

Publican 2.6

Guida Utente

Publicare libri, articoli, relazioni e
raccolte di volumi con DocBook XML



Publican

BOOK PUBLISHING TOOL

Don Domingo

Brian Forté

Rüdiger Landmann

Joshua Oakes

Joshua Wulf



Publican 2.6 Guida Utente

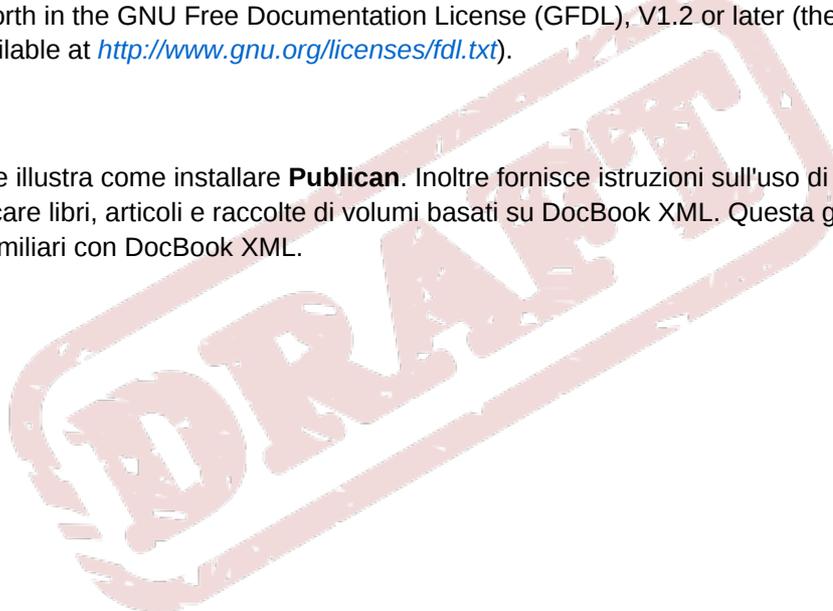
Publicare libri, articoli, relazioni e raccolte di volumi con DocBook XML

Edizione 0

Autore	Don Domingo	ddomingo@redhat.com
Autore	Brian Forté	bforte@redhat.com
Autore	Rüdiger Landmann	r.landmann@redhat.com
Autore	Joshua Oakes	joakes@redhat.com
Autore	Joshua Wulf	jwulf@redhat.com
Editor	Brian Forté	bforte@redhat.com
Editor	Rüdiger Landmann	r.landmann@redhat.com
Revisore tecnico	Jeff Fearn	jfearn@redhat.com
	Josef Hruška	

Copyright © 2010 Red Hat, Inc This material may only be distributed subject to the terms and conditions set forth in the GNU Free Documentation License (GFDL), V1.2 or later (the latest version is presently available at <http://www.gnu.org/licenses/fdl.txt>).

Questo manuale illustra come installare **Publican**. Inoltre fornisce istruzioni sull'uso di Publican per creare e pubblicare libri, articoli e raccolte di volumi basati su DocBook XML. Questa guida assume che si sia già familiari con DocBook XML.



DRAFT

Prefazione	vii
1. Convenzioni del documento	vii
1.1. Convenzioni tipografiche	vii
1.2. Convenzioni del documento	viii
1.3. Note ed avvertimenti	ix
2. Occorrono commenti!	x
Introduzione	xi
1. Installare Publican	1
1.1. Sistemi Operativi Linux	1
1.1.1. Fedora	1
1.1.2. Red Hat Enterprise Linux 5	1
1.1.3. Ubuntu	2
1.1.4. Debian	2
1.2. Sistemi Operativi Windows	3
2. Comandi di Publican	5
2.1. Opzioni di comando	5
2.2. Azioni	5
3. Creare un documento	9
3.1. I file nella directory del libro	10
3.1.1. Il file publican.cfg	11
3.1.2. Book_Info.xml	21
3.1.3. Author_Group.xml	26
3.1.4. Chapter.xml	27
3.1.5. <i>Nome_Doc.xml</i>	28
3.1.6. <i>Nome_Doc.ent</i>	29
3.1.7. Revision_History.xml	32
3.2. Aggiungere immagini	32
3.3. Aggiungere codice	34
3.4. Aggiungere file	35
3.5. Preparare un documento per la traduzione	35
3.6. Creare un documento	37
3.6.1. Compilare un documento senza controllo di validità	39
3.6.2. Compilare un documento creato con Publican 0	39
3.7. Creare il pacchetto di un documento	40
3.7.1. Tipi di pacchetti RPM	40
3.7.2. Il comando publican package	44
3.8. Tagging condizionale	46
3.8.1. Tagging condizionale e traduzione	48
3.9. Software pre-release e documentazione draft	49
3.9.1. Denotare il software pre-release	49
3.9.2. Denotare la documentazione draft	50
3.9.3. Denotare come draft la documentazione per software pre-release	50
4. Branding	51
4.1. Installare un brand	51
4.2. Creare un brand	52
4.3. File nella directory brand	53
4.3.1. Il file publican.cfg	53
4.3.2. I file defaults.cfg e overrides.cfg	54
4.3.3. File publican-brand.spec	54
4.3.4. README	54
4.3.5. COPYING	54

4.3.6. Common Content per il brand	54
4.3.7. La sotto-cartella css	55
4.3.8. La sotto-cartella images	55
4.4. Creare il pacchetto di un brand	56
5. Usare i set	59
5.1. Set a sé stanti	59
5.2. Set distribuiti	60
6. Creare un sito web con Publican	63
6.1. Creare manualmente un sito web	64
6.1.1. Creare la struttura del sito web	64
6.1.2. Creare, installare ed aggiornare la home page	66
6.1.3. Creare, installare ed aggiornare pagine di prodotto e di versione	68
6.1.4. Installare, aggiornare e rimuovere documenti	69
6.2. Creare un sito web usando pacchetti RPM	70
6.2.1. Creare la struttura del sito web	70
6.2.2. Creare, installare ed aggiornare la home page	72
6.2.3. Creare, installare ed aggiornare pagine di prodotto e di versione	73
6.2.4. Installare, aggiornare e rimuovere documenti	73
6.2.5. Automatizzare la pubblicazione	74
7. Frequently Asked Questions	77
Domande ricorrenti	77
A. Elementi ed attributi non permessi	81
A.1. Elementi non permessi	81
A.2. Attributi non permessi	83
B. Sommario dei comandi	87
C. Parametri di publican.cfg	91
D. Contenuto del file dump di un sito	97
D.1. Ricavare gli URL dai file dump	99
E. File spec d'esempio per il pacchetto desktop-menu	101
F. Codici di lingua	103
G. Cronologia revisioni	109

Prefazione

1. Convenzioni del documento

Questo manuale utilizza numerose convenzioni per evidenziare parole e frasi, ponendo attenzione su informazioni specifiche.

Nelle edizioni PDF e cartacea questo manuale utilizza caratteri presenti nel set *Font Liberation*¹. Il set Font Liberation viene anche utilizzato nelle edizioni HTML se il set stesso è stato installato sul vostro sistema. In caso contrario, verranno mostrati caratteri alternativi ma equivalenti. Da notare: Red Hat Enterprise Linux 5 e versioni più recenti, includono per default il set Font Liberation.

1.1. Convenzioni tipografiche

Vengono utilizzate quattro convenzioni tipografiche per richiamare l'attenzione su parole e frasi specifiche. Queste convenzioni, e le circostanze alle quali vengono applicate, sono le seguenti.

Neretto monospazio

Usato per evidenziare l'input del sistema, incluso i comandi della shell, i nomi dei file ed i percorsi. Utilizzato anche per evidenziare tasti e combinazione di tasti. Per esempio:

Per visualizzare i contenuti del file `my_next_bestselling_novel` nella vostra directory di lavoro corrente, inserire il comando `cat my_next_bestselling_novel` al prompt della shell e premere **Invio** per eseguire il comando.

Quanto sopra riportato include il nome del file, un comando della shell ed un tasto, il tutto riportato in neretto monospazio e distinguibile grazie al contesto.

Le combinazioni di tasti possono essere distinte dai tasti tramite il trattino che collega ogni parte della combinazione. Per esempio:

Premere **Invio** per eseguire il comando.

Premere **Ctrl+Alt+F2** per smistarsi sul primo virtual terminal. Premere **Ctrl+Alt+F1** per ritornare alla sessione X-Windows.

Il primo paragrafo evidenzia il tasto specifico singolo da premere. Il secondo riporta due combinazioni di tasti, (ognuno dei quali è un set di tre tasti premuti contemporaneamente).

Se si discute del codice sorgente, i nomi della classe, i metodi, le funzioni i nomi della variabile ed i valori ritornati indicati all'interno di un paragrafo, essi verranno indicati come sopra, e cioè in **neretto monospazio**. Per esempio:

Le classi relative ad un file includono **filesystem** per file system, **file** per file, e **dir** per directory. Ogni classe possiede il proprio set associato di permessi.

Proportional Bold

Ciò denota le parole e le frasi incontrate su di un sistema, incluso i nomi delle applicazioni; il testo delle caselle di dialogo; i pulsanti etichettati; le caselle e le etichette per pulsanti di selezione, titoli del menu e dei sottomenu. Per esempio:

¹ <https://fedorahosted.org/liberation-fonts/>

Selezionare **Sistema** → **Preferenze** → **Mouse** dalla barra del menu principale per lanciare **Preferenze del Mouse**. Nella scheda **Pulsanti**, fate clic sulla casella di dialogo **mouse per mancini**, e successivamente fate clic su **Chiudi** per cambiare il pulsante primario del mouse da sinistra a destra (rendendo così il mouse idoneo per un utilizzo con la mano sinistra).

Per inserire un carattere speciale in un file **gedit**, selezionare **Applicazioni** → **Accessori** → **Mappa carattere** dalla barra menu principale. Successivamente, selezionare **Cerca** → **Trova...** dalla barra del menu **Mappa carattere**, inserire il nome del carattere nel campo **Cerca** e cliccare **Successivo**. Il carattere ricercato verrà evidenziato nella **Tabella caratteri**. Fare un doppio clic sul carattere evidenziato per posizionarlo nel campo **Testo da copiare**, e successivamente fare clic sul pulsante **Copia**. Ritornare ora al documento e selezionare **Modifica** → **Incolla** dalla barra del menu di **gedit**.

Il testo sopra riportato include i nomi delle applicazioni; nomi ed oggetti del menu per l'intero sistema; nomi del menu specifici alle applicazioni; e pulsanti e testo trovati all'interno di una interfaccia GUI, tutti presentati in neretto proporzionale e distinguibili dal contesto.

Corsivo neretto monospazio o ***Corsivo neretto proporzionale***

Sia se si tratta di neretto monospazio o neretto proporzionale, l'aggiunta del carattere corsivo indica un testo variabile o sostituibile. Il carattere corsivo denota un testo che non viene inserito letteralmente, o visualizzato che varia a seconda delle circostanze. Per esempio:

Per collegarsi ad una macchina remota utilizzando ssh, digitare **ssh** **username@domain.name** al prompt della shell. Se la macchina remota è **example.com** ed il nome utente sulla macchina interessata è john, digitare **ssh** **john@example.com**.

Il comando **mount -o remount file-system** rimonta il file system indicato. Per esempio, per rimontare il file system **/home**, il comando è **mount -o remount /home**.

Per visualizzare la versione di un pacchetto attualmente installato, utilizzare il comando **rpm -q package**. Esso ritornerà il seguente risultato: **package-version-release**.

Da notare la parola in Corsivo neretto — nome utente, domain.name, file-system, pacchetto, versione e release. Ogni parola racchiude il testo da voi inserito durante l'emissione di un comando o per il testo mostrato dal sistema.

Oltre all'utilizzo normale per la presentazione di un titolo, il carattere Corsivo denota il primo utilizzo di un termine nuovo ed importante. Per esempio:

Publican è un sistema di pubblicazione per *DocBook*.

1.2. Convenzioni del documento

Gli elenchi originati dal codice sorgente e l'output del terminale vengono evidenziati rispetto al testo circostante.

L'output inviato ad un terminale è impostato su **tondo monospazio** e così presentato:

```
books      Desktop  documentation  drafts  mss      photos  stuff  svn
```

books_tests Desktop1 downloads images notes scripts svgs

Gli elenchi del codice sorgente sono impostati in **tondo monospazio** ma vengono presentati ed evidenziati nel modo seguente:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
    public static void main(String args[])
        throws Exception
    {
        InitialContext iniCtx = new InitialContext();
        Object          ref    = iniCtx.lookup("EchoBean");
        EchoHome        home   = (EchoHome) ref;
        Echo             echo   = home.create();

        System.out.println("Created Echo");

        System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
    }
}
```

1.3. Note ed avvertimenti

E per finire, tre stili vengono usati per richiamare l'attenzione su informazioni che in caso contrario potrebbero essere ignorate.



Nota Bene

Una nota è un suggerimento o un approccio alternativo per il compito da svolgere. Non dovrebbe verificarsi alcuna conseguenza negativa se la nota viene ignorata, ma al tempo stesso potreste non usufruire di qualche trucco in grado di facilitarvi il compito.



Importante

Le caselle 'importante' riportano informazioni che potrebbero passare facilmente inosservate: modifiche alla configurazione applicabili solo alla sessione corrente, o servizi i quali necessitano di un riavvio prima di applicare un aggiornamento. Ignorare queste caselle non causa alcuna perdita di dati ma potrebbe causare irritazione e frustrazione da parte dell'utente.



Avvertenza

Un Avvertimento non dovrebbe essere ignorato. Se ignorato, potrebbe verificarsi una perdita di dati.

2. Occorrono commenti!

Se in questo manuale si trovano errori tipografici, oppure se si hanno suggerimenti su come migliorare questo manuale, noi saremmo lieti di ascoltarli! Si prega di inviare un report in [Bugzilla](#)², per il componente **Publican**. Inviando il bug report, assicurarsi di menzionare l'identificatore del manuale: *publican*.

Se si hanno suggerimenti su come migliorare il documento, si cerchi di essere il più possibile specifici nella descrizione. Se si trova un errore, si prega di indicare la sezione e di includere anche parte del testo circostante, in modo da poterlo intercettare facilmente.



² https://bugzilla.redhat.com/bugzilla/enter_bug.cgi?product=Fedora&version=rawhide&component=publican

Introduzione

Publican

è uno strumento per pubblicare materiale scritto in DocBook XML. Questa guida spiega come creare e compilare libri ed articoli usando **Publican**. Non si tratta di un tutorial su DocBook XML; per supporto su DocBook XML, fare invece riferimento alla Guida *DocBook: The Definitive Guide* di Norman Walsh e Leonard Mueller, disponibile su <http://www.docbook.org/tdg/en/html/docbook.html>.

Publican è nato come strumento interno al Red Hat's Documentation Group (ora noto come Engineering Content Services). All'occorrenza questa eredità verrà evidenziata.

Progetto

Publican è un sistema di pubblicazione, non solo uno strumento di elaborazione di DocBook. Oltre ad assicurare la validità di un DocBook XML, **Publican** assicura che ogni file XML sia conforme allo standard di pubblicazione.

Le funzionalità di brand consentono di creare regole di presentazione e look personalizzati, in alternativa allo stile predefinito, per soddisfare le proprie esigenze editoriali. Le scelte effettuate nel codice, tuttavia, non sono modificabili.

Per esempio, le entità possono essere validamente definite in ogni file XML. Tuttavia per garantire che la dichiarazione DTD sia presente, valida e standardizzata, **Publican** riscrive la dichiarazione in ogni file XML prima di compilare un testo o articolo. Di conseguenza, tutte le entità dichiarate nei file XML vengono perse. Quindi **Publican** richiede di definire le entità nel file **Nome_Doc.ent** (vedere la [Sezione 3.1.6, «Nome_Doc.ent»](#)).

Al crescere del lavoro editoriale, la definizione di entità senza restrizioni porta alla duplicazione di entità e ad altre pratiche che causano difficoltà di mantenimento. Consolidare la definizione delle entità in un unico posto noto, serve ad alleviare i problemi di mantenimento e contribuisce ad irrobustire l'automazione del processo di compilazione.

Inoltre le entità presentano un ostacolo sostanzialmente insormontabile sulla qualità della traduzione (fare riferimento alla [Sezione 3.1.6.1, «Entità e traduzione»](#)). Quindi, si ritiene opportuno mantenere le attuali funzionalità del file **Nome_Doc.ent** senza aggiungere altre funzionalità o caratteristiche associate all'uso delle entità.

DRAFT

Installare Publican

1.1. Sistemi Operativi Linux



Importante — Disponibilità nei repository

Le procedure indicate in questa sezione assumono che **Publican** e le sue varie dipendenze siano disponibili nei repository cui ha accesso il proprio sistema.

1.1.1. Fedora

1. Aprire un terminale
2. Divenire l'utente root: `su -`
3. Eseguire il seguente comando per installare il pacchetto *publican* ed il pacchetto *publican-doc* della documentazione:

```
yum install publican publican-doc
```

Con **Publican** sono disponibili anche diversi pacchetti di brand. Eseguire il seguente comando come utente root, per installare i pacchetti per la costruzione di libri brand:

```
yum install publican-brand
```

Sostituire *brand* con **redhat**, **fedora**, **jboss**, **ovirt**, o **gimp**. Vedere il [Capitolo 4, Branding](#) per una spiegazione del branding in **Publican**.

1.1.2. Red Hat Enterprise Linux 5



Importante — Software non supportato

Publican non fa parte della distribuzione Red Hat Enterprise Linux. Quindi Red Hat non offre supporto per **Publican**.



Importante — Dipendenze disponibili soltanto all'interno di Red Hat

L'installazione di **Publican** su Red Hat Enterprise Linux 5, richiede un numero di dipendenze che attualmente sono disponibili solo nei repository yum interni a Red Hat.

1. Aprire un terminale
2. Divenire l'utente root: `su -`

3. Eseguire il seguente comando per installare il pacchetto *publican* ed il pacchetto *publican-doc* della documentazione:

```
yum install publican publican-doc
```

Con **Publican** sono disponibili anche diversi pacchetti di brand. Eseguire il seguente comando come utente root, per installare i pacchetti per la costruzione di libri brand:

```
yum install publican-brand
```

Sostituire *brand* con **redhat**, **fedora**, **jboss**, **ovirt**, o **gimp**. Vedere il [Capitolo 4, Branding](#) per una spiegazione del branding in **Publican**.

1.1.3. Ubuntu



Importante — Novità in 10.4 "Lucid Lynx"

Publican fa il suo ingresso in Ubuntu 10.4 "Lucid Lynx".

1. Aprire un terminale
2. Eseguire il seguente comando per installare il pacchetto *publican*:

```
sudo apt-get install publican
```

1.1.4. Debian



Avviso — Completare questa procedura

Completare ogni passaggio di questa procedura. Se non si effettuano, come qui indicate, le operazioni di ripristino alle modifiche apportate al file `/etc/apt/sources.list`, il sistema potrebbe diventare instabile.

Publican non è disponibile nella corrente versione stabile di Debian (version 5.0, "Lenny"), ma è disponibile nella corrente versione di test ("Squeeze"). Per installare **Publican** su un computer che esegue Debian, occorre abilitare temporaneamente l'accesso al repository **squeeze**. Abilitando l'accesso a questo repository, si autorizza il computer ad installare nuovo software e nuove versioni di software, rispetto alla corrente versione stabile di Debian. Tuttavia, non tutto il software disponibile nel repository di test ha già superato il controllo di qualità. Quindi, se dopo l'installazione di **Publican** non si disabilita l'accesso a questo repository, al successivo aggiornamento del sistema, i pacchetti software esistenti verranno sostituiti con nuove versioni scaricate dal repository, e molto probabilmente risulteranno essere versioni non ancora testate.

1. Aprire un terminale
2. Con un editor di testo aprire il file `/etc/apt/sources.list`. Per esempio, per modificare il file con **gedit**, eseguire:

```
sudo gedit /etc/apt/sources.list
```

3. Aggiungere la seguente riga alla fine del file:

```
deb http://ftp.debian.org/debian/ squeeze main
```

4. Salvare il file e chiudere l'editor.
5. Eseguire il seguente comando per aggiornare la lista dei pacchetti disponibili nel sistema:

```
sudo apt-get update
```

6. Eseguire il seguente comando per installare il pacchetto *publican*:

```
sudo apt-get install publican
```

7. Riaprire il file `/etc/apt/sources.list` e rimuovere la riga precedentemente inserita.

Notare che finchè la release "Squeeze" non viene rilasciata come versione stabile di Debian, occorre manualmente abilitare e disabilitare l'accesso al repository di test come fin qui descritto, ogniqualvolta diventa disponibile, nel repository di test, una nuova versione di **Publican**. Per informazioni aggiornate sullo status di **Publican** per Debian, visitare <http://packages.debian.org/squeeze/publican>, contenente il numero di versione di **Publican** disponibile nel repository (versione 2.1 al tempo di scrittura di questo documento).

Nel momento in cui "Squeeze" diventa la versione stabile di Debian, per installare **Publican** su sistemi che eseguono questa versione del sistema operativo, non occorre più abilitare o disabilitare l'accesso a repository extra.

1.2. Sistemi Operativi Windows

1. Scaricare il programma di installazione di Publican da <https://fedorahosted.org/releases/p/u/publican/>.
2. Spostarsi nella cartella in cui si è scaricato l'eseguibile **Publican-Installer-versione.exe**.
3. Fare doppio click sul file **Publican-Installer-versione.exe**.
4. Il programma di installazione inizia presentando una serie di accordi di licenza. Tutti i file che fanno parte di una installazione di **Publican** sono disponibili sotto licenza libera. Tuttavia, poichè diverse licenze sono più adatte a certe parti di **Publican** di altre, i file di **Publican** non vengono tutti distribuiti sotto la stessa licenza libera. Ogni licenza conferisce differenti diritti e responsabilità sulla copia e sulla modifica dei file di una installazione di **Publican**. Noi usiamo questa combinazione di licenze per consentire di usare **Publican** il più liberamente possibile e per consentire di scegliere qualunque licenza per i propri documenti, pubblicati usando **Publican**.

Leggere le condizioni dei vari accordi di licenza. Se si concorda con le condizioni, premere **I Agree** su ciascuna di esse, altrimenti premere **Cancel**.

5. Il programma di installazione propone di installare diversi componenti: **Publican** (denominato **Main** nella finestra d'installazione), alcuni *brand* (**RedHat**, **JBoss**, e **fedora**), e due componenti

di DocBook (il *DTD* o Data Type Definition, e l'*XSL* o Extensible Stylesheet Language). I tre brand si trovano raggruppati sotto il controllo espandibile **Brands**, mentre i componenti di DocBook si trovano nel controllo espandibile **DocBook** nella finestra di installazione. Vedere il [Capitolo 4, Branding](#) per una spiegazione dei brand in **Publican**. Per rendere i documenti XML in presentazioni di altri formati (come HTML e PDF), **Publican** usa il DTD e gli *stylesheet* di XSL. Se non si installano questi componenti, **Publican** deve scaricare questi dati da Internet ogniqualvolta elabora un documento, il che comporta lunghi ritardi.

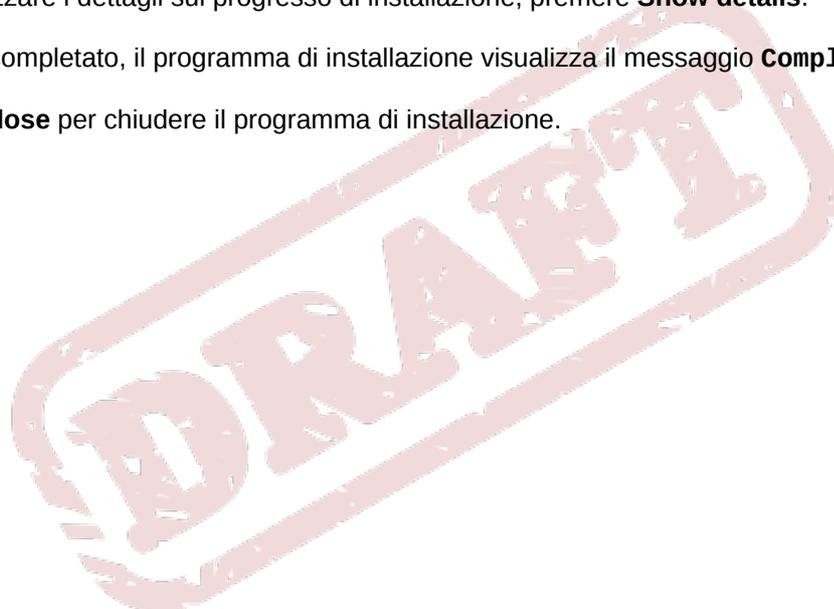
Tutti i componenti sono selezionati per impostazione. Deselezionare i componenti eventualmente *non* richiesti e poi premere **Next**, per continuare.

6. Per impostazione, il programma di installazione crea una cartella denominata **Publican** nella cartella **%ProgramFiles%** del proprio computer — tipicamente **C:\Program Files \Publican**. Per selezionare una cartella differente, inserire manualmente il percorso alla cartella, nella casella di testo etichettata **Destination Folder**.
7. Dopo aver impostato la cartella di destinazione, premere **Install**.

A questo punto, durante l'installazione di **Publican**, viene visualizzata una barra di progressione. Per visualizzare i dettagli sul progresso di installazione, premere **Show details**.

8. Una volta completato, il programma di installazione visualizza il messaggio **Completed**.

Premere **Close** per chiudere il programma di installazione.



Comandi di Publican

Publican è uno strumento da riga di comando. Per usare **Publican** su un computer con Sistema Operativo Linux, occorre avviare un emulatore di terminale (come **GNOME Terminal** o **Konsole**) oppure passare ad una console virtuale. Per usare **Publican** su un computer con sistema operativo Windows, avviare una shell a riga di comando, digitando **cmd** nello **Start menu**

I comandi di **Publican** hanno uno dei seguenti formati:

publican *opzione_comando*

opzione_comando è una delle opzioni accettate dal comando **publican**.

publican *azione* *opzioni_azione*

azione è una richiesta di elaborazione per **Publican**, come creare i file XML per un nuovo documento o creare un documento in HTML dai corrispondenti file XML. Le *opzioni_azione* si applicano ad una *azione*, per specificare per esempio la lingua di un documento.

publican *opzione_comando* *azione* *opzioni_azione*

Alcune *opzioni_comando* influenzano il risultato di una *azione*, come quando si richiede, per esempio, a **Publican** di usare nell'output la colorazione ANSI.

2.1. Opzioni di comando

Le opzioni per il comando **publican**, sono:

--help

Questa opzione visualizza i formati del comando ed un sommario delle azioni valide, descritte in questo capitolo.

--man

Questa opzione visualizza la pagina di man su **Publican** integrando le informazioni dell'opzione **--help**, oltre a fornire informazioni su licenze e dipendenze.

--help_actions

Questa opzione visualizza un elenco di *azioni* valide di **Publican**.

-v

Questa opzione visualizza il numero di versione di una installazione di **Publican**.

--config file

Questa opzione specifica per un documento, un file di configurazione alternativo al file predefinito **publican.cfg**.

--nocolours

Questa opzione disabilita la colorazione ANSI nei messaggi di log di **Publican**.

--quiet

Questa opzione disabilita tutti i messaggi di log.

2.2. Azioni

Publican è in grado di effettuare le seguenti elaborazioni:

build

trasforma i file XML in altri formati (per esempio: PDF, HTML su pagina singola o HTML su pagine multiple). Fare riferimento alla [Sezione 3.6, «Creare un documento»](#) per maggiori dettagli ed una descrizione delle opzioni disponibili.

clean

rimuove tutti i file e le cartelle nella sotto-cartella **tmp/**. La cartella **tmp/** viene creata durante l'esecuzione del comando **publican build**, usato per generare un documento, come **publican build --formats=html --langs=it-IT**.

clean_ids

modifica tutti gli ID in un formato standard. Questo formato è *Nome_Libro-titolo*. Per esempio, una sezione con titolo **First Section**, in un libro con titolo **Test Book** avrà il seguente ID dopo l'esecuzione del comando **clean_ids**: `<section id="Test_Book-First_Section">`.

**Importante — Possono verificarsi conflitti tra ID**

Il comando **publican clean_ids** serve a facilitare la costruzione di una struttura di DocBook in quei documenti provenienti da altri formati come HTML. Tuttavia, il comando **publican clean_ids** ricava le informazioni basandosi solo su file XML e sul nome del documento. Perciò, i nodi dello stesso tipo che hanno lo stesso identico titolo ricevono lo stesso ID. Questa duplicazione di ID inevitabilmente impedirà la compilazione del documento.

Usare il comando **publican clean_ids** per creare una struttura del documento, ed aggiustare manualmente eventuali duplicazioni di ID. Si raccomanda di non eseguire **publican clean_ids** su un documento già ben strutturato.

clean_set

rimuove le copie locali di libri remoti facenti parte di un set distribuito. Vedere la [Sezione 5.2, «Set distribuiti»](#) per i dettagli sull'uso di set distribuiti.

create

crea un nuovo libro, articolo o un nuovo set. Vedere il [Capitolo 3, Creare un documento](#) per i dettagli su come creare un libro o articolo, ed il [Capitolo 5, Usare i set](#) per i dettagli sull'uso dei set.

create_brand

crea un nuovo brand. Fare riferimento alla [Sezione 4.2, «Creare un brand»](#) per i dettagli.

create_site

crea un sito web di documentazione. Fare riferimento alla [Capitolo 6, Creare un sito web con Publican](#) per i dettagli.

help_config

visualizza un elenco di parametri di configurazione del file **publican.cfg**, contenuto in ciascun libro o brand. Fare riferimento alla [Sezione 3.1.1, «Il file publican.cfg»](#) per maggiori dettagli.

install_book

installa un documento su un sito web di documentazione. Vedere il [Capitolo 6, Creare un sito web con Publican](#) per i dettagli.

install_brand

configura l'installazione di un brand. Fare riferimento alla [Sezione 4.1, «Installare un brand»](#) per i dettagli.

lang_stats --lang=codice_lingua

genera un report statistico di traduzione per la lingua specificata da *codice_lingua*. Per ogni file PO generato da **Publican**, una tabella mostra il numero delle stringhe non tradotte in tutti i **msgid**; il numero delle stringhe fuzzy (contegge le stringhe contenute in **msgid**, il cui contenuto è variato dall'ultima generazione dei POT), ed il numero delle stringhe tradotte, coincidente, a traduzione avvenuta, con il numero delle stringhe contenute nel **msgid**.

old2new

crea un file **publican.cfg** a partire da un **Makefile** di un libro, articolo o set, originariamente creato con una versione sperimentale di **Publican** (versioni fino a **Publican 0.45**, inclusa). Fare riferimento alla [Sezione 3.6.2, «Compilare un documento creato con Publican 0»](#) per maggiori dettagli.

package

crea un pacchetto RPM di un libro, articolo, set o brand. Vedere la [Sezione 3.7, «Creare il pacchetto di un documento»](#) e la [Sezione 4.4, «Creare il pacchetto di un brand»](#) per maggiori dettagli.

print_banned

visualizza l'elenco dei tag di DocBook non raccomandati da **Publican**. Vedere l'[Appendice A, Elementi ed attributi non permessi](#) per una discussione sull'uso di tag non raccomandati.

print_known

visualizza l'elenco dei tag di DocBook supportati da **Publican**. I tag *supportati* sono quelli che hanno superato una minima verifica di qualità per l'uso in **Publican**. Fare riferimento all'[Appendice A, Elementi ed attributi non permessi](#).

print_tree

visualizza la struttura ad albero dei file XML inclusi, con il tag `<xi:include>`, in un libro, articolo o set.

print_unused

visualizza l'elenco dei file XML *non* inclusi, con il tag `<xi:include>`, in un libro, articolo o set.

remove_book

rimuove un documento da un sito web di documentazione. Vedere il [Capitolo 6, Creare un sito web con Publican](#) per i dettagli.

site_stats

genera un report statistico di un sito web di documentazione.

update_po

aggiorna i file *PO* (Portable Object). Vedere la [Sezione 3.5, «Preparare un documento per la traduzione»](#) per maggiori dettagli.

update_pot

aggiorna i file *POT* (Portable Object Template). Vedere la [Sezione 3.5, «Preparare un documento per la traduzione»](#) per maggiori dettagli.

update_site

aggiorna i template del sito web di documentazione. Consultare il [Capitolo 6, Creare un sito web con Publican](#) per i dettagli.

DRAFT

Creare un documento

Questo capitolo descrive come creare libri ed articoli: i file di configurazione principali, i file in un documento d'esempio, e come compilare un documento.

Usare il comando **publican create** per creare un nuovo documento corredato di tutti i file necessari.

Di seguito si riporta un elenco di opzioni che si possono appendere al comando **publican create**. Per esempio, **publican create --help**, **publican create --name New_Book**, e così via.

--help

visualizza l'elenco di tutte le opzioni del comando **publican create**.

--name Nome_Doc

assegna *Nome_Doc* al nome del libro o articolo. Questa variabile non deve contenere spazi. Per esempio, il comando **create_book --name Test_Book** crea un libro di nome **Test_Book** con tutti i file necessari per creare il libro, ed imposta il parametro *BOOKID* nel file **Test_Book.ent**.

--lang Codice_Lingua

imposta la lingua, con il *Codice_Lingua*, in cui il libro o articolo verrà redatto. Se non si specifica un codice linguistico, **Publican** per impostazione usa **en-US** (inglese americano). L'opzione **--lang** imposta il parametro *xml_lang* nel file di configurazione **publican.cfg**. Fare riferimento alla [Sezione 3.1.1, «Il file publican.cfg»](#) per maggiori informazioni sui parametri di **publican.cfg** ed all'[Appendice F, Codici di lingua](#) per i dettagli sui codici linguistici.

--version versione

imposta il numero di *versione* del prodotto descritto dal libro. Per esempio, per Red Hat Enterprise Linux 5.1 si userà **5.1**. Il valore predefinito è **0.1**. L'opzione **--version** imposta il tag `<productnumber>` nel file **Book_Info.xml** o nel file **Article_Info.xml**. Per maggiori informazioni vedere la [Sezione 3.1.2, «Book_Info.xml»](#).

--edition edizione

assegna il numero di *edizione* del libro. Questo numero serve ad indicare il rilascio di una nuova edizione di un libro. Il primo rilascio pubblico (*general availability* o GA) di un libro, dovrebbe avere l'edizione **1.0**. Il valore predefinito è **0**. L'opzione **--edition** imposta il tag `<edition>` nel file **Book_Info.xml** o nel file **Article_Info.xml**. Per maggiori informazioni fare riferimento alla [Sezione 3.1.2, «Book_Info.xml»](#).

--product Nome_Prodotto

assegna *Nome_Prodotto* al nome del prodotto descritto dal libro. Questa variabile non deve contenere spazi. Per esempio, usare il valore **Fedora** per la documentazione Fedora di base, ed il nome del prodotto per gli altri documenti, per esempio **Fedora_Directory_Server**. Il valore predefinito è **Documentation**. L'opzione **--product** imposta il tag `<productname>` nel file **Book_Info.xml** o **Article_Info.xml** e il parametro *PRODUCT* nel file **Doc_Name.ent**.

--type Article --name Nome_Articolo

crea un articolo invece di un libro, assegnando *Nome_Articolo* al nome dell'articolo. Questa variabile non deve contenere spazi. L'opzione **--type** imposta il parametro *type* nel file di configurazione **publican.cfg**. Fare riferimento alla [Sezione 3.1.1, «Il file publican.cfg»](#) per maggiori informazioni sui parametri del file **publican.cfg**.

--type Set --name Nome_Set

crea un set di documenti invece di un libro, assegnando *Nome_Set* al nome del set. Questa variabile non deve contenere spazi. L'opzione **--type** imposta il parametro *type* nel file di

configurazione **publican.cfg**. Fare riferimento alla [Sezione 3.1.1, «Il file *publican.cfg*»](#) per maggiori informazioni sui parametri del file **publican.cfg** ed al [Capitolo 5, Usare i set](#) per i dettagli sull'uso dei set.

--brand *brand*

assegna lo stile di presentazione o *brand*, per esempio **RedHat**, **fedora**, **JBoss**, **oVirt** o **GIMP** del documento. Il valore predefinito è **common**, il brand integrato in **Publican**. L'opzione **--brand** imposta il parametro *brand* nel file di configurazione **publican.cfg**. Fare riferimento alla [Sezione 3.1.1, «Il file *publican.cfg*»](#) per maggiori informazioni sui parametri del file **publican.cfg**. Questa opzione richiede che sia installato l'appropriato pacchetto di brand di **Publican**. Per esempio, per compilare libri con *brand* Red Hat, occorre installare il pacchetto *publican-redhat*. Fare riferimento alla [Sezione 4.1, «Installare un brand»](#) per istruzioni su come installare pacchetti di brand per l'uso in **Publican**. Vedere il [Capitolo 4, Branding](#) per maggiori informazioni.



Specificare opzioni

Nello specificare le opzioni indicate, è anche possibile inserire un segno "=" tra le opzioni d'azione ed i loro valori. Quindi i comandi **publican azione opzioni_azione valori_opzione** e **publican azione opzioni_azione=valori_opzioni**, sono del tutto equivalenti.

Prima di eseguire il comando **publican create**, spostarsi (con il comando shell **cd**), nella directory in cui si vuole venga creato il libro. Per esempio, per creare un libro di nome **Libro_di_Prova** nella directory **miei_libri/**, eseguire i seguenti comandi:

```
cd miei_libri/
publican create --name Libro_di_Prova --lang=it-IT
```

Per visualizzare i risultati di questo comando su un computer con Sistema Operativo Linux, eseguire il comando shell:

```
ls
```

Il risultato dovrebbe assomigliare a:

```
Libro_di_Prova/
```

Per visualizzare il contenuto della nuova cartella **Libro_di_Prova/** su un computer con Sistema Operativo Linux, eseguire i comandi:

```
cd Libro_di_Prova/
ls
```

Il risultato dovrebbe assomigliare a:

```
it-IT/ publican.cfg
```

3.1. I file nella directory del libro

Se si esegue il comando **publican create --name Libro_di_Prova --lang it-IT**, **Publican** crea una directory con i file richiesti, che generalmente sono:

- **publican.cfg**
- **it-IT** (una directory)
 - **Libro_di_Prova.xml**
 - **Libro_di_Prova.ent**
 - **Revision_History.xml**
 - **Preface.xml**
 - **Chapter.xml**
 - **Book_Info.xml**
 - **Author_Group.xml**
- **images** (una directory)
 - **icon.svg**

3.1.1. Il file publican.cfg



Nota — Personalizzare l'output

Se si mantengono diverse versioni di un documento, si può creare un file di configurazione per ogni versione. Quando si crea un documento o il pacchetto relativo, si può usare l'opzione **--config** per specificare un file di configurazione diverso dal file **publican.cfg**, e quindi usare un insieme differente di parametri per una particolare compilazione. Per esempio:

```
publican build --formats html,pdf --langs de-DE,en-US,it-IT --config community.cfg
```

Il file **publican.cfg** configura le opzioni di compilazione, e si trova nella cartella radice del libro. Di seguito si riporta un esempio di file **publican.cfg**, con una descrizione dei parametri ivi presenti:

```
# Config::Simple 4.59
# Mon Sep 28 16:38:14 2009

xml_lang: en-US
type: Book
brand: common
```

Parametri predefiniti

xml_lang

specifica la lingua dei file XML sorgenti, per esempio **en-US**, come impostato con l'opzione **--lang** nel comando **publican create**.

type

specifica il tipo di documento — un `<article>` DocBook, un `<book>` DocBook o un `<set>` DocBook, come impostato con l'opzione **--type** nel comando **publican create**.

brand

imposta il *brand* del documento, per esempio **RedHat**, **fedora**, **JBoss**, **oVirt** o **GIMP**, come impostato con l'opzione **--brand** nel comando **publican create**. Se non si specifica un brand, **Publican** usa il brand predefinito. Fare riferimento al [Capitolo 4, Branding](#) per maggiori informazioni.

Parametri avanzati

arch

filtra l'output in base all'*architettura* della macchina. Per esempio, impostando **arch: x86_64** nel file **publican.cfg**, l'applicazione **Publican** include solo gli elementi XML contenenti l'attributo equivalente, per esempio `<para arch="x86_64">`.

**Usare con cautela**

Come accade più in generale con i tag condizionali, il parametro **arch** può causare notevoli problemi in fase di traduzione di un documento. Vedere la [Sezione 3.8.1, «Tagging condizionale e traduzione»](#) per una spiegazione di questi problemi.

**parametro *arch* in nodi radice**

Se il nodo radice di un file XML viene escluso dal parametro *arch*, il documento non può essere compilato, poichè file vuoti non sono file XML validi. Per esempio, se il file **Installation_and_configuration-PPC.xml** è costituito da un solo capitolo:

```
<?xml version='1.0' encoding='utf-8' ?>
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN" "http://www.oasis-
open.org/docbook/xml/4.5/docbookx.dtd" [
]>
<chapter id="chap-Installation_and_configuration_on_PowerPC" arch="PowerPC">
<title>Installation and configuration on PowerPC</title>

[text of chapter]

</chapter>
```

ed il capitolo è incluso nel file **User_Guide.xml** con un tag `<xi:include>`, il documento non compilerà se viene impostata la **condition: x86** nel file **publican.cfg**.

Per escludere il capitolo, aggiungere il parametro *arch* al tag `<xi:include>` in **User_Guide.xml**, invece che al tag `<chapter>` in **Installation_and_configuration-PPC.xml**.



xrefs e parametro *arch*

Se un `<xref>` punta ad un contenuto escluso dalla compilazione dal parametro *arch*, la compilazione fallisce. Per esempio, impostando **arch: x86** nel file **publican.cfg**, il comando **publican build --formats=pdf --langs=en-US** fallisce se il libro ha il tag `<xref linkend="Itanium_installation">` che punta a `<section id="Itanium_installation" arch="IA64">`.

books

specifica un elenco di libri, separati da spazio, usati in un set remoto. Vedere la [Sezione 5.2, «Set distribuiti»](#) per maggiori informazioni sui set distribuiti.

brew_dist

specifica il target da usare per creare il pacchetto RPM desktop in **Brew**, il sistema di creazione di pacchetti interno a Red Hat. Il valore predefinito è **docs-5E**. Vedere la [Sezione 3.7.2, «Il comando publican package»](#) e la [Sezione 4.4, «Creare il pacchetto di un brand»](#) per maggiori informazioni sulla compilazione di pacchetti RPM.

bridgehead_in_toc

specifica se includere gli elementi DocBook `bridgehead` (o intestazioni svincolate) nelle tabelle dei contenuti. Per abilitare questa proprietà, impostare **bridgehead_in_toc: 1**. Per impostazione, quest'ultimo parametro è impostato a **0** e gli elementi `bridgehead` non sono inclusi nel sommario dei contenuti.

chunk_first

controlla se visualizzare la prima sezione in una nuova pagina, nel rendering HTML. Per visualizzare la sezione in una nuova pagina HTML, impostare il parametro su **chunk_first: 1**. Per impostazione, il valore predefinito è **0**, e la prima sezione viene visualizzata nella stessa pagina del proprio capitolo.

chunk_section_depth

controlla il livello di sotto-sezione a partire da cui queste non vengono riportate su una nuova pagina, nel rendering HTML. Per impostazione, il valore predefinito è **4**.



chunk_section_depth - Esempi di suddivisione di sotto-sezioni

Come si presentano alcune successioni di pagine HTML in un documento strutturato con capitoli e sotto-sezioni, per vari valori del parametro *chunk_section_depth*.

chunk_section_depth: 0

nessuna suddivisione di sezioni. Tutte le sezioni e sotto-sezioni appaiono nella stessa pagina del capitolo cui appartengono. La successione delle pagine è capitolo 1, capitolo 2, capitolo 3, ...

chunk_section_depth: 1

la suddivisione di sezione è a "livello 1". Ogni sezione di livello uno, con le relative sotto-sezioni, appaiono su una nuova pagina. La successione delle pagine è capitolo 1, 1.2, 1.3, 1.4 ... capitolo 2, 2.1, 2.2, ... 2.3 ...

chunk_section_depth: 2

la suddivisione di sezione è a "livello 2". La successione delle pagine è capitolo 1, 1.2, 1.2.2, 1.2.3, 1.2.4 ... 1.3, 1.3.2, 1.3.3 ...

chunk_section_depth: 3

la suddivisione di sezione è a "livello 3". La successione delle pagine è capitolo 1, 1.2, 1.2.2, 1.2.2.2, 1.2.2.3, 1.2.2.4 ... 1.3, 1.3.2, 1.3.2.2, 1.3.2.3 ...

chunk_section_depth: 4 (predefinito)

la suddivisione di sezione è a "livello 4". La successione delle pagine è capitolo 1, 1.2, 1.2.2, 1.2.2.2, 1.2.2.2.2, 1.2.2.2.3, 1.2.2.2.4 ... 1.2.3, 1.2.3.2, 1.2.3.2.2, 1.2.3.2.3 ...

classpath

imposta il percorso ai file *jar* (Java archive) per **FOP** (Formatting Objects Processor). **Publican** si basa su Apache **FOP** — una applicazione Java — per rendere i documenti in file PDF. Il percorso predefinito ai file *jar* di **FOP**, su un computer con Sistema Operativo Linux è: **/usr/share/java/ant/ant-trax-1.7.0.jar:/usr/share/java/xmlgraphics-commons.jar:/usr/share/java/batik-all.jar:/usr/share/java/xml-commons-apis.jar:/usr/share/java/xml-commons-apis-ext.jar**.

common_config

imposta il percorso ai file d'installazione di **Publican**. La locazione predefinita su un Sistema Operativo Linux è **/usr/share/publican**. Su un computer con sistema operativo Windows, la locazione predefinita è **%SystemDrive%\%ProgramFiles%\publican** — solitamente **C:/Program Files/publican**.

common_content

imposta il percorso alla cartella dei file *comuni* di **Publican**. I file contenuti forniscono formattazione predefinita, alcuni modelli e grafica generica. La locazione predefinita su un Sistema Operativo Linux è **/usr/share/publican/Common_Content**. Su un computer con sistema operativo Windows, la locazione predefinita è **%SystemDrive%\%ProgramFiles%\publican/Common_Content** — solitamente **C:/Program Files/publican/Common_Content**.

condition

specifica, prima di una trasformazione, le condizioni per escludere file XML. Vedere la [Sezione 3.8, «Tagging condizionale»](#) per maggiori informazioni.



Nodi root e tag condizionale

Se il nodo di root di un file XML viene escluso da un tag condizionale, il documento non compila, poiché file vuoti non sono file XML validi. Per esempio, se il file **Installation_and_configuration_on_Fedora.xml** contiene un solo capitolo:

```
<?xml version='1.0' encoding='utf-8' ?>
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN" "http://www.oasis-
open.org/docbook/xml/4.5/docbookx.dtd" [
]>
<chapter id="chap-Installation_and_configuration_on_Fedora" condition="Fedora">
<title>Installation and configuration on Fedora</title>

[text of chapter]

</chapter>
```

ed il capitolo è incluso in **User_Guide.xml** con un tag `<xi:include>`, il documento non compila se è presente l'impostazione **condition: Ubuntu** nel file **publican.cfg**.

Per escludere il capitolo, aggiungere un attributo condizionale al tag `<xi:include>` in **User_Guide.xml**, e non al tag `<chapter>` in **Installation_and_configuration_on_Fedora.xml**.



xref e tag condizionale

Se un `<xref>` punta ad un contenuto escluso nella compilazione da un tag condizionale, la compilazione fallisce. Per esempio, con l'impostazione **condition: upstream** nel file **publican.cfg**, il comando **publican build --formats=pdf --langs=en-US** fallisce se il libro ha un tag `<xref linkend="betasection">` che punta alla `<section id="betasection" condition="beta">`.

confidential

contrassegna un documento come confidenziale. Impostando su **1** questo parametro, **Publican** aggiunge il testo specificato nel parametro *confidential_text* (per impostazione, **CONFIDENTIAL**) a piè di pagina o in testa ad ogni pagina di un documento HTML o PDF, rispettivamente. Il valore predefinito è **0** (nessuna intestazione o piè di pagina).

confidential_text

specifica il testo da usare quando il parametro *confidential* è impostato ad **1**. Il testo predefinito è **CONFIDENTIAL**.

cvs_branch

il branch (ramo) in CVS in cui importare l'SRPM. Specificare questo parametro quando si crea il pacchetto di un documento con l'opzione **--cvs** — fare riferimento alla [Sezione 3.7.2, «Il comando publican package»](#).

cvs_pck

il pacchetto in CVS in cui importare l'SRPM. Specificare questo parametro quando si crea il pacchetto di un documento con l'opzione **--cvs** — fare riferimento alla [Sezione 3.7.2, «Il comando publican package»](#).

*cv*s_root

la radice in CVS in cui importare l'SRPM. Specificare questo parametro quando si crea il pacchetto di un documento con l'opzione `--cv`s — fare riferimento alla [Sezione 3.7.2, «Il comando *publican* package](#)».

debug

controlla se visualizzare il messaggi di debug durante l'elaborazione. Con il valore predefinito impostato a **0**, **Publican** non visualizza messaggi. Modificare il valore ad **1** per vedere i messaggi di debug.

def_lang

imposta la lingua predefinita per un sito web gestito da **Publican**. La tabelle dei contenuti delle altre lingue fanno riferimento ai documenti della lingua predefinita, se non sono disponibili traduzioni. Fare riferimento alla [Sezione 3.7, «Creare il pacchetto di un documento](#)» per maggiori informazioni.

doc_url

fornisce un URL al team di documentazione del pacchetto. In documenti HTML, **Publican** crea un link a questo URL in alto a destra di ogni pagina, attraverso l'immagine `image_right.png` nella cartella `Common_Content/images` del brand. Il valore predefinito è `https://fedorahosted.org/publican`.

docname

specifica il nome del documento. Se impostato, questo parametro non tiene conto del contenuto del tag `<title>` nel file `Book_Info.xml` in fase di costruzione del pacchetto del documento. Questo valore può contenere solo lettere maiuscole/minuscole non accentate, cifre, il carattere trattino_basso ed il carattere spazio ('a-z', 'A-Z', '0-9', e '_' e ' ').

dt_obsoletes

il pacchetto reso obsoleto dal pacchetto desktop.

dt_requires

il pacchetto richiesto dal pacchetto desktop, per esempio, il pacchetto del menu di una documentazione. Fare riferimenro alla [Sezione 3.7.1.3, «Voci nel menu del desktop per i documenti](#)».

*dt*dver

specifica la versione del DTD (Document Type Definition) di DocBook XML su cui si basa il progetto. **Publican** fa riferimento alla versione 4.5. Le specifiche della versione DTD 4.5 di DocBook XML sono disponibili su <http://www.docbook.org/specs/docbook-4.5-spec.html>.

**Un DTD differente potrebbe rallentare la compilazione**

Quando si installa **Publican**, si installa anche una copia locale della definizione DTD versione 4.5 di DocBook XML in accompagnamento ad XSL (Extensible Stylesheet Language). Se si imposta una versione di DTD per cui non risulta disponibile una versione locale, **Publican** deve scaricare DTD ed XSL appropriati da una sorgente in rete, ad ogni compilazione di un documento. In tal caso la compilazione del documento risulta ritardata dal completamento di questo scaricamento. La dimensione complessiva dei file è di circa 70 MB.

ec_id

imposta l'ID per un plugin d'aiuto di **Eclipse**. Ogni plugin deve possedere un unico ID che generalmente segue le convenzioni sui nomi dei pacchetti JAVA (<http://java.sun.com/docs/>

[codeconv/html/CodeConventions.doc8.html](#)). Per impostazione, **Publican** imposta l'ID con *org.prodotto.nomedoc*. L'ID impostato determina anche il nome della cartella del plugin, nella cartella **plugin**.

ec_name

imposta il nome per un plugin d'aiuto di **Eclipse**. E' un nome leggibile che compare nell'elenco d'aiuto di **Eclipse**. Il nome non deve essere unico o rispettare particolari convenzioni. Per impostazione, **Publican** imposta il nome con *prodotto.nomedoc*.

ec_provider

imposta il nome del fornitore per un plugin d'aiuto di **Eclipse**. Può essere un nome di persona, o il nome di un progetto o organizzazione. Questo è visibile agli utenti e non deve essere unico o rispettare particolari convenzioni. Per impostazione, **Publican** imposta il nome del fornitore con *Publican-version di Publican*.

edition

specifica il numero di edizione del documento. Se impostato, questo parametro non tiene conto del contenuto del tag `<edition>` nel file **Book_Info.xml** in fase di costruzione del pacchetto. Questo valore può contenere solo cifre ed il carattere punto ('0'-'9' e '.').

generate_section_toc_level

controlla se generare un sommario dei contenuti all'inizio delle sezioni ed il livello di profondità; con il valore predefinito **0**, non viene generato alcun sommario nelle sezioni, ma soltanto all'inizio del documento e nelle parti, nei capitoli e in appendice.



generate_section_toc_level - Esempi

`generate_section_toc_level: 0` (predefinito)

Publican genera le tabelle dei contenuti all'inizio del documento e nelle parti, nei capitoli e in appendice, ma non nelle sezioni.

`generate_section_toc_level: 1`

Publican genera le tabelle dei contenuti anche all'inizio delle sezioni di "livello 1", come le sezioni 1.1, 1.2 ... 2.1, 2.2 ...

`generate_section_toc_level: 2`

Publican genera le tabelle dei contenuti anche all'inizio delle sezioni di "livello 2", come le sezioni 1.1.1, 1.1.2, 1.1.3 ... 1.2.1., 1.2.2, 1.2.3 ...

ignored_translations

specifica le traduzioni da ignorare. Se si crea un libro o il pacchetto di un libro per una lingua filtrata da questo parametro, **Publican** ignora ogni traduzione in questa lingua, e crea invece il libro o il pacchetto relativo, nella lingua dei sorgenti XML. Fare riferimento alla [Sezione 3.5, «Preparare un documento per la traduzione»](#). I valori accettati dal parametro *ignored_translations* sono codici linguistici come it-IT, ca-ES, ecc. Per maggiori dettagli sui codici di lingua, fare riferimento all'[Appendice F, Codici di lingua](#).

license

specifica la licenza usata dal pacchetto. Per impostazione, **Publican** seleziona la licenza GFDL (GNU Free Documentation License). Fare riferimento alla [Sezione 3.7, «Creare il pacchetto di un documento»](#).

max_image_width

per le immagini in un documento, specifica la massima larghezza possibile, in pixel. Per impostazione, **Publican** ridimensiona le immagini più larghe di 444 pixel fino ad adattarle a questo limite. Il limite di 444 pixel assicura che le immagini non eccedano nel margine destro delle pagine HTML e si adattino all'interno delle pagine PDF. Fare riferimento alla [Sezione 3.2, «Aggiungere immagini»](#) per maggiori informazioni sull'uso delle immagini.

**Importante — 444 pixel è la massima larghezza di sicurezza**

Non usare il parametro *max_image_width* se le immagini contengono importanti informazioni. Le immagini più larghe di 444 pixel potrebbero presentarsi male nei documenti HTML e PDF e rendersi inusabili, in quanto superando i margini esse verrebbero rappresentate incomplete.

Viceversa, le immagini più larghe di 444 pixel che vengono scalate in un browser web o in un visualizzatore PDF, perdono in qualità.

Per preservare la qualità delle immagini, si raccomanda di tagliare o riscalare le immagini ad una larghezza inferiore a 444 pixel, prima di includerle in un documento.

menu_category

la categoria del menu del desktop (come definito dal file `.menu` corrispondente), in cui inserire il documento installato con un pacchetto RPM desktop. Fare riferimento alla [Sezione 3.7.1.3, «Voci nel menu del desktop per i documenti»](#).

os_ver

specifica il sistema operativo per cui costruire il pacchetto. **Publican** appende questo valore al nome del pacchetto RPM. Per esempio, `.fc15` for Fedora 15. Il valore predefinito è `.e15`, che significa Red Hat Enterprise Linux 5 e sistemi operativi derivati. Fare riferimento alla [Sezione 3.7, «Creare il pacchetto di un documento»](#) ed alla [Sezione 4.4, «Creare il pacchetto di un brand»](#).

prod_url

fornisce un URL per il prodotto a cui fa riferimento il documento. In documenti HTML, **Publican** crea un link a questo URL nella parte in alto a sinistra, usando l'immagine `image_left.png` nella cartella `Common_Content/images` del brand. Il valore predefinito è `https://fedorahosted.org/publican`.

product

specifica il prodotto cui fa riferimento il documento. Se impostato, questo parametro non tiene conto del contenuto del tag `<productname>` nel file `Book_Info.xml`, durante la creazione del pacchetto. Questo valore può contenere solo lettere maiuscole/minuscole non accentate, cifre, il carattere trattino-basso ed il carattere spazio ('a-z', 'A-Z', '0-9', e '_' e ' ').

release

specifica il numero di rilascio del pacchetto. Se impostato, questo parametro non tiene conto del contenuto del tag `<pubsnumber>` nel file `Book_Info.xml`, durante la creazione del pacchetto. Il valore può contenere solo cifre ('0-9').

repo

specifica il repository da cui prelevare i libri remoti che fanno parte di un set distribuito. Fare riferimento alla [Sezione 5.2, «Set distribuiti»](#).

scm

specifica il *sistema di controllo versione* (o source code management), usato nel repository contenente i libri remoti di un set distribuito. Al momento, **Publican** può usare solo **SVN** (Subversion), e quindi il valore predefinito è **SVN**. Fare riferimento alla [Sezione 5.2, «Set distribuiti»](#).

show_remarks

controlla se visualizzare gli elementi remark nel documento; il testo incluso è evidenziato con colore viola. Per impostazione, il parametro è impostato sul valore **0** che nasconde i remark. Impostare questo valore su **1** per visualizzare i remark.

show_unknown

controlla se segnalare tag sconosciuti durante la trasformazione dei file XML. Per impostazione, questo parametro ha il valore **1** segnalando i tag sconosciuti. Impostare questo valore su **0** per non visualizzare questi avvisi. **Publican** ignora questo parametro in *strict mode*.

src_url

specifica l'URL in cui trovare i tarball dei file sorgente. Questo parametro completa il campo **Source**: nell'intestazione del file *spec* dell'RPM. Fare riferimento alla [Sezione 3.7, «Creare il pacchetto di un documento»](#).

strict

imposta **Publican** in modalità *strict* per impedire l'uso di tag inutilizzabili in documenti professionali e traduzioni. Per impostazione, il parametro **strict** è impostato sul valore **0**, disabilitando la modalità *strict*. Per abilitare la modalità *strict*, impostare il valore su **1**. Correntemente la modalità *strict* non viene applicata.

tmp_dir

specifica la cartella dei prodotti di **Publican**. Per impostazione, il valore è **tmp** corrispondente ad una cartella di nome **tmp**, inclusa nella cartella contenente l'articolo o libro.

toc_section_depth

controlla fino a che livello rappresentare le sotto-sezioni nel sommario principale dei contenuti. Per impostazione, il valore predefinito è **2**.

**toc_section_depth - Esempi**

Nota che la prima cifra negli esempi, rappresenta un capitolo.

`toc_section_depth: 0`

Publican genera un sommario principale solo di capitoli.

`toc_section_depth: 1`

Publican genera un sommario principale solo per i capitoli e le sezioni di "livello 1", come 1, 1.1, 1.2, 1.3 ... 9, 9.1, 9.2 ... ma non per sezioni 1.1.1, 1.1.2 ...

`toc_section_depth: 2 (predefinito)`

Publican genera un sommario principale per i capitoli e le sezioni di "livello 1" e "livello 2", come 1, 1.1, 1.1.1, ... 1.2, 1.2.1, 1.2.2 ... ma non per le sezioni più interne tipo x.x.x.x .

version

specifica il numero di versione del prodotto a cui fa riferimento il documento. Se impostato, questo parametro non tiene conto del contenuto del tag `<productnumber>` nel file **Book_Info.xml**, per la creazione del pacchetto. Il valore può contenere solo cifre ed il carattere punto ('0'-'9' and '.').

web_brew_dist

specifica il target di compilazione **brew** da usare per la creazione di pacchetti RPM per il web. **Brew** è il sistema di creazione di pacchetti interno a Red Hat. Per impostazione, questo valore è impostato su **docs-5E**, rappresentando i pacchetti per la documentazione Red Hat Enterprise Linux 5. Fare riferimento alla [Sezione 3.7, «Creare il pacchetto di un documento»](#).

web_formats

una lista di formati, separati da virgola, da includere nel pacchetto RPM per il web. Fare riferimento alla [Sezione 3.7.2, «Il comando **publican package**»](#).

web_home

specifica che il documento è la home page di un sito web di documentazione, non un documento standard. Fare riferimento al [Capitolo 6, Creare un sito web con Publican](#).

**Importante — *web_home* è deprecato**

In **Publican 2.2**, **web_home** è stato sostituito da **web_type: home**. Supporto a **web_home** verrà interrotto in future versioni di **Publican**.

web_name_label

visualizza il valore impostato, invece del nome del libro, nel menu di un sito web gestito da **Publican**. Fare riferimento al [Capitolo 6, Creare un sito web con Publican](#).

web_obsoletes

specifica i pacchetti resi obsoleti da questo RPM per il web. Fare riferimento alla [Sezione 3.7, «Creare il pacchetto di un documento»](#).

web_product_label

visualizza il valore impostato, invece del nome del prodotto, nel menu di un sito web gestito da **Publican**. Fare riferimento al [Capitolo 6, Creare un sito web con Publican](#).

web_type

specifica che si tratta di un documento descrittivo per un sito web gestito da **Publican**, e non del documento di un prodotto. Il contenuto include la home page del sito web (**web_type: home**), pagine descrittive di prodotto (**web_type: product**), e pagine descrittive di versione (**web_type: version**). Fare riferimento al [Capitolo 6, Creare un sito web con Publican](#).

web_version_label

visualizza il valore impostato, invece del numero di versione, nel menu di un sito web gestito da **Publican**. Impostare il valore su **UNUSED** per una documentazione generale che non si applica ad una particolare versione di un prodotto. Fare riferimento al [Capitolo 6, Creare un sito web con Publican](#).



Aiuto da riga di comando

Eseguire il comando **publican help_config** nella cartella radice di un libro per un elenco di questi parametri.

3.1.2. Book_Info.xml



Article_Info.xml e Set_Info.xml

Questa descrizione del file **Book_Info.xml** si applica anche ai file **Article_Info.xml** e **Set_Info.xml**. Quindi, per semplificare, nel corso di questa sezione si farà riferimento al file **Book_Info.xml**.



Pacchetti non pacchetti RPM

Questa sezione descrive i pacchetti di documenti distribuiti con il **Gestore di pacchetti RPM**. Quando si usa il comando **publican package**, **Publican** genera un tarball che può essere usato per ricavare un pacchetto, da distribuire con un gestore di pacchetti software differente. Se si esegue **publican package** su un sistema senza **rpmbuild** installato, **Publican** genera ancora il tarball anche se non può creare da esso, il pacchetto RPM.

Il file **Book_Info.xml** contiene i metadati chiave di un libro: ID del libro; titolo; sottotitolo; autore e numero editoriale. Contiene anche nome e versione del prodotto documentato, ed un abstract.

Oltre a costituire gli elementi introduttivi di un libro, questi metadati sono usati anche per creare il pacchetto RPM di un libro. Solitamente, se si distribuisce un libro come un pacchetto RPM, i vari tag inclusi in maniera predefinita in **Book_Info.xml** devono contenere dati che siano conformi alle richieste del formato RPM. E' possibile non tenere conto di questi tag, usando i campi equivalenti nel file **publican.cfg**, come discusso in questa sezione.

A meno che non siano specificati nel file **publican.cfg**, per realizzare l'RPM di un libro, sono necessari i dati di sette tag predefiniti in **Book_Info.xml**. Per lo più, il nome di file del pacchetto RPM di un libro è costruito come:

nome_prodotto-titolo-numero_prodotto-codice_lingua-edizione-numero_pub.src.rpm

Ogni dato, escluso *codice_lingua*, è ricavato dal file **Book_Info.xml** — la *lingua* è specificata durante la creazione del libro. Come pure `<subtitle>` e `<abstract>` usati nel file *spec* dell'RPM per fornire il campo *Summary*: nell'intestazione ed il campo *%description*, rispettivamente.

Appresso, si riporta un esempio di file **Book_Info.xml**, per un **Libro di Prova**. Seguono i dettagli riguardanti questo file, e le richieste di conformità al formato RPM per ciascun tag.

```
<?xml version='1.0' encoding='utf-8' ?>
<!DOCTYPE bookinfo PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN" "http://www.oasis-open.org/docbook/xml/4.5/docbookx.dtd" [
<!ENTITY % BOOK_ENTITIES SYSTEM "Users_Guide.ent">
%BOOK_ENTITIES;
]>
<bookinfo id="book-Users_Guide-Users_Guide" lang="it-IT">
```

```

<title>Guida Utente</title>
  <subtitle>Pubblicare libri, articoli, relazioni e raccolte di volumi con DocBook XML</
subtitle>
  <productname>Publican</productname>
  <productnumber>2.6</productnumber>
  <edition>0</edition>
  <pubsnumber>0</pubsnumber>
  <abstract>
  <para>
    Questo manuale illustra come installare <application>Publican</application>. Inoltre
    fornisce istruzioni sull'uso di Publican per creare e pubblicare libri, articoli e raccolte
    di volumi basati su DocBook XML. Questa guida assume che si sia già familiari con DocBook
    XML.
  </para>

</abstract>
  <keywordset>
  <keyword>publican</keyword>
  <keyword>docbook</keyword>
  <keyword>publishing</keyword>

</keywordset>
  <subjectset scheme="libraryofcongress">
  <subject>
    <subjectterm>Electronic Publishing</subjectterm>

  </subject>
  <subject>
    <subjectterm>XML (Computer program language)</subjectterm>

  </subject>

</subjectset>
  <corpauthor>
  <inlinemediaobject>
  <imageobject>
    <imagedata fileref="Common_Content/images/title_logo.svg" />
  </imageobject>
  <textobject>
    <phrase>Team Publican</phrase>
  </textobject>

  </inlinemediaobject>

</corpauthor>
  <mediaobject role="cover">
  <imageobject remap="lrg" role="front-large">
    <imagedata fileref="images/cover_thumbnail.png" width="444" />
  </imageobject>
  <imageobject remap="s" role="front">
    <imagedata fileref="images/cover_thumbnail.png" width="444" />
  </imageobject>
  <imageobject remap="xs" role="front-small">
    <imagedata fileref="images/cover_thumbnail.png" width="444" />
  </imageobject>
  <imageobject remap="cs" role="thumbnail">
    <imagedata fileref="images/cover_thumbnail.png" width="444" />
  </imageobject>

</mediaobject>
  <xi:include href="Common_Content/Legal_Notice.xml" xmlns:xi="http://www.w3.org/2001/
XInclude" />
  <xi:include href="Author_Group.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
</bookinfo>

```

```
<bookinfo id="id_libro">,<articleinfo id="id_articolo">,<setinfo id="id_set">
```

L'ID del documento è usato internamente e non viene visualizzato ai lettori. Se si esegue il comando **publican clean_ids**, ogni ID inserito manualmente, inclusi questi, viene modificato in un formato *Nome_Doc-Titolo*, dove *Titolo* è il titolo associato al libro, articolo, sezione o capitolo.

```
<productname>nomeprodotto</productname>
```

Il nome del prodotto o gruppo di prodotto cui si riferisce il libro, articolo, o set, per esempio: **Red Hat Enterprise Linux** o **JBoss Enterprise Application Platform**. Quando si crea il pacchetto RPM di un libro, il valore nel tag `<productname>` viene usato come parte del nome di file dell'RPM.

Per non tenere conto di questo tag, usare il parametro *product* nel file **publican.cfg**, in particolare se il nome_prodotto contiene caratteri non-latini, caratteri accentati o caratteri di punteggiatura diversi dal trattino_basso.



Caratteri permessi

Publican usa i dati in questo tag per generare gli elementi del nome di file per i pacchetti RPM, a meno che non siano superati dai dati nel file **publican.cfg**. Se i dati nel tag non sono superati da quelli in **publican.cfg**, questo tag può contenere solo caratteri non accentati minuscoli e maiuscoli, cifre, il segno meno, il trattino-basso, il punto ed il carattere somma ('a-z', 'A-Z', '0-9', '-', '.', '_', e '+').

```
<title>titolo</title>
```

Abbastanza ovvio, il titolo del documento (per esempio *Server Configuration Guide*). Il titolo compare sotto il nome del prodotto in entrambe le presentazioni, HTML e PDF. Un titolo è necessario per la creazione del pacchetto RPM. Quando si crea l'RPM di un libro, il titolo è usato come parte integrante nel nome di file del pacchetto.

Per non tenere conto di questo tag, usare il parametro *docname* nel file **publican.cfg**, in particolare se il titolo del documento contiene caratteri non-latini, caratteri accentati o caratteri di punteggiatura diversi dal trattino_basso.



Caratteri permessi

Publican usa i dati in questo tag per generare gli elementi del nome di file per i pacchetti RPM, a meno che non siano superati dai dati nel file **publican.cfg**. Se i dati nel tag non sono superati da quelli in **publican.cfg**, questo tag può contenere solo caratteri non accentati minuscoli e maiuscoli, cifre, il segno meno, il trattino-basso, il punto ed il carattere somma ('a-z', 'A-Z', '0-9', '-', '.', '_', e '+').

```
<subtitle>sottotitolo</subtitle>
```

Analogamente ovvio come il precedente, il sottotitolo del libro; un titolo alternativo, generalmente esplicativo per il libro (per esempio: *Server Configuration Guide: Preparing Red Hat Enterprise Linux for Production Use*). Il sottotitolo compare sotto il titolo in entrambe le presentazioni, HTML e PDF. Un sottotitolo è necessario anche per la creazione del pacchetto RPM. In tal caso, il

sottotitolo è usato nel *Summary* del file *spec* dell'RPM. Ciò rende disponibile il **Summary** di un pacchetto RPM, con il comando `rpm -qi`.

`<productnumber>numero_prodotto</productnumber>`

Il numero di versione del prodotto cui si riferisce il documento, per esempio '5.2' per Red Hat Enterprise Linux 5.2 e '4.3' per JBoss EAP 4.3.

L'esecuzione del comando `publican create --name Nome_Doc --version versione` configura propriamente il numero di prodotto.

Per non tenere conto di questo tag, usare il parametro *version* nel file `publican.cfg`, in particolare se il termine versione di prodotto, non contiene solo cifre.



Caratteri permessi

Publican usa i dati in questo tag per generare parti del nome di file per i pacchetti RPM, a meno che non siano superati dai dati nel file `publican.cfg`. Se i dati nel tag non sono superati da quelli in `publican.cfg`, questo tag può contenere solo cifre ed il carattere punto ('0-9' e '.').

`<edition>edizione</edition>`

Il numero di edizione del libro. La prima edizione di un libro dovrebbe coincidere con 1.0 (a meno di usare 0.x per versioni pre-release di un libro). Le edizioni successive dovrebbero incrementare 1.x per indicare ai lettori una nuova edizione del libro. Questo tag imposta il numero di versione nel nome di file di un RPM, creato con `publican package`. Per esempio, impostando `edition` ad **1.2** e compilando il libro con il comando `publican package --binary --lang=en-US`, si crea un file RPM di nome `nomeprodotto-titolo-numeroprodotto-en-US-1.2-0.src.rpm`.

L'esecuzione del comando `publican create --name Nome_Doc --edition x.y` configura propriamente l'edizione.

Per non tenere conto di questo tag, usare il parametro *edition* nel file `publican.cfg`, in particolare se il termine edizione non contiene solo cifre.



Caratteri permessi

Publican usa i dati in questo tag per generare parti del nome di file per i pacchetti RPM, a meno che non siano superati dai dati nel file `publican.cfg`. Se i dati nel tag non sono superati da quelli in `publican.cfg`, questo tag può contenere solo cifre ed il carattere punto ('0-9' e '.').

`<pubsnumber>numero_pub</pubsnumber>`

Il `pubsnumber` imposta il numero di release (l'ultima cifra) nel nome di file di un RPM, creato con `publican package`. Per esempio, impostando il `pubsnumber` a **1** e compilando il libro con il comando `publican package --binary --lang=en-US`, si crea un file RPM di nome `nomeprodotto-titolo-numeroprodotto-en-US-edizione-1.src.rpm`.

Per non tenere conto di questo tag, usare il parametro *release* nel file `publican.cfg`, in particolare se il numero di release del documento non contiene solo cifre.



Caratteri permessi

Publican usa i dati in questo tag per generare gli elementi del nome di file per i pacchetti RPM, a meno che non siano superati dai dati nel file **publican.cfg**. Se i dati nel tag non sono superati da quelli in **publican.cfg**, questo tag può contenere solo cifre ('0–9').

`<abstract><para>abstract</para></abstract>`

Una breve descrizione e sintesi sull'argomento e sulla finalità del libro, generalmente non più lungo di un paragrafo. L'abstract compare prima del sommario dei contenuti nelle presentazioni HTML e nella seconda pagina nelle presentazioni PDF. Se si compila il pacchetto RPM per un libro, il tag `abstract` imposta il campo *Description* nel file *spec* dell'RPM. Ciò rende disponibile l'abstract con il comando `rpm -qi`.

Si possono aggiungere metadati extra al file **Book_Info.xml** di un documento, a supporto di specifiche proprietà nei vari formati d'uscita:

`<keywordset>`, `<keyword>`

I termini con il tag `<keyword>` contenuti in un `<keywordset>`, sono inseriti all'interno di tag `<meta name="keywords">`, presenti nel tag head dei file HTML e nel campo **Keywords** delle proprietà di un documento PDF.

`<subjectset>`, `<subject>`

I termini con il tag `<subject>` contenuti in un `<subjectset>` sono aggiunti al campo **Subject** delle proprietà di un documento PDF e nei metadati di un e-book in formato EPUB.

Si consideri di usare un *vocabolario controllato* per la definizione del soggetto di un documento, per esempio, il descrittore di soggetto dell'*LCSH* (Library of Congress Subject Headings). Si identifichi il vocabolario scelto con l'attributo *scheme* nel tag `<subjectset>`, per esempio `<subjectset scheme="libraryofcongress">`. In tal modo è possibile ricercare tra i descrittori di LCSH, nella pagina *Authorities & Vocabularies* di Library of Congress: <http://id.loc.gov/authorities/search/>.

`<mediaobject role="cover" id="epub_cover">`

Usare un tag `<mediaobject>` con attributi **role="cover"** e **id="epub_cover"** per impostare cover-art per e-book in formato EPUB. Per esempio:

```
<mediaobject role="cover" id="epub_cover">
  <imageobject role="front-large" remap="lrg">
    <imagedata width="600px" format="PNG" fileref="images/front_cover.png"/>
  </imageobject>
  <imageobject role="front" remap="s">
    <imagedata format="PNG" fileref="images/front_cover.png"/>
  </imageobject>
  <imageobject role="front-small" remap="xs">
    <imagedata format="PNG" fileref="images/front_cover.png"/>
  </imageobject>
  <imageobject role="thumbnail" remap="cs">
    <imagedata format="PNG" fileref="images/front_cover_thumbnail.png"/>
  </imageobject>
</mediaobject>
```

Come per le altre immagini in documenti, salvare le cover-art nella sotto-cartella **images**.

3.1.2.1. Pacchetti RPM, edizioni, ristampe e versioni

Come già notato in precedenza, il file predefinito **Book_Info.xml** usato da **Publican** include un tag `<edition>`. Se si distribuisce un libro come un pacchetto RPM, i dati all'interno di questo tag impostano, nel nome di file dell'RPM, le prime due cifre della versione. Quindi, una edizione '1.0' diventa una versione '1.0'. Il file **Book_Info.xml** include anche il tag `<pubsnumber>`. I dati all'interno di questo tag impostano la release del pacchetto RPM di un libro.

Quindi, per esempio, un libro con edizione 1.0 e con pubsubmer 5, coinciderebbe con la versione 1.0 e la release 5 (1.0-5), come **nomeprodotto-titolo-numeroprodotto-codice_lingua-1.0-5.src.rpm**.

I tag `edition` e `pubsnumber` non sono correlati al tag `<productnumber>`, anch'esso presente in **Book_Info.xml**: infatti `<productnumber>` specifica la versione del prodotto documentato o descritto. Del resto, è naturale avere la II edizione di un libro per una particolare versione di un prodotto.

In bibliografia, due copie di un libro fanno parte della stessa edizione se risultano stampati usando sostanzialmente la stessa composizione tipografica o di pagina. ('Sostanzialmente' sono tollerati solo correzioni tipografiche ed altre correzioni minori). Diversamente, i collezionisti di libri solitamente si riferiscono alla 'prima edizione' come alla 'prima uscita di stampa'; i bibliografi invece prestano attenzione al testo solitamente situato nelle prime pagine interne di un libro, in cui si sepcifica per esempio, 'II ristampa' o 'IV edizione'. Noi raccomandiamo di seguire il metodo seguito dai bibliografi.



Ripubblicazione: Incrementare `<pubsnumber>` and `<edition>`

Quando si usa **Publican** per ripubblicare un libro da un file XML *sostanzialmente identico*, incrementare il tag `<pubsnumber>`. Esso ha una funzione molto simile alla ristampa nella editoria tradizionale. Per il cambio di `<edition>`, si raccomanda di usare lo stesso criterio degli editori tradizionali: nel caso di revisione o di riscrittura *significativa*. Su cosa sia significativo e su quanto debba essere consistente una riscrittura, da richiedere un incremento intero o decimale nel numero di edizione, resta a discrezione dell'editore.

3.1.3. Author_Group.xml

Il file **Author_Group.xml** non è richiesto ed è la locazione standard in cui inserire autore, editore, grafico ed altri dettagli di merito. Il seguente è un esempio di file **Author_Group.xml**:

```
<?xml version='1.0'?>
<!DOCTYPE authorgroup PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN" "http://www.oasis-open.org/docbook/xml/4.5/docbookx.dtd" [
]>

<authorgroup>
  <corpauthor>FF0000 Headgear Documentation Group</corpauthor>
  <author>
    <firstname>Dude</firstname>
    <surname>McDude</surname>
    <affiliation>
      <orgname>My Org</orgname>
      <orgdiv>Best Div in the place</orgdiv>
    </affiliation>
    <email>dude.mcdude@myorg.org</email>
  </author>
</authorgroup>
```

Il file **Author_Group.xml** non necessariamente deve contenere tutte queste informazioni: inserire a discrezione quelle richieste.

3.1.4. Chapter.xml



Articoli e Capitoli

Gli articoli di DocBook non possono contenere capitoli. Se si usa l'opzione **--type=article** con il comando **publican create**, **Publican** non crea anche un file **Chapter.xml**. Usare le sezioni per organizzare il contenuto degli articoli.

Fare riferimento alla Guida *DocBook: The Definitive Guide* di Norman Walsh e Leonard Muellner, disponibile su <http://www.docbook.org/tdg/en/html/docbook.html>, per i dettagli sui vari modi di interazioni tra set, book, articoli, part, capitoli e sezioni. In particolare, notare che gli articoli possono essere documenti a sè stanti, o possono essere incorporati in libri.

Il file **Chapter.xml** è un modello per creare file di capitoli. Questi file costituiscono il contenuto di un libro. Di seguito si riporta un modello di capitolo (**Chapter.xml**) creato dal comando **publican create**. Notare che **DOCTYPE** è impostato a **chapter**:

```
<?xml version='1.0'?>
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN" "http://www.oasis-open.org/docbook/xml/4.5/docbookx.dtd" [
]>

<chapter id="MYBOOK-Test">
  <title>Test</title>
  <para>
    This is a test paragraph
  </para>
  <section id="MYBOOK-Test-Section_1_Test">
    <title>Section 1 Test</title>
    <para>
      Test of a section
    </para>
  </section>

  <section id="MYBOOK-Test-Section_2_Test">
    <title>Section 2 Test</title>
    <para>
      Test of a section
    </para>
  </section>

</chapter>
```

Il capitolo presenta due sezioni, **Section 1 Test** e **Section 2 Test**. Per ulteriori informazioni sui capitoli, fare riferimento a <http://docbook.org/tdg/en/html/chapter.html> della sopra citata guida.

**Nota**

Il file di capitolo dovrebbe essere rinominato in modo da rispecchiare l'argomento contenuto. Per esempio, un capitolo sull'installazione di un prodotto dovrebbe essere denominato **Installation.xml**, mentre un capitolo sull'impostazione di un software sarebbe meglio denominato, **Setup.xml** o **Configuration.xml**.

3.1.5. Nome_Doc.xml

Il file **Nome_Doc.xml** contiene le direttive **xi:include** per includere gli altri file XML indispensabili al documento, tra cui i capitoli e le sezioni contenute nei vari file XML. Per esempio, il file **Nome_Doc.xml** di un libro riunisce i capitoli contenuti in distinti file XML.

Ecco un esempio di **Nome_Doc.xml** che descrive un libro di DocBook — notare il parametro **DOCTYPE** impostato con **book**.

Esempio 3.1. Un libro DocBook

```
<?xml version='1.0'?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN" "http://www.oasis-open.org/docbook/xml/4.5/docbookx.dtd" [
]>

<book>
  <xi:include href="Book_Info.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
  <xi:include href="Preface.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
  <xi:include href="Chapter.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
  <xi:include href="Revision_History.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
  <index />
</book>
```

Questo esempio carica i file XML **Book_Info.xml**, **Preface.xml**, **Chapter.xml** e **Appendix.xml**.

**Importante**

L'ordinamento dei capitoli è importante. La creazione di questo libro, prevede che **Book_Info.xml** preceda **Preface.xml** che a sua volta preceda **Chapter.xml**, e così via.

Il file **Nome_Doc.xml** non si limita all'uso delle direttive **xi:include**. Si possono creare documenti anche con un solo file XML. Di seguito si riporta un esempio di libro, usando un singolo file XML:

```
<book>

<chapter>
<title>Chapter 1</title>
<para>
  A paragraph in Chapter 1.
</para>
<section id="section1">
<title>Chapter 1 Section 1</title>
<para>
  A paragraph in Section 1.
```

```

</para>
</section>
<section id="section2">
<title>Chapter 1 Section 2</title>
<para>
  A paragraph in Section 2.
</para>
</section>
</chapter>

<chapter>
<title>Chapter 2</title>
<para>
  A paragraph in Chapter 2.
</para>
</chapter>

</book>

```

Questo libro contiene due capitoli, in cui il I capitolo è costituito da due sezioni. Per ulteriori informazioni su sezioni e book vedere <http://docbook.org/tdg/en/html/section.html> e <http://docbook.org/tdg/en/html/book.html>, rispettivamente.

3.1.6. Nome_Doc.ent

Il file **Nome_Doc.ent** è usato per definire entità locali. Le entità *YEAR* e *HOLDER* sono usate per informazioni di copyright. Per impostazione, **Publican** imposta *YEAR* con l'anno corrente, ed inserisce un messaggio in *HOLDER* che richiama di specificare la licenza per il documento. Se mancano entrambe le entità *YEAR* e *HOLDER*, il documento non compila.

Altre entità potrebbero essere richieste dal *brand* applicato al documento. Per esempio, il brand per i documenti Fedora, usa l'entità **BOOKID** per indicare l'identificativo del documento ai lettori che desiderano inviare commenti.

```

<!ENTITY PRODUCT "MYPRODUCT">
<!ENTITY BOOKID "MYBOOK">
<!ENTITY YEAR "2008">
<!ENTITY HOLDER "YOUR NAME GOES HERE">

```

3.1.6.1. Entità e traduzione



Usare le entità con particolare attenzione

Le entità sono convenienti, ma dovrebbero essere usate con particolare attenzione in quei documenti che saranno tradotti. Scrivere per esempio, &FDS; invece di **Fedora Directory Server** è un vantaggio per il redattore del documento; tuttavia le entità non risultano trasformate nei file *PO* (Portable Object) usati dai traduttori. Di conseguenza, risulta impossibile una traduzione completa di un documento contenente entità.

Le entità rappresentano degli ostacoli per i traduttori, precludendo la possibilità di realizzare traduzioni di qualità. La natura propria di una entità è di rendere esattamente, in ogni occorrenza del documento ed in ogni lingua, la parola o frase rappresentata. Questa scarsa flessibilità comporta che la parola

o gruppo di parole, rappresentate dall'entità, possa essere illeggibile o incomprensibile in certe lingue e non possa modificarsi al cambiare delle regole grammaticali richieste dalla lingua. Inoltre, poichè durante la conversione in PO dei file XML, le entità non vengono trasformate, i traduttori non possono selezionare correttamente, secondo le regole grammaticali della propria lingua, le parole da inserire intorno all'entità.

Se si definisce una entità — `<!ENTITY LIFT "Liberty Installation and Formatting Tome">` — nel file XML si può inserire un riferimento all'entità definita, `&LIFT;` e in ogni compilazione HTML, PDF o semplice testo, si visualizzerà l'entità **Liberty Installation and Formatting Tome**.

Le entità non vengono trasformate durante la conversione in PO dei file XML. Quindi, i traduttori non vedono mai **Liberty Installation and Formatting Tome**, ma soltanto `&LIFT;` che non possono tradurre.

Si consideri per esempio la traduzione in tedesco, del seguente frammento in lingua inglese:

As noted in the Liberty Installation and Formatting Tome, Chapter 3...

Una traduzione potrebbe essere:

Wie in dem Wälzer für die Installation und Formatierung von Liberty, Kapitel 3, erwähnt...

Poichè non esistono entità, il titolo può essere tradotto in un tedesco corretto. Inoltre, notare in questo contesto linguistico, l'uso di 'dem', la forma corretta dell'articolo determinativo ('il') quando riferito a Wälzer ('tomo')

Per contrasto, usando le entità, la stessa frase nel file PO risulta:

```
#. Tag: para
#, no-c-format
msgid "As noted in the <citetitle>&LIFT;</citetitle>, Chapter 3..."
msgstr ""
```

La traduzione di ciò, probabilmente sarebbe:

```
#. Tag: para
#, no-c-format
msgid "As noted in the <citetitle>&LIFT;</citetitle>, Chapter 3..."
msgstr "Wie in <citetitle>&LIFT;</citetitle>, Kapitel 3, erwähnt..."
```

E nella presentazione si avrebbe:

Wie in *Liberty Installation and Formatting Tome*, Kapitel 3, erwähnt...

In tal caso, ovviamente il titolo rimane in inglese, incluse le parole come 'Tome' e 'Formatting' che il lettore difficilmente comprende. Inoltre, il traduttore è costretto ad omettere l'articolo determinativo 'dem', per un costrutto più generico che si avvicina ad un ibrido tra inglese e tedesco, definito Denglisch o Angleutsch, dai madrelingua tedesca. Molti di coloro che parlano il tedesco, ritengono scorretto questo approccio e quasi tutti un modo poco elegante.

Problemi analoghi emergono con un frammento come questo:

However, a careful reading of the *Liberty Installation and Formatting Tome* afterword shows that...

Senza testo nascosto da entità, una traduzione in tedesco potrebbe essere:

Jedoch ergibt ein sorgfältiges Lesen des Nachworts für den *Wälzer für die Installation und Formatierung von Liberty*, dass...

Se, per salvare tempo di scrittura, si fosse usata un'entità, il traduttore si sarebbe trovato con questo:

```
#. Tag: para
#, no-c-format
msgid "However, a careful reading of the <citetitle>&LIFT;</citetitle> afterword shows that..."
msgstr ""
```

E la traduzione sarebbe stata differente, in modo sottile ma rilevante:

```
#. Tag: para
#, no-c-format
msgid "However, a careful reading of the <citetitle>&LIFT;</citetitle> afterword shows that..."
msgstr "Jedoch ergibt ein sorgfältiges Lesen des Nachworts für <citetitle>&LIFT;</citetitle>, dass..."
```

Presentata al lettore, questa apparirebbe come:

Jedoch ergibt ein sorgfältiges Lesen des Nachworts für *Liberty Installation and Formatting Tome*, dass...

Di nuovo, notare l'assenza dell'articolo determinativo (den in questo contesto grammaticale). Ciò è poco elegante ma necessario, in quanto il traduttore può solo tentare di indovinare l'articolo (den, die o das), generando molto probabilmente un errore.

Infine, tener presente che anche se un termine è invariabile in inglese, ciò non necessariamente è vero in altre lingue, anche quando si tratta di un *nome proprio* come quello di un prodotto. In molte lingue, i nomi cambiano la loro forma (*flessione*) in accordo al ruolo nel periodo (*caso grammaticale*). Una entità in un file XML, impostata per rappresentare un nome inglese o un sintagma nominale (*noun phrase*) rende praticamente impossibile una traduzione corretta in tali lingue.

Per esempio, se si scrive un documento che si può applicare a più di un prodotto, si potrebbe essere tentati di impostare una entità come &PRODUCT; . Il vantaggio di questo approccio è che cambiando semplicemente questo valore nel file **Nome_Doc.ent**, si può facilmente adattare il libro a una documentazione, per esempio, per Red Hat Enterprise Linux, Fedora, o CentOS. Comunque, mentre il nome proprio *Fedora* non subisce variazioni nella lingua inglese, (per esempio) in ceco, presenta sei forme differenti, una per ciascuno dei possibili ruoli in un periodo:

Tabella 3.1. 'Fedora' in ceco

Caso	Uso	Forma
Nominativo	il soggetto di un periodo	Fedora
Genitivo	indica possesso	Fedory
Accusativo	l'oggetto diretto di un periodo	Fedoru
Dativo	l'oggetto indiretto di un periodo	Fedoře
Vocativo	rivolto direttamente al soggetto	Fedoro
Locativo	riguarda un argomento o luogo	Fedoře
Strumentale	riguarda un mezzo	Fedorou

Per esempio:

- Fedora je linuxová distribuce. — Fedora è una distribuzione Linux.
- Inštalácia Fedory — Installazione di Fedora.
- Stáhnout Fedoru — Ottieni Fedora.
- Přispějte Fedoře — Contribuire a Fedora.
- Ahoj, Fedoro! — Ciao Fedora!
- Ve Fedoře 10... — In Fedora 14...
- S Fedorou získáváte nejnovější... — Con Fedora, puoi ottenere gli ultimi...

Una frase che inizia con S Fedora získáváte nejnovější... rimane comprensibile al lettore ceco, ma risulta sintatticamente scorretta. Lo stesso effetto si può aver con la lingua inglese, infatti anche se i sostantivi hanno perso l'uso delle desinenze intorno al Medio Evo, i pronomi hanno conservato la flessione. La frase, 'Me see she' risulta completamente comprensibile ad un inglese, ma non è la forma che ci si aspetterebbe di leggere, poichè non è corretta la forma dei pronomi **me** e **she**. **Me** è la forma accusativa del pronome, ma poichè è il soggetto del periodo, il pronome dovrebbe assumere la forma nominativa, **I**. Analogamente, **she** è il caso nominativo, ma come oggetto diretto del periodo il pronome dovrebbe assumere la forma accusativa, **her**.

I sostantivi nella maggior parte delle lingue slave come russo, ceco, polacco, serbo e croato, hanno sette differenti casi. I sostantivi nelle lingue finno-ugriche come finlandese, ungherese ed estone hanno tra quindici e diciassette casi. Altre lingue modificano i sostantivi per altre ragioni. Per esempio, le lingue scandinave flettono i sostantivi per indicare *determinazione* — se il sostantivo si riferisce ad 'una cosa' o 'alla cosa' — ed alcuni dialetti di queste lingue flettono i sostantivi per determinazione e per declinazione.

Ora si estendano tali problemi a tutte le lingue (più di 40), attualmente supportate da **Publican**. Oltre alle poche stringhe non tradotte che **Publican** specifica per impostazione nel file **Nome_Doc.ent**, le entità possono risultare utili per specificare i numeri di versione dei prodotti. Al di là di ciò, l'uso delle entità sembra quasi un desiderio di inibire e ridurre la qualità delle traduzioni. Inoltre, il lettore del documento, tradotto in una lingua con inflessione di sostantivi (declinazione, determinazione o altro), non sa di certo che l'errore grammaticale è generato dalle entità impostate nell'XML — concludendo, con buona probabilità, che si tratta di incompetenze del traduttore.

3.1.7. Revision_History.xml

Durante la sua elaborazione, il comando **publican package** individua nella directory dei file XML, il primo file contenente un tag **<revhistory>**. Successivamente **publican** usa questo file per compilare la cronologia di revisione del pacchetto RPM.

3.2. Aggiungere immagini

Salvare le immagini nelle sottocartella **images** della cartella contenente i file XML. Usare **./images/nome_immagine** per inserire le immagini in un libro. Di seguito si riporta un esempio che inserisce l'immagine **testimage.png**:

```
<mediaobject>
<imageobject>
  <imagedata fileref="./images/testimage.png" />
</imageobject>
<textobject><phrase>alternate text goes here</phrase></textobject>
```

```
</mediaobject>
```

Assicurarsi di fornire un `<textobject>` in modo da rendere accessibile il contenuto alle persone con disabilità visiva. In alcuni Stati, potrebbe essere una responsabilità legislativa permettere questa accessibilità — per esempio negli Stati Uniti alle organizzazioni si richiede di rispettare la Section 508 del *Rehabilitation Act of 1973*.¹

Se il libro contiene immagini che occorre localizzare — per esempio, gli screenshot di una interfaccia utente in una altra lingua da quella originale del libro — salvare queste immagini nelle sottocartelle **images** delle directory linguistiche. Assicurarsi che il file dell'immagine tradotta abbia lo stesso nome del file della lingua originale. Quando si compila un libro per una lingua, **Publican** usa il file della sottocartella **images/** nella directory linguistica pertinente e non quello nella sottocartella **images/** della lingua originale.

Immagini grandi si presentano male in HTML perchè spesso invadono il margine destro del testo. Analogamente, le immagini più larghe di 444 pixel, spesso si estendono al di là del margine destro delle pagine PDF e risultano tagliate, lasciando visibile solo la parte sinistra dell'immagine. Per questo, per impostazione, **Publican** crea presentazioni HTML e PDF istruendo i browser web e i visualizzatori PDF a scalare le immagini più larghe di 444 pixel. Notare, comunque, che le immagini scalate in questo modo perdono significativamente in qualità. Per migliori risultati, scalare o tagliare le immagini in un software di editazione immagini in modo da garantire una larghezza non superiore a 444 pixel, prima di inserirle in un documento.

Per non tenere conto della limitazione di 444 px imposta da **Publican**, specificare la larghezza di un'immagine nel tag `<imagedata>`. Per esempio, per impostare una larghezza di 670 pixel:

```
<imagedata fileref=".images/image.png" width="670px">
```

In tal caso, rivedere accuratamente la presentazione per preservare gli standard di qualità richiesti.



Locazioni delle immagini

Publican usa soltanto le immagini nella sottocartella **images** della cartella contenente i file XML e le corrispondenti immagini nelle sottocartelle **images** pertinenti alle lingue. Immagini salvate in altre directory non vengono prese in considerazione.



File PNG in documenti PDF

Publican dipende da un'applicazione esterna, **FOP**, per rendere documenti PDF. Al momento, alcune versioni di **FOP** contengono un bug che altera i colori in certe immagini in formato PNG. Nello specifico, le immagini PNG a 32-bit vengono visualizzate correttamente, mentre quelle a 24-bit presentano problemi.

Se si nota che **Publican** produce un file PDF contenente immagini con colori alterati, convertire i file PNG originali nel formato PNG a 32-bit aggiungendo un *canale alpha* all'immagine e poi ricompilare il libro. Se il software di elaborazione immagine, non include un'opzione specifica denominata **Aggiungi canale alpha**, usare l'opzione **Aggiungi trasparenza**.

¹ Fare riferimento a <http://www.section508.gov/>

3.3. Aggiungere codice

Se il libro contiene pezzi di codice, salvare il file in una sotto-cartella denominata **extras/** nella cartella della lingua originale, ed usare un `<xi:include>` per caricare il file del codice nella struttura XML del documento. Ogni file contenuto nella cartella **extras/** non viene analizzato sintatticamente (parsed) come XML da **Publican**.

Alcuni caratteri sono riservati in XML, in particolare, **&** e **<**. Se si inserisce un pezzo di codice direttamente in un file XML di un documento, occorre fare l'*escaping* di questi caratteri, rendendoli **CDATA** o sostituendoli con entità (**&**; e **<**; rispettivamente).² Posizionando questi file nella cartella **extras/**, non occorre alcun escaping di questi caratteri. Questo approccio risparmia tempo, riduce la possibilità di introdurre errori nell'XML e nel codice; oltre a semplificare il mantenimento del documento e del codice.

Per includere nel documento, un pezzo di codice contenuto nella cartella **extras/**, seguire questa procedura:

1. Creare la cartella **extras**

```
mkdir en-US/extras
```

2. Copiare il file del codice nella cartella **extras**

```
cp ~/samples/foo.c en-US/extras/
```

3. Nel file XML, includere il file di codice in un tag `xi:include`

```
<programlisting>
<xi:include parse="text" href="extras/foo.c" xmlns:xi="http://www.w3.org/2001/
XInclude" />
</programlisting>
```

4. Ora si può modificare il file **en-US/extras/foo.c** nel proprio editor preferito, senza doversi preoccupare di eventuali effetti nell'XML.

Lo stesso approccio funziona annotando il codice con *callout*. Per esempio:

```
<programlistingco>
<areaspec>
  <area id="orbit-for-parameter" coords='4 75' />
  <area id="orbit-for-magnitude" coords='12 75' />
</areaspec>
<programlisting language="Fortran"><xi:include parse="text" href="extras/orbit.for"
xmlns:xi="http://www.w3.org/2001/XInclude" /></programlisting>
<calloutlist>
  <callout id="callout-for-parameter" arearefs="orbit-for-parameter">
    <para>
      The <firstterm>standard gravitational parameter</firstterm>
      ( $\mu$ ) is a derived value, the product of Newton's gravitational
      constant (G) and the mass of the primary body.
    </para>
  </callout>
  <callout id="callout-for-magnitude" arearefs="orbit-for-magnitude">
    <para>
```

² Fare riferimento alla sezione 2.4 "Character Data and Markup" dello standard XML 1.0, disponibile su <http://www.w3.org/TR/2008/REC-xml-20081126/>.

```

    Since the mass of the orbiting body is many orders of magnitude
    smaller than the mass of the primary body, the mass of the
    orbiting body is ignored in this calculation.
  </para>
</callout>
</calloutlist>
</programlistingco>

```

Notare gli elementi `<area>` che definiscono la posizione dei callout che compaiono nel codice. Gli attributi in **coords** specificano un numero di riga ed un numero di colonna, separati da uno spazio. In questo esempio, i callout sono applicati alle righe 4 e 12 del codice, entrambi allineati sulla colonna 75. Anche se questo approccio prevede di adattare gli attributi **coords** ad ogni modifica apportata al codice, ciò evita di combinare tag `<coords>` nel codice.

3.4. Aggiungere file

Publican permette di includere file arbitrari all'interno di un documento. Questi file vengono inclusi nel pacchetto RPM compilato da **Publican** e vengono installati sul sistema dell'utente insieme al documento. Per esempio, si possono includere file multimediali di tutorial a complemento del documento, o file di codice sorgente o altro materiale utile agli utenti per lavorare con gli esempi o i tutorial.

Per distribuire file arbitrari con un documento, includerli in una cartella denominata **files**, e salvarla nella cartella della lingua originale (p.e. **en-US**) del libro (p.e. **My_Book**).

Nella cartella **My_Book**:

```
mkdir en-US/files
```

Copiare i file nella cartella **files**:

```
cp arbitrary_files en-US/files
```

3.5. Preparare un documento per la traduzione

Supporto per la localizzazione di documenti è stata una considerazione chiave del progetto di **Publican**. Il workflow di traduzione generale dei documenti sviluppati in **Publican** procede come segue:

1. Completare l'XML di un documento.

Questa versione XML del documento dovrebbe essere considerata 'frozen'. Se il documento si trova in un repository con controllo di versione, a questo punto, la versione verrebbe spostata in una cartella separata o *branch*. In tal modo i redattori possono iniziare a lavorare su versioni successive del documento in un branch, e fornire una base stabile per traduzione in un altro branch.

2. Generare i file *POT* (Portable Object Template) dai file XML:

```
$ publican update_pot
```

Se è la prima volta che si creano i file POT per il documento, **Publican** crea una nuova sottocartella, denominata **pot**. La sottocartella **pot** contiene un file pot per ciascun file XML nel documento. Se **Publican** ha creato i file POT precedentemente, **Publican** aggiorna i file POT esistenti per rispecchiare i cambiamenti apportati ai file XML dall'ultimo aggiornamento dei file POT.



Rimuovere i file XML non usati

Publican genera un file POT per ogni file XML presente della cartella dei file XML, che siano usati o meno nel documento. Se si trasformano in POT file XML non usati, si spreca il tempo e lo sforzo dei volontari traduttori, e si spreca denaro per le traduzioni commissionate a pagamento.

Usare il comando **publican print_unused** per visualizzare l'elenco dei file XML non usati nel documento.

3. Generare dai file POT, i file *PO* (Portable Object) per poter avviare la traduzione in una particolare lingua.

```
$ publican update_po --langs=codice_lingua
```

dove *codice_lingua* è il codice per la lingua target. Fare riferimento all'[Appendice F, Codici di lingua](#) per maggiori informazioni su questi codici. Si possono fornire codici linguistici multipli, separati da virgole, in modo da generare i file PO per più lingue. Per esempio:

```
$ publican update_po --langs=hi-IN,it-IT,ru-RU,zh-CN
```

Se è la prima volta che i file PO vengono creati per una lingua, **Publican** crea una nuova sottocartella, il cui nome coincide con il codice linguistico specificato con l'opzione **--langs=**. La sottocartella contiene un file PO per ciascun file POT nella sottocartella **pot**. Se **Publican** ha creato i file PO precedentemente, **Publican** aggiorna i file PO esistenti per rispecchiare i cambiamenti apportati ai file POT dall'ultimo aggiornamento dei file PO. E' possibile aggiornare i file PO esistenti in tutte le sottocartelle, usando l'opzione **--langs=all**:

```
$ publican update_po --langs=all
```



Rimuovere i file POT non usati

Publican genera un file PO per ogni file POT presente della cartella **pot**, a prescindere se il file POT corrisponda o meno ad un file XML usato nel documento, o corrisponda ad un file XML esistente. Se si trasformano in file PO, i POT da file XML non usati od eliminati, si spreca il tempo e lo sforzo dei volontari traduttori, e si spreca denaro per le traduzioni commissionate a pagamento.

Quando si generano i file PO, **Publican** mostra un avviso per i file POT che non presentano un corrispondente file XML, tuttavia genera comunque il file PO. Comunque, **Publican** non avvisa se esiste un file POT corrispondente ad un file XML non usato nel documento.

4. Tradurre le *stringhe* contenute nei file PO.
5. Creare il documento per la lingua desiderata, per esempio:

```
$ publican build --formats=html,html-single,pdf --langs=it-IT,nb-NO
```

o il pacchetto per la lingua desiderata, per esempio:

```
$ publican package --lang=it-IT
```

E' possibile creare il documento in tutte le lingue di cui si dispone una traduzione, usando l'opzione `--langs=all`, mentre i pacchetti vanno compilati individualmente. Fare riferimento alla [Sezione 3.6, «Creare un documento»](#) ed alla [Sezione 3.7, «Creare il pacchetto di un documento»](#) per maggiori informazioni.



Importante — impostare *Project-Id-Version* per il pacchetto

Il numero di release dei pacchetti RPM per i documenti tradotti viene impostato con il parametro *Project-Id-Version* nel file **Article_Info.po** o **Book_Info.po**. Per esempio, la release **3** di un libro in lingua giapponese avrebbe la seguente impostazione nel file **ja-JP/Book_Info.po**:

```
"Project-Id-Version: 3  
"
```

Notare che il numero di release di un pacchetto in una certa lingua non condivide nessuna relazione con il numero di rilascio del pacchetto relativo allo stesso documento in lingua originale o in altra lingua. Questo numero si riferisce esclusivamente ad una lingua.

3.6. Creare un documento



Nota — Personalizzare la presentazione

I parametri impostati nel file di configurazione del documento (per impostazione **publican.cfg**), permettono di controllare molti aspetti riguardanti la presentazione di un documento ([Sezione 3.1.1, «Il file publican.cfg»](#)).

Se si mantengono versioni multiple di un documento, si può creare un file di configurazione per ciascuna versione. Quando si crea un documento, si può usare l'opzione `--config` per specificare il file da usare per una particolare compilazione, per esempio:

```
publican build --formats html,pdf --langs en-US,de-DE,it-IT --config community.cfg
```

Per creare un documento:

1. Verificare che nel file **Nome_Doc.ent**, siano state configurate le entità *YEAR* e *HOLDER*, come descritto nella [Sezione 3.1.6, «Nome_Doc.ent»](#).
2. Spostarsi nella cartella radice del documento. Per esempio, se il documento ha il nome **Libro_di_Prova** e si trova in `~/libri/`, eseguire il seguente comando:

```
cd ~/libri/Libro_di_Prova
```

3. Eseguire un test, per evitare che ci siano errori di compilazione nel linguaggio desiderato, per esempio:

```
publican build --formats=test --langs=it-IT
```

4. Eseguire il seguente comando per creare il libro:

```
publican build --formats=formati --langs=lingue
```

Sostituire *formati* con la lista dei formati d'uscita, separati da virgole, per esempio **html,html-single,pdf**. Sostituire *lingue* con la lista dei codici linguistici, separati da virgole, per esempio **en-US,sv-SE,it-IT,ko-KR**.

Formati per l'azione **build**

html

Publican presenta il documento in multiple pagine HTML, con ciascun capitolo e le principali sezioni su pagine separate. **Publican** crea un indice all'inizio del documento e posiziona controlli di navigazione su ogni pagina.

Usare i parametri *chunk_first* e *chunk_section depth* nel file **publican.cfg**, per controllare il tipo di suddivisione delle sezioni in questo formato d'uscita.

html-single

Publican presenta il documento in una singola pagina HTML con la tabella dei contenuti posta in cima alla pagina.

html-desktop

Publican presenta il documento in una singola pagina HTML con la tabella dei contenuti posta in un pannello separato a sinistra del documento.

man

Publican presenta il documento in pagine di manuale (o "man page"), in uso nei Sistemi Operativi Linux, Unix e simili.

pdf

Publican presenta il documento in un file PDF.

test

Publican controlla la validità della struttura XML del libro, senza effettuare nessuna trasformazione.

txt

Publican presenta il documento in un singolo file di testo.

epub

Publican presenta il documento in un e-book in formato EPUB.

eclipse

Publican presenta il documento come un plugin d'aiuto di **Eclipse**. Vedere la [Sezione 3.1.1, «Il file publican.cfg»](#) per i dettagli sui parametri d'impostazione *ec_id*, *ec_name* ed *ec_provider*.

I seguenti esempi mostrano alcuni comandi comunemente usati con **publican build**:

publican build --help

Elenca le opzioni disponibili in **publican build** per creare un libro.

publican build --formats=test --langs=lingue

Controlla che il libro compili correttamente. Compilare con **--formats=test** prima di eseguire ogni altro comando **publican build**, e prima di riportarlo su un repository di controllo versione, da cui altri contributori possano scaricarlo.

publican build --formats=html --langs=lingue

Compila il libro in formato HTML su pagine multiple. Il documento in HTML viene salvato nella directory **Nome_Doc/tmp/lingua/html/**. Ogni capitolo e ogni sezione principale viene disposto in un file HTML separato. E' possibile controllare il livello di suddivisione delle sotto-sezioni da presentare su pagine HTML separate, con il parametro **chunk-section-depth** in **publican.cfg** — fare riferimento alla [Sezione 3.1.1, «Il file publican.cfg»](#).

publican build --formats=html-single --langs=lingue

Crea il libro in formato HTML su una pagina singola. Il documento è un unico file HTML salvato nella directory **Nome_Doc/tmp/lingua/html-single/**.

publican build --formats=pdf --langs=lingue

Crea il libro in formato PDF. **Publican** si appoggia ad una applicazione esterna, **FOP** per rendere il PDF. Quindi, la compilazione per PDF potrebbe non essere disponibile su tutti i sistemi, dipendendo dalla disponibilità di **FOP**. Il documento è un unico file PDF salvato nella directory **Nome_Doc/tmp/lingua/pdf/**.

publican build --formats=html,html-single,pdf --langs=lingue

Crea il libro nei formati HTML su pagine multiple e su pagina singola, e PDF.

3.6.1. Compilare un documento senza controllo di validità

Publican controlla la validità della struttura XML con il *DTD* (Document Type Definition) prima di compilare il contenuto. Tuttavia in certe situazioni, come quando un documento è in fase di sviluppo, si potrebbe voler saltare la validazione durante la compilazione, soprattutto se il documento contiene riferimenti incrociati (`<xref>`) a sezioni del documento che ancora non esistono. Per saltare la validazione, eseguire **publican build** con l'opzione **--novalid**. I riferimenti a contenuto non esistente, nel documento creato, appaiono con tre punti interrogativi, **???**.

Poichè il documento non risulta validato con il DTD, la qualità della presentazione prodotta usando l'opzione **--novalid** è quantomeno sospetta. Non pubblicare i documenti compilati con l'opzione **--novalid**.

3.6.2. Compilare un documento creato con Publican 0

I documenti prodotti con precedenti versioni di **Publican** (versioni fino alla 0.45, inclusa) non hanno un file **publican.cfg**; un insieme di parametri simili veniva definito in un **Makefile**. Prima di compilare un tale documento con una versione corrente di **Publican** (versione 0.99 in poi), occorre convertire il **Makefile** in un file **publican.cfg**. **Publican** è in grado di fare automaticamente questa conversione:

1. Spostarsi nella cartella del documento, contenente il **Makefile**.
2. Eseguire **publican old2new**. **Publican** analizza il **Makefile** e crea un file **publican.cfg** con parametri equivalenti, quando possibile.

Eseguendo il comando `publican old2new`, **Publican** non modifica o elimina il **Makefile** originale. Un **Makefile** ed un file `publican.cfg` possono coesistere nello stesso documento.

3.7. Creare il pacchetto di un documento



Pacchetti non pacchetti RPM

Questa sezione descrive i pacchetti di documenti distribuiti con il **Gestore di pacchetti RPM**. Quando si usa il comando `publican package`, **Publican** genera un tarball che può essere usato per ricavare un pacchetto, da distribuire con un gestore di pacchetti software differente. Se si esegue `publican package` su un sistema senza `rpmbuild` installato, **Publican** genera ancora il tarball anche se non può creare da esso, il pacchetto RPM.



Nota — Personalizzare l'output

I parametri impostati nel file di configurazione del documento (per impostazione `publican.cfg`) permettono di controllare molti aspetti riguardanti il pacchetto di un documento ([Sezione 3.1.1, «Il file publican.cfg»](#)).

Se si mantengono versioni multiple di un documento, si può creare un file di configurazione per ciascuna versione. Quando si crea il pacchetto di un documento, si può usare l'opzione `--config` per specificare il file da usare per una particolare compilazione, per esempio:

```
publican package --lang it-IT --config community.cfg
```

Publican non solo crea documenti, ma può creare anche *pacchetti RPM* di questi contenuti, da distribuire su workstation o server web. I pacchetti RPM sono usati per distribuire software nei computer con sistemi operativi Linux che usano un **Gestore di pacchetti RPM**. Tra questi sistemi operativi si annoverano Red Hat Enterprise Linux, Fedora, Mandriva Linux, SUSE Linux Enterprise, openSUSE, Turbolinux e Yellow Dog Linux, per citarne alcuni.

3.7.1. Tipi di pacchetti RPM

Publican può produrre sia *pacchetti RPM di sorgenti* (*pacchetti SRPM*) sia *pacchetti RPM di binari*. Inoltre, sia i pacchetti SRPM sia i pacchetti RPM binari possono essere configurati per essere portati su workstation o server web.

3.7.1.1. Pacchetti RPM di sorgenti e pacchetti RPM di binari

An SRPM package contains the source code used to generate software rather than the software itself. To use an SRPM package, a computer must *compile* the source code into software — or in this case, into documents. SRPM packages of **Publican** documents contain XML files rather than finished documents (HTML, PDF, EPUB, etc) and a `spec` schema file to instruct the system how to build binary package.

Diversamente, i pacchetti RPM binari contengono software — o in questo caso, un documento — pronto per essere salvato nel file system del computer ed essere immediatamente usato. I contenuti di un pacchetto RPM binario non necessitano di essere compilati dal sistema su cui vengono installati e perciò, il sistema non ha bisogno di installare **Publican**.

3.7.1.2. Pacchetti per il desktop e pacchetti per il web

Publican può creare pacchetti di documenti che possono essere consultati su una workstation (*pacchetto RPM desktop*) o che possono essere installati su un server web e resi pubblici sul World Wide Web (*pacchetto RPM web*). Il pacchetto RPM desktop generato da **Publican**, e il pacchetto RPM web dello stesso documento differiscono per il fatto che il pacchetto RPM desktop installa solo la documentazione per l'uso sul computer locale, mentre l'RPM web installa anche il contenuto necessario a servire il World Wide Web.

I pacchetti RPM (binari) desktop di documenti creati con **Publican**, contengono la documentazione in formato HTML, su pagina singola. I documenti distribuiti con questi pacchetti, vengono installati in una sotto-cartella di `/usr/share/doc/`, secondo la specifica *FHS (Filesystem Hierarchy Standard)* della 'Miscellaneous documentation'.³ Il pacchetto RPM desktop contiene anche un file *desktop*, salvato in `/usr/share/applications/`. Questo file abilita gli *ambienti desktop* come GNOME e KDE, ad aggiungere il documento al menu del desktop, facilitando l'accesso agli utenti. In GNOME, per impostazione predefinita, la voce viene inserita sotto il menu **Sistema** → **Documentazione**. Se si desidera personalizzare il posizionamento della voce nel menu, occorre creare un pacchetto per il menu dei documenti, che fornisce i file `.directory` e `.menu` ed occorre impostare nel file `publican.cfg`, i parametri `dt_requires`, per richiedere l'uso del pacchetto per il menu dei documenti, ed il parametro `menu_category` per fornire la categoria di menu appropriata. Fare riferimento alla [Sezione 3.7.1.3, «Voci nel menu del desktop per i documenti»](#).

Per impostazione, i pacchetti RPM web dei documenti di **Publican**, contengono la documentazione in formato HTML, su pagina singola e pagine multiple, e nei formati EPUB e PDF. I formati inclusi variano se:

- si pubblica documentazione in una lingua, al momento priva di supporto per la presentazione in PDF. **Publican** si basa su **FOP** per generare il PDF. Allo stato attuale, **FOP** non supporta la scrittura sinistrorsa, come d'uso nella lingua araba, iraniana o ebraica. Inoltre, poichè allo stato attuale, **FOP** non è in grado di specificare i font carattere per carattere, una deficienza di font in scritture indo-arie, tra cui anche alcuni glifi latini, impediscono a **Publican** di generare presentazioni PDF in lingue indo-arie. Per impostazione, **Publican** non include i file PDF nei pacchetti web generati per la lingua araba, iraniana, ebraica o altra lingua indo-aria.
- il libro o brand contiene il parametro `web_formats`. Il valore di questo parametro, scavalca i formati creati, in modo predefinito, da **Publican**. Per esempio, per pubblicare un documento solo nei formati HTML su pagina singola, PDF e testo, impostare `web_formats: html-single,pdf,txt`.

Il contenuto dei pacchetti è installato in sotto-cartelle di `/var/www/html/`, una comune *radice di documenti* per server web. Notare che un pacchetto SRPM web genera sia il pacchetto RPM binario per il web sia il pacchetto RPM binario per il desktop.

3.7.1.3. Voci nel menu del desktop per i documenti

Per impostazione, i pacchetti RPM dei documenti di **Publican** per il desktop, appaiono negli ambienti GNOME, sotto il menu **Sistema** → **Documentazione**. All'aumentare del numero di documenti installati, questo menu diventa intricato e di difficile navigazione. Ciò porta ad un menu con molti documenti per differenti prodotti ed in alcuni casi, in diverse lingue, che crea non poca confusione.

Per raggruppare i documenti per un prodotto, in un sotto-menu separato, nel menu **Sistema** → **Documentazione** di GNOME, occorre:

- creare e distribuire un pacchetto desktop-menu (per il menu del desktop), che crea il sotto-menu.

³ Fare riferimento a <http://www.pathname.com/fhs/pub/fhs-2.3.html#USRSHAREARCHITECTUREINDEPENDENTDATA>

- specificare nel documento, la *category* del menu, ed opzionalmente, nel pacchetto di documentazione, specificare una direttiva *require* per richiedere anche l'installazione del pacchetto `desktop-menu`.

Notare che il menu **Documentazione** non raggruppa le voci in sotto-menu finchè non sono presenti due o più documenti appartenenti al sotto-menu. Il primo documento compare sotto **Sistema** → **Documentazione**.

3.7.1.3.1. Creare un pacchetto `desktop-menu`

Un pacchetto `desktop-menu` consiste di:

- un file *desktop entry* (**.directory**) che fornisce i metadati sul nuovo sotto-menu.
- un file *desktop menu* (**.menu**) che definisce la posizione del sotto-menu nel menu **Documentazione**.

La struttura del file **.directory**, per documentazione generata da **Publican**, è la seguente:

- il *group header* [**Desktop Entry**]
- il parametro *Name*, impostato con il nome del sotto-menu da posizionare nel menu **Documentazione**.
- opzionalmente, le localizzazioni per il parametro *Name*, nel formato *Name[codice_lingua]*, dove *codice_lingua* è un codice linguistico in formato glibc, *non* nel formato XML usato da **Publican**.
- il parametro *Comment*, impostato con una descrizione del sotto-menu.
- opzionalmente, le localizzazioni per il parametro *Comment*, nel formato *Comment[codice_lingua]*, dove *codice_lingua* è un codice linguistico in formato glibc, *non* nel formato XML usato da **Publican**.
- il parametro *Type*, impostato a **Directory**.
- il parametro *Encoding*, impostato a **UTF-8**.

Esempio 3.2. Esempio di file `.directory`

Il seguente file, **menu-example.directory** mostra la struttura di un file di desktop entry:

```
[Desktop Entry]
Name=Example
Name[fr]=Exemple
Name[it]=Esempio
Comment=Example Documentation menu
Comment[fr]=Exemple d'une menu de documentation
Comment[it]=Esempio di un menù di documentazione
Type=Directory
Encoding=UTF-8
```

Il file di desktop entry, è salvato in `/usr/share/desktop-directories/`

Per una descrizione completa di come funzionano le desktop entry, fare riferimento alla *Desktop Entry Specification*, disponibile su <http://standards.freedesktop.org/entry-spec/latest/>

Un file di desktop menu, è un file XML che contiene:

- un DTD (Document Type Declaration), per la Desktop Menu Specification di freedesktop.org:

```
<!DOCTYPE Menu PUBLIC "-//freedesktop//DTD Menu 1.0//EN"
"http://www.freedesktop.org/standards/menu-spec/1.0/menu.dtd">
```

- un *elemento radice*, <Menu>, contenente:
 - un elemento <Name> contenente **Documentation**
 - un altro elemento <Menu> che a sua volta, contiene:
 - un elemento <Name> contenente **Documentation** (come l'elemento radice)
 - un elemento <Directory> contenente il nome del file di desktop entry creato, per esempio:

```
<Directory>menu-example.directory</Directory>
```

- un elemento <Includes> contenente **X-nome_categoria**, dove *nome_categoria* è un identificatore per i documenti da raggruppare nel sotto-menu. Per esempio:

```
<Includes>X-Example-Docs</Includes>
```

Esempio 3.3. Esempio di file .menu

Il seguente file, **menu-example.menu** mostra la struttura di un file di desktop menu.

```
<!DOCTYPE Menu PUBLIC "-//freedesktop//DTD Menu 1.0//EN"
"http://www.freedesktop.org/standards/menu-spec/1.0/menu.dtd"
<Menu>
  <Name>Documentation</Name>
  <Menu>
    <Name>Documentation</Name>
    <Menu>
      <Name>Example</Name>
      <Directory>menu-example.directory</Directory>
      <Include>
        <Category>X-Example-Docs</Category>
      </Include>
    </Menu>
  </Menu>
</Menu>
```

Il file di desktop menu è salvato in `/etc/xdg/menus/settings-merged/`

Per una descrizione completa di come funzionano i desktop menu, fare riferimento alla *Desktop Menu Specification*, disponibile su <http://standards.freedesktop.org/desktop-menu-spec/latest/>

3.7.1.3.2. Impostare un categoria di desktop menu

Per posizionare un documento in un sotto-menu di **Sistema** → **Documentazione**, impostare il parametro `menu_category` nel proprio file `publican.cfg`, in modo da corrispondere al contenuto dell'elemento <Includes> nel relativo file di desktop menu. Per esempio, si consideri il file di desktop menu, contenente l'elemento:

```
<Includes>X-Example-Docs</Includes>
```

Per posizionare un documento nel sotto-menu definito da questo file di desktop menu, il file **publican.cfg** del documento, dovrebbe contenere:

```
menu_category: X-Example-Docs
```

Notare che si può includere questo parametro nel file **defaults.cfg** o nel file **overrides.cfg** di un brand, cosicché tutti i documenti creati con il brand siano automaticamente raggruppati nel sotto-menu, senza dover impostare il parametro *menu_category* in ogni documento.

Se si distribuiscono i file di desktop menu e di desktop entry in un pacchetto RPM, si può impostare che i pacchetti RPM di documenti *richiedano* il pacchetto di desktop-menu. Impostando questa dipendenza, il pacchetto di desktop-menu è installato automaticamente sul sistema degli utenti, durante l'installazione del pacchetto di documentazione, in modo da apparire nel sotto-menu creato per il progetto. Impostare la dipendenza con il parametro *dt_requires*, nel file **publican.cfg** del documento. Per esempio, se si distribuisce un pacchetto di desktop-menu di nome *foodocs-menu*, impostare:

```
dt_requires: foodocs-menu
```

Notare che si può includere il parametro nel file **defaults.cfg** o nel file **overrides.cfg** di un brand, cosicché tutti i documenti creati con il brand richiedano lo stesso pacchetto di desktop-menu.

3.7.2. Il comando **publican package**

Usare il comando **publican package --lang=Codice_Lingua** per creare il pacchetto del documento, da distribuire nella lingua specificata con l'opzione **--lang**. Fare riferimento all'[Appendice F, Codici di lingua](#) per maggiori ragguagli sull'uso dei codici linguistici.

Se si esegue **publican package** senza altre opzioni oltre alla obbligatoria opzione **--lang**, **Publican** genera un pacchetto SRPM web. La lista completa delle opzioni per **publican package** è la seguente:

--lang lingua

specifica la lingua per cui creare il pacchetto della documentazione.



Traduzioni incomplete

Se la traduzione in una particolare lingua è incompleta alla scadenza della release, si consideri di segnare la lingua con il parametro *ignored_translations* nel file **publican.cfg** del documento. Il pacchetto viene creato con un nome appropriato per la lingua, ma contiene la documentazione nella lingua originale del XML, invece di una traduzione parziale. A traduzione completata, rimuovere il parametro *ignored_translations*, incrementare il numero di release del campo **Project-Id-Version**, nel file **Book_Info.po** per la lingua, e rigenerare il pacchetto. Al momento della distribuzione del pacchetto revisionato, esso sostituisce il pacchetto originale non tradotto.

--config nome_file

specifica di usare un file di configurazione diverso dal predefinito, **publican.cfg**.

--desktop

specifica di creare un pacchetto RPM desktop invece di un pacchetto RPM web.

--brew

specifica di trasferire il pacchetto completato su **Brew**. **Brew** è il sistema di compilazione usato all'interno di Red Hat; questa opzione non ha effetto al di fuori di Red Hat.

--scratch

quando usato con le opzioni **--brew** e **--desktop**, specifica di compilare il pacchetto SRPM come uno *scratch build* (compilazione di lavoro) da trasferire su **Brew**. Gli *scratch build* sono usati per verificare la correttezza strutturale del pacchetto SRPM, senza aggiornare il database dei pacchetti con il pacchetto risultante.

--short_sighted

specifica di creare il pacchetto senza includere il numero di versione del software (*version* nel file **publican.cfg**), nel nome del pacchetto.



Omettere il numero di versione del software

Molta documentazione per software è versione specifica. Nella migliore delle ipotesi, le procedure descritte nella documentazione per una versione di un prodotto potrebbero non essere d'aiuto per installare, configurare o usare una versione differente del prodotto. Nella peggiore, le procedure descritte per un prodotto potrebbero essere fuorvianti o addirittura dannose se applicate ad una versione differente.

Se si distribuisce documentazione attraverso pacchetti RPM, senza numeri di versione nei nomi dei pacchetti, il **Gestore dei Pacchetti RPM** sostituisce automaticamente, sui sistemi degli utenti, la documentazione esistente con l'ultima versione disponibile; gli utenti non potrebbero avere accesso a più di una versione alla volta. Assicurarsi di volere realmente questo risultato.

--binary

specifica di compilare il pacchetto come un pacchetto RPM binario.

--cvs

specifica di importare il pacchetto SRPM nel CVS. Ciò richiede di impostare i parametri *cvs_root*, *cvs_pkg* e *cvs_branch* nel file di configurazione, **publican.cfg** del documento.

Dopo aver eseguito il comando **publican package**, **Publican** salva i pacchetti SRPM completati nella cartella **tmp/rpm** del documento, e i pacchetti RPM binari nella cartella **tmp/rpm/noarch**.

Per impostazione, i pacchetti di documentazione generati da **Publican**, sono denominati

nomeprodotto-titolo-numeroprodotto-[web]-lingua-edizione-numeropubblicazione.build_target.noarch.estensione_file.

Publican usa le informazioni nel file **publican.cfg** del documento, per fornire i vari parametri nel nome di file, e poi le informazioni nel file **Book_Info.xml** per altre informazioni non presenti in **publican.cfg**. Fare riferimento alla [Sezione 3.1, «I file nella directory del libro»](#) per i dettagli su questi file di configurazione. Inoltre:

- i pacchetti RPM per il web includono l'elemento **-web-** tra la versione del prodotto ed il codice della lingua.
- i pacchetti SRPM hanno l'estensione **.src.rpm** mentre i pacchetti RPM binari l'estensione **.rpm**.
- i pacchetti RPM binari includono **build_target.noarch** prima della estensione del file, dove *build_target* rappresenta sistema operativo e versione relativa, per cui il pacchetto è stato

compilato, come specificato dal parametro `os_ver`, nel file `publican.cfg`. L'elemento `noarch` specifica che il pacchetto può essere installato su ogni sistema, indipendentemente dall'architettura di sistema.

- l'uso dell'opzione `--short_sighted`, rimuove il termine `-numero-prodotto-` dal nome del pacchetto.
- i pacchetti di documenti tradotti, prendono i numeri di release dal parametro `Project-Id-Version` nei file `Article_Info.po` o `Book_Info.po`. Questo numero è specifico alla particolare lingua e non ha alcuna relazione con i numeri di release dello stesso documento, nella lingua originale o in altra lingua.

3.7.2.1. Il comando `publican package` — Esempi d'uso

I seguenti esempi mostrano alcune opzioni comuni, riferite alla II edizione, VI revisione del documento *Foomaster 9 Configuration Guide*.

`publican package --lang=it-IT`

genera un pacchetto SRPM desktop di nome `Foomaster-Configuration_Guide-9-web-it-IT-2-6.src.rpm` contenente i sorgenti della documentazione in italiano ed uno spec file per creare i documenti.

`publican package --desktop --lang=it-IT`

genera un pacchetto SRPM desktop di nome `Foomaster-Configuration_Guide-9-it-IT-2-6.src.rpm` contenente la documentazione in italiano.

`publican package --binary --lang=it-IT`

genera sia un pacchetto RPM web di nome `Foomaster-Configuration_Guide-9-web-it-IT-2-6.el5.noarch.rpm` sia un pacchetto RPM binario desktop di nome `Foomaster-Configuration_Guide-9-it-IT-2-6.el5.noarch.rpm` contenenti la documentazione in italiano, compilati per il sistema operativo Red Hat Enterprise Linux 5.

`publican package --desktop --binary --lang=it-IT`

genera un pacchetto RPM binario desktop di nome `Foomaster-Configuration_Guide-9-it-IT-2-6.el5.noarch.rpm` contenente la documentazione in italiano, compilati per il sistema operativo Red Hat Enterprise Linux 5.

`publican package --desktop --short_sighted --lang=it-IT`

genera un pacchetto SRPM desktop di nome `Foomaster-Configuration_Guide-it-IT-2-6.src.rpm`, contenente la documentazione in italiano. Questo pacchetto sostituisce ogni precedente versione della Configuration Guide di **Foomaster** esistente su un sistema. In tal caso, gli utenti non possono avere accesso contemporaneamente alla *Foomaster 8 Configuration Guide* ed alla *Foomaster 9 Configuration Guide*.

3.8. Tagging condizionale

In alcuni casi occorre mantenere multiple versioni di un libro; per esempio, un HOWTO per il prodotto FOO può avere una versione upstream ed una versione enterprise, con piccole differenze tra loro.

Publican semplifica la gestione delle differenze tra versioni multiple di un libro, permettendo di usare una sorgente unica per tutte le versioni. Il *tagging condizionale* garantisce che il contenuto di una versione specifica, compaia soltanto nella versione stabilita; ossia, permette di *condizionare* il contenuto.

Per condizionare il contenuto di un libro, usare il tag **condition**. Per esempio, si supponga che il libro *How To Use Product Foo* abbia una versione "upstream", una versione "enterprise" e una versione "beta".

```
<para condition="upstream">
Content that only appears in the upstream version of How To Use Product Foo.
</para>

<para condition="enterprise">
Content that only appears in the enterprise version of How To Use Product Foo.
</para>

<para condition="beta">
Content that only appears in the beta version of How To Use Product Foo.
</para>

<para condition="beta,enterprise">
Content that only appears in the beta and enterprise versions of How To Use Product Foo.
</para>
```

Per compilare una versione specifica (e quindi catturare solo il contenuto relativo a questa versione), aggiungere il parametro **condition: versione** al file **publican.cfg** ed eseguire il comando **publican build**, ormai noto. Per esempio, aggiungendo **condition: upstream** al file **publican.cfg** dell'HOWTO *How To Use Product Foo*, e poi eseguendo:

```
publican build --formats=pdf --langs=en-US
```

Publican filtra il contenuto in tutti gli altri tag condizionali, escluso il tag con l'attributo **condition="upstream"** e compila l'*How To Use Product Foo* in un file PDF in lingua inglese statunitense.



Nodi root e tag condizionale

Se il nodo di root di un file XML viene escluso da un tag condizionale, il documento non compila, poichè file vuoti non sono file XML validi. Per esempio, se il file **Installation_and_configuration_on_Fedora.xml** contiene un solo capitolo:

```
<?xml version='1.0' encoding='utf-8' ?>
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN" "http://www.oasis-
open.org/docbook/xml/4.5/docbookx.dtd" [
]>
<chapter id="chap-Installation_and_configuration_on_Fedora" condition="Fedora">
<title>Installation and configuration on Fedora</title>

[text of chapter]

</chapter>
```

ed il capitolo è incluso in **User_Guide.xml** con un tag `<xi:include>`, il documento non compila se è presente l'impostazione **condition: Ubuntu** nel file **publican.cfg**.

Per escludere il capitolo, aggiungere un attributo condizionale al tag `<xi:include>` in **User_Guide.xml**, e non al tag `<chapter>` in **Installation_and_configuration_on_Fedora.xml**.



xref e tag condizionale

Se un `<xref>` punta ad un contenuto escluso nella compilazione da un tag condizionale, la compilazione fallisce. Per esempio, con l'impostazione **condition: upstream** nel file **publican.cfg**, il comando **publican build --formats=pdf --langs=en-US** fallisce se il libro ha un tag `<xref linkend="betasection">` che punta alla `<section id="betasection" condition="beta">`.

3.8.1. Tagging condizionale e traduzione



Usare con cautela il tagging condizionale

Usare il tagging condizionale con una certa cautela soprattutto nei libri che si presume vengano tradotti, poichè ciò crea ulteriori difficoltà ai traduttori.

Il tagging condizionale crea difficoltà ai traduttori in due modi: oscura il contesto nei file *PO* (Portable Object) e rende più ardua la rilettura/correzione del documento, a quei traduttori non molto familiari con i tag condizionali.

I file *PO* non includono gli attributi dei tag, o al momento, il contenuto dei tag condizionali, escluso nel documento finale, viene anche incluso nei file *PO*. Per esempio, se un traduttore apre il file *PO*, relativo all'*How To Use Product Foo* della [Sezione 3.8, «Tagging condizionale»](#), egli vede:

```
#. Tag: para
#, no-c-format
```

```
msgid "Content that only appears in the upstream version of How To Use Product Foo."
msgstr ""

#. Tag: para
#, no-c-format
msgid "Content that only appears in the enterprise version of How To Use Product Foo."
msgstr ""

#. Tag: para
#, no-c-format
msgid "Content that only appears in the beta version of How To Use Product Foo."
msgstr ""

#. Tag: para
#, no-c-format
msgid "Content that only appears in the beta and enterprise versions of How To Use Product
  Foo."
msgstr ""
```

In questo esempio, i soli paragrafi in cui il senso scorre logicamente da un paragrafo al successivo è tra il terzo e il quarto. Poichè entrambi i paragrafi sono presenti nella versione beta del libro, essi (si spera), sono i soli ad avere senso. Quindi, l'uso dei tag condizionali costringe i traduttori a lavorare su brevi pezzi di brani, senza possibilità di seguire il contesto globale del discorso, da un paragrafo al successivo. In queste condizioni, ne risente la qualità della traduzione, aumentando di conseguenza il tempo richiesto — ed anche il costo della traduzione.

Inoltre, a meno che i traduttori non sappiano come configurare il file **publican.cfg** di **Publican** e non siano a conoscenza dei tag condizionali, essi non riuscirebbero a rivedere il documento finale tradotto. Se i traduttori non sono informati sull'uso di questi tag, quando procedono a rivedere un documento, rimarrebbero sorpresi nel non riuscire a trovare il testo che hanno tradotto e che facilmente trovano nei file PO. Quindi, se occorre usare i tag condizionali, ricordarsi di informare i propri traduttori del loro uso, fornendo loro il necessario supporto.

In alternativa ai tag condizionali, si consideri di mantenere versioni separate di un libro, in branch separati di un repository con controllo di versione. In tal modo, è anche possibile condividere file XML e PO tra i vari branch; inoltre alcuni sistemi di controllo versione permettono di condividere correttamente le modifiche tra branch.

3.9. Software pre-release e documentazione draft

La documentazione completata per software pre-release non coincide con la documentazione draft (bozza).

Un documento draft è una versione incompleta di un libro o articolo, ed il cui stato incompleto non è correlato allo stato del software documentato.

Tuttavia in ogni caso, è importante comunicare chiaramente lo stato del software, della documentazione o di entrambi a tutti gli interessati: utenti, sviluppatori, lettori e revisori.

3.9.1. Denotare il software pre-release

La documentazione per software pre-release, soprattutto quello distribuito a tester, clienti e partner, dovrebbe portare un contrassegno, a denotare lo status beta del software.

Per creare il contrassegno, seguire la seguente procedura:

1. Aggiungere il numero di versione del software pre-release, o un informazione di stato equivalente, al tag `<subtittle>` nel file **Book_Info.xml**. Inserire questa informazione tra tag `<remark>`. Per esempio:

```
<subtitle>Using Red Hat Enterprise Warp Drive<remark> Version 1.1, Beta 2</remark></
subtitle>
```

2. Aggiungere `show_remarks` al file `publican.cfg` ed impostarlo ad '1':

```
show_remarks: 1
```

Compilando il libro con questo tag `<remark>` e con l'impostazione `show_remarks`, si definisce in modo chiaro ed inequivocabile la natura pre-release del software. Le compilazioni PDF visualizzano il contrassegno sulla copertina e sulla pagina del titolo. Le compilazioni HTML (sia su pagina singola sia su pagine multiple), visualizzano il contrassegno sulla pagina iniziale del file `index.html`.

Poichè questo approccio non modifica le informazioni nel file `Book_Info.xml` usato per generare gli RPM, esso garantisce anche che non ci sia alcuna ambiguità nelle operazioni del sottosistema RPM.



Importante

Rimane una responsabilità del redattore rimuovere il tag `<remark>` con il suo contenuto e rimuovere o disattivare l'attributo `show_remarks` quando la documentazione viene aggiornata con la versione di release del software.

3.9.2. Denotare la documentazione draft

La documentazione incompleta resa disponibile per revisione dovrebbe essere indicata chiaramente come tale.

- Per far comparire il contrassegno *draft* ad un documento, aggiungere l'attributo `status="draft"` al tag `<article>`, `<book>` o `<set>` nel nodo radice del documento. Per esempio:

```
<book status="draft">
```

Per impostazione, il nodo radice coincide con il tag `<book>` nel file `Nome_Doc.xml`.

Nel caso di un articolo o set di volumi, il nodo radice coincide con il tag `<article>` o `<set>`, rispettivamente, nel file `Nome_Doc.xml`.

L'aggiunta dell'attributo `status="draft"` causa la comparsa su ciascuna pagina del documento del contrassegno *draft*.

Anche se si modificano solo brevi porzioni di un documento da revisionare, contrassegnare le pagine come *draft* incoraggia i lettori a riportare gli errori ed i refusi intercettati. Serve anche ad assicurare che lettori occasionali evitino di far danni scambiando un *draft* per una versione finale.

3.9.3. Denotare come draft la documentazione per software pre-release

Per denotare propriamente la documentazione incompleta relativa a software pre-release, seguire entrambe le procedure indicate in precedenza.

Branding

I *brand* sono collezioni di file usati da **Publican** per applicare alle presentazioni HTML e PDF, un look ed uno stile consistente. Essi forniscono modelli testuali con cui iniziare un documento, immagini come loghi ed elementi stilistici come schemi di colore. **Publican** viene distribuito con un unico brand, **common/**. I team che realizzano documentazione possono produrre e distribuire brand ai loro contributori, sia mediante pacchetti (per esempio pacchetti RPM), sia mediante archivi (per esempio file tarball o ZIP).

4.1. Installare un brand

In Fedora, i brand **Publican** per documenti Fedora, Genome e oVirt sono disponibili come pacchetti RPM. Analogamente, Red Hat distribuisce pacchetti RPM contenenti brand di **Publican** per documenti GIMP, JBoss e Red Hat. Accedendo ai repository dedicati, è possibile installare questi brand su computer che eseguono Red Hat Enterprise Linux o Fedora — o un sistema operativo derivato — con il comando **yum install publican-brand** o con un gestore grafico di pacchetti come **PackageKit**.

Se si usa **Publican** su un sistema operativo che non utilizza pacchetti RPM, il progetto di documentazione può fornire il proprio brand in un altro formato. Qualunque sia il formato di distribuzione del brand, i file di brand devono trovarsi in una sotto-cartella della cartella **Common_Content** di **Publican**. Per impostazione, questa cartella si trova in **/usr/share/publican/Common_Content** nei Sistemi Operativi Linux, e in **%SystemDrive%/%ProgramFiles%/Publican/Common_Content** nei sistemi operativi Windows — tipicamente **C:/Program Files/Publican/Common_Content**.

Ciascun brand attualmente disponibile, viene distribuito sotto una licenza specifica per brand, come indicato di seguito.

Installare il brand:

1. Se il brand è distribuito come un archivio, per esempio, un file tarball o ZIP, estrarre il brand in una nuova cartella.
2. Spostarsi nella cartella contenente il brand estratto:

```
cd publican-brand
```

dove *brand* è il nome del brand.

3. Compilare il brand:

```
publican build --formats xml --langs all --publish
```

4. Installare il brand:

```
sudo publican install_brand --path path
```

dove *path* è il percorso ai file Common Content di **Publican**. Per esempio, su un sistema Linux, eseguire:

```
sudo publican install_brand --path /usr/share/publican/Common_Content
```

o su un sistema Windows, eseguire:

```
publican install_brand --path "C:/Program Files/Publican/Common_Content"
```

Tabella 4.1. Brand correnti e relativi pacchetti

Brand	Licenza sui file in Common Content	Licenza predefinita sui documenti	Pacchetto	Commento
common	CC0 1.0 ¹	GFDL Version 1.2 ²	publican	Licenza GPL compatibile. Nessuna opzione.
RedHat	CC-BY-SA 3.0 ³	CC-BY-SA 3.0	publican-redhat	
Fedora	CC-BY-SA 3.0	CC-BY-SA 3.0	publican-fedora	
JBoss	CC-BY-SA 3.0	CC-BY-SA 3.0	publican-jboss	Nessuna opzione.
oVirt	OPL 1.0 ⁴	OPL 1.0	publican-ovirt	Nessuna opzione.
GIMP	GFDL Version 1.2 ⁵	GFDL Version 1.2	publican-gimp	Coincidente con la licenza sulla documentazione GIMP esistente.
Genome	OPL 1.0	OPL 1.0	publican-genome	Nessuna opzione.

Notare, nel brand common, la differenza di licenza tra i file di common content (CC0) e la licenza predefinita (GFDL), impostata sui libri generati con il brand common. La licenza CC0 consente di re-distribuire e licenziare i file (inclusi i file CSS e le immagini), che fanno parte del common brand per soddisfare alle proprie esigenze progettuali. **Publican** suggerisce la licenza GFDL per la documentazione, per impostazione predefinita, poiché **Publican** è sviluppato in primo luogo per creare documentazione per software. La licenza GFDL è compatibile con la licenza GPL, che è la licenza più comunemente usata per il software open-source.

4.2. Creare un brand

Usare l'azione `create_brand` per creare un nuovo brand. Quando si crea un nuovo brand, occorre fornire un nome e specificare la lingua originale per i file XML del brand. L'opzione `--name` fornisce il nome, e l'opzione `--lang` specifica la lingua. Quindi il comando completo è:

```
publican create_brand --name=brand --lang=codice_linguistico
```

Publican crea una nuova sotto-cartella di nome **publican-brand**, dove *brand* è il brand specificato con l'opzione `--name`.

Per esempio, per creare un brand di nome **Acme**, con i file XML in Common Content redatti in inglese statunitense, eseguire:

```
publican create_brand --name=Acme --lang=en-US
```

In tal caso, **Publican** crea il brand nella sotto-cartella **publican-Acme**.

Per configurare il nuovo brand, individuare il termine **SETUP** nei file predefiniti creati da **Publican** e modificare i file inserendo i necessari dettagli. Sui Sistemi Operativi Linux, la ricerca in questi file, del termine **SETUP** può essere effettuata con il comando:

```
grep -r 'SETUP' *
```

4.3. File nella directory brand

L'esecuzione del comando **publican create_brand --name=brand --lang=codice_linguistico** crea la struttura delle cartelle e i file necessari. La cartella brand, inizialmente contiene:

- **COPYING**
- **defaults.cfg**
- **overrides.cfg**
- **publican.cfg**
- **publican-brand.spec**, dove *brand* è il nome del brand.
- **README**
- una sotto-cartella contenente i file XML di brand, fogli di stile CSS e immagini predefinite. La sotto-cartella ha lo stesso nome del codice linguistico della lingua originale del brand (per esempio, **en-US**). Questi file sono:
 - **Feedback.xml**
 - **Legal_Notice.xml**
 - la sotto-cartella **css**, contenente:
 - **overrides.css**
 - la sotto-cartella **images**, contenente 43 immagini in formato raster o bitmap (PNG) e vettoriale (SVG).

4.3.1. Il file publican.cfg

Il file **publican.cfg**, in un brand, svolge una funzione simile al file **publican.cfg** in un documento — configura un certo numero di opzioni di base per definire il brand.

version

specifica il numero di versione del brand. Quando si crea un brand con il comando **publican create_brand**, la versione è impostata a **0.1**. Aggiornare questo numero di versione presente in **publican.cfg** e in **publican.spec** del brand, in occasione della preparazione di una nuova versione del brand.

Notare che questo parametro non è correlato al numero di versione dei documenti creati con questo brand. Per esempio, *Fedora 12 Installation Guide* ha la versione impostata a **12** nel proprio file **publican.cfg**, ma potrebbe essere compilato con la versione 1.0 del brand *publican-fedora*.

xml_lang

specifica la lingua dei file sorgenti XML per il Common Content del brand, per esempio **en-US**, come impostato nell'opzione **--lang** del comando **publican create_brand**.

release

specifica il numero di release per il brand. Quando si crea un brand con il comando **publican create_brand**, la release è impostata a **0**. Aggiornare il numero di release nei file **publican.cfg** e **publican.spec** nel brand, in occasione della preparazione di una nuova release di una versione esistente del brand.

type

impostato a **type: brand**, questo parametro identifica il contenuto nella cartella come un brand invece che come un libro, articolo o set.

brand

specifica il nome del brand, come impostato dall'opzione **--name** nel comando **publican create_brand**.

4.3.2. I file **defaults.cfg** e **overrides.cfg**

Ogni documento creato in **Publican**, ha un file **publican.cfg** nella cartella radice, per configurare le opzioni di creazione dei documenti. Fare riferimento alla [Sezione 3.1.1, «Il file publican.cfg»](#) per una completa descrizione di queste opzioni. I file **defaults.cfg** e **overrides.cfg** in un brand, forniscono valori predefiniti ai parametri, che è possibile impostare nel file **publican.cfg** di un documento.

Quando si compila un documento con un particolare brand, **Publican** dapprima applica i valori nel file **defaults.cfg** del brand e poi i valori impostati nelle file **publican.cfg** del documento. Quindi i valori di **publican.cfg** hanno la precedenza su quelli impostati nel file **defaults.cfg** del brand.

Poi **Publican** applica i valori impostati nel file **overrides.cfg** del brand, per cui quest'ultimo non tiene conto dei valori impostati in **defaults.cfg** del brand e in **publican.cfg** del documento.

Usare il file **defaults.cfg** per impostare valori da applicare per routine al brand e per consentire ai redattori di modificarli nei libri; usare il file **overrides.cfg** per impostare quei valori che non si desidera vengano modificati dai redattori.

4.3.3. File **publican-brand.spec**

Alcuni Sistemi Operativi Linux, usano un **Gestore di pacchetti RPM** per distribuire software, sotto forma di *pacchetti RPM*. In termini generali, un pacchetto RPM contiene i file di software compressi in un archivio, insieme ad un *file spec* che indica al **Gestore di pacchetti RPM** come e dove installare questi file.

Quando si crea un brand, **Publican** genera lo schema di un file spec di RPM per il brand. Il file spec generato (automaticamente), fornisce uno schema di partenza da cui creare il pacchetto RPM con cui distribuire il brand. Fare riferimento alla [Sezione 4.4, «Creare il pacchetto di un brand»](#) per sapere come configurare il file spec e come usarlo per produrre un pacchetto RPM.

4.3.4. README

Il file **README** dovrebbe includere una breve descrizione del brand.

4.3.5. COPYING

Il file **COPYING** dovrebbe fornire i dettagli riguardanti la licenza di COPYRIGHT sul brand.

4.3.6. Common Content per il brand

All'interno della cartella del brand, si trova una sotto-cartella denominata con il nome della lingua predefinita dei file XML, come impostata con l'opzione **--lang** al momento della creazione del brand. Questa cartella contiene i file XML e le immagini, che si sovrappongono al Common Content predefinito fornito con **Publican**. La personalizzazione di questi file permette ad un brand di acquisire il suo aspetto distintivo, con il suo schema di colori e i suoi loghi.

4.3.6.1. Feedback.xml

Il file **Feedback.xml** è incluso, per impostazione predefinita, nella prefazione di ogni libro prodotto con **Publican**. Con questo file si invitano i lettori ad inviare commenti sul documento. Personalizzare questo file inserendo i recapiti del progetto. Se il progetto usa un sistema di tracciamento di bug come **Bugzilla**, **JIRA** o **Trac** si potrebbe includere qui questo tipo di informazione.

4.3.6.2. Legal_Notice.xml

Il file **Legal_Notice.xml** contiene le informazioni legali che appaiono all'inizio di ogni documento prodotto con **Publican**. Inserire i dettagli riguardanti la propria licenza sul diritto d'autore in questo file. Tipicamente, ciò potrebbe includere il nome della licenza, con un breve descrizione ed un link ai dettagli completi sulla licenza.

4.3.7. La sotto-cartella **css**

La sotto-cartella **css** contiene un solo file: **overrides.css**.

4.3.7.1. overrides.css

Il file **overrides.css** imposta l'apparenza stilistica del brand. I valori in questo file sopravanzano i valori nel file **Common_Content/common/lingua_xml/css/common.css** di **Publican**.

4.3.8. La sotto-cartella **images**

La sotto-cartella **images** contiene 43 immagini in formato **PNG** (Portable Network Graphics) ed **SVG** (Scalable Vector Graphics). Queste immagini sono segnaposti per varie icone di navigazione, riquadri contenenti note/suggerimenti/avvisi/, e loghi di brand. Essi includono:

image_left

è un logo per il prodotto cui fa riferimento il documento. Esso compare in alto, nell'angolo sinistro delle pagine HTML, in cui contiene un link alla pagina web per il prodotto, come definito nel parametro *prod_url* del file **publican.cfg** del documento. Si consideri di impostare *prod_url* nei file **defaults.cfg** o **overrides.cfg** del brand.

image_right

è un logo per il team che ha prodotto la documentazione. Esso compare in alto, nell'angolo destro delle pagine HTML, in cui contiene un link alla pagina web per il team, come definito nel parametro *doc_url* del file **publican.cfg** del documento. Se tutta la documentazione per questo brand è prodotta dallo stesso team, si consideri di impostare *doc_url* nei file **defaults.cfg** o **overrides.cfg** del brand.

title_logo

è una versione più grande del logo del prodotto, che compare sulla pagina del titolo nei documenti PDF ed all'inizio nei documenti HTML.

note, important, warning

sono icone che si accompagnano agli avvisi XML `<note>`, `<important>` e `<warning>`.

dot, dot2

sono i punti elenco usati con `<listitem>` in `<itemizedlist>`.

stock-go-back, stock-go-forward, stock-go-up, stock-home

sono le icone di navigazione per le pagine HTML.

h1-bg

è uno sfondo per l'intestazione, contenente il nome del prodotto, come appare all'inizio di un documento HTML.

watermark_draft

è un contrassegno che appare nelle pagine di documentazione in draft (bozza). Fare riferimento alla [Sezione 3.9.2, «Denotare la documentazione draft»](#).

4.4. Creare il pacchetto di un brand



Pacchetti non pacchetti RPM

Questa sezione descrive i pacchetti di documenti distribuiti con il **Gestore di pacchetti RPM**. Quando si usa il comando **publican package**, **Publican** genera un tarball che può essere usato per ricavare un pacchetto, da distribuire con un gestore di pacchetti software differente. Se si esegue **publican package** su un sistema senza **rpmbuild** installato, **Publican** genera ancora il tarball anche se non può creare da esso, il pacchetto RPM.

Dopo aver creato un brand (come descritto nella [Sezione 4.2, «Creare un brand»](#)), **Publican** permette di distribuire il brand ai membri di un team di documentazione come *pacchetti RPM*. I pacchetti RPM vengono impiegati per distribuire software ai computer con sistemi operativi Linux che usano un **Gestore di pacchetti RPM**. Tra questi sistemi operativi figurano Red Hat Enterprise Linux, Fedora, Mandriva Linux, SUSE Linux Enterprise, openSUSE, Turbolinux e Yellow Dog Linux, per citarne solo alcuni.

Publican è in grado di produrre sia *pacchetti RPM di sorgenti (pacchetti SRPM)* sia *pacchetti RPM di binari*. Come parte del processo, crea anche il *file spec* — il file contenente i dettagli di come configurare ed installare il pacchetto.

Un pacchetto SRPM contiene il codice sorgente da cui generare il software, invece del software stesso. Per usare un pacchetto SRPM, un sistema deve *compilare* il codice sorgente trasformandolo in software. I pacchetti SRPM di brand creati con **Publican**, contengono file di configurazione, file XML e file di immagini che definiscono il brand nella sua lingua originale, più un file **spec** come modello da cui generare il pacchetto di brand binario. Per installare un brand da un pacchetto di sorgenti, seguire il procedimento indicato nella [Sezione 4.1, «Installare un brand»](#).

Diversamente, i pacchetti RPM di binari contengono software — in questo caso, un brand di **Publican** — pronto per essere salvato nel file system del computer ed immediatamente usato. I contenuti del pacchetto RPM binario non devono essere prima compilati sul sistema da installare, e quindi il sistema non deve aver **Publican**, installato.

Per creare il pacchetto di un brand, usare il comando **publican package** nella cartella del brand. Quando usato senza opzioni, **Publican** crea un pacchetto SRPM. Le opzioni disponibili sono:

--binary

specifica di compilare il pacchetto come un pacchetto RPM binario.

--brew

specifica di inviare su **Brew** il pacchetto completato. **Brew** è il sistema di compilazione interno a Red Hat; questa opzione non ha senso all'esterno di Red Hat.

--scratch

se usato insieme all'opzione **--brew**, specifica di compilare il pacchetto SRPM da inviare su **Brew**, come uno *scratch build* (compilazione di prova). Uno *scratch build* è usato per verificare la correttezza strutturale del pacchetto SRPM, senza aggiornare il database dei pacchetti con il pacchetto risultante.

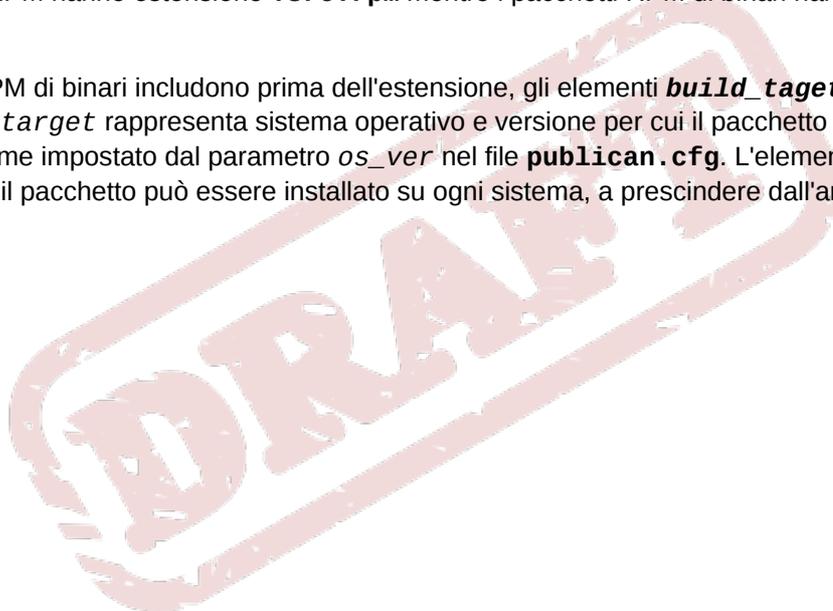
Le opzioni **--lang**, **--desktop** e **--short_sighted** che si applicano quando si crea il pacchetto di un libro (descritto nella [Sezione 3.7, «Creare il pacchetto di un documento»](#)) non hanno senso nel caso dei brand. In particolare, notare che sebbene l'opzione **--lang** sia necessaria per la creazione del pacchetto di un libro, essa non occorre quando si crea il pacchetto di un brand.

Per impostazione, i pacchetti di brand **Publican** sono denominati

publican-brand-versione-release.build_target.noarch.estensione_file

Publican usa le informazioni nel file **publican.cfg** per fornire i vari parametri nel nome di file. Fare riferimento alla [Sezione 4.3.1, «Il file publican.cfg»](#) per i dettagli sulla configurazione di questo file. Inoltre:

- i pacchetti SRPM hanno estensione **.src.rpm** mentre i pacchetti RPM di binari hanno estensione **.rpm**
- i pacchetti RPM di binari includono prima dell'estensione, gli elementi **build_target.noarch**, dove *build_target* rappresenta sistema operativo e versione per cui il pacchetto viene compilato, come impostato dal parametro *os_ver* nel file **publican.cfg**. L'elemento **noarch** specifica che il pacchetto può essere installato su ogni sistema, a prescindere dall'architettura presente.



DRAFT

Usare i set

Un *set* è una raccolta di libri pubblicati come una unica presentazione. *Services Plan*, per esempio, è un set composto da molti libri come *Developer Guide*, *Engineering Content Services Guide* ed *Engineering Operations Guide*, per citarne solo alcuni. Il comando `create_book` crea un modello per un set, impostando il parametro `type` con il valore `Set`.

Esistono due tipi di set:

- *set a sé stanti*
- *set distribuiti*

5.1. Set a sé stanti

Un set a sé stante contiene i file XML di ogni libro, i quali si trovano all'interno della cartella del set.

Il seguente procedimento indica come creare un set a sé stante di nome *My Set*, salvato nella cartella `books/My_Set/`. Il set è composto da due libri, *Book A* e *Book B*, entrambi creati manualmente all'interno della cartella `books/My_Set/en-US`.

Procedura 5.1. Creare un set a sé stante

1. In una shell, eseguire il seguente comando nella cartella `books/` per creare un set denominato `My_Set` con brand in stile Red Hat e in cui i file XML vengono redatti in inglese americano.

```
publican create --type=Set --name=My_Set --brand=RedHat --lang=en-US
```

2. `cd` nella cartella `My_Set/en-US` e creare due cartelle denominate `Book_A` e `Book_B`.

```
cd My_Set/en-US
mkdir Book_A Book_B
```

3. `cd` nella cartella `books/My_Set/en-US/Book_A`. Creare e modificare i file `Book_A.xml`, `Book_Info.xml` e gli altri file XML richiesti dal libro e gli altri dei vari capitoli. Assicurarsi che il file `Book_A.xml` contenga i corretti riferimenti in `xi:include` a tutti i file XML nella cartella. Per esempio, se *Book A* contiene `Book_Info.xml` e `Chapter_1.xml`, il file `Book_A.xml` potrebbe essere simile a:

```
<?xml version='1.0'?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN"
"http://www.oasis-open.org/docbook/xml/4.5/docbookx.dtd" [
]>

<book>
  <xi:include href="Book_Info.xml" xmlns:xi="http://www.w3.org/2001/XInclude"></
xi:include>
  <xi:include href="Chapter_1.xml" xmlns:xi="http://www.w3.org/2001/XInclude"></
xi:include>
</book>
```

4. Usare lo stesso procedimento per il `Book_B`, salvato nella cartella `books/My_Set/en-US/Book_B`, come indicato sopra.

5. Modificare il file `books/My_Set/en-US/My_Set.xml`. Per ciascun libro nel set, aggiungere un riferimento `xi:include` al file XML principale del libro. Il file XML principale per il *Book A* è `Book_A.xml` e quello per il *Book B*, `Book_B.xml`. Il file `My_Set.xml` dovrebbe assomigliare a:

```
<?xml version="1.0"?>
<!DOCTYPE set PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN"
"http://www.oasis-open.org/docbook/xml/4.5/docbookx.dtd" [
]>

<set>
  <xi:include href="Set_Info.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
  <xi:include href="Preface.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
  <xi:include href="Book_A/Book_A.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
  <xi:include href="Book_B/Book_B.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
  <xi:include href="Revision_History.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
</set>
```

6. Verificare il set, con il comando `publican build --formats=test --langs=en-US`.

5.2. Set distribuiti

Un *set distribuito* contiene libri localizzati in un repository di controllo versione. Sebbene esistano diversi sistemi di controllo versione, questa versione di **Publican** supporta solo un sistema: **SVN (Subversion)**. Impostando, nel file `publican.cfg`, la locazione del repository e i titoli dei libri contenuti nel set, ogni libro può essere esportato per creare il set completo. Il seguente procedimento, indica come creare unset denominato *My Set* contenente il *Book A* ed il *Book B*.



Importante

Di seguito so assume che il *Book A* ed il *Book B* siano già esistenti e disponibili in un repository **SVN**. Al momento, **Publican** supporta solo **SVN**.

Procedura 5.2. Creare un set

1. In una shell, eseguire il seguente comando, per creare un set denominato **My_Set** con brand in stile Red Hat e in cui i file XML vengono redatti in inglese americano.

```
$ publican create --type=Set --name=My_Set --brand=RedHat --lang=en-US
```

2. Aggiungere le seguenti righe al file `publican.cfg`:

```
books = Book_A Book_B
repo = http://PATH-TO-YOUR-SVN-REPOSITORY
scm = SVN
```

3. Modificare il file `My_Set.xml`. Per ciascun libro del set, aggiungere un riferimento `xi:include` al file XML principale del libro. Il file XML principale per il *Book A* è `Book_A.xml` e quello per il *Book B*, `Book_B.xml`. Il file `My_Set.xml` dovrebbe assomigliare a:

```
<?xml version="1.0"?>
```

```
<!DOCTYPE set PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN"
"http://www.oasis-open.org/docbook/xml/4.5/docbookx.dtd" [
]>

<set>
  <xi:include href="Set_Info.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
  <xi:include href="Preface.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
  <xi:include href="Book_A/Book_A.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
  <xi:include href="Book_B/Book_B.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
  <xi:include href="Revision_History.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
</set>
```

4. Verificare il set, con il comando **publican build --formats=test --langs=en-US**.



Importante

Quando si crea un set, il comando **publican clean_ids** verrà eseguito su ogni libro poichè la condizione di unicità degli ID deve essere valida su tutti i libri. Prestare particolare attenzione alla creazione di ID basati su contenuto che potrebbero rendersi indisponibili se i libri vengono creati isolatamente dal set.

DRAFT

DRAFT

Creare un sito web con Publican

Publican non solo crea documenti per pubblicazione ma può creare e gestire anche un sito web di documentazione. Se occorre mantenere una raccolta di documenti, si può usare **Publican** per creare un sito sul sistema locale e poi caricare il sito su un server web. Tuttavia questo approccio non è molto scalabile, per cui nei progetti di documentazione basati su team, **Publican** permette di creare pacchetti RPM di documentazione da installare sul server web. Per installare su un server i pacchetti RPM creati con **Publican**, occorre che sul server sia installato **Publican** (versione 2.1 o superiore) e l'applicazione **rpm**. Se si crea e mantiene il sito su una workstation e poi lo si carica su un server web per pubblicazione, allora non occorre che **Publican** ed **rpm** siano installati sul server web.

Il sito web creato da **Publican** consiste di quattro parti: la struttura del sito, una home page, pagine descrittive su prodotto e versione, e i documenti pubblicati sul sito. La struttura del sito, a sua volta consiste di:

- un file di configurazione
- un file di database SQLite
- una sottocartella per i documenti pubblicati, contenente:
 - **index.html** — una pagina indice che indirizza alle versioni localizzate di una home page per il sito.
 - **interactive.css** — un foglio di stile CSS contenente gli stili per il menu di navigazione, per le pagine map e per le statistiche sul sito.
 - **opds.xml** — un catalogo OPDS (Open Publication Distribution System) per permettere ai lettori di eBook compatibili di individuare facilmente i documenti sul sito.
 - **Sitemap** — una mappa del sito in XML che coadiuva i motori di ricerca nella scansione del sito.
 - **site_overrides.css** — un foglio di stile CSS alternativo a quelli presenti in **interactive.css**, per offrire stili specifici per il sito. Questo file non è creato dal processo di creazione del sito, ma deve essere aggiunto manualmente, successivamente o fornito dalla *home page* del sito.
 - **toc.html** — una mappa statica dei documenti presenti sul sito, che coadiuva i visitatori con JavaScript non abilitato nei loro browser.
 - **toc.js** — uno script JavaScript che in base all'impostazione nel browser, indirizza i visitatori al contenuto in lingua locale e che controlla la presentazione del menu di navigazione.
- sottocartelle per ogni lingua in cui pubblicare. Inizialmente, essa contiene solo **Site_Tech.html**, poi:
 - **opds.xml** — un catalogo OPDS di documenti EPUB in questa lingua.
 - **opds-prodotto.xml** — un catalogo OPDS di documenti EPUB per ogni prodotto da pubblicare in questa lingua. All'interno di ogni catalogo, per versioni differenti dello stesso prodotto, la documentazione è suddivisa in `<category>`.
 - **Site_Statistics.html** — un sommario sulla documentazione installata sul sito.
 - **Site_Tech.html** — un sommario sulla tecnologia usata per creare il sito.
 - **toc.html** — l'indice ai contenuti per la lingua, inizialmente senza alcun link a documento.

- Una sotto-cartella per ciascun prodotto da pubblicare in questa lingua.

Opzionalmente, la struttura del sito può contenere anche un *file dump* — un file XML che fornisce il dettaglio completo sul contenuto del sito per servire altri servizi, come feed web o ricerca personalizzata di pagine. La struttura potrebbe contenere anche una versione zippata del file dump. Fare riferimento alla [Sezione 6.1.1, «Creare la struttura del sito web»](#) ed alla [Sezione 6.2.1, «Creare la struttura del sito web»](#) per i dettagli sulla creazione di un file dump; mentre per una descrizione del contenuto di un file dump, vedere l'[Appendice D, Contenuto del file dump di un sito](#).

6.1. Creare manualmente un sito web

6.1.1. Creare la struttura del sito web

Per creare la struttura del sito web:

1. Sulla propria macchina, creare una nuova cartella, in cui poi spostarsi. Per esempio, su un sistema Linux, eseguire:

```
mkdir ~/docsite
cd ~/docsite
```

2. Eseguire **publican create_site**, specificando i seguenti parametri:

- **--site_config** — il nome del file di configurazione, per il sito, con estensione **.cfg**
- **--db_file** — il nome del file di database SQLite, per il sito, con estensione **.db**
- **--toc_path** — il percorso alla cartella in cui saranno salvati i documenti

Su un computer con sistema operativo diverso da Linux, impostare anche:

- **--temp_path** — il percorso alla cartella **templates/** nella directory di installazione di **Publican**. Sui computer con sistemi operativi Windows, il percorso tipicamente coincide con **%SystemDrive%\%ProgramFiles%\Publican\templates**.

Per esempio:

```
publican create_site --site_config foomaster.cfg --db_file foomaster.db --toc_path html/
docs
```

E' possibile assegnare dei nomi ai file di configurazione e di database del sito per un miglior riconoscimento del sito di appartenenza. Per esempio, per il sito di documentazione **FooMaster**, si potrebbe assegnare a questi file, rispettivamente, i nomi **foomaster.cfg** e **foomaster.db**. Il parametro **--toc_path** può essere impostato a piacere.

3. Modificare il file di configurazione specificando il nome del sito, il web host ed opzionalmente, impostare i parametri di ricerca, la lingua predefinita e il file dump per il sito:
 - a. Specificare il titolo con il parametro *title*, per esempio:

```
title: "Foomaster Documentation"
```

Normalmente, i visitatori del sito non visualizzano questo titolo perchè il JavaScript del sito li reindirizza alla homepage. Comunque il titolo è facilmente individuato ed indicizzato dai motori di ricerca.

- b. Specificare con il parametro *host*, il web host indicando l'URL completo, incluso il protocollo. Per esempio:

```
host: http://docs.example.com
```

Publican usa il valore impostato in *host* per costruire gli URL nel file XML, **Sitemap**, usato dai motori di ricerca e per limitare le ricerche dalla casella relativa nel menu di navigazione, soltanto ai documenti presenti nel sito.

- c. Opzionalmente, costruire un motore di ricerca da usare con la casella di ricerca nel menu di navigazione, e specificare l'intero contenuto del `<form>` HTML con il parametro *search*. Se non si specifica alcun criterio di ricerca, **Publican** crea una ricerca basata su Google, limitando la ricerca all'host specificato dal parametro *host*.

Per esempio, per costruire una ricerca basata su Yahoo!, limitatamente all'host **docs.example.com**, impostare:

```
search: '<form target="_top" method="get" action="http://search.yahoo.com/search"> <div class="search"> <input type="text" name="p" value="" /> <input type="hidden" name="vs" value="docs.example.com" /> <input type="submit" value="###Search###" /> </div> </form>'
```

Fare riferimento alla documentazione del motore di ricerca scelto, per i dettagli su come creare criteri di ricerca personalizzati.

Se nel codice di un pulsante di ricerca, si imposta **value="###Search###"**, **Publican** visualizza il termine **Search** sul pulsante, ovviamente tradotto in ogni lingua supportata da **Publican**.



Importante — il parametro *search* non è validato

Publican non controlla la validità del parametro *search*, ma crea il valore di questo parametro nel menu di navigazione, così come specificato. Prestare particolare attenzione nell'usare questa proprietà.

- d. Opzionalmente, impostare la lingua predefinita del sito web. **Publican** crea un menu di navigazione distinto e tradotto, per ogni lingua in cui si pubblica la documentazione. Comunque, se un documento non è disponibile in una particolare lingua, **Publican** indirizza i visitatori alla versione in lingua originale del documento. Per specificare la lingua predefinita, originale del sito, impostare il parametro *def_lang* con il codice linguistico relativo. Per esempio:

```
def_lang: it-IT
```

Impostando il parametro *def_lang* con il codice **it-IT**, i visitatori che visualizzano il menu di navigazione, per esempio in spagnolo, vengono diretti alla versione italiana del documento se non è presente una traduzione in spagnolo.

- e. Opzionalmente, configurare un *file dump* per il sito web. **Publican** può produrre un file XML con i dettagli completi del contenuto del sito per offrire altri servizi, come feed web o pagine di ricerca personalizzate. Questo file viene aggiornato ad ogni installazione o rimozione di un libro dal sito, oppure eseguendo il comando **publican update_site**. Configurare come indicato di seguito, i parametri *dump*, *dump_file* e *zip_dump*:

dump

Impostare **dump**: **1** per abilitare la funzione di file dump. Il parametro, in modo predefinito, è impostato su **0** (disattivato).

dump_file

Impostare **dump_file**: *nome* per specificare il nome del file dump e la cartella in cui salvare il file. Il parametro, in modo predefinito, è impostato su **/var/www/html/DUMP.xml**.

zip_dump

Impostare **zip_dump**: **1** per creare anche una versione zippata del file XML. Il parametro, in modo predefinito, è impostato su **0** (disattivato).

Fare riferimento all'[Appendice D, Contenuto del file dump di un sito](#) per una descrizione del contenuto del file dump.

4. Creare un file vuoto di nome **site_overrides.css** nella cartella specificata con l'opzione **doc_path** (la cartella che contiene il foglio di stile **interactive.css** e le cartelle linguistiche). Se si desidera usare degli stili specifici alternativi a quelli forniti da **interactive.css**, aggiungere un file **site_overrides.css** al documento che fornisce la *home page* del sito — vedere la [Sezione 6.1.2, «Creare, installare ed aggiornare la home page»](#). Se non si usa uno stile specifico, il file vuoto aggiunto impedirà la comparsa dell'errore 404 sul server. Su un sistema Linux, spostarsi nella cartella specificata con l'opzione **doc_path** ed eseguire:

```
touch site_overrides.css
```

Per aggiornare la struttura del sito, eseguire:

```
publican update_site --site_config percorso_al_file_di_configuration_del_sito.cfg
```

6.1.2. Creare, installare ed aggiornare la home page

La home page generata da **Publican** è la pagina di localizzazione a cui sono indirizzati i visitatori dallo JavaScript del sito e che fornisce lo stile alla struttura del sito web. La home page è strutturata come un <article> DocBook ma con un parametro ulteriore *web_type*: *home* nel proprio file **publican.cfg**. Nella sua struttura e presentazione, la home page assomiglia ad un articolo di **Publican**. Per creare la home page:

1. Spostarsi in una cartella opportuna ed eseguire il comando di **publican create**:

```
publican create --type Article --name nome_pagina
```

Per esempio:

```
publican create --type Article --name Home_Page
```

La maggior parte dei brand (incluso il brand **common**), presentano il nome del documento in grandi lettere a colori in cima alla pagina, al di sotto del banner contenente il nome del prodotto (l'opzione **--name** imposta il tag `<title>`). Quindi, per impostazione, il valore impostato con l'opzione **--name** viene presentata in primo piano ai visitatori del sito; nel precedente esempio, i visitatori vengono salutati con le parole **Home Page** sotto il banner del prodotto.

2. Spostarsi nella cartella dell'articolo:

```
cd nome_pagina
```

Per esempio:

```
cd Home_Page
```

3. Scollegare il file **Article_Info.xml** dal file XML radice.

Il contenuto del file **Article_Info.xml** serve a ben poco alla home page del sito web. Quindi, modificare il file XML radice relativo alla home page, rimuovendo il tag `<xi:include>` che collega **Article_Info.xml**. Ora **Publican** usa ancora le informazioni in **Article_Info.xml** per creare pacchetti, ma senza includerlo nella home pagina stessa.

4. Modificare il file **publican.cfg**.

Come minimo, aggiungere il parametro `web_type` ed impostarlo a **home**:

```
web_type: home
```

Il parametro `web_type: home` indica di elaborare questo documento non come una documentazione di prodotto. Questa è l'unica modifica necessaria al file **publican.cfg**. Altre modifiche opzionali al file **publican.cfg**, frequentemente utili a siti web creati con **Publican** includono:

brand

Per assegnare alla home page lo stile dei propri documenti, aggiungere:

```
brand: nome_di_brand
```

docname, product

Se il tag `<title>` o `<product>` impostati nel file **Article_Info** includono caratteri diversi dai caratteri di base, come carettri accentati, impostare come occorre i parametri `docname` e `product`.

5. Modificare il contenuto del file **nome_pagina.xml** (per esempio, **Home_Page.xml**) come ogni altro documento DocBook.

Se si rimuove il tag `<xi:include>` che collega a **Article_Info.xml**, specificare un titolo per la pagina nel formato seguente:

```
<title role="producttitle">FooMaster Documentation</title>
```

6. Se si pubblica documentazione in più lingue, creare un set di file POT ed un set di file PO per ciascuna lingua usando i comandi **publican update_pot** e **publican update_po**.

7. Per personalizzare il logo in cima al menu di navigazione, contenente un link alla home page, creare una immagine PNG 290 px × 100 px, di nome **web_logo.png**. Salvare l'immagine nella cartella **images/** della directory dei file XML del documento, per esempio, **en-US/images/**.
8. Se si desidera usare degli stili specifici alternativi a quelli forniti da **interactive.css** nel sito web, aggiungere gli stili a un file **site_overrides.css** e salvarlo nella radice dei file sorgenti (la cartella che contiene il file **publican.cfg** e le cartelle linguistiche).
9. Creare la home page in formato HTML su singola pagina con l'opzione **--embedtoc** ed installarla nella struttura del sito. Per esempio:

```
publican build --publish --formats html-single --embedtoc --langs all
publican install_book --site_config ~/docsite/foomaster.cfg --lang Codice_Lingua
```

Notare che è possibile creare con un unico comando, gli HTML per tutte le lingue, ma occorre installare la home page di ciascuna lingua con un comando **publican install_book**, individuale.

6.1.3. Creare, installare ed aggiornare pagine di prodotto e di versione

Le pagine di prodotto e le pagine di versione generate con **Publican**, sono pagine localizzabili che offrono una panoramica generale, rispettivamente, su un prodotto o una versione. I visitatori accedono a queste pagine cliccando su prodotto o versione nel menu di navigazione. Le pagine sono strutturate come un `<article>` DocBook con un ulteriore parametro *web_type*: *product* o *web_type: version* nel proprio file **publican.cfg**. Nella loro struttura e presentazione, le pagine prodotto e quelle di versione sono pressochè identiche ad ogni altro articolo prodotto con **Publican**. Per creare una pagina prodotto o di versione:

1. Spostarsi in una cartella opportuna ed eseguire il seguente comando di **publican create**:

```
publican create --type Article --name nome_pagina
```

Per esempio, per una pagina prodotto si avrebbe:

```
publican create --type Article --name FooMaster
```

o per una pagina di versione:

```
publican create --type Article --name FooMaster_3
```

2. Spostarsi nella cartella dell'articolo:

```
cd nome_pagina
```

Per esempio:

```
cd FooMaster
```

3. Scollegare il file **Article_Info.xml** dal file XML radice.

Il contenuto del file **Article_Info.xml** serve a ben poco alle pagine prodotto o versione. Quindi, modificare il file XML radice relativo alla pagina, rimuovendo il tag `<xi:include>` che

collega **Article_Info.xml**. Ora **Publican** usa ancora le informazioni in **Article_Info.xml** per creare pacchetti, ma senza includerlo nella home pagina stessa.

4. Modificare il file **publican.cfg**.

Come minimo, aggiungere il parametro *web_type* ed impostarlo a **product** o **version**:

```
web_type: product
```

o

```
web_type: version
```

Il parametro *web_type* indica di elaborare questo documento non come una documentazione di prodotto. Questa è l'unica modifica necessaria al file **publican.cfg**. Altre modifiche opzionali al file **publican.cfg**, frequentemente utili a pagine prodotto o di versione includono:

brand

Per assegnare alla pagina lo stile dei propri documenti, aggiungere:

```
brand: nome_di_brand
```

docname, product

Se il tag `<title>` o `<product>` impostati nel file **Article_Info** includono caratteri diversi dai caratteri di base non accentati, impostare come occorre i parametri *docname* e *product*.

5. Modificare il contenuto del file **nome_pagina.xml** (per esempio, **FooMaster.xml**) come ogni altro documento DocBook.

Se si rimuove il tag `<xi:include>` che collega a **Article_Info.xml**, specificare un titolo per la pagina nel formato seguente:

```
<title role="producttitle">FooMaster Documentation</title>
```

6. Se si pubblica documentazione in più lingue, creare un set di file POT ed un set di file PO per ciascuna lingua usando i comandi **publican update_pot** e **publican update_po**.

7. Creare la pagina prodotto o di versione in formato HTML su singola pagina con l'opzione **--embedtoc** ed installarla nella struttura del sito. Per esempio:

```
publican build --publish --formats html-single --embedtoc --langs all
publican install_book --site_config ~/docsite/foomaster.cfg --lang codice_lingua
```

Notare che è possibile creare con un unico comando, gli HTML per tutte le lingue, ma che occorre installare la pagina prodotto o di versione di ciascuna lingua con un comando **publican install_book**, individuale.

6.1.4. Installare, aggiornare e rimuovere documenti

Per installare un documento su un sito web da creare manualmente, spostarsi nella cartella contenente il sorgente del documento ed eseguire:

```
publican build --embedtoc --formats=lista_dei_formati --langs=codici_lingua --publish
```

```
publican install_book --site_config percorso_al_file_di_configurazione_del_sito.cfg --
lang codici_lingua
```

Notare che è possibile creare con un unico comando di **publican build**, le pubblicazioni per tutte le lingue, ma che occorre un comando **publican install_book**, individuale per ciascuna lingua. Nel comando **publican build**, occorre includere **html** come uno dei formato d'uscita ; opzionalmente, è possibile includere anche uno o più dei seguenti formati, separandoli con virgole: **html-single**, **pdf** ed **epub**.

Per aggiornare un documento, spostarsi nella cartella contenente i sorgenti del documento ed eseguire gli stessi comandi relativi ad una prima installazione di un documento. In tal modo, **Publican** sostituisce la precedente con la nuova versione.

Per rimuovere un documento, spostarsi nella cartella contenente i sorgenti del documento ed eseguire:

```
publican remove_book --site_config percorso_al_file_di_configurazione_del_sito.cfg --
lang codice_lingua
```

Una volta installati i documenti, il sito è pronto per essere caricato sul server web attraverso un qualsiasi metodo solitamente usato, come **scp**, **rsync** o un client FTP.

6.2. Creare un sito web usando pacchetti RPM

6.2.1. Creare la struttura del sito web



Avviso — Questa procedura sostituisce i file

Quando si costruisce la struttura del sito web, **Publican** salva i file nella cartella **/var/www/html/docs**. Quindi usando questa procedura, i file esistenti nella cartella vengono sovrascritti.

Sul server web, effettuare i seguenti passaggi. Occorre accedere con un account con privilegi di root.

1. Loggarsi sul server web.
2. Divenire l'utente root:

```
su -
```

3. Installare **Publican**. Per esempio, su un server con Sistema Operativo Fedora, eseguire:

```
yum install publican
```

4. Modificare il file **/etc/publican-website.cfg**, specificando il nome del sito, il web host ed opzionalmente impostare i parametri di ricerca, la lingua predefinita e il file dump per il sito:
 - a. Specificare il titolo con il parametro *title*, per esempio:

```
title: "Foomaster Documentation"
```

Normalmente, i visitatori del sito non visualizzano questo titolo perchè rediretti alla home page dallo JavaScript del sito. Comunque questo titolo è individuato ed indicizzato dai motori di ricerca.

- b. Specificare il web host con il parametro *host* come un URL completo, includente il protocollo. Per esempio:

```
host: http://docs.example.com
```

Publican usa il valore impostato in *host* per costruire gli URL nel file XML, **Sitemap**, usato dai motori di ricerca e per limitare le ricerche dalla casella relativa nel menu di navigazione, soltanto ai documenti presenti nel sito.

- c. Opzionalmente, costruire un motore di ricerca da usare con la casella di ricerca nel menu di navigazione, e specificare l'intero contenuto del `<form>` HTML con il parametro *search*. Se non si specifica alcun criterio di ricerca, **Publican** crea una ricerca basata su Google, limitando la ricerca all'host specificato nel parametro *host*.

Per esempio, per costruire una ricerca basata su Yahoo!, limitatamente all'host **docs.example.com**, impostare:

```
search: '<form target="_top" method="get" action="http://search.yahoo.com/search"> <div class="search"> <input type="text" name="p" value="" /> <input type="hidden" name="vs" value="docs.example.com" /> <input type="submit" value="###Search###" /> </div> </form>'
```

Fare riferimento alla documentazione del motore di ricerca scelto, per i dettagli su come creare criteri di ricerca personalizzati.

Se nel codice di un pulsante di ricerca, si imposta **value="###Search###"**, **Publican** visualizza il termine **Search** sul pulsante, ovviamente tradotto in ogni lingua supportata da **Publican**.



Importante — il parametro *search* non è validato

Publican non controlla la validità del parametro *search*, ma crea il valore di questo parametro nel menu di navigazione, così come specificato. Prestare particolare attenzione nell'usare questa proprietà.

- d. Opzionalmente, impostare la lingua predefinita del sito web. **Publican** crea un menu di navigazione distinto e tradotto, per ogni lingua in cui si pubblica la documentazione. Comunque, se un documento non è disponibile in una particolare lingua, **Publican** indirizza i visitatori alla versione in lingua originale del documento. Per specificare la lingua predefinita, originale del sito, impostare il parametro *def_lang* con il codice linguistico relativo. Per esempio:

```
def_lang: it-IT
```

Impostando il parametro *def_lang* con il codice **it-IT**, i visitatori che visualizzano il menu di navigazione, per esempio in spagnolo, vengono diretti alla versione italiana del documento se non è presente una traduzione in spagnolo.

- e. Opzionalmente, configurare un *file dump* per il sito web. **Publican** può produrre un file XML con i dettagli completi del contenuto del sito per offrire altri servizi, come feed web o pagine di ricerca personalizzate. Questo file viene aggiornato ad ogni installazione o rimozione di un libro dal sito, oppure eseguendo il comando **publican update_site**. Configurare come indicato di seguito, i parametri *dump*, *dump_file* e *zip_dump*:

dump

Impostare **dump**: **1** per abilitare la funzione di file dump. Il parametro, in modo predefinito, è impostato su **0** (disattivato).

dump_file

Impostare **dump_file**: *nome* per specificare il nome del file dump e la cartella in cui salvare il file. Il parametro, in modo predefinito, è impostato su **/var/www/html/DUMP.xml**.

zip_dump

Impostare **zip_dump**: **1** per creare anche una versione zippata del file XML. Il parametro, in modo predefinito, è impostato su **0** (disattivato).

Fare riferimento all'[Appendice D, Contenuto del file dump di un sito](#) per una descrizione del contenuto del file dump.

5. Creare un file vuoto di nome **site_overrides.css**. Se si desidera usare degli stili specifici alternativi a quelli forniti da **interactive.css**, aggiungere un file **site_overrides.css** al documento che fornisce la *home page* del sito — vedere la [Sezione 6.2.2, «Creare, installare ed aggiornare la home page»](#). Se non si usa uno stile specifico, il file vuoto aggiunto impedirà la comparsa dell'errore 404 sul server. Su un sistema Linux, eseguire:

```
touch /var/www/html/docs/site_overrides.css
```

Per aggiornare la struttura del sito, eseguire:

```
publican update_site --site_config /etc/publican-website.cfg
```

6.2.2. Creare, installare ed aggiornare la home page

La home page generata da **Publican** è la pagina di localizzazione a cui sono indirizzati i visitatori dallo JavaScript del sito e che fornisce lo stile alla struttura del sito web. La home page è strutturata come un <article> DocBook ma con un parametro ulteriore *web_type*: *home* nel proprio file **publican.cfg**. Nella sua struttura e presentazione, la home page assomiglia ad un articolo di **Publican**, il cui pacchetto viene creato allo stesso modo.

1. Usando il procedimento indicato nella [Sezione 6.1.2, «Creare, installare ed aggiornare la home page»](#), su una macchina creare una home page.
2. Nella cartella contenente la home page creata, eseguire:

```
publican package --binary
```

Publican crea un pacchetto RPM, salvandolo nella cartella **/tmp/rpms/noarch/** della home page. Notare che per impostazione predefinita, **Publican** genera un pacchetto RPM da installare su un server che esegua Red Hat Enterprise Linux 5. Per creare un RPM da

installare su un server con un sistema operativo differente, impostare il parametro *os_var* nel file **publican.cfg** della home page.

3. Caricare il pacchetto sul server web ed installarlo con **rpm -i** o **yum localinstall**, oppure salvare il pacchetto in un repository e configurare il server perchè possa installarlo dal repository eseguendo il comando **yum install**.

Per aggiornare la home page, creare un nuovo pacchetto con un numero più alto di <edition> o di <pubsnumber>, presenti in **Article_Info.xml**. **Publican** usa questi valori per impostare i numeri di versione e di release del pacchetto RPM. In tal modo, quando si installa il pacchetto sul server, **yum** può sostituire la versione precedente con la nuova, sia usando **yum localinstall** per un pacchetto locale, sia **yum update** per un pacchetto scaricato da un repository.

6.2.3. Creare, installare ed aggiornare pagine di prodotto e di versione

Le pagine di prodotto e le pagine di versione generate con **Publican**, sono pagine localizzabili che offrono una panoramica generale, rispettivamente, su un prodotto o una versione. I visitatori accedono a queste pagine cliccando su prodotto o versione nel menu di navigazione. Le pagine sono strutturate come un <article> DocBook con un ulteriore parametro *web_type: product* o *web_type: version* nel proprio file **publican.cfg**. Nella loro struttura e presentazione, le pagine prodotto e quelle di versione sono pressochè identiche ad ogni altro articolo prodotto con **Publican**, i cui pacchetti vengono creati allo stesso modo.

1. Usando il procedimento della [Sezione 6.1.3, «Creare, installare ed aggiornare pagine di prodotto e di versione»](#), su una macchina creare una pagina prodotto o di versione.
2. Nella cartella contenente la pagina prodotto o di versione, eseguire:

```
publican package --binary
```

Publican crea un pacchetto RPM, salvandolo nella cartella **/tmp/rpms/noarch/** della pagina prodotto o di versione. Notare che per impostazione predefinita, **Publican** genera un pacchetto RPM da installare su un server che esegua Red Hat Enterprise Linux 5. Per creare un RPM da installare su un server con un sistema operativo differente, impostare il parametro *os_var* nel file **publican.cfg** della pagina prodotto o di versione.

3. Caricare il pacchetto sul server web ed installarlo con **rpm -i** o **yum localinstall**, oppure salvare il pacchetto in un repository e configurare il server perchè possa installarlo dal repository eseguendo il comando **yum install**.

Per aggiornare la pagina prodotto o di versione, creare un nuovo pacchetto con un numero più alto di <edition> o di <pubsnumber>, presenti in **Article_Info.xml**. **Publican** usa questi valori per impostare i numeri di versione e di release del pacchetto RPM. In tal modo, quando si installa il pacchetto sul server, **yum** può sostituire la versione precedente con la nuova, sia usando **yum localinstall** per un pacchetto locale, sia **yum update** per un pacchetto scaricato da un repository.

6.2.4. Installare, aggiornare e rimuovere documenti

Sulla workstation, spostarsi nella cartella contenente i sorgenti del documento ed eseguire:

```
publican package --binary --lang codice_lingua
```

Publican crea un pacchetto RPM, salvandolo nella cartella `/tmp/rpms/noarch/` del documento. Notare che per impostazione predefinita, **Publican** genera un pacchetto RPM da installare su un server che esegua Red Hat Enterprise Linux 5. Per creare un RPM da installare su un server con un sistema operativo differente, impostare il parametro `os_var` nel file `publican.cfg` del documento.

Caricare il pacchetto sul server web ed installarlo con `rpm -i` o `yum localinstall`, oppure salvare il pacchetto in un repository e configurare il server perchè possa installarlo dal repository eseguendo il comando `yum install`.

Per aggiornare un documento, creare un nuovo pacchetto con un numero più alto di `<edition>` o di `<pubsnumber>`, presenti in `Book_Info.xml` o in `Article_Info.xml`. **Publican** usa questi valori per impostare i numeri di versione e di release del pacchetto RPM. In tal modo, quando si installa il pacchetto sul server, **yum** può sostituire la versione precedente con la nuova, sia usando `yum localinstall` per un pacchetto locale, sia `yum update` per un pacchetto scaricato da un repository.

Rimuovere un documento dal server web, con il comando `rpm -e` or `yum erase`.

6.2.5. Automatizzare la pubblicazione

Giacchè **Publican** è in grado di creare pacchetti RPM, **Publican** può essere usato insieme ad un sistema di creazione di RPM come **Koji** per automatizzare il processo di pubblicazione. Quando un documento è pronto per la pubblicazione, il redattore o il traduttore eseguono il seguente comando:

```
publican package --cvs
```

A questo punto, **Publican** effettua le seguenti operazioni:

1. genera un SRPM
2. importa l'SRPM nella radice del CVS secondo l'istanza di **Koji** specificata nel file `publican.cfg` del documento
3. esegue `cvs up`
4. esegue `make build`

Se il server web controlla periodicamente la presenza di aggiornamenti nel repository in cui **Koji** salva i pacchetti creati, il server può automaticamente installare le versioni aggiornate dei documenti.

Si può anche implementare un sistema di tag in cui un redattore o traduttore può eseguire il comando `koji tag-pkg` per applicare un'etichetta o *tag*, al pacchetto creato. In tal modo è possibile avere differenti webserver su cui installare i pacchetti, in base ai tag applicati. Per esempio, si potrebbe configurare un server su una intranet per installare tutti i pacchetti con tag **draft** per renderli disponibili internamente per revisione e rilettura tecnica. E poi si potrebbe configurare il server che pubblica i documenti sul web in modo da installare i pacchetti solo se con tag **public**.

Per preparare un documento da compilare tramite una istanza di **Koji**, modificare il file `publican.cfg`, specificando:

`cvs_branch`

il nome del *branch* per il pacchetto: tipicamente una particolare versione software.

`cvs_pkg`

il nome del pacchetto in CVS. Per impostazione, in **Publican** coincide con:

prodotto-titolo-versione_prodotto-web-lingua-edizione-numeropub.

cvsv_root

la radice di CVS in cui importare l'SRPM. Per esempio, la radice CVS per l'istanza **Koji** del Fedora Project è `:ext:USERNAME@cvsv.fedoraproject.org:/cvsv/pkgs`.

Per maggiori informazioni su **Koji**, visitare <https://fedorahosted.org/koji/wiki>.



DRAFT

Frequently Asked Questions

Domanda Come si aggiunge una lingua ad un libro?

Risposta Eseguendo il comando `publican update_po --langs=codice_linguistico`, dove `codice_linguistico` è il codice della lingua che si vuole aggiungere. Si possono aggiungere più lingue per volta, sperando i codici linguistici con virgole. Per esempio, `publican update_po --langs=ja-JP` crea la cartella per la lingua giapponese ed i file PO in giapponese, mentre `publican update_po --langs=ja-JP,ko-KR` crea cartelle e file PO per la lingua giapponese e coreana.

Domanda non si specifica il codice di nazione? Per esempio, si può eseguire `publican update_po --langs=es,de,fr`?

Risposta Anche così il comando funziona. Tuttavia, se si omette il codice di nazione, il risultato potrebbe essere indefinito nel caso in cui **Publican** o un brand abbia definizioni per varietà nazionali di una lingua — come nel caso di **zh-CN** (cinese semplificato della Repubblica Popolare Cinese) e **zh-TW** (cinese tradizionale della Repubblica di Cina, o Taiwan). Anche nel caso sia definita correntemente una sola varietà linguista, è sempre opportuno includere anche il codice di nazione, cosicché un futuro aggiornamento di **Publican** non trasformi inaspettatamente un documento in lingua tedesca (**de-DE**) nel Common Content dello Schweizerdeutsch (**de-CH**) o svizzero tedesco.

Domanda Come si aggiornano tutti i file PO?

Risposta Eseguendo il comando `publican update_po --langs=all`.

Domanda Come si ottiene l'elenco completo delle opzioni di `build` di **Publican**?

Risposta Eseguendo il comando `publican build --help`.

Domanda Come si ottiene l'elenco completo dei parametri configurabili nel file `publican.cfg`?

Risposta Eseguendo il comando `publican help_config` in una cartella contenente un documento di **Publican**.

Domanda Dove si trovano i file comuni di **Publican**?

Risposta In impostazione, nella directory `/usr/share/publican/` nei Sistemi Operativi Linux e in `%SystemDrive%/%ProgramFiles%/publican/Common_Content` nei sistemi operativi Windows — tipicamente, `C:/Program Files/publican/Common_Content`.

Domanda È possibile includere nei tarball e nei pacchetti RPM un file qualsiasi?

Risposta Sì. Creando una sotto-cartella denominata `files` nella cartella del linguaggio originale, essa viene inclusa in tutti i tarball o pacchetti SRPM creati da **Publican**.



Importante

La cartella **files** non è disponibile durante la fase di validazione, per cui non è possibile `xi:include` o inglobare diversamente, i file contenuti in questa cartella nell'XML.

Domanda: Perché **Publican** segnala con warning i tag sconosciuti?

Risposta: Questi warning informano che si sta usando dei tag il cui effetto non è stato testato per attrattiva, conformità a XHTML 1.0 Strict o conformità a Section 508 (Standard d'Accessibilità USA).

Domanda: Quali brand abilitano la modalità strict? Al momento la modalità strict non è applicata.

Risposta: Al momento i brand Red Hat e JBoss abilitano la modalità strict.

Domanda: I documenti HTML vengono creati regolarmente, ma provando a creare documenti PDF, si solleva un'eccezione **java.lang.NullPointerException** senza produzione di file PDF. Cos'è che non va?

Risposta: Provare a creare un file PDF con un diverso documento — meglio se con un nuovo documento creato con il comando **publican create**. Se il problema persiste, allora probabilmente nel sistema in uso, esiste un conflitto tra **JRE** (Java Runtime Environment) e **JDK** (Java Development Kit). Se è presente una JDK, **FOP** richiede che la JDK abbia la stessa versione della JRE. Inoltre, **FOP** non può usare la **GCJ** (GNU Compiler for Java).

Eseguire **alternatives --config java** e **alternatives --config javac** per determinare la JRE e JDK in uso, poi selezionare le versioni corrispondenti e quelle prive dell'elemento **gcj** nel loro nome. Per esempio, la seguente configurazione Java visualizza una JRE ed una JDK corrispondente che consentono di creare file PDF:

```
$ alternatives --config java

There are 3 programs which provide 'java'.

  Selection    Command
-----
    1          /usr/lib/jvm/jre-1.5.0-gcj/bin/java
*   2          /usr/lib/jvm/jre-1.6.0-openjdk/bin/java
+   3          /usr/lib/jvm/jre-1.6.0-openjdk.x86_64/bin/java

Enter to keep the current selection[+], or type selection number:
```

```
$ alternatives --config javac

There are 3 programs which provide 'javac'.

  Selection    Command
-----
*+  1          /usr/lib/jvm/java-1.6.0-openjdk.x86_64/bin/javac
    2          /usr/lib/jvm/java-1.6.0-openjdk/bin/javac
    3          /usr/lib/jvm/java-1.5.0-gcj/bin/javac

Enter to keep the current selection[+], or type selection number:
```

Potrebbe essere necessario installare una JDK extra, nel caso non sia presente una JDK corrispondente alle versioni JRE presenti.

Alcune installazioni Java non configurano correttamente gli ambienti con il comando **alternatives**. Per questo problema non è stata determinata una soluzione.

Domanda: Un errore segnala che **Batik** non è presente nel classpath eppure **Batik** è installato! Cos'è che non va?

Risposta: Si ritiene che ciò sia dovuto a problemi di classpath causati dall'utilizzo di differenti versioni di JRE e JDK. Fare riferimento alla domanda precedente di questa FAQ, riguardante l'eccezione **java.lang.NullPointerException** ed usare il comando **alternatives** per assicurarsi di avere JRE e JDK corrispondenti.

Domanda: Quando un file PDF si solleva un'eccezione **Exception in thread "main" java.lang.OutOfMemoryError: Java heap space**. Cos'è che non va?

Risposta: La memoria allocata per Java, in modo predefinito, è insufficiente a creare il PDF. Occorre aumentare la memoria allocata per **FOP**. Prima di eseguire **publican build**, eseguire **echo "FOP_OPTS=' -Xms50m -Xmx700m'" > ~/.foprc**. Ciò imposta lo spazio iniziale della memoria heap a 50 MB e permette di aumentare fino ad un massimo di 700 MB.

Domanda: Le versioni precedenti di **Publican** rimuovevano i tag <para> vuoti. E le versioni di **Publican** correnti?

Risposta: **Publican** nel passato, rimuoveva i tag <para> vuoti durante la trasformazione dei file XML, in quanto i tag <para> vuoti creavano problemi agli strumenti di traduzione usati in precedenza in Red Hat e nel Fedora Project. Tag <para> vuoti, sono tag validi in DocBook XML ed ora non vengono più rimossi da **Publican**.

Domanda: Chi ha fatto il controllo ortografico?

Risposta: Le precedenti versioni di **Publican** (fino alla versione 0.45, inclusa) eseguivano un controllo ortografico durante la trasformazione dei file XML. In seguito ai giudizi negativi degli utenti, questa funzione è stata rimossa.

Eseguire il seguente script bash nella cartella radice del documento, per controllare l'ortografia nei file XML con **aspell**, un controllore ortografico da riga di comando.

```
#!/bin/sh
# Jeff Fearn 2010

ASPELL_EXCLUDES=programlisting,userinput,screen,filename,command,computeroutput,abbrev,accel,orgnam

for file in `find en-US -wholename '*/extras/*' -prune -o -name \*.xml -print`; do
  echo "Processing $file";
  aspell --list --lang=en-US --mode=sgml --add-sgml-skip=${ASPELL_EXCLUDES} < $file
  | sort -u;
  echo;
done
```

Domanda: Perché quando si creano file PDF, i tag <segmentedlist> non funzionano?

Risposta: Contare il numero di colonne presenti nei tag `<segmentedlist>`. Se i `<segmentedlist>` sono formattati come tabelle, l'XSL di DocBook limita il numero delle colonne a due soltanto, e **Publican** formatta i `<segmentedlist>` come tabelle.

Domanda: Come mai, nei PDF, si altera il colore delle immagini?

Risposta: È dovuto ad un bug in **FOP** che distorce i colori nelle immagini PNG a 24-bit. Convertire le immagini, in immagini PNG a 32-bit per circuire questo problema.

Domanda: Quando un documento, si ottiene un errore del tipo 'undefined language' — cos'è che non va?

Risposta: **Publican** l'evidenziazione del codice è generato con il modulo Perl, **Syntax::Highlight::Engine::Kate**. Se in un tag `<programlisting>` si specifica un linguaggio non riconosciuto da **Syntax::Highlight::Engine::Kate**, in fase di creazione del libro si riceve un errore. Le prime righe del messaggio d'errore sono simili a:

```
undefined language: JAVA at /usr/lib/perl5/vendor_perl/5.10.0/Syntax/Highlight/Engine/
Kate.pm
line 615.
cannot create plugin for language 'JAVA'
```

Notare che il modulo **Syntax::Highlight::Engine::Kate** è molto rigoroso con i nomi dei linguaggi ed è case sensitive. Quindi, `<programlisting language="Java">` funziona, ma `<programlisting language="java">` e `<programlisting language="JAVA">` non funzionano. Il messaggio d'errore specifica il problema relativo all'attributo del linguaggio.

Fare riferimento a <http://search.cpan.org/~szabgab/Syntax-Highlight-Engine-Kate-0.06/lib/Syntax/Highlight/Engine/Kate.pm#PLUGINS> per l'elenco completo dei linguaggi supportati da **Syntax::Highlight::Engine::Kate**, insieme al corretto uso delle lettere maiuscole e della punteggiatura.

Domanda: Come abilitare il completamento dei comandi bash per Publican?

Risposta: Il supporto al completamento dei comandi bash è una nuova funzionalità presente in **Publican 2.2**. Per abilitarla:

1. Installare il pacchetto o i pacchetti che offrono il completamento dei comando bash per il proprio sistema operativo. Per esempio, in Fedora, eseguire il comando **sudo yum install bash-completion**.
2. Aggiungere al proprio file `~/.bashrc`:

```
# Use bash-completion, if available
if [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
fi
```

3. Riavviare il terminale o eseguire **source ~/.bashrc**.

Domanda: Perché Jeff chiama Isaac 'Ivan'?

Risposta: Perché la memoria di Jeff è in mutande!

Appendice A. Elementi ed attributi non permessi



Supportato, non supportato e non permesso

Non tutti gli *elementi* (tag) ed attributi che funzionano con **Publican** sono supportati. Nello specifico, non tutti i tag sono stati testati riguardo agli effetti sulla presentazione di un documento in formato HTML o PDF.

Publican funziona con quasi tutti gli elementi e relativi attributi di DocBook 4.5, la maggior parte dei quali sono *supportati*. Gli elementi ed attributi supportati sono quelli la cui presentazione in HTML e PDF di **Publican** sono stati testati e conservano un livello di qualità accettabile.

Altri elementi ed attributi non riconosciuti dannosi o ridondanti, ma che non sono stati testati per qualità sono *non supportati*. Se il contenuto all'interno di un particolare tag DocBook non viene visualizzato correttamente in un documento HTML o PDF, il problema può derivare da una mancanza di test sulla logica di trasformazione del relativo tag. Provare a ricreare il documento e controllare l'output generato da **Publican** durante la creazione del documento. **Publican** presenta avvisi sui tag non supportati man mano che vengono elaborati i file XML.

Infine, un ristretto gruppo di elementi ed attributi sono *non permessi*. Questi elementi ed attributi sono elencati in basso, accompagnati da una spiegazione del motivo del divieto.

Usare il comando `publican print_known` per visualizzare l'elenco dei tag supportati da **Publican**, ed il comando `publican print_banned` per visualizzare l'elenco dei tag non permessi in **Publican**.

A.1. Elementi non permessi

`<caution>`, `<tip>`

DocBook XML supporta cinque tipi di *avvisi* per vari di livelli di severità: `<tip>`, `<note>`, `<important>`, `<caution>` e `<warning>`. Complessivamente, essi rappresentano un insieme di distinzioni a grana molto fine. E' improbabile che queste sottili distinzioni possano applicarsi consistentemente in un documento, in specie quando un documento è redatto o mantenuto da più persone. Inoltre, questo livello di granularità può sembrare insignificante ai lettori. Per progetto, **Publican** disabilita gli elementi `<tip>` e `<caution>`, essendo gli elementi più ridondanti del gruppo.

Usare `<note>` invece di `<tip>`, ed usare `<important>` o `<warning>` invece di `<caution>`. Alcuni criteri, in base ai quali selezionare i livelli di severità sono indicati nella sezione 'Convenzioni del Documento' della Prefazione dei libri prodotti con il brand predefinito di **Publican**.

`<entrytbl>`

Publican dipende da un'applicazione esterna, **FOP**, per rendere documenti PDF. Al momento, **FOP** non supporta l'annidamento delle tabelle (nested tables), quindi ogni tentativo di creare file PDF da documenti **Publican** contenenti tabelle annidate fallisce.

L'annidamento delle tabelle è perciò vietato almeno fino a quando non supportato in **FOP**. Se si prevede di includere una tale tabella, si riconsideri di ristrutturare la struttura dei dati

<glossdiv>, <glosslist>

Questo tag presenta i termini nei glossari in ordine alfabetico; tuttavia, i termini vengono ordinati secondo la lingua originale dell'XML, a prescindere se i termini siano tradotti in altre lingue. Per esempio, un glossario prodotto con <glossdiv> che in lingua inglese appare come:

A

Apple — an *apple* is...

G

Grapes — *grapes* are...

O

Orange — an *orange* is...

P

Peach — a *peach* is...

in italiano appare come:

A

Mela — una *mela* è...

G

Uva — l'*uva* è...

O

Arancia — un'*arancia* è...

P

Pera — la *pera* è...

Quindi tradotto in una lingua che non condivide lo stesso sistema di scrittura della lingua originale in cui è redatto l'XML, il risultato del tag è privo di senso.

<inlinegraphic>

Questo elemento presenta le informazioni contenute in un oggetto grafico, privo di una opzione per una presentazione alternativa in modalità testuale. Perciò questo tag limita l'accesso alle informazioni alle persone con ridotte capacità visive. Negli Stati in cui vige una legislazione che garantisce l'accessibilità ai contenuti elettronici alle persone con ridotte capacità visive, i documenti contenenti questo tag quindi non soddisfano queste richieste. La Section 508 del *Rehabilitation Act of 1973*¹ è un esempio di tali richieste predisposte alle Agenzie Federali degli Stati Uniti d'America.

Notare che il tag <inlinegraphic> non è valido in DocBook versione 5.

<link>

Il tag <link> offre un generico collegamento ipertestuale e quindi nulla di più dei tag <xref> e <ulink>, rispettivamente, per collegamenti interni ed esterni. Quindi il tag <link> è disabilitato per ridondanza.

<olink>

Il tag <olink> offre riferimenti tra documenti XML. Per usare il tag <olink> per riferimenti ai documenti esterni facenti parte della stessa libreria di file XML, occorre fornire l'URL del

¹ Fare riferimento a <http://www.section508.gov/>

documento da linkare. In ambienti che usano il tag `<olink>`, questi URL possono essere forniti sia con entità XML sia con script server-side. **Publican** produce documenti di vasta diffusione in cui gli URL sono sempre indispensabili per riferimenti incrociati. Quindi, il tag `<olink>` non offre alcun vantaggio sul tag `<ulink>` e perciò per ridondanza, è disabilitato.

A.2. Attributi non permessi

`<[element] xreflabel="[ogni_altra_stringa]">`

La presenza di un attributo `<xreflabel>` riduce l'usabilità delle versioni stampate di un libro. Inoltre, i valori dell'attributo non sono visibili ai traduttori, e perciò non sono traducibili.

Per esempio, il seguente pezzo:

```
<chapter id="ch03" xreflabel="Chapter Three">
  <title>The Secret to Eternal Life</title>
  <para>The secret to eternal life is...</para>
</chapter>

[more deathless prose here]

...see <xref linkend="ch03"> for details.
```

durante la trasformazione in HTML del file XML, il tag `<xref>` diventa un tag anchor, come indicato di seguito:

```
...see <a href="#ch03">Chapter Three</a> for details.
```

Il testo contenuto nel tag anchor coincide con quello nell'attributo `<xreflabel>`. In questo caso, ciò vuol dire che i lettori delle versioni stampate hanno una perdita di informazione.

Per evitare questo problema si può far coincidere il valore dell'attributo `<xreflabel>` con il testo contenuto tra i tag `<title></title>`. Tuttavia questa duplicazione aumenta il rischio di errori di battitura, senza in effetti alcun miglioramento. E riduce anche la quantità di informazione presentata ai lettori delle copie stampate.

Il seguente XML:

```
<chapter id="ch03" xreflabel="The Secret to Eternal Life">
  <title>The Secret to Eternal Life</title>
  <para>The secret to eternal life is...</para>
</chapter>

[more deathless prose here]

...see >xref linkend="ch03"> for details.
```

viene trasformato in tag anchor HTML come segue:

```
...see <a href="#ch03">The Secret to Eternal Life</a> for details.
```

Ciò non è così informativo come il testo presentato senza usare l'attributo `<xreflabel>`. Il seguente:

```
<chapter id="ch03">
  <title>The Secret to Eternal Life</title>
  <para>The secret to eternal life is...</para>
</chapter>

[more deathless prose here]

...see <xref linkend="ch03"> for details.
```

trasforma l'elemento `<xref>` come segue in formato HTML:

```
...see <a href="#ch03">Chapter 3: The Secret to Eternal Life</a> for details.
```

Comunque, ancora più importanti, sono i problemi di traduzione causati dal tag `<xreflabel>`. I valori degli attributi non sono visibili ai traduttori. Di conseguenza, non possono essere tradotti. Si consideri di nuovo il secondo esempio, di cui sopra:

```
<chapter id="ch03" xreflabel="The Secret to Eternal Life">
  <title>The Secret to Eternal Life</title>
  <para>The secret to eternal life is...</para>
</chapter>

[more deathless prose here]

...see <xref linkend="ch03"> for details.
```

In inglese, `<xref>` è ancora trasformato in un tag anchor come segue:

```
...see <a href="#ch03">The Secret to Eternal Life</a> for details.
```

Mentre i lettori della versione in lingua tedesca, visualizzeranno il seguente HTML:

```
...Sehen Sie <a href="#ch03">The Secret to Eternal Life</a> für Details.
```

Rimuovendo l'attributo `<xreflabel>`, i tag del titolo e del capitolo, appaiono propriamente tradotti al lettore. Cioè il seguente pezzo:

```
<chapter id="ch03">
  <title>The Secret to Eternal Life</title>
  <para>The secret to eternal life is...</para>
</chapter>

[more deathless prose here]

...see <xref linkend="ch03"> for details.
```

per una traduzione in lingua tedesca, viene visualizzato come:

```
...Sehen Sie <a href="#ch03">Kapitel 3: Das Geheimnis des ewigen Lebens</a> für Details.
```

E ciò, senza meravigliarsi più di tanto, è ciò che si vuole ottenere!

Quindi, per quanto fin qui detto, l'attributo *xreflabel* è vietato.

<[element] endterm="[any_string_here]">

L'attributo *endterm* permette di presentare il testo relativo all'ipertesto con un nome diverso dalla sezione o capitolo cui punta il collegamento. E ciò riduce l'usabilità della versione stampata del documento e genera difficoltà anche ai traduttori.

Il testo presentato in un elemento (come un *<xref>*), contenente l'attributo *endterm* è ricavato dal tag *<titleabbrev>* nel capitolo o sezione target. Sebbene il contenuto del tag *<titleabbrev>* sia traducibile nei file PO del documento, esso viene di fatto rimosso dal contesto del tag *<xref>*. L'assenza di questo contesto rende praticamente impossibile la traduzione di articoli e preposizioni, preservando genere e numero.

Per esempio, il seguente pezzo:

```
<chapter id="The_Secret">
  <title>The Secret to Eternal Life</title>
  <titleabbrev id="final">the final chapter</titleabbrev>

  <para>The secret to eternal life is...</para>
</chapter>

[more deathless prose here]

The solution is in <xref linkend="The_Secret" endterm="final"/>.
```

Come si vede, il testo che precede il tag *<xref>* presente nella versione inglese del documento è:

The solution is in **the final chapter**.

Un traduttore vede il tag *<titleabbrev>* nel file PO come:

```
#. Tag: titleabbrev
#, no-c-format
msgid "the final chapter"
msgstr ""
```

e vede il testo contenuto in *<xref>* in qualche altra parte del file PO (o, addirittura in un PO completamente diverso), come:

```
#. Tag: para
#, no-c-format
msgid "The solution is in <xref linkend="The_Secret" endterm="final"/>."
```

```
msgstr ""
```

Il traduttore non ha idea di come verrà sostituito il tag `<xref linkend="The_Secret" endterm="final"/>` durante la creazione del documento, per cui una traduzione in italiano potrebbe leggersi come:

```
#. Tag: para
#, no-c-format
msgid "The solution is in <xref linkend="The_Secret" endterm="final"/>."
msgstr "La soluzione è in <xref linkend="The_Secret" endterm="final"/>."
```

Notare la preposizione **in**.

Se il traduttore di lingua italiana traduce **the final chapter** in **l'ultimo capitolo**, il documento risultante si leggerebbe come:

La soluzione è in **l'ultimo capitolo**.

Il risultato è comprensibile, ma poco elegante, per l'assenza di combinazione tra preposizione ed articolo. Una traduzione più elegante sarebbe stata:

La soluzione è nell'**ultimo capitolo**.

Senza conoscere il testo che compare al posto di `<xref linkend="The_Secret" endterm="final"/>`, il traduttore in italiano non sa se usare la preposizione **in** invariata, o una delle sette possibili combinazioni con l'articolo determinativo: **nel, nei, nello, nell', negli, nella** o **nelle**.

Inoltre, notare che la combinazione della preposizione con l'articolo pone anche una problema se inserire la preposizione articolata nel tag `<xref>` o nel tag `<titleabbrev>`. E comunque qualunque sia la soluzione scelta dal traduttore, ulteriori problemi si verificano quando *endterm* è presente in altri contesti grammaticali, poichè sarebbe richiesta un'altra preposizione articolata.

A causa di queste difficoltà, in fase di traduzione di *endterm*, **Publican** non permette l'uso di questo attributo.

Appendice B. Sommario dei comandi

Opzioni di comando

publican --help

visualizza l'help

publican --man

visualizza le pagine di man relative

publican --help_actions

visualizza l'elenco delle *azioni*

publican --v

visualizza il numero di versione di **Publican**

--config file

specifica un file di configurazione per un documento, in alternativa al file **publican.cfg**, predefinito

--nocolours

disabilita la colorazione ANSI nei messaggi di log di **Publican**

--quiet

disabilita tutti i messaggi di log

Azioni

publican build

trasforma gli XML in un documento. Opzioni:

--formats

una lista di formati di compilazione, separati da virgole (necessario)

--langs

una lista di lingue, separate da virgole (necessario)

--embedtoc

inserisce la tabella dei contenuti in una presentazione HTML

--publish

imposta per pubblicazione il contenuto creato

--novalid

salta il controllo di validità DTD durante la creazione di un documento

publican bump --lang=CODICE_LINGUA

incrementa di una unità, <pubsnumber> nel file **Book_Info.xml**, dove *CODICE_LINGUA* è la lingua in cui sono redatti i file XML



Importante — Nessun supporto alle lingue tradotte

In **Publican 2.3**, **bump** funziona solo con la versione originale, non tradotta, del documento. Future versioni di **Publican** includeranno supporto anche per le lingue tradotte.

publican clean

rimuove le cartelle temporanee dalla cartella di un documento

publican clean_ids

indenta ordinatamente i file XML e ricrea gli ID degli elementi

publican clean_set

rimuove le copie locali di libri remoti, parte di un set

publican create

crea un nuovo libro, articolo o set. Opzioni:

--name

il nome del documento (necessario)

--product

il prodotto documentato

--version

la versione del prodotto documentato

--edition

l'edizione del documento

--brand

il brand per il documento

--lang

la lingua in cui gli XML verranno redatti

--type

il tipo di documento — article, book, o set

publican create_brand

crea un nuovo brand. Opzioni:

--name

il nome del documento (necessario)

--lang

la lingua in cui gli XML verranno redatti

publican create_site

crea un sito web di documentazione. Opzioni:

--site_config

il nome del file di configurazione del sito da creare (necessario)

--db_file

il nome del file di database del sito da creare (necessario)

--toc_path

percorso alla cartella in cui creare il file **toc.html** di primo livello (necessario)

--tmpl_path

percorso alla cartella template (**/usr/share/publican/templates**, per impostazione)

publican help_config

visualizza l'elenco dei parametri per il file **publican.cfg**

publican install_book

installa un documento sul sito web di documentazione

--site_config

il nome del file di configurazione del sito (necessario)

--lang

la lingua del documento da installare (necessario)

publican install_brand

configura un brand per installazione. Opzioni:

--path

percorso ai file Common Content di **Publican**. Per impostazione predefinita, **/usr/share/publican/Common_Content** nei Sistemi Operativi Linux e **%SystemDrive %/ProgramFiles/Publican/Common_Content** nei sistemi operativi Windows — tipicamente, **C:/Program Files/Publican/Common_Content**

publican lang_stats

genera un rapporto di traduzione per una lingua

--lang

la lingua per cui generare il rapporto

publican old2new

crea un file **publican.cfg** basato sul **Makefile** di un documento di **Publican 0**

publican package

crea il pacchetto di un documento o di un brand, per distribuzione. Opzioni:

--lang

la lingua in cui creare il pacchetto (necessario per documenti, irrilevante per brand)

--desktop

specifica che il pacchetto RPM di un documento è creato per uso desktop (irrilevante per brand)

--brew

invia il pacchetto al sistema di compilazione **Brew** (irrilevante al di fuori di Red Hat)

--cvs

specifica di importare il pacchetto SRPM nel CVS specificato

--scratch

usato insieme all'opzione **--brew** per specificare uno *scratch build* (irrilevante al di fuori di Red Hat)

--short_sighted

crea il pacchetto RPM privo del numero di versione di prodotto nel nome di file dell'RPM

--binary

crea il pacchetto RPM binario invece del pacchetto SRPM sorgente

`publican print_banned`

visualizza l'elenco dei tag DocBook non permessi da **Publican**

`publican print_known`

visualizza l'elenco dei tag DocBook supportati da **Publican**.

publican print_tree

visualizza la struttura ad albero dei file XML di un documento

`publican print_unused`

visualizza l'elenco dei file XML *non* inclusi in tag `<xi:include>` in un libro, articolo o set

`publican remove_book`

rimuove un documento da un sito web di documentazione

`--site_config`

il nome del file di configurazione del sito (necessario)

`--lang`

il documento in lingua da rimuovere (necessario)

`publican site_stats --site_config=nome_del_file_di_configurazione_del_sito`

genera un report statistico di un sito per documentazione. Opzione

`--site_config`

il nome del file di configurazione del sito (necessario)

publican update_pot

aggiorna i file POT in un documento

publican update_po

aggiorna i file PO in un documento

`--langs`

lista delle lingue, separate da virgole, da aggiornare o 'all' per aggiornare tutte le lingue (necessario)

`publican update_site --site_config=nome_del_file_di_configurazione_del_sito.cfg`

rigenera il contenuto modellato di un sito di documentazione. Opzione:

`--site_config`

il nome del file di configurazione del sito (necessario)

Appendice C. Parametri di `publican.cfg`

Ogni libro, articolo, documento, set o brand ha un file `publican.cfg` nella propria cartella radice. I parametri che possono essere configurati nel file `publican.cfg` sono:

docname

il nome del documento, impostato con l'opzione `--name`

version

la versione del prodotto, impostata con l'opzione `--version`

xml_lang

la lingua dei file XML sorgenti, impostata con l'opzione `--lang`

edition

il numero di edizione della documentazione, impostato con l'opzione `--edition`

type

il tipo di documento — un `<article>`, `<book>` o `<set>` DocBook, impostato con l'opzione `--type`

brand

il *brand* del documento, impostato con l'opzione `--brand`

product

il prodotto a cui è rivolta la documentazione, impostato con l'opzione `--product`

arch

l'*architettura* hardware del computer per il documento

books

una lista di book, separati da virgole, parte di un set remoto

brew_dist

il target usato per compilare il pacchetto RPM desktop in **Brew**. (Per impostazione: **docs-5E**)

bridgehead_in_toc

specifica se includere bridgehead nelle tabelle dei contenuti. (Per impostazione: **0** — bridgehead non sono inclusi)

chunk_first

specifica se visualizzare la prima sezione sulla stessa pagina del parent nelle presentazioni in HTML. (Per impostazione: **0** — la prima sezione inizia su una nuova pagina HTML)

chunk_section_depth

il livello a partire da cui le sottosezioni sono visualizzati su una stessa pagina, nelle presentazioni HTML. (Per impostazione: **4**)

classpath

il percorso ai file jar indispensabili a **FOP**. (Per impostazione, nei Sistemi Operativi Linux: `/usr/share/java/ant/ant-trax-1.7.0.jar:/usr/share/java/xmlgraphics-commons.jar:/usr/share/java/batik-all.jar:/usr/share/java/xml-commons-apis.jar:/usr/share/java/xml-commons-apis-ext.jar`)

common_config

il percorso all'installazione di **Publican**. (Per impostazione nei Sistemi Operativi Linux: **/usr/share/publican**; nei sistemi operativi Windows: **%SystemDrive%/%ProgramFiles%/publican** — solitamente **C:/Program Files/publican**)

common_content

il percorso ai file *Common Content* di **Publican**. (Per impostazione nei Sistemi Operativi Linux: **/usr/share/publican/Common_Content**; nei sistemi operativi Windows: **%SystemDrive%/%ProgramFiles%/publican/Common_Content** — solitamente **C:/Program Files/publican/Common_Content**)

condition

condizioni di filtraggio degli XML prima della trasformazione

confidential

contrassegna un documento come confidenziale. (Per impostazione: **0** — non confidenziale)

confidential_text

imposta il testo con cui contrassegnare come confidenziale un documento. (Per impostazione: **CONFIDENTIAL**)

cvs_branch

il branch CVS in cui importare il pacchetto SRPM

cvs_pkg

il pacchetto CVS in cui importare l'SRPM. Il parametro accetta la stringa **__LANG__**, sostituendola con il codice della lingua durante l'elaborazione del pacchetto

cvs_root

la root CVS in cui importare il pacchetto SRPM

debug

specifica se visualizzare i messaggi di debug generati da **Publican**. (Per impostazione: **0** — messaggi disabilitati)

def_lang

la lingua predefinita per il sito web gestito via **Publican**. Le tabelle dei contenuti, per le altre lingue, sono collegate ai documenti in lingua predefinita in assenza di traduzioni. (Per impostazione: **en-US** — inglese degli Stati Uniti d'America)

doc_url

URL del team di documentazione del pacchetto. (Per impostazione: **https://fedorahosted.org/publican**)

dt_obsoletes

un pacchetto che il pacchetto desktop rende obsoleto

dt_requires

un pacchetto richiesto dal pacchetto desktop, per esempio, un pacchetto di menu del desktop, per documentazione. Fare riferimento alla [Sezione 3.7.1.3, «Voci nel menu del desktop per i documenti»](#).

dtdver

la versione del DTD (Document Type Definition) di DocBook XML su cui si basa il progetto. (Per impostazione: **4.5**)

ec_id

l'ID del plugin d'aiuto di **Eclipse**. (Per impostazione: *prodotto.nomedoc*)

ec_name

il nome del plugin d'aiuto di **Eclipse**. (Per impostazione: *prodotto.nomedoc*)

ec_provider

il nome del fornitore del plugin d'aiuto di **Eclipse**. (Per impostazione: *Publican-versione Publican*)

generate_section_toc_level

il livello di profondità delle sezioni per cui generare una tabella di contenuti. (Per impostazione: **0** — nessun indice nelle sezioni)

ignored_translations

traduzioni da ignorare

license

la licenza usata dal pacchetto. (Per impostazione: GNU Free Documentation License)

max_image_width

la larghezza massima consentita alle immagini nel documento, se non specificato diversamente con il tag `<imagedata>` per una certa immagine. (Per impostazione: **444** — larghezza di 444 pixel)

**Importante — 444 pixel è la massima larghezza di sicurezza**

Non usare il parametro *max_image_width* se le immagini contengono importanti informazioni. Le immagini più larghe di 444 pixel potrebbero presentarsi male nei documenti HTML e PDF e rendersi inusabili, in quanto superando i margini esse verrebbero rappresentate incomplete.

Viceversa, le immagini più larghe di 444 pixel che vengono scalate in un browser web o in un visualizzatore PDF, perdono in qualità.

Per preservare la qualità delle immagini, si raccomanda di tagliare o riscalare le immagini ad una larghezza inferiore a 444 pixel, prima di includerle in un documento.

menu_category

la categoria del desktop menu (come definito dal file `.menu` corrispondente), in cui far comparire un documento quando installato da un pacchetto RPM per desktop. Fare riferimento alla [Sezione 3.7.1.3, «Voci nel menu del desktop per i documenti»](#).

os_ver

il sistema operativo per cui creare il pacchetto. (Per impostazione: **.e15** — Red Hat Enterprise Linux 5)

prod_url

URL del prodotto cui si applica il documento. (Per impostazione: **https://fedorahosted.org/publican**)

release

il numero di release del pacchetto. (Per impostazione: il valore in `xml_lang`, ricavato dal tag title in `xml_lang/TYPE_Info.xml` o `Project-Id-Version` in `lang/TYPE_Info.po`)

repo

il repository da cui scaricare libri remoti, parte di un set distribuito

scm

il sistema controllo di versione usato nel repository che contiene i libri remoti, parte di un set distribuito. (Per impostazione: **SVN**)

show_remarks

specifica se visualizzare gli avvisi nelle presentazioni. (Per impostazione: **0** — avvisi non visibili)

show_unknown

specifica se segnalare i tag sconosciuti durante l'elaborazione degli XML. (Per impostazione: **1** — tag sconosciuti segnalati)

src_url

URL in cui trovare i tarball dei file sorgenti

strict

use *strict mode* (Per impostazione: **0** — non strict) attualmente la modalità strict non è abilitata

tmp_dir

la cartella di output di **Publican**. (Per impostazione: **tmp**)

toc_section_depth

la profondità di livello delle sezioni incluse nella tabella dei contenuti. (Per impostazione: **2**)

web_brew_dist

il target di compilazione **brew** da usare per il pacchetto RPM web. (Per impostazione: **docs-5E**)

web_formats

un elenco di formati, separati da virgole, da includere nel pacchetto RPM per web. Vedere la [Sezione 3.7.2, «Il comando `publican package`»](#).

web_home

specifica che il documento è la home page di un sito di documentazione, non un documento standard



Importante — `web_home` è deprecato

In **Publican 2.2**, `web_home` è sostituito da `web_type: home`. Supporto a `web_home` sarà interrotto in future versioni di **Publican**.

web_name_label

sostituisce il nome di book che appare nel menu di un sito web gestito da **Publican**

web_obsoletes

pacchetti resi obsoleti dal pacchetto RPM web

web_product_label

sostituisce il nome di prodotto che appare nel menu di un sito web gestito da **Publican**

web_type

specifica che il documento è un contenuto descrittivo di un sito web gestito da **Publican** e non una documentazione di prodotto. Questo contenuto include la home page del sito

(**web_type: home**), pagine descrittive di prodotto (**web_type: product**) e di versione (**web_type: version**). Vedere il [Capitolo 6, Creare un sito web con Publican](#).

web_version_label

sostituisce il numero di versione che appare nel menu di un sito web gestito da **Publican**



DRAFT

Appendice D. Contenuto del file dump di un sito

Il file dump di un sito web generato da **Publican** contiene i dettagli di alcune configurazioni di base, insieme con i dettagli di ogni documento pubblicato sul sito. I dettagli di configurazione del sito sono:

<host>

L'URL alla radice del sito di documentazione, come impostato dal parametro *host*, nel file di configurazione del sito.

<def_lang>

La lingua predefinita della documentazione sul sito web, come impostato dal parametro *def_lang*, nel file di configurazione del sito.

Ogni documento, in ogni lingua, in ogni formato ha un record separato. Questi record contengono i seguenti dati:

<name>

Il titolo del documento, generato dal tag <title> in **Book_Info.xml**, **Article_Info.xml** o **Set_Info.xml**, superato dal parametro *docname* nel file **publican.cfg**. Ogni spazio presente nel titolo è sostituito con un carattere trattino basso.

<ID>

Un numero ID unico per il documento, in un certo formato e lingua.

<abstract>

Una breve descrizione generale del contenuto del documento, generato dal tag <abstract> in **Book_Info.xml**, **Article_Info.xml** o **Set_Info.xml**. **Publican** usa questo stesso contenuto per generare la sezione **%description** del file spec, usato per creare il pacchetto RPM di un documento. Se l'<abstract> è tradotto, questo campo contiene il testo tradotto.

<format>

Il formato in cui il documento è prodotto — **html** per html in pagine multiple, **html-single** per html in pagina singola, **pdf** per PDF ed **epub** per EPUB.

<language>

Il codice linguistico per il documento. Fare riferimento all'[Appendice F, Codici di lingua](#), per maggiori informazioni su questi codici in XML.

<name_label>

Il nome del documento che appare nella tabella dei contenuti del sito. Il nome può essere impostato con il parametro *web_name_label* nel file **publican.cfg** del documento. Diversamente, il campo è vuoto per un documento in lingua originale o contiene il titolo del documento tradotto. Ogni spazio presente nel nome è sostituito con il carattere trattino basso.

<product>

Il prodotto descritto dal documento, generato dal tag <productname> in **Book_Info.xml**, **Article_Info.xml** o **Set_Info.xml**, se non generato dal parametro *product* nel file **publican.cfg**. Ogni spazio presente nel nome è sostituito con il carattere trattino basso.

<product_label>

Il nome del prodotto che appare nella tabella dei contenuti del sito. Il nome può essere impostato con il parametro *web_product_label* nel file **publican.cfg** del documento. Diversamente,

il campo è vuoto per un documento in lingua originale o contiene il titolo del documento tradotto. Ogni spazio presente nel nome è sostituito con il carattere trattino basso.

Se il nome di prodotto è impostato con **UNUSED**, non viene visualizzata alcuna intestazione nella tabella dei contenuti del sito.

<subtitle>

Un'unica riga descrittiva del contenuto del documento, generato dal tag <subtitle> in **Book_Info.xml**, **Article_Info.xml** o **Set_Info.xml**. **Publican** usa questo stesso contenuto per generare la sezione **Summary** del file spec, usato per creare il pacchetto RPM di un documento. Se il <subtitle> è tradotto, questo campo contiene il testo tradotto.

<update_date>

La data di più recente installazione del documento, sul sito, nel formato YYYY-MM-DD.

<version>

La versione del prodotto descritto dal documento (*non* la versione del documento), generata dal tag <productnumber> in **Book_Info.xml**, **Article_Info.xml** o **Set_Info.xml**, se non generata dal parametro *version* nel file **publican.cfg**.

<version_label>

La versione del prodotto che compare nella tabella dei contenuti del sito. La versione può essere impostata con il parametro *web_version_label* nel file **publican.cfg** del documento.

Se la versione è impostata con **UNUSED**, non viene visualizzata alcuna intestazione per questa versione del prodotto, nella tabella dei contenuti del sito.

Esempio D.1. Record d'esempio da un file **DUMP.xml**

Questi due record da un file **DUMP.xml**, visualizzano lo stesso libro, *Red Hat Enterprise Linux 5 Installation Guide*, in due formati e due lingue differenti — una versione PDF in lingua inglese ed una versione HTML multi-pagine, in lingua francese.

```
<record>
  <name>Installation_Guide</name>
  <ID>22</ID>
  <abstract>This manual explains how to boot the Red Hat Enterprise Linux 5 installation
program (anaconda) and to install Red Hat Enterprise Linux 5 on 32-bit and 64-bit x86
systems, 64-bit POWER systems, and IBM System z. It also covers advanced installation
methods such as kickstart installations, PXE installations, and installations over VNC.
Finally, it describes common post-installation tasks and explains how to troubleshoot
installation problems.</abstract>
  <format>pdf</format>
  <language>en-US</language>
  <name_label>Installation_Guide</name_label>
  <product>Red_Hat_Enterprise_Linux</product>
  <product_label>Red_Hat_Enterprise_Linux</product_label>
  <subtitle>Installing Red Hat Enterprise Linux 5 for all architectures</subtitle>
  <update_date>2010-10-07</update_date>
  <version>5</version>
  <version_label></version_label>
</record>
<record>
  <name>Installation_Guide</name>
  <ID>149</ID>
  <abstract>Ce manuel explique comment lancer le programme d'installation Red Hat
Enterprise Linux 5 et comment installer Red Hat Enterprise Linux 5 sur les systèmes
x86 32-bit et 64-bit, sur les systèmes POWER 64-bit, et sur les systèmes IBM System z.
Il couvre aussi des méthodes d'installation avancées telles que les installations
kickstart, PXE, et les installations au moyen de VNC. Finalement, ce manuel décrit les
```

```
tâches communes post-installation et explique comment résoudre les problèmes liés à une
installation.</abstract>
<format>html</format>
<language>fr-FR</language>
<name_label>Guide_d'installation</name_label>
<product>Red_Hat_Enterprise_Linux</product>
<product_label>Red_Hat_Enterprise_Linux</product_label>
<subtitle>Installation de Red Hat Enterprise Linux 5 pour toutes les architectures</
subtitle>
<update_date>2010-10-19</update_date>
<version>5</version>
<version_label></version_label>
</record>
```

D.1. Ricavare gli URL dai file dump

Usando i seguenti campi, si può ricavare l'URL di ogni documento sul sito:

- <host>
- <name>
- <format>
- <language>
- <product>
- <version>

HTML multi-pagine

```
<host><language><product><version><format><name>/index.html
```

Per esempio, http://docs.fedoraproject.org/en-US/Fedora/14/html/Accessibility_Guide/index.html

HTML singola-pagina

```
<host><language><product><version><format><name>/index.html
```

Per esempio, http://docs.fedoraproject.org/en-US/Fedora/14/html-single/Accessibility_Guide/index.html

PDF

```
<host><language><product><version><format><name><product>-<version>-
<name>-<language>.pdf
```

Per esempio, http://docs.fedoraproject.org/en-US/Fedora/14/pdf/Accessibility_Guide/Fedora-14-Accessibility_Guide-en-US.pdf

EPUB

```
<host><language><product><version><format><name><product>-<version>-
<name>-<language>.epub
```

Per esempio, http://docs.fedoraproject.org/en-US/Fedora/14/pdf/Accessibility_Guide/Fedora-14-Accessibility_Guide-en-US.epub

Notare che i campi <product_label>, <version_label> e <name_label> sono privi di significato per gli URL, anche quando sono esplicitamente rimossi dalla tabella dei contenuti con l'impostazione **UNUSED**.

DRAFT

Appendice E. File spec d'esempio per il pacchetto desktop-menu

Il seguente file spec è un esempio di come creare un pacchetto RPM contenente un file di *>desktop entry* (file **.directory**) ed un file di *desktop menu* (file **.menu**). Fare riferimento alla [Sezione 3.7.1.3, «Voci nel menu del desktop per i documenti»](#), per la struttura di questi file.

Questo esempio assume che si sia creato un file desktop-entry, **menu-example.directory** ed un file desktop-menu, **menu-example.menu**, con un file **README** localizzati in una cartella di nome **menu-example-0**, archiviata come **menu-example-0.tgz**.

Una volta creato, il pacchetto risultante avrà nome *menu-example*.

```
Name: menu-example
Version: 0
Release: 8%{?dist}.t1
Summary: Example of how to do a documentation menu package
Group: Development/Tools
License: GPLv2+
URL: http://engineering.redhat.com
Source0: %{name}-%{version}.tgz
BuildRoot: %[_tmppath]/%{name}-%{version}-%{release}-root-%(%{__id_u} -n)
BuildArch: noarch

%description
Example of how to do a documentation menu package

%prep
%setup -q

%build

%install
rm -rf %[_buildroot]
mkdir -p $RPM_BUILD_ROOT%[_datadir]/desktop-directories
mkdir -p $RPM_BUILD_ROOT/etc/xdg/menus/settings-merged

install -m644 menu-example.directory $RPM_BUILD_ROOT%[_datadir]/desktop-directories/menu-example.directory
install -m644 menu-example.menu $RPM_BUILD_ROOT%[_sysconfdir]/xdg/menus/settings-merged/menu-example.menu

%{_fixperms} $RPM_BUILD_ROOT/*

%clean
rm -rf %[_buildroot]

%files
%defattr(-,root,root,-)
%doc README
%[_datadir]/desktop-directories/menu-example.directory
%config(noreplace) %[_sysconfdir]/xdg/menus/settings-merged/menu-example.menu

%changelog
* Tue Nov 23 2010 Jeff Fearn <jfearn@redhat.com> 0-8
- Creation
```

DRAFT

Appendice F. Codici di lingua



Subtag di nazione

L'unica parte del tag linguistico XML, obbligatorio in **Publican**, è il *subtag linguistico*. Tuttavia, **Publican** è progettato con l'assunzione che l'identificazione delle lingue includa regolarmente anche i *subtag nazionali*. In molte lingue, ortografia e vocaboli variano significativamente da nazione a nazione. Se non si specifica la varietà nazionale di una lingua, in cui il documento viene redatto o in cui viene tradotto, si potrebbero ottenere risultati inaspettati, alla creazione del documento in **Publican**.



Altri codici linguistici

Il sistema di codici nello standard XML, usato per identificare le lingue, non è l'unico sistema di codici linguistici, attualmente in uso nel mondo. Tuttavia, poichè **Publican** si sforza di essere compatibile con lo standard XML, questi sono gli unici codici supportati da **Publican**. In particolare, notare che i codici usati nei prodotti GNU (notabili per l'uso del carattere trattino basso e del simbolo @ per separare gli element — per esempio, **en_GB** o **sr_RS@latin**), non sono compatibili con lo standard XML e perciò non funzionano con **Publican**.

Publican è uno strumento di pubblicazione basato su XML e perciò progettato per l'uso di codici linguistici — o *tag* — delineati dal W3C (World Wide Web Consortium)¹ nelle specifiche XML. Questi codici sono definiti nel documento *BCP 47: Tags for Identifying Languages*,² dell'IETF (Internet Engineering Task Force).

I tag linguistici sono creati a partire da uno o più *subtag*, separati da trattini. In ordine di presentazione all'interno di un tag linguistico, questi subtag sono:

lingua-script-regione-variante

Il BCP 47 permette anche, con l'uso di subtag *extension* e subtag *private-use*, di creare notevoli tag linguistici per casi speciali. Un subtag *extension* consente una regolazione calibrata di subtag esistenti, ma che occorre registrare presso l'IETF (attualmente non esiste nessun tag registrato). Un subtag *private-use* è preceduto da **x-** e non necessita di registrazione. Subtag *private-use* a parte, un subtag è valido se è presente nel registro dei subtag mantenuti dall'IETF, attraverso l'autorità IANA (Internet Assigned Numbers Authority).³ Sebbene **Publican** accetti ogni tag linguistico valido, secondo le regole stabilite nel BCP 47, esso è progettato con l'assunzione che i tag linguistici, per i documenti, assumano la forma ***lingua-nazione***. Di seguito si riporta una breve descrizione dei subtag:

subtag lingua

Il subtag linguistico è composto da due o più lettere minuscole ed è l'unica parte obbligatoria di ogni tag linguistico. Per le principali lingue parlate, il subtag linguistico è un codice di due lettere identico ai codici linguistici specificati nell'ISO 639-1,⁴ per esempio **it** (italiano), **hi** (hindi), **es** (spagnolo) ed **en** (inglese). Dove non esiste un codice di due lettere nell'ISO 639-1, il subtag

¹ <http://www.w3.org/TR/REC-xml/#sec-lang-tag>

² <http://tools.ietf.org/html/bcp47>

³ <http://www.iana.org/assignments/language-subtag-registry>

⁴ http://www.infoterm.info/standardization/iso_639_1_2002.php

linguistico solitamente è identico al codice specificato nell'ISO 639-2,⁵ per esempio **ba1** (baluchi: lingua iranica), **apk** (kiowa apache: lingua apache delle pianure) e **tpi** (Tok Pisin: lingua della Papua Nuova Guinea). Infine, un piccolo numero di subtag linguistici presenti nel registro presso l'IANA, sono privi di codici corrispondenti sia in ISO 639-1 sia in ISO 639-2, come i subtag per le lingue inventate **qya** (quenya: lingua elfica inventata da J.R.R.Tolkien) e **t1h** (klington: lingua extraterrestre della serie Star Trek), e per la lingua occulta **i-enochian** (enochiano: lingua degli angeli inventata da E.Kelley). Quest'ultimo esempio, mostra anche un ristretto numero di subtag linguistici *eccezionalmente* inseriti nel registro, senza corrispondere al modello delle due o tre lettere derivato dagli standard ISO 639.



Subtag estesi di lingua

Il documento *RFC 5646: Tags for Identifying Languages*⁶ pubblicato nel settembre del 2009, permette ai *subtag estesi di lingua* di aderire al subtag linguistico. I subtag estesi sono codici di tre lettere, rappresentanti lingue, che condividono una stretta relazione con una lingua già rappresentata da un subtag linguistico. Per esempio, **yue**, rappresentante la lingua cantonese, deve essere usato sempre con il subtag associato (cinese), quindi: **zh-yue**. L'IETF non riconosce l'*RFC 5646* come la "Miglior Regola d'Arte", nè tantomeno questi tag fanno già parte dello standard XML.

subtag script

Il subtag di script è composto da quattro lettere — la prima maiuscola le altre minuscole — e definisce un sistema di scrittura o alfabeto. Questi codici sono identici alle quattro lettere di codice specificati in ISO 15924.⁷ Il subtag di script è usato per identificare le lingue che comunemente usano più di un sistema di scrittura; il subtag viene ommesso quando non aggiunge alcun valore distintivo al tag linguistico globale. Per esempio, **sr-Latn** rappresenta la lingua serba scritta con l'alfabeto latino, mentre **sr-Cyr1** rappresenta ancora la lingua serba ma scritta con l'alfabeto cirillico; quindi **az-Arab** e **az-Cyr1** rappresentano la lingua azera (dell'Azerbaijani), scritta rispettivamente, in alfabeto arabo e cirillico. D'altro canto, l'italiano non ha bisogno di specificare **it-Latn** in quanto, comunemente nel mondo, l'italiano è scritto solo con l'alfabeto latino.

subtag regione

Il subtag di regione è composto da due lettere maiuscole (per le regioni che coincidono con i confini nazionali) o da tre cifre (per altre aree, come le regioni trans-nazionali). I tag di due lettere sono identici a quelli definiti in ISO 3166-1⁸, per esempio **IT** (Italia), **TZ** (Tanzania) e **VE** (Venezuela). I tag di tre cifre si basano su quelli definiti in UN M.49,⁹ per esempio, **015** (Nord Africa), **061** (Polinesia) e **419** (America latina e Caraibi).

subtag variante

I subtag di variante, composti da lettere maiuscole, minuscole e cifre, identificano varianti riconoscibili, ben definite di una lingua o di una scrittura. I subtag di variante che iniziano con una lettera devono essere lunghi almeno cinque caratteri, mentre quelli che iniziano con una cifra, lunghi almeno quattro caratteri. I principali subtag di variante possono essere usati solo in combinazione con subtag specifici oppure in combinazione di subtag. I subtag di variante non si armonizzano con gli altri standard; essi sono il risultato di una registrazione separata, presso l'IETF, da parte di una persona o gruppo interessato.

⁵ <http://www.loc.gov/standards/iso639-2/>

⁶ <http://tools.ietf.org/html/rfc5646>

⁷ <http://www.unicode.org/iso15924/>

⁸ http://www.iso.org/iso/country_codes.htm

⁹ <http://unstats.un.org/unsd/methods/m49/m49.htm>

Nello standard attuale, i dialetti di varie lingue sono designati con subtag di variante, per esempio, **nedis** denota un dialetto sloveno del Natisone o Nadiza. Questo subtag deve essere usato in congiunzione con il subtag della lingua slovena, quindi si ha **sl-nedis**. Nel settembre 2009, l'IETF ha pubblicato un RFC (Request for Comments) che tra le altre cose, propone di rappresentare i dialetti con i subtag di lingua estesa, da aggiungere ai subtag di lingua.¹⁰

I principali subtag di variante contrassegnano una particolare ortografia, la maggior parte usualmente dopo una riforma ufficiale di ortografia o dopo un significativo lavoro di documentazione sulla lingua. Esempi (con i relativi subtag di lingua) comprendono: **fr-1606nicot** (per la lingua francese come documentata da Jean Nicot nel 1606), **de-1901** (per la lingua tedesca la cui ortografia è stata codificata dal 2nd Orthographic Conference nel 1901) e **be-1959acad** (per la lingua bielorusca come codificata dall'Orthography Commission nel 1959).

Infine, alcuni tag di variante denotano una particolare variante di un sistema di scrittura o di traslitterazione. Per esempio, **zh-Latn-wadegile** rappresenta la lingua cinese scritta con l'alfabeto latino, in accordo la sistema di traslitterazione sviluppato da Thomas Wade ed Herbert Giles; **ja-Latn-hepburn** la lingua giapponese scritta con l'alfabeto latino, usando il sistema di traslitterazione di James Curtis Hepburn.

Publican include supporto per le seguenti lingue:

- ar-SA — arabo
- as-IN — assamese
- ast-ES — asturiano
- bg-BG — bulgaro
- bn-IN — bengalese
- bs-BA — bosniaco
- ca-ES — catalano
- cs-CZ — ceco
- da-DK — danese
- de-CH — tedesco (Svizzera)
- de-DE — tedesco (Germania)
- el-GR — greco
- es-ES — spagnolo
- fa-IR — iraniano
- fi-FI — finlandese
- fr-FR — francese
- gu-IN — gujarati

¹⁰ <http://tools.ietf.org/html/rfc5646>

- he-IL — ebraico
- hi-IN — hindi
- hr-HR — croato
- hu-HU — ungherese
- id-ID — indonesiano
- is-IS — islandese
- it-IT — italiano
- ja-JP — giapponese
- kn-IN — kannada
- ko-KR — coreano
- lv-LV — lettone
- ml-IN — malayalam
- mr-IN — marathi
- nb-NO — norvegese (ortografia Bokmål)
- nl-NL — olandese
- or-IN — oriya
- pa-IN — punjabi
- pl-PL — polacco
- pt-BR — portoghese (Brasile)
- pt-PT — portoghese (Portogallo)
- ru-RU — russo
- si-LK — singalese
- sk-SK — slovacco
- sr-Cyrl-RS — serbo (alfabeto cirillico)
- sr-Latn-RS — serbo (alfabeto latino)
- sv-SE — svedese
- ta-IN — tamil
- te-IN — telugu
- th-TH — thailandese o thai
- uk-UA — ucraino



- zh-CN — cinese (Repubblica Popolare Cinese, alfabeto Han semplificato)
- zh-TW — cinese (Repubblica di Cina, alfabeto Han tradizionale)



DRAFT

Appendice G. Cronologia revisioni

Revisione **Wed Dec 15 2010**
2.6-1

Rüdiger Landmann
r.landmann@redhat.com

Corretto i comandi d'esempio -- BZ#663211

Revisione **Wed Dec 1 2010**
2.4-1

Rüdiger Landmann
r.landmann@redhat.com

Incorporato correzioni in rilettura di Luigi Votta lewis41@fedoraproject.org BZ#657576
Documentato per lingue note, i problemi con PDF
Documentato il parametro `web_formats`
Documentato la creazione dei desktop-menu
Documentato `site_overrides.css`

Revisione **Mon Oct 25 2010**
2.3-0

Rüdiger Landmann
r.landmann@redhat.com

Documentato il file dump di sito web
Documentato il comando `bump`
Aggiornato l'informazione sulla larghezza immagini

Revisione **Tue Oct 5 2010**
2.3-0

Rüdiger Landmann
r.landmann@redhat.com

Aggiornato l'inclusione di più lingue in `lang_stats`
Corretto i dettagli su `web_logo.png` BZ#638153
Corretto l'elenco dei caratteri usabili nel nome di prodotto e nel titolo di documento
Documentato il nuovo parametro `web_type` e ricollocato i parametri `web_host` e `web_search` nel file di config di sito
Descritto i cataloghi OPDS
Documentato le pagine di prodotto e di versione
Documentato la presentazione delle pagine di man
Documentato il parametro `bridgehead_in_toc`
Correzione -- `def_langs` è un parametro del file config di sito, non un parametro del file di config di home page

Revisione **Thu Aug 19 2010**
2.2-0

Rüdiger Landmann
r.landmann@redhat.com

Espano gli esempi di codice BZ#604255
Chiarito l'uso di `clean_ids` BZ#612819
Documentato `--novalid` BZ#616142

Revisione **Fri Jul 16 2010**
2.1-1

Rüdiger Landmann
r.landmann@redhat.com

Corretto e chiarificato le istruzioni su sito web BZ#614259
Chiarificato l'uso di `Product-Version-Id` per i pacchetti

- Revisione 1.6-1** **Mon May 24 2010** **Rüdiger Landmann**
r.landmann@redhat.com
 Aggiornato le istruzioni di installazioni per Ubuntu
- Revisione 1.6-0** **Fri May 7 2010** **Rüdiger Landmann**
r.landmann@redhat.com
 Rivisto la nomenclatura di azioni ed opzioni
 Documentato le azioni print_known, print_banned, e print_unused
 Corretto ed espanso la documentazione su installazione di brand
 Documentato i parametri max_image_width e confidential_text
 Documentato il formato dei plugin di aiuto di Eclipse e i parametri supportati
- Revisione 1.5-0** **Fri Feb 26 2010** **Rüdiger Landmann**
r.landmann@redhat.com
 Documentato le opzioni di --config
- Revisione 1.4-0** **Wed Feb 17 2010** **Jeff Fearn** jfearn@redhat.com
 Rimosso obsoleto riferimento al path al file di catalogo di DocBook BZ#565498
 Documentato le opzioni di CVS
- Revisione 1.3-0** **Mon Dec 7 2009** **Rüdiger Landmann**
r.landmann@redhat.com
 Aggiunto una FAQ sulla evidenziazione degli errori nel codice
 Aggiunto una sezione sui formati validi
 Aggiornato l'elenco degli autori
 Aggiunto istruzioni di installazione per Debian; istruzioni di installazione più specifiche per Ubuntu BZ#542711
 Metadati nel file Book_Info.xml
- Revisione 1.2-0** **Fri Nov 27 2009** **Jeff Fearn** jfearn@redhat.com
 Documentato l'azione lang_stats BZ#540696
- Revisione 1.1-1** **Thu Nov 26 2009** **Jeff Fearn** jfearn@redhat.com
 Risolto le condizioni d'uso di errati doc BZ#540691
- Revisione 1.1-0** **Thu Oct 22 2009** **Rüdiger Landmann**
r.landmann@redhat.com
 Risolto varie piccole inconsistenze e pulizia generale

Revisione **Tue Oct 13 2009**
1.0-0

Aggiornato per Publican 1.0

Rüdiger Landmann
[*r.landmann@redhat.com*](mailto:r.landmann@redhat.com)

Revisione **Thu Dec 18 2008**
0.5-0

Aggiunto appendice per i parametri di Makefile
Aggiunto una FAQ sullo spazio di heap in Java

Jeff Fearn [*jfearn@redhat.com*](mailto:jfearn@redhat.com)

Revisione **Tue Nov 25 2008**
0.4-0

Aggiunto la sezione "Pre-release and draft documentation"

Brian Forté [*bforte@redhat.com*](mailto:bforte@redhat.com)

Revisione **Fri Oct 10 2008**
0.3-0

Aggiunto la sezione "Conditional Tagging"

Don Domingo [*ddomingo@redhat.com*](mailto:ddomingo@redhat.com)

Revisione **Fri Sep 05 2008**
0.2-0

Modifiche generali e aggiornato alla release Publican 0.36. Aggiunto anche una nuova sezione al capitolo 3.3.

Brian Forté [*bforte@redhat.com*](mailto:bforte@redhat.com)

Revisione **Fri Jun 06 2008**
0.1-1

Aggiornato il capitolo Branding con una nota sull'aggiunta dei brand oVirt e GIMP

Murray McAllister [*mmcallis@redhat.com*](mailto:mmcallis@redhat.com)

Revisione **Fri May 16 2008**
0.1-0

Aggiornato FAQ

Jeff Fearn [*jfearn@redhat.com*](mailto:jfearn@redhat.com)

Revisione **Thu Dec 13 2007**
0.0-0

Contenuto iniziale alla release

Murray McAllister [*mmcallis@redhat.com*](mailto:mmcallis@redhat.com)

DRAFT