# Concept Document v1.0

## www.hornget.net

### Overview

Based on the discussion previously within the HornGet Google group I have compiled this document to target HornGet moving to a server based platform.

This document contains my ideas regarding implementation and UX design for [www.hornget.net](http://www.hornget.net) to allow visitors to easily browse through our supported OSS packages, based on the categories and descriptors found within our package DSL, and download their desired OSS packages, pre-built in zip file format.

### Concept

The website must automatically update itself with the most recent built/added packages to ensure that the content and packages are always up to date, and that there is minimum maintenance required on the part of the HornGet development team to allow for their time to be focused on the development of the HornGet core services.

I believe this can be achieved very quickly and easily based on the following assumptions.

1. The HornGet core services will run automatically as windows (or other) service regularly.
2. The ability to add a step to this service, or have another service running on the server to generate the web sites package data.
3. Both the website and core services will run on the same server.

### Implementation of dynamic XML data for website

I would like to implement the data generation for the website in a flat file XML format under 2 folders.

* Categories
* Packages

The generated package XML data would be re-created each time during the packaging process of the core services.

The categories XML data files would be created once, if they did not exist already, and then never re-created dynamically.

The reason that the categories XML should only be created once is to allow us to manually adjust the XML to add in html descriptions for rendering on the front end – I cannot currently see any easy way to add category descriptions to the current horn infrastructure so I believe this to be the easiest way to achieve the desired outcome.

Each generated file name will conform to a convention to allow direct routing based on expression to calculate which file to load and render to the user.

**For example:**

*If the URL was* [*www.hornget.net/packages*](http://www.hornget.net/packages) *-*   
all of the categories XML data would be loaded and a list of all available categories would be displayed to the user.

*If the URL was* [*www.hornget.net/packages/orm*](http://www.hornget.net/packages/orm) *-*   
the details for the ORM category would be loaded, and a list of all available packages would be loaded and displayed to the user, based on the convention:

Get all packages XML data files whose file name starts with “orm\_”, i.e. the URL of the category.

*And finally, if the URL was* [*www.hornget.net/packages/orm/nhibernate-2-1*](http://www.hornget.net/packages/orm/nhibernate-2-1) *-*  
 the details of the package would be loaded and displayed to the user. E.G. orm\_nhibernate-2-1.xml

### Category XML data structure

**URL Structure**: [www.hornget.net/packages/orm](http://www.hornget.net/packages/orm)  
**XML File name**: orm.xml  
  
<category>  
 <name>ORM</name>  
 <url>orm</url>  
 <description>Object-relational mapping (ORM, O/RM, and O/R mapping)....</description> <image>images/orm-image.jpg</image>  
</category>

### Package XML data structure

**URL Structure**: [www.hornget.net/packages/orm/nhibernate-2-1](http://www.hornget.net/packages/orm/nhibernate-2-1)  
**XML File name**: orm\_nhibernate-2-1.xml  
  
<package>  
 <category>ORM</category>  
 <name>NHibernate</name>  
 <url>orm/nhibernate-2-1</url>   
 <description>NHibernate handles persisting plain ....</description>  
 <notes>Trunk Build</notes>  
 <forum>http://groups.google.co.uk/group/nhusers?hl=en</forum>  
 <homepage></homepage>  
 <version>1.0.0.0</version>  
 <created>03/01/2009 17:55</created>   
 <filename>nhibernate.zip</filename>  
 <contents>  
 <file>ABC.hbm.xml</file>  
 <file>antlr.runtime.dll</file>  
 <file>Antlr3.Runtime.dll</file>  
 <file>Castle.Core.dll</file>  
 <file>Castle.Core.pdb</file>  
 <file>Castle.Core.xml</file>  
 ....  
 </contents>  
</package>

### HornGet Logo

I in no way class myself as a graphic designer ☺ but in the spirit of things I through together a quick logo for feedback.

I have no idea what the HornGet colour scheme is as we have yet to discuss it, but here goes.



Obviously this will change to adapt to the web design.