

---

# **multilingual-ng to hvad guide Documentation**

*Release 1*

**Benjamin Wohlwend**

September 21, 2012



# CONTENTS

<b>1</b>	<b>Indices and tables</b>	<b>3</b>
----------	---------------------------	----------



Contents:



# MODELS.PY

multilingual-ng and django-hvad share the same database schema. This means that converting a project from ng to hvad should generally be possible without schema migrations. A ng model can be converted into a hvad model like this:

## 1.1 multilingual-ng

```
class MyModel(models.Model):
    is_published = models.BooleanField()

    class Translation(multilingual.Translation):
        name = models.CharField(max_length=100)
        slug = models.SlugField(max_length=100)

    def __unicode__(self):
        return self.name
```

## 1.2 hvad

```
from nani.models import TranslatableModel, TranslatedFields

class MyModel(TranslatableModel):
    is_published = models.BooleanField()

    translations = TranslatedFields(
        name = models.CharField(max_length=100),
        slug = models.SlugField(max_length=100),
    )

    def __unicode__(self):
        return self.safe_translation_getter('name', unicode(self.pk))
```

### 1.2.1 Notes

- a field is only accessible if it is translated in the current language (inefficient fallback methods notwithstanding). You should never try to access a translated field directly in `unicode()`. Instead, use `self.safe_translation_getter()`.

- Don't try to mix hvad models with django-polymorphic, GeoDjango, django-mptt or generally model subclasses that depend on their own `Manager` and `QuerySet` implementations. You will fail. Hard. Really. Don't do it. Honestly.



# VIEWS.PY

This is probably where you have to change the most code

## 2.1 Ordering on translated fields

If you want to order on translated fields, you have to use the `using_translations()` on the manager:

```
ordered_by_name = MyModel.objects.using_translations().order_by('name')
```

## 2.2 Querying related fields on non-translated models

Given this model:

```
class Hobbies(TranslatedModel):
    user = models.ForeignKey('auth.User')

    translations = TranslatedField(
        name = models.CharField(max_length=20),
    )
```

If you want to have a translated list of hobbies from user a, you need to use `get_translation_aware_manager()`:

```
hobbies =
```



# ADMIN

The general transformation of an ng-Admin to a hvad-Admin:

mutlingual-ng:

```
class MyModelInline(multilingual.admin.MultilingualInlineAdmin):
    model = MyInlineModel
    extra = 3

class MyModelAdmin(multilingual.admin.MultilingualInlineAdmin)
    list_display = ('title', 'is_published')
    inlines = [
        MyModelInline,
    ]

    search_fields = [
        'translations__title',
        'translations__destination',
        'translations__lead_text',
    ]

    use_fieldsets = (
        (_("Common"), {
            'fields': (('is_published',),
        })),
        (_("Language dependent"), {
            'fields': ('title', 'description',),
        })),
    )
```

hvad:

```
class MyModelInline(TranslatableTabularInline):
    model = MyInlineModel
    extra = 3
    ordering = None # necessary because of a bug in hvad, might be resolved later on

class MyModelAdmin(TranslatableAdmin)
    list_display = ('title', 'is_published')
    inlines = [
        MyModelInline,
    ]

    use_fieldsets = (
```

```
(_("Common"), {
    'fields': (('is_published',)
}),
(_("Language dependent"), {
    'fields': ('title', 'description',),
}),
)

def get_fieldsets(self, request, obj=None):
    return self.use_fieldsets
```

## 3.1 Notable changes

### 3.1.1 queryset method

the ng-inline might have a queryset method like this:

```
def queryset(self, request):
    return super(multilingual.admin.MultilingualInlineAdmin, self).queryset(request)
```

This is not necessary anymore with hvad.

### 3.1.2 get\_fieldsets method

Django validates the `fieldsets` attribute, and doesn't detect the translated fields. To circumvent the validation, we use the `get_fieldsets()` method instead.

### 3.1.3 ordering = None on inline admin class

This is necessary because of a bug in hvad which might be resolved at a further stage