

Xaraya Usability Recommendations

Status of this Memo

This memo provides information for the Xaraya community. It does not specify an Xaraya standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Copyright © The Digital Development Foundation (2003). All Rights Reserved.

Abstract

This document is intended to be the starting point of a specific series of initiatives designed to improve overall usability of Xaraya from all points-of-view (POV). This is a living document and, as such, is subject to change.

Specifically, we expect that some of the issues raised herein may have already been addressed and/or are in the process of being resolved. As these issues are resolved, they will be moved to the "Resolved Issues" Appendix. They will not be removed as this document is intended to be the document of record for all usability concerns.

Table of Contents

1 Preface	3
1.1 Underlying Assumptions.....	3
1.2 Terminology.....	3
2 Developer POV	5
2.1 Core Architecture.....	5
2.1.1 General Issues.....	5
2.1.2 Directory Structure.....	5
2.1.3 BlockLayout and Overall UI Issues.....	6
2.2 Third Party Developer (3PD) Standards.....	7
2.2.1 xApproved Plugins.....	7
2.2.2 xApproved Themes.....	7
3 Administrator (GOD) POV	8
3.1 Installation and Upgrades.....	8
3.1.1 BASE INSTALLATION.....	8
3.1.2 Upgrades and Additions.....	9
3.2 Administrative Utilities Toolset.....	10
3.3 Default User Roles.....	11
3.3.1 Administrator (GOD User).....	11
3.3.2 Editor (Super User).....	11
3.3.3 Member (Base User).....	11
3.3.4 Anonymous (Guest User).....	11
4 End-User (Member/Editor) POV	12
4.1 Role-Specific Behaviors.....	12
4.1.1 Editor Role.....	12
4.1.2 Member Role.....	13
4.2 Module-Specific Behaviors.....	13
4.2.1 Articles.....	13
4.2.2 Content Management.....	13
4.2.3 Community Management.....	13
4.2.4 Dynamic User Data.....	14
5 Revision history	15
Authors' Addresses	16
A Installation Reporting and Automated Update Tools	17
B Proposed Component Subcategories	18
C Site Kits	19
D References and Resources	21
E Package Installations (RPM/MSI Format)	22
F Resolved Issues	23
G Additional Credits	24
Intellectual Property and Copyright Statements	25

1. Preface

1.1 Underlying Assumptions

1. *Markets and User Base Defined*

Xaraya's target markets/user base are as follows:

1. *Web Designers/Developers*: Arguably the largest immediate audience of Xaraya, independent Web Designers/Developers are looking for tool that will allow them to quickly deploy a robust, yet simple-to-use content management system for their customers. This market will act as Administrators for client sites and their clients will act as Editors/Members.
2. *Individuals, Clubs & Special Interest Groups (SIGs)*: This market likely mirrors Web Designers/Devs closely. The differentiator is, in most cases, a small group or individual will be primarily responsible for application, content and user administrations
3. *Small-Medium Sized Business Web Presence*: The end-customer of the Web Designers/Developers who have enough knowledge in house to install and administer Xaraya as a public web presence. In addition to core CMS and User Management functionality, this market will be looking for 3PD eCommerce Plugins as well to allow them to conduct business on the web.
4. *Corporate Intranet*: Like the Small-Medium Business market, the Corporate Intranet market has the technical know-how on staff to install and administer Xaraya themselves. This market will look for specific Core and Plug-In components to enable Interop with existing systems. Examples include extending Core User components to use LDAP authentication and support for Oracle and Microsoft SQL Server.

2. *90% of end-users never touch code*

As developers first, we all tend to think too technically and, as a result, assume too much know-how on the part of the end-users of the code. While 3PDs are a valuable audience which needs active and aggressive support, they will be best served if we make the underlying Xaraya platform as user-friendly as possible for the non-technical browser/member and content owner/editor.

3. *Clear, Common Nomenclature Required*

To create clear, effective documentation which enhances usability for all target audiences, clear and consistent nomenclature is required. Where appropriate, we have attempted to establish naming conventions and labels we think will be most effective. All of this terminology is open for discussion. In the end, we need simple, concise definitions that work.

1.2 Terminology

1. *Xaraya Components*

Common terminology is a critical component to effectively communicating with one another and with the end-user. In the past, we have used the term *Module* relatively loosely. As we more definitively isolate core components more clearly from add-on components, we suggest deprecating the term *Module* and adopting the term *Component*. With this change, we further define Xaraya Components as follows:

1. *Core*: Core components include all files (module, SQL, docs, themes, etc.) considered an essential part of the core distribution of Xaraya.
2. *Plugins*: Commonly referred to as *Modules*, Plugins include any add-on which either extends a core Component or adds entirely new functionality to Xaraya.
3. *Themes*: Also referred to as *Skins*, Themes use BlockLayout to reconfigure the visual user interface (UI) of Xaraya. Themes can be as subtle as a simple color change and as radical as a complete redesign of the way th page functions and how information is arranged.

2. *General Terms Used in this Document*

1. *Interop*: Short for *Inter-Operability*, Interop refers to the increasing need for Web Services in general and Websites in specific, to interact with one another in meaningful ways. The primary goals of Interop initiatives generally serve both the developer and the end-user by establishing standards which make it simpler to share data and eliminate repetitive tasks.
2. *Third Party Developer (3PD)*: Any individual or team that develops Plugins and/or Themes yet is not member of the Xaraya Development Team.
3. *xApproved*: Working name for "blessed" Plugins and Themes. xApproved components have be reviewed by the Xaraya Team and are certified as compliant with the standards outlined in the Xaraya Developer's Guide.

2. Developer POV

2.1 Core Architecture

2.1.1 General Issues

1. *Platform Awareness:*

To enable ease of installation and a variety of administrative utilities, Xaraya needs to auto-detect the following information about the installation environment. *Note:* This need may already be addressed by calling PHPInfo in the installer.

1. *Operating System:* Brand & Version
2. *Database Platform:* Brand & Version
3. *Site Kit Dependencies:* Any packages which may be required for components of a specific site kit.
4. *???:*

2. *Component Subcategories:*

In addition to the macro-level classification of Xaraya Components (Core, Plug-In & Theme), Core & Plug-In Components can and should be further subcategorized to facilitate several initiatives. Refer to [Appendix B](#) for proposed Component Subcategories. Initiatives enabled by Component Subcategories include:

1. xApproved Component Registration
2. Auto-generated Component Information/Download/Install Directory
3. Automated Component Update
4. Site Kit Development & Distribution
5. Default Component Permissions based on User Role
6. *???*

2.1.2 Directory Structure

To enable quick, selective back ups, the application should be organized within a simple, yet specific directory structure. As new components (Core or Plug-In) are created, they should adhere to this directory structure. One possible solution is listed below. Within */var/www/html*, the top-level directories would be:

1. */backups:* If a backup utility is included in the Core, this would be the target directory for the local tar/zip files created.
2. */core:* Only Core components and their supporting files are held here. Subdirectories may include:
 1. */core/components*
 2. */core/language*
 3. */core/sql*
 4. *???*
3. */configs:* Xaraya's analog to */etc*, this directory includes all custom configuration files.

4. */content*: Isolated primarily to give one central, standard location for developers to put user-added content. By centralizing user added content in one standard directory, we enable simpler back-ups of content not contained within the DB. One simple example would be */albums* created by *Gallery*. Additional detailed examples include ...
 1. */content/articles*
 2. */content/albums*
 3. */content/downloads*
 4. */content/home/*.*username/* (*possibly used by Dynamic User Data*)
 5. */content/images*
 6. */content/music*
 7. */content/reviews*
 8. */content/uploads*
 9. ???
5. */docs*: This directory holds all docs for Xaraya. It would NOT contain docs for individual Plugins or Themes. Those docs would be held in a subdir of the directory created by and containing the Plug-In or theme.
6. */plugins*: All 3PD Plugins and their supporting files (*/docs*, */lang*, etc.) go here. This isolates them from Core components and reduces the opportunity for corruption of Core by accidental overwrites or installs gone awry. Unless installing a Site Kit containing non-Core Plugins, this directory would be empty with the base Xaraya installation.
7. */themes*: This directory would initially contain the default Xaraya theme and would be the target directory for installation of all 3PD Themes.

2.1.3 BlockLayout and Overall UI Issues

1. *Typeface*: Theme templates should consistently apply typeface across the application. Basically, all fonts should be controlled by the theme CSS. This may already have been addressed in the BL implementation. If so, we still need to keep an eye out for legacy font controls from PN which had very inconsistent font controls.
2. *Link Style*: If links are not underlined in theme CSS definition, rollover must include underline.
3. *Supported Browsers*: The following list covers all browsers in which Xaraya has been developed and tested. This does not mean Xaraya will not function in older browsers, only that it has been designed to leverage technologies available in more current offerings. Browsers not on this list are not officially support by Xaraya.
 1. Internet Explorer 5.5+
 2. Mozilla 0.9.8+
 3. Netscape 6.2+
 4. Opera 6.5+
4. *Proprietary CSS Attributes*: Xaraya does not endorse proprietary CSS attributes such as *scrollbar-*s* in Internet Explorer, or *-moz-** in Mozilla.
5. *General Enhancements to "How To" Language*: There are some changes that should be made to the English language file that will make X more clearly understandable. *See Drew*.
6. *Use Non-technical Language*: Remove all techno-speak from site. Example: Don't refer to things as "date/timestamp" -- say "date field", "time field", etc.

7. *Input Field Width*: Width of text input fields should be configurable by the Administrator in Site Settings.
8. *Breadcrumb Support*: Not sure if bread crumb support is planned as a BL Widget. If not, it should be.
9. *Page Weight*: Page weight on an un-modified configuration should be ~50K. Exceptions include articles, reviews, etc.
10. *Consistent Element Labels*: Consistent use and guidelines for naming of all UI elements, including, but not limited to buttons (SEARCH, GO, BACK, etc) and Field Names (ex. Username, Password, Address, etc.) Final approved list to be determined.
11. *Delete Confirmations*: Confirm *ALL* deletions from the site database.
12. *HTML Language*: All text in lang/xxx.php should be parsed as HTML, and displayed in HTML if codes are present. There are a few locations in which this is not the case. *See Drew*
13. *Printer Friendliness*: All pages should offer a link to a printer friendly version.
14. *Consistent Content Add/Delete Process*: With the exception of content-specific fields, the UI tools used to add content should have consistent look-and-feel, instructions and other UI elements and processes. In short, adding WebLinks should function the same as adding Reviews.

2.2 Third Party Developer (3PD) Standards

2.2.1 xApproved Plugins

Plug-In standards still need to be defined in the Xaraya Developer's Guide. Once established, we need an infrastructure to review and approve new *xApproved* Plugins.

1. *xarAPI Compliance*: All xApproved Plugins must comply with the xarAPI as outlined in the forthcoming Xaraya Developer's Guide.
2. *phpDoc Style Comments*: All xApproved Plugins must be suitably commented in the phpDoc style so comments can be extracted as needed for further development by Xaraya and/or other 3PDs.

2.2.2 xApproved Themes

Theme standards still need to be defined in the Xaraya Developer's Guide. Once established, we need an infrastructure to review and approve new *xApproved* Themes. Some preliminary standards include:

1. *Web Standards*: XHTML 1.0 (Strict) and CSS 2.0 compliant

3. Administrator (GOD) POV

3.1 Installation and Upgrades

The following describes the ideal Installation experience. Some features may not be feasible by Release 1.0 and/or may not make sense when described technically. New steps/screens/functionality are *emphasized* and, as a rule, are explained in greater detail later in the document.

3.1.1 BASE INSTALLATION

1. *Pre-Install Questions/Checks*

1. *Pre-Install Checklist*
2. Language
3. Accept GPL?
4. [Permit Installation Logging?](#)
5. Systems Setting Check
6. Database Settings
7. [Site Kit Selection](#)

2. *Base Configuration*

1. *Pre-Install Checklist*: Collect any non-automated configurations processes into one printable *DO-THIS-FIRST* file, and tell the user what they have to do (chmod, etc). Strongly recommend they print it out before proceeding
2. *phpMyAdmin*: In either basic or xModular form, phpMyAdmin should be distributed and installed with Xaraya Core. It is a very mature and useful tool for Administrators. As such, we should embrace it. As phpMyAdmin is a MySQL-specific tool, it would need adapted (presumably via ADODB) to work with other Xaraya-supported databases (PostGRESQL, Oracle). If this is not possible, the feature would only be made available for MySQL users or an alternative tool would be used
3. *Default Theme*: Xaraya distribution should contain only *ONE* default theme. Other themes may be readily available on Xaraya.com, but the single default theme is required to create a consistent set of documentation and help reduce the learning curve.
4. *Default Core Modules*:
 1. Articles
 2. *Credits*: A simple way to put ALL credits for EVERYTHING on the site in one place.
 3. Comments
 4. Help
 5. ?
 6. ?
 7. Search
 8. Site Map (*NEW*)
5. *Component Configuration*: Remove the LIST option from the start of Component (Modules) configuration in AdminPanels. Use "Regenerate" every time.
6. *Lorem Ipsum (Default Content)*: At installation, there should be default data in ALL areas of the site --

Reviews, Topics, Articles, FAQs, Sections, etc. -- if for no other reason than it provides the documentation team material with which to work when teaching how to edit or delete stuff.

7. ???

3. *Site Kits*

Site Kits are explained in far greater detail in [Appendix C](#). The following are high-level recommended behaviors of Site Kits.

1. Install ALL approved Modules, but only activate/configure those defined in the Site Kit.
2. Tell the user what modules will be ACTIVATED (all are installed, selection above determines which are activated by the installer)
3. Display INACTIVE modules (with descriptions) and allow user to add them.
4. Any modules that are activated which require additional configuration, the configuration of that module is added to the installation process.
5. ???

3.1.2 Upgrades and Additions

Again, Red Hat Network and MS Windows Update are the functionality benchmarks for ease of use. Specifically, Xaraya would ideally have a built in mechanism through which the Administrator can click a button and have their installation compared to a database residing on Xaraya servers. Once this comparison is complete, a report is generated describing available updates and giving the Administrator the option of downloading and installing the updates automatically.

1. *Overall Considerations*

1. *Xaraya Registration DB*: Centralized database housed on the Xaraya web server (Sasquatch). This DB contains two primary types of information:
 1. *Component Registration*: All Xaraya components (Core, as well as Official/Approved Plugins and Themes) require registration with a unique identifier in the Xaraya Registration DB.
 2. *Install Registration*: If we elect to go this route, the Registration DB would also contain unique registration keys for each Xaraya installation.
2. *Available Updates Profile (Xaraya Server)*: Profile derived from Component Registration tables. Used in Comparison Pass to determine which updates are available for installation.
3. *Installation Profile (Local Install)*: After install/update is completed, an Installation Profile is created/modified on the user's server. This file will be used in the Comparison Pass and, optionally, it could be pushed to the Install Registration tables as well.
4. *Comparison Script*: Admin Panels provides *Check for Updates* button. When clicked, the following occurs:
 1. *Pre-Upgrade Checklist*: Check list is displayed which includes, among other things, a detailed description of what is about to occur and what, if any, server configurations (ex. Enable write permissions) are required to proceed. It also includes links to recommended pre-upgrade utilities (ex. [File/DB Backup](#)). After reviewing this checklist and taking action as needed, the user clicks the *Next* button. *Note*: All Upgrade screens have and *Cancel* button which exits out to AdminPanels.
 2. *Pass Installation Profile to Server* Passes the local Installation Profile to the Comparison Script on Xaraya's

- server. Script compares profiles and reports on available updates.
3. *Select Desired Updates*: From the report, user uses checkboxes to select desired updates for installation.
 4. *Package and Transfer Files*: Based on user selection, script builds a tar/zip file containing the selected updates and writes it to the install directory on the target server.
 5. *Install Updates*: Once the file has transferred, the script expands the archive and writes files to the appropriate location. If needed, the script also updates the *SQL server
2. *Core Specific*
 1. *TBD As Needed*
 3. *Plug-In Specific*
 1. *TBD As Needed*
 4. *Theme Specific*
 1. *TBD As Needed*

3.2 Administrative Utilities Toolset

To truly administer a Xaraya site well, the Administrator needs a strong toolset. Some of the tools are be discussed elsewhere in this document. The following is a current best guess at the desired Administrative Utilities Toolset.

1. *Backup Utilities*: Using platform awareness and available tools, the Administrator need the ability to quickly and efficiently back up his site prior to upgrading. These tools include Database Backup (phpMyAdmin or similar tool) and File Back Up for the following Xaraya directories (at a minimum).

1. /configs
2. /content
3. /plugins
4. /themes

2. *Custom Language Editor*: One interface to edit ALL language files, even those in component directories (Plugins & Themes). As part of the component registration process, it would announce the presence/location of the component language files.

Drew's Notes:

It would be great if there was a default file, say *eng_override.php* (different prefix for other languages, of course) in the /configs that would allow over-ride of what's in the language files, to prevent having to edit the distribution files directly. For example, in /plugins/<plugin_name>/lang/eng/global.php, it has `'define("_PRODUCTTITLE","Product title");'`, but I would like to override that.

So, in *eng_override.php*, I would be able to do `'define("_PRODUCTTITLE","Product/Event title");'` and have *eng_override.php* included at the appropriate times to override what's in the distribution file if the *eng_override.php* file exists. As I edit my distribution language files more and more, the more I think this feature would be very helpful. If the *eng_override.php* file doesn't exist, it would just use the defaults in the distribution language file.

3. *Groups & Permissions GUI*: Groups & Permissions GUI was badly in need of an overhaul. Marcinmilan is already hard at work on this. We are just calling it out here to support his work in progress.
4. *Administration Menu Block*: The default installation should display Administration menu items in a menu block, not as a module list. Everyone uses the menu block method; it's the most powerful option. *Note*: This method is already being employed in AdminPanels. Like Permissions GUI, we are calling it out here as a show of support.
5. ???

3.3 Default User Roles

One of the core failings of the *Nuke platform is that it only provided two default user roles (Administrator and User). This approach assumes that the Administrator was also the Content editor in all cases. Regular usage of *Nukes and user feedback indicates the need for a third default role (Editor). Each proposed user role is defined below.

3.3.1 Administrator (GOD User)

No real role change here. This is the Xaraya equivalent of */root*. Just as no Server Admin will give out */root* to just anyone, so to should the Administrator role be tightly guarded. By default, the Administrator can do *anything* on the system.

3.3.2 Editor (Super User)

This is an entirely new default role. Most Administrators end up creating this sort of role anyway, so it makes sense to include it as a default. By default, this role has *ONLY* content and user management (add/approve/modify/delete) rights. They have no system configuration, install/upgrade, database access or other global rights to impact the system.

3.3.3 Member (Base User)

In the past, this role has been generically referred to as *User*. This level of access gives permission to additional content and/or services and local user configuration options. This role has no Administrator/Editor rights. This role can submit content for publication, but it must be approved by an Administrator or Editor.

3.3.4 Anonymous (Guest User)

No real change here. This is a single account which is used by generic browsers. Beyond accessing publicly available content and services, this user has *NO* rights on the system.

4. End-User (Member/Editor) POV

4.1 Role-Specific Behaviors

4.1.1 Editor Role

1. *Context Sensitive Editing Tools:* All Components should provide access to Administrator/Editor tools from *WITHIN* the module if the user logged in is an Administrator or Editor with the appropriate permissions for that Component.
2. *Reorganize Content Management Links/Pages:* Organize Editor pages by long-term usability and put the options that are most likely to be frequently used at the top of the page. For example, WebLinks is currently ordered as follows:
 1. Add a category
 2. Modify a category
 3. Add a new link
 4. Modify a link
 5. Web Links Configuration

It is more logical, when long-term usage is considered, to order it as follows:

1. Add a new link
2. Modify a link
3. Add a category
4. Modify a category
5. Web Links Configuration

Alternatively, it would be very useful (though probably difficult to implement) to track usage of Editing Tools based, perhaps, on page hits. Armed with this usage data, the system could, upon logon, present the Editor with a *Frequently Used Tools* Menu Block or AdminPanel.

3. *Article MetaCommands:*
 1. *Story Links:* Provide an easy method within an article to link to another article on the same site. Metacommands like "<readprev=520>" and "<readnext=525>" would be VERY useful. Each metacommand would be replaced as follows:
 1. "<readfirst=519>" would display user-defined text (like "<i>Click here to read the first article</i>") in this series link to article #519.
 2. "<readprev=520>" would display user-defined text (like "<i>Click here to read the previous article</i>") and link to article #520.
 3. "<readnext=525>" would display user-defined text (like "<i>Click here to read the next article</i>") and link to article #525.
 2. *Read More:* Add a metacommand for NEWS STORIES like "<readmore>". When this text is found in the body of an article, automatically replace it with a user-defined string (like "<i>Click here to read more...</i>") that links to the full article view. In the full article view or in print view, the <readmore> string is NOT displayed.
 3. *Page Break:* Where ever this appears, a single article is broken into multiple pages with auto-generated

navigation links (ex. PREV, NEXT, etc.)

4. *Configurable URL/HREF Targets*: Ideally, Xaraya would be aware if the contents of an HREF are local (onsite) or external (offsite) as determined by the presence or absence of *http://*. The Administrator/Editor would then be able to set a configuration variable which would determine if the HREF should pop a new window to display the target URL (analogous to *target="blank"*). This setting would be global across the site, but could be over-ridden on a case-by-case basis.

4.1.2 Member Role

1. *my.<domainname>.com*: On login, default page is *my.<domainname>.com*. This page would display things like:
 1. Waiting private messages,
 2. Custom RSS feeds,
 3. User's admin tools (gallery upload, change password, etc),
 4. *New Items* (See Below)
 5. Bookmarked Forums, Links etc (see vBulletin)
 6. *Alerts & Notifications*: Derived from choices made in Pub/Sub/Unsub component.
 7. Other Dynamic User Data (DUD) enabled content
 8. ???
2. *"NEW" Block*: A category called "NEW" that displays, based on item timestamp and date/time of user's last login, new items. Period of time for "new" status is configurable by admin.
3. *Content Sorting*: Users can set PRIORITY of items on the site (display SPORTS topic items first, display ARTS topic items second, etc.) and the page will show only the items that the user has configured to see (turn OFF SPORTS topics (they are not displayed)..

4.2 Module-Specific Behaviors

4.2.1 Articles

1. *PREVIOUS, NEXT, and TOP Links* : Display these at the bottom of the front page. Get the count of the number of articles to display from the value in Administration --> Settings --> Stories # on homepage.
2. ???

4.2.2 Content Management

1. *DMOZ/Yahoo Index*: All modules which gather and classify information (links, downloads, etc) should use a DMOZ/Yahoo style index.
2. *New Item Indicator*: New items in sub-categories need some "new" indicator (icon, font color, etc.) which is configurable at the Administrator and/or Theme level.
3. ???

4.2.3 Community Management

1. *User/Context Sensitive Help System:* Ideally, the help system will be aware of both the Component it is being called from (to determine Help Content) and the Role of the User logged in (Admin, Editor, Member) and display the appropriate level of Help detail based on user Role.
2. ???

4.2.4 Dynamic User Data

1. ???

5. Revision history

2003-01-15: First Draft (Daulton/Vogel)

2003-02-03: Second Draft (Daulton/Vogel)

2003-02-11: Release 0.9 (Daulton/Vogel)

Authors' Addresses

Douglas Daulton

Xaraya Development Group
EMail: apakuni@xaraya.com
URI: <http://www.xaraya.com>

Drew Vogel

Xaraya Development Group
EMail: drew@drewvogel.com
URI: <http://www.drewvogel.com>

A. Installation Reporting and Automated Update Tools

It has been suggested that, upon completion of a Xaraya install, the application either automatically ping Xaraya's server and collect information (perceived as spyware by many) or we ask users to register which does the same thing. For real world examples ... think Windows Update, Red Hat Network and the Eclipse Updater Tool.

This Reporting Installer could collect all sorts of valuable information which, if positioned correctly and paired with a clean and simple opt-out process could achieve the following goals:

1. Accurate Install/Usage statistics with which we can use in ...
 1. Forecasting future installs,
 2. Defining Market Share,
 3. Fund-raising,
 4. Marketing
2. Establishes useful user database with as much (or as little) information as we feel is appropriate to maintain about our install base. This would not be spyware because we would allow folks a simple way to opt-out at the beginning (no optin by default) and if they optout, we would simply add a install tick to the counter and collect no data.

As users elect to opt-in, they share more information about their install and intended usage which allows us to do some really cool stuff like ...

1. Automated core updates (again, think Eclipse Updater, RHN, WU)
2. Detailed market analysis (who is using the site for what ... blogs, music sites, pron, etc)
3. Detailed system stats for install base (windows, LAMP, etc)
4. Automated YANXS directory
5. Security Alerts and Patch Notices via email even SMS or pager

We can work out the details, but you get the idea. It is not spyware if we give them the option to opt-out and tell them if they do, no information beyond a "Install Counter tick" will be passed and we could even forgo that if we wanted. IMO, the benefits to everyone (core, user and dev communities alike) and hurts no one.

B. Proposed Component Subcategories

1. Core Components

1. *Setup (Base??)*: Single or infrequent-use tools for configuring the site ex. Site Configuration, Theme Selection
2. *Utilities (AdminPanels)*: Backend Management (db adds, cleanup, archiving, etc) ex. Translations, Autolinks, Web Services, Categories
3. *Content Admin*: Any tools which setup how content is viewed/managed ex. Articles, Links, Downloads, RSS, Pbsub, Polls Reviews, WSW Editor
4. *Membership (Users)*: Any and all tools for user management ex. Groups, Permissions, DUD, Messages, Mail Blast
5. *Site Promotion*: ex. Recommend, Stats, HitCount, Keywords, Mail Blast
6. *Online Help*: ex. Search, FAQs, Help

2. Plug-In Components

1. *eBusiness/eCommerce*: ex. OpenRealty, OSCommerce, CafePress, Amazon bookstore, etc.
2. *Scheduling/PIM*: ex. xCalendar, xProject, ToDo List, BKView, Seminar Scheduler
3. *Communication/Social Networking*: ex. Forums, NNTP UI, Chat/ShoutBox, WebRing
4. *Content Expansions*: ex. Pinup, Gallery, MeG, Comic, AvantGo, MP3 Managers/Streamers, Fan Fiction, Collaborative Stories, MyHaiku, Bible/Torah/Quran Verse DB, pnEncyclopedia, pnDictionary, CookBook/Recipe Modules
5. *Games*: ex. Nascar, Fantasy Football, Flash Games, Half-Life/Counterstrike/Unreal Server Stats
6. *Data Collection*: ex. Surveys, Quizzes, Extended Polls
7. *Dynamic User Data*: ex. xResume, xPortfolio, My <sitename> HomePage, Blog, DUD DB Expansions for specific groups (Church, School, Club, Game Clan, etc)
8. *Core Extensions*: These are really Core Components Hacks and/or Replacements. May not be prudent to list them, but would include stuff like ... Advanced Search, Mass Mailer, KnowledgeBase(expanded FAQs)

C. Site Kits

As the name implies, *Site Kits* are a series of pre-configured Xaraya tarballs (Core + Modules). As frame of reference, think of the Red Hat GUI-based Installer. For example ...

Red Hat: What sort of install is this?

1. *Server*
2. *Workstation*
3. *Laptop*
4. *Custom*
5. *Everything*

Like the Red Hat example, Xaraya Site Kits would install and configure pre-built packages based on user choice. The Xaraya example may look like this ...

Xaraya: How will you use your Xaraya installation?

1. *Generic Community* : Includes Links, Calendar, Downloads, Forum, Articles, Users
2. *News Site*: Includes Articles, Users, Reviews, Calendar
3. *Personal Site*: Includes News, BLOG, Links, Gallery
4. *Family Site*: Includes Users, Gallery, Articles, Reviews, Links
5. *Online Store*: Includes Users, Forums, OSCommerce, Reviews
6. *Gaming Clan*: Includes Calendar, Articles, Reviews, Forums, Users, Half-Life Server Stats (or other game server stats)
7. *Corporate Intranet*: Includes LDAP, SQL2000 support, Forums, Articles, Gallery, FAQs, Calendar, etc.
8. *School Homeroom*: Includes Users /Students, GradeBook, Forums, Articles, Reviews, Gallery, etc.
9. *Non-Profit*: Includes Users, Forums, Paypal Donations, Reviews, Articles, etc.
10. *INFINITE COMBINATIONS*: This list can (and will) grow.
11. *Custom*
12. *Everything*

As you can see, the possibilities are boundless. The advantages of this sort of functionality are manifold ...

1. It engages 3PDs in much the same way the Red Hat installer and RHN does. It gives them a framework within which to simply distribute their work and a clear incentive to use that framework guidelines (xDeveloper Guide) to the letter.
2. It provides a distinct competitive advantage over other OS CMS as it makes us the most configurable tool while making said configuration as simple for the end user as possible.
3. It shows foresight and planning which will invite/attract other, bigger development partners (IBM, etc) and helps us cement Xaraya as something more than garage project like PN or PHP-Nuke.
4. Along with things like LDAP authentication, SOAP support and other Interop tools, it provides a possible entree into larger corporate install base.
5. Driven by BL, it makes Xaraya the platform of choice for independent small to medium sized web developers looking for a way to provide dynamic websites without knowing how to code. They do the install, retheme and viola.
6. It makes us a viable option for small to medium business owners looking for a way to establish a web presence at low cost. If they have a bit of tech savvy they can do it themselves.
7. Opens the doors for schools (K through University) and other Non-Profits to establish web presence in a simple, cost-efficient manner. They may pay their ad agency to develop their theme/look, or better yet call on the body of Xaraya developers/designers to do so, and save the development costs. At the HS and

University level, they may also grow a new generation of Xaraya devs by engaging their students in module development.

8. Instantly breaks the current OS CMS mold which will certainly lead to considerable positive exposure for this project in the OS media and in

Lots of food for thought here ... but the bottom line is this sort of feature/module will go a long way to further killing installer usability concerns, maintaining an active, engaged an loyal community and rebuilding a loyal install base for Xaraya.

D. References and Resources

1. [AQUA: Apple's Human Interface Guidelines](#)¹
2. [Nielsen Norman Group: Web Usability Guidelines](#)²
3. [SitePoint: Do-It-Yourself Accessibility](#)³
4. [Software Usability Research Lab: Wichita State University](#)⁴
5. [Usability and Open Source Software by David M. Nichols and Michael B. Twidale](#)⁵
6. [WC3: User Agent Accessibility Guidelines 1.0](#)⁶
7. [www.useit.com: Usability Guru Jakob Nielsen](#)⁷

¹ <http://developer.apple.com/techpubs/macosx/Essentials/AquaHIGuidelines/index.html>

² <http://www.nngroup.com/reports/>

³ <http://www.webmasterbase.com/article/744/16>

⁴ <http://psychology.wichita.edu/optimalweb/default.htm>

⁵ <http://www.cs.waikato.ac.nz/~daven/docs/oss-wp.html>

⁶ <http://www.w3.org/TR/UAAG10/>

⁷ <http://www.useit.com>

E. Package Installations (RPM/MSI Format)

These ideas require a more mature, stable Xaraya release to begin feasibility review. They are captured here because ultimately, they serve both the project (wider distribution channels) and the end-users (ease of installation).

1. *Linux Distributions:* Work with Red Hat & other major Linux distros to integrate Xaraya into their distribution as an on-install package that check dependencies, etc.
2. *Windows Distributions:* Work with FoxServ project to integrate Xaraya into their installation.

F. Resolved Issues

This space reserved for future use.

G. Additional Credits

Many thanks to the following folks who helped research or otherwise prepare these materials.

- Chris Dudley (Miko)
- Marcel Van der Boom (MrB)
- Marty Vance (Dracos)
- Tomislav Grabic (Tomi)

Intellectual Property Statement

The DDF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the DDF's procedures with respect to rights in standards-track and standards-related documentation can be found in RFC-0.

The DDF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the DDF Board of Directors.

Acknowledgement

Funding for the RFC Editor function is provided by the DDF