

A Short Comparison on .NET Open-Source Portals

Cătălin MAICAN, Radu LIXĂNDROIU
Transilvania University Of Braşov

This paper tries to present two of the most important open source portals based on the .NET technology. The first part makes a short presentation of the open source current, while the second part presents the “commandments” of the open source taking into account Windows-based open source software. The third part presents the main features of a portal. These features are further detailed for the two main portals based on the ASP.NET technology in the forth and the fifth part of the paper.

Keywords: open source, portal, DotNetNuke, Passage, comparison.

Why open-source portals?

It seems that making the argument for open-source software becomes easier every year. This is mainly due to open-source products such as Linux, MySQL, or the Apache Web Server (just to name a few), which have been very successful. Furthermore, quality software organizations such as the Apache Software Foundation boast a history of consistent, stable, and successful software deliveries.

Many open-source projects are commercially developed code bases turned over to the open market as a business decision, because they no longer have to pay to maintain the code.

We cannot forget the best part about open-source software: it is free. Although, we should note that *open source* and *free* are not synonymous. Just because most open-source products are free, it doesn't mean that they are all free. All open source means that you can receive the source in order to identify bugs [1].

The greatest thing about capable open-source software is that it solves the classic “build versus buy” engineering dilemma. Software had become so complex that it was simply not practical to choose to build anything. Now, with reliable open-source alternatives, developers can start with the open-source products and build the features sets needed.

Another advantage might be that open-source tools lower barriers to a developer's entry into the market. With open-source products, developers can become proficient with emerging tools and technologies, cultivating an ability to build solutions with them, rather

than be tied to a purchasing decision.

Building a portal is about integrating an organization with all its business processes, people and locations, this being a procedure that one cannot take lightly or accomplish easily. Business processes are complex because of the competitive forces and the ever-shifting variety of people associated with them. Building a portal enables information movement through the organization. Therefore, it is better to invest in a person, rather than a product that will be outdated in a year.

Does .NET /Windows open-source exist?

There is no doubt that the "Open Source" term coined by Eric Raymond, founder of the Open Source Initiative, has been subjected to a lot of interpretation in recent years. Since the term is descriptive, it could not be protected by a trademark; therefore, companies and organizations are free to leverage the term in their marketing campaigns to promote their own interests. As a result, "Open Source" has started to lose some of its idealist values. These values are best summed up by the Open Source Definition's 10 Commandments [2]:

1. Free Redistribution
2. Source Code
3. Derived Works
4. Integrity of The Author's Source Code
5. No Discrimination Against Persons or Groups
6. No Discrimination Against Fields of Endeavor
7. Distribution of License
8. License Must Not Be Specific to a Product

9. License Must Not Restrict Other Software

10. License Must Be Technology-Neutral

There are Open Source zealots who believe that unless an application is part of a stack which includes 100% Open Source services and components, it can not claim to be Open Source. This "stack" typically includes an Open Source operating system at its foundation (ie. Linux), an Open Source web server (ie. Apache), an Open Source database (ie. mySQL), and an Open Source application layer (ie. PHP, Perl, Python) or LAMP.

But, what happens with projects like DotNetNuke because it is not part of a fully Open Source stack (at this point in time DotNetNuke, Passage Portal, Rainbow or other .NET portals with an Open Source license run on ASP.NET, a services layer which is only available for the Windows platform - a situation which the Mono project is trying to address).

Does this "stack" argument actually make any sense? In the true sense of the OSD, it certainly does not. Each application is supposed to be judged independently based on its own licensing scheme. And there is no specification as to someone being restricted from combining Open Source components with proprietary ones to produce a comprehensive solution.

Portal features

According to [4], the main portal features are:

- **Portal Core Services** - Provides core portal application functions. These functions are the common services (such as user interface design and search) that are utilized by other components. These services can be supported using a 3rd party tool.
- **Content Services** - Content components are Information (knowledge) from internal sources, external feeds, databases, etc. 3rd party tools or application service providers (ASPs) can provide this component.
- **Portal Interface Framework** - The interface framework provides a structure or container for other applications (or portlets/modules). The framework allows new applications to follow a consistent visual interface and programming standard.
- **Collaboration Services** - Collaboration components are those that facilitate employees working together. Many components facilitate virtual meetings through application sharing, community chat rooms or videoconferencing.
- **Integration Services** - Integration components are those that link existing services to the portal. These components provide the application and transactional linkage to enterprise systems (such as email or corporate directories). Adapters can facilitate this integration, but integration is usually developed or customized on the basis of requirements.

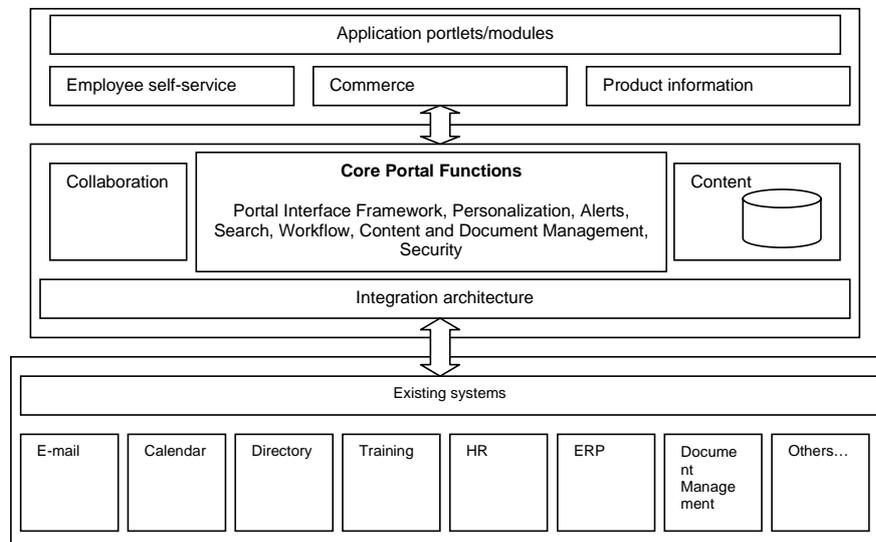


Figure 1. Portal features

The portal platform services provide the structure and core functions for an organizational portal. Without these services, the portal is merely a static web site. Value added services such as alerts and preferences can

provide the user with convenience and contact. Although many packages provide portal platform services, it is critical that the engine provides adapters to extend the platform as requirements change.

Table 1: A short description of core portal features

Component	Description
Portal interface framework	Provides a container for application portlets/modules. A framework may be used to make the user interface across portals
Personalization/ Filters	Applies user-defined filters to content to refine what information is presented. Identifies and categorizes content to improve personalization
Alerts	Automatic notification of changes, additions of preferences within the portal. This includes the possibility to search the content across multiple data sources
Search	Infrastructure component that indexes and organizes links to portal content
Workflow	Move assets and information around the organization based on business rules and relations
Document Management / Content Management	Manages the creation, editing, publishing, access and archiving of business and technical content/documents. The Document management should be integrated with content management for providing easier access to content creation procedures. The Content Management component should contain access to a content repository and to external content
Reporting	Real-time monitoring and reporting of business activity
Security	Provides the proper level of access security and permission structure.

Collaboration Services allow users to communicate and solve business problems irrespective of location, space and time. As organizations become more geographically diverse, alternate forms of communication are

required. Because face-to-face communication is not always possible, collaboration applications such as application sharing, whiteboard, video-conferencing and chat can enhance the standard voice communication.

Table 2: Collaboration services within a portal

Component	Description
Communication / messaging	Tools used by users for communicating between them. Includes applications such as web-mail, discussion forums, chat and instant messaging.
Application sharing	Enable multiple parties to share and collaborate on applications at a distance. Includes web-based meeting facilities.
Community building	Creates virtual community for users with similar interests and needs.

Integration Architecture provides the infrastructure for existing systems integration. In order to integrate, it is imperative to buy/develop adapters for existing applica-

tions. The following picture shows the most common systems that may be needed to be integrated.

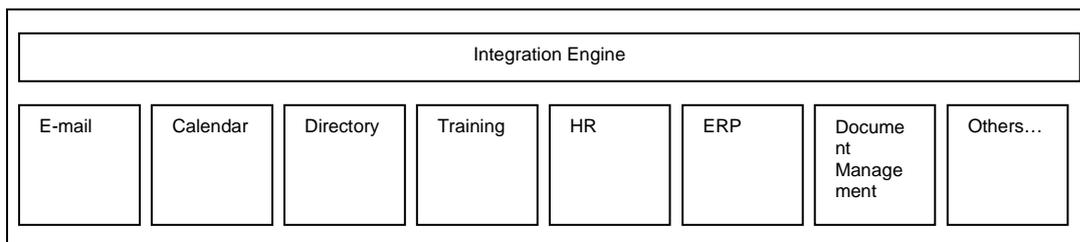


Figure 2: Integration services through a portal

Content services provide the infrastructure for hosting web and document based-content. It is the single database of information that is

populated, integrated and synchronized with internal and external sources. In many cases, the content repository will be a component of

the portal engine that allows the organization to extend the data structure and/or integrate information for external source to provide context sensitive information.

DotNetNuke

“In DotNetNuke, a portal is one site, of *n* possible sites, in a single instance of DNN, with data and functionality completely discreet from other portals in the same instance. For the purposes of this discussion, a portal can be defined as the related data for one web

site hosted within your DotNetNuke installation. The application natively provides the capability to host multiple web sites from the same code base, each containing different information and presented to the user at runtime based on the URL the user enters to access the code base.” [5]

“The ultimate goal for DNN is to create a minimal core of portal services which is easily extendable.”[6]

Component	Description
Portal interface	<p>DotNetNuke support security at page and module levels enabling each area in the portal to be delegated to a department and locked for other users.</p> <p>Regarding the URL, DotNetNuke permits “Friendly URLs” that are still composed of page name and the ID of that page.</p> <p>A single installation of DotNetNuke permits virtual portals based on the DNS name for the “parent portals” and a virtual folder for the “child portals”</p> <p>DotNetNuke contains a “Skin engine” that enables various themes for each page of the portal. The skin is applied at Page level while for module a Container may be used. Regular users cannot change or customize the page. “Skin” installation is enabled at “host” level for portals. Once a “skin” is created it must be uploaded and parsed by the DotNetNuke before being used. Portal administrators can choose skins and containers from a preview window for the whole site, while at the page level only the name of the skin/container is available.</p>
Personalization/ Filters	<p>There is no rule-based or filtering personalization for users. There is though a simple Personalization API that permits saving/retrieving values from the personalization store. Customization is not available for users in the default package. Although, a commercial module enables a “My Page” feature.</p>
Alerts	<p>Not available in the default package.</p>
Search	<p>DotNetNuke uses its own search engine.</p> <p>This function uses the role-based security for accessing the content. All the modules that offer information to be searched need to implement an interface that goes through the records to be included in the search. The search engine in also integrated with a RSS function which pulls information from the search engine.</p> <p>Other search APIs can be used like Lucene for searching and indexing the site, but they are available only in commercial versions produced by other organizations.</p>
Workflow	<p>DotNetNuke does not contain a Workflow engine by its own. A free add-on is available</p>
Document Management / Content Management	<p>DotNetNuke includes in the default installation functionalities such as asset management, content scheduling, content staging.</p>
Reporting	<p>A limited audit trail function is included. It provides auditing for login/logout, exceptions.</p> <p>Also, a Report module is available that can connect to various data-sources for obtaining data and displaying it in a table-based format.</p>
Security	<p>Security is based on providers, meaning that developers can create other authentication and authorization components without modifying the original source-code.</p> <ul style="list-style-type: none"> • Authentication packages offered with the original source-code: LDAP, NTLM. A free plug-in is available for Kerberos • Authentication is only role-based - access to content is granted to users that belong on certain roles. There is no possibility to create a role hierarchy. Any number of roles can be created within a portal; these roles may belong to groups of roles. • Granular permissions are granted to different roles that access modules within pages. There are two default “permissions”, namely “View” and “Edit”; new

Component	Description
	<p>permissions can be created programmatically.</p> <ul style="list-style-type: none"> • User profiles are extensible on portal basis. • Others: SSL support, e-mail verification for the first-time access, captcha.
Communication / messaging	DotNetNuke has a number of projects having various communication features: Forums, Wiki
Application sharing	DotNetNuke uses reusable components. It supports a wide variety of web component but the preferred type of component is the ASP.NET user control.
Community building	Forum module
E-mail Adapter	Not available in the default package. A web-mail add-on is available with a commercial license
Calendar Adapter	DotNetNuke includes a Calendar module that may be used for group (role) level event scheduling using a user-configurable repetitive scheme. Users that are in roles with "Edit" permission can create repetitive events. Users can enroll to events and content editors can moderate the events and enrollees.
Remote learning Adapter	Not available
Content Repository	A Repository Module is available with the default DotNetNuke install. Administrators may define Categories, Attributes and Attribute values for those categories. This module further enhances portal permissions by permitting selections for Moderation, Upload, Download, Rating and Comments. As a negative aspect, this module is not integrated with other modules – other modules cannot use the categories and attributes created within this module
External Content	External Content may be accessed using web services and, for news, RSS. Also, DotNetNuke can push content to various applications using Web Services.
Performance	DotNetNuke includes support for caching, load balancing, web-farm.

Passage Portal

Passage Portal provides content management, enterprise portal, and digital dashboard functionality in a single, easy to install and cus-

tomize web server solution. Passage has been in use in a variety of industries and countries since 2001.

Component	Description
Portal interface	<p>A single installation of Passage Portal permits multiple virtual portals only in the Enterprise Edition (not Open Source)</p> <p>Regarding the URL, Passage permits "Human Friendly URLs" by using Page Name Aliases. By default each page is accessed by using an ID composed of a Guid value. By using aliases the pages can be accessed by its alias (e.g. alias home, access home.aspx) instead of view.aspx?id=[guid]</p> <p>Passage Portal is based on the Composite Design Pattern and uses Object-Relational data model for managing hierarchies of objects. From a metadata perspective, Passage applications are simply trees of entities and attributes. Portal objects are typically managed logically within the hierarchy so that a Portal is the parent of several Page objects and each page has several Web Part objects.</p> <p>Passage Portal uses a simple theme engine which enables the administrator to select different CSS styles for the site. Page Templates can be used for creating several pages that share common look and feel characteristics.</p>
Personalization/ Filters	<p>There is no rule-based personalization for users. Filtering can be achieved using the Segmentation feature that permits creating hierarchical rules for content. Segments are similar to roles in that they may be used to conditionally grant access or display content based on segmentation rules. For example, a user in segment "Platinum Customer" may be delivered different content or ads than a user in the "New Customer" segment. In addition, Dialog Rules may be used for separating the content based on Segment rules and return any number of types of objects to be used by the calling object. Examples of using dialogue rules include dynamically displaying content in a web page or email, or routing a web form on submit.</p>
Alerts	Passage Portal uses an Alert system

Component	Description
Search	The search engine within Passage Portal is based on Best bets which intercepts common search keywords, misspellings, or other variations of search terms and map them to search results. The search engine may be integrated with database full-text indexing for obtaining better results. External, public web site built using Passage can be evaluated by a Search Engine Optimizer that analyzes all web pages and suggest fundamental SEO best practices.
Workflow	A two-phase workflow with offline sandbox is available by default.
Document Management / Content Management	Passage provides an array of content management features to make any type of content easily accessible to customers: web-based page creation and editing, web forms and surveys, image gallery.
Reporting	An audit trail function is included. It provides auditing for page views, exceptions and security. The Events are detailed on user and portal-object basis. A Chart Web Part is available; it can connect to various data-sources for obtaining data and displaying it in a graphical format (bar, line, area, pie).
Security	The portal role based security model allows administrators to associate up to 5 permissions with any portal object, such as pages and web part components. The native properties are View, Add, Edit, Delete and Subscribe. Permissions are decoupled from Web Parts so that administrators can repurpose, provision and reuse web parts in a variety of configurations. Extended properties can be created by code for associating custom properties with Web Parts on global or per-user basis.
Communication / messaging	A web-mail Web Part is available for sending messages within the portal. Also, a simple discussion form may be used for communication. Passage also uses a Blog WebPart for displaying messages.
Application sharing	Passage Portal uses reusable "web parts" components. It supports a wide variety of web component formats such as Perl scripts and CGI executables, but the preferred type of component is an ASP.NET user control (web part). Passage Portal is also tightly integrated with the Salesforce.com CRM applications and may be used for creating customer portals, partner portals, eMarketing landing pages, microsites.
Community building	A minimal Forum module is included
E-mail Adapter	A web-mail Web Part is available.
Calendar Adapter	Passage proposes a Calendar Web Part that may be used for displaying and editing various events.
Remote learning Adapter	Not available
Content Repository	Users can edit their own documents based on a quota using a Web Part called "My Documents". The documents available for the whole organization are available with the "Organizational documents"
External Content	External Content may be accessed using web services and, for news, RSS.
Performance	Passage Portal includes support for caching, load balancing, web-farm.

Conclusions

Both reviewed portal have strong and weak points. DotNetNuke has the following strong points: a very good skin engine that enables DotNetNuke powered sites to have professional designs, a good security, strong content management features and a lot of open source or free modules. Also, being a true open-source software, DotNetNuke has a very important and string community. Passage Portal has the following string points:

security, personalization and strong content management features based on their web parts. Also, we must not forget that there are other open source ASP.NET based portals like Cuyahoga, OmniPortal, Rainbow, Chrome Portal, SharpNuke (an incomplete C# port of DotNetNuke) and mojoPortal that runs on Mono.

Reference

- [1] Richardson, C.; Avondolio, D.; Vitale, J.; Len, P.; Smith, K.: *Professional Portal Development with Open Source Tools: Java Portal API, Lucene, James, Slide*, Wiley Technology Publishing, 2004, ISBN 0-471-46951-3;
- [2] <http://opensource.org/docs/definition.php>
- [3] http://www.mono-project.com/Main_Page
- [4] DiamonCluster: Enterprise Portals: Building Value through Organizational Knowledge, <http://marcrosenberg.com/images/Enterprise%20Portals.pdf>
- [5] Walker, S.; Brinkman, J.; Hopkins, B.; McCulloch, S.; Paterra, C.; Santry, P.J.; Willhite, W.; Caron, D.: *Professional DotNetNuke 4: Open Source Web Application Framework for ASP.NET 2.0*, Wrox, 2006, ISBN 0-471-78816-3;
- [6] <http://www.dotnetnuke.com/Default.aspx?tabid=970&EntryID=117>
- [7] Golden, B.: *Succeeding with Open Source*, Addison Wesley, 2004, ISBN 0-321-26853-9;
- [8] <http://www.dotnetnuke.com>
- [9] <http://www.cubiccompass.com>