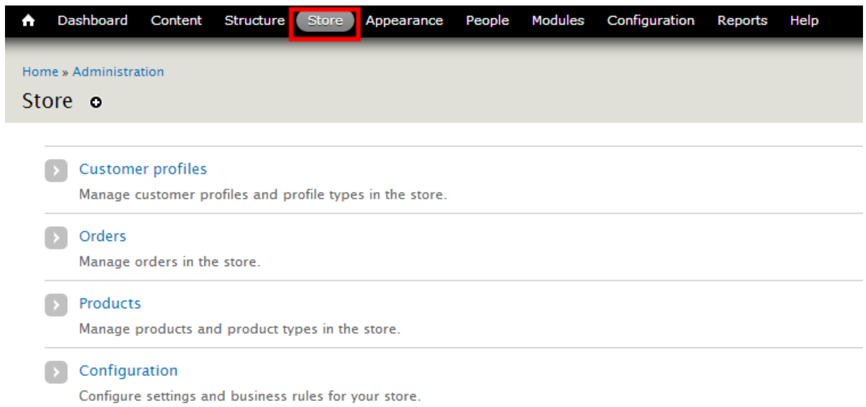
Administration ⇨ Store

The administration area of the store is initially divided into groups of options: **Orders, Customer Profiles, Products** and **Configuration**. These groups and their included options may be increased depending on the installation of additional modules. F36.63

F36.63

Store

The basic installation of Drupal Commerce implies an area for the complete management of an online store, from the Store option of the Administration menu.



Each group contains a set of options for performing different actions:

- **Customer profiles**. Management of profile types and existing profiles.
- **Orders**. Order management.
- **Products**. Management of product types and existing products.
- **Configuration**. Includes several configuration options:
    - Payment methods.
    - Taxes.
    - Currency settings.
    - Checkout settings.
    - Order settings.
    - Product pricing rules.
    - Line ítem types.

Next we visit the various options available in the basic installation of **Commerce**. We begin with options in the **Configuration** section.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Payment methods

Payment methods are configured from:

Administration ⇨ Store ⇨ Configuration ⇨ Payment methods

URL Payment methods
/admin/commerce/config/payment-methods

Depending on the payment methods we want to use on the site, we will have to install additional modules available at:

http://www.drupalcommerce.org/contrib/payment

Some payment modules available are:

- **Commerce Bank Transfer**. Payment by bank transfer.

  Available at:
  http://drupal.org/project/commerce_bank_transfer

- **Commerce PayPal**. Payment via Paypal.

  Available at:
  http://drupal.org/project/commerce_paypal

The configuration of payment modules is done via Rules. The module has the following **event** to be triggered when selecting a method of payment before submitting an order:

- **A payment gets entered for an order**.

Installed payment methods will be available as **actions**:

- **Enable payment method: Example payment**. This is an example action, enabled by default. It serves as an example to set other payment methods or to develop through programming custom payment methods.

- **Enable payment method: Bank Transfer**.

- **Enable payment method: PayPal WPP – Credit Card**. Enables Paypal's Website Payments Pro (WPP). Allows for credit card payment via Paypal.

- **Enable payment method: PayPal WPS**. Enables PayPal's Website Payments Standard (WPS) method.

For each enabled payment method, needed individual configuration parameters will be requested.

As we can also add conditions to each defined rule, this system is very flexible, allowing to install different variants of the same payment method which are activated under different conditions. For example, it is possible to activate under different conditions several bank accounts for bank transfer payments or several PayPal accounts, which could be necessary for receiving micropayments.

Taxes

From the Taxes section we can configure the types of tax rates and for each type, the applicable percentages.

URL Taxes
/admin/commerce/config/taxes

Administration ⇨ Store ⇨ Configuration ⇨ Taxes

Tax rates are also configured through rules. By default two **tax types** are defined: **Sales tax** (sales tax system used in the U.S.) and **VAT** (value added tax).

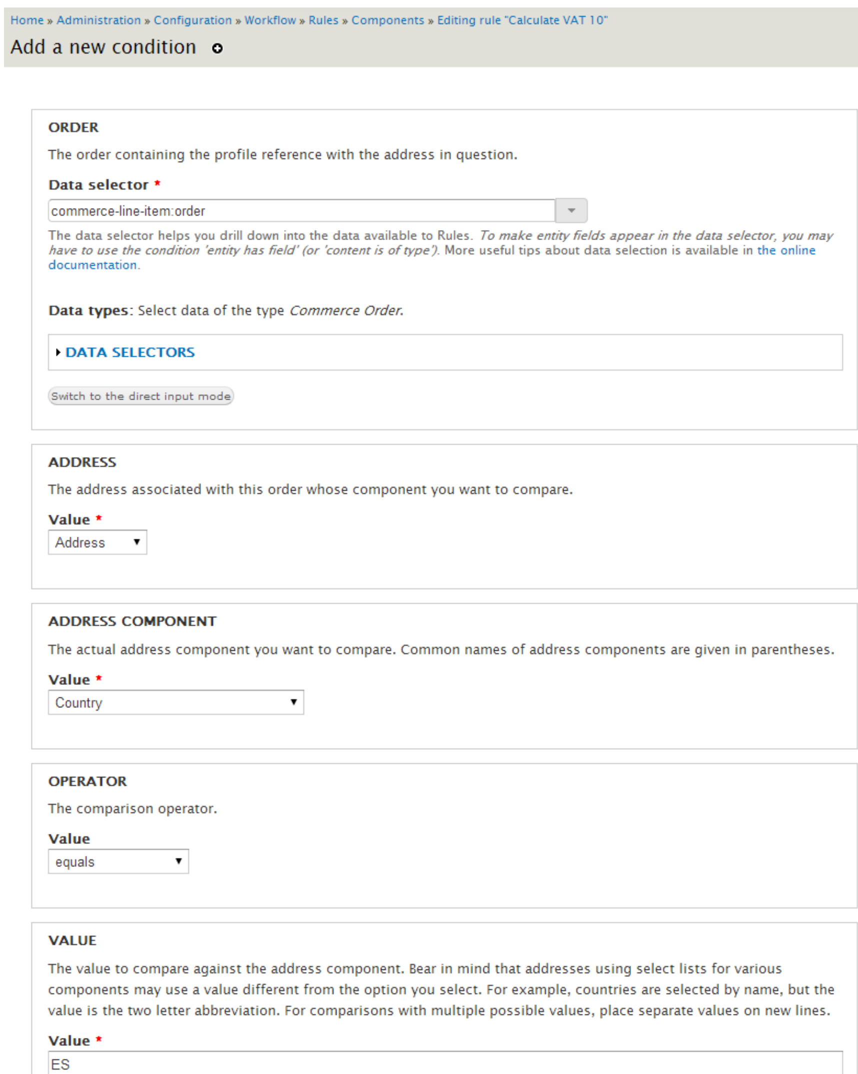
From the **Tax Rates** tab we can create tax rates, indicating the type and percentage rate, in addition to the special conditions required. In this case rates are defined as components of Rules, without an associated event, which will be evaluated by the main rule defined in the rate type.

Generally rates applied may depend on the customer's address. When appropriate, we will create different rates for each country (or state/province), and make a comparison with the address entered by the customer, adding the **Order address component comparison** condition. F36.64

F36.64

Taxes

Tax configuration is done from Rules, using events, conditions and actions specific to the module, combined with other options available in Rules.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

When comparing countries use the 2-digit code corresponding to the country (US, ES, etc.). As already discussed, this system through rules may seem complex, but provides a flexibility that cannot always be achieved with other solutions like Ubercart.

Currency settings

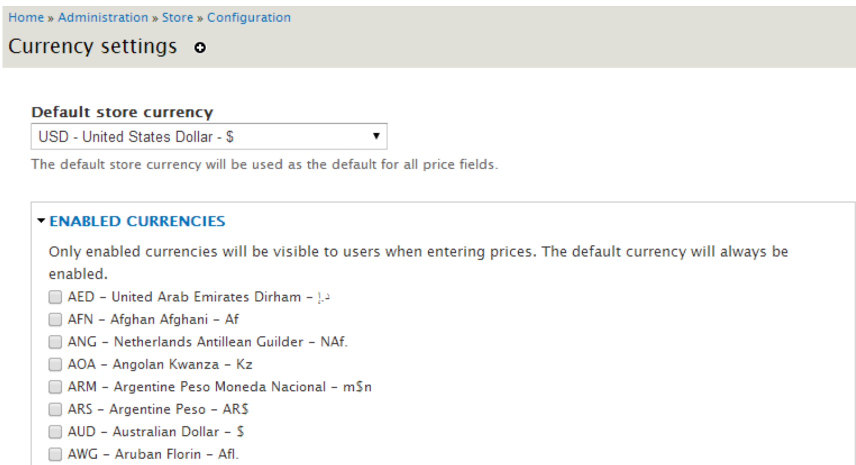
To enable different types of currency we go to:

Administration ⇨ Store ⇨ Configuration ⇨ Currency settings

URL Currency settings
/admin/commerce/config/currency

Only activated currencies are visible. The default currency is always active, even if not select in the list of enabled currencies.

Since the U.S. dollar ($) is the default currency, we could convert it to the pertinent currency (eg. Euro - €). F36.65



F36.65
Currency settings
Enable which currencies will be available in the store.

Payment options

From payment options you can configure the checkout form and the rules that apply to complete a purchase. Configuration is done from:

Administration ⇨ Store ⇨ Configuration ⇨ Checkout settings

URL Checkout settings
/admin/commerce/config/checkout

From the **Checkout form** tab you can configure and rearrange the elements of the form. This form is divided into several steps: F36.66

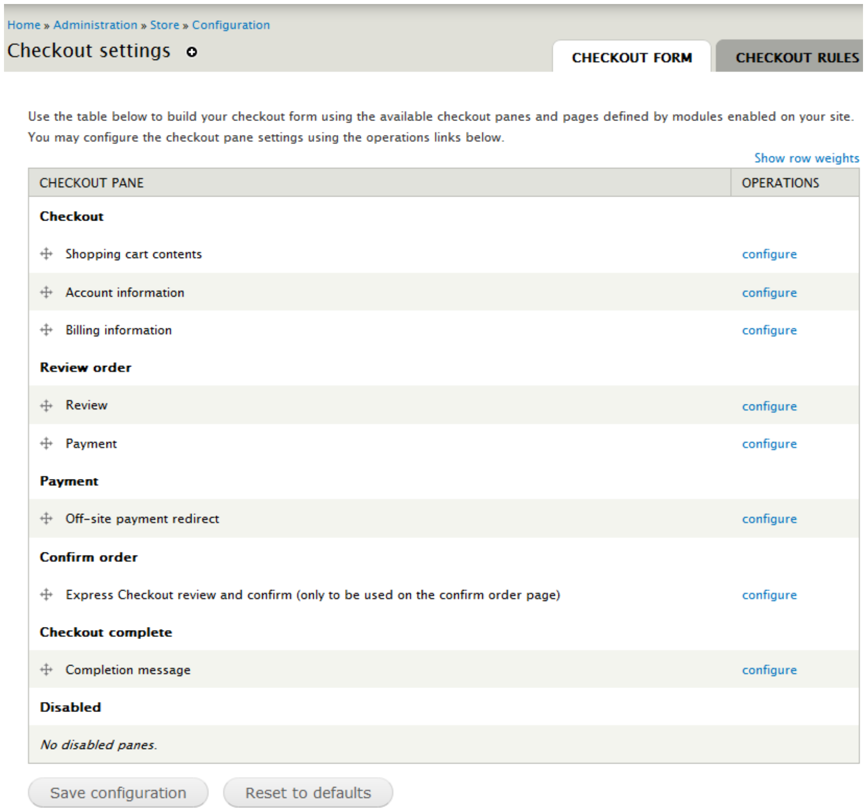
- **Checkout**:
  - Shopping cart contents
  - Account information (de usuario)
  - Billing information
- **Review order**:
  - Review
  - Payment (payment method selection)
- **Payment**:
  - Redirect to external payment site (if applicable)
- **Checkout complete**:
  - Completion message.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F36.66

Checkout settings

The checkout form consists of four steps, which can be configured from this page.

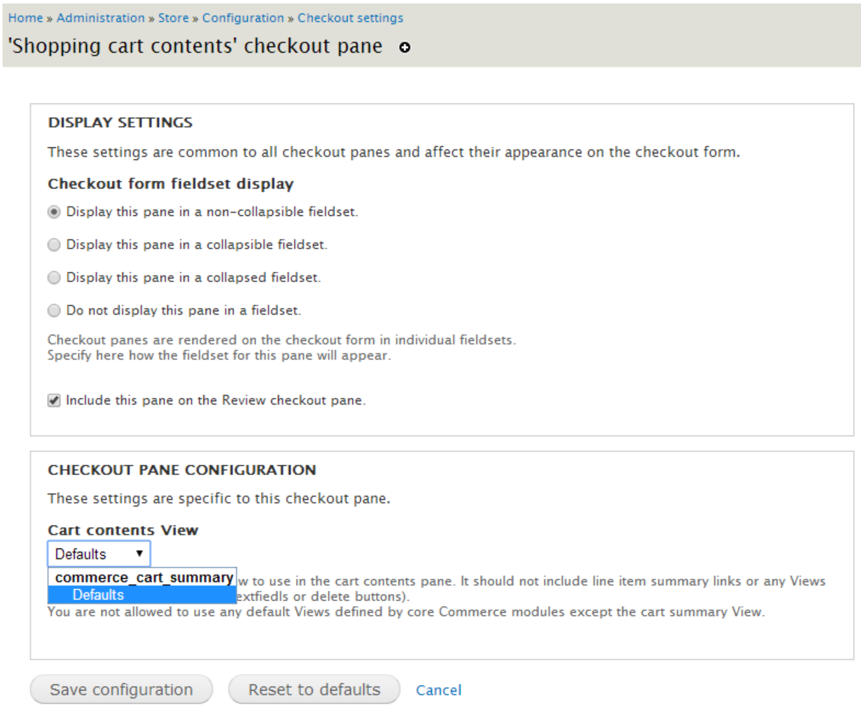


In the configuration of each element find **Display settings**, which are similar for all elements, and other specific options. **F36.67**

For example, in the **Cart contents** element you can select the view that displays the order items. By default, the commerce_cart_summary view is used. Modifying this view or selecting another view can modify the way in which this element is displayed.

Along with rules, the use of views promotes flexibility in the construction and configuration of the online store.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



F36.67

Cart contents

The shopping cart can be configured by modifying the view added by the module.

From the **Checkout Rules** tab you can configure the rules to be evaluated (and run if applicable) at checkout.

Initially we find the following rules: **F36.68**

- **Update the order status on checkout completion**. Assigns pending status to the order.
- **Assign an anonymous order to a pre-existing user**. This rule checks the email address entered by the anonymous user against the address of a registered site user and, if it matches, assigns the order to that user.
- **Create a new account for an anonymous order**. If the email entered does not exist in the database, a user account is created for the customer.
- **Set an order notification e-mail**. Sends an email to the customer.

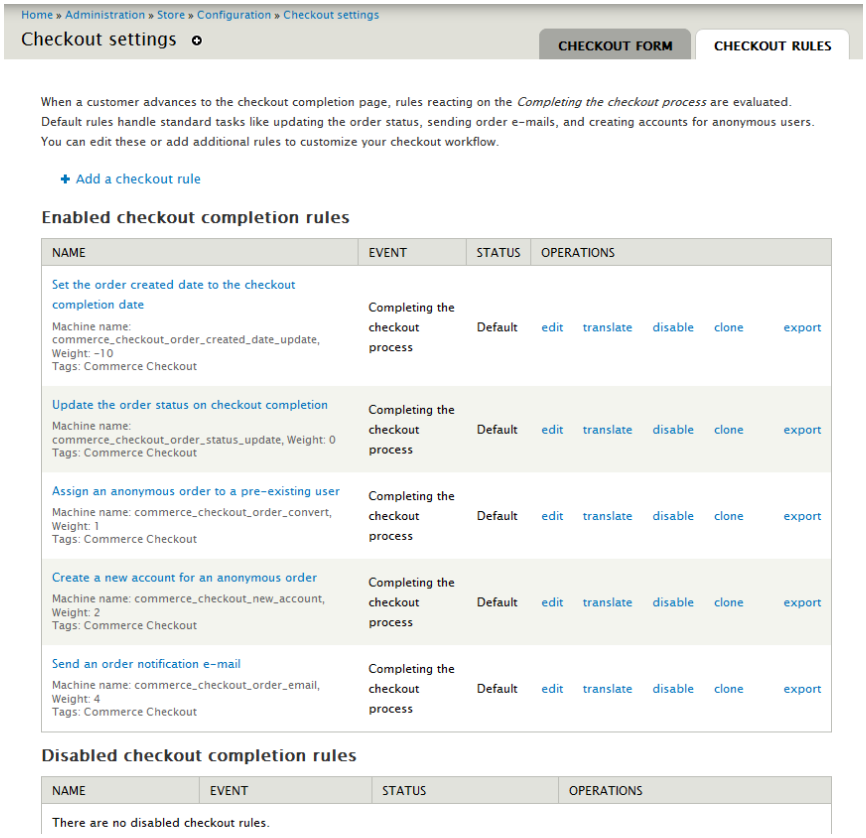
Again, the system allows you to add new rules or modify existing ones. We need only consider that the **event** that fires during the completion of the purchase is **Completing the checkout process**.

F36.68

Checkout rules

Includes all the rules defined by the module to be applied during the checkout process.

We may modify these rules or add additional ones, all through Rules.



Home » Administration » Store » Configuration » Checkout settings

Checkout settings

CHECKOUT FORM | CHECKOUT RULES

When a customer advances to the checkout completion page, rules reacting on the *Completing the checkout process* are evaluated. Default rules handle standard tasks like updating the order status, sending order e-mails, and creating accounts for anonymous users. You can edit these or add additional rules to customize your checkout workflow.

+ Add a checkout rule

Enabled checkout completion rules

NAME	EVENT	STATUS	OPERATIONS				
Set the order created date to the checkout completion date Machine name: commerce_checkout_order_created_date_update, Weight: -10 Tags: Commerce Checkout	Completing the checkout process	Default	edit	translate	disable	clone	export
Update the order status on checkout completion Machine name: commerce_checkout_order_status_update, Weight: 0 Tags: Commerce Checkout	Completing the checkout process	Default	edit	translate	disable	clone	export
Assign an anonymous order to a pre-existing user Machine name: commerce_checkout_order_convert, Weight: 1 Tags: Commerce Checkout	Completing the checkout process	Default	edit	translate	disable	clone	export
Create a new account for an anonymous order Machine name: commerce_checkout_new_account, Weight: 2 Tags: Commerce Checkout	Completing the checkout process	Default	edit	translate	disable	clone	export
Send an order notification e-mail Machine name: commerce_checkout_order_email, Weight: 4 Tags: Commerce Checkout	Completing the checkout process	Default	edit	translate	disable	clone	export

Disabled checkout completion rules

NAME	EVENT	STATUS	OPERATIONS
There are no disabled checkout rules.			

Order settings

In Drupal Commerce orders are entities. The main advantage is that we can manage fields and presentation as we have done with other entities (eg. content types).

The Drupal Commerce suite adds specific fields for the entities it creates, which also can be used in any other entity:

- Line item reference.
- Producto reference.
- Customer profile reference.
- Precio. Enables inclusion of price and currency type.
- Postal added (added by the **Address field** module).

From Order Options we can manage fields and display of the Order entity. We can also add a **help text** to be displayed in the order creation form.

Creating orders will be studied in **section 36.10.**

Product pricing rules

Product pricing rules allow calculations to get the final price of products ordered. Depending on installed modules, these rules allow applying discount coupons, currency conversions, tax, etc.

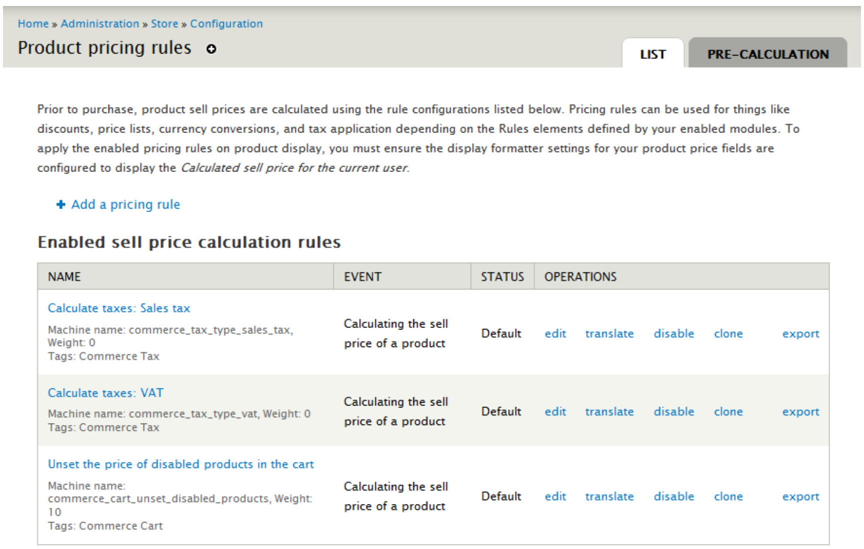
Configuration of product pricing rules is made from:

Administration ⇨ Store ⇨ Configuration ⇨ Product pricing rules

URL Product pricing rules
/admin/commerce/config/product-pricing

The initially available rules are: F36.69

- **Unset the Price of disabled products in the cart**.
- **Calculate taxes: (Sales tax)**. If this tax type will not be used this rule should be disabled.
- **Calculate taxes (VAT)**. If this tax type will not be used this rule should be disabled.



NAME	EVENT	STATUS	OPERATIONS				
Calculate taxes: Sales tax Machine name: commerce_tax_type_sales_tax, Weight: 0 Tags: Commerce Tax	Calculating the sell price of a product	Default	edit	translate	disable	clone	export
Calculate taxes: VAT Machine name: commerce_tax_type_vat, Weight: 0 Tags: Commerce Tax	Calculating the sell price of a product	Default	edit	translate	disable	clone	export
Unset the price of disabled products in the cart Machine name: commerce_cart_unset_disabled_products, Weight: 10 Tags: Commerce Cart	Calculating the sell price of a product	Default	edit	translate	disable	clone	export

F36.69

Product pricing rules

These rules allow recalculation of prices based on defined taxes, coupons, etc.

Line items

A line item is each product added to an order. A line item includes, in addition to the product reference (through its SKU), the amount of products ordered, the unit price and the total price.

Line items have also been defined as entities, so we can manage their fields and display. By default there is a line item type called Product, which lets you add a product to an order. We can not directly add other types of line items, but we can change the line item Product. We do so from:

Administration ⇨ Store ⇨ Configuration ⇨ Line items

URL Line items
/admin/commerce/config/line-items

36.9 Commerce: Products

Unlike Ubercart, where product types are built as content types, in Drupal Commerce product types constitute a new entity, thus avoiding the accidental mix of entities but enabling access to all the features associated with them, such as additional fields and presentation management.

In this section we will see how to create products and product types.

Product types

Product type management is available at:

URL Product types
/admin/commerce/products/types

Administration ⇨ Store ⇨ Products [Product types]

Drupal Commerce automatically adds a product type called **Product** (with machine name **product**). If we access **Manage fields** in this product type, we see that the entity has the following: F36.70

- **Product SKU**. Stans for Stock Keeping Unit, a unique reference to the product.
- **Title**. Name of product.
- **Price**. It is a field of type Price which lets you add in addition to price, currency and applicable tax rate.
- **Status**. Active or Inactive.

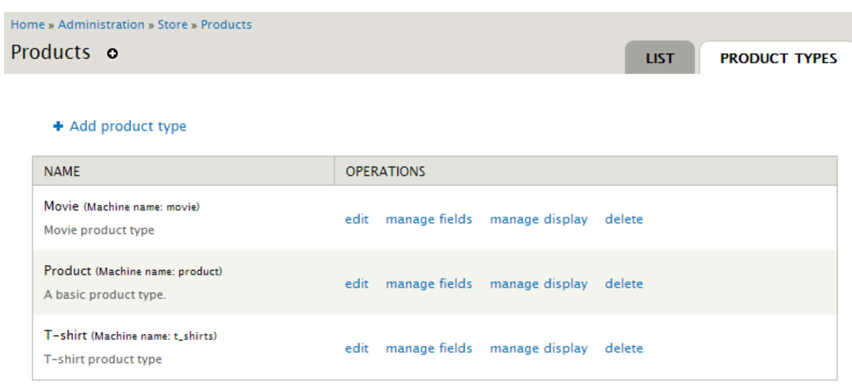
We can create new product types by clicking on **Add product type**. The new product type (T-shirts, books, movies, etc.) will initially have the same basic fields. Both in the Product and in the new types we can include additional fields, for example, images or any other product information.

Note: If upon creation of a new product type links to **manage fields** and **manage display** are not shown, check that the user (Administrator role) has the new permissions that have been generated. **Clear all caches before and after changing permissions.**

F36.70

Product types

Unlike Ubercart, in Drupal Commerce product types are not content types, but a new entity.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Product creation

To create products of any of the available types we need to access: F36.71

Administration ⇨ Store ⇨ Products [Add a product]

URL Add a product
/admin/commerce/products/add

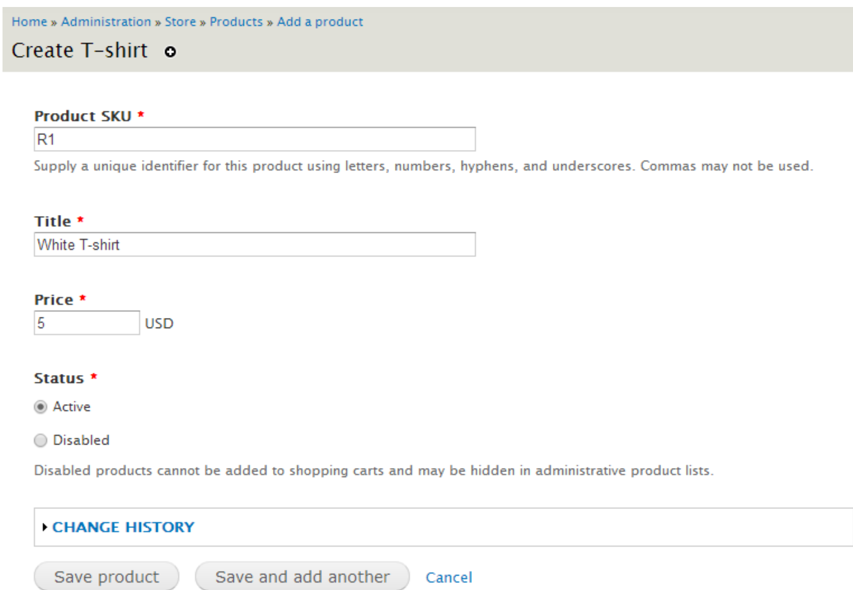
After selecting the product type, we will complete the edit form of the product.

F36.71

Add a product

To create a product first select the product type.

If product price includes taxes, these will not be recalculated during checkout. F36.72



F36.72

Add a product

The product may have specific fields, depending on the definition made in the product type.

Items added to an order cannot be deleted, only disabled so they will not be shown in product listings.

Product catalog

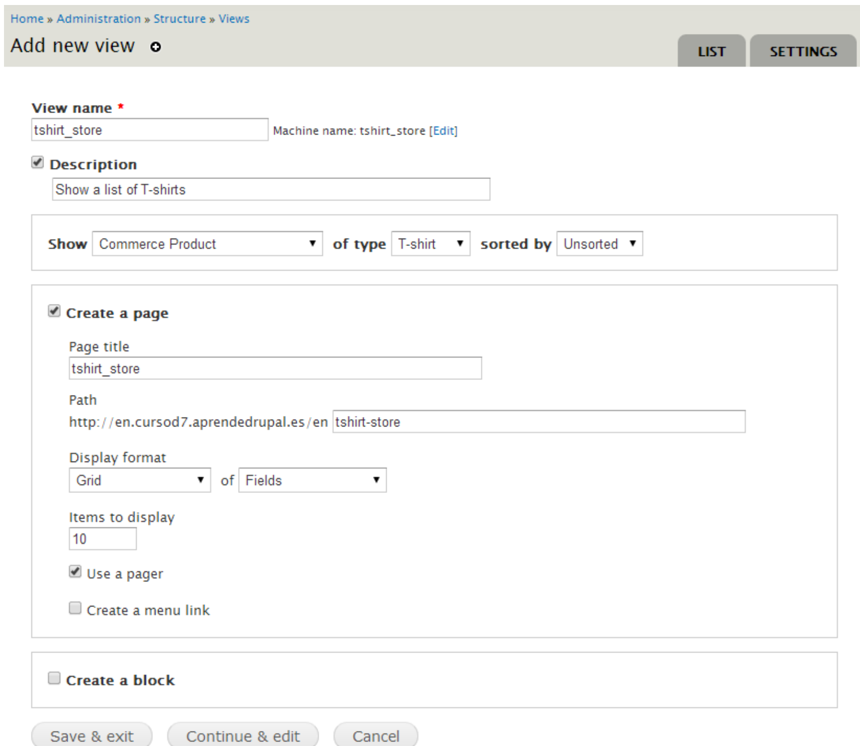
Drupal Commerce does not provide a catalog of products like Ubercart. To build a list of products we use Views.

When creating the view select the **Commerce Product** as the main element, specifying the type of product to be displayed. F36.73

F36.73

Product catalog

Drupal Commerce does not provide a catalog like Ubercart does. To build our list of products we must use Views.

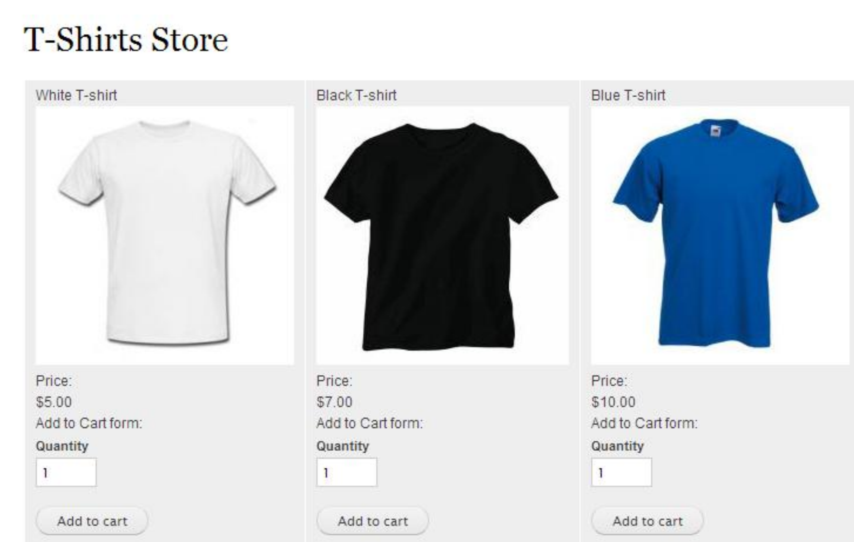


The **Figure** F36.74 shows a simple example of a view in grid format with the added fields:

F36.74

Product catalog

Example catalog created from a view.



- **Commerce product: Title**. Name of product.
- **Commerce product: Image**. New field we have previously added to the product type.

- **Commerce product: Price**. We have selected option **Formatted amount**.
- **Commerce product: Add to cart form**. We have selected option **Display a textfield quantity widget on the add to cart form**.

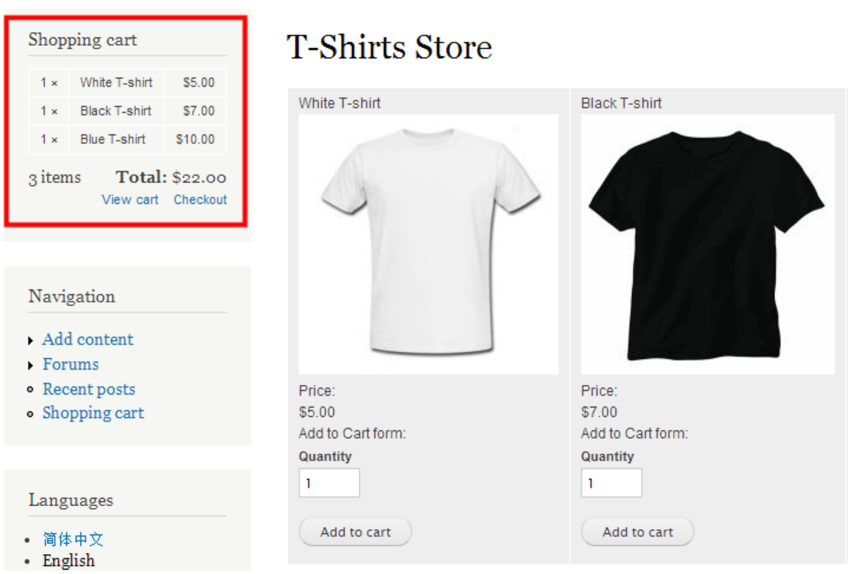
Commerce: Purchases and orders

36.10

Shopping cart Block

The **Shopping cart** block, initially disabled, shows the user-added products to the shopping cart.

The **Figure** F36.75 shows the Shopping cart block, with details of the items the customer has added.



F36.75
Shopping Cart block
The Shopping Cart block always shows the detail of added items, amount per product and total.

The purchase process

Next we go over the purchase process from the point of view of the customer.

Customer adds to shopping cart the items he intends to purchase by clicking the **Add to cart** buttons to be found in both the product page and in the catalog (custom views).

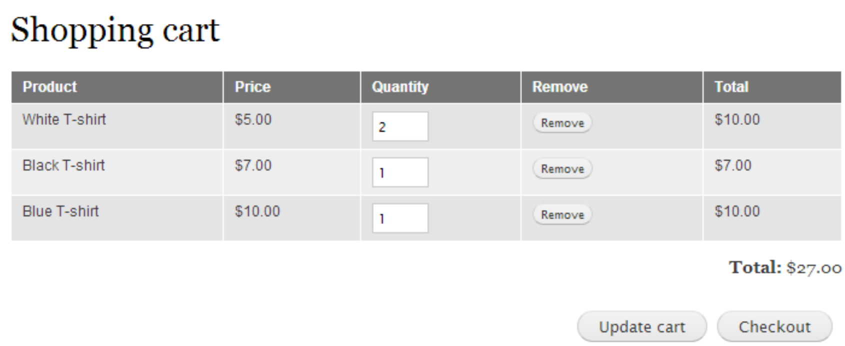
At any time the customer can check the status of the shopping cart by clicking the **View Cart** link (from the **Shopping cart** block). F36.76

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F36.76

Shopping cart

The Shopping Cart always shows the detail of added items, amount per product and total.



From the shopping cart, users can change quantities or delete items. Upon purchase completion they must click on **Checkout** to proceed.

As we saw in the configuration section, the checkout process is divided into multiple pages or steps:

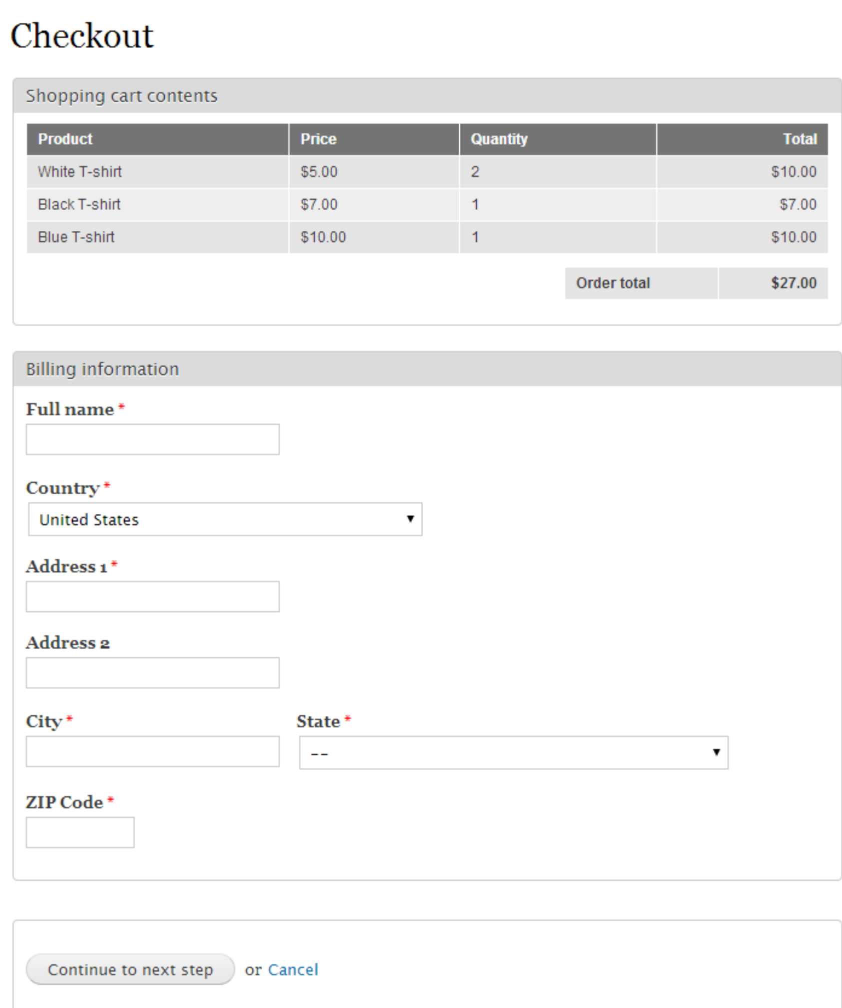
- The first step is the **checkout** and shows the cart content and customer's billing information. F36.77

F36.77

Checkout

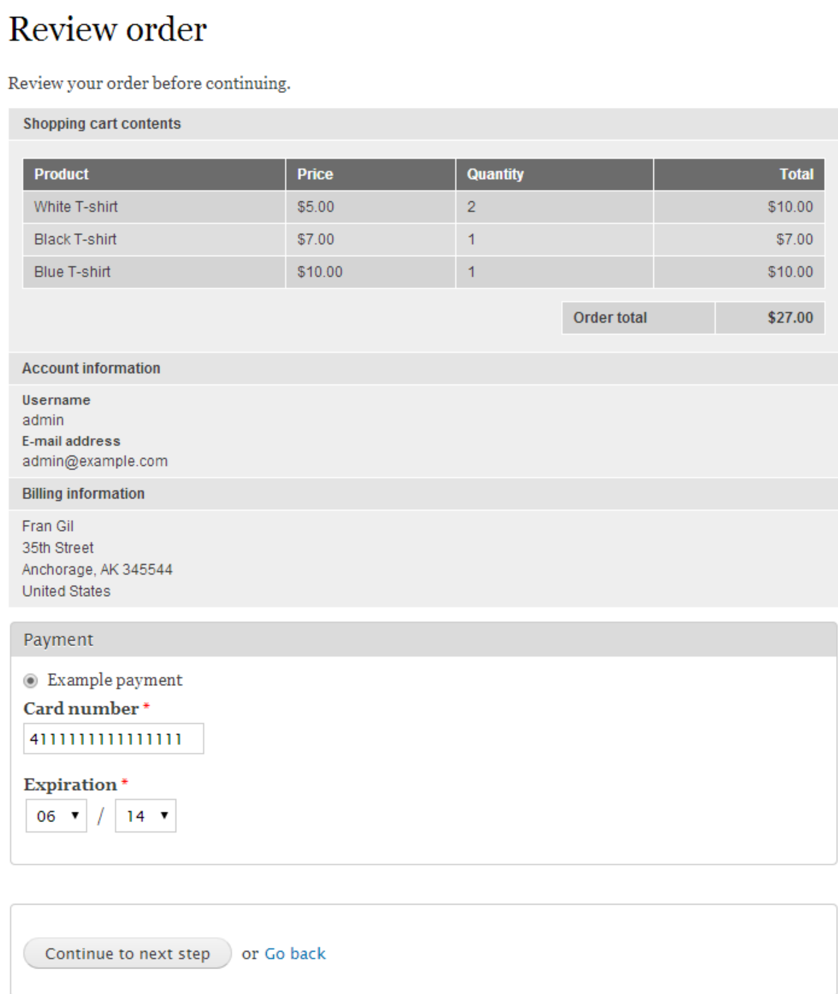
It is the first step in the checkout process.

This step shows the order and requests billing information.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- The second step is to review the order. Order detail is displayed, along with billing information. The customer must select one of the enabled payment methods. F36.78

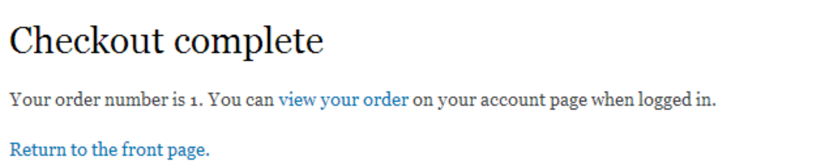


F36.78
Review order
Ordering and billing information is displayed for review.

Also, in this step the payment method is selected.

Depending on the payment method, user pays at the site's gateway or is redirected to an external site.

The process concludes with the display of a message with details of the completed purchase. F36.79



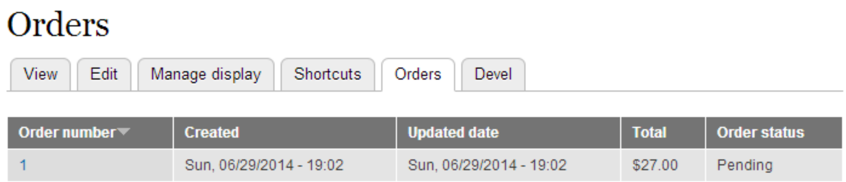
F36.79
Checkout complete
Final message of successful completion.

The user can review the order from the **Orders** tab in his profile. F36.80

F36.80

Orders

List of user's orders from his account.



Order management

An administrator can review orders at: F36.81

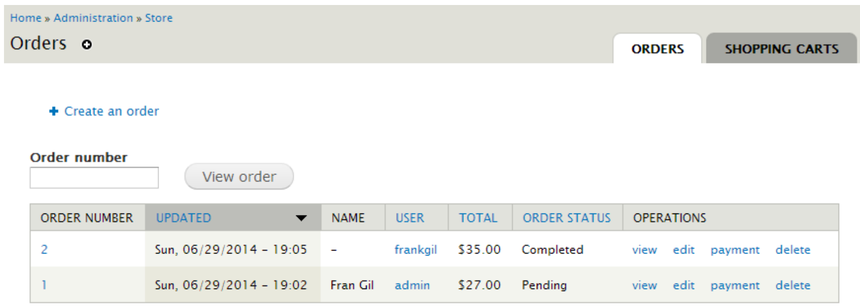
URL Orders
/admin/commerce/orders

Administration ⇨ Store ⇨ Orders

F36.81

Order management

Order management area.



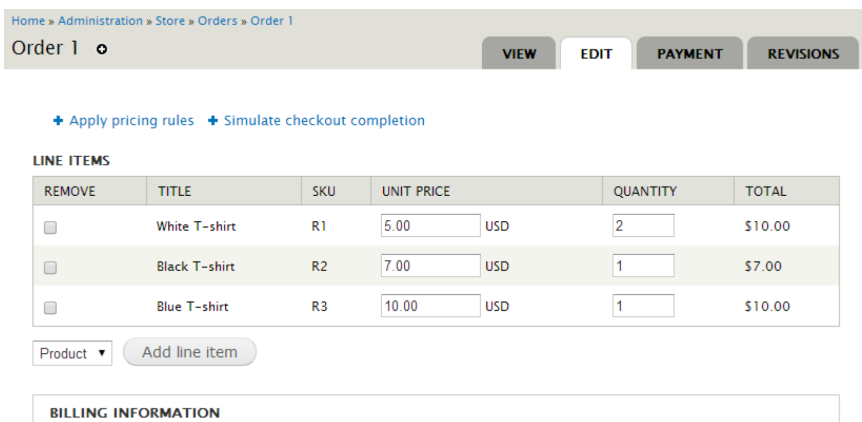
Administrators can edit the order, changing any information regarding billing or ordered products (quantity, price, delete products, etc.). Administrators can also add new line items to the order (link **Add line item**). F36.82

From the order edition page we can also manually change the status of the order.

F36.82

Order edition

When editing an order we can modify any product, add new products, etc.. We can also change the order status or manually enter the payment.



We can confirm payment from the **Payment** tab. For example, if the user has selected to pay by bank transfer, we will have to wait to receive payment in our bank account before proceeding to manually confirm payment.

Payment confirmation does not automatically change the status of the order, which may continue pending awaiting shipment.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Customer profiles

Drupal Commerce creates the default customer profile type **Billing information**. If we want to add new fields to this profile type or browse profiles, we will accomplish this task at:

Administration ⇨ Store ⇨ Customer profiles

URL Customer profiles
/admin/commerce/customer-profiles

Commerce: Other functionalities

36.11

In the official Drupal repository we can find additional modules that increase the functionality of Drupal Commerce. These modules are also available at:

http://www.drupalcommerce.org/contrib

In this section we briefly introduce some of the available options, bear in mind that many of these modules are still under development.

Shipping and handling

The **Commerce Shipping** module provides a basis for other modules that add additional shipping methods and calculate the costs involved.

The Commerce Shipping modules is available at:

http://drupal.org/project/commerce_shipping

Stock

The Commerce Stock module adds stock control capabilities to the virtual store.

This module is available at:

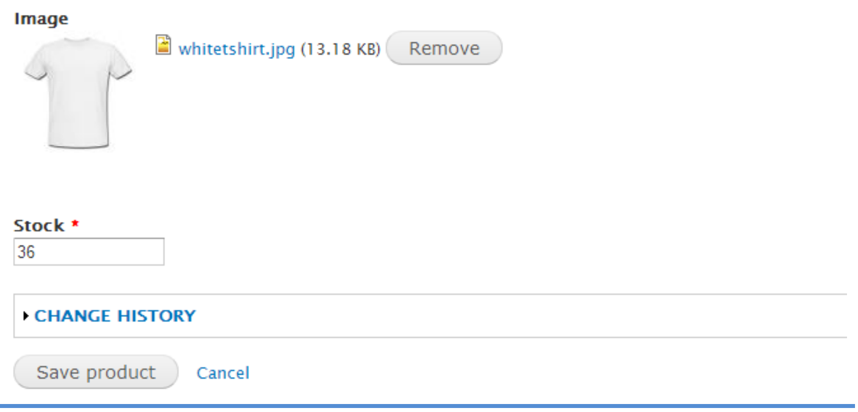
http://drupal.org/project/commerce_stock

Once installed, we can configure product types whose stocks we want to manage from:

Administration ⇨ Store ⇨ Configuration ⇨ Stock management

URL Stock management
/admin/commerce/config/stock

A new **Stock** field will be shown in the selected products, where units sold will be discounted from the total. F36.83



F36.83

Stock control
Control of stock can be performed with the Commerce Stock module.

Discount coupons

The **Commerce Coupon** module, based on Rules, allows for creation of coupons.

This module is available at:

http://drupal.org/project/commerce_coupon

This module consists of the following modules:

- **Commerce Coupon**. Is the main module. It allows creating coupon types and adding rules.
- **Commerce UI**. Interface to insert a coupon during the purchase process.
- **Basic Coupon**. Coupon type with rules and conditions.
- **Coupon Batch**. Allows for mass creation of coupons.

Coupons are managed from:

URL Cupones
/admin/commerce/coupons

Administration ⇨ Store ⇨ Coupons

The module adds two new events that we can use to create the appropriate rules:

- Redeem a coupon.
- Validate a coupon.

It is up to administrators to set the rules properly so that coupons are applied to products in the expected conditions.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

37 SEO Positioning with Drupal

Web positioning, also known as SEO (*Search Engine Optimization*), is a group of techniques that when applied to a website, improve its positioning or visibility in different search engines.

In this unit we will review some modules that can be useful for the optimization of a webpage in Drupal and improve it's positioning.

Search engines information modules:

- **XML sitemap**, generates a site map for search engines.
- **Search 404**, substitutes the error 404 page with a search with the URL not found.
- **Global Redirect**, permanently redirects duplicate URL's to the adequate URL alias.

Content Structure modules:

- **Meta tags**, adds structured metadata to each page including **Open Graph** tags.
- **Page Title**, to generate the page title (<title>).
- **Pathauto** and **Custom breadcrumbs**, previously reviewed.

Analysis and Optimization modules:

- **SEO Checklist**, provides a verification list with modules and good SEO practices.
- **SEO Compliance Checker**, verifies if the content complies with the defined SEO rules.
- **SEO Watcher**, verifies the site ranking in search engines.
- **Content Optimizer** and **Content Analysis**, to analyze and optimize the content.

Comparative D7/D6

Some modules available for Drupal 6 still do not have a version in Drupal 7. Nevertheless, the group of available modules for Drupal 7 extensive and allow the optimization of the site to fulfill the positioning recommendations of SEO.

Unit contents

37.1 Introduction to SEO Positioning 522
37.2 Information For Search Engines 523
37.3 Content Structure 529
37.4 Analysis and Optimization 534

37

37.1 Introduction to SEO Positioning

Web positioning, also known as SEO (*Search Engine Optimization*), is the group of techniques that when applied, improves a websites positioning or visibility in the different search engines.

The optimization of a page includes many elements: titles, contents, URL, design, tags, etc. In this unit we'll see some modules that can be useful for optimizing a webpage in Drupal in order to improve it's positioning.

This unit's objective is not to guarantee positioning in the first positions of a search, but to avoid good content from not being positioned because of bad technical design.

Actions to Improve Web Positioning

There are external actions that we can carry out to position our webpage, such as obtaining links from pages with related subject matter and publishing content in forums, blogs, portals, and social networks. It is also important to generate a good experience for the user with original quality content, and content design and content structure that facilitates navigation.

In this unit we will focus on actions that can be carried out directly on the webpage, such as:

- **Key Words**. Produce webpage content with the key words used to search for the services from the site. In addition, we can highlight the key words in bold or italics.

- **Web accessible**. Making the web accessible, specially limiting the use of Flash, Frameworks, JavaScript and other non-accessible element. In any case, we'll always offer alternatives so the search robot can cover all of the page content.

- **Titles**. The titles and descriptions should be unique to every page and representative of the true content.

- **Sitemap**. The map facilitates the task of navigating and indexing of the search engine.

- **Original Content**. It's important that the content is original.

- **Load time**. It's important to optimize the page load time.

Positioning is a specialization that requires continuous actualization. Search engines evolutionize and modify their positioning algorithm so that only the knowledge of those changes will keep us up to date on the positioning techniques used.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Information For Search Engines

37.2

In this section we will review some modules that provide information to the search engines related to the content structure of the site.

XML sitemap Module

The XML sitemap module generates a sitemap in XML form according to the protocol defined in sitemaps.org. Many search engines use this protocol to maintain the indexed results updated. In addition, some search engines such as Google, Bing or Yahoo allow the sitemap to be sent directly so that it's used in the indexation of its contents.

The XML sitemap module is available at:

http://drupal.org/project/xmlsitemap

It's composed of a group of modules that allow very complete maps to be created:

- **XML sitemap**. Is the base module necessary to generate the map to the XML site.
- **XML sitemap custom**. Allows personalized links to be added to the map.
- **XML sitemap engines**. Allows the map to be sent to different search engines.
- **XML sitemap internationalization**. Sitemaps in language function.
- **XML sitemap menu**. Adds menu elements to the sitemap.
- **XML sitemap node**. Adds the node links to the sitemap.
- **XML sitemap taxonomy**. Adds term taxonomy links to the sitemap.
- **XML sitemap user**. Adds user profile links to the sitemap.

Administration of the XML site map is carried out from:

Administration ⇨ Configuration ⇨ Search and metadata ⇨ XML sitemap

URL XML Site Map
/admin/config/search/xmlsitemap

Maps created will be displayed in the **List** tab. The module will create a map by default in the site root: F37.1

http://www.example.com/**sitemap.xml**

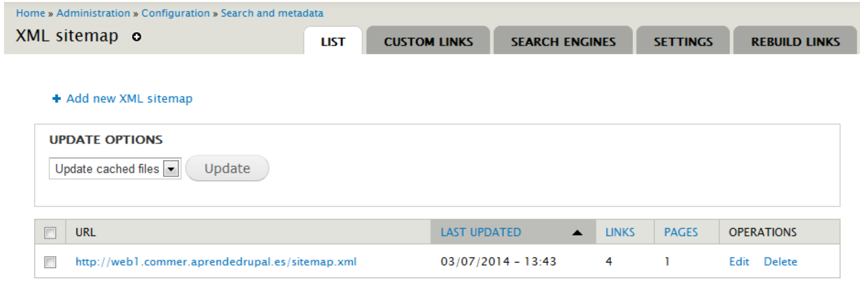
The sitemap generates or is actualized with every clock execution, which is why an initial execution of time is required to generate the first version of the map. It's also possible to actualize the maps by using the actualization option **Update cached files**.

Initially only the main site page will be indexed in the map. We will have to activate and configure the rest of the modules included in the XML sitemap to include nodes, menus, etc.

F37.1

XML Sitemap Map List

The XML maps created are displayed on the list.



We can generate some general options from the **Settings** tab, such as the **minimum sitemap lifetime**, which will establish the minimum time that should transpire between each clock execution. The recommended value is 1 day, although this value will depend on the degree of actualization of the site contents. **F37.2**

In the **Frontpage** group options we can configure the **Priority** and **Change frequency** parameters of the site's front page.

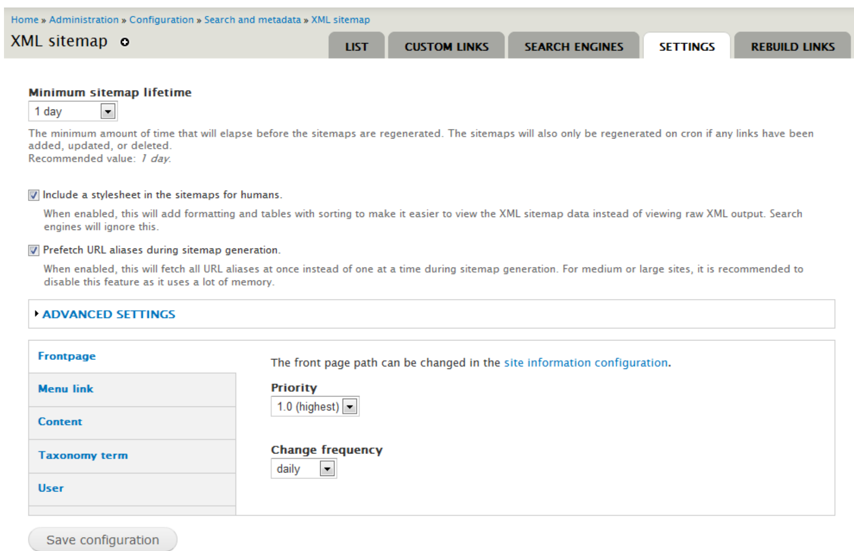
Priority is a parameter that is added to every link indexed in the XML map, and will have a value between 0.0 (the lowest) and 1.0 (the highest), 0.5 being the established value for Normal priority. Priority establishes the order or importance of a page in comparison to the rest. This is why the site's front page is assigned maximum priority by default.

F37.2

XML Sitemap Options

General options such as minimum sitemap lifetime.

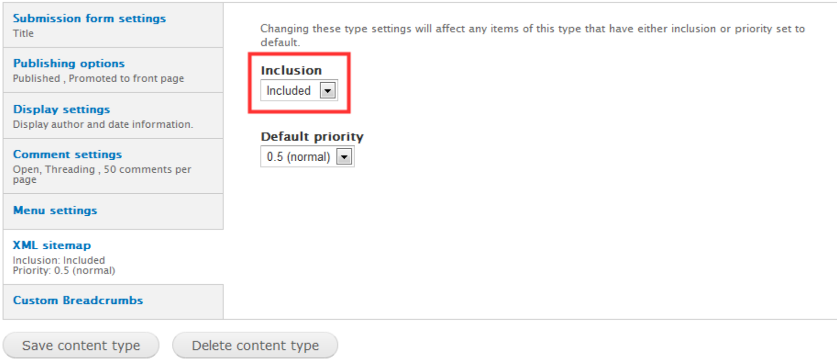
Priority allows XML map contents to be organized in the order of importance.



To **index nodes** we must indicate what type of content will be included in the XML sitemap. First we must have activated the **XML sitemap node** module.

In each **content type** a new option group XML map of the site will be displayed. By default, all of the content types will be excluded, which is why we must edit each one and change the option **Inclusion** to **included**. **F37.3**

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



F37.3

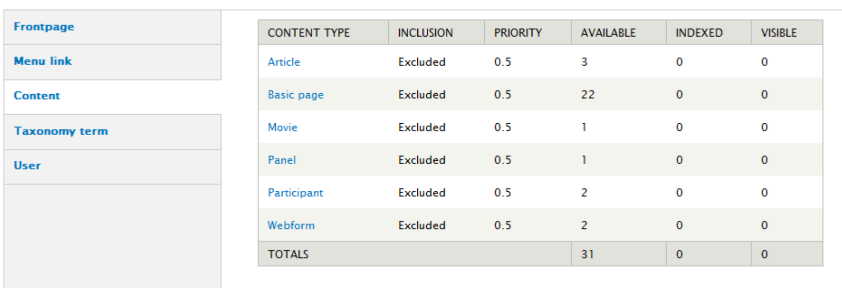
XML Sitemap Content Type

We can configure the default values in each content type. If the node of this type will be included in the map and what will be the priority.

Once included, we must actualize the XML map manually or simply wait for it to expire.

Returning to the **Settings** tab of the **XML sitemap**, in the options **Content** group the actual state of the indexation will be displayed. The available content types in the site and the number of nodes available and indexed will be displayed.

To modify the inclusion of the content type and its priority, we must edit the content type. The links displayed in **Content** will take us directly to the editing of each content type. F37.4



CONTENT TYPE	INCLUSION	PRIORITY	AVAILABLE	INDEXED	VISIBLE
Article	Excluded	0.5	3	0	0
Basic page	Excluded	0.5	22	0	0
Movie	Excluded	0.5	1	0	0
Panel	Excluded	0.5	1	0	0
Participant	Excluded	0.5	2	0	0
Webform	Excluded	0.5	2	0	0
TOTALS			31	0	0

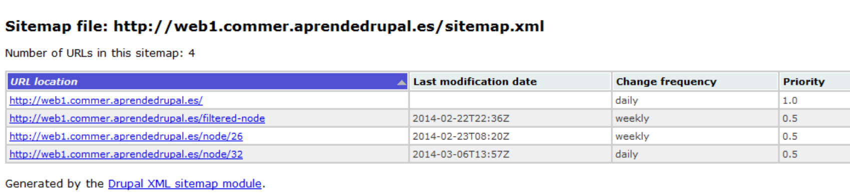
F37.4

XML Sitemap Content

From the Content tab the content is displayed indexed in the map by content type.

We can also modify the particular parameters of each node (inclusion and priority) from its edition.

The XML map generated can be consulted by accessing it directly from the navigator at the URL **www.example.com/sitemap.xml**. F37.5


Sitemap file: http://web1.commer.aprendedrupal.es/sitemap.xml

Number of URLs in this sitemap: 4

URL location	Last modification date	Change frequency	Priority
http://web1.commer.aprendedrupal.es/		daily	1.0
http://web1.commer.aprendedrupal.es/filtered-node	2014-02-22T22:36Z	weekly	0.5
http://web1.commer.aprendedrupal.es/node/26	2014-02-23T08:20Z	weekly	0.5
http://web1.commer.aprendedrupal.es/node/32	2014-03-06T13:57Z	daily	0.5

Generated by the Drupal XML sitemap module.

F37.5

XML Sitemap Map in XML

The XML map is displayed loaded directly on the navigator.

Following a similar procedure, we will add other elements to the map such as menus, taxonomy term links or user profile links. These options will be available in the function of the activated modules.

Once we have generated the XML sitemap, we can include its URL in the search engines. Generally the search engines offer this type of tools for Webmasters.

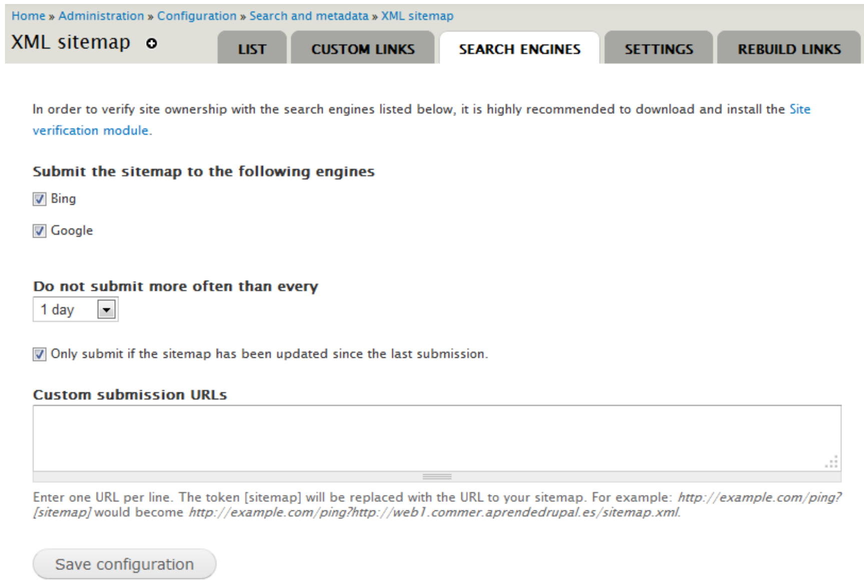
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

The **XML sitemap engines** module facilitates the delivery of the XML sitemap to some search engines such as Bing or Google. From the **Search engines** tab we can indicate what engines and with what frequency the map will be sent. F37.6

F37.6

XML Sitemap Search Engines

Configuration of the search engines to which the actualized XML sitemap will be sent periodically.



For the correct functionality of this service, installing the **Site verification** module is recommended, that allows configuration of the site so that the search engines can verify the authenticity of the delivery.

The **Site verification** module is available at:

http://drupal.org/project/site_verify

Search Module 404

The **Search 404** module substitutes the error page 404 with a search operation with the result of search the URL not found on the site.

For example, if the page **http://www.example.com/noticias** is not found in the site, the module will display the result search the chain "news".

The Search 404 module is available at:

http://drupal.org/project/search404

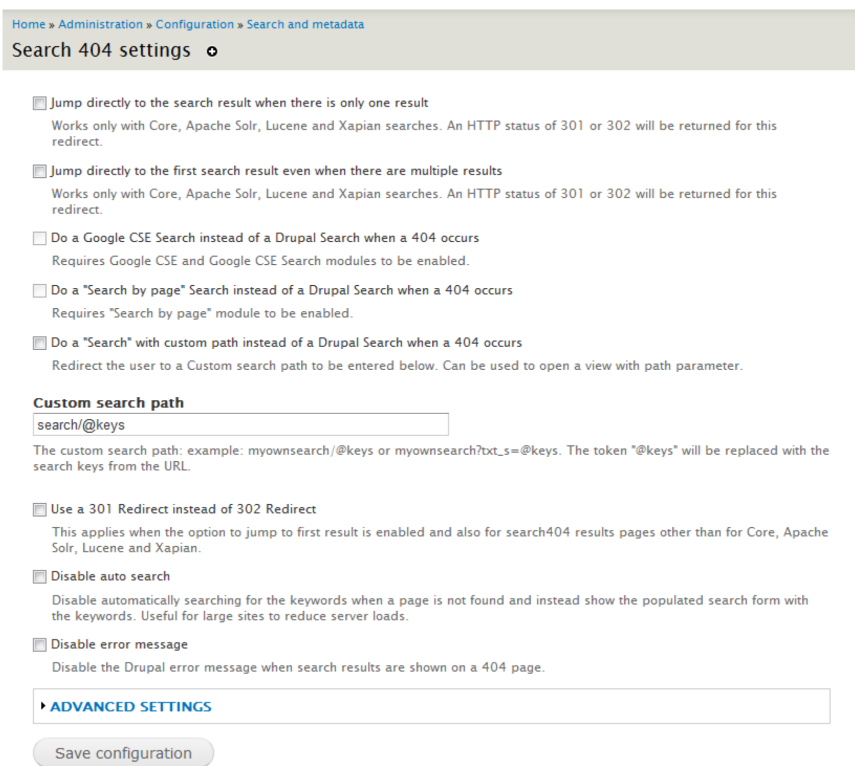
Once installed we can configure the module from:

Administration ⇨ Configuration ⇨ Search and metadata ⇨ Search 404 Settings

URL Search 404
/admin/config/search/search404

Although we will not need to modify these options so that the module functions, we can configure some parameters such as: F37.7

- **Jump directly to the search result when there is only one result**.
- **Jump directly to the first search result** of the search.
- **Disable auto search**. In this case, the search form will be displayed with the option to search, but without actually executing the search. The user must click on Search if he wants to obtain the results.
- **Disable error message**. By default, a message is displayed indicating the page doesn't exist and it proceeds to carry out the search.



F37.7
Search 404
The Search 404 module substitutes the error 404 page for a search. The string to search will be the URL that hasn't been found, previous transformation in words.

It's important to indicate that in order for the search 404 to be displayed, the user must have permission to carry out the searches on the site. Otherwise, the error 404 page will be displayed.

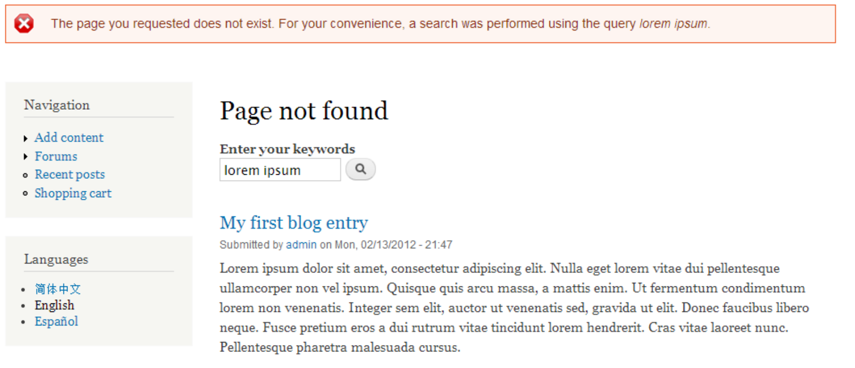
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

As an example the results obtained by the search 404 will be displayed when we attempt to load the URL www.example.com/**lorem/ipsum**, page that does not exist on the site. F37.8

F37.8

Search 404 Result

As an example, the result obtained when attempting to access the URL /lorem/ipsum is displayed. Since the URL does not exsitst on the site, it has been substituted for a search of the words: lorem ipsum.



Global Redirect Module

With the **Global Redirect** module, we avoid offering the search engine duplicate URL's on the same content. In Drupal, when we generate an URL alias, the original URL (for example, node/1234), is still accessible.

The Global Redirect module generates on the pages with the URL without alias, indicating to the search engine that the old URL has changed and a permanent link to the new one is established.

This procedure is also useful when we change the URL alias of any page, in such a way that all Accesses to the previous page are permanently associated to the new address.

The **Global Redirect** module is available at:

http://drupal.org/project/globalredirect

And we can Access its configuration from:

URL Global Redirect
/admin/config/system/globalredirect

Administration ⇨ Configuration ⇨ System ⇨ Global Redirect

If we attempt to Access an internal URL, such as node/1234, the system will automatically redirect us to the page with the adequate alias. The behavior is not what's most important for the user, but that the search engine vision will also be adequate, registering only the final URL.

Content Structure

37.3

In this section we look at other modules that improve the structure or presence of the page content.

Meta Tags Module

The **Meta tags** module adds structured metadata on every page on the site

The module is displayed at:

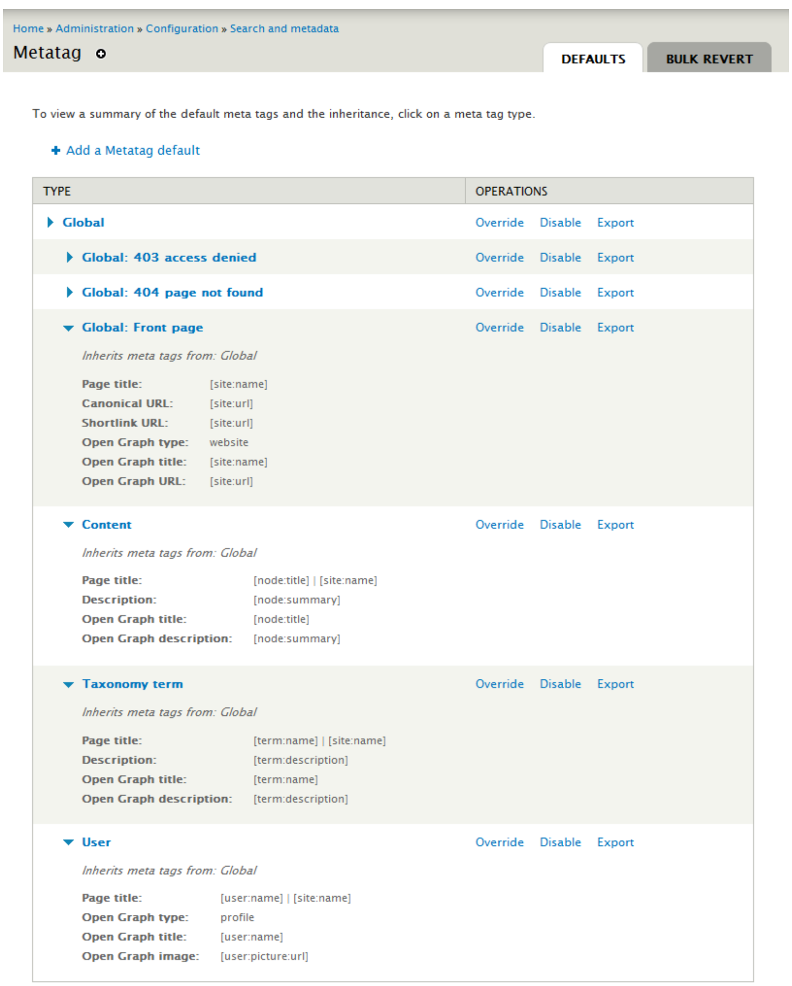
http://drupal.org/project/metatag

It also included support for **Open Graph tags** (through the **Open Graph meta tags** module), Which is the protocol used by Facebook to obtain information from the page when sharing it on the social network.

We access the module's general configuration from: F37.9

Administration ⇨ Configuration ⇨ Search and metadata ⇨ Meta tags

URL Meta tags
/admin/config/search/metatags



F37.9

Meta Tags

Configuration of meta tags with Meta Tags. By default the established values can be modified for each individual element.

For example, establishing patterns for the elements of one content type, when creating each node we will be able to specify other tag values.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

The module adds values by defect for the main page meta tags, the contents, taxonomy terms, and user profiles.

We can also add tags by default for predetermined elements, such as, for a specific content type. We will do this by clicking on **Add a meta tag default** and selecting the type or element for which the tags will be defined. F37.10

F37.10

Meta tags

New Type

We can add new values by default for different elements of the site like content type and taxonomy terms.

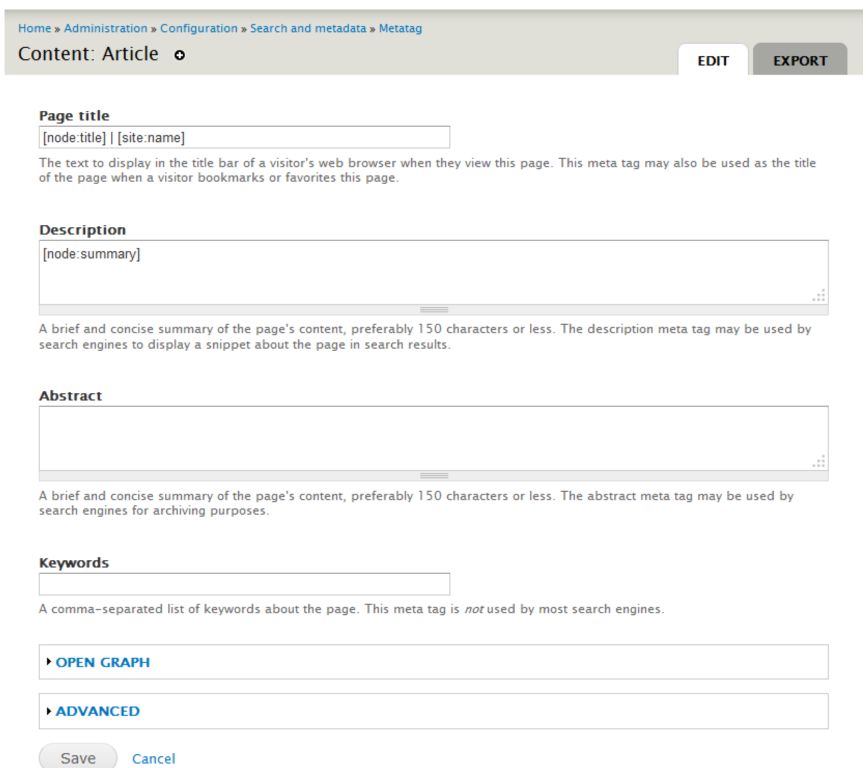
We can add values by making use of special characters, **Description** tags, **Keywords**, **Page title**, in addition to **Open Graph** tags. These fields will not be visible on the page, but will generate as tags in HTML code. For example, Title does not refer to the node title, but to the tag <title></title>, which is the value that is displayed in the navigator tab and is used in the search results as a title for the link page. F37.11

F37.11

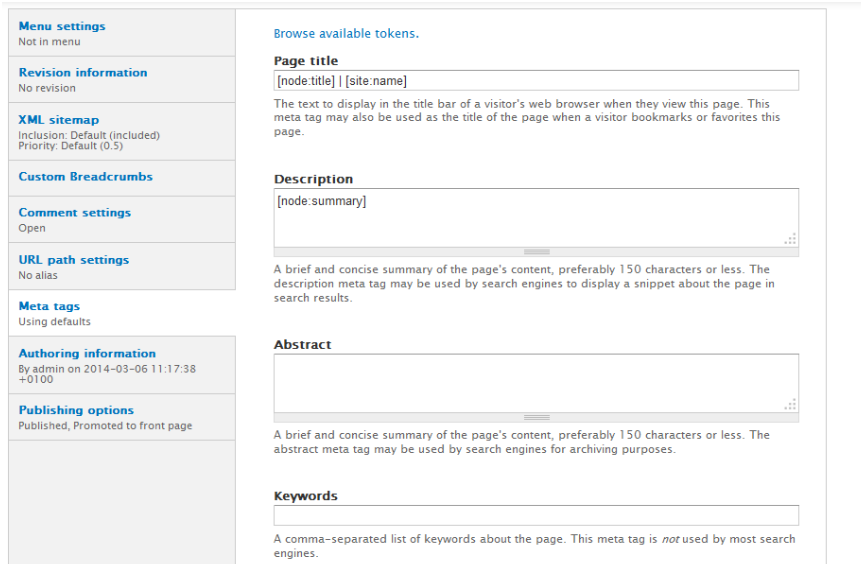
Meta tags

Characters

Through replacement characters we can establish the structure that each meta tag will have.



Meta tags can also be personalized for each node from the **Metatags** option. F37.12



F37.12

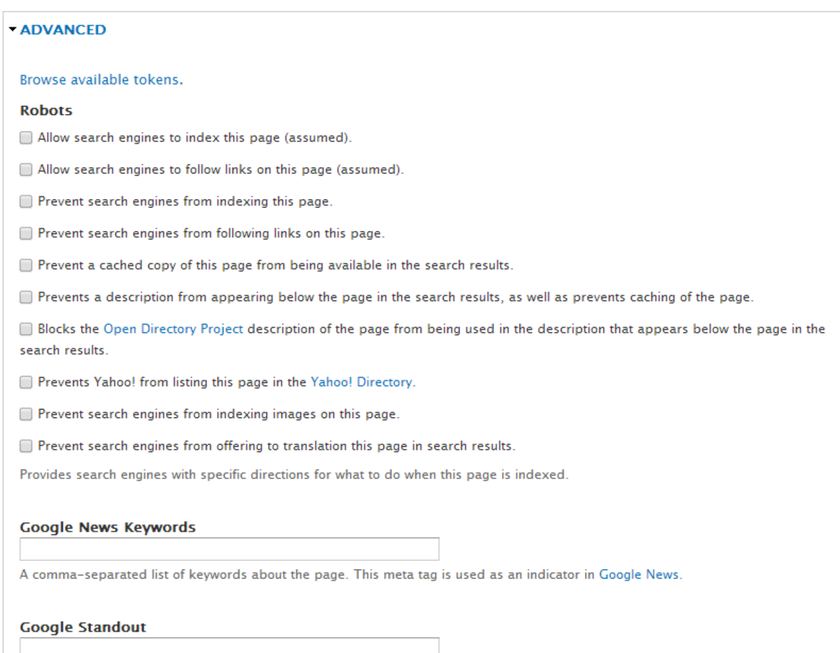
Meta tags

Node

From node editing, we can modify the values of its meta tags by default.

From the editing from any of the types we can also access other advanced options, such as: F37.13

- Prevent search engines from indexing this page.
- Prevent search engines from following links on this page.
- Prevent a cached copy of this page from being available in the search results.
- Indicate Copyright values, Canonical URL, Shortlink URL and Editor URL



F37.13

Meta tags

Advanced Options

From the content type editing we can establish advanced options such as do not index content, do not follow links, etc.

All of the values will embed themselves in the page's HTML code in tag form that will be interpreted by the search engines.

Page Title Module

The **Page Title** module allows the addition of patterns to generate the title of any page automatically. Instead of Meta tags, we refer to the tag <title></title> and not to the node title.

The tag <title> is found in HTML within the header (<head>) of each page. The value of this tag is displayed on the **navigator tab** and is also the value displayed by the search engines.

The Page Title module is available at:

http://drupal.org/project/page_title

We access the module's general configuration from: F37.14

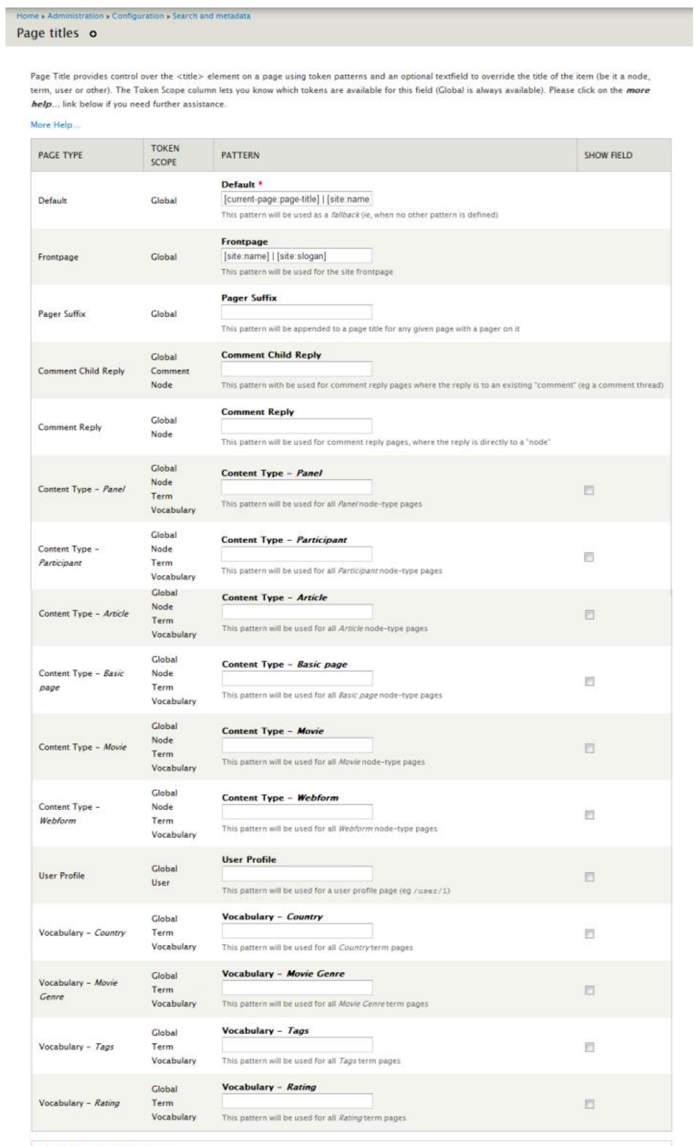
URL Page Title
/admin/config/search/page-title

Administration ⇨ Configuration ⇨ Search and metadata ⇨ Page titles

F37.14

Page Title Configuration

The Page Title module allows configuration of the value of the tag <title>.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Using tokens, we can fix the pages' titles for different elements:

- Frontpage.
- Pager Sufifx.
- Comment reply pages.
- Specific values according to content type.
- User profiles.
- Taxonomy terms pages.

For example, using the **Article** type content pattern:

Article [current-page:page-title] | [site:name]

The value of the tag <title> from the Drupalcamp Spain 2012 content would be:

Article Drupalcamp Spain 2012 | Expert in Drupal 7 - Intermediate

The **Figure** F37.15 shows the title in the navigation tab (when placing the cursor over the title it will be displayed completely). As we can see, the title of the tag <title> doesn't have to correspond to the content title.

F37.15

Page Title

The page title is not the node title, but the value of the tag HTML <title>, that is used as a title in the navigator tab and as page title on the search results.

The page title doesn't have to coincide with the node title. Generally the page title contains the node title.

Pathauto Module

The **Pathauto** module allows URL alias to be automatically generated, in such a way that it won't be necessary to complete the **URL path settings** field each time we create a node.

The **Pathauto** module was studied in detail in **Unit 22** (**section 22.7**).

Custom Breadcrumbs Module

The **Custom breadcrumbs** module allows personalized breadcrumbs to be created according to the content type

The Custom breadcrumbs module was in the previous **Unit 25** (**section 25.2**).

37.4 Analysis and Optimization

In this section we will introduce some modules for site analysis and in this way verify if it's configured correctly so that it has adequate positioning.

SEO Checklist Module

The **SEO Checklist** module generates a checklist with recommended Drupal modules and good practices for the SEO positioning

The module is available at:

http://drupal.org/project/seo_checklist

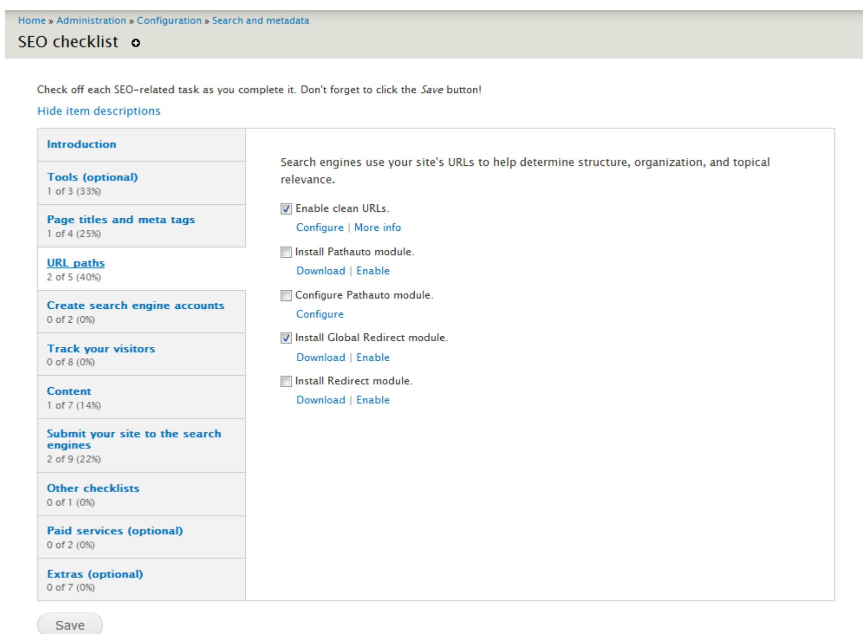
We will access the module checklist from:

URL SEO Checklist
/admin/config/search/seochecklist

Administration ⇨ Configuration ⇨ Search and metadata ⇨ SEO Checklist

The checklist facilitated by the module should be used as a recommendation and not as an obligation. For example, it can happen that the functionality proposed through a particular module is already being applied with a different module not referenced in the checklist.

The checklist is divided in element groups: Page Titles, URL paths, Content, Spam protection, etc. It has to do with a good starting point so that we don't forget to install and configure any of the important modules to create a website with Drupal that is well structured and "understandable" by search engine crawlers. F37.16



F37.16

SEO Checklist Verification List

The checklist module generates a checklist of recommendations (basic Drupal modules for SEO positioning and good practices).

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

SEO Compliance Checker Module

The **SEO Compliance Checker** module verifies when creating or editing content, if it complies with the defined SEO rules.

The module is available at:

http://drupal.org/project/seo_checker

SEO Compliance Checker is composed of **three modules**:

- **SEO Compliance Checker**. It deals with the base module that verifies the defined rules when creating or editing content. This module doesn't add rules, which is why we must jointly install the other included modules, or additional modules that make use of their API.
- **Basic SEO Rules**. Includes basic positioning rules.
- **Keyword Rules**. Includes rules to verify the correct utilization of key words in the content.

Once the three modules have been installed, we will Access the configuration from:

Administration ⇨ Configuration ⇨ Content authoring ⇨ SEO Checker

URL SEO Checker
/admin/config/content/seo_checker

The options are available at: F37.17

- Carry out the verification of the SEO rules only in content **preview** or also when the node is saved.
- Select the **content types** in which the SEO rules are verified.
- **Modify SEO rules parameters**. The module initially applies reference parameters, but we can modify them to adapt them to the real needs of the site.

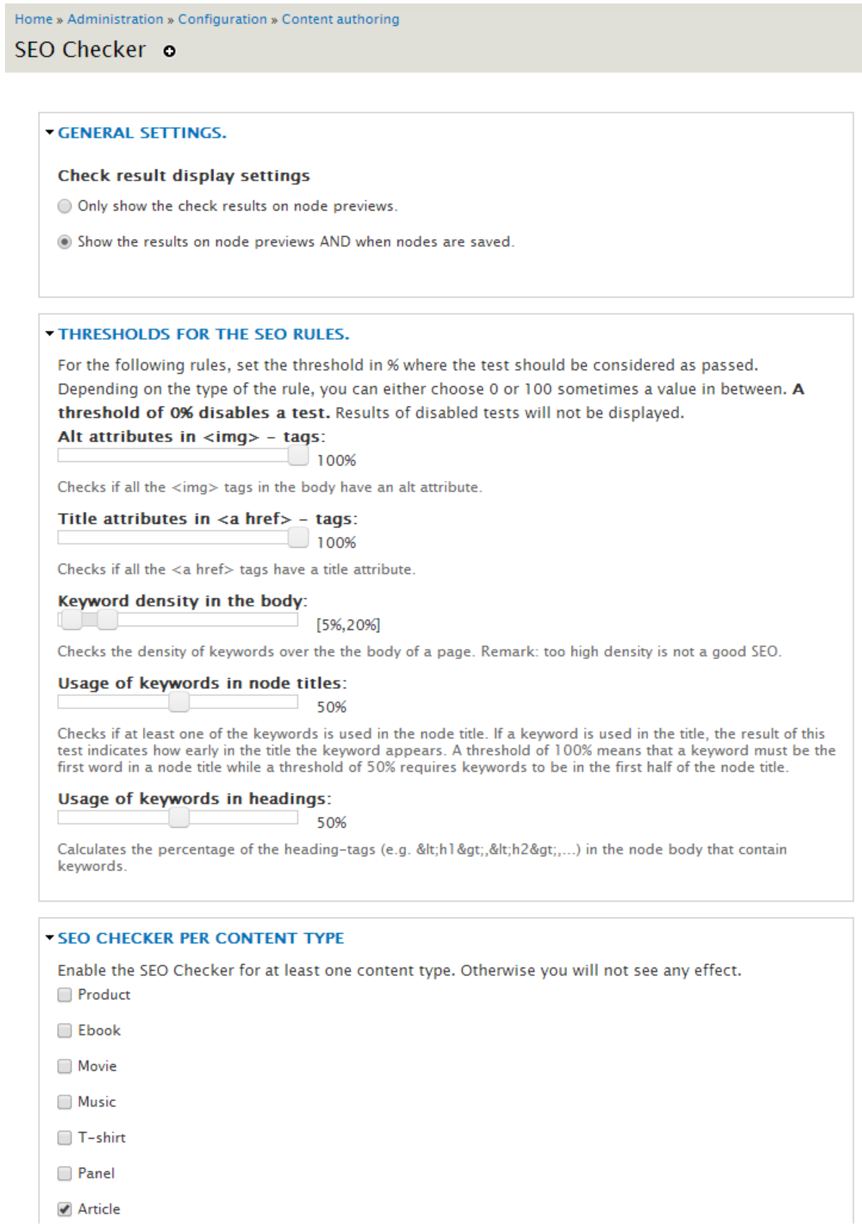
The following verifications will be carried out:

- Verify that the **images** (<img>) have the attribute **alt**: 100% (all images should fulfill the rule).
- Verify that all the **links** (<a href> have the attribute **title**: 100% (all links should fulfill the rule).
- **Density of key words** in the body. Initially the established value is between 5 and 20%. A moderate value is recommended since a massive presence of the key word can jeopardize the page positioning.
- Use of the **key words in the node title**. The 100% indicates the key word can be found in the beginning of the title. The 50% indicates the key word can be found in the first half of the title.
- Use of **key words in the page title** (<title>). The 100% indicates the key word can be found in the beginning of the title. The 50% indicates the key word can be found in the first half of the title. This rule will be available when using the module **Page Title**.

F37.17

SEO Compliance Checker Configuration

The module carries out a verification of the available rules in the function of the indicated parameters. The initial parameter serve as reference athough we may modify them.



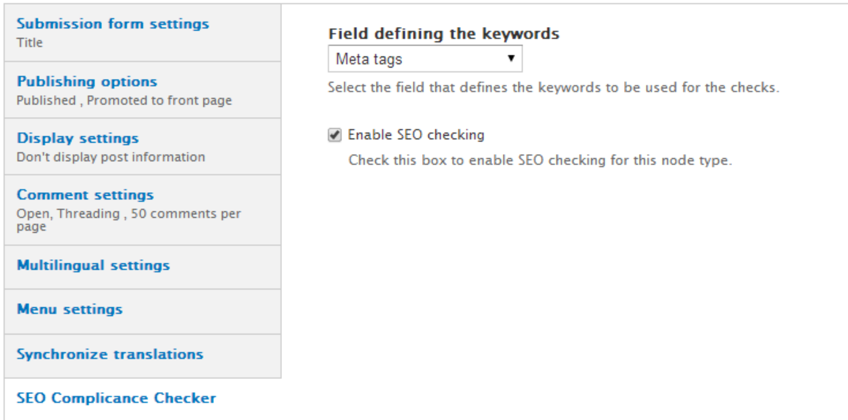
Prior to putting the module into practice, we must configure the content types for which the SEO verification will be done.

From the **content type editing**, we must configure: F37.18

- **SEO Compliance Checker**. We activate the verification and must select the field where the key words will be defined.

  For example, if we are using the **Meta tags** module, we will select this field (Meta tags), so that the value of the key words will be obtained from the **Key word** field of Meta tags.

- **Page Title Settings**. So that the verification rule is available, we must activate the **Show field**.



F37.18

SEO Compliance Checker

Content type

From content type editing, we can indicate if the module analysis is active and in what field the key words are stored.

Lets look at an application example by creating a new article where the key word will be "drupalcamp spain". Initially, we'll include some positioning errors and we will correct them later.

When creating the content we will indicate these values:

- **Title**: Article
- **Body**: Initially we won't include the string "drupal" in the body (add a lorem ipsum text).
  We test <img>, <a href> and headings by including this code:

```
<h2>Drupal header</h2>
<a href="http://drupal.org" title="drupal">drupal link</a>
<img src="https://www.drupal.org/files/druplicon.small_.png"
alt="drupal logo">
```

- **Keywords** (Meta tags group): drupal
- **Page title** (**Page title options** group): There will be Drupalcamp Spain in 2012.

When carrying out verification of the SEO rules **3 errors are detected**: F37.19

- The density of the keywords in the **body** is not within the established range.
- The usage of the keywords in the **node title** is not correct.
- The usage of the keywords in the **page title** is not correct.

• SEO Check Results:

RULE	MESSAGE	ACHIEVED	REQUIRED	PASSED
Alt attributes in <img> - tags	Test passed.	100%	≥ 100%	✔
Title attributes in <a href> - tags	Test passed.	100%	≥ 100%	✔
Keyword density in the body	Test failed, please make sure you use your keywords in the body but not too often.	2%	∈ [5%,20%]	✖
Usage of keywords in node titles	Test failed, place your keywords early in your node title.	0%	≥ 50%	✖
Usage of keywords in page titles (Page Title Module)	Test failed, place your keywords early in your page title.	22%	≥ 50%	✖
Usage of keywords in headings	Test passed.	100%	≥ 50%	✔

F37.19

SEO Compliance Checker

Analysis With Errors

A content example is displayed that has not passes some of the rules.

To correct these errors and improve the page positioning, we will apply the following changes to the content:

- **Title**: Drupal article.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

In this case the keyword is in the first half of the title string (50%). When placing the keyword at the beginning, we are in reality applying a 100%.

- **Page Title**: Drupal article.

  As in the Title field, the keyword is in the first half of the page Title (50%). When placing the keyword in the beginning, we are in reality applying 100%.

- **Body**: We will add the string "drupal" several times until reaching the minimum density. In a real text the inclusion of the keyword must make sense and be coherent with the content.

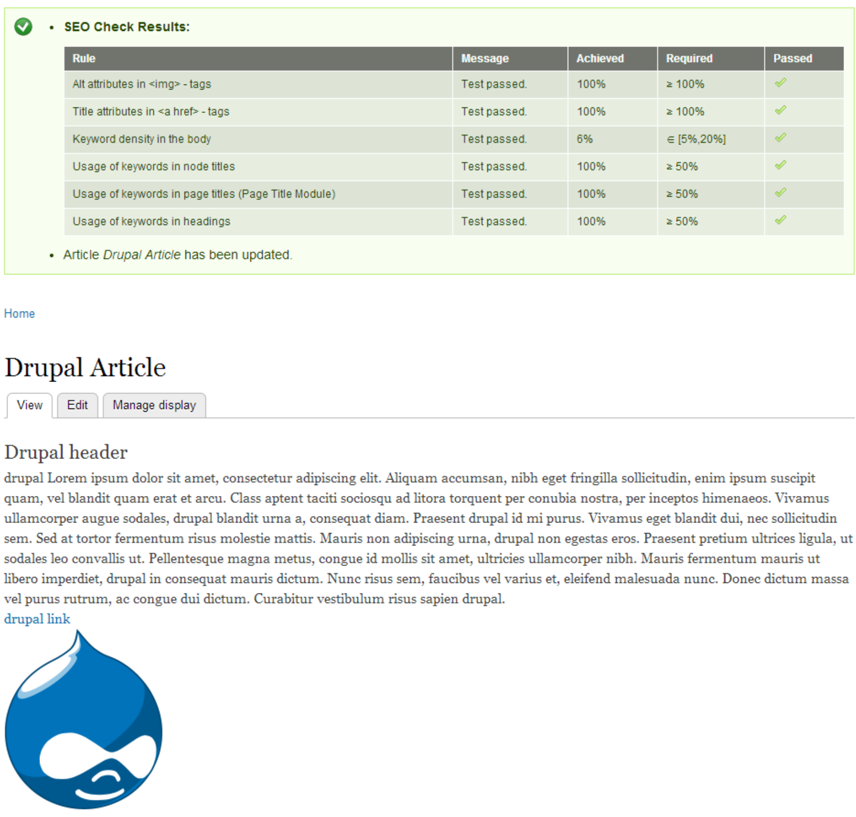
Once the content has been saved, we will verify that it now passes all of the carried out SEO validations. F37.20

F37.20

SEO Compliance Checker

Analysis Without Errors

Once the errors or recommendations are corrected, the content has passed all of the SEO validation rules.



- SEO Check Results:

Rule	Message	Achieved	Required	Passed
Alt attributes in <img> - tags	Test passed.	100%	≥ 100%	✓
Title attributes in <a href> - tags	Test passed.	100%	≥ 100%	✓
Keyword density in the body	Test passed.	6%	∈ [5%,20%]	✓
Usage of keywords in node titles	Test passed.	100%	≥ 50%	✓
Usage of keywords in page titles (Page Title Module)	Test passed.	100%	≥ 50%	✓
Usage of keywords in headings	Test passed.	100%	≥ 50%	✓

- Article *Drupal Article* has been updated.

Home

Drupal Article

View | Edit | Manage display

Drupal header

drupal Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam accumsan, nibh eget fringilla sollicitudin, enim ipsum suscipit quam, vel blandit quam erat et arcu. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Vivamus ullamcorper augue sodales, drupal blandit urna a, consequat diam. Praesent drupal id mi purus. Vivamus eget blandit dui, nec sollicitudin sem. Sed at tortor fermentum risus molestie mattis. Mauris non adipiscing urna, drupal non egestas eros. Praesent pretium ultrices ligula, ut sodales leo convallis ut. Pellentesque magna metus, congue id mollis sit amet, ultricies ullamcorper nibh. Mauris fermentum mauris ut libero imperdiet, drupal in consequat mauris dictum. Nunc risus sem, faucibus vel varius et, eleifend malesuada nunc. Donec dictum massa vel purus rutrum, ac congue dui dictum. Curabitur vestibulum risus sapien drupal.

drupal link

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

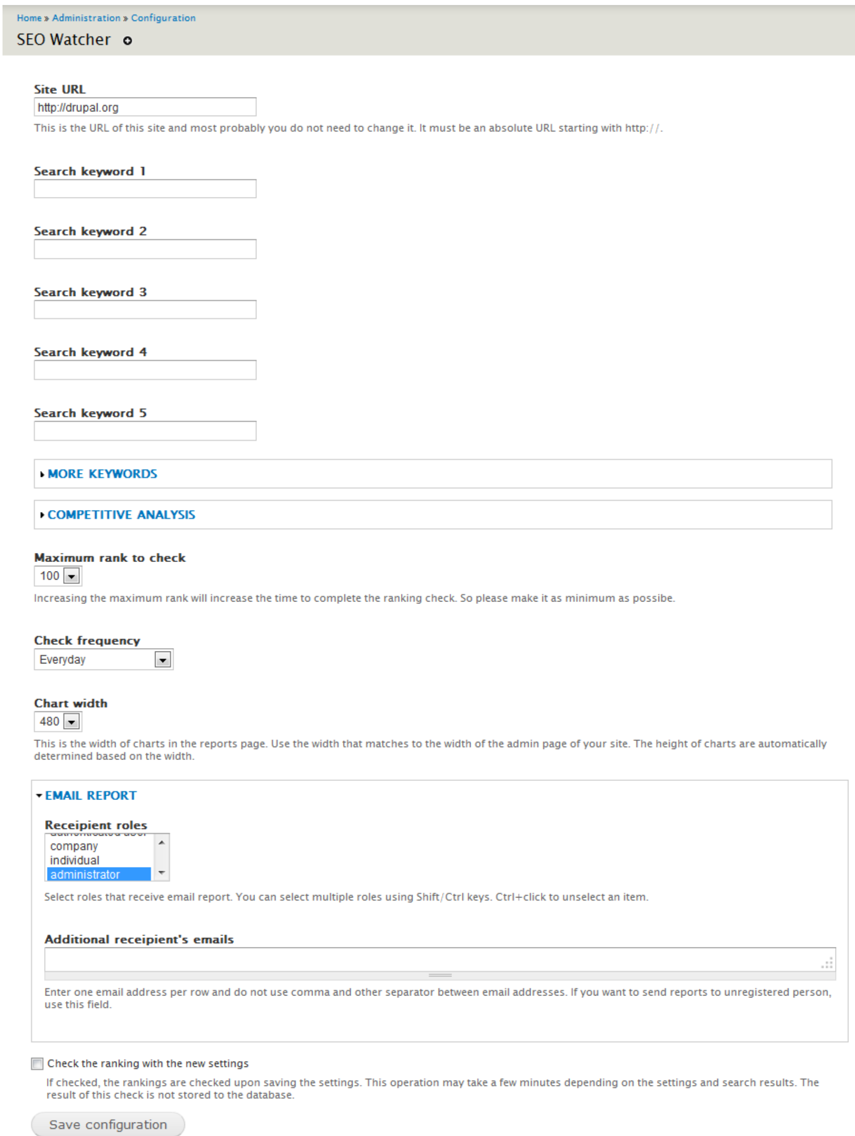
SEO Watcher Module

The **SEO Watcher** module looks for the key words specified in some search engines and returns a report with the position of the site with respects to the carried out search.

The SEO Watcher module is available at:

http://drupal.org/project/seowatcher

Once installed, we will be able to access the module configuration from the administration index, **/admin/index** (SEO Watcher), or directly in the URL **/admin/config/seowatcher**.



F37.21

SEO Watcher Configuration

The module allows programming of a ranking verification in different search engines for predetermined key words.

This verification will be executed periodically, by sending an e-mail to the administrator (or indicated directions) with the result report.

With this module we can program a ranking verification in the main search engines (Google, Yahoo and Bing) that will periodically execute, sending e-mail with the information to the indicated destinations.

The parameters we must define are: F37.21

- **Site URL**. URL of the site that we want to verify, beginning with http://. Generally, the URL of the site where the module is installed will be verified. Which is why we don't need to modify the pre-established value.

- **Search keywords**. Up to 10 words or strings may be established.

- **Competitive analysis**. We can indicate the URL o the sites with which we compete to compare the positioning results.

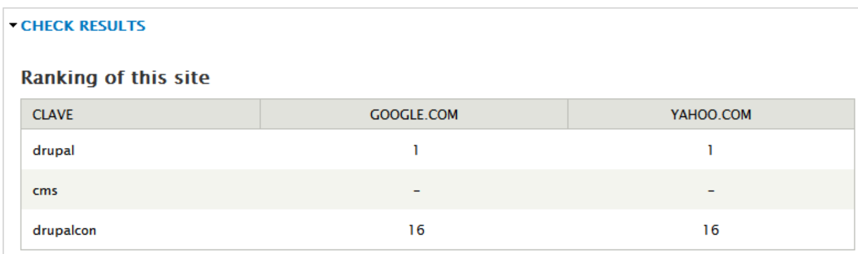
- **Maximum rank to check**. A value will only be returned if the search result can be found below the indicated maximum.

- **Chart width**. Refers to the width of the images on the graphics of the report.

- **Email destinations**. We can indicate we want an email to be sent to all users with a predetermined role or include other destinations manually.

- **Check the ranking**. Finally, when saving the configuration we can activate this option so the module carries out the verifications and returns a result. F37.22

F37.22
SEO Watcher Ranking
Site ranking for the indicated key words.


▾ CHECK RESULTS

Ranking of this site

CLAVE	GOOGLE.COM	YAHOO.COM
drupal	1	1
cms	-	-
drupalcon	16	16

Reports are generated through the system clock according to the frequency established in the module configuration. We can consult the generated reports from:

URL Report SEO Watcher
/admin/reports/seowatcher

Administration ⇨ Reports ⇨ SEO Watcher

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Content Optimizer and Content Analysis Modules

The **Content Optimizer** module analyses the content of any page on the site and returns a report with recommendations so the content complies with good SEO practices.

The **Content Optimizer** module is available at:

http://drupal.org/project/contentoptimizer

The module requires the API facilitated by **Content Analysis**, available at:

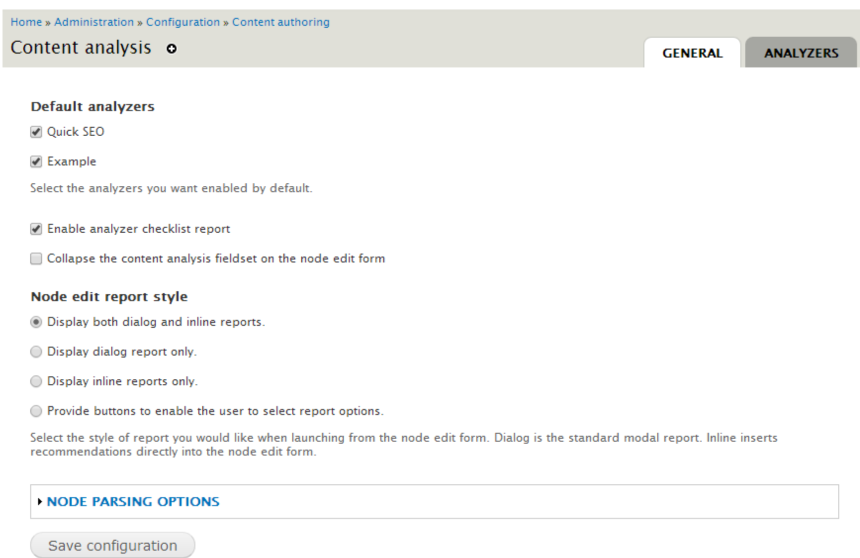
http://drupal.org/project/contentanalysis

Once installed, we will access the module configuration from:

Administration ⇨ Configuration ⇨ Content authoring ⇨ Content analysis

URL Content analysis
/admin/config/content/contentanalysis

From the **General** tab we can activate the analysis and configuration as displayed in the node. The option by default (Display both dialog and inline reports) will display the analysis information in a new window and in the same node, next to every affected field. F37.23



F37.23

Content Analysis and Content Optimizer

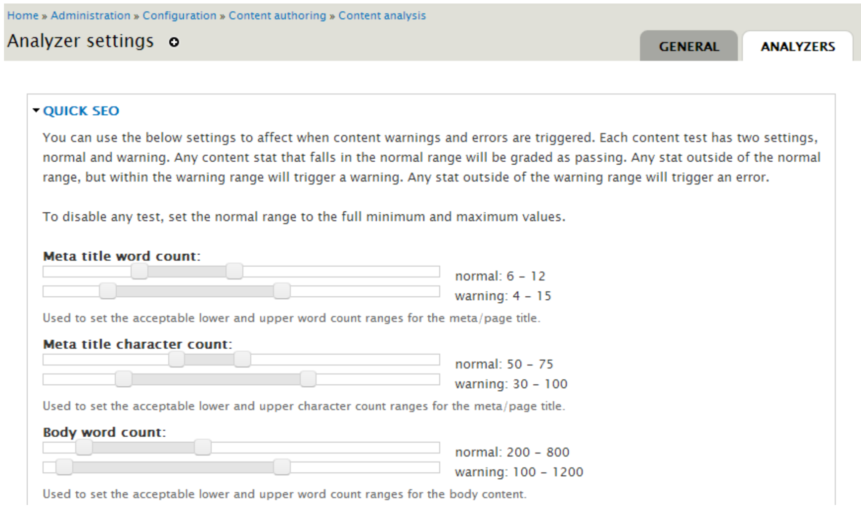
The Content Optimizer module uses the API of Content Analysis to analyze the content of any page on the site, returning a report with recommendations so that the content complies with the SEO good practices.

From the **Analyzers** tab we will access the group of verification that will be carried out during the analysis. From this page we can change the ranges of each parameter. F37.24

F37.24

Content Analysis Analyzers

From the Analyzers tab we can configure the rules added by the additional modules installed, such as Content Optimizer.



When creating or editing a node, a frame with the title **Content analysis report,** will be displayed next to each field affected by a verification rule, where the module will indicate if we should carry out improvements in the field or if it's optimized. F37.25

There will be three types of messages:

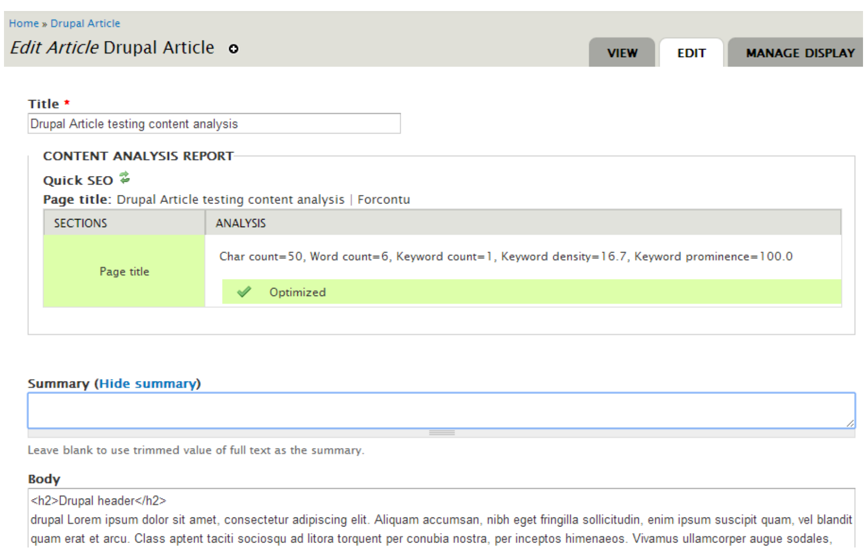
- **Green**, the field is optimized.
- **Yellow**, the field is outside of the parameters. It's a notice of recommendation.
- **Red**, the field is outside of the parameters and it's an alert for the need to modify.

In each frame, next to the subtitle Quick SEO, **an icon with two green arrows** is displayed. Clicking on this icon will update the report for that particular field. We will use it once the value of a field has been modified, to verify if the new value is optimized.

F37.25

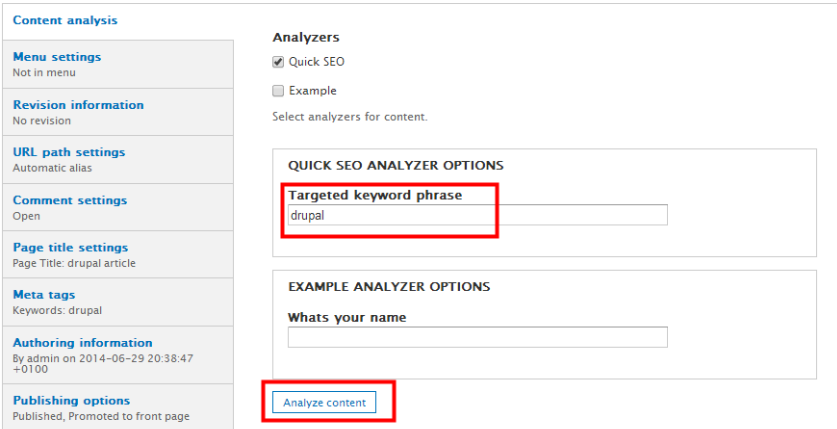
Content Analysis Node Analysis

The analysis indicates, for each field, if it complies or not with the verification rules.



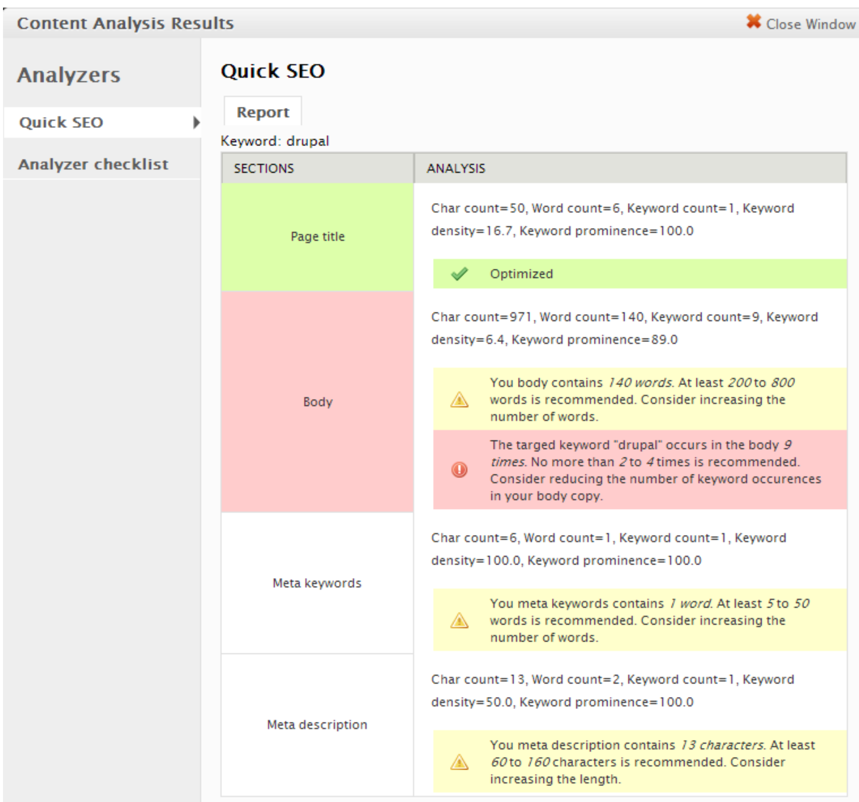
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

From the node option group **Content analysis**, we can launch a complete analysis of the content function of the indicated key words. F37.26



F37.26
Content Analysis
Analyze Content
We can launch a complete analysis of the content function of the key words indicate.

This report will open up in a superposed window. The tabs and content of the report will depend on the additional modules installed that like **Content Optimizer** make use of the **Content Analysis** API. F37.27



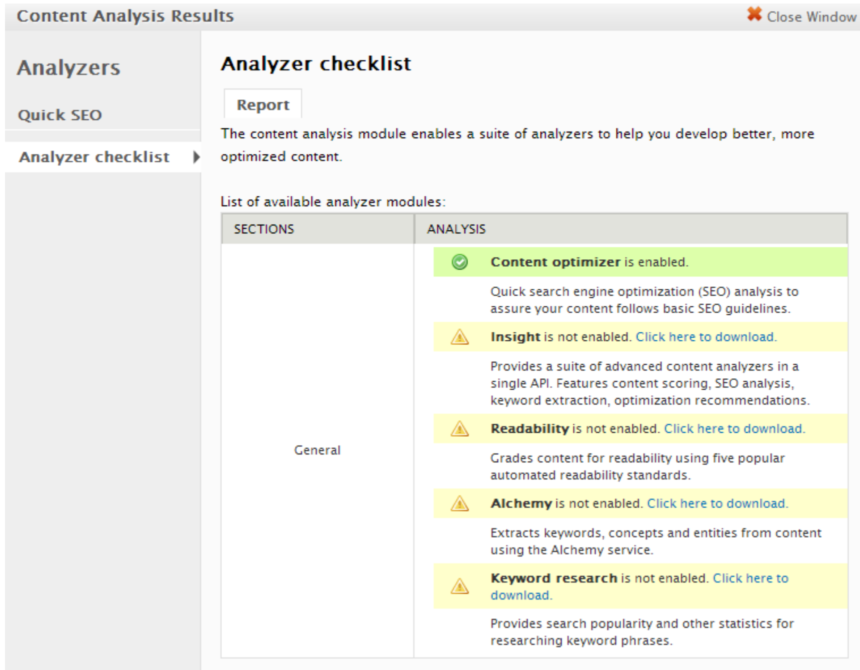
F37.27
Content Analysis
Content Analysis
The report opens up in an emerging window. This report can have different tabs in function of the analysis modules installed.

From the **Analyzer checklist** tab the Content Analysis module verifies that additional modules are activated and make use of their API. Other modules are displayed, that can be installed to widen the analysis and improve the optimization of the site content from the SEO positioning pint of view. F37.28

F37.28

Content Analysis Analysis Module

In the Analyzer checklist tab, a list is displayed with the analysis module installed (those that use the Content Analysis API).



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

38 Social tools: voting and sharing content

In this unit we will go over some social tools that we can incorportate into a website using additional modules.

In the **Vote Content** paragraph we will see modules:

- **Fivestar**, which adds a voting system typically represented by 5 stars.
- **Rate**, which adds other voting types like Like/Dislike, +1/-1, Yes/No, etc.

User **Scoring Points**:

- **User Points**, with other additional modules allows users to win or lose points according to their actions.

To share content on social networks we study modules:

- **ShareThis**, which adds a bar of buttons to share content in several social networks.
- **Service Links**, an alternative which uses buttons to share content on social networks and other online tools.
- **Facebook Like Button**, which adds the Facebook Like button.

Lastly we analyze some modules that enable **communication within the site's users**:

- **Shoutbox**, allows users to send short messages.
- **Chat Room**, enables chat rooms.
- **Privatemsg**, sends internal messages between users.

Comparative D7/D6

Some of the modules in this unit are still under development in Drupal 7.

In Drupal 6 we find important changes in some of them, like Fivestar, where configuration of the voting control is carried out in Drupal 7 via a Fivestar field added by the module.

Unit contents

38.1 Content voting 544
38.2 User scoring 556
38.3 Share in social media 561
38.4 Communication tools 566

38

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

38.1 Content voting

We start this unit by analyzing two modules for content voting: Fivestar and Rate. We accomplish this by two practical cases which will walk us through the process of installation and configuration.

Practical case 38.1 Voting content with Fivestar

The **Fivestar** module allows to add a voting system for the content published. This voting system is typically represented using 5 stars, where 1 star is the lowest value (very bad) and 5 stars is the highest possible value (excellent). In this practical case we will study other configurations.

Step 1. Installation of necessary modules.

The **Fivestar** module is available at:

http://drupal.org/project/fivestar

This module depends on the **Voting API** module, available at:

http://drupal.org/project/votingapi

Next we will install and activate both modules.

Step 2. Configuration of Voting API

Voting API allows for configuration of all the modules that make use of its API, like **Fivestar**.

The Voting API configuration can be accessed at: F38.1

URL Voting API
/admin/config/search/votingapi

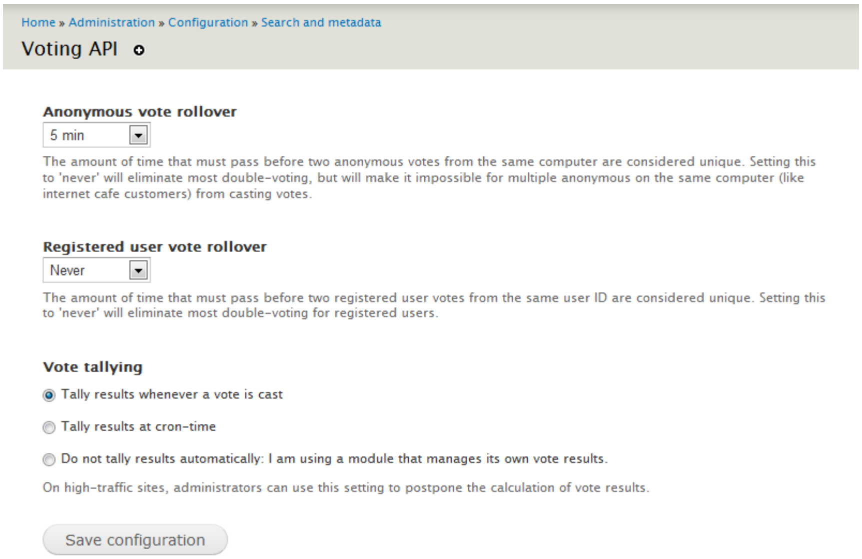
Administration ⇨ Configuration ⇨ Search and metadata ⇨ Voting API

The first parameter to define is the **Anonymous vote rollover**. It is about the time that must lapse for votes from anonymous users to be counted from the same IP, in order to prevent repetitive anonymous voting. That is, if we enter a value of 1 day, we are setting that voting will not be allowed from the same IP for an anonymous user. We will set a 5 minutes value.

The **Vote tallying** option indicates how cast votes shall be counted:

- Tally results whenever a vote is cast.
- Tally results at cron-time
- Do not tally results automatically: I am using a module that manages its own vote results.

We will activate the first option, so votes will be counted immediately.



F38.1

Voting API

Voting API configuration, Fivestar base module.

Step 3. Fivestar configuration

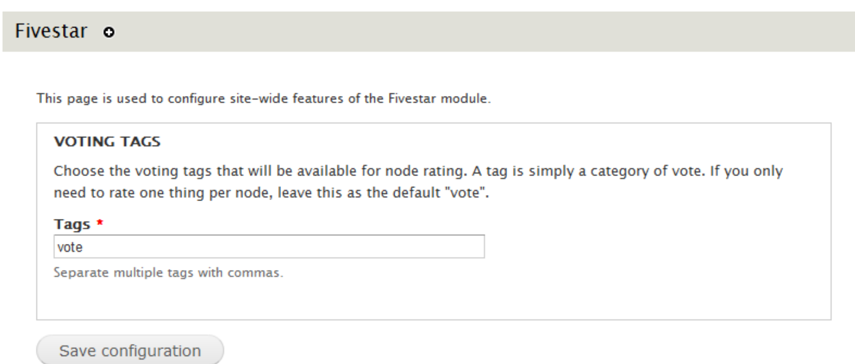
Fivestar configuration is available at:

Administration ⇨ Configuration ⇨ Content authoring ⇨ Fivestar

URL Fivestar
/admin/config/content/fivestar

In Drupal 7 Fivestar configuration is achieved through fields added to entities. We will not establish a default configuration like in Drupal 6, so in the configuration area of the module we will only find a **voting tags** option. F38.2

Each label works like a **voting category**, for example if we want to classify votes according to content. The **vote** term is created by default, in this practical case it will not be necessary to modify this term or to add new labels.



F38.2

Fivestar configuration

We only have to set labels or voting categories at the module's configuration page.

Step 4. Configure permissions.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

For users to be able to vote content we must configure permissions in Fivestar. The module only has one permission: **Rate Content**, in the **Fivestar** group. We will activate this permission both for anonymous and registered users.

Step 5. Add Fivestar field to content type

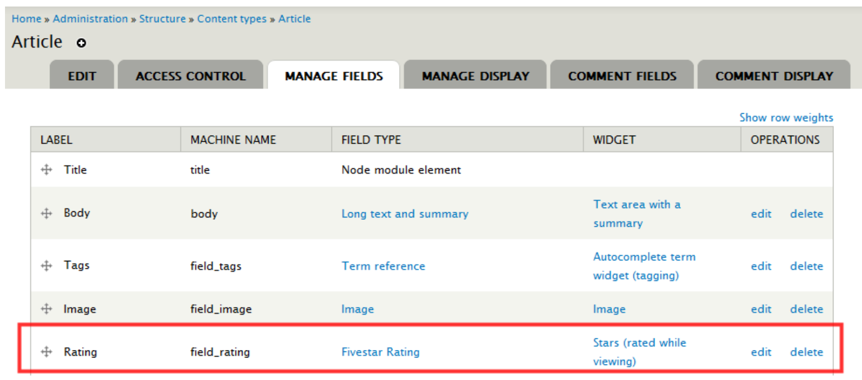
Lastly we must configure which content types can be voted on and how to show the voting system for each one.

We anticipated earlier that the way Fivestar works in Drupal 7 has changed considerably with respect to Drupal 6. In Drupal 7 the voting element is added to the content type (or entity) like a **Fivestar Rating type field.** F38.3

F38.3

Fivestar widget

The widget is added to an entity through a Fivestar Rating field.



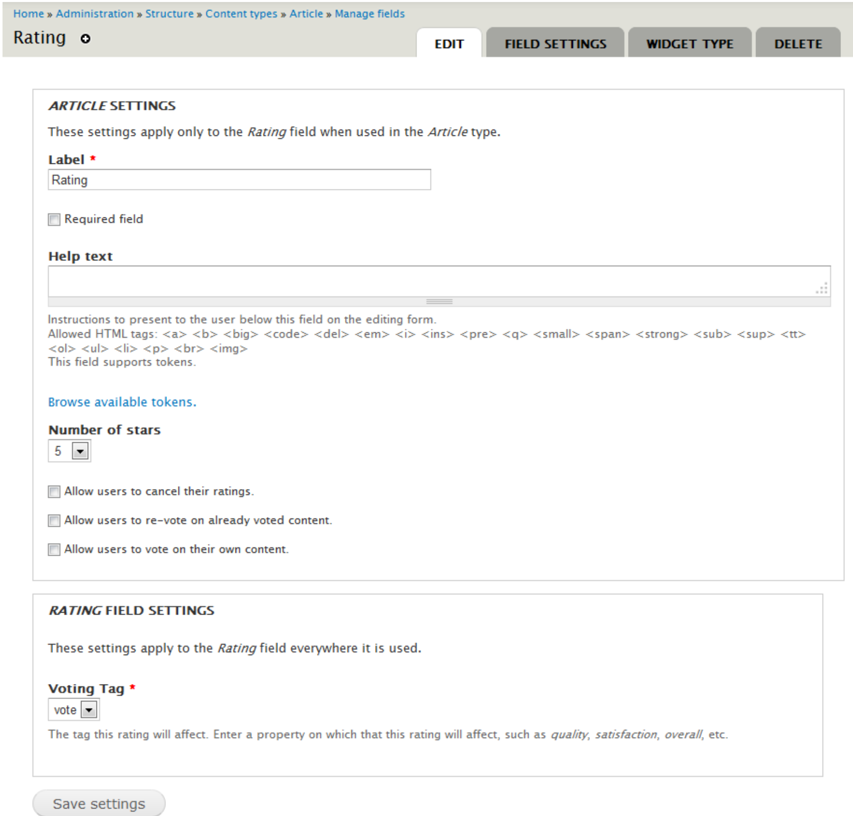
Upon creating the field we can set the number of stars that the voting control will have (between 1 and 10). We also select the voting label (in this practical case only the *vote* label will be available) and if the user will be able to cancel her vote *(Allow users to cancel their ratings).* F38.4

The **widgets** that we can use in this field are:

- **Stars (rated while viewing)**. Users who are viewing a content can rate it if they have the right permissions. This is the default option and the usual operation.

- **Stars (rated while editing)**. In this case voting is performed from content edition, thus only users with edit permissions will be able to vote.

- **Select list (rated while editing)**. Similar to previous but the editor is shown, instead of stars, a select list with the numerical value of each option.



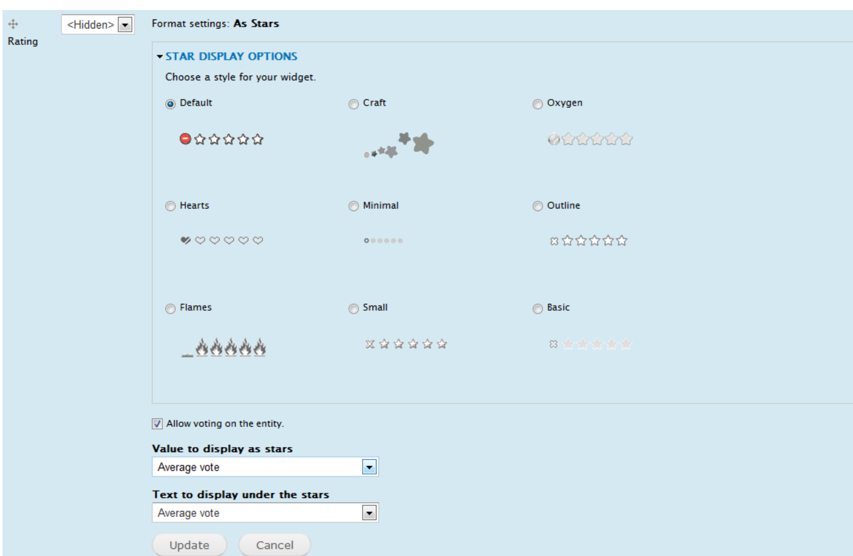
F38.4

Fivestar field

In this field we set the configuration of the vote: stars, tag, category, etc.

Step 6. Configure the Fivestar field presentation

From the **Manage display** tab of the content type we configure the presentation style of the Fivestar Rating field. **F38.5**



F38.5

Field presentation

From Manage Presentation we configure Fivestar's field presentation.

We can choose between the provided sets of images, in addition to indicating what information will be displayed (average vote, user's vote, etc..).

We can select any of the image sets included in Fivestar.

For the voting element to be available in the content type the **Expose this Fivestar field for voting** option must be active.

Lastly, we can determine which elements will be shown as images.

- Average vote.
- User´s vote.
- User´s vote if available, average otherwise.
- Both user´s and average vote. In this case two sets of stars are shown.

And the text to be shown under the stars:

- No text.
- Average vote.
- User´s vote.
- User´s vote if available, average otherwise.
- Both user´s and average vote.

We select **Average vote** in both fields.

Step 7. Vote content

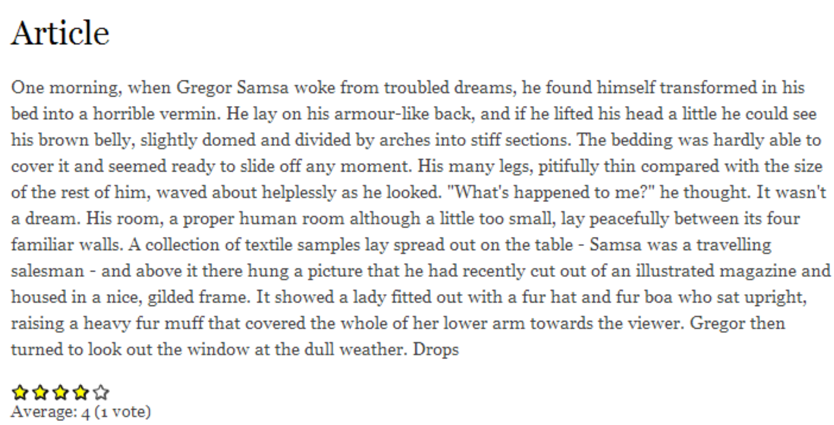
Once the voting system is configured, users can cast votes on any content type where we have added the Fivestar field.

Figure F38.6 shows an example content with the voting element active, showing an average vote.

F38.6

Content of one voting element

A content with an active voting element is shown, displaying the average vote.


Article

One morning, when Gregor Samsa woke from troubled dreams, he found himself transformed in his bed into a horrible vermin. He lay on his armour-like back, and if he lifted his head a little he could see his brown belly, slightly domed and divided by arches into stiff sections. The bedding was hardly able to cover it and seemed ready to slide off any moment. His many legs, pitifully thin compared with the size of the rest of him, waved about helplessly as he looked. "What's happened to me?" he thought. It wasn't a dream. His room, a proper human room although a little too small, lay peacefully between its four familiar walls. A collection of textile samples lay spread out on the table - Samsa was a travelling salesman - and above it there hung a picture that he had recently cut out of an illustrated magazine and housed in a nice, gilded frame. It showed a lady fitted out with a fur hat and fur boa who sat upright, raising a heavy fur muff that covered the whole of her lower arm towards the viewer. Gregor then turned to look out the window at the dull weather. Drops

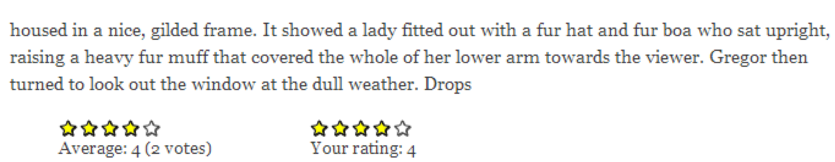
Average: 4 (1 vote)

F38.7

Content with many voting element

Both elements are shown: user's and average vote.

Figure F38.7 shows another example with a different field presentation, both for images and text in the voting elements: **Both user's and average vote**.

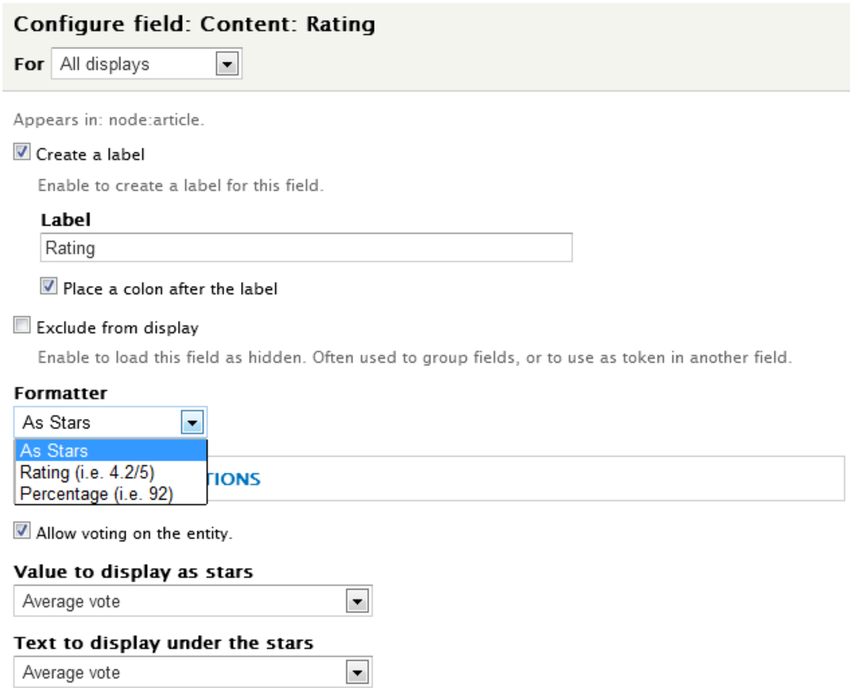

housed in a nice, gilded frame. It showed a lady fitted out with a fur hat and fur boa who sat upright, raising a heavy fur muff that covered the whole of her lower arm towards the viewer. Gregor then turned to look out the window at the dull weather. Drops

Average: 4 (2 votes) Your rating: 4

Step 8. Best voted movies view

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

The **Fivestar** module can be integrated into the **Views** module through the Views field **Content: Vote**. This field can be shown as **stars, score or percentage**. F38.8



F38.8

Views

Configuration of a fiel content type. The vote is included in a view.

Figure F38.9 shows an example of a view with a list of movies sorted by voting average.

F38.9

View with votes

List of movies with the associated voting widgets.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Step 9. Modulos related to Fivestar

If we are using the Fivestar voting system, we can complement it with additional modules.

For example, the **Fivestar Recommender** module adds 4 additional views with the following results:

- **Similar nodes (by votes).** Recommends to the user similar nodes to the one visited. This recommendation is the same for all visitors and is only based on the average score of other nodes.

- **Recommendations (from similar nodes).** Recommends other nodes that could be of interest to the visitor, taking into account other nodes voted by the user with the same score. In this view results are different for each user.

- **Similar users (by votes).** Shows a list of users with the same tastes taking into account the contents voted on by the present user. Results shown are different for each user.

- **Recommendations (from similar users).** Recommends nodes that previous similar users have valued positively. Results shown are different for each user.

Fivestar Recommender is available at:

http://drupal.org/project/fivestar_rec

It depends on **Recommender API module**, available at:

http://drupal.org/project/recommender

This module is charged with implementing the recommendation algorithms used by Fivestar Recommender and other modules.

Practical case 38.2 Voting contents with Rate

The **Rate** module contributes other voting types different from Fivestar. Rate includes 7 types of voting:

- **Thumbs up**. Only allows positive votes (equivalent to Like).
- **Thumbs up/down**. Votes can be positive (Like) or negative (Dislike).
- **Number up/down**. Voting of type +1/-1.
- **Fivestar**. Voting by stars, similar to Fivestar.
- **Emotion**. Shows a list of emoticons (amused, sad, angry, etc.), so that users can choose the best match for their state of mind.
- **Yes/No**. A simple self-explanatory control.
- **Custom**. Allows to create a completely custom control starting from scratch, which can work by scoring points to content, percentages, or options. It is recommended to use first the other controls to become familiar with their operation.

Step 1. Instalation of necessary modules

The **Rate** module depends on the **Voting API** module, already installed in the previous step with Fivestar.

The Rate module is available at:

http://drupal.org/project/rate

The **Voting API** module is available at:

http://drupal.org/project/votingapi

Rate includes two other modules, **Rate** and **Rate Expiration**. **Rate** allows to vote on content, and **Rate Expiration** allows to set expiration dates for votes, so that a deadline is set to vote on a certain content.

It is possible to activate the Rate module without activating Rate Expiration. In order to activate Rate Expiration, we must resolve its dependency on the Date module, which provides Drupal with the necessary tools to handle dates.

The **Date** module is available at:

http://drupal.org/project/date

We will activate both modules, Rate and Rate Expiration, together with their dependencies.

Step 2. Creation of rate widgets

We can add voting widgets in content types and comments

Administration of these widgets is made from:

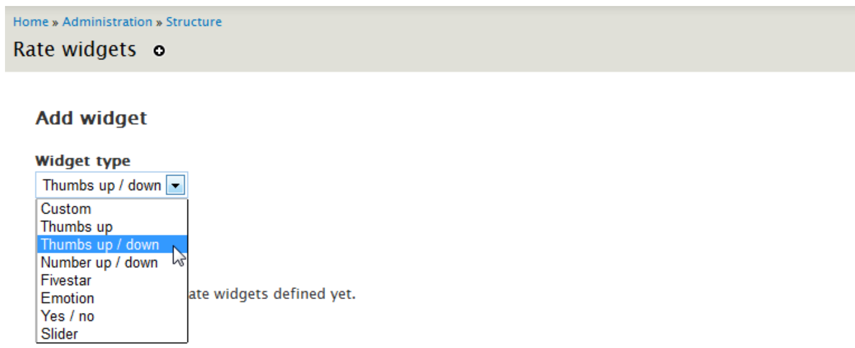
URL Voting widgets
/admin/structure/rate

Administration ⇨ Structure ⇨ Voting Widgets

First we add a widget of type **Thumbs up/down**. F38.10

F38.10
Voting widgets

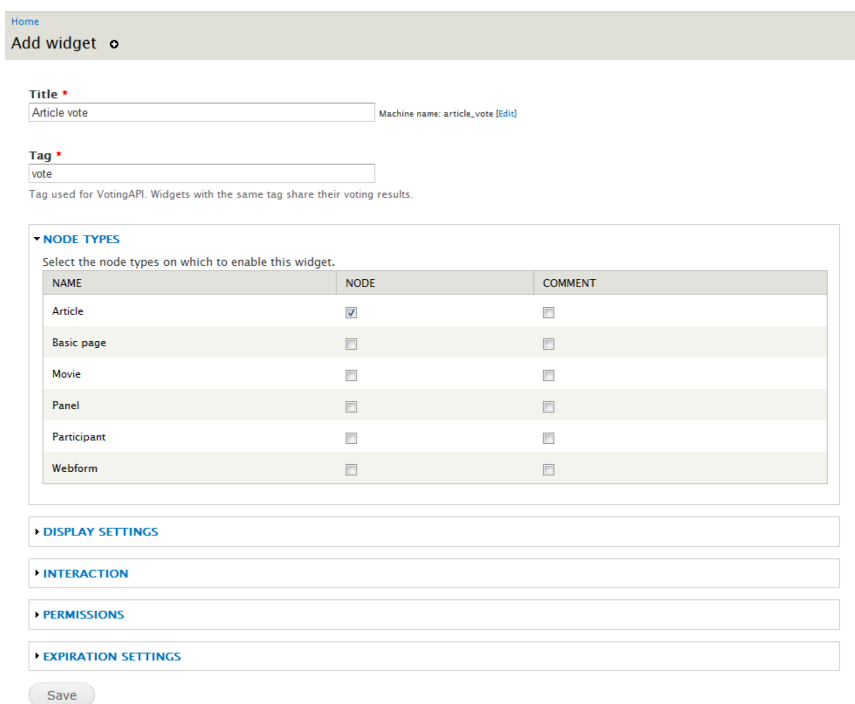
The Rate modules allows to add several voting widgets: Thumbs up/down, +1/-1, Yes/No, Emoticon, etc.



We must enter a **title**, to be used only in administration lists. Similar to Fivestar, the **tag** will store the votes collected by Voting API. If two controls share the same label, votes will be counted jointly. F38.11

F38.11
Add widget
Widget configuration.

We can select in which content types the element will be displayed, both in nodes and comments.

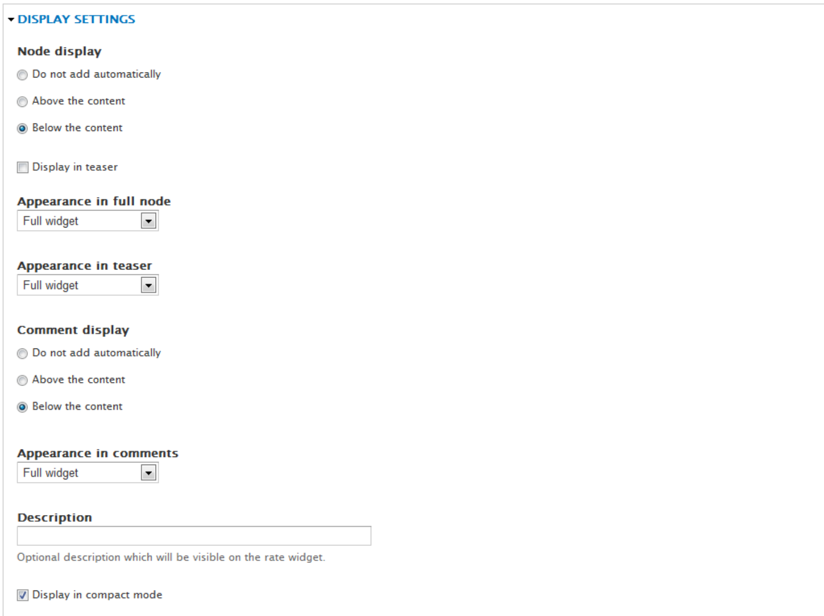


We select within the available content types in which nodes or comments we show the widget. In this practical case we only show the widget created for nodes of type Article.

Step 3. Presentation options

In the **Display settings** section we can control how to show the widget in nodes or comments. By default widgets are shown under content. F38.12

It is possible to add an optional description alongside the widget.



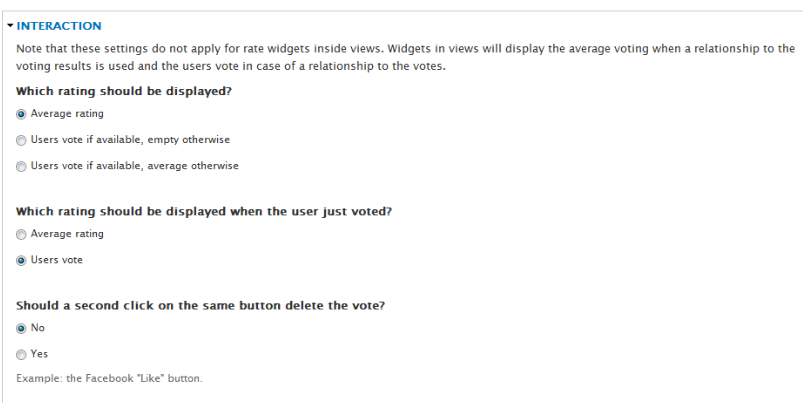
F38.12

Presentation options

Configuration of voting widget presentation both in nodes and comments.

Step 4. Interaction options

From the **Interaction** section we can specify which value will be shown to the user before and after voting (average value or user's vote). F38.13



F38.13

Interaction Options

We can set which value will be displayed on the control before and after the user's vote.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Step 5. Permissions

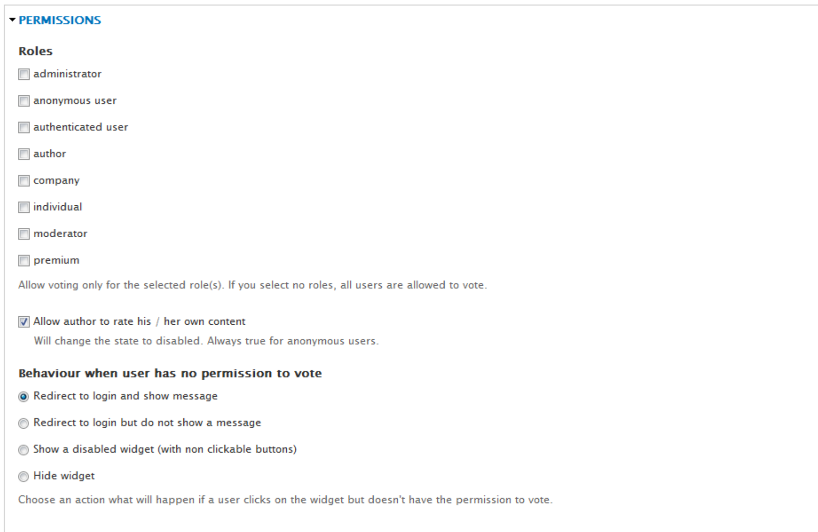
In the **Permissions** section we can set which roles can vote and what will the behavior be when the user does not have such permission. F38.14

For example, we can set to show the widget inactive, so that the user will not be able to vote.

F38.14

Permissions

Which roles can vote and what will be the behavior when the user does not have permission to vote.



Step 6. Expiration settings

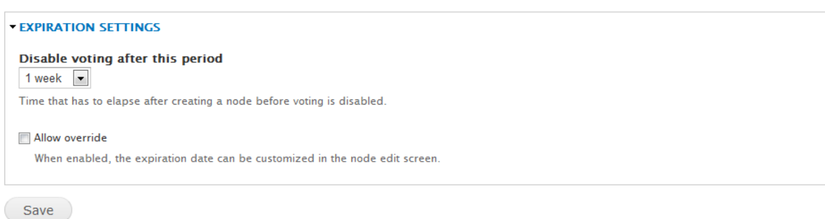
Expiration settings on the voting will only available if we have enabled the **Rate Expiration** module before. F38.15

We select a time period (1 hour, 1 day, 1 week, etc.) of Never if we do not want automatic closure of poll. Activating option **Allow Overwrite** allows for modification of the expiration setting for a particular node.

F38.15

Expiry options

Voting on a content is closed past a certain time period.



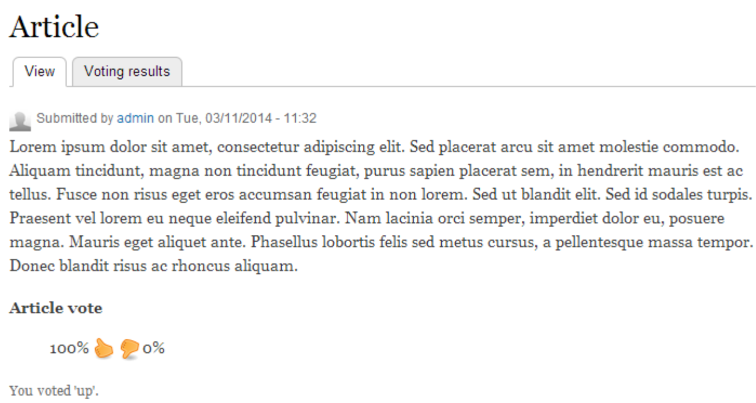
Once the widget is configured we can wrap up its creation by clicking on Save.

The widget will show in the list of poll widgets, we can edit or delete it from there.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Step 7. Vote content

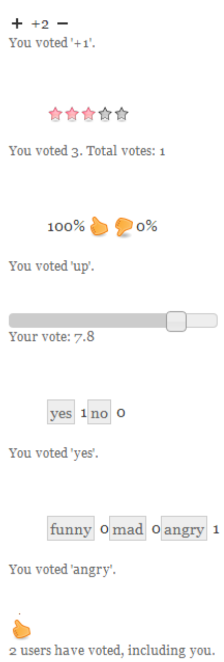
Once the voting widget is configured we can cast our vote from every content with type Article. F38.16



F38.16
Content with vote
This example shows a content with a voting widget of type Thumbs up/down, equivalent to Like/Dislike.

Step 8. Other widgets

We show the rest of available widgets with Rate. F38.17



F38.17
Other widgets
The figure shows widgets allowed by Rate. In order:

- Thumbs up/down
- Number up/down
- Thumbs up
- Yes/No
- Emoticons
- Fivestar

Step 9. Access to results

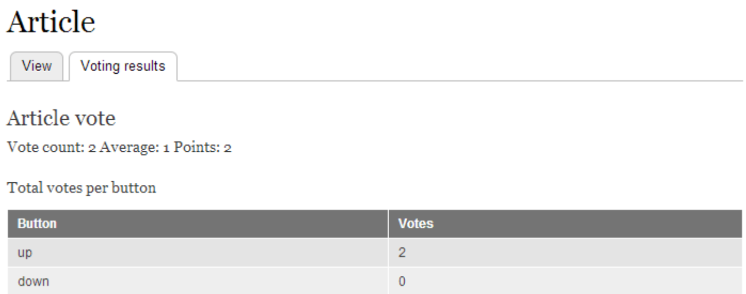
Users with Rate permissions **View rate results page**, will be able to access the vote tally of each node from the **Poll Results** tab. F38.18

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F38.18

Voting results

In contents which have a voting system enabled, a tab is displayed with the results of votes.



38.2 User scoring

The **User Points** module contributes an API so that users can win or lose points according to their actions. Other modules will be charged of implementing the functions that add or subtract points.

The **User Points** module is available at:

http://drupal.org/project/userpoints

As an example of an additional module that uses the User Points API, we will install the **User points Nodes and Comments**, available at:

http://drupal.org/project/userpoints_nc

This module enables adding points to users who publish a node or comment, or moderate comments published by other users.

Module User Points also integrates with Rules, enabling the generation of customized rules to add or subtract points to users according to their actions.

We will actvate the following modules and their dependencies:

- Userpoints.
- Userpoints Nodes and Comments.
- Userpoints rules integration.

Configuration of user points

URL Points
/admin/config/people/userpoints

Configuration and administration of user points is done from:

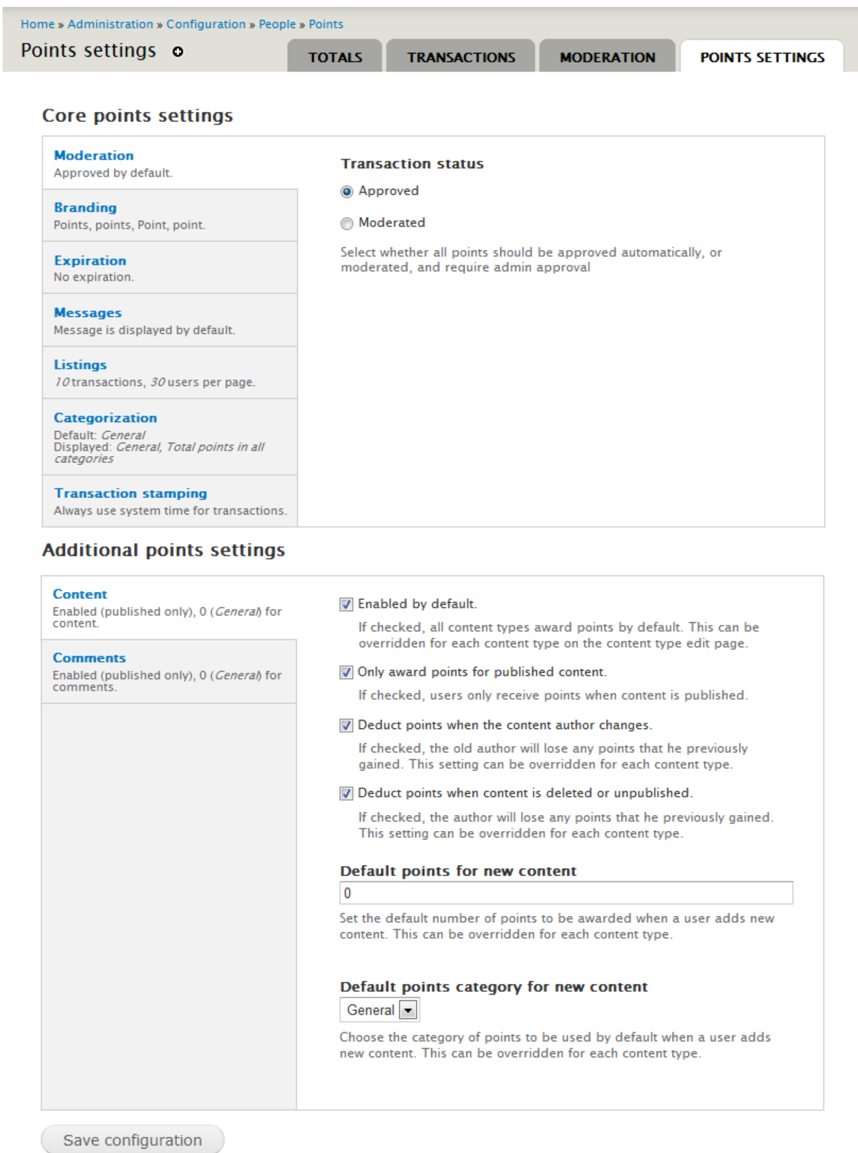
Administration ⇨ Configuration ⇨ People ⇨ Points

From the **Points settings** tab it is possible to configure: F38.19

- **Moderation**. We can set if points will be allocated automatically or if moderation is required. Points that require approval will be shown in the **Moderation** tab.

- **Branding**. Refers to what names points will have.

- **Expiration**. Enables to set an expiration date for points.
- **Messages**. We can set if a message will be sent to the user when points are granted.
- **Listings**. Configuration of user lists and points transactions.
- **Categorization**. The module adds a new vocabulary to taxonomy, called Userpoints. If we add new terms to this vocabulary they will be available in the assignment rules for points with the purpose of categorizing or grouping points. Points can be shown sorted by category in the user's profile.
- **Transaction stamping**. This sets a time stamp for transactions.



F38.19

User Points configuration

Multiple configuration options of the User Points module.

- **Additional points settings: Content**. Default options for content. These options can be modified for a certain content type, they set points granted for creating new content, associated category, etc.

- **Additional points settings: Comments**. Default options for comments. The options can be modified for a certain content type, they set points for comments, the associated category, etc.

Configuration of content types

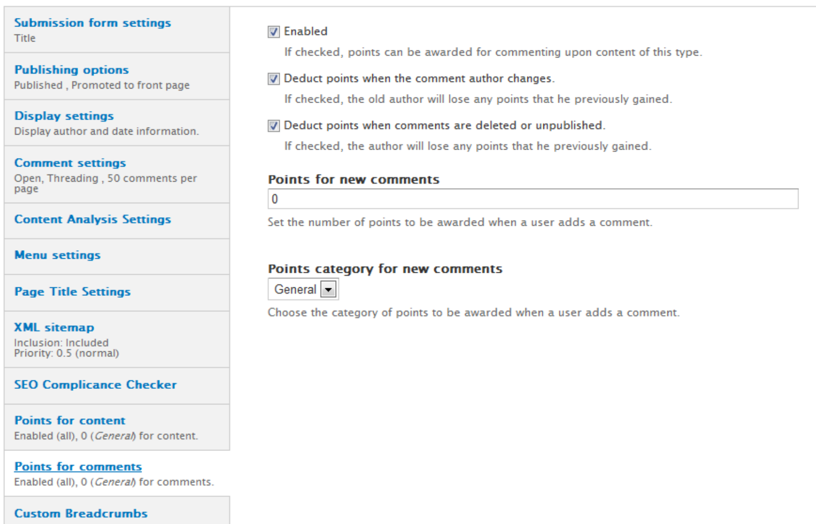
From each content type we can configure, both for nodes and comments: F38.20

- If the points system is active.
- If points are withdrawn from the user when the author of the content changes.
- If points are withdrawn from the user when the content is unpublished.
- Points to be granted upon creation of a new user.
- Category where points will be added.

F38.20

Content type configuration

In each content type we can set the configuration of User Points when new content or new comments are created.



Lists of user points

Back to configuration of user points: F38.21

URL Points
/admin/config/people/userpoints

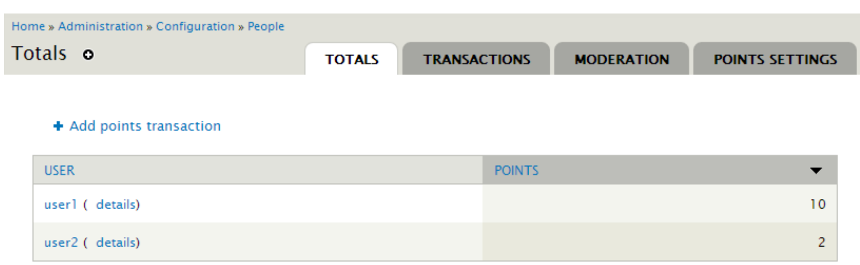
Administration ⇨ Configuration ⇨ People ⇨ Points

From the **Totals** tab we can check the total of user points.

F38.21

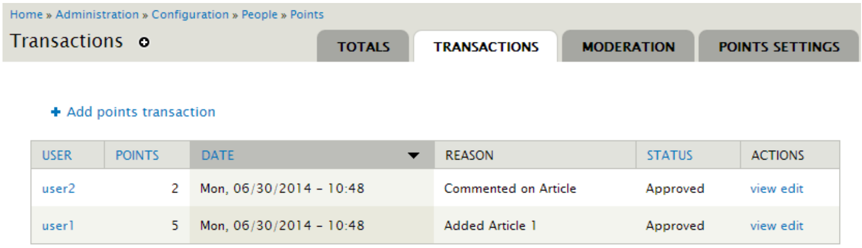
Total points

List of users and total points scored.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

In the **Transactions** tab we can see the details of point transactions, both for addition and subtraction. F38.22



F38.22

Transactions

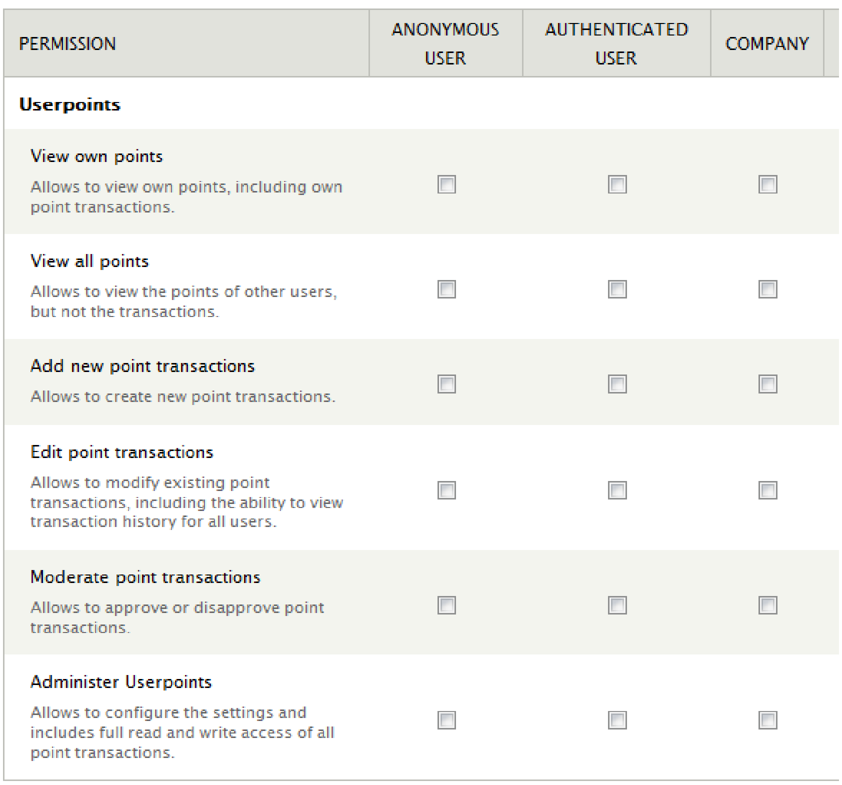
Detail of point transactions per user, both positive and negative.

We can edit each transaction or create new point transactions manually (**Add points transaction**).

Permissions

Available permissions to configure User Points are: F38.23

- View own points.
- View all points.
- Add new point transactions.
- Edit point transactions.
- Moderate point transactions.
- Administer Userpoints.



F38.23

Userpoints permissions

Set of permissions available to configure access to the various functionalities of the module.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Integration with Rules

By installing **Rules** together with the **Userpoints rules integration** module, we can create rules to add or subtract points to users.

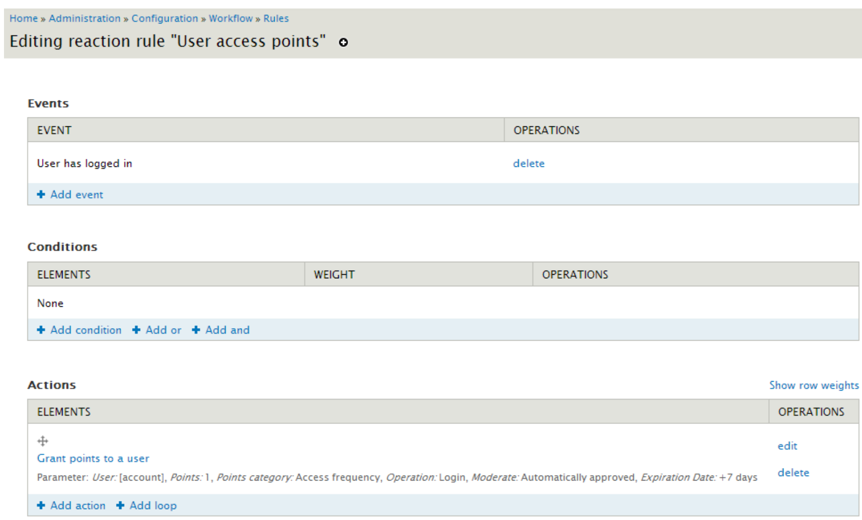
In **Rules** we will have two new events at our disposal:

- **User has been assigned points**. Executes after granting (or withdrawing) points.
- **User will be assigned points**. Executes before granting (or withdrawing) points.

and two actions:

- **Load points of a user**. Gets the user's points. It can get the total or the subtotal of a specific category.
- **Grant points to a user**. From this action it is possible to grant (positive action) or withdraw (negative action) points.

As an example the **User access points** rule has been configured to add 1 point to users each time their login on the site. These points are added to the **Access frequency** category (new term added to the **Userpoints** vocabulary), and expires after 7 days. F38.24 F38.25



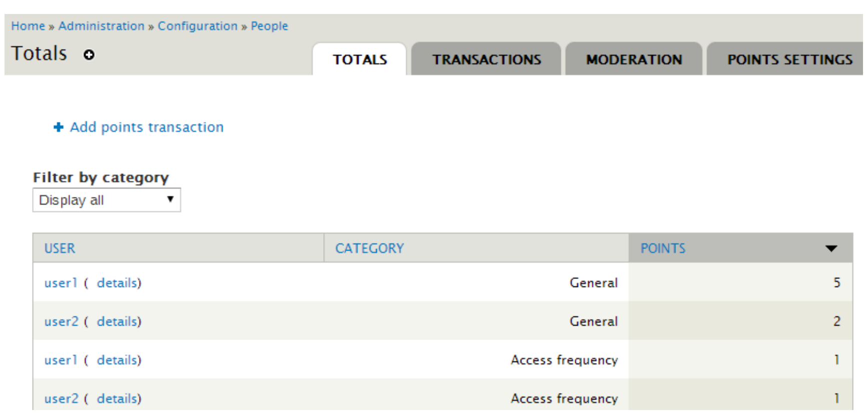
F38.24

User Points rule

Example of rule using User Points system to award points to the user.

Users receive 1 point every time they log in to the site. These points expire after 7 days.

Points are added on an additional category called Frequent Access.



F38.25

Total points

Total points grouped by user and category.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Share in social media

38.3

ShareThis module

The **ShareThis** module adds a toolbar to share content in social media.

The ShareThis module is available at:

http://drupal.org/project/sharethis

Once activated, we will find the configuration option directly through URL /admin/config/sharethis. We can also access it through the administration index /admin/index, by selecting option **ShareThis Settings**.

From the configuration page we can set the following options: F38.26

- **Choose a widget type**:
    - Multi-Post widget. The user can share content in multiple social tools without leaving the widget.
    - Direct-Post widget. This option redirects the user to the selected social tool, where he can finish to publish his shared content.
- **Choose a button style**:
    - Large Chicklets.
    - Small Chicklets.
    - Classic Buttons.
    - Vertical Counters.
    - Horizontal Counters.
- **Choose your services**. We select the social services to share content. In the left column the selected services are shown, and in the right column the rest of available services.

  To move services between columns we select the services and then click on the matching date.

  We can also rearrange services with the up and down arrows.
- **Select the content types.** Lastly we select the content types where the ShareThis toolbar will be shown.

F38.26

ShareThis configuration

ShareThis module configuration options

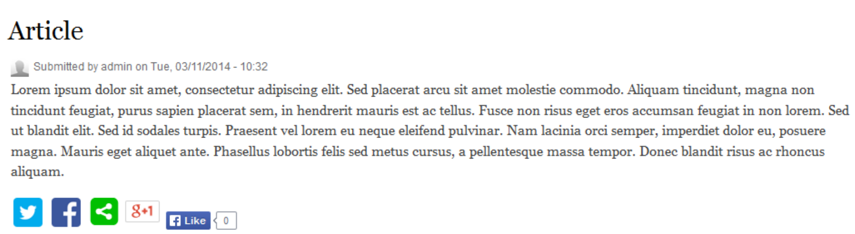
Figure **F38.27** shows the **ShareThis** toolbar.

F38.27

Content with ShareThis

Content that we can share using ShareThis buttons.

Next to each application the number of times its content has been shared is shown.



Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Módulo Service Links

The **Service Links** module is an alternative to share contents across different social networks.

The module is available in:

http://drupal.org/project/service_links

Service Links includes several general purpose modules and, independently, multilingual services. We will activate the following modules.

- **Service links**. Contributes the engine to integrate services.
- **General Services**. General services (international).
- **Widget Services**. Widgets for Facebook, Google Plus, Linkedin, Twitter, etc.

Once installed, we access the configuration of Service Links from:

Administration ⇨ Configuration ⇨ Web services ⇨ Service Links

URL Service Links
/admin/config/services/service-links

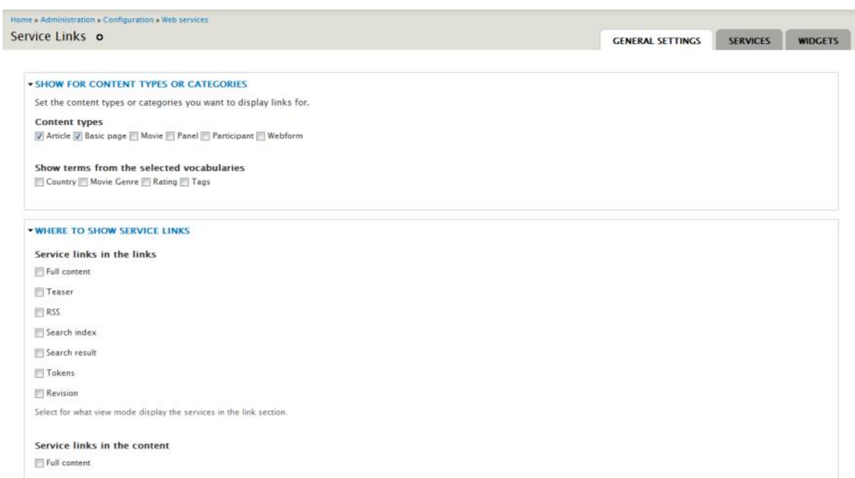
From the **General Settings** tab we can configure the following options: F38.28

- **Show for content types or categories.** We select the content types and taxonomy vocabularies where links will be shown.

- **Where to show service links**. Service links can be shown in the Links section of the theme or after the content.

- **Page specific visibility settings**. Settings to show or hide in certain pages.

- **How to display service links**. For example we can set if links will be shown as images or also text with the name of the services.



F38.28

Service Links configuration

From the general configuration of the module we can establish in which content types and taxonomy categories the service links bar will be displayed.

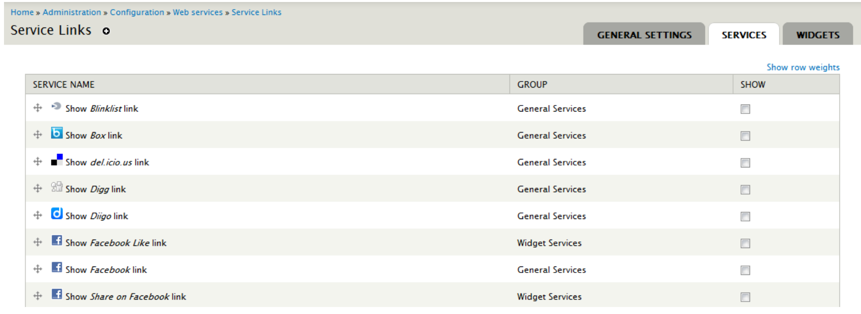
From the **Services** tab we can set which services will be shown in the links bar. The list of services will depend on installed modules. All services will be inactive by default. F38.29

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F38.29

Services

From available services we activate those that will be displayed in the link bar.

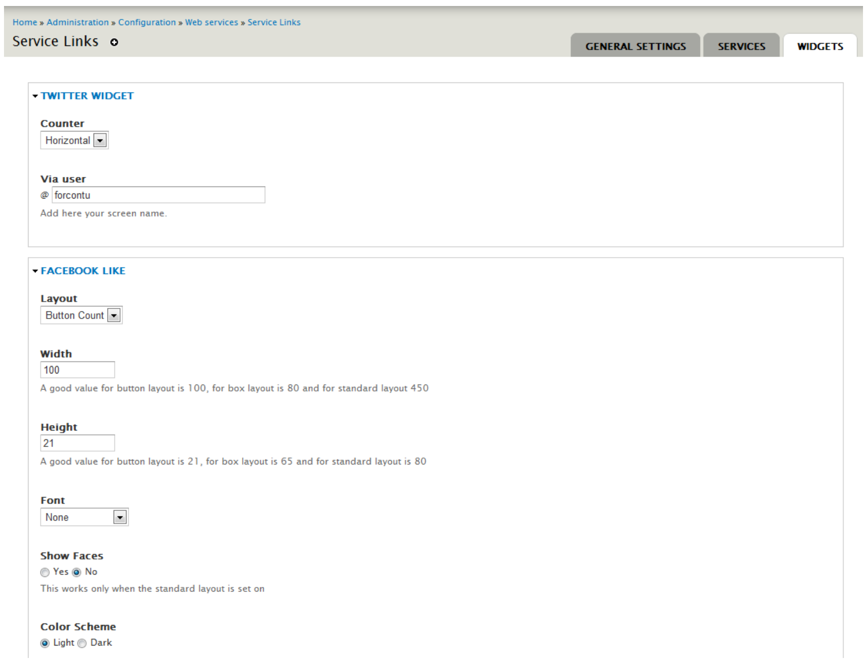


From the **Widgets** tab we can configure the buttons of some of the services, not all services have additional configuration. F38.30

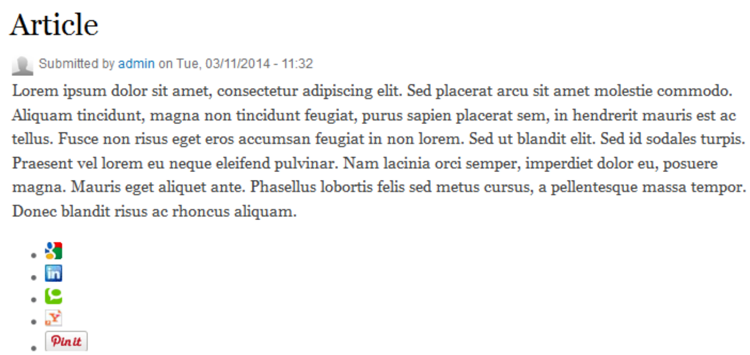
F38.30

Controls

Some of the services may require additional configuration.



For the service links bar to show in the set content types, the user must have been assigned the **Access to Service Links**. F38.31



F38.31

Content with Service Links

Example content with an active links bar.

Facebook Like Button module

The **Facebook Like Button** module adds the Like Facebook button to each node. This functionality. This functionality is included in the previously shown modules.

The **Facebook Like Button module** is available at:

http://drupal.org/project/fblikebutton

Once installed, we access configuration from:

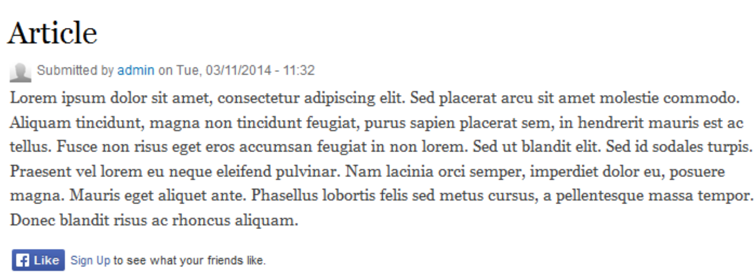
Administration ⇨ Configuration ⇨ FB Like Button ⇨ Opciones generales

URL FB Like Button
/admin/config/fblikebutton/general

It is possible to set in which content types will the button appear, height and width, style, language, etc.

The module also includes the **FB Like button** block, oriented to show a general **Like** for the site.

Figure F38.32 shows the **Like** Facebook button added to an Article content type.



F38.32

Facebook Like Button

Example content to which we added the Facebook Like button. The number of people who liked the content is shown alongside the button.

38.4 Communication tools

Shoutbox module

The **Shoutbox** module shows a block where users can publish short messages.

The module is available at:

http://drupal.org/project/shoutbox

Shoutbox includes, among others, the following modules:

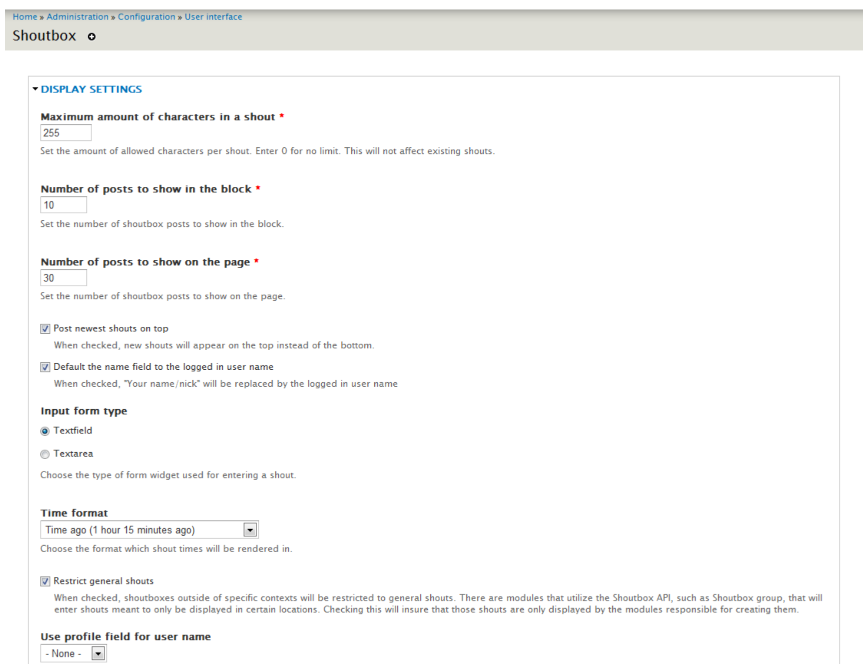
- **Shoutbox**. Provides the page and the block for the messages.
- **Shoutbox points**. Integration with User points.
- **Shoutbox rules**. Integration with Rules.

We can access Shoutbox administration from **/admin/index** (Shout box) or directly via URL **/admin/config/shoutbox**.

F38.33

Shoutbox configuration

Presentation options of the Shoutbox module.



In **Display settings** we can configure: F38.33

- Maximun amount of characters in a shout.
- Number of posts to show in the block.
- Number of posts to show on the page.
- Post newest shouts on top.
- Input form type (textfield or text area).
- Time format.
- Use profile field for user name: It can be use to select a profile field to write the user's first name instead of its username.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

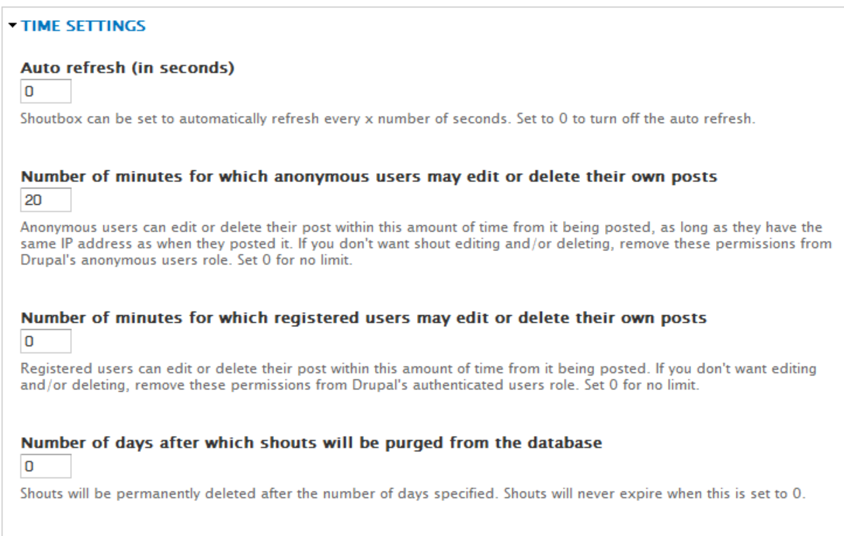
From **Time Settings** we can configure the following parameters: F38.34

- **Auto refresh (in seconds)** of the block or page. If the value is zero this functionality will not be active. It is recommended to set a value greater than 0, so the block will refresh and allow for conversations between users.

- Number of minutes for which anonymous users may edit or delete their own posts.

- Number of minutes for which registered users may edit or delete their own posts. If value is set to 0 there will be no limits, thus a registered user will be able to modify his messages anytime.

- Number of days after which shouts will be purged from the database. 0 value disables deletion of messages.



F38.34

Shoutbox configuration

Time options to configure parameters such as the update rate of the page.

The module provides the following **permissions** to configure access to functionalities: F38.35

- Administer shoutbox.
- Moderate shouts.
- View shoutbox.
- Post shouts.
- Skip shout approval.
- Edit own shouts.
- Delete own shouts.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F38.35

Permissions

Permissions set to configure access to Shoutbox.

PERMISSION	ANONYMOUS USER	AUTHENTICATED USER	ADMINISTRATOR
Shoutbox			
Administer shoutbox Administer shouts and shout settings.	☐	☐	☑
Moderate shouts Enable/Disable posts from other users.	☐	☐	☑
View shoutbox	☑	☑	☑
Post shouts	☑	☑	☑
Skip shout approval	☑	☑	☑
Edit own shouts	☐	☑	☑
Delete own shouts	☐	☑	☑

Once permissions are configured we will be able to activate the Shoutbox block or access page **/shoutbox**. In either case the same messages will be shown. F38.36

F38.36

Shoutbox block

The Shoutbox block running.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Privatemsg module

The **Privatemsg** module enables the sending of private messages between users.

This module is available at:

http://drupal.org/project/privatemsg

Privatemsg is a composite of several modules. Some of them are:

- **Private messages**. The main module that enables the sending of private messages.

- **Privatemsg Email Notification**. Sends an email notification to the recipient of a private message.

- **Privatemsg filter**. Enables filtering of messages.

- **Privatemsg roles**. Enables sending of messages to all users with a certain role.

- **Privatemsg Limits**. Sets limits to the sending of messages.

- **Privatemsg Rules Integrations**. Integration with the Rules module to perform certain actions when a private message is sent.

This module is configured from:

Administration ⇨ Configuration ⇨ Messaging ⇨ Private message settings

URL Private messages
/admin/config/messaging/privatemsg

Private messages are a new entity, so we can manage fields and presentation.

From the **Private message settings** tab we configure the following group of options, which can vary according to the modules that are active:

- **Message lists**. Options related to how a list of messages is shown is shown to users.

  - Threads per page.
  - Fileds that wil be shown. At a bare minimum the Subject and Last Update fields will be shown.
  - Predetermined list. Inbox for all messages.
  - Number of messages on thread pages.

- **Display**. Enables selection of a template to show private messages, inform the user of new messages upon session start, etc.

- **Links**. Ooptions related to links for private messages sending. For example, links can be shown in the profile page or on published nodes, in order to send a message to the author. In this case we will select in which content types the link will be shown.

- **Deleted messages**. By default deleted messages are only hidden. From this tab it is possible to configure the interval for permanent deletion.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **E-mail notify**. Configuration of email message to be sent to users upon reception of a private message.

From the **Límits** tab we can set limits in the private messages sending system. Some options are:

- **Maximum number of messages/conversations per time period**. It is possible to limit the number of messages as well as the time period (for example, 3 messages per hour).

- Limit the **Maximum number of recipients per message.**

Basic **permissions** that enable reading and writing of private messages are:

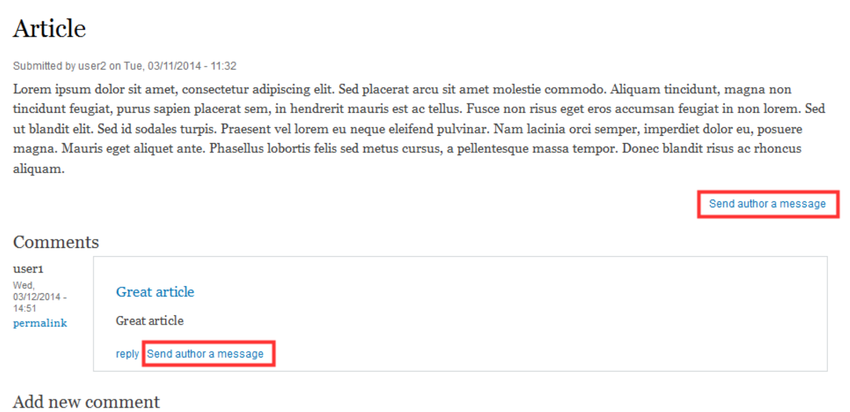
- **Read private messages**.

- **Write new private messages**. The user will be able to write new messages and answer to previously sent messages.

- **Reply to private messages**. This permission allows to answer private messages but not write new ones, unless the previous permission is also active.

- **Delete private messages**.

Figure F38.37 shows links **Send author a message** both in the node and in comments.



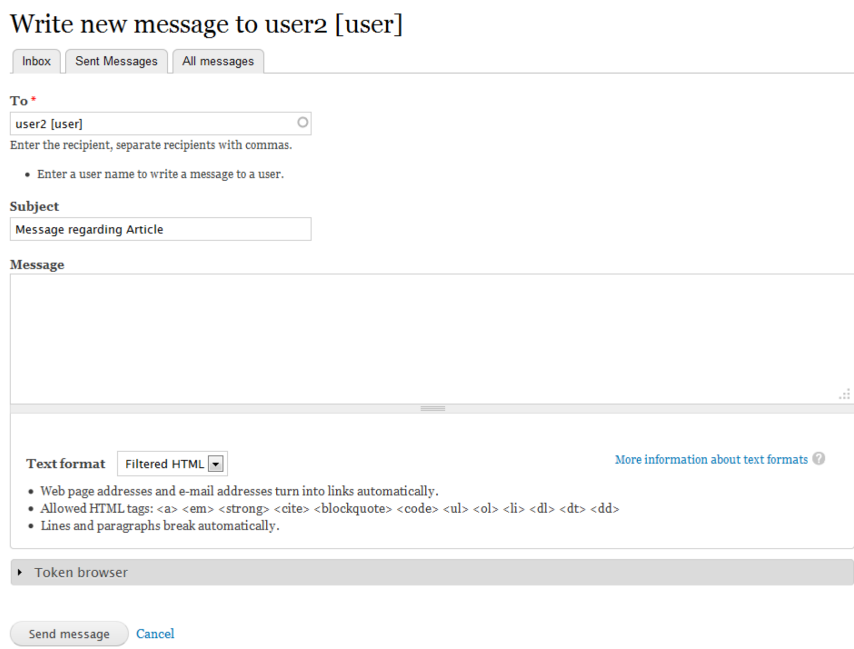
F38.37

Sneding private messages

The Privatemsg module allows for communication via private messages between users of the site.

For example, this system allows a user to get in touch with the author of a node or comment.

By clicking on some of the links we will be sending a private message to the matching user, author of a node or comment. By default the private message will be assigned a recipient and a subject. We can modify these values including several recipients (separated with commas). F38.38



F38.38

Composing a private message

If we start with a Send Message to Author link, the submission form will already have assigned a message recipient (To field) and a subject. We just have to type the message and click on Send Message.

If we have activated email notification, the user who receives the private message will also receive an email notification. Independently of this notification, the system will show new messages in the user's menu, next to the link to the inbox. F38.39

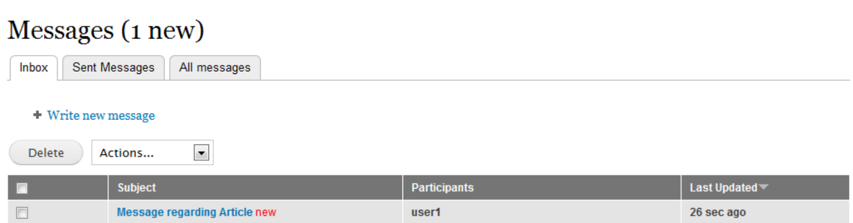
F38.39

New messages

Access to the messages inbox.

The **Inbox** shows received messages like an email manager. F38.40

Apart from reading the message we can delete it, mark it as read or unread and archive it.



F38.40

Inbox

Messages are managed as email, they can be deleted or archived.

Upon clicking on the message users will be able to, apart from reading the message, answer it by completing the text area and clicking on **Send message**. The main message and subsequent answers are stored like a unique conversation. F38.41

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F38.41

View message

By viewing the contents of the message we can send a reply.

Messages and responses are stored as a single conversation.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

39 Newsletter Subscription

In this unit we will study the Simplenews module as a solution for creating email newsletter subscriptions.

Simplenews allows you to create distinct newsletters or categories of newsletters that users, both registered and anonymous can subscribe to. The newsletters created are associated with a category so that only those users who have subscribed to that particular topic will recieve them.

Simplenews allows emails to be sent in HTML format, in order to do so you will need to install the Mime Mail plugin.

Although Simplenews can be complemented by various contributed modules, there are still few additional modules that have a stable release for Drupal 7.

Comparative D7/D6

Simplenews

Simplenews is very mature module with many additional contributed modules. However, the adaptation of the module to Drupal 7 has been slow and many of these modules do not have a stable version for Drupal 7.

Unit contents

39.1 Simplenews installation and Configuration 576
39.2 Subscribing to newsletters 581
39.3 Create and Send newsletters 588

39

39.1 Simplenews installation and Configuration

The **Simplenews** module is used to publish and send newsletters to users who have previously subscribed to them.

Simplenews keeps various subscription lists. Both registered and anonymous users can subscribe to newsletters by simply providing a vaild email address.

The Simplenews module is available at:

http://drupal.org/project/simplenews

Simplenews is comprised of two modules:

- **Simplenews**, this is the main module and is required to create subscriber lists and to send newletters to those subscribers.

- **Simplenews rules**, integrates Simplenews with Rules.

If you wish to send newsletters in HTML format, you will need to install the **Mime Mail** module, available at:

http://drupal.org/project/mimemail

Mime Mail, in turn, is dependent on the **Mail System** module, available at:

http://drupal.org/project/mailsystem

From Mine Mail, activate the modules included: Mime Mail and Mime Mail CSS Compressor.

Once we have installed and activated these modules, we can configure the newsletter system from:

URL Simplenews
/admin/config/services/simplenews

Administration ⇨ Configuration ⇨ Web Services ⇨ Simplenews

From the **Settings** tab we can configure the module; the configuration is divided into three tabs: Newsletter, Subscription and Send Mail.

Newsletter configuration

From the **Newsletter** tab we can configure the following parameters: F39.1

- **Format**. Can be Simple (plain text) or HTML. As we have already mentioned, for HTML format operation to work we must install the Mime Mail module.

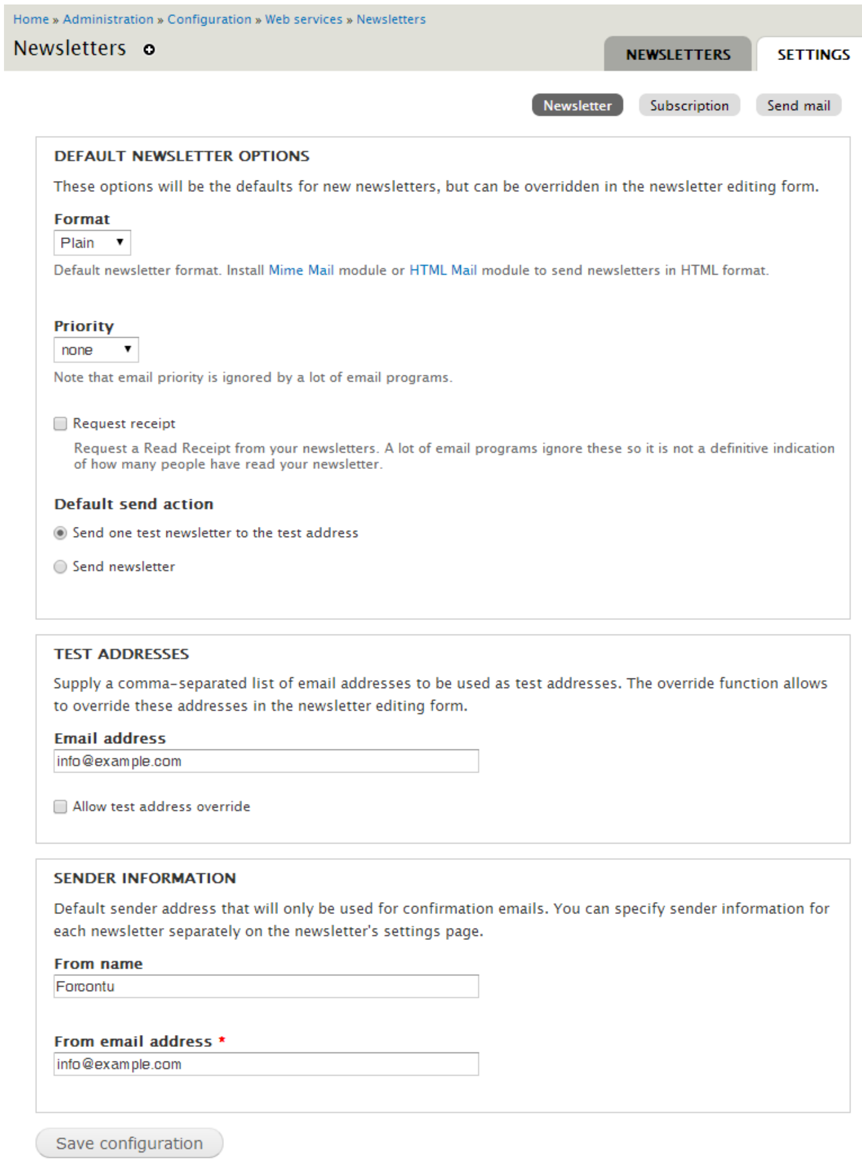
- **Priority** and **Request receipt**. When using these options be aware that they might be ignored by some mail clients.

- **Default send action**. It is recommended that you always send a test news letter to the email address indicated. This will allow you to check the newsletter before it is shipped to users. Leaving the default (send one test newsletter to the test address) helps avoid sending the newsletter before it is complete.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Test addresses**. Comma-separated list of emails to send the test newsletter to. If we allow the test address to be overwritted during newsletter creation, other test addresses can be selected at that time.

- **Sender information**. This is the information used in emails related to membership (additions, deletions and other notices), not for sending the newsletter. As discussed in the following sections, each newsletter will have it own Sender configured when created.



F39.1

Newsletter Configuration

General configuration options for newsletters, such as format, test address and sender information.

Subscription settings

From the **Subscription** tab we are able to configure the following options: F39.2

- **Synchronize with user account**. Subscriptions are synchronized with user accounts, so if you delete or block a user, it will also remove that user's subscriptions to newsletters.

- **Confirmation emails**. Configures the emails sent to confirm subscription requests as well as subscription removal confirmations.

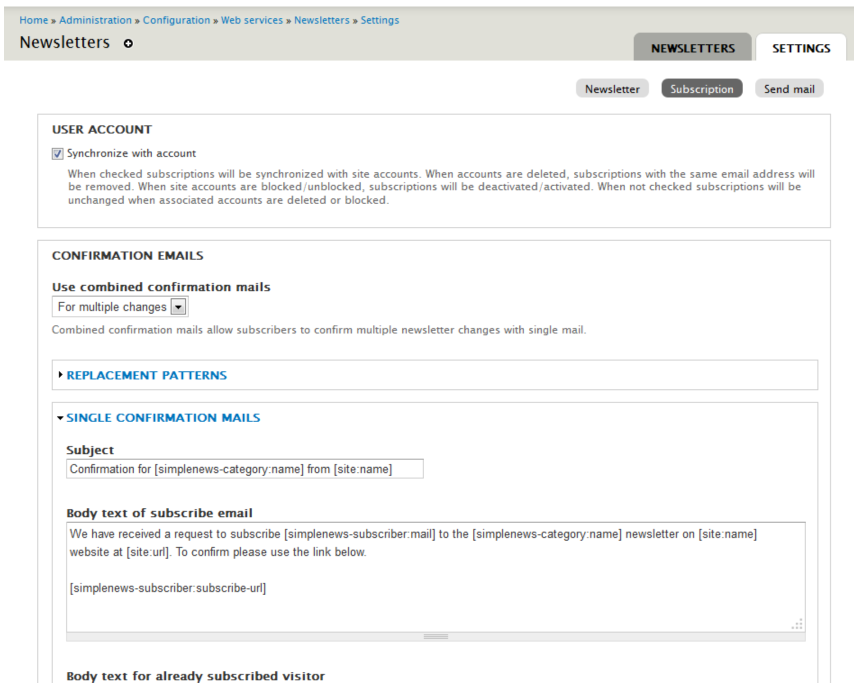
- **Confirmation pages**. We can indicate the URL that the user will be redirected to when confirming the subscription or cancellation. If no URL is indicated the user is redirected to the site home page.

F39.2

Subscription Configuration

Options related to user subscriptions.

From this page you can configure the confirmation messages sent to the user when subscribing or unsubscribing to a newsletter.



Send Mail Configuration

From the **Send Mail** tab we are able to configure the following options: F39.3

- **Use cron to send newsletters**. By activating this option (recommended), the newsletters will be sent when cron executes. Only test newsletters and confirmation emails will be sent automatically.

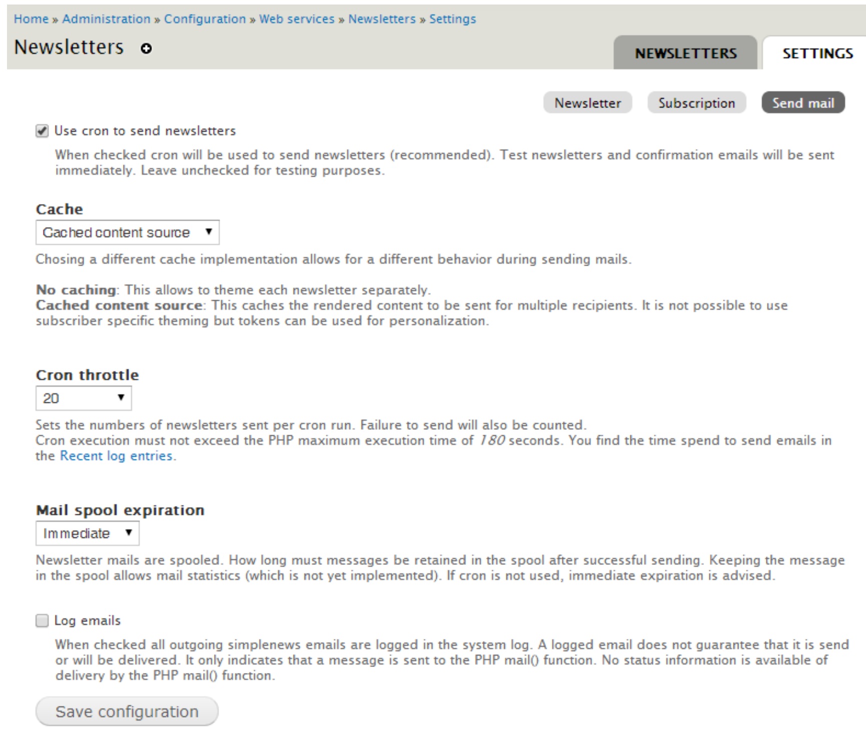
- **Cache**. System Cache is used. Using the system cache, the newsletter content will only be prepared once for all recipients, so that the final design will be the same for all users, although small variations in content is allowed through wildcard replacement.

- **Cron throttle**. Indicates the number of emails sent per cron execution.

- **Mail spool expiration**. By default, once the emails are sent they are removed from the queue. By modifying this parameter, we can keep the emails in the queue for a set time, usually so they can be reviewed.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Log emails**. If you activate this option, sent mails are stored in the system log.



F39.3

Send Mail Configuration

Send mail related options. The test and confirmation emails are sent immediately, but newletters wait for cron execution.

Mime Mail configuration

As already discussed, in order to send emails in HTML format, installing Mine Mail is required, we can configure Mime mail here:

Administration ⇨ Configuration ⇨ System ⇨ Mime Mail

URL Mime Mail
/admin/config/system/mimemail

Note that Mime Mail settings affect all emails sent from the site (including system emails sent through Rules, Webform emails, etc.), and not only those sent by Simplenews.

The options available for configuration are: F39.4

- **Sender name and e-mail address**. This is the default address which can be modified from the newsletter.

- **Include site style sheets**. If the theme does not include a specific style sheet for email (mail.css), all the styles from the site are included in the mail.
- **E-mail format**. Select Full HTML to avoid filters in the HTML content.

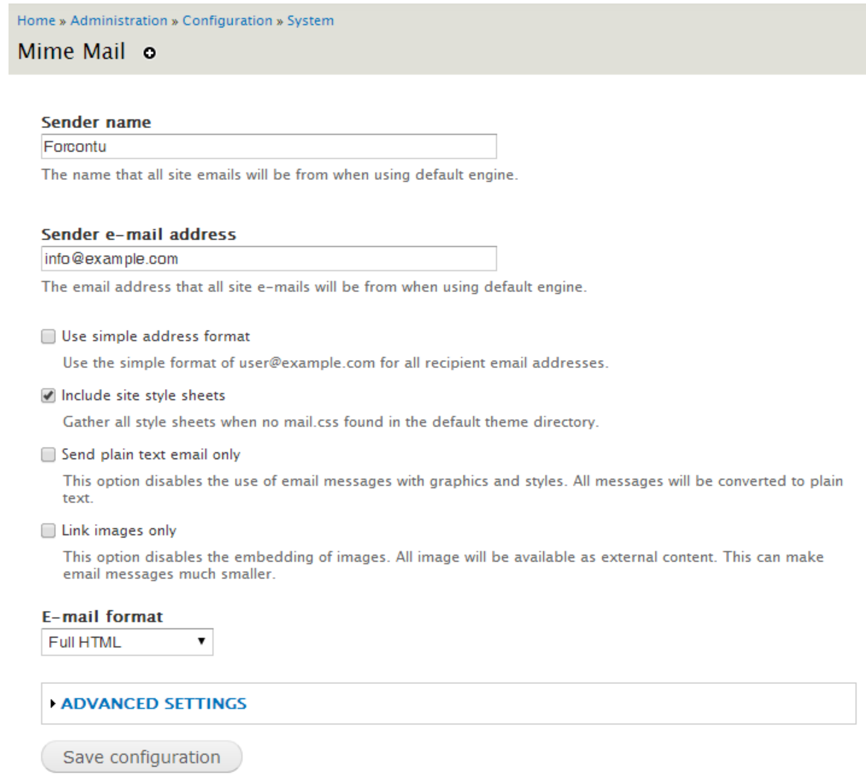
- **Theme to render the emails**. Utilize the theme active at the time the email is sent (Actual).

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F39.4

Mime Mail

The settings for Mime Mail affect all the emails on the site and not just the Simplenews newsletters.



Permissions

To complete the overall configuration of Simplenews we must consider the permissions added by the module:

Within the **Node** permissions we find permissions related to the content type **Simplenews newsletter**, which permits the creation of a new newsletter:

- *Simplenews newsletter*: Edit own content
- *Simplenews newsletter*: Edit any content
- *Simplenews newsletter*: Delete own content
- *Simplenews newsletter*: Delete any content

Within **Simplenews** we find these other permissions:

- Administer newsletters
- Administer simplenews subscriptions
- Administrar simplenews settings
- Send newsletter
- Subscribe to newsletters. This last permission should be given to users of the site that you wish to be able to subscribe to newsletters (for example, Anonmyous users or registered users)

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Subscribing to Newsletters

39.2

Create newsletter categories

In Simplenews we can define different **newsletter categories**. Users can select which categories they subscribe to so they will only recieve emails related to the topics they are interested in.

Newletter categories are created and configured from the Simplenews configuration page by accessing the Newsletter tab.

Once you have installed and activated the modules you can configure Newsletters from here:

Administration ⇨ Configuration ⇨ Web Services ⇨ Simplenews

URL Simplenews
/admin/config/services/simplenews

By default the module creates a primary category, assigning it the name of the site (*My Site* newsletter).

To create a **new category** click on **Add newsletter category** and configure the following parameters: F39.5

- **Name and description**. Both values will be displayed to users.
- **Subscription settings**:
    - **Subscribe new account**. Sets how the newsletter subscription option is displayed to new users on the registration form.
    - **Opt-in/out method**. Utilizes the double method for subscribing and unsubscribing by sending a confirmation email for each.
    - **Subscription block**. Creates a subscription block for this category. This block can be managed from the block management area of the site.
- **Email settings**. Format (Simple or HTML), priority and acknowledgement.
- **Sender information**. Specific values to be applied to all newsletters sent for this category.
- **Newsletter subject**. By default the category name (in brackets) and the title of the Newsletter node will be send in the email subject (for example, "**[News] Upcoming Courses**").
- **HTML to text conversion**. Options that apply when the newsletter is sent as plain text.

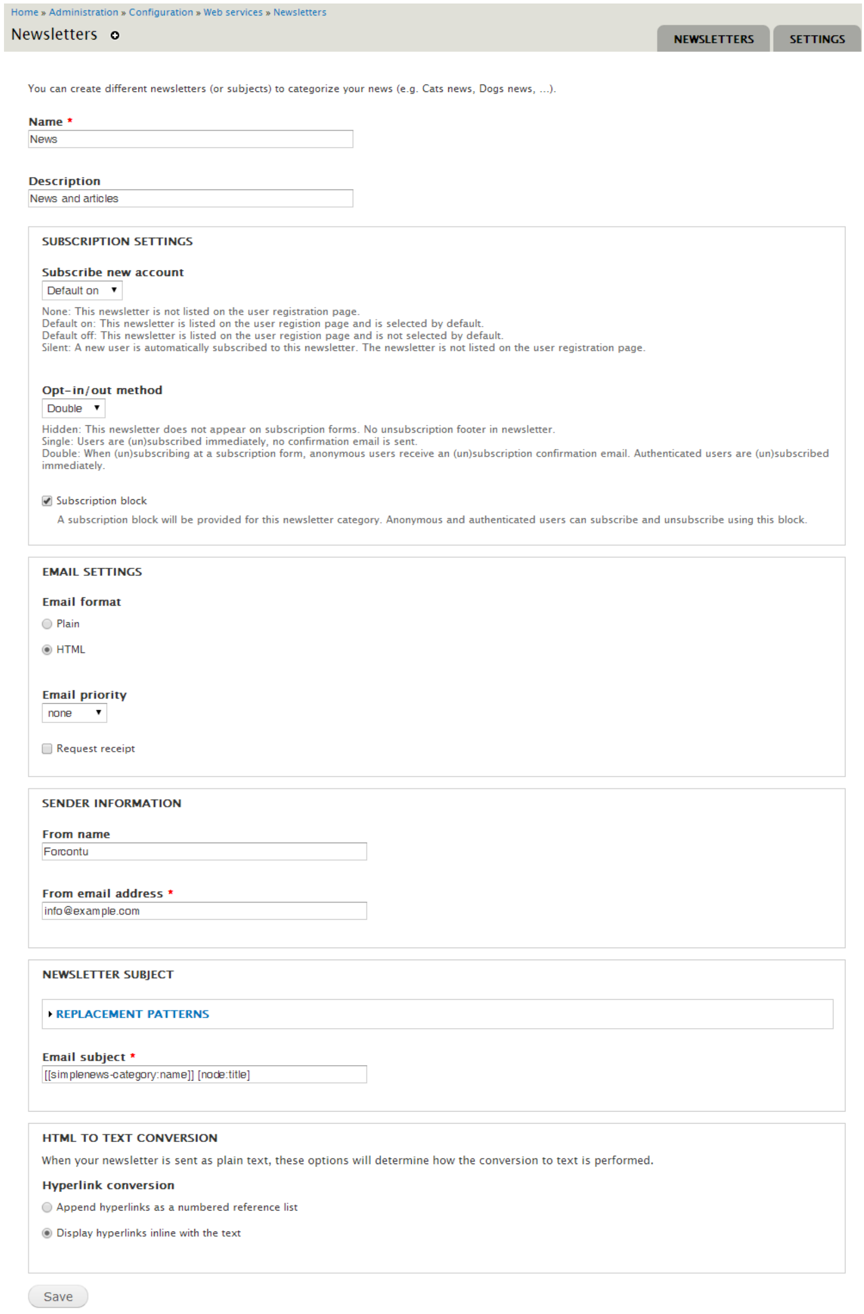
F39.5

Newsletter Category

Users can subscribe to newsletter categories.

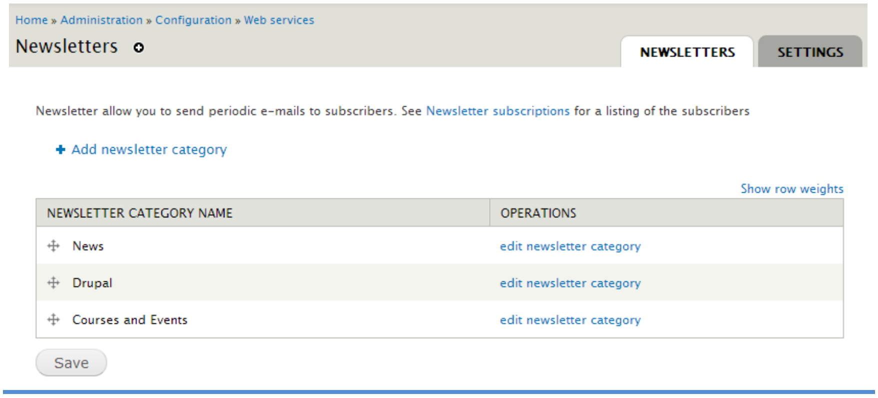
This allows newsletters by subject to reach only those subscribed users (and stakeholders).

For each Newsletter category we can establish an independent configuration.



As an example we can create the following categories of newsletters: F39.6

- News
- Drupal
- Courses and Events



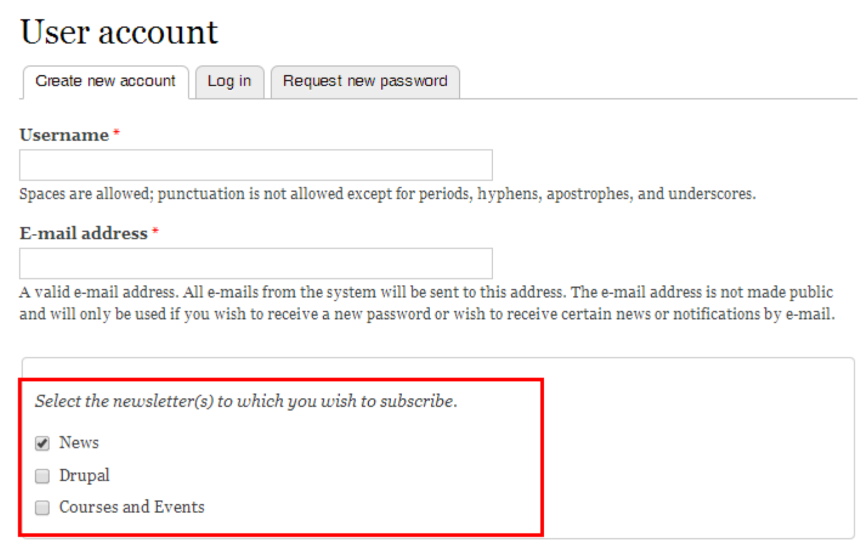
F39.6

Newsletter Categories

Shown is the adminstration list for the newsletter categories we created.

Subscription during registration

When configuring the category if under **Subscribe new account**, **Default on** or **Default off** was selected, the user will be able to subscribe to this category during registration of a new user account. The difference between each choice is whether or not the category subscription is selected by default or not. In any case, the user can change the selection before submitting the form. F39.7



F39.7

User Registration

You can include the option for new users to subscribe to Newsletter categories on the new user registration form.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

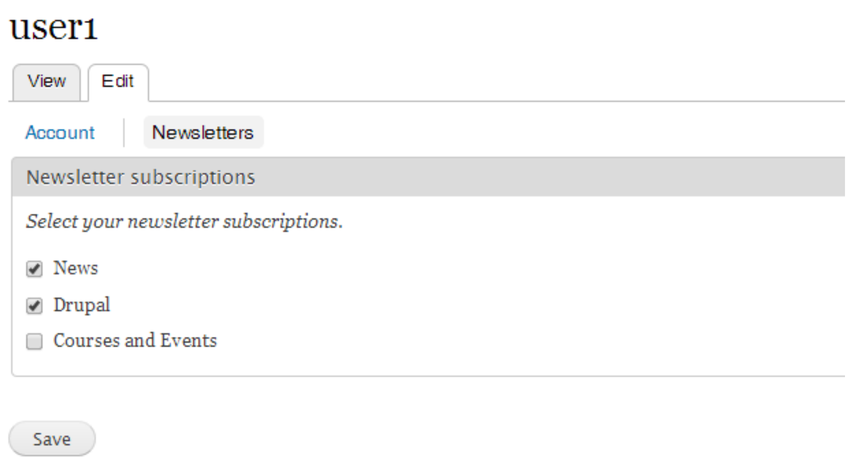
Management of user subscriptions

Registered users can modify newsletter subscriptions at any time by editing their profile under the Newsletter tab. F39.8

F39.8

User Newsletters

Newsletter subscriptions are avaialbe under the Newsletter tabs in the user profile. Users can manage subscriptions from there.



Subscription Block

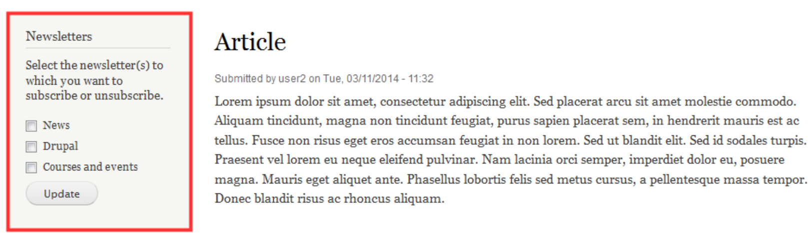
The module includes a block that allows users to subscribe to any of the Newsletter Categories (Newsletter: Multi Subscription). F39.9

If the user is already registered and logged into the site, they can just select the category and click Update.

F39.9

Newsletter Multi Subscription block (authenticated users)

The block shows all the newleter categories users can subscribe to or unsubscribe from.



For anonymous users the block asks for an email address and allows anonymous users to both subscribe to and unsubscribe from any category. F39.10

F39.10

Newsletter Multi Subscription block (anonymous users)

If you have set permissions for anonymous users to subscribe and unsuscribe the block will allow them to by requesting an email address.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

If you so indicated in the general Simplenews settings, the user will recieve a confirmation email to the address indicated (they will recieve an email for each category selected). The subscription or unsubscription will not be effective until the link on the confiormation email is clicked or accessed.

If in the configuration of the category you have activated the **Subscription block** option, a separate module for the category is created, identifed by the name **Newsletter: *Name of category***. This block can be used to by registered or anonymous users to subscribe or unsubscribe to the newsletter. F39.11

F39.11

Newsletter Block

Through the configuration of the category you can indicate if a seperate block should be created to manage subscription for that newsetter category.

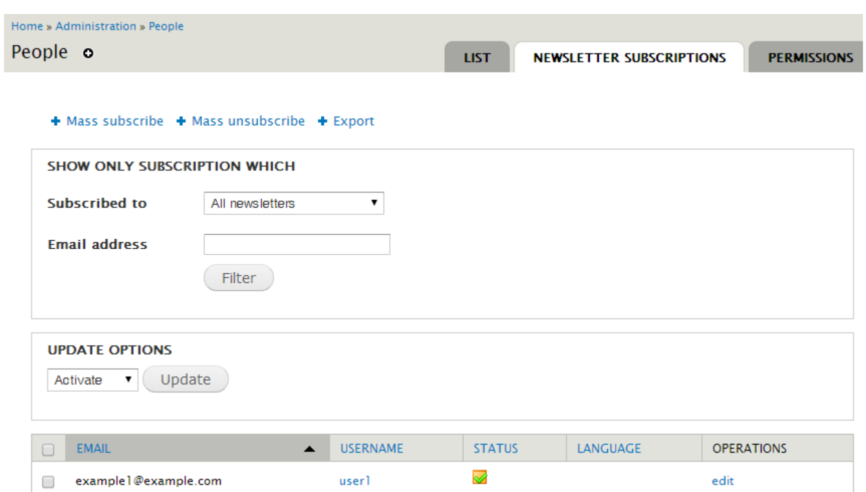
Subscription Administration

Users with the permission "Manage subscriptions Simplenews: can perform the following administration options from: F39.12

Administration ⇨ People [Newsletters subscriptions]

URL Subscription Adminstration
/admin/people/simplenews

- View and email users subscribed to the newsletter. This can be filtered by category or a specific email address search.
- Perform operations on the selected accounts. The operatiions availabe are: Enable, Disable, Delete.
- Edit each subscription. By accessing a particular subscription the administrator can modify user subscriptions.



F39.12

Subscription Administration

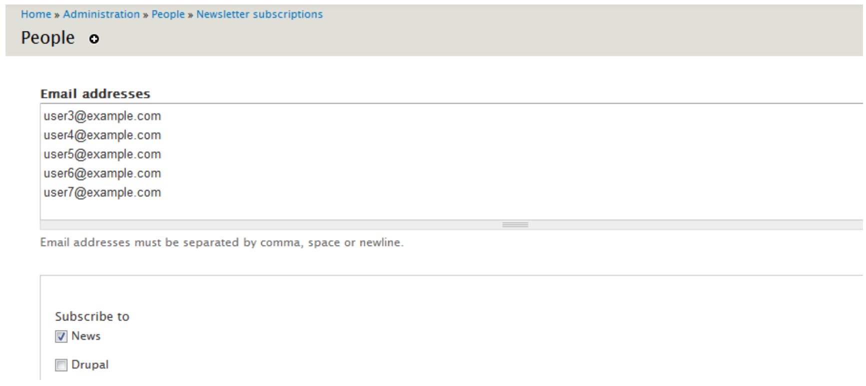
From the Subscriptions tab of the newsletter we can check and manage all the subscriptions

The administrator can also perform a **bulk subscribe or unsubscribe**, by including a list of email addresses to apply the action to. From this page you can also **Export** the list of subscribers for any category either for back up or to be used with an external application. F39.13

F39.13

Bulk Subscribe

The administrator can bulk subscribe or unsubscribe by inputing a list of email addresses .



Subscribe by Roles

Through additional moduels we can add new functionality to Simplenews. Although there are few modules available for Drupal 7, we can find some with a stable version.

The **Simplenews Private Roles** allows you to restrict certain categories of newletters based on certain roles.

The Simplenews Private Roles module is available at:

http://drupal.org/project/simplenews_private_roles

Once the module is activated, within each newsletter category is a new block **Restrict to roles**, where we can select the roles that can subscribe to that category. F39.14

F39.14

Simplenews Private Roles

The module allows you to restrict access to certain categories of newsletter based on user roles.

In categories where there is no role selected the subscription will be open to all roles. Once you select any of the roles, the subscription is available only for those roles.

Subscription through Rules

Through the **Simplenews Rules** module, included in Simplenews, we can integrate the system of Rules with Simplenews.

This module adds the following **events**, within the Simplenews group, allowing us to act during the subscribe/unsubscribe process:

- **A user has been subscribed**
- **A user has been unsubscribed**

There are not specific conditions for this module but we can use the generic condition for Rules. For example, if we want to check which categories have been subscribed to or unsubscribed from, we can use a comparison condition (Data comparison) and select a data selector **tid (Simplenews category)**, which will return the ID for the category.

Here are the specific **actions** for Simplenews that we find:

- **Send newsletter**
- **Subscribe an e-mail adress to a newsletter**
- **Unsubscribe an e-mail adress from a newsletter**

Integration with Rules adds great flexibility by allowing you to create specific rules to manage the subscribe/unsubscribe process.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

39.3 Create and Send Newsletters

Now that we have configured the module, added categories and subscribed users, the last step is to create and send newsletters. As mentioned before, users will only recieve newsletters for the categories they have subscribed to.

Create and send a newsletter

Simplenews adds a new content type called **Simplenews newsletter**, which lets you create and send newsletters. We can create newsletters from:

URL Create newsletter
/node/add/simplenews

Content ⇨ Add content ⇨ Simplenews newsletter

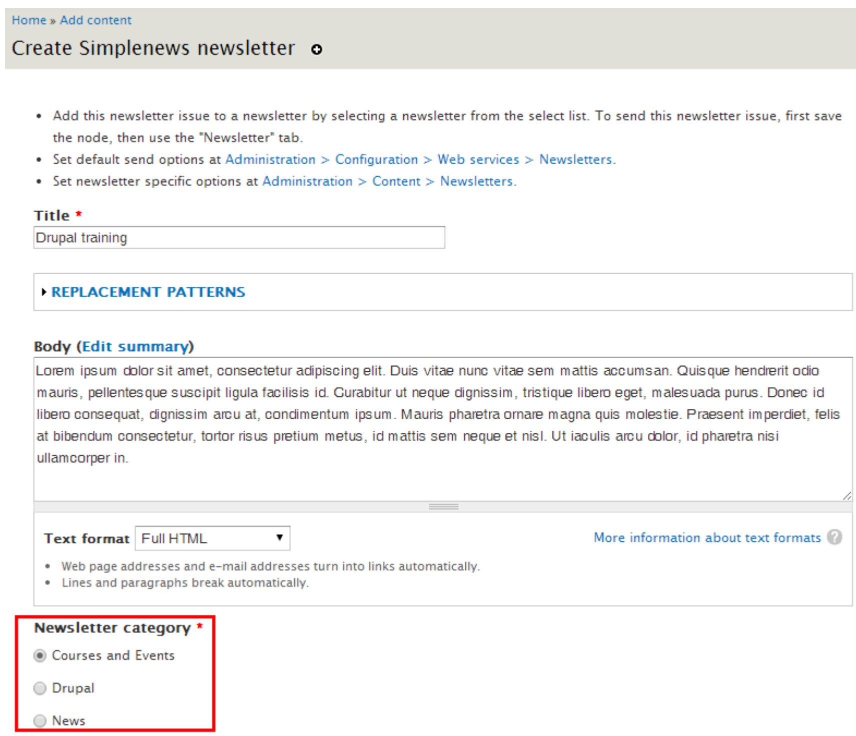
We will create a normal node in content; The only difference is that we have to select the **Newsletter category**. F39.15

F39.15

Create Newsletter

The content type Simplenews newsletter allows the creation of newsletter nodes which can be sent to subscribed users.

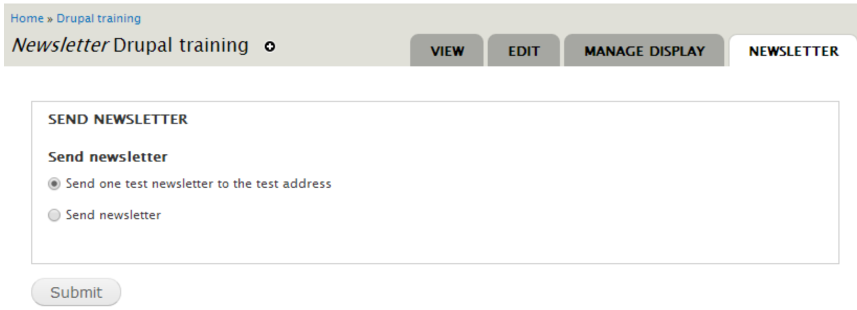
When you create a newsletter type node you must select the category of the newsletter.



The newsletter in the form of a node will also be accesible to the users of the site. This allows new members to access previous mailings or simply consult newsletters without having to subscribe to the newsletter.

From the **Newsletters** tab of the node we can send the newsletter. If we send the newsletter to a test account, this will immediately send a copy of the newsletter to the address assigned in the module configuration. F39.16

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



F39.16

Send the Newsletter

From the Newsletter node tab we can send the newsletter. You should always send a test newsletter before sending the newsletter to subscribers.

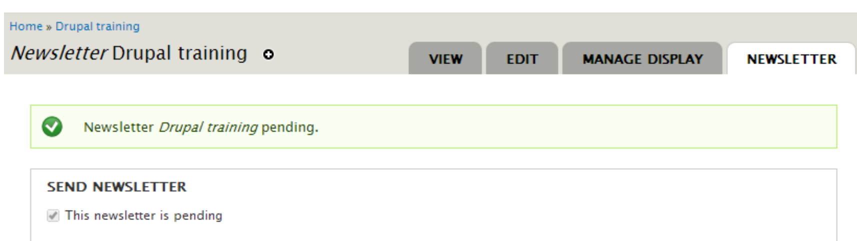
Once we have reviewed the content of the newsletter we can send it to the subscribed users. Select the option "Send Newsletter" and click Submit. Emails with the newsletter **will be sent in each run of the cron**, depending on the general settings on the module. F39.17

F39.17

Newsletter recieved

Example of a newsletter recieved by email.

Initially, the newsletter will be in a **pending** state. F39.18



F39.18

Newsletter Pending

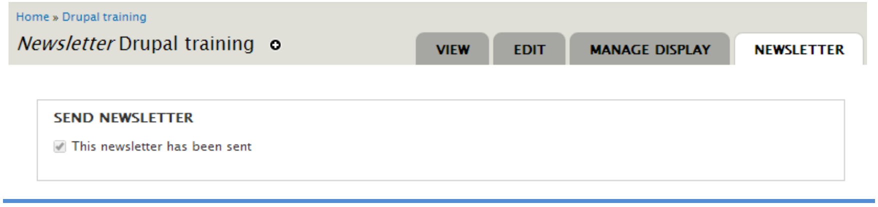
Once you have sent the newsletter it will remain in a Pending status.

Once cron has sent all of the newsletters the newsletter will change to a **Sent** status. F39.19

F39.19

Newsletter Sent

When cron has finished sending all the newsletter emails the status of the newsletter will change to Sent.



Once a newsletter has been sent, whether it is in Pending or Sent status, it cannot be sent again.

If you want to resend a newsletter, you have to create a new one (or clone the newsletter with Node Clone) and follow the steps again.

List of Newsletters

The list of newsletters that have been created can be managed from the following area: **F39.20**

URL Managing Newsletters
/admin/content/simplenews

Administration ⇨ Content [newsletter]

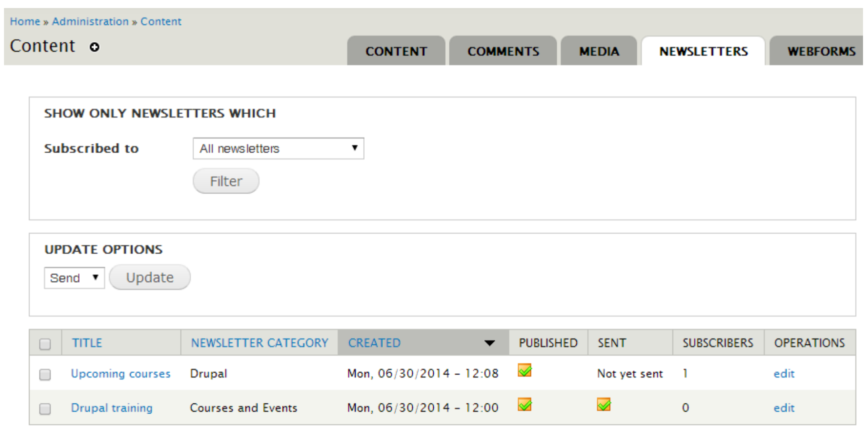
From this page you can edit newsletters and even send multiple newsletters at once (select the newsletters and change the **update options** to Send and click Update).

F39.20

Newsletter list

Administration area for newsletters. All newsletters created are shown here, including those not yet sent.

From this page you can send multiple newsletters at a time by selecting the newsletters and using the Send option.



Beside each newsletter the delivery status is displayed: Sent, Pending (0), not yet sent.

40 Other Features

In this last unit for the intermediate level, we will review some additional modules that allow us to complete the functionality of any website.

In this section we will review **Content Presentation** features of the following modules:

- **ThemeKey**, allows you to change the site theme based on different rules.
- **Printer, e-mail and PDF versions**, allows you to add options for printing, email and PDF creation.
- **Invisimail**, adds a filter to hide email addresses from spam bots.
- **Nice frontpage**, redirects the page /node to any other page on the site.
- **Site map**, adds a web site map.
- **Views Slideshow**, allows for the creation of slideshows.

Inside the **Mega Menus** module we will view the **OM Maximenu** module, which allows you to add blocks within the menu links.

We will continue with the **Calendar** module to create views of event calendars.

And we will end with two Administration modules:

- **Admin**, which adds an adminstration menu.
- **Elysia Cron**, which allows the execution of the tasks included in the cron, independently and at different times.

Comparative D7/D6

Many of the modules in this unit have a corresponding Drupal 6 version, but have been adapted to use the new features of Drupal 7.

Unit contents

40.1 Content Presentation 592
40.2 Mega Menus 606
40.3 Scheduling events with Calendar 610
40.4 Administration Modules 613

40

40.1 Content Presentation

In this section, we will introduce modules related to the presentation of content.

ThemeKey Modules

The **ThemeKey** module permits changing the theme based on defined rules. To construct these rules you can use the current path, taxonomy terms, language, content type, etc..

There are various applications that ThemeKey can provide:

- Have different themes by section.
- Different themes for special occasions.
- Themes adapted to certain browsers.
- Themes specific to the user's role.

ThemeKey is also a good solution to **test a new theme** while developing in a production environment because we can assign the theme to a particular user role without it being visibal to other site users.

The **ThemeKey** module is available at (install version 7.x-2.x):

http://drupal.org/project/themekey

Of the modules available install the following:

- **ThemeKey**, which is the base moduled and required for changing themes according to rules.
- **ThemeKey UI**, provides the management interface for the module.
- **ThemeKey User Profile**, allows users to select their own theme from those available. The functionality was part of Drupal 6 core, but was eliminatd in Drupal 7.

Once installed, access the configuration here:

URL ThemeKey
/admin/config/user-interface/themekey

Administration ⇨ Configuration ⇨ User interface ⇨ ThemeKey

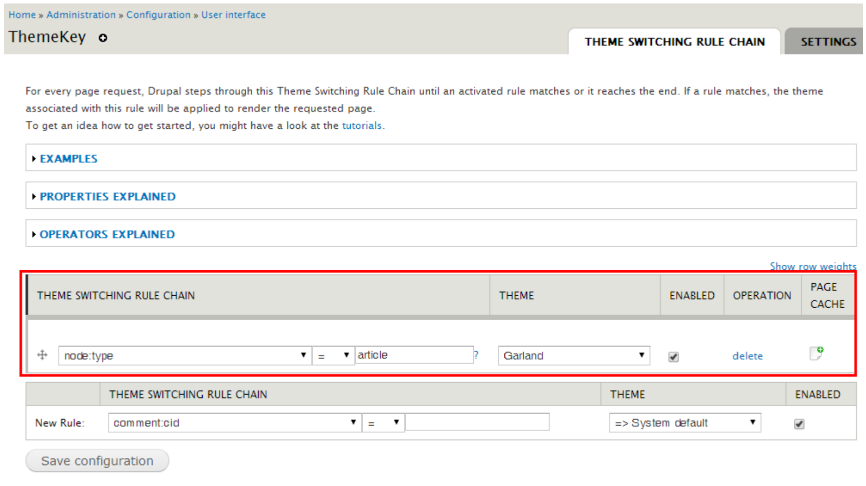
All the rules defined have the following structure: **properties (operators) value**.

The **properties** are explained in the "Properties explained" section. Some of the available properties are:

- **Drupal:path**. Path of the loaded page.
- **user:role**. Role of the current user.
- **system:date**. System date.
- **node:type**. Content type of the loaded page (system name).

The **operators** are explained in the "Operators explained" section. Some of the operators available are **equals** (=), **less than** (<), **greater** (>), **contains** (*), **not equals** (!), etc..

To create a new rule use the form on the page by selecting the property, operator, value and theme that will be activated if the rule is met.
For example, to make the **Garland** theme active when the content type (**node:type**) is equal (**=**) to **Article**, write the rule in the **Figure**. F40.1



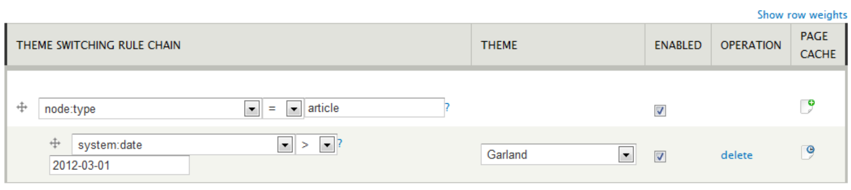
F40.1

ThemeKey

Creation of rules with Themekey. In each rule we have to select a property, an operator and a value in addition to the theme that will be activated if the condition is met.

Click **Save configuration** to save the rule and it will be added to the list of rules for changing the theme.

Rules **can be nested or grouped**, so that the theme change will only occur if all the rules are met (**AND**). To nest a rule we have to drag it using the **list order** icon, as we have done with other elements such as menu links. **F40.2**



F40.2

ThemeKey
Nesting of rules

Rules can be nested, so that the theme will be applied only if all the grouped conditions are met.

Rules should be ordered property because the system will check the rules in the order until it finds a valid rule. The theme change is made according to the first rule found, ignoring the remainder of the rules.

We recommend consulting the **Examples** section and reviewing the sample rules shown.

From the **Settings** tab, we can configure the other options for ThemeKey. For example, by enabling the **Add theme option to user profile** (in the subtab **User interface**), the user can select an item from the **Selectable themes**, which we can also configure from this page.

The user must **F40.3** also have permission to "Select a different theme" (**ThemeKey User Profile** Module). The user can then select the theme by editing their profile.

As mentioned this functionality was available as part of Drupal 6 core, but was eliminated in Drupal 7. With ThemeKey we can recover that functionality for Drupal 7 sites.

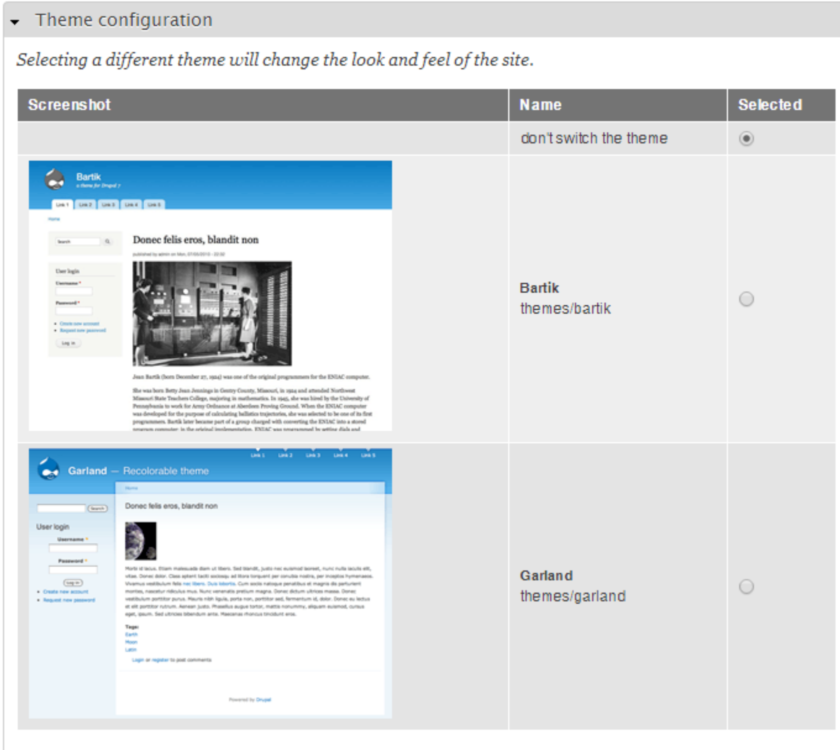
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F40.3

ThemeKey User Theme

Using the ThemeKey User Profile we can configure the system so that users can select a theme from the "selectable" themes available.

This functionality is very similar to Drupal 6 core, which was elimniated in Drupal 7.



Printer, e-mail and PDF versions module

The **Printer, e-mail and PDF versions** module allows you to add options for printing, emailing or creating PDF versions of site content. These options add the following icons in content where they are enabled.

The module is available at:

http://drupal.org/project/print

It includes the following modules:

- **Printer-friendly pages**, this is the main module and it is responsible for changing the page to the print version.
- **PDF version**, allows page export to PDF.
- **Send by e-mail**, allows pages to be emailed.

URL Printer, e-mail and PDF versions
/admin/config/user-interface/print

Once installed you can configure the module here:

Administration ⇨ Configuration ⇨ User interface ⇨ Printer, e-mail and PDF versions

This allows for individual configuration of all the options through different tabs in the administration page:

- **Web page**, for print view options.
- **Email**, for emailing options.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **PDF**, for exporting a PDF.

- **Settings**, general options, but mainly affecting the print version (Web page).

From **Web Page options** we can configure:

- **Where we show the link for** the print version (Links area and/or Content corner).

- **Link style** (text, icon or both), CSS class and other visibility options.

- If the print version will be opened in a **new window**.

- We can also set how **search engines** will behave when encountering these pages. Generally, and it is the default, we will indicate that they should not index (noindex) nor follow any links on the page (nofollow).

To use the **PDF export option** you must install an external library that is not included in the module. We have three tools available:

- **Wkhtmltopdf**, requires complex installation on the server but is also the best choice. It is available at: http://code.google.com/p/wkhtmltopdf/

- **TCPDF** is a very hip solution, available at www.tcpdf.org.

- **dompdf**, supports CSS 2.1 and some properties of CSS 3, giving very good conversion results. The latest version of dompdf is available at: http://code.google.com/p/dompdf/

Installation instructions for each of these tools are available in the **INSTALL.txt** file included with the module. Once at least one tool has been installed, you must select the one you want to use from the PDF tab.

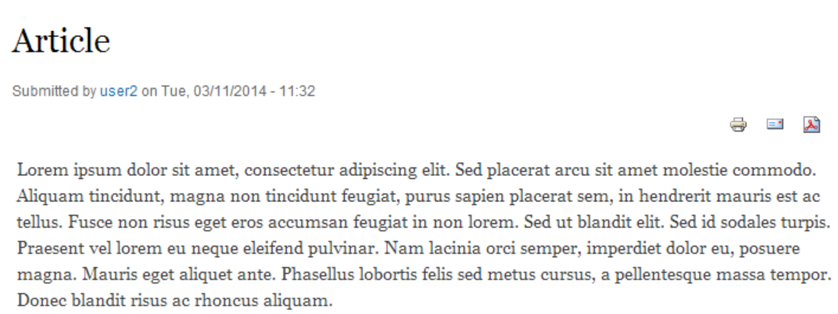
F40.4

F40.4

Printer, e-mail and PDF PDF

To use the export to PDF function, install and select one of the PDF conversión tools.

Figure F40.5 shows an example of content with the icon links displayed.



F40.5

Printer, e-mail and PDF Icons

Example content is shown with icons to print, email or export to a PDF.

Figure F40.6 shows an example of the print version of a page. As you can see it is a simple HTML page, without a frame or other site designs.

F40.6

Printer, e-mail and PDF
Print Version

The print version is a clean version of the page, in HTML format, minus the frame or site design.

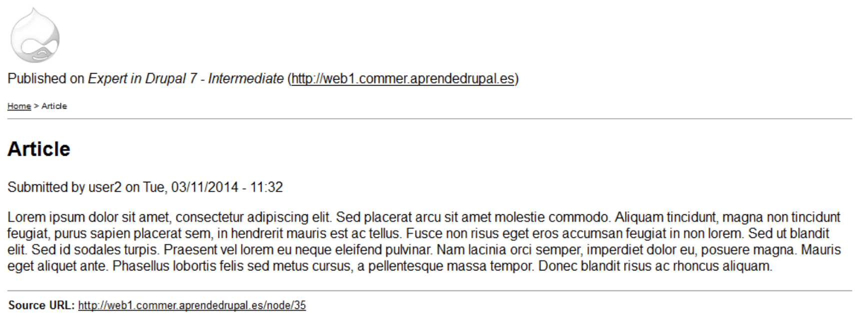
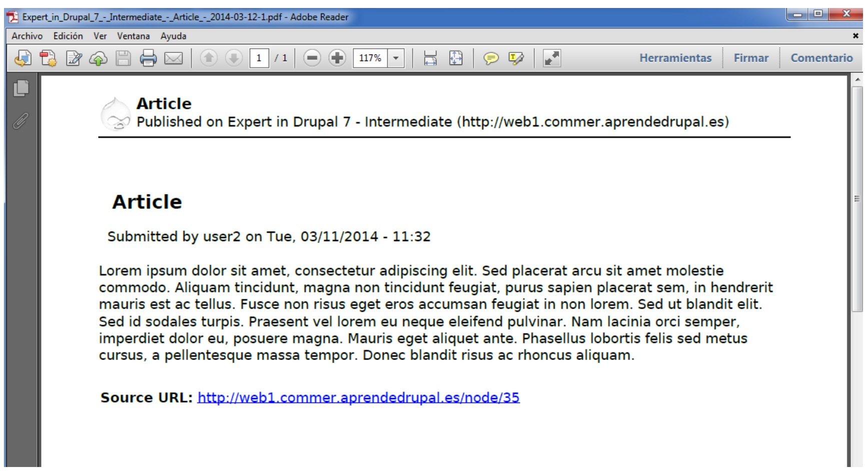


Figure F40.7 shows the results of exporting to a PDF.

F40.7

Printer, e-mail and PDF
PDF

The results of exporting to PDF. The look may vary depending on the HTML content of the page and the PDF conversión tool that is being used.

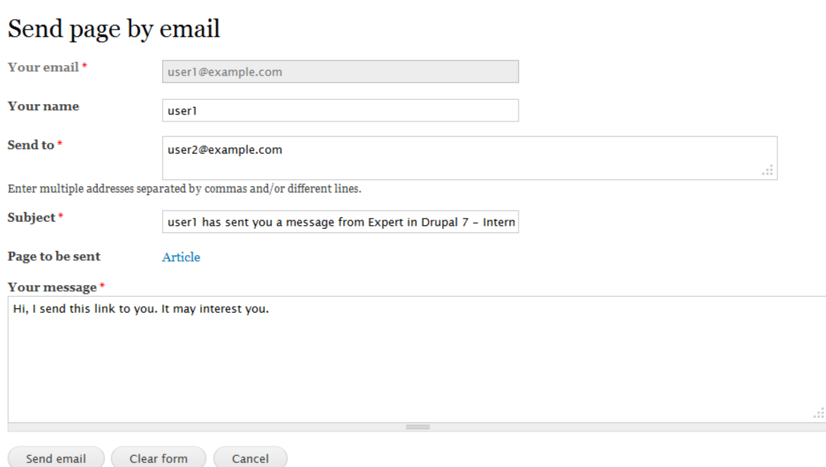


Finally **Figure** F40.8 shows the form the user completes to send content in an email to another person (or their own email account).

F40.8

Printer, e-mail and PDF
Email to a friend

The form used to send content via email.



In order for users to see the links and be able to use these tools they should be

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

assigned the following permissions:

- **Access the PDF version**, included in the **PDF version** module.
- **Access the printer-friendly page**, included in the **Printer-friendly pages** module.
- **Access the Send To Friend functionality**, included in the **Send by e-mail** module.

Finally, we can indicate **default options for each content,** through the options group **printer-friendly version, send by email and PDF version**.

Later, when you create a node, you can use the default or set **specific options for the node**.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Invisimail Module

The **Invisimail** module adds a content filter to hide email address from spam bots.

SPAM bots navigate through the pages looking for email addresses, which are then added to a database to send SPAM email to those addresses.

Email addresses are encoded in HTML so SPAM bots cannot directly detect them but the user can view them normally.

The **Invisimail** module is available at:

http://drupal.org/project/invisimail

Once installed, the module adds two filters which are available in the **text format** editor: F40.9

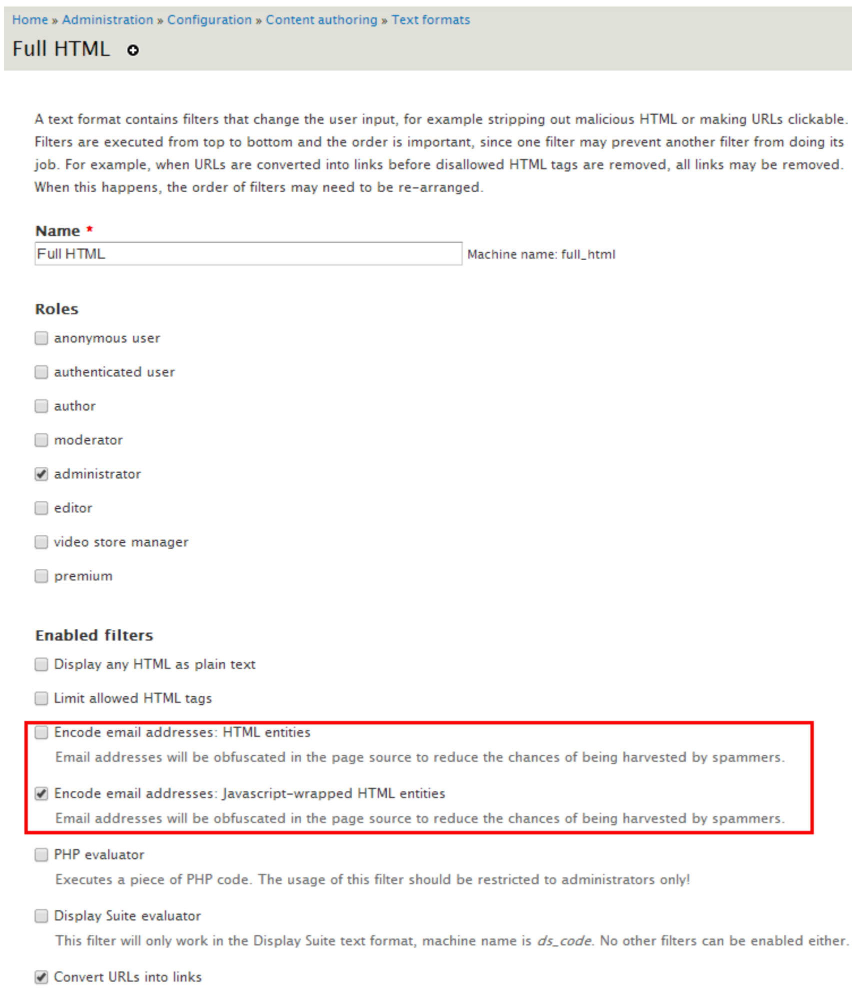
URL Text Formats
/admin/config/content/formats

F40.9

Invisimail Filters

Two new filters are added in the text format editor. These filters perform the same function with different methods so it is only necessary to activate one of them.

Administration ⇨ Configuration ⇨ Content authoring ⇨ Text formats

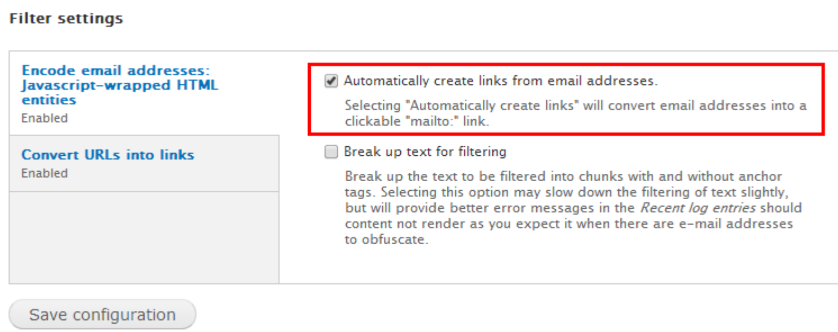


Both filters perform the same action, but use different procedures, so you only need to activate one of the two.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

It is important that the email encoding filter is placed **above the other filters**, especially above the **Convert URLs into links** filter.

Depending on the active filter, a group is added to the filter options, where we can activate the option to **Automatically create links from email addresses**. F40.10

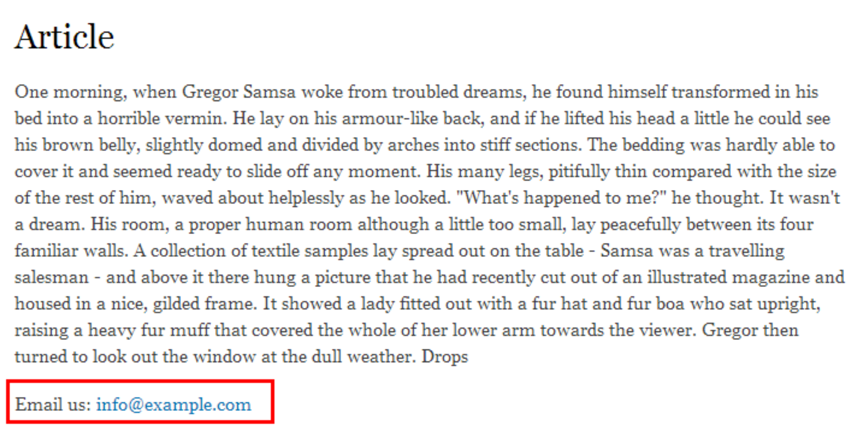


F40.10

Invisimail Filter Options

We must activate the option to automatically create links from email addresses. We also need to ensure that the filter is executed first (from the list of filters).

Figure F40.11 shows content with a link to an email address. The user will see the link to any email (mailto: info@example.com), and may even use the link to send an email directly.



Article

One morning, when Gregor Samsa woke from troubled dreams, he found himself transformed in his bed into a horrible vermin. He lay on his armour-like back, and if he lifted his head a little he could see his brown belly, slightly domed and divided by arches into stiff sections. The bedding was hardly able to cover it and seemed ready to slide off any moment. His many legs, pitifully thin compared with the size of the rest of him, waved about helplessly as he looked. "What's happened to me?" he thought. It wasn't a dream. His room, a proper human room although a little too small, lay peacefully between its four familiar walls. A collection of textile samples lay spread out on the table - Samsa was a travelling salesman - and above it there hung a picture that he had recently cut out of an illustrated magazine and housed in a nice, gilded frame. It showed a lady fitted out with a fur hat and fur boa who sat upright, raising a heavy fur muff that covered the whole of her lower arm towards the viewer. Gregor then turned to look out the window at the dull weather. Drops

Email us: info@example.com

F40.11

Invisimail Content

Sample content with a mailing address. The user will see the email address as normal.

However, if we analyze the generated HTML, you will see that the link is displayed as <a href="mailto: info@example.com"...>, but in a codified version. F40.12

```
<p>Email us         <span id="7fcabfe4911afec8b5b8fd80ef54fc7b1cfad1d3"></span></p>
<script type="text/javascript">
<!--//--><![CDATA[// ><!--
 <!--
  document.getElementById('7fcabfe4911afec8b5b8fd80ef54fc7b1cfad1d3').innerHTML = '<a
href="&#109;&#97;&#105;&#108;&#116;&#111;&#58;&#105;&#110;&#102;&#111;&#64;&#101;&#120;&#97;
&#109;&#112;&#108;&#101;&#46;&#99;&#111;&#109;">&#105;&#110;&#102;&#111;&#64;&#101;&#120;&#9
7;&#109;&#112;&#108;&#101;&#46;&#99;&#111;&#109;';
// -->
//--><!]]>
</script>
```

F40.12

Invisimail HTML Code

Analyzing the HTML code we see that the email address is actually encoded.

Nice frontpage Module

It is very common when developing a website to develop an alternative home page to the default page (**/node**). From that moment we forget about /node, although it is still accessable through the URL.

The **Nice frontpage** module provides a very basic but very helpful functionality: redirecting the page **/node** to any other page of the site, generally to the **home**

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

page that we have defined. In this way, we avoid accessing the page /node, which we normally don't access but also don't remove.

The functionality provided by **Nice frontpage** is available with other modules (eg with Rules).

The Nice frontpage module is available at:

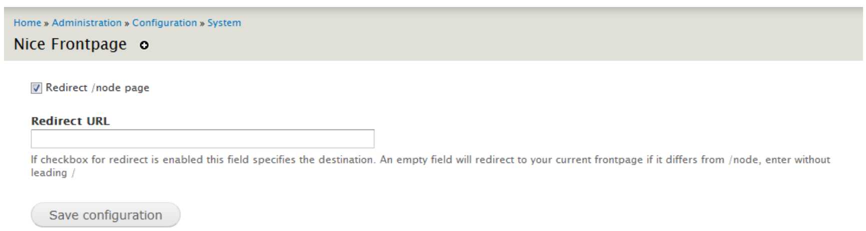
http://drupal.org/project/bs_nice_frontpage

There is no need for any specific configuration; the module will redirect the URL **/node** to the configured **home page**. If we want to configure another URL or disable URL redirection, we can do it from the module configuration in:

URL Nice Frontpage
/admin/config/system/bs_nice_frontpage

F40.13

Administration ⇨ Configuration ⇨ System ⇨ Nice Frontpage



F40.13

Nice Frontpage

By default, the module redirects the page /node to the designated home page. From the configuration area we can specify any other URL.

Site map module

A site map is simply a list of links to all the pages of a site, ordered according to the menu structure defined in the site.

The site map allows visitors to see the overall structure of the site and quickly access various content. It is also very useful to help search engine bots find and index all the content on our website.

We studied in **Unit 37** the **XML Sitemap** module, which generates a site map in XML, expecially formatted for search engine bots to understand. Now we will review the **Site map** module, which we will use to create a site map for site users.

The **Site map** module is found at:

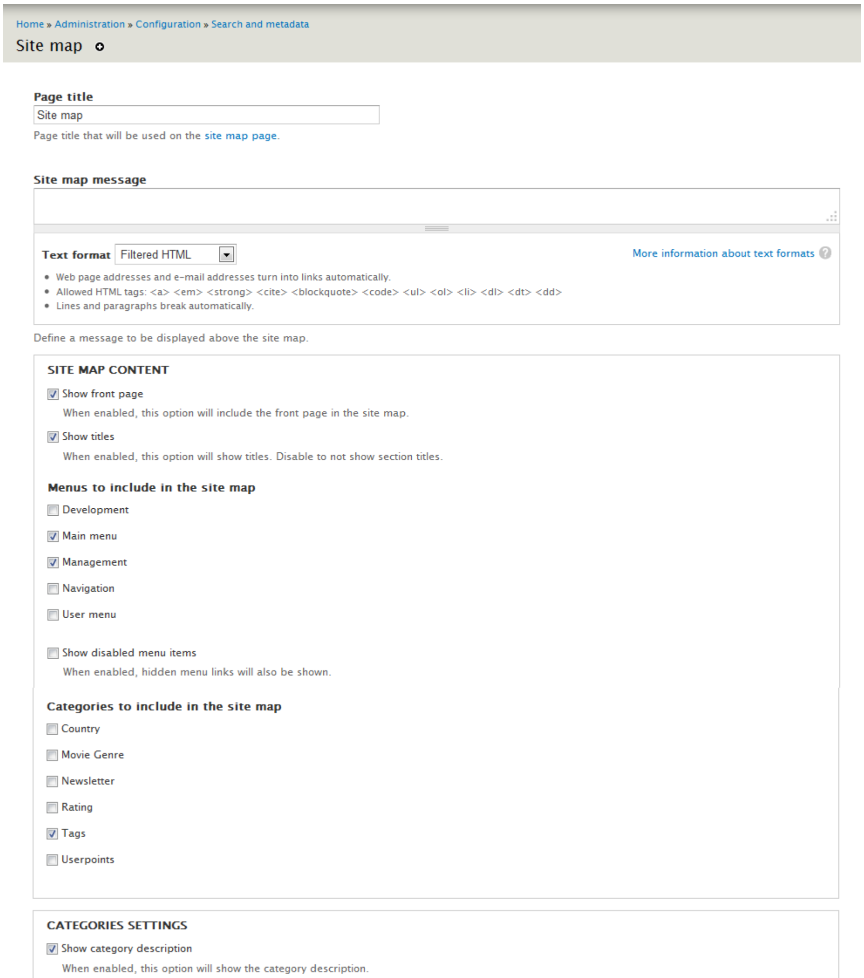
http://drupal.org/project/site_map

The **Site map** module allows you to create site maps from created content structures in the site. In particular, it allows the addition of menus and taxonomy categories, as well as to other elements like books and blogs to the site map. The configuration options for site map are found in:

Administration ⇨ Configuration ⇨ Search and metadata ⇨ Site map

URL Site Map
/admin/config/search/sitemap

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



F40.14

Site map

Configuring the site map. We can select which menus and which categories are displayed on the map.

Some of the options you can configure are: F40.14

- **Page title and site map message**. Sets the title to be displayed on the site map page and the text to be displayed before the links.
- **Show front page**. The link to the home page will be included in the menu. This can be useful if the homepage is not already linked from any of the menus of the site.
- **Menus to include in the site map**. Selects which menus to display. It is not recommended, for example, to display the navigation menus as they are essentially administration options.
- **Categories to include in the site map**. Selects which vocabularies will be added to the site map. For the categories you can configure other parameters like the **number of nodes shown by category**, the depth etc..

Once configured, the sitemap will be available at the URL **/sitemap**. **Figure** F40.15 shows an example of a site map.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

F40.15

Site map

Site map example.

Site map

Front page

Front page of *Forcontu*

Main menu

- Home

Country

- United States (15)

Movie Genre

- Action (2)
- Comedy (1)
- Drama (3)
- Horror (2)
- Sci-Fi (2)
- Thriller (1)

Rating

Movie rating

- General Audiences. All Ages Admitted. (1)
- Parental Guidance Suggested. Some Material May Not Be Suitable For Children. (2)
- Parents Strongly Cautioned. Some Material May Be Inappropriate For Children Under 13. (2)
- Restricted. Children Under 17 Require Accompanying Parent or Adult Guardian. (1)

Tags

Use tags to group articles on similar topics into categories.

- Drupal 7 (3)
- eLearning (1)
- News (2)

Views Slideshow module

The **Views Slideshow** module allows you to create blocks of content that rotate like a slide show. Although this functionality is generally applied to image rotation, the module allows any content that can be shown through a view to be used.

The **Views Slideshow** module is available at:

http://drupal.org/project/views_slideshow

The module requires the installation of these other modules:

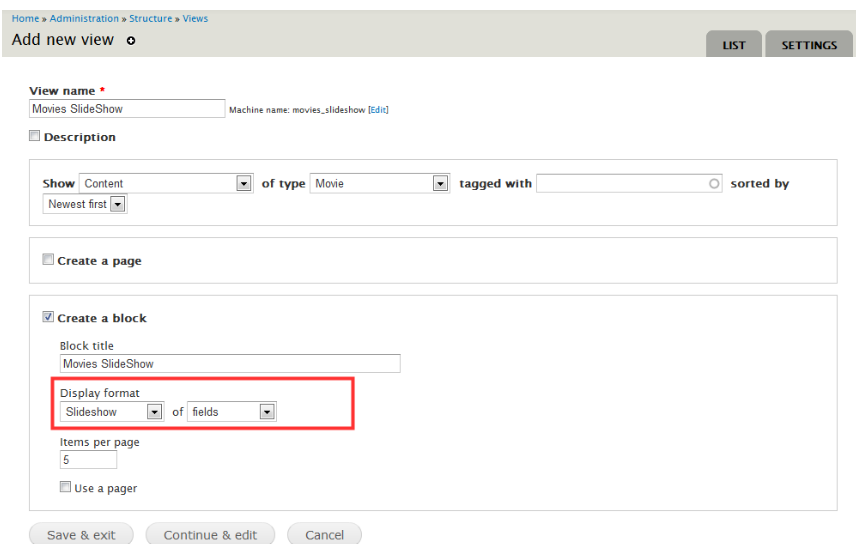
- **Views 3**, available at: http://drupal.org/project/views
- **Chaos tool suite (ctools)**, available at: http://drupal.org/project/ctools
- **Libraries API**, available at: http://drupal.org/project/libraries

We also need to install the **Jquery Cycle** plugin, which we can download from: http://jquery.malsup.com/cycle/download.html

To install the plugin create the folder **/sites/all/libraries/jquery.cycle**, and add under it the downloaded file, **jquery.cycl.all.js**.

Finally activate the modules, **Views slideshow** and **Views slideshow: Cycle**.

Once installed, when **creating a view** we will have **a presentation format** available called **Slideshow**. F40.16



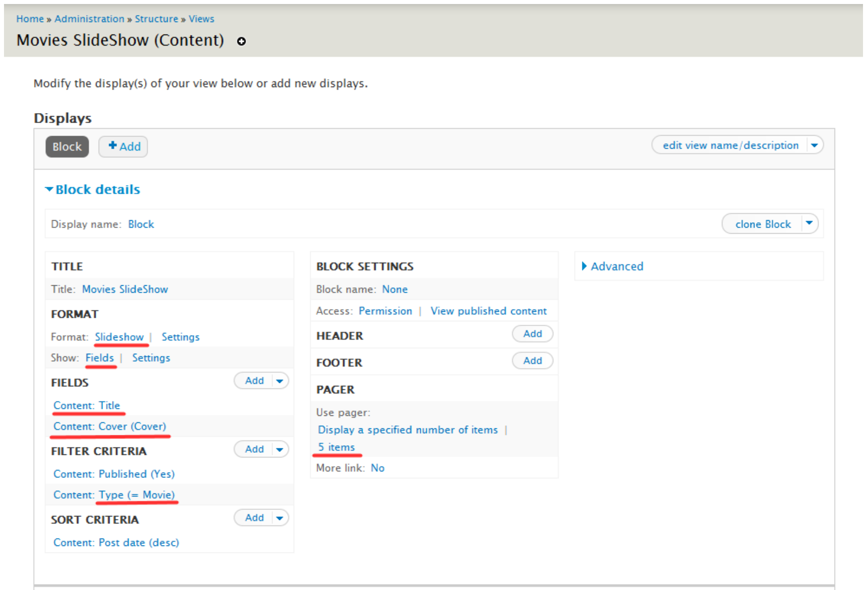
F40.16
Views Slideshow
Slide presentation
When creating the view, select the display format called Slideshow.

As an example we have created a block type view that displays the latest 6 movies added to the site. It shows the **Slideshow** format, showing the **Title** and **Cover**, linking both to the node. Select an appropriate **picture style** for the cover in the F40.17 sidebar of the site.

F40.17

Views Slideshow
Configuring the view

For this example we have created a view with a Slideshow presentation format, displaying the Title and Cover of nodes of the type Movie. Only five ítems are displayed.



Finally activate the block on one side of the site to test it. **F40.18**

F40.18

Views Slideshow
Block

This shows a block where content will be rotated. Clicking on the title or cover will Access the corresponding node.

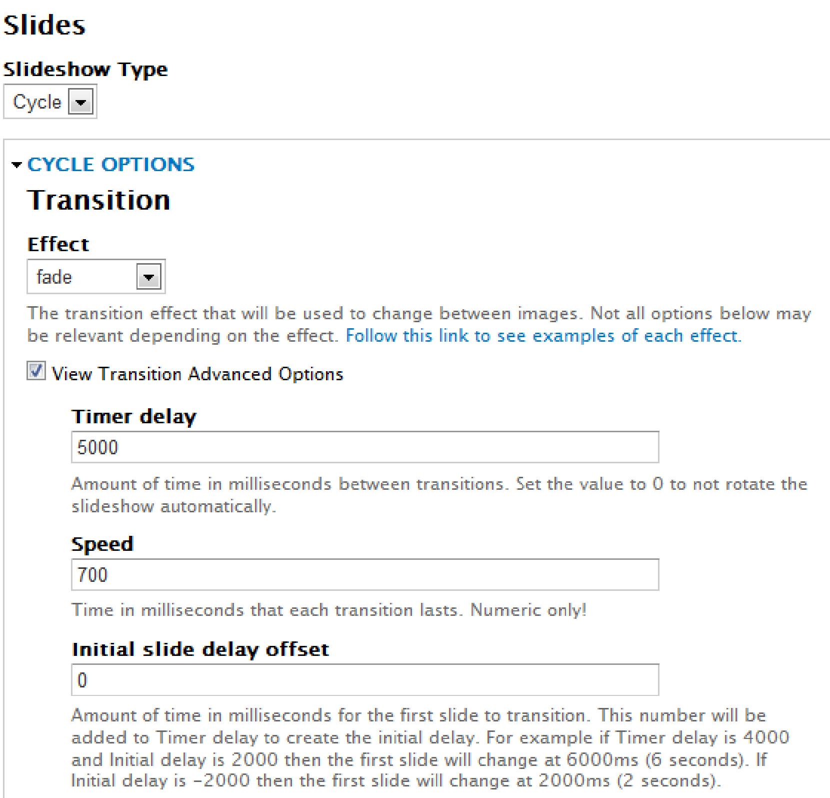
We have activated a slide counter.

Returning to the view editor, from the **Settings** for **Slideshow format** you can configure some other parameters related to slide transition: **F40.19**

- **Effect**. The module includes 27 configurable transition effects. By enabling the **advanced options** you can configure options just as **transition speed** and **time between transitions**.

- **Widgets**. We can add additional controls, such as a **pager** or a **Slide counter,** in the header and/or footer of each slide.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



F40.19

Views Slideshow Presentation Options

Among the options for the slideshow format, we can set some transition related parameters.

We can also add additional controls such as a pager or slide counter, which displays the current slide number and the total slide number.

The module is very flexible because it uses views to generate the presentation of the slideshow. Further more, **Views Slideshow** can be supplemented with additional modules:

- **Views Slideshow Xtra**, which allows you to overlay HTML elements (such as text, links, icons, etc.) on each slide.

  The module is available at:

  http://drupal.org/project/views_slideshow_xtra

- **Views Slideshow: Galleria**, to create image galleries

  The module is available at:

  http://drupal.org/project/views_slideshow_galleria

40.2 Mega menus

A **mega menu** is a menu that allows you to display a lot of information in an easy to read way. While classic drop down menus display a list of links, mega menus can contain more complex structures (multiple columns, images, texts, etc.).

To help you better understand what a mega menu is, the following figures show some good examples. F40.20 F40.21 F40.22

F40.20

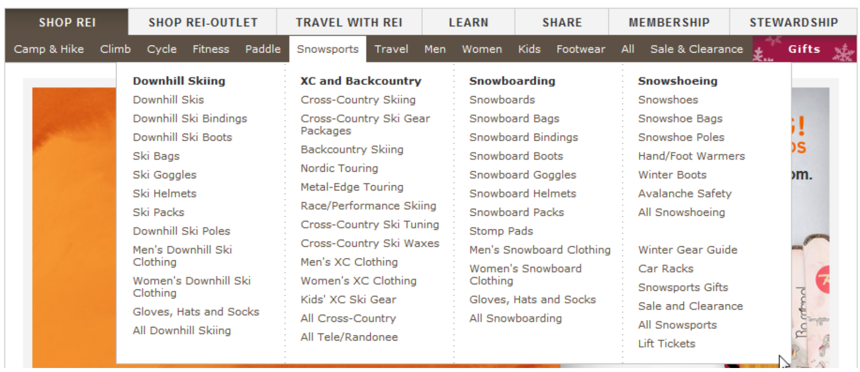
Mega menu Example

Mega menu with images at sony.es.

F40.21

Mega menu Example

Mega menu with columns at www.rei.com



F40.22

Mega menu Example

Mega menu with block and columns at www.whitehouse.gov

There are different modules for Drupal to create **Mega menus** on a site. Some of them are:

- **Megamenu**, available at: http://drupal.org/project/megamenu
- **Giga Menu**, available at: http://drupal.org/project/gigamenu
- **OM Maximenu**, available at: http://drupal.org/project/om_maximenu
- **Menu Views**, available at: http://drupal.org/project/menu_views
- **Menu Minipanels**, available at: http://drupal.org/project/menu_minipanels

OM Maximenu Module

Among those modules, we have selected the **OM Maximenu** module as an example, which we will discuss in this section.

OM Maximenu allows you to add blocks to menus.

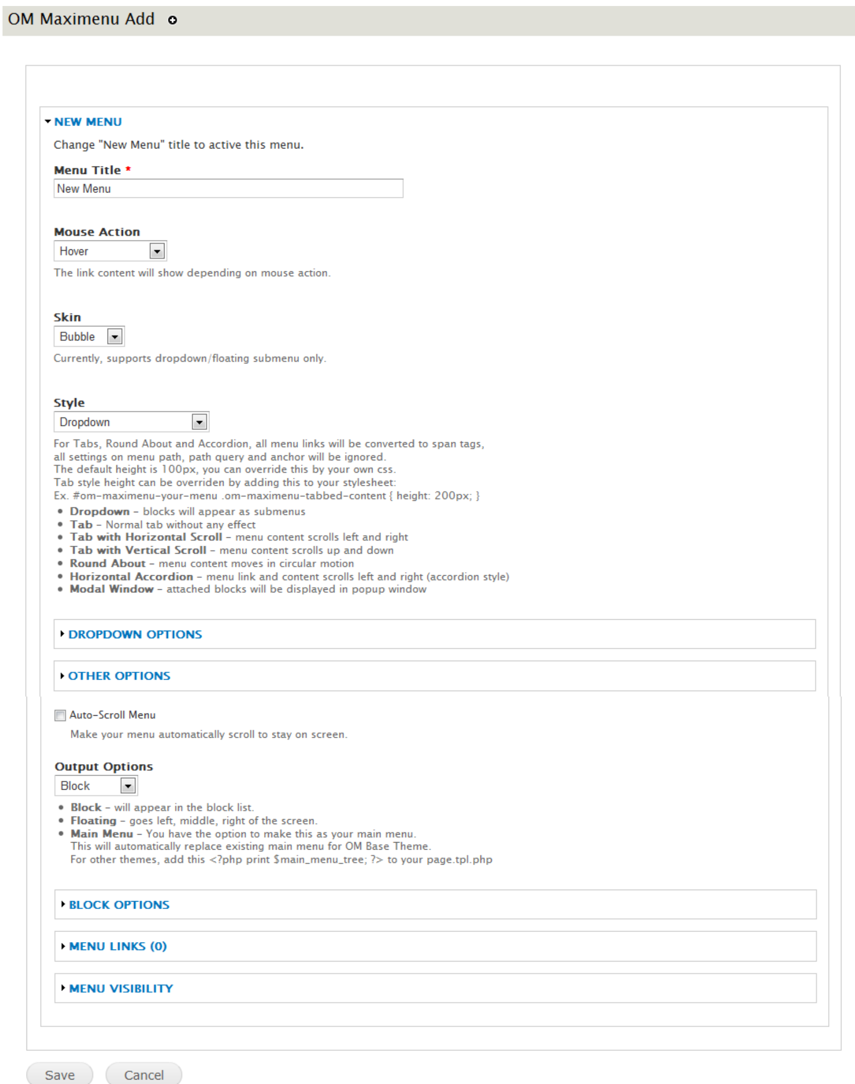
The **OM Maximenu** module is available at:

http://drupal.org/project/om_maximenu

Once installed, we can access the configuration from:

Administration ⇨ Structure ⇨ OM Maximenu

URL OM Maximenu
/admin/structure/
om-maximenu


OM Maximenu Add

▾ NEW MENU
Change "New Menu" title to active this menu.
Menu Title *
New Menu

Mouse Action
Hover
The link content will show depending on mouse action.

Skin
Bubble
Currently, supports dropdown/floating submenu only.

Style
Dropdown
For Tabs, Round About and Accordion, all menu links will be converted to span tags,
all settings on menu path, path query and anchor will be ignored.
The default height is 100px, you can override this by your own css.
Tab style height can be overriden by adding this to your stylesheet:
Ex. #om-maximenu-your-menu .om-maximenu-tabbed-content { height: 200px; }

- **Dropdown** – blocks will appear as submenus
- **Tab** – Normal tab without any effect
- **Tab with Horizontal Scroll** – menu content scrolls left and right
- **Tab with Vertical Scroll** – menu content scrolls up and down
- **Round About** – menu content moves in circular motion
- **Horizontal Accordion** – menu link and content scrolls left and right (accordion style)
- **Modal Window** – attached blocks will be displayed in popup window

▸ DROPDOWN OPTIONS

▸ OTHER OPTIONS

☐ Auto-Scroll Menu
Make your menu automatically scroll to stay on screen.

Output Options
Block

- **Block** – will appear in the block list.
- **Floating** – goes left, middle, right of the screen.
- **Main Menu** – You have the option to make this as your main menu.
This will automatically replace existing main menu for OM Base Theme.
For other themes, add this <?php print $main_menu_tree; ?> to your page.tpl.php

▸ BLOCK OPTIONS

▸ MENU LINKS (0)

▸ MENU VISIBILITY

Save Cancel

F40.23

OM Maximenu
Create Maximenu
Options for configuring an new Maximenu.

First we must create a Maxi Menu from **Add menu**, and give it the general configuration: **F40.23**

- **Menu title**. Title for administration listings.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Mouse Action**. Indicates what mouse action displays the menu. We can select **hover** (hover over the option) or **click**.

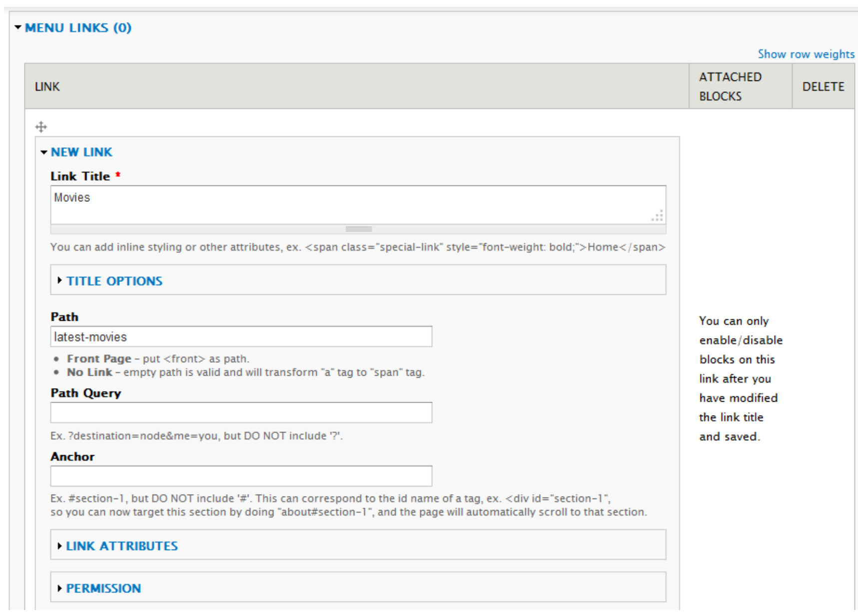
- **Skin**. Available styles for the window that is displayed when the menu is selected.

- **Style**. Some available styles. For example, the dropdown style deploys elements downward.

- **Output options**. By selecting Block, the menu created will be available as a block, which gives us the flexibility to display the menu in any region of the site.

From **Menu links** we can add menu links. First, we have to create the link, at minimum giving it a title and once we have saved it, we assign the block or blocks that belong to the link. F40.24

F40.24
OM Maximenu
Create link
First create the link and once it is saved the link can be added to the block.

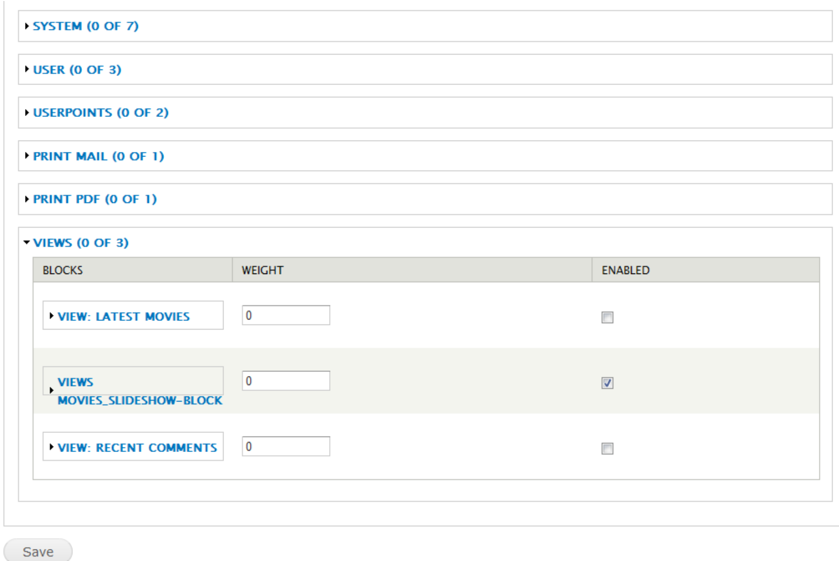


To configure the blocks for each link we have to click on the **Edit attached blocks** link. F40.25

F40.25
OM Maximenu
Adding Blocks
A link is able to contain multiple blocks.

A list of the available blocks, grouped by their origin (module) are displayed together (Block, Menu, System, User, Views, etc.).

A menu link can contain multiple blocks. Activate all blocks that contain the link and then order them according to weight. For each block added we can set the title, path, etc.. If you don't not want a title name displayed, write <none> in the title field. F40.26



F40.26

OM Maximenu Available Blocks

In the list of available blocks you will find the site blocks grouped by the modules that create them. (Block, Menu, Views, etc.).

A link can contain multiple blocks We can actívate them and sort them by weight.

Figure F40.27 shows an example of a link where we have added 2 views; 1 block type menu is displayed. To achieve proper arrangement of the elements it may be necessary to modify the **CSS styles of the theme**.

F40.27

OM Maximenu Menu results

An example of an expanded link showing containing 2 views and one menu.

40.3 Scheduling events with Calendar

Amoing the modules available in Drupal for event management and schedule presentation, we have selected the Calendar module for easy integration with the views and content of the site.

The **Calendar** module is available at:

http://drupal.org/project/calendar

Calendar requires the installation of the following modules:

- **Views 3**, available at: http://drupal.org/project/views
- **Chaos tool suite (ctools)**, available at: http://drupal.org/project/ctools
- **Date API** and **Date Views**, both are include in **Date**, available at: http://drupal.org/project/date

Once you have installed the **Calendar** module, create a new calendar view from the link "**Add a view from template**", selecting the right calendar depending on the content type and date field that must be shown in the calendar.

The Calendar view includes several page type displays representing **Monthly, Weekly, Daily** and **Yearly** calendar formats.

The paths for each of the dispays are as follows:

- **Month view**: /calendar/month
- **Week view**: /calendar/week
- **Day view:** /calendar/day
- **Year view:** /calendar/year

The view also includes two **Blocks**:

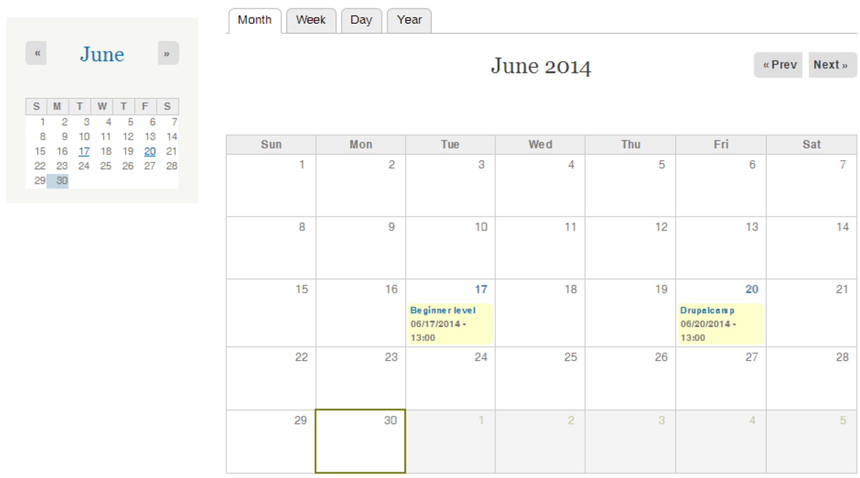
- **Block**: Displays a monthly calendar.
- **Upcoming**: displays a list of upcoming events.

By default the calendar displays all the published nodes according to the modification date of the node. We can modify the view and apply new filters so that the calendar will display only the desired content.

Figure F40.28 shows the calendar is available at **/calendar/month**. The calendar has several views, shown by tabs: month, week, day and year.

In the same Figure we show the **View: Calendar: Block**. It shows that in the block when there is an event published the link style is changed. Clicking on a calendar day will display a list of the available events for that day.

As an example we will create a new content type **Event** and modify the calendar to only display Event content types depending on the date of the event (and not on the publication event of the node).



F40.28

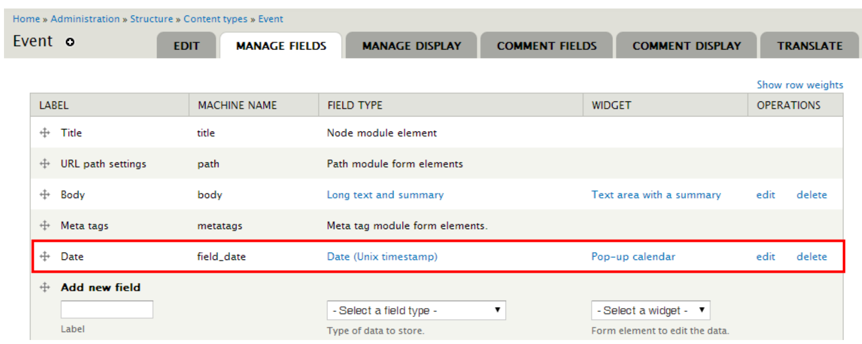
Calendar

Calendar view

The Calendar view is shows with the montly page display and the block added by the Calendar module.

Create a content type Event

We will create a content type **Event** and add a new field called **Date** (system name **field_date**), of the type **Date** and of control type **Pop-up calendar** (requires the activation of the **Date Popup** module, included in Date). F40.29



F40.29

Calendar

Event Content Type

Event content type with Date field.

In the options field indicate that the field will have a mandatory end date.

To complete this step we will create some events with different dates.

Calendar view

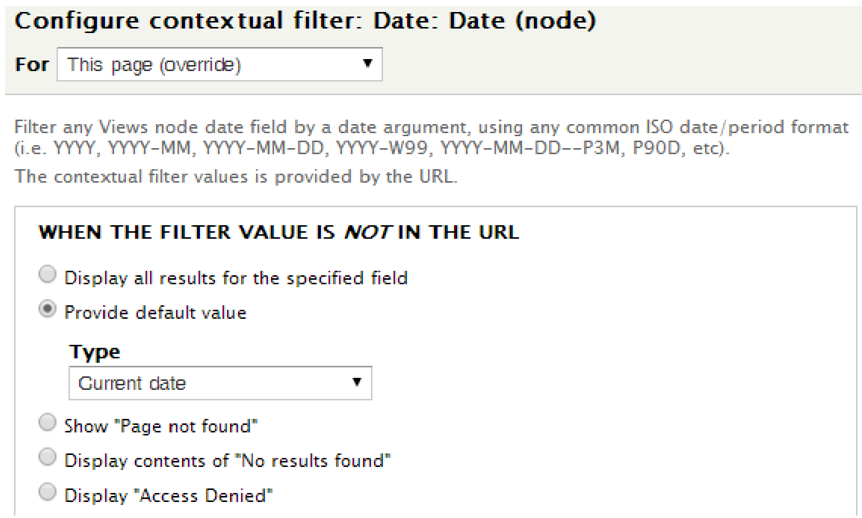
As we created the calendar view from template we don't need to modify the next parameters. Use this only if you need to modify the view by adding other date fields (for example if we need to add different content types into the same calendar).

- In **contextual filters** edit the current filter, while keeping the configuration: **Provide default value** to the type **Actual Date**. In **Dates fields** select the start and end dates of the **Date** field for the Event content type. F40.30 F40.31

F40.30

Calendar Contextual Filter

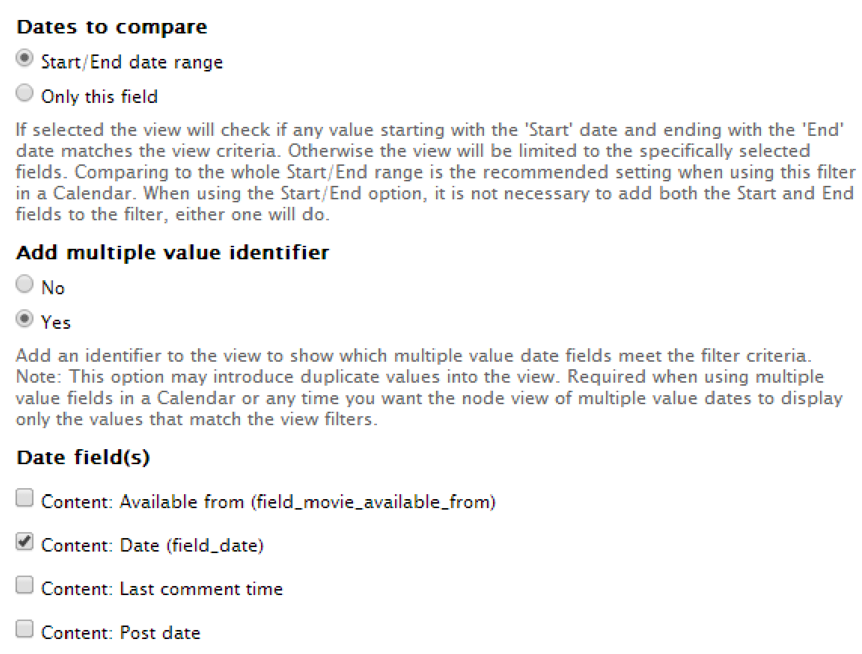
Change the contextual filter to filter the new field Date in the Event content type.



F40.31

Calendar Contextual Filter

Select the start and end dates for the Date field in the Event content type.



Administration Modules

40.4

In this final section, we will review some related site administration modules.

Admin Module

The **Admin** module improves administration navigation by providing a configurable sidebar.

The **Admin** module is available at:

http://drupal.org/project/admin

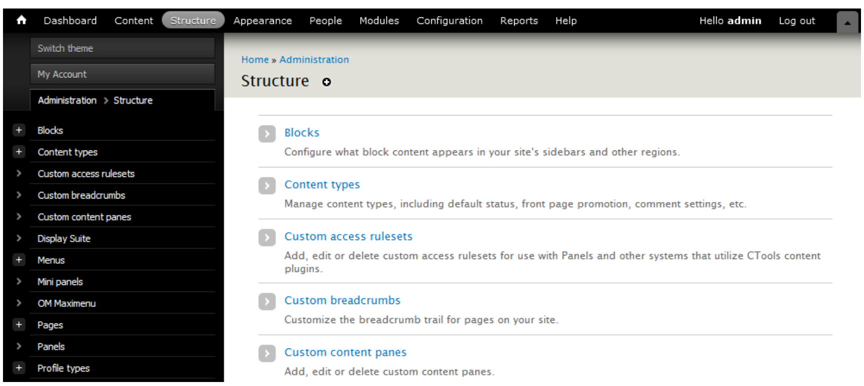
The menu is activated automatically without the need to configure it, but it can be personalized here: F40.32

Administration ⇨ Configuration ⇨ User interface ⇨ Administration tools

URL Administration tools /admin/config/user-interface/admin

From the settings, you can specify whether the menu will be displayed Vertically or Horizontally as well as where it will be placed (by default **top left corner**).

We can also configure the blocks that are displayed, disabling them or adding new ones.



F40.32
Admin
General admin menu for the Admin module

Elysia Cron Module

The **Elysia Cron** module allows you to manage cron tasks independently. Another utility included is the ability to run cron on a user visit, without having to configure the cron system.

The module is available at:

http://drupal.org/project/elysia_cron

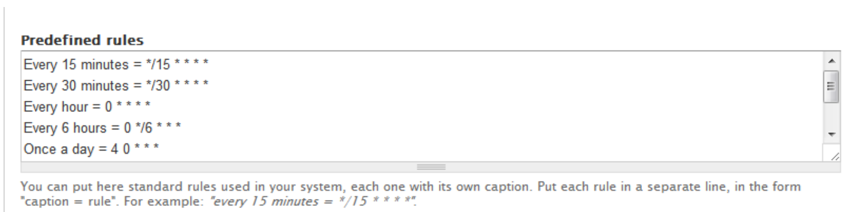
We can access the configuration here:

URL Cron Settings
/admin/config/system/cron

Administration ⇨ Configuration ⇨ System ⇨ Cron Settings

From the configuration, we can define a set of preconfigured rules, which consist of a title and its corresponding temporal rule. F40.33

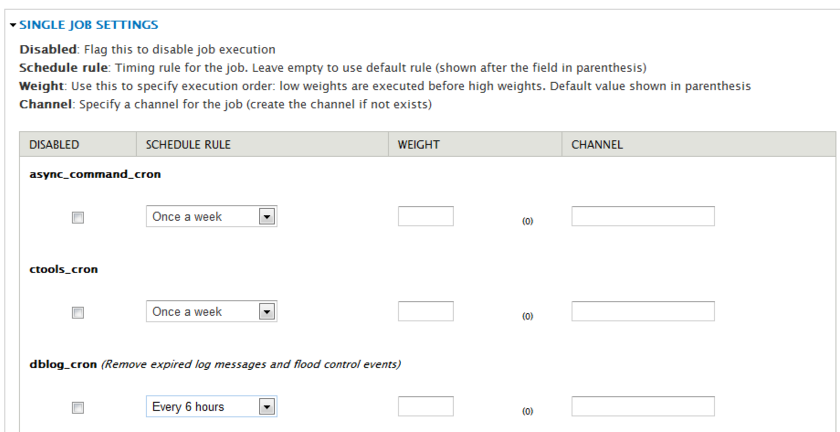
F40.33
Elysia Cron Rules
We can define execution rules for each cron job.



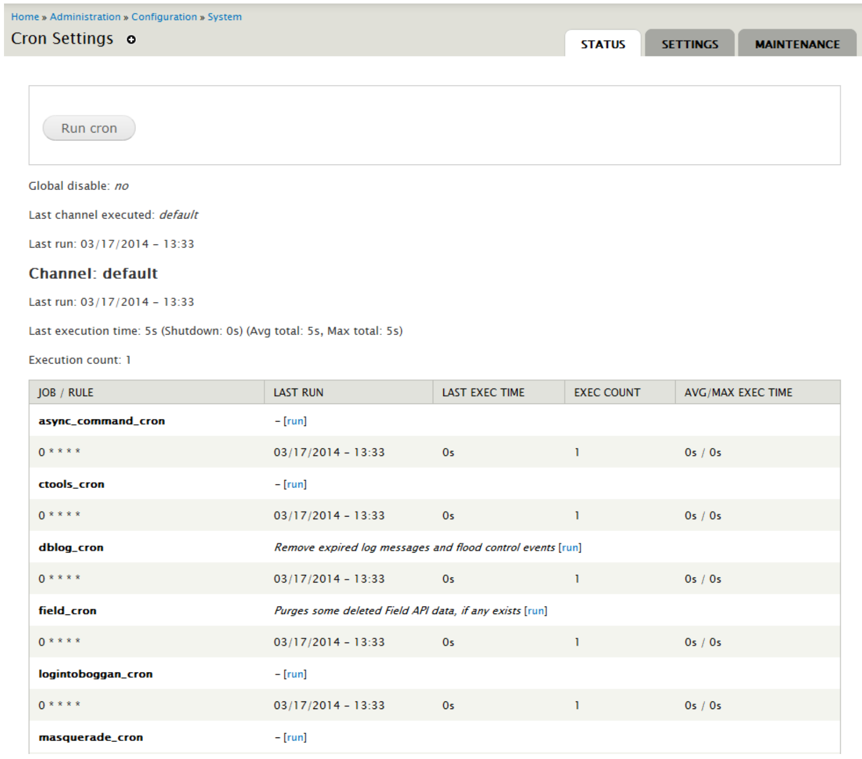
These rules can then be applied to each cron job independently. In addition, the tasks can be organized by the order of execution.

From the configuration, you can also disable the cron jobs you don't want to run. F40.34

F40.34
Elysia Cron Independent Jobs
A list of the jobs implemented in Cron. We can arrange execution based on rules and weight.



On the **Status** tab, the execution status of each cron job is displayed. From here, we can also run the complete cron list by clicking on **Run cron**, or run each job independently by clicking on its [run] link. F40.35



F40.35

Elysia Cron Status

On the status tab, the list of tasks along with information from the last run is shown.

From this tab you can run the complete cron or execute each task individually.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

A Configuration of the hosting and necessary tools

In this annex we present a short introduction to the tools that will be necessary for managing the hosting. Reference will only be made to the functionalities that are useful for managing the website with Drupal.

Hosting dashboard A.1

In this annex we present a short introduction to the tools that will be necessary for managing the hosting. Reference will only be made to the functionalities that are useful for managing the website with Drupal.

When we contract a Hosting service, the provider will give use access to a **Dashboard** from where we will be able to administer the different functionalities available, depending on the hosting plan contracted.

Some typical actions that we will be able to do from the dashboard are:

- Check the status of our account: disk space occupied, monthly bandwidth transfer, number of e-mail accounts, number of sub-domains, FTP accounts, Databases, etc.

- Register domains and sub-domains.

- Create FTP accounts.

- Create MySQL users and databases, and administer them using phpMyAdmin.

- Create e-mail accounts with the domains available.

- Configure Cron tasks that will be run automatically every now and then.

There are different dashboard applications for hostings, of which the most well-known are **cPanel** and **Plesk**. The application available will be the one facilitated by the service provider.

In this section we will study some functionalities of the dashboard with the **cPanel** software, although these are fully applicable to other types of dashboards.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Dashboard access

Generally access to the dashboard will be done from the browser via an URL of the www.example.com/cPanel type, with www.example.com being the main domain of our website. This piece of data, along with the user name and password, will be provided by the service provider.

In **Figure** FA.1 we see the typical form for access to cPanel.

FA.1

cPanel. Access form

cPanel access form.

Reaction: Regions

Disable theme regions when the context is active.

The dashboard is divided into two main sections. The column on the left usually shows the status of the hosting and the resources used and available. The column on the right shows all of the functionalities available, grouped together by categories. The available settings will vary depending on the services contracted. FA.2

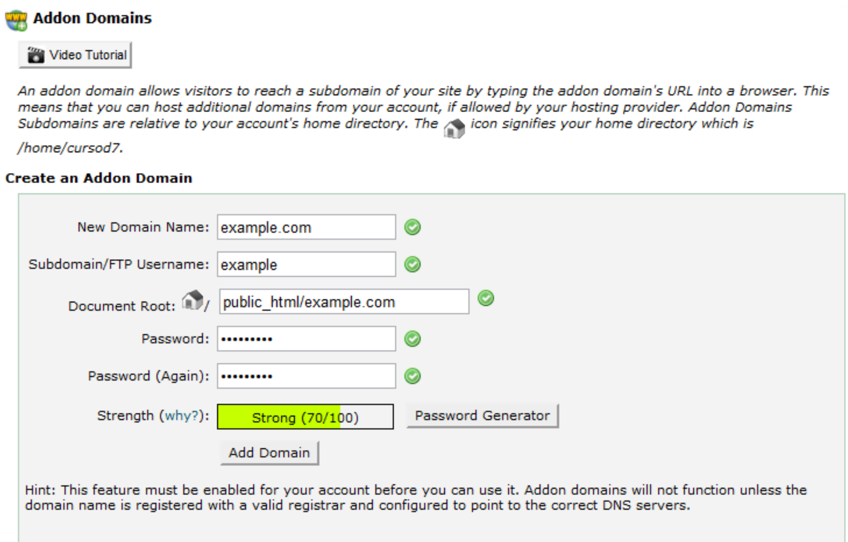
FA.2

cPanel. Overview

Overview of the cPanel dashboard.

Add domains and sub-domains

From the **Add on domains** setting we can add new domains to the hosting server, which we will have acquired beforehand via a domains provider. FA.3 When an add on domain is created, we have to state the folder that this domain will point to. A main FTP account will be created along with the domain, with the user name and password that are indicated.



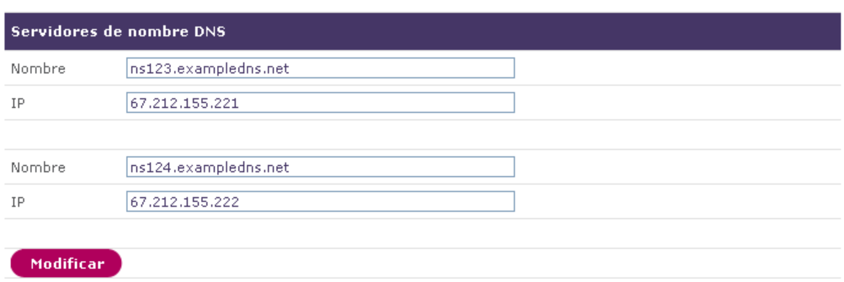
FA.3

cPanel. Add a domain

When a domain is added, an FTP account will be automatically created for accessing the folder that is indicated.

In order for the domain to begin functioning, we have to indicate this to the hosting server. This is done by means of the name servers or DNS. In **Figure** **FA.4** we see an example of the registration of DNS servers associated with the domain. This configuration is not applied using the dashboard, but rather in the administration area of the domains provider where we have acquired ours.

The DNS names (typically 2) and their corresponding IP addresses must have provided by the hosting provider.

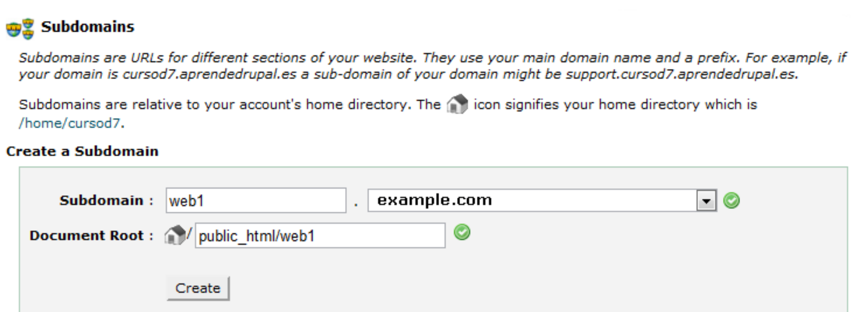


FA.4

cPanel. Point to the domain

We will use the name servers or DNSs to point to the domain of our hosting.

Once a domain has been added, we can also create subdomains (such as for example, web1.example.com, web2.example.com, etc.). We will add these from the Subdomains setting, indicating the name of the subdomain, the associated domain and the folder that this will point to. **FA.5**



FA.5

cPanel. Create sub-domains

We can add sub-domains to any available domain or sub-domain that has previously been created.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Once a domain or subdomain has been created, the files uploaded in the corresponding files will be available via the URLs:

- http://www.example.com
- http://web1.example.com

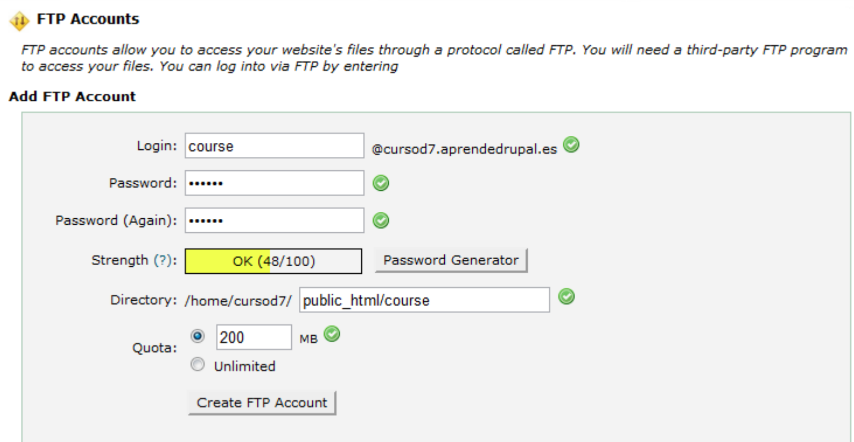
Create FTP accounts

An FTP account allows us to access the files of particular folders of the server, by using a FTP software (such as **FileZilla**). To create an FTP account we have the **FTP Accounts** setting available, where we will have to state the user name, the password and the folder that he will have access to. Optionally, we can also indicate a quota or limit on the disk space that the account created will use. FA.6

FA.6

cPanel. FTP Accounts

The FTP accounts will enable us to manage the files of the server (upload and download files).



Create MySQL databases

From the **MySQL Databases** setting FA.7 we can create databases and the users associated with these. FA.8

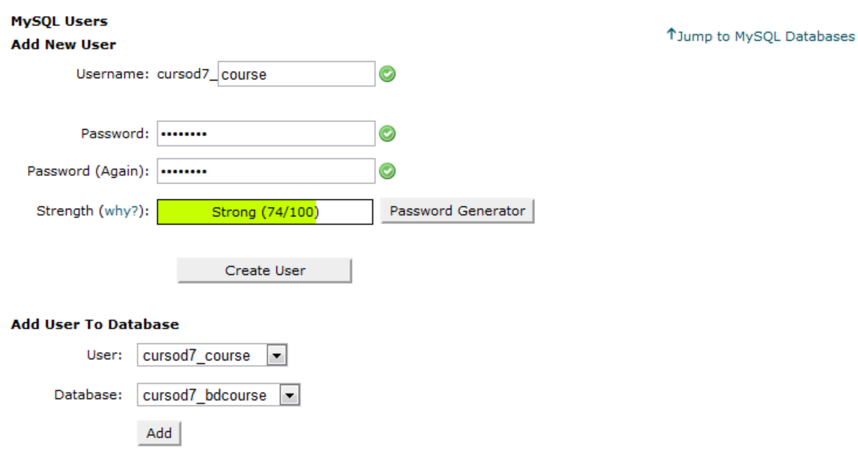
FA.7

cPanel. Create databases

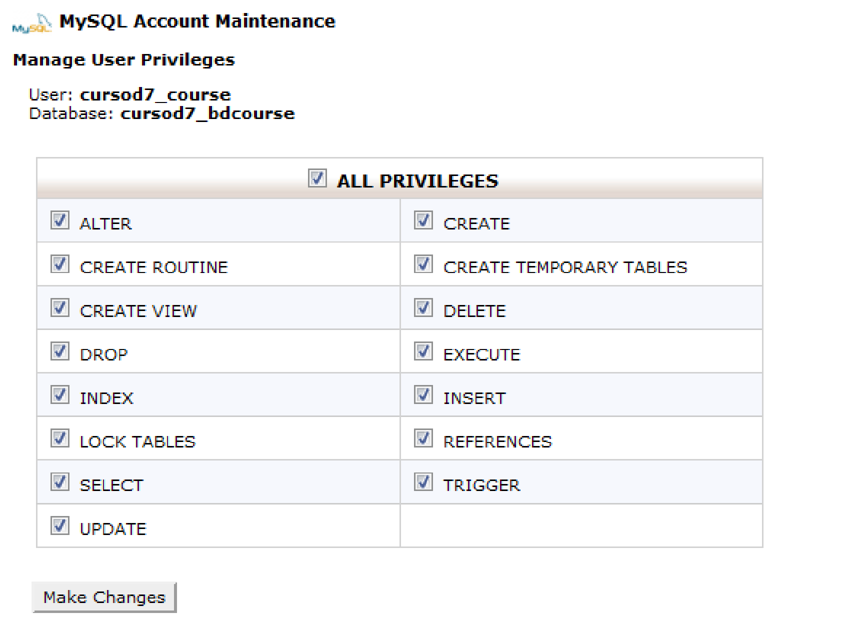
FA.8

cPanel. Create database user

To access a database it is necessary to create a MySQL user, with the right permissions.



The next step will be Add User to database, which means that we will give permissions to the user so that he can access that database, stating the exact privileges that he will have with respect to this. FA.9



FA.9

cPanel. Assign permissions to the database user

When the database is assigned to the user we have to state which privileges he will have over it.

Practical Case 31.1 Adding Action

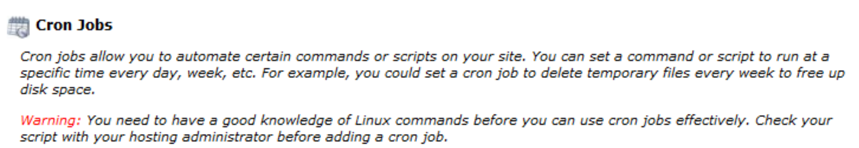
In the SUBJECT field of mail, we may use replacement patterns.

Any available variable may be used, using brackets [variable].

Once the user privileges have been granted, this will be ready to make use of the databases, whether directly or in any application that makes use of databases, such as is the case with Drupal.

Enable the Cron

We can create tasks that will periodically run commands from the **Cron jobs** setting. Of the two methods available, **Standard** and **Advanced**, we will only see the **Standard** option in this course, which is sufficient to enable and configure the Drupal cron. FA.10



FA.10

cPanel. Cron Mode

We will only see the Standard cron configuration mode.

Practical Case 31.1 Adding Action

Lastly, we will indicate that the mail is sent from the current user in the FROM field.

Practical Case 31.1 Adding Action

Lastly, we will indicate that the mail is sent from the current user in the FROM field.

Figure A.11 shows the form for configuring a cron task. We must firstly enter the command that is going to be run.

Generally speaking, the Linux curl command is used to load an URL, and hence the full command for running the cron of our site (www.example.com) will be:

curl http://www.example.com/cron.php

We will then select the time interval that we want for running the cron on our site.

In **Figure** FA.11 this has been configured to run every day at 3:05 in the morning. As far as possible it is recommended that the cron is run at times at which the site activity level is low, for example in the early hours of the morning.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

FA.11

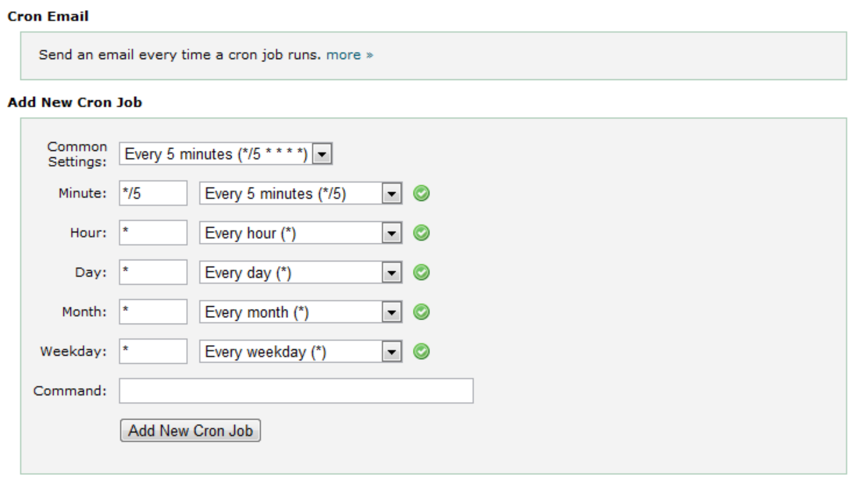
cPanel

The curl command allows us to run any URL of the system. To configure the cron, it is sufficient to state which periods we want to run it in.

Practical Case 31.1 Adding Action

An e-mail will be sent to the node author where the comment has been published.

We will find the variable that has the node author's e-mail in the TO field.



If the **curl** command does not function on your hosting, check which command you should use with your provider, because this may vary depending on the operating system and the configuration of the server.

Once the cron has been enabled, check in the site **status report** that the cron is being run correctly, in the set time periods.

A.2 Managing databases with phpMyAdmin

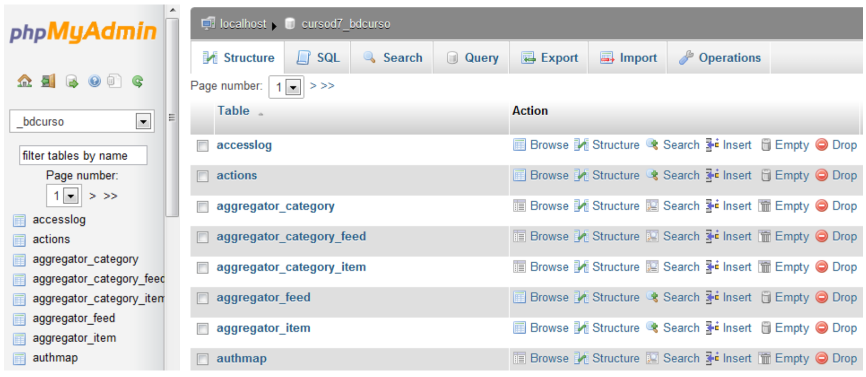
From the dashboard it is possible to access the management of the databases created using the **phpMyAdmin** software. To enlarge upon the information about this tool, please consult the official page of **phpMyAdmin** at http://www.phpmyadmin.net. **FA.12**

The main settings that we will find in **phpMyAdmin** are:

- **List of databases and tables** (in the left-hand column). This makes it possible to select a particular database and/or table.
- **Structure.** If a database is selected, this displays the set of tables that makes it up. If one particular table is selected, it displays the fields and items that make it up.
- **Query.** This allows us to navigate through the data or records contained in the selected table.
- **SQL.** This allows us to launch SQL sentences (select, insert, update, delete, etc.).
- **Search.** Help for searching for data contained in the table, using different filters related to the fields of the table.
- **Insert.** Form for inserting data or logs into the table.
- **Export.** This makes it possible to export a database or a table. In both cases it is possible to export from the structure, from the data, or both. To make a database backup we will carry out an export of both the structure and of the data. The resulting file is a file with SQL sentences, which can be used in order to Import, into the same database or into another one.
- **Import.** This allows us to import an item and data from previous exports (SQL files).
- **Operations.** This allows us to carry out additional operations on a database or on a table (change name, copy, etc.)

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

- **Clear.** Deletes all of the data from a table.
- **Delete.** Deletes a table (structure and data).



FA.12

phpMyAdmin

Main view once a database has been selected.

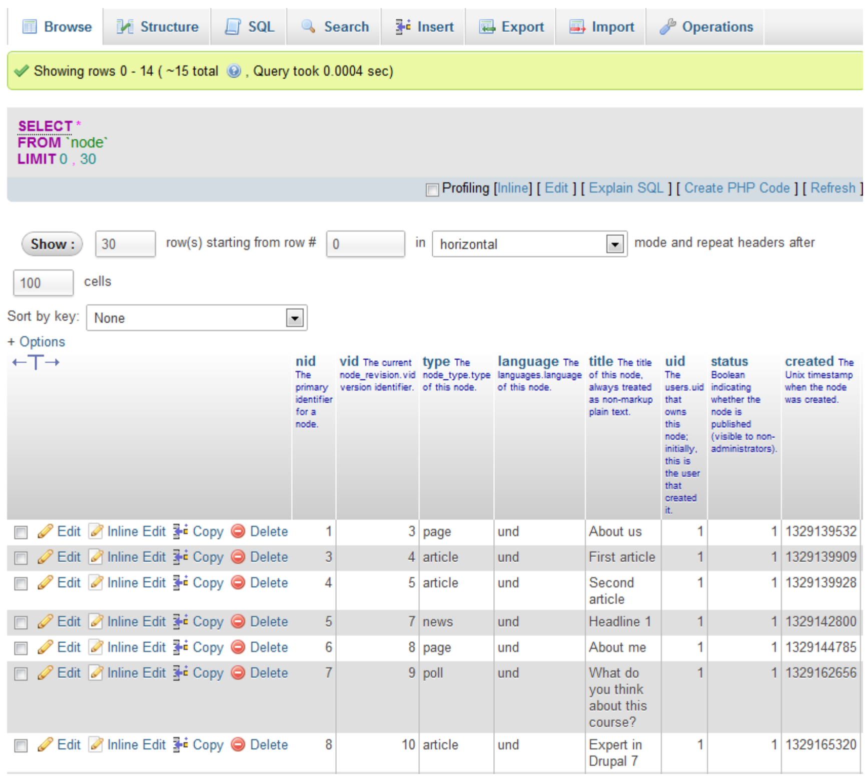
Practical Case 31.1 Adding Action

We can also use replacement patterns in the MESSAGE field of mail.

Practical Case 31.1 Adding Action

We can also use replacement patterns in the MESSAGE field of mail.

In **Figure** **FA.13** we see the content of the **node** table of Drupal, once the contents have been created on the site. To access the content of the table we have first selected the database, then the **node** table and lastly the **Query** tab.



FA.13

phpMyAdmin

Data contained in the node table (Query tab).

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

A.3 Upload and download files via FTP

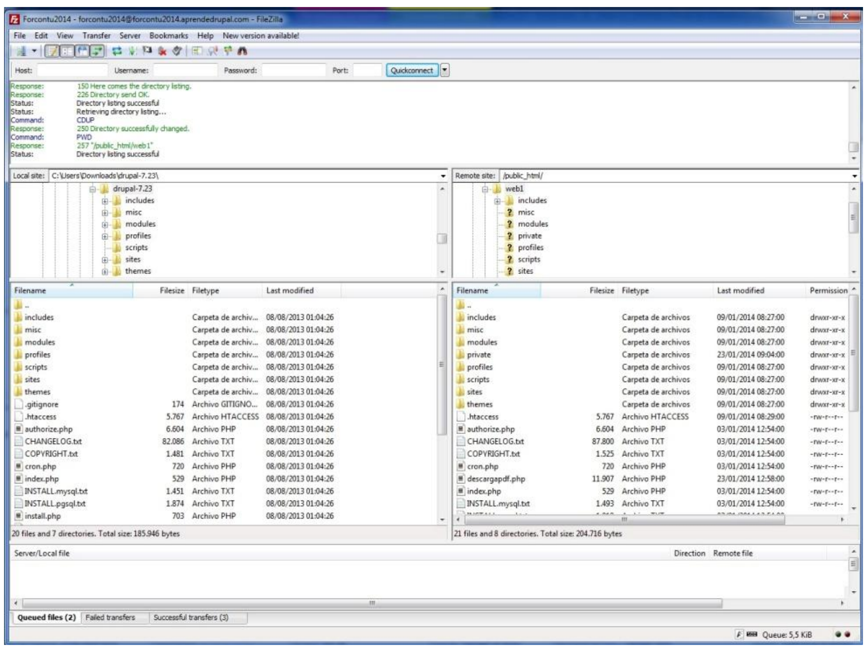
The software that enables us to upload and download files to a remote server is known as **FTP** (*File Transfer Protocol*). There are many alternatives, and they all work in practically the same way. Here we will analyse how the **FileZilla** software works, as this is freeware that is widely used.

FileZilla is available for downloading at http://filezilla-project.org/, by using the *Download FileZilla Client* link.

In **Figure** FA.14 we see an overview of the software. The central part of the window is split in two. On the left we see the files and folders of our operator (local) and on the right we see those belonging to the server. We can browse through both structures, and move the files and folders from one site to another one.

FA.14
FileZilla
FileZilla is a piece of freeware for uploading or downloading files by FTP.

Practical Case 31.1
Testing the Rule
The rule will be executed when the registered user publishes a comment. The node author will receive an e-mail like the one shown.

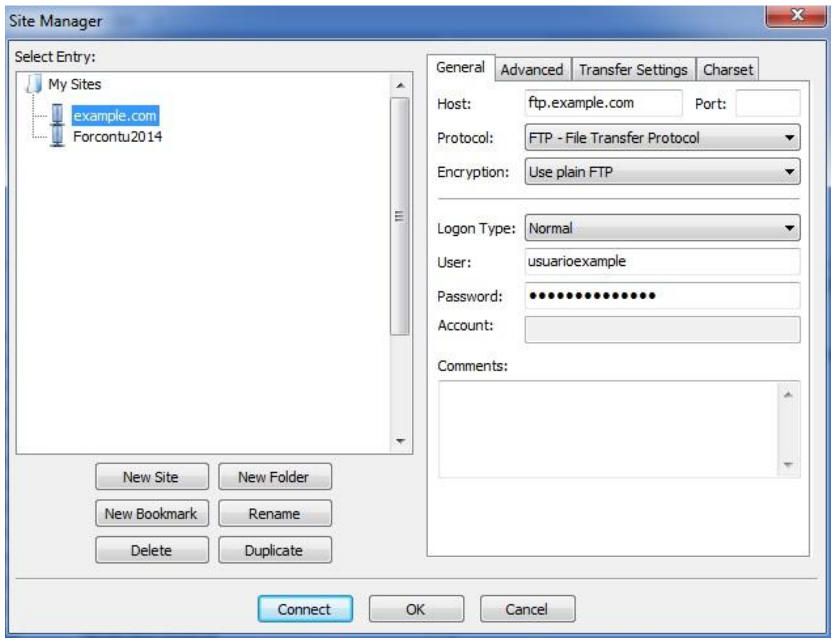


Set up an FTP connection

In addition to the server's name or IP address, to set up a connection with an FTP server we need a valid user and password. In **section A.1** we saw how to create an FTP user to access the files of our hosting server.

From **File⇨Site manager** we can access the connections management window that enables us to create and save the usual connections. FA.15

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



FA.15

FileZilla

The **Sites manager** enables us to create and save the usual connections.

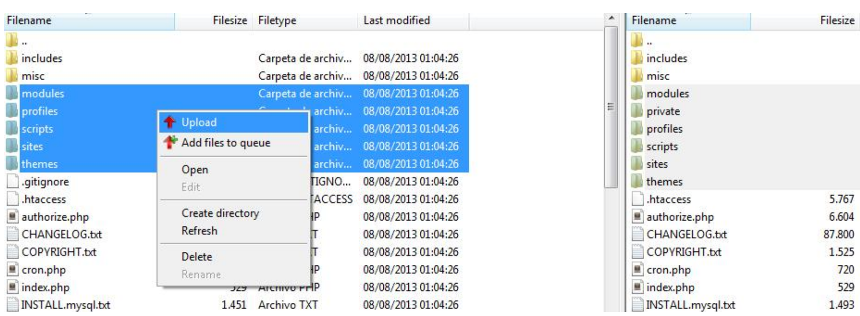
Filter rules

We can filter by event or tag from the rules administration area.

Upload and download files

We will take the following steps to **upload files to the server** from our local computer: **FA.16**

- In the window on the right (server) we will move through the structure of folders till we reach the folder that we want to upload the files into.
- In the window on the left (local computer), we will move in the same way through the structure of folders till we locate the files that we want to upload to the server. We will then select the files and folders and by clicking on the right-hand button we will choose the **Upload** option.



FA.16

FileZilla

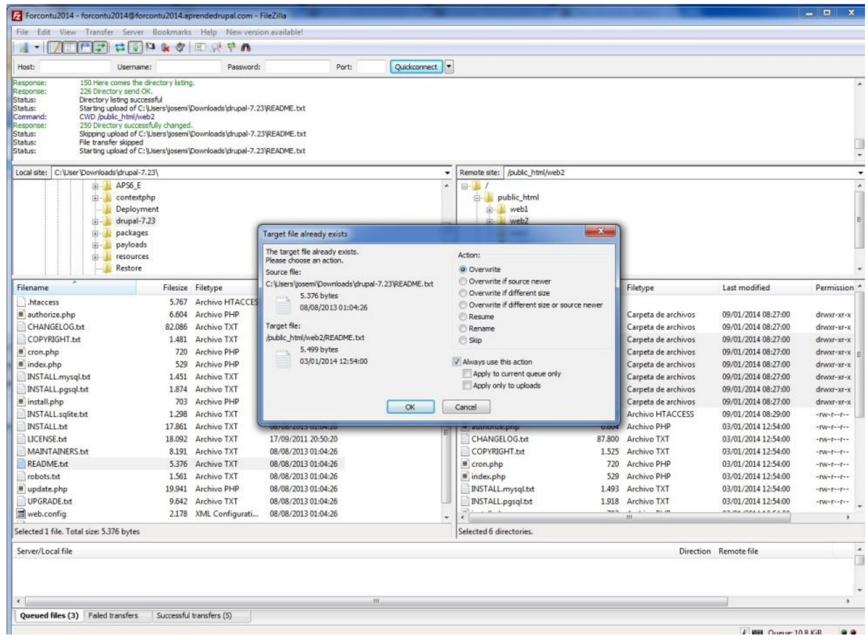
To **Upload files** to the server, select the files to be uploaded on the left (local).

If the files or folders exist on the server, a window will be displayed that asks us whether we want to overwrite the files. The "Always use this action" option will stop us being asked this question for every individual file. **FA.17**

FA.17

FileZilla

Use the "Always use this action" option to overwrite multiple files.



To **download files from server** to our local computer, we will take the following steps: **FA.18**

- In the window on the left (local computer), we will move through a structure of folders until we reach the folder that we want to download the files to.

- In the window on the right (server), we will move in the same way through the structure of folders till we locate the files that we want to download. We will then select the files and folders and clicking with the right-hand button we will choose the Download option.

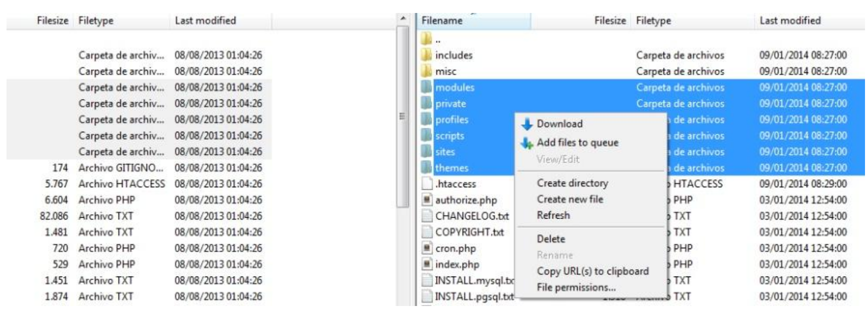
FA.18

FileZilla

To **Download files** from the server, select the files to be downloaded on the right (server).

Login one time Module

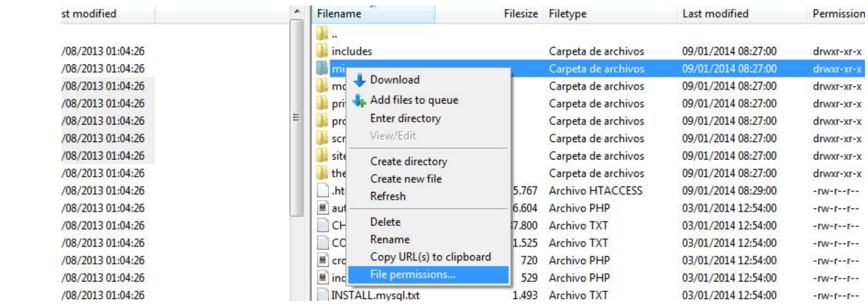
The e-mail containing the one-time use access link that will be sent to the user, can be edited from Account Options



Modify the permissions of files and folders of the server

A value is shown on the server side, along with each file or folder, that relates to the permissions that are established (775, 644, etc). To modify the permissions of a file or folder, select it and click with the right-hand button, and then choose the **File permissions** option from the dropdown menu. **FA.19**

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



FA.19

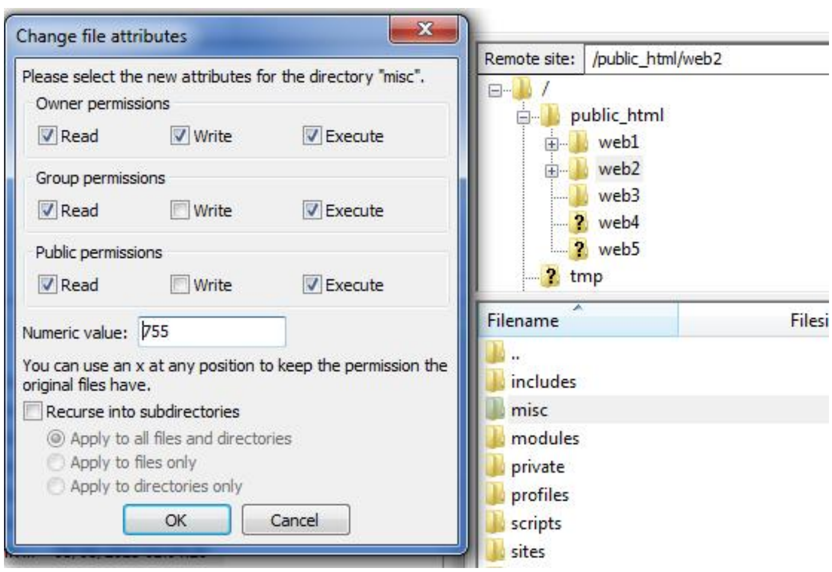
FileZilla

Access **File permissions** to modify the file reading, writing or run permissions

Login one time Module

When sending a link to a user we will establish what page the user will be redirected to when accessing the site.

In **Figure** **FA.20** we see the permissions that can be assigned to a file or folder, which can be read, write or run. We have to take special care with the write permissions.



FA.20

FileZilla

Special care must be taken with the writing permissions that we assign to the files. Drupal protects some files and folders, and so if we want to make any modification in these we will first have to modify the permissions using FileZilla.

Login one time Module

Configuration of the elements that will be displayed on the user's profile page.

Drupal protects certain folders and files of the system (such as /sites/default and /sites/default/settings.php) so as to stop it being possible for those files to be modified maliciously. If we want to create a folder within /sites/default, we first have to change the permissions to the default folder, in such a way that allows for reading (owner permissions and group permissions, generally speaking). Once the folder (for example, modules or themes) has been created, we will be able to change the permissions of the default folder again.

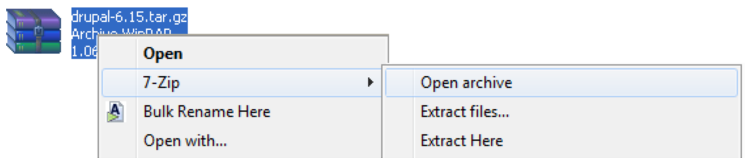
The same will happen when we want to modify the settings.php file. It will be necessary to modify the permissions of both the folder that contains it (default) and of the settings.php file itself. Once we have made the changes we will apply the initial permissions again.

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

A.4 Uncompress files with 7zip

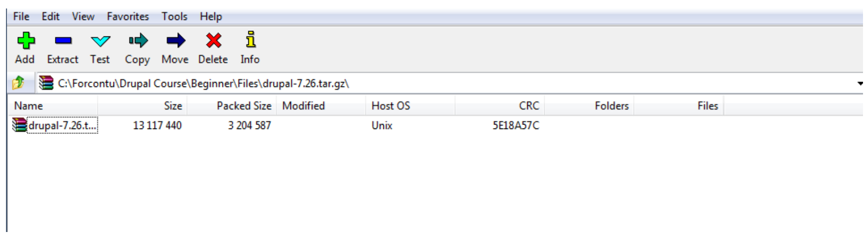
During the course you will find that the files downloaded from drupal.org are compressed in the **.tar.gz** format. We can use any file compression and uncompression software to uncompress these files, such as WinZip or WinRAR. In this section we will see how to use **7zip**, a piece of freeware available at n http://www.7-zip.org/.

To uncompress a **.tar.gz** file, it is necessary to take account of the fact that this is a dual compression: first the files and folders are grouped together into one single file (**tar**) and then the resulting file (**gz**) is compressed. This is why we will have to do a dual uncompression.



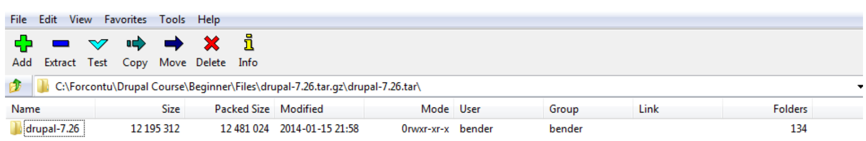
FA.21
7zip
Uncompress a .tar.gz file with 7zip.
Login one time Module
Users with permission to send access links will see the corresponding button on each users profile, along with the rest of the options that have been selected in the module configuration.

When the compressed file is opened, as shown in **Figure** FA.21 (right click on the file and open with **7-zip** or first opening **7-zip** and selecting the file to decompress), the file will be uncompressed on one occasion, and giving the result of the **.tar** file. FA.22



FA.22
7zip
In the first unzipping we obtain the .tar file, which continues being a compressed file.

We will reach the end folder by double clicking on the **.tar** file. From this point we will browse through the structure of files and folders contained in the compressed file and uncompress or extract just the files that we want. To do this it is enough to select them and drag them to the folder we want (on our computer) or to click on the **Extract button**, indicating the destination folder. FA.23



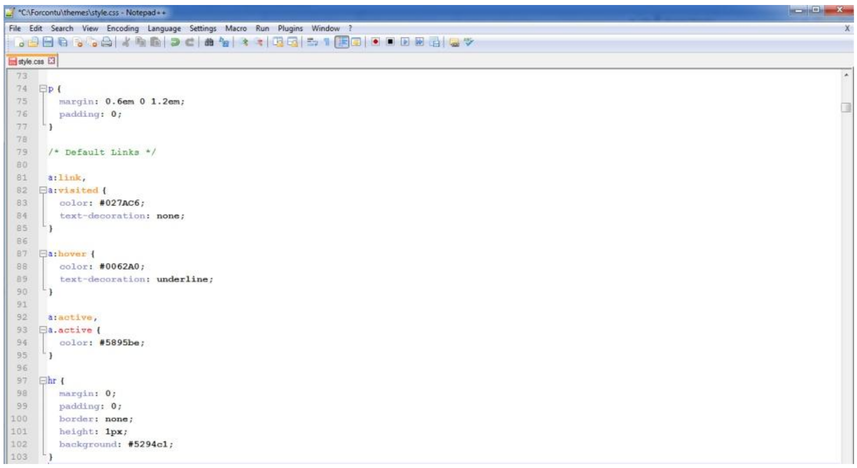
FA.23
7zip
On the website, by double clicking on the .tar file we arrive at the content of this. To finish the unzipping we can use Extract or drag and drop the content into any folder of the system.

Editing text files with Notepad++

A.5

Notepad++ is a freeware text editor for Windows, available to be downloaded at the URL http://notepad-plus.sourceforge.net. It is a very useful editor for editing PHP, HTML and/or CSS files, because this shows in the syntax in colour, the line number and the nestings between items. It is moreover possible to choose the language in which the file is written from among a large range of options (PHP, CSS, XML, HTML, SQL, etc.).

Figure FA.24 shows us a CSS file, edited with **Notepad++**.



FA.24

Notepad++

Notepad++ shows the structure of the file in diagrammatical file, colouring the key words depending on the type of the language: PHP, HTML, CSS, etc.

B Links and resources of interest

In this annex you will find links to pages and resources of interest so as to properly follow this course and to learn Drupal in general.

forcontu.com B.1

www.forcontu.com is the official page of the course. From this page, you will be able to download the materials, access the classrooms and obtain **additional resources.**

Furthermore, you will be able to **contract other services that are supplementary** to this book:

- **Supervised course.** By contracting the supervise course, you will have the continuous support of the tutors, who will guide you in carrying out the activities proposed and in your personal project. The customised course includes the student assessment and the certificate of **Expert in Drupal 7**, once you have passed this. In addition, you will be able to contract this by separate or grouped levels, depending on your needs and previous knowledge.

- **Continuous training.** You do not stop learning Drupal when the courses end. Sign up for continuous training so as to continue periodically receiving Activities, Practical cases and all of the new features on new modules, themes and about Drupal in general. You get all of this with the same training focus as that applied in the course.

- **Consultancy.** If you consider that you need help to complete your website project or start a new project once you have finished the course, you will be able to contract the Consultancy service. This service is conceived of to help you, from a training focus, to make the best choice about modules and the configuration of your site, depending on the particular project that you are going to carry out. Your advisors will also help you to resolve the errors and conflicts that you may encounter when you install and configure new modules.

- **Website development.** At Forcontu, we have made a clear investment in sustainable development, and this is what we want clients to actively participate in the process of building their projects. In this way, we aim to convey to you the need to keep your project alive, adding new functionalities and innovations that are being adapted to the inevitable progress of the business and of the market, thus achieving an alignment between the business and the ITC tools that are implemented. If your company needs to develop a website project, do not hesitate to contact us.

B.2 Official Drupal page, drupal.org

On the official Drupal page, drupal.org, you will find very many of the resources studied (in English) on the course and that are necessary to carry out the installation and enlargement of Drupal.

- **Latest Drupal version.** You will be able to download the latest Drupal version from the site home page.

- **Additional modules.** From the **Modules** section, you can consult and download all of the free modules that are available.

  http://drupal.org/project/modules

- **Graphical themes.** You can consult and download all of the graphical themes available from the **Themes** section.

  http://drupal.org/project/themes

- **Translations.** You can consult and download all of the available translations of the core from the **Translations** section (section B.6).

  http://localize.drupal.org

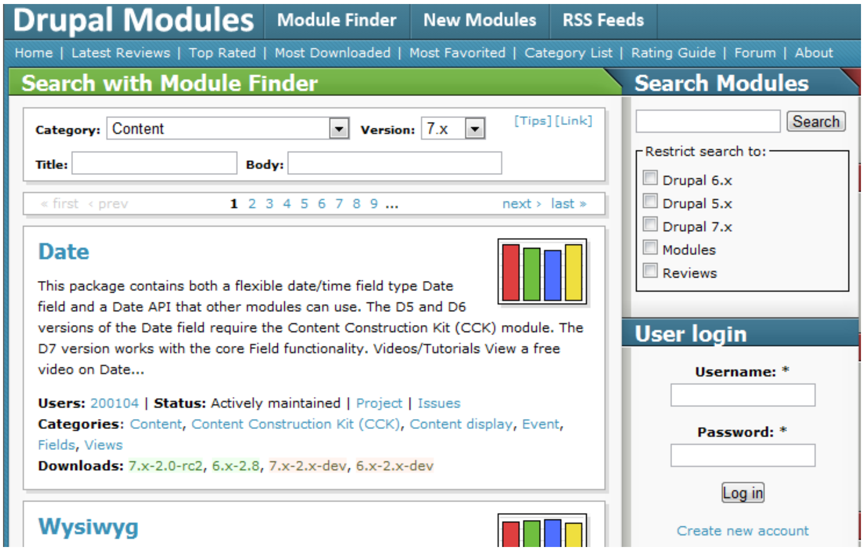
- **Forums.** In the Drupal forums you will find discussions (in English) about different issues related to Drupal (installation, updating, development of modules and themes, etc.).

  http://drupal.org/forum

B.3 drupalmodules.com

The drupalmodules.com website is the site that is recommended for searching for and downloading Drupal modules (in English). Unlike with drupal.org, this site is completely focused on modules for Drupal and it incorporates filters and search engines that make it possible to locate modules more smoothly and in a more straightforward way. In addition, it incorporates graphics that show the functionalities, security, ease of use and documentation on a scale from 1 to 5 for every module and, in some cases, comments from the users. The repositories are synchronised, and the versions of the modules available on both sites are completely identical. FB.1

Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079



FB.1

Repository of modules at drupalmodules.com

The drupalmodules.com website is exclusively orientated towards the Drupal modules. A lot of modules can be found with very useful search and query tools.

Drupal Association

B.4

The Drupal Association is an organization dedicated to helping the open-source Drupal CMS project flourish. We help the Drupal community with funding, infrastructure, education, promotion, distribution and online collaboration at Drupal.org. **FB.2**

Funds to support these programs and the Association staff come from memberships, supporting partners, sponsorships, donations, and volunteers.

If you want to work together with the Drupal Association you can become a member from https://association.drupal.org/.

FB.2

Drupal Association

Website of the *Drupal Association.*

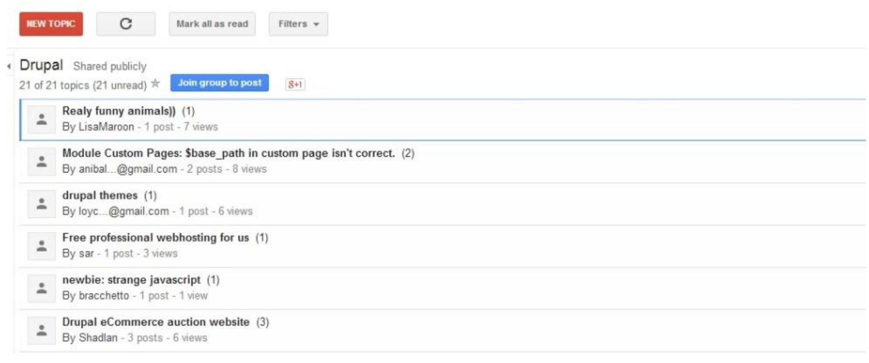
Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

B.5 Drupal Group on Google

Join the ***Google Drupal*** group, a meeting point for making technical enquiries and getting to know other members of the community. You will just need a Google account to be able to take part.

http://groups.google.com/group/drupal FB.3

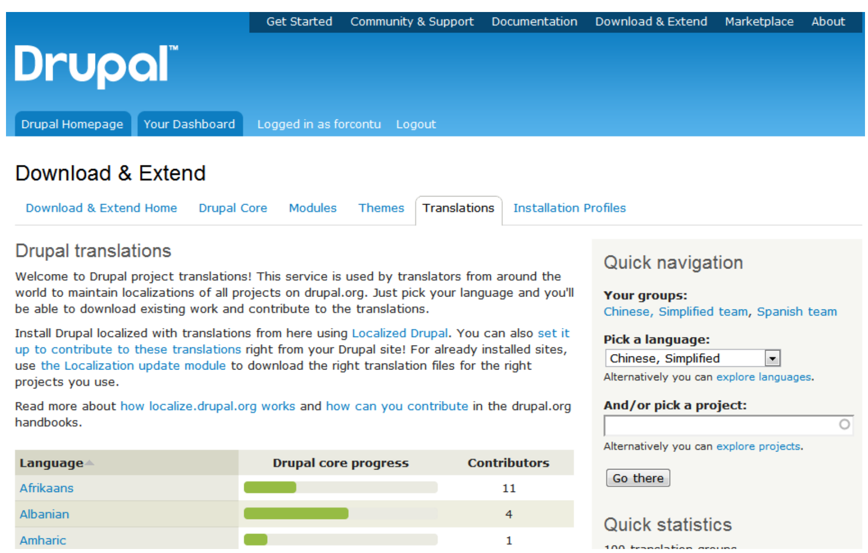
FB.3
Drupal Group
You will find the Drupal group in Google Groups.



B.6 Translations of Drupal, localize.drupal.org

Localize Drupal (http://localize.drupal.org/) is a new website created by Drupal and that is wholly devoted to translating both the Drupal core and the additional modules shared by the community into many languages. The translations are also done in community, where the participants propose translations that are then accepted and incorporated into the site. FB.4

FB.4
localize.drupal.org
The Localize Drupal website is a new Drupal website dedicated to the translation of Drupal and of the additional modules.

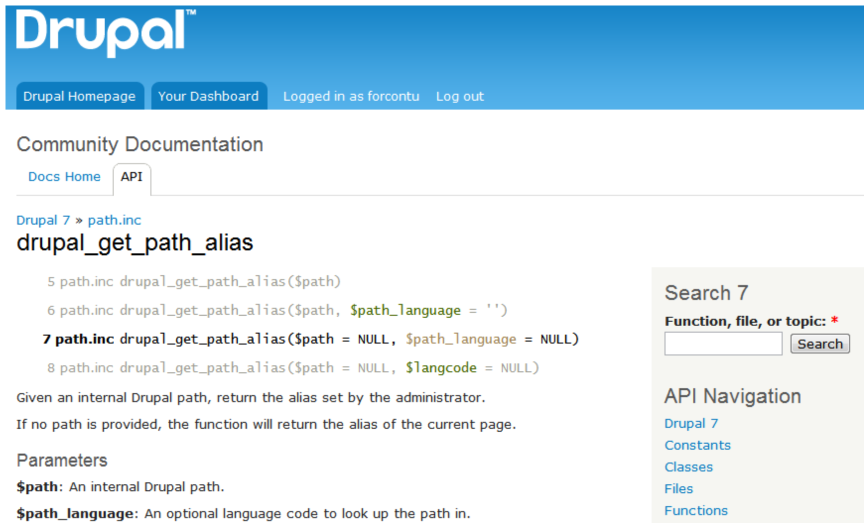


Copyright 2011-2015 Forcontu S.L. All Rights Reserved. No part of this book may be reproduced or transmitted in any form.
This book was prepared exclusively for Nidhi Badani. Verification code: D7INTPDFEN00037738007079

Drupal API, api.drupal.org

B.7

An API (Applications Programming Interface) is a library of functions that makes it possible to access particular data and functionalities of the system with no need to understand their internal structure. The Drupal API contributes communication functions with the system and with the core modules. The details of the Drupal API are available at api.drupal.org. Consult the **Advanced Level of the course.** FB.5



FB.5

api.drupal.org

The Drupal API is necessary to schedule Drupal modules. This will be exclusively studied on the Advanced Level of the course.

Drupal paid-for themes

B.8

In addition to the repositories of paid-for themes, it is possible to find sites that offer paid-for themes. Some of these sites are:

- **Template Monster**, http://www.templatemonster.com/drupal-themes.php
- **Themeforest**, http://themeforest.net/category/cms-themes/drupal

Generic texts for tests

B.9

When we design a website it may be very useful to have generic texts of tests available that simulate actual texts with paragraphs, lists, etc.

One standard that is very widespread among website design and layout professionals on Internet is the text known as **Lorem Ipsum**. Through the www.lipsum.com page, it is possible to generate **Lorem Ipsum** text blocks with different characteristics (number of paragraphs, number of words, number of characters or lists).