Come sviluppare un widget complesso (3/3)

Siamo giunti all'ultimo tutorial di questa serie, nel quale ti mostrerò la procedura grazie alla quale é possibile **tradurre in diverse lingue un plugin o un tema di WordPress.**

Avrai certamente già notato che nelle cartelle di linguaggio troviamo dei files con estensione .mo (Machine Object, il file di linguaggio compilato) e .po (Portable Object, il sorgente editabile). Questi file fanno capo al sistema GNU Gettext che é il sistema scelto da WordPress per implementare le traduzioni.

Vedremo più tardi come creare questi files; ora partiamo con le funzioni di base.

Quali sono le funzioni di base?

__() (doppio underscore)

Questa funzione passa due argomenti:

- Il testo da tradurre
- Il domain (una stringa univoca che fa riferimento al file di linguaggio che svilupperemo in seguito). Se il secondo parametro viene omesso, significa che vogliamo utilizzare il file di linguaggio del core di WordPress.

La funzione ritorna la traduzione nel linguaggio impostato su WordPress.

Se dovesse presentarsi qualunque problema (il file di linguaggio non esiste, il file di linguaggio esiste ma non viene trovata una corrispondenza, ….), la funzione ritorna comunque quanto scritto nel primo argomento.

_e()

Esattamente come la funzione appena vista, ma invece di ritornare la traduzione la stampa a video. In pratica é un'abbreviazione di

echo __()

Come predisporre il plugin per la traduzione

Inizia ora con l'individuare le espressioni da tradurre. Utilizzeremo come dominio *pp-textdomain*.

Nel costruttore rendi traducibile la descrizione in questo modo:

Per convenzione, la lingua di default é l'inglese. Nessuno ci vieta di scriverlo in italiano e poi tradurlo in inglese. Ma io sono un cultore delle convenzioni e degli standard.

Come vedi ho utilizzato la funzione __(), in quanto dovevo valorizzare una variabile.
Il discorso cambia quando dobbiamo predisporre per la traduzione le label del form

In questo caso serve stampare a video, quindi userai la funzione _e().

Fai ora la stessa cosa per tutti i campi del form

Come indicare a WordPress dove deve cercare i files di linguaggio?

A questo punto dovrai dire a WordPress dove si trovano i files di linguaggio del widget.
Per farlo utilizzeremo la funzione load_plugin_textdomain() che passa tre parametri:

- il dominio (nel nostro caso pp-textdomain)
- percorso assoluto (deprecato, quindi scriveremo false)
- percorso relativo della cartella del linguaggio

Dichiara quindi questa funzione all'inizio del costruttore di classe, in questo modo:

Ed ora crea la cartella *languages* all'interno della cartella del widget.

Ora puoi provare il widget che verrà mostrato in inglese. Questo in quanto se hai WordPress impostato in italiano, verrà cercato il file di linguaggio italiano. Ma visto che non viene trovato, come detto, mostra il default.

Prova ora a togliere *pp-textdomain* dall'espressione *Title* nel form e ricarica.

Se hai WordPress impostato in italiano, *Title* verrà tradotto con *Titolo*, perché?

Come ho detto, se non passo nessun dominio, WordPress andrà a cercare le corrispondenze nel file di linguaggio del core, il quale comprende la traduzione del termine *Title*.

Perché ti faccio notare questa cosa?

Perché se il tuo plugin non contiene termini o frasi troppo specifiche, potrebbe non essere necessario creare un file di linguaggio personalizzato. Parole come *Search, Title, Save*, ecc...

fanno sicuramente parte della libreria di base (assieme a tantissime altre parole o frasi).

Se riesci a tradurre il tuo plugin utilizzando il file di linguaggio del core avrai un grande ed immediato vantaggio (oltre a non dover fare la traduzione): **il tuo plugin é tradotto in tutte le lingue del mondo.**

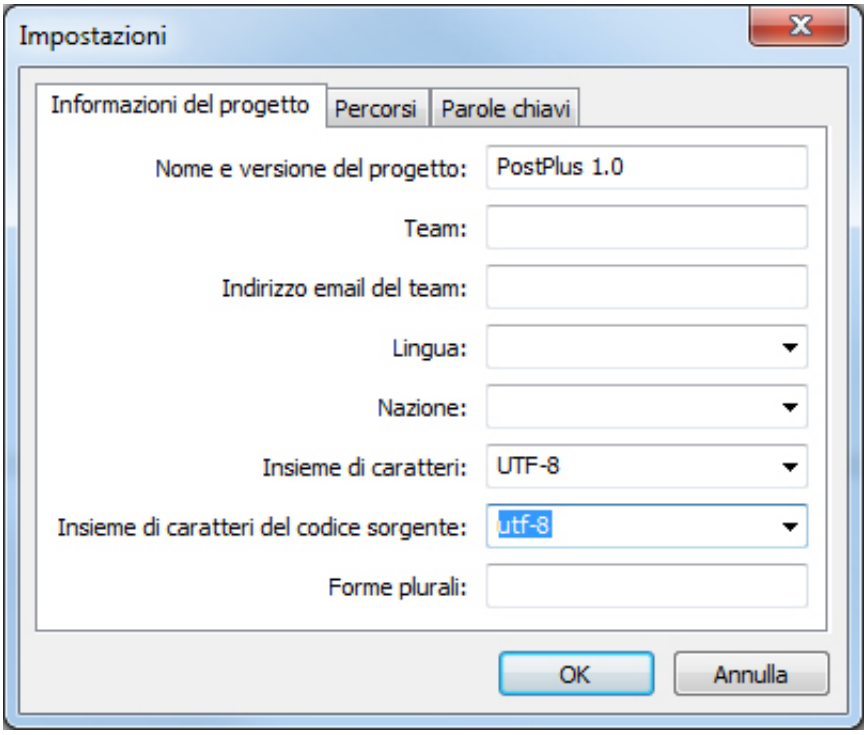
Roba da farci qualche compromesso eventualmente!

Comunque: ora tutto é pronto. Traduciamo.

Come creare i files di linguaggio

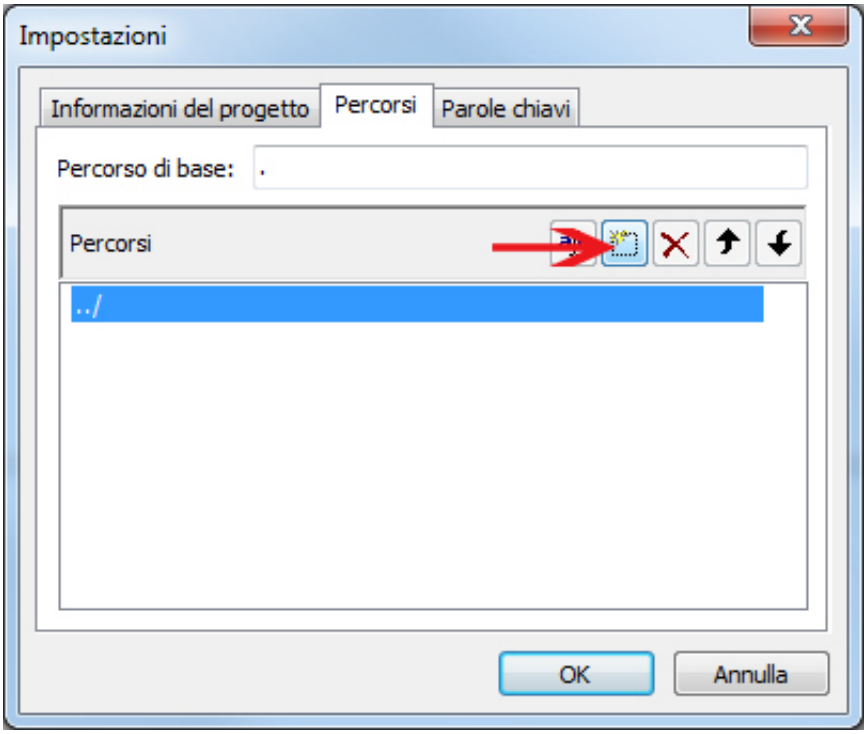
Come prima cosa avrai bisogno di un editor. Ti consiglio poedit che tra l'altro gira su tutte le piattaforme. Sacricalo ed installalo.

Ora apri poedit e vai su *file -> nuovo catalogo*. Inserisci il nome del progetto e se lo desideri altri dati.



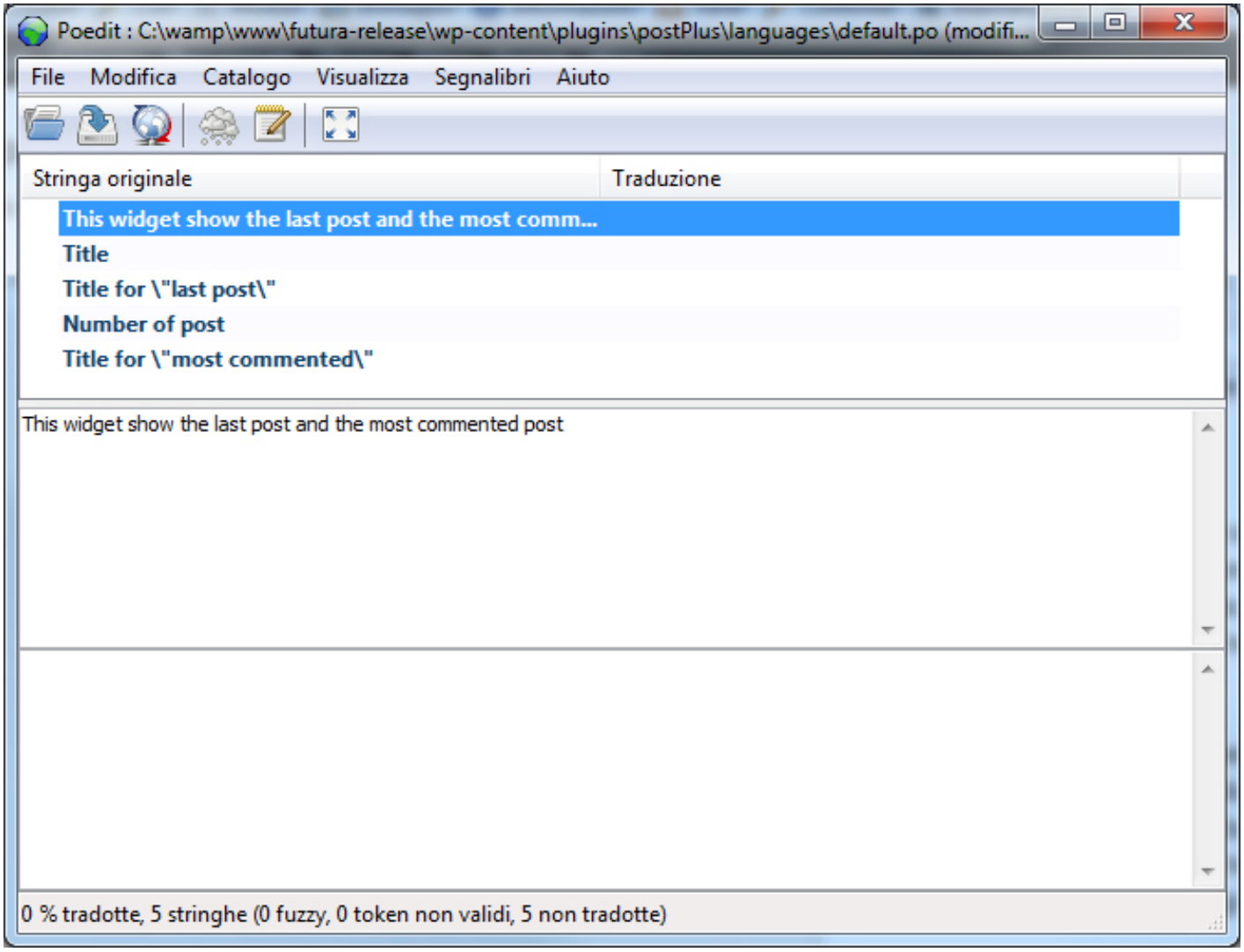
Quindi portati sulla scheda *percorsi*. Dobbiamo indicare dove si trovano i files da tradurre rispetto alla posizione del file che stiamo creando. Questo file lo salveremo nella cartella *languages*, quindi la posizione dei files da tradurre sarà ../

Clicca dunque su "*nuovo oggetto*", inserisci ../ e premi su enter.



Passa ora alla scheda *“parole chiavi”* nella quale dobbiamo indicare le funzioni che abbiamo utilizzato per le traduzioni, in modo tale che poedit andrà **automanticamente a scovare le espressioni da tradurre all’interno del codice.**

Clicca dunque su *“nuovo oggetto”* ed aggiungi __ . Ed in seguito clicca nuovamente su *“nuovo oggetto”* ed aggiungi _e.



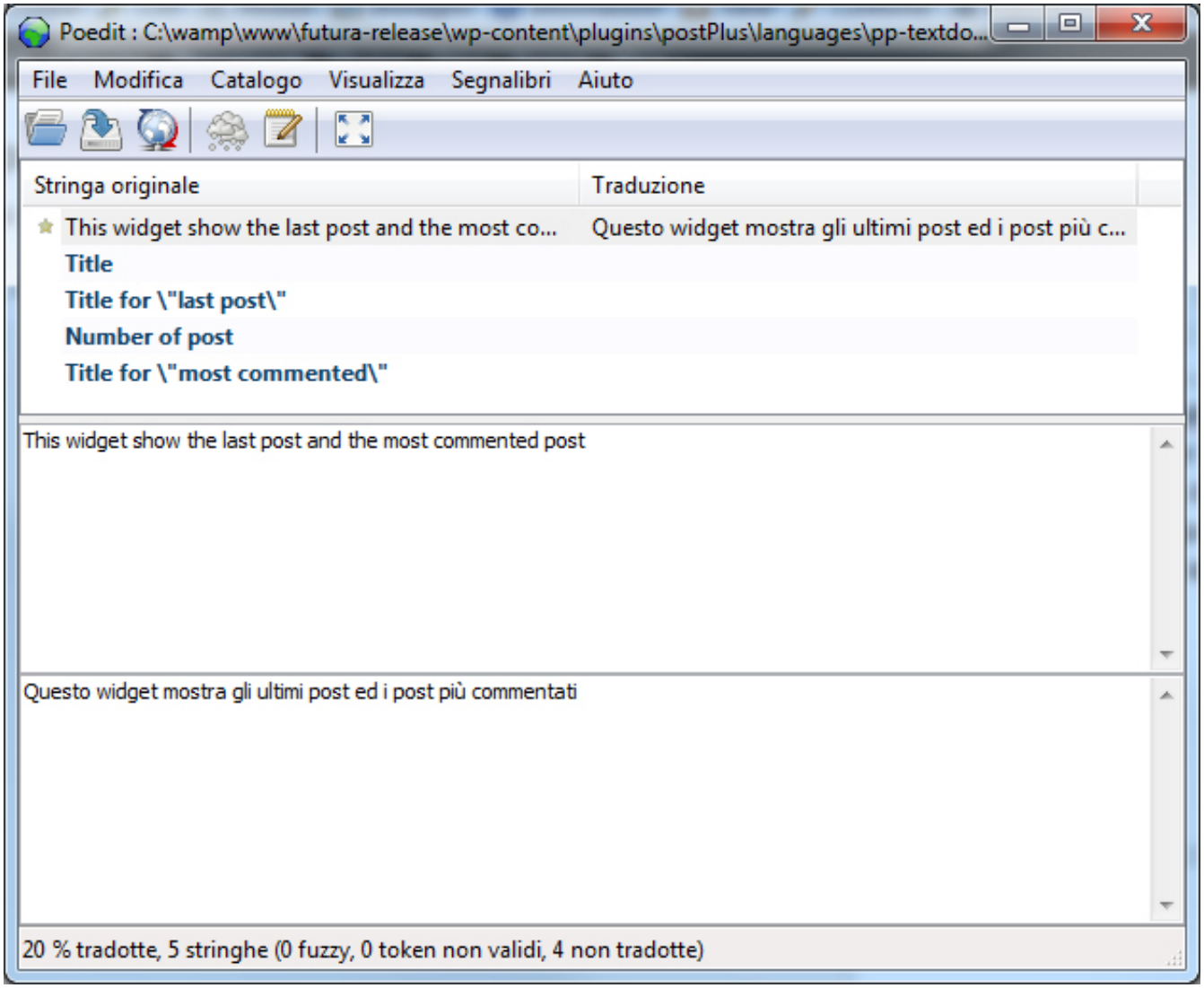
Premi su ok. Ti viene richiesto di salvare il file *default.po*. Posizionati nella cartella *languages* e salvalo con il seguente nome:

pp-textdomain-it_IT.po

Come vedi i file di linguaggio vanno salvati in questo formato:

dominio-sigla del linguaggio.po

A questo punto le frasi da tradurre vengono importate automaticamente nel’editor
Non ti resta che selezionare la frase che vuoi tradutte e scrivere la traduzione nell’area di testo in basso.



Quando hai finito premi su salva catalogo. Finito. Il tuo widget é tradotto in italiano, e con qualsiasi altra lingua impostata, mostrerà la versione inglese (default).

Conclusione

In questo articolo abbiamo visto la procedura per localizzare un plugin di wordpress. Siamo così giunti alla fine di questa serie di articoli dove ho mostrato alcuni strumenti avanzati nello sviluppo di plugins e widget. Sono tra l'altro **strumenti applicabili anche nello sviluppo di temi**; nel file *functions.php* di un tema possiamo infatti implementare funzionalità nello stesso modo che abbiamo visto fino ad ora. Anzi, magari in futuro vedremo anche come sviluppare un pannello opzioni per un tema. Ti interessa?