



# FarCry 4.0 CMS Developer Training Course

Version 2.1

FarCry CMS



BY NC ND

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 2.5 License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-nd/2.5/au/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

This work is copyrighted Daemon Pty Limited 2007, http://www.daemon.com.au/

# 1 Table of Contents

1 Table of Contents
2 Course Overview
2.1 Course Objectives4
2.2 Course Prerequisites4
2.3 Solar Safaris Website4
3 FarCry Introduction
3.1 Programming Environment
3.2 Dynamic Authoring and Publishing
3.3 Content Object API (COAPI)
3.4 Tree Model6
3.5 Services6
3.6 Security Model
3.7 Presentation Logic
3.8 Container Management
3.9 Code BaseS
3.10 Extensibility
3.11 Third Party Plug-ins
3.12 Coding Standards
4 Administration Interface
4.1 Home Tab14
4.2 Site Tab
4.3 Content Tab
4.4 Admin Tab22
4.5 Security Tab
4.6 Reporting Tab
5 Webskins
5.1 MOLLIO
5.2 Home Page
5.3 Header
5.4 Footer
5.5 Standard Page
5.6 News page
5.7 Unit Review
6 Include Objects
6.1 Sitemap
6.2 Search
6.3 Contact Us

6.4 Unit Summary	
7 Custom Types	
7.1 Component Definition	
7.2 Properties	
7.3 Deployment	
7.4 Adding new properties	91
7.5 Deleting properties	
7.6 Edit destinations	
7.7 Adding Custom Type to the Navigation Tree	92
7.8 Status	
7.9 Versioning	94
7.10 Webskin	95
7.11 Child Links Rule	96
7.12 Dynamic Custom Type	
7.13 Custom Admin	
7.14 Webskins	
7.15 Lab	
7.16 Unit Review	
8 Custom Rules	
8.1 Rule Definition	<u>107</u>
8.2 Deploy	
8.3 Teaser webskin	
8.4 Execute Method	
8.5 Lab	
8.6 Unit Review	113
9 Advanced Topics	114
9.1 Caching	
9.2 Security	116
9.3 Extending Core Types	
9.4 Friendly URLs	
10 ADDITIONAL ONLINE MATERIAL	
10.1 FARCRY CMS OPEN SOURCE COMMUNITY	
10.2 farcry-dev (public)	124
10.3 farcry-user (public)	
10.4 farcry-beta (public)	
10.5 BUG TRACKING	124
10.6 FARCRY WIKI	

# 2 Course Overview

The FarCry Developers Course is designed to introduce the concepts of the FarCry Open Source Content Management System and teach experienced ColdFusion developers how to install, develop, customise and optimise a FarCry application.

The course is task-based, with students building their own FarCry application over the three-day course period. Each unit will introduce a new concept, allow for students to follow a demonstration, and then be walked through a lesson with their instructor and finally complete a task on their own to reinforce the concept.

# 2.1 COURSE OBJECTIVES

By the end of the course, students should be able to:

- Install FarCry
- Use the FarCry Administration Interface
- Customise the look and feel of a FarCry Application
- Create custom Include objects
- Create custom administration functions
- Create custom types
- Create custom rules
- Optimise a FarCry application

# 2.2 COURSE PREREQUISITES

The prerequisites for this course are:

- Experience with HTML
- Experience with ColdFusion
- Experience with SQL

## 2.3 SOLAR SAFARIS WEBSITE

Throughout the 3-day course, students will be developing a site for a futuristic travel company called Solar Safaris. On completion of the course this website will contain the following functionality:

- Destination Profiles
- Package Tour information
- Online Reservation
- Contact Form
- Search Form
- Site Map
- Latest News

# 3 FarCry Introduction

# 3.1 PROGRAMMING ENVIRONMENT

FarCry is built on top of the award winning Macromedia ColdFusion application server. ColdFusion is a Sun Certified java application. ColdFusion script and mark-up compiles to java byte code and runs on the majority of J2EE application servers. Although the underlying Java environment can be leveraged, FarCry developers only need to understand the CFML language in order to develop within the application framework.

# 3.2 DYNAMIC AUTHORING AND PUBLISHING

FarCry CMS follows a dynamic authoring and publishing model. Content is composited on the production server and delivered to the visitor "on the fly". FarCry does not publish static HTML files. Although FarCry does have a series of sophisticated caching services that can be used to reduce load on the application server as required.

Content authoring, staging and delivery can all be served from a single server instance. Alternatively these functions can be distributed across multiple servers depending on the deployment. FarCry features a virtual-staging mode that allows contributors to preview draft content in the context of the live, published website.

The authoring environment is feature rich and a more comprehensive discussion of authoring and publishing can be found in the FarCry Feature FAQ.

# 3.3 CONTENT OBJECT API (COAPI)

The FarCry CMS Content Object API (COAPI) forms the foundation for all content built and managed on FarCry CMS. The COAPI is a database abstraction layer that uses an object-based programming model for managing content. The site object model is a blueprint for the underlying data schema of any FarCry application.

The COAPI provides a content storage model that enables developers to transparently store all forms of content in a structured object storage system. The FarCry Content Object database is built on top of a standard relational-database model, enabling customers to use almost any standard relational database as the actual storage infrastructure. Support as of v4.0 includes MySQL, MS SQL Server, Oracle and PostgreSQL.

Storing content in a core repository meets the key design goal of a clean separation of data or content from presentation. This allows developers to easily repurpose content based on a given user's browser, or for delivery using other formats, such as XHTML, XML, RTF, e-mail, etc.

Using the COAPI shields developers from having to program or manipulate the database schema; instead they use an elegant ColdFusion component-based API to interact with the Content Object database. This API facilitates the deployment of new content types and the modification of existing types. In addition it manages the evolution of the underlying relational database schema, including the modification of data types.

Because the COAPI is extensible to support any form of back-end system, developers can simultaneously access content stored in either the Content Object database or in their own external databases. The FarCry content model supports storing most popular content types, including

files, images, Flash animations. Binary media such as files, images, and audio/video are stored in filesystem based media directories for fast retrieval and delivery. Developers can also extend the content model to include content and data stored in any external databases.

## 3.3.1 Content Types

A site object model is comprised of a collection of Content Types. Each object type contains a set of properties and methods. Properties are the elements of data associated with an object type. For instance, a news object type might have title, teaser, body and publishdate properties. Methods are the activities that surround the object, such as edit and display. The code that manages the activities is built using a ColdFusion component. The properties of information or data can be stored and managed by FarCry, using the Content Object database, or in any external RDBMS accessed via SQL or object middleware.

Core types are those content types that ship as part of the FarCry core libraries. They include Navigation, HTML, Files, Images and many others. These by and large make up the default installation and administration interfaces within FarCry CMS. FarCry developers are also able to take advantage of "plugins" which are discrete modules projects can leverage, gaining access to additional functionality. There are many public and freely available plugins, the most common being "farcrycms" which contains News, Events, Links and Facts. A "vanilla" FarCry deployment is one that only uses the FarCry core types and has no customisations. Custom types are content types that are specific to the FarCry application in question. This term can also refer to core content types that have been extended and modified.

Content types are based on ColdFusion components. These specialised components extend (or inherit) an abstract class (types.cfc) within the FarCry core library that provides the component with system properties and methods needed to operate within the FarCry framework. This allows custom objects to immediately leverage the plethora of services within the FarCry framework. Obviously specific properties and methods can be overridden as required.

## 3.3.2 Content Objects

Content object is a term used to represent a specific instance of a content type. For example a News content type would define the relevant properties and methods available – like a boiler template. The content object would represent a specific instance of news content. Although all content objects of a particular type will have different data associated with them, they will nevertheless share a common set of property fields and methods.

## 3.4 NESTED TREE MODEL

Hierarchical information within the FarCry CMS environment is modelled in SQL using a variation of the nested tree model approach (championed by Joe Celko, <u>http://www.celko.com/</u>). This provides a very fast and powerful mechanism for managing tree related information. For example, the site information hierarchy, the relationship of one fixed page to another and the categorisation service, the relationship of one keyword to another, are managed within the FarCry tree model.

The tree modelling approach allows for all kinds of sophisticated content management including the ability to easily rearrange positional information and rapidly determine essential metadata such as breadcrumbs, ancestry, descendents and so on. Developers don't need to understand the tree model to leverage its power – everything is exposed through a simple component API. Common tasks like generating navigation and positional information for the presentation layer are simplified further still via a library of custom tags.

## 3.5 SERVICES

FarCry, as you might expect, contains a library of services for handling content management functions. These include workflow, versioning, Verity free text search integration and so on. The FarCry administration area or webtop is effectively a sample application that leverages these services to provide the visible out of the box FarCry solution for the default content types. These services can also be leveraged to support custom content types, publishing rules and other customisations. Effectively anything that is possible in the default administration area should be accessible as a service through the component API.

Components are generally self-documenting, and it's worth using the ColdFusion component browser to review the functions available by default in the application framework. Many integration examples are available for review from the FarCry developer community.

## 3.5.1 Versioning and archiving

By default only HTML objects are versioned and archived in the core framework. These services can be leveraged for any content type within the system.

#### 3.5.2 Verity free text search services

FarCry leverages the built-in Verity K2 engine shipped on the ColdFusion application server to provide free-text searching across all content types. The individual content type properties you require searching for can be configured at will through the web based administration. FarCry also provides options for searching external file libraries.

## 3.5.3 Scheduled tasks

FarCry has its own task management subsystem that uses the underlying ColdFusion cron service. This is typically used for scheduling maintenance events, periodic reporting and the like.

## 3.5.4 Categorisation

There is a central categorisation service that can be applied to any content object. The categorisation service is exposed by default to many of the core content types. Categorisation is based on a keyword tree that can be edited by appropriately privileged users. Content can be selected for by specific keyword allocation or by keyword tree branch.

## 3.5.5 Export

FarCry ships with an export service that allows the content instances of any content type registered in the system to be exported according to a developer defined template. By default we incorporate a basic XML template and a more specialised RSS template for content syndication. Developers can of course extend this service by providing additional templates as required.

## 3.5.6 Reporting

FarCry has an observation architecture that records the detail of every object access. Detailed web statistics reports are available for all kinds of activity throughout the website. FarCry ships with a standard set of reports that can be supplemented as required by processing the captured data.

FarCry applications generate a standard web log (from the web server itself) for third-party logging tools to process.

## 3.6 SECURITY MODEL

FarCry incorporates a very sophisticated role based security model. FarCry authenticates against a user directory expecting a true/false on the login credentials and group membership for the user. User groups are then mapped to internal policies or roles, thereby determining the privileges of the user in the system.

#### 3.6.1 User directories

FarCry supports multiple user directories including ActiveDirectory, NTDomain, LDAP and JDBC/ODBC directories. Significantly FarCry can manage users in multiple and mixed directories at once in a single application. Furthermore, user directories can be shared across multiple FarCry application instances.

## 3.6.2 Policy Groups or Roles

FarCry policy groups are effectively a collection of permissions within the system. FarCry has a library of permissions covering all aspects of the framework. Policy groups are mapped to specific user groups within the associated user directories configured for the application.

Policy groups are completely customisable. FarCry ships with five default policy groups. These can be modified, substituted or added to as required. "Anonymous" is a special role identified to the system as effectively the absence of any group information – its very handy to be able to apply permissions for users who have not been authenticated.

Developers can readily add additional permissions to cater for customisations they may need to make. In addition, developers can use permissions to provide visibility on tabs and menu items within the web based administration as well as access to specific functions or actions.



# 3.7 PRESENTATION LOGIC

FarCry isolates the presentation layer into a special area of the code called the "webskin". We've done this to provide the most simplistic offering possible for template designers to composite presentation templates within the system. As such, templates can be developed by anyone confident with HTML.

Literally any sort of presentation layer can be developed. By way of example only, FarCry installs with Mollio (http://www.mollio.org/), a set of XHTML compliant templates developed by Daemon.

Templates are ColdFusion pages. As such they can incorporate any functionality available to the FarCry component API or the underlying ColdFusion server itself. Consequently a template in the FarCry sense can do all sorts of interesting things. The template code has access to the properties of the object instance, its positional information (including an understanding of its parents, children, siblings and related links) environmental variables (including any session profile for the current user), and all the underlying FarCry services. This essentially allows for an extremely rich environment for the development of the presentation layer. For example, a page breadcrumb can be dynamically built by asking the FarCry API where in the site the currently viewed content sits.

Each content type, for example a web page or news article can have any number of templates associated with them. For instance, a web page content type might have templates for "Full Page Display", "Teaser Display", and "Text Only Display". A single content item could be rendered in several different ways, just by changing the processing template.



FarCry has a template caching sub-system that allows template designers to cache different regions of the template for any specified time. For example, one might cache the primary navigation for 6 hours but a random testimonial for only 5 minutes. Any combination of caching regions and periods are possible.

#### 3.7.2 CSS support

Template developers can reference their own fixed style sheets in the presentation layer.

In addition, FarCry has a specific content type for managing external style sheets. This can be referenced in the template to provide a specific set of cascading style sheets for different regions of the web site. Furthermore, these style sheets can be managed independently of the template engine by non-technical users through the web based administration area.

# 3.8 CONTAINER MANAGEMENT



FarCry has a special sub-system for managing dynamic content within the presentation layer. Specific regions of any template can be designated as a "container" and have dynamic content scheduled into them by non-technical authors. This behaviour in other systems is sometimes called, portlets, page behaviours, pods or a variety of other names.

The concept is important to understand because it represents some of the most powerful aspects of the FarCry framework. Containers are regions of a page where some sort of programmatic behaviour can be placed. For example, a list of the latest news items, an event calendar, content syndicated via XML from another website, a randomly picked fact and so on. These programmatic behaviours in FarCry are called Publishing Rules.

#### 3.8.1 Containers

Containers are placed in a template with a simple custom tag. FarCry takes care of the rest of the required machinery. Based on the name given to the container, they can be unique to a specific page, shared across a specific section of the website or shared globally throughout the website. They are populated by contributors selecting from a predefined set of publishing rules.

#### 3.8.2 Publishing Rules

Publishing rules are a special content type that capture some parameters in edit mode from a contributor and then at run-time, in execution mode, dynamically display content based on some predefined behaviour. FarCry ships with a library of publishing rules for contributors to choose from. Developers can customise these rules or build their own.

For example, the news rule allows a contributor to select a type of display, categories for filtering and the number of news items. When a visitor accesses the page with the rule, the predefined behaviours goes off and grabs the very latest news items matching the specified categorisation and displays them with the right template. When even new news content is added to the system, this rule will automatically update the page without any further intervention.

For example, the child links rule will display teasers from all a pages underlying child pages. This sort of behaviour is used to populate a summary or landing page dynamically based on the underlying pages of that section of the website. Add, update or remove a page and the landing page is automatically updated.

For example, the "random fact rule" will randomly pick and display a predefined number of fact content objects. This rule would update and refresh its content every page request. This is often used as a great way to show off testimonials or other snippets of information.

## 3.9 FRIENDLY URLS

The FriendlyURL servlet is used to generate beautiful user and search engine friendly URLs for any FarCry application. FriendlyURL now ships as part of the standard FarCry distribution but is turned off by default. FriendlyURL is very easy to configure and is highly recommended.

## 3.10 CODE BASES

One of FarCry's distinct architectural advantages is the separation of core CMS services from unique project-by-project customisations. We work exceptionally hard to make the core libraries configurable by extension and not be replacement. You should not have to make modifications to the core library as part of normal development. This significantly minimises the impact of core changes on existing deployments. Furthermore, it ensures that development teams can continue to benefit from the fruits of ongoing open source development in the core code whilst providing uniquely customised deployments.

#### 3.10.1 Core

"core" is the primary code base. It holds all the content management services and methods. It effectively provides the application framework and the presentation layer for the FarCry Administration area. It has been specifically isolated as the core, managed code base -- you should never have to modify this code base except to patch bugs or extend core functionality. All project specific customisations should be done elsewhere!

The core library contains all the generic services that make up a "vanilla" FarCry deployment. It is worth noting that the sample application that comes as part of the installation only represents a small sub-set of functionality available overall.

The Component API is well documented, and you can review this online at: http://www.farcrycore.org/cfcdoc/

#### 3.10.2 FarCry Project

Every FarCry instance has its own unique project directory. This directory holds the site-specific configuration files, the webskin (presentation layer) and anything else that makes a project unique.

On installation, new applications contain skeleton templates from "Mollio". You start with this very basic application structure and then modify it to your hearts content. This repository contains basic templates such that your project is operational on deployment. This basic structure includes configuration files, directory structures for customisation and a presentation layer for the core CMS elements.

#### 3.10.3 Mollio

This code base is most often used for installation of new projects. The installation routine generates a new application (including populating a fresh database) using the base templates stored in Mollio. Although Mollio can be used as a standalone application, it really should be treated as a template for building upon.

## 3.10.4 FourQ

"fourq" is the content object API (COAPI) used by FarCry as a database abstraction layer. It is this abstraction layer that makes the deployment and evolution of content types within the framework so flexible. fourQ is an interesting use of CFC concepts and well worth a look -- even if it's just to spark your own database abstraction ideas. "core" provides a web-based front end to fourq actions to make life easy for FarCry developers.

# 3.11 EXTENSIBILITY

FarCry CMS is more than a content management solution. FarCry is a web application framework that is well suited to the construction of web applications, especially those dealing with content. The framework provides a range of integration and customisation options.

## 3.11.1 Templates & Webskin

Developers can implement any look and feel through the FarCry presentation engine. Templates are simplified further by being self-registering, and supplemented by a custom tag library for truly painless deployment. Template construction only requires good HTML skills and a passing knowledge of ColdFusion.

## 3.11.2 Included Objects

The "include" content type allows developers to run any block of ColdFusion code within the template engine of FarCry. For example, the sitemap and search pages within the sample applications are both Include objects.

## 3.11.3 Extending the webtop

The FarCry web-based administration is customisable. Developers can supplement the existing functionality with their own administration interfaces as they require. The security model, audit and other services can all be used to extend the administration area.

## 3.11.4 Custom Content Types

Completely new and unique content types can be built and deployed within FarCry, leveraging the same core services available to core types.

## 3.11.5 Custom Publishing Rules

Developers can build publishing rules to supplement the existing library of publishing rules within the system.

# 3.12 THIRD PARTY PLUG-INS

FarCry leverages many third-party code libraries, to provide a best of breed solution. These plug-ins are optional but recommended. FarCry provides out-of-the-box integration for following solutions and is adding more as the community requires. These libraries are isolated from the primary code base as they may be distributed under alternative licenses and/or their source code is managed by a separate project. The FarCry milestone builds released by Daemon provide commercial support for the integration specific versions of these code libraries.

## 3.12.1 Rich Text Editor Support

v4.0 FormTools provides support for tinyMCE and is considered the editor of choice, as such it is the only editor shipped with official FarCry installation packages. Of course developers are still able to plugin a different editor if they so desire.

## 3.12.2 Javascript Libraries

FarCry uses a variety of JavaScript libraries for client side validation in forms and wizards, and other UI effects. FarCry distribution currently includes Prototype, mootools, YUI, extJS and qForms libraries.

## 3.12.3 geoLocator

geoLocator is a Java/ColdFusion library used to lookup country code and language from a visitor's IP address. It uses a local copy of the WHOIS database to perform fast, accurate lookups of country codes. geoLocator is useful for log analysis, internationalisation, geolocation, and more.

geoLocator is maintained by Paul Hastings (http://sourceforge.net/projects/javainetlocator/).

## 3.13 CODING STANDARDS

The following variable naming conventions are used within FarCry:

Variable Type	Prefix	Example
Structure	st	stObj
Array	а	aObjects
List		lObjects
Integer	i	iAmount
Boolean	b	bFound
Object	0	oTree

As of v4.0 variables, filenames and directories are preferred to be in lowerCamelCase. Although ColdFusion (CFML) is not a case sensitive language, FarCry is available on any platform that ColdFusion can run on including the \*nix operating systems where case sensitivity with filenames and directories is enforced.

For this reason a concerted effort is being made to standardise naming conventions. The lowerCamelCase states that words are joined without spaces and the first letter is capitalised (with the exception of the first word which is all lower case).

For example:

displayPageStandard.cfm types.cfc

The use of <cfsetting enablecfoutputonly="true"> is highly encouraged to reduce "white space" in the resulting HTML source code. Only code that is to be generated as HTML should be inside <cfoutput> tags.

As the use of these white space techniques can make the code quite complex this course (for the sake of simplicity) will not go to great lengths to reduce white space, but it should be noted that this is regarded as a best practice and should be implemented in a real world production situation.

# 4 The "webtop": Administration Interface

To login to the FarCry Administration Interface for the first time use the following details:

Username: farcry Password: farcry

# 4.1 HOME TAB

The Home Tab used as an overview/reporting page for your site. Here you will find information like your profile details, objects you have locked or waiting approval, recently added pages and a number of graphs to give you an overview of site.

Daemon Generic -	Content Items Pending Approval Content Type: All Content Type: S				
on - tment	CONTENT ITEM	CREATED BY		AST UPDATED	
-	Currently Non In [pending] Status				
- en_AU SETTINGS	Content Items In Draft Content Type: All 🛛 🖌 Latest: 5 💌				
	CONTENT ITEM	CREATED BY	LAST UPDATED		
	Careers at Daemon	daemon	Monday 05 June 2006 8	3:29:08 PM EDT	
	The Daemon Sandpit	daemon	Monday 05 June 2006 8	3:28:28 PM EDT	
	Macromedia User Groups	daemon	Monday 05 June 2006 8	3:27:46 PM EDT	
	Solutions	daemon	Tuesday 30 May 2006 1	2:55:25 AM EDT	
	Web Developer Role	daemon	Friday 26 May 2006 12:	41:03 AM EDT	
	Items you have locked	Түре	CREATED BY	LAST UPDATED	(and a state of the state of th
	(incompleta)	dminime	daemon	5 June 2006	[unlock]
	(Electripiere)	animago	dapmon	30 May 2000	[childen]
	Web Developer Role	depHTM			I I IDIOCKI
	Web Developer Role FarCry CMS Annual Support Packages	dmHTML dmHTML	daemon	20 Way 2000	[unlock]

#### 4.1.1 Your Profile

Displays your currently profile details including name, email and locale. Each user in FarCry has an associated profile. An email address is required if you wish to make use of the workflow reporting system within FarCry, which notifies you when your object has changed status, or someone is requesting your approval for one of their objects.

#### 4.1.2 Edit Profile

Editing your profile allows you to update your personal details including name and email address. It also allows to you to choose if you want to receive email notifications and what locale you belong to. The default locale is en\_AU (Australian English). Changing this to will affect the way dates are formatted, text direction etc within the FarCry admin system, and if a translation file is present it will also present the admin system in your local language.

#### 4.1.3 Change Password

This allows you to change your login password. Passwords should be changed on a regular basis for security reasons.

## 4.1.4 Objects Waiting Your Approval/Draft/Locked

This will show as a table in the centre column of the page. Listed will be objects that are awaiting your approval. Each item is linked to the object so to allow quick viewing/approval.

## 4.1.5 Objects You Have Locked

This will also show as a table in the centre column of the page if you have any objects that are locked and preventing other users from editing. Objects are locked automatically when you begin the edit process and if for some reason you leave the edit process without saving the object will remain locked. You can unlock each item by simply clicking the "[unlock]" link in the table.

## 4.1.6 Customised Overview

The entire webtop overview for any content type can be replaced and customised.

# 4.2 SITE TAB

The site tab is where you manage the structure of your site and all navigational content.

daemon serious web solution	V25		FarCry Control Logged in: Daeme
Overview Site Conten	Admin Reporting	Security	
stre V V kk Zeem: Hone V V Sectors Web Solutions Sectors Web Solutions Sectors Studies About Us	APPROVED Content Rem Title: Content Rem Type: Ravigation Alias(es): Created by: Date Created: Locking: Last Updated: Last Updated By: Current Status: ObjectID • <u>Store Corrent Frender U</u> • <u>Store Corrent Frender U</u> • <u>Store Corrent Frender U</u> • <u>Store Corrent Frender U</u>	Home Navigation home farchy 1906/005 Unitocked 1906/2005 farchy approved 96F1F41F-EA69-0EC7-F55A923A8431E51D REL Minister	Eit this content item Senditiis content Bern back to draft Create Navigation Create Navigation Create Include Create Include Create File Create File Create File Create CSS Create Link Proview

#### 4.2.1 Home

The Home tree structure is the main navigational structure for your site, from your home page down. Use this tree to:

- add/edit pages
- change hierarchy structure
- set security permissions for content
- approve/request approval/send back to draft
- view page/branch statistics
- send to trash/delete pages
- view dump of page contents
- preview page

#### 4.2.2 Utility

Offers the same functionality as the Home tree but is generally used for content that does not fit into the main site hierarchy, eg search pages, utility menus etc.

#### 4.2.3 Trash

All objects that have been "Sent to Trash" are stored here. Objects can be moved back into the main tree or deleted permanently from the system.

#### 4.2.4 Quick Zoom

A drop down list of navigation points that when selected quickly zooms to the selected point in the tree. Values are added to the drop down list if they have a nav alias set.

## 4.2.5 Refresh

Clicking the refresh icons refreshes the tree without reloading the other frames.

## 4.2.6 Prune and graft, copy and paste

Navigation nodes can be moved, copied or pasted onto other navigation folders in the tree to rearrange the site hierarchy.

#### 4.2.7 Overview

The overview screen show basic object details along with a number of options depending on object type, status and security permissions. These options include:

- Edit
- Approve/Request Approval/Send back to draft
- Create draft version
- Add comment
- View comments
- Delete
- Send to trash
- Preview
- Dump
- Create new object (if object being viewed is a Navigation Node)

#### 4.2.8 Edit

The edit tab calls the edit method for the object, automatically locks the object and presents the user with the edit form, usually in the form of a one page form or a multi step form called a PLP (Process Logic Path). Completing the edit form unlocks the object, updates its contents and returns you to the overview screen.

#### 4.2.9 Archive

The archive screen lists all previous versions of the object. If an object is approved you can create a draft version of the object whilst keeping the live version, allowing you to make changes without effecting the live site. When the changes are complete you can send the draft version live, replacing the current live version with the draft version. The existing live version gets archived and will be listed under the archive tab. You can roll back to archived versions at any time.

## 4.2.10 Audit

The audit tab lists all activity against the object. Each audit log shows the date of the activity, what the activity was eg create, update etc, the location it was edited from (IP address), any notes and the user who performed the change.

#### 4.2.11 Stats

The stats tab shows a number of statistical graphs based on the number of views the object has had. There are three graphs:

- Views per hour of the last 3 days
- Views per day of the last 4 weeks
- Views per day between a start and end date (user entered)

If the object being viewed is a navigation node, then the statistics shown will be generated from all objects under that node (called a navigation branch).

## 4.2.12 Properties

The properties tab is primarily for developers and shows a dump of actual values stored in the database for that object.

# 4.3 CONTENT TAB

The content tab is primarily used for adding content that does not sit within the site tree hierarchy. This could include things like news items, facts, links and events. Typically this kind of information is listed on a page via a container and publishing rule and may be filtered on publish dates and categories. For example, only show news items that are currently published and are in a category called "New Destinations". This is known as dynamic content. Different objects may appear on the page without any changes to code or the page itself.



The content tab is split into four sub sections:

## 4.3.1 Media Library Content

The Media Library allows content editors to add and edit images, files and flash movies within their project from a centralised location. Images for example would leverage the Image content type

The Media Library is available to content editors and is an easy choice for accessing a central repository for your project assets.

#### 4.3.2 Site Tree Content

This area provides a summary of all "HTML" pages (dmHTML) and "Included" content objects. If you have a large site this can be a handy way to find a page you need to edit, or to give you an indication of the number of these content types in your project.

## 4.4 CONTENT UTILITIES

#### 4.4.1 RSS Feeds

FarCry has a utility to generate RSS feeds from any object type. Multiple RSS feeds are permitted and each feed is generated via a simple edit form that includes standard RSS properties. Users can select which object type to use, how many objects to be listed and filter by category if needed.

The RSS feed is generated when the user clicks on the "Preview" button. An xml file will be generated (using the name given in edit form) and saved in the <a href="https://www.xml"><a href="https://www.xml">applicationName</a>/www/xml</a> folder.

The validate button is used to validate the RSS feed against standards. This function is only available if the resulting xml file is available on the Internet, i.e. not when using localhost.

#### 4.4.2 Export

The Export utility generates an xml file of all the content in the system for the selected content type and sends it via email to the address entered.

#### 4.4.3 Manage Keywords

Categorisation allows users to administer the site's Category tree. A category tree is defined so that site content can be grouped into similar topics. Eg you may have a page in your site called "Destinations". On this page you want to list all of your destinations but also to show news related to destinations and a random fact on one of your destinations listed. To do this you would create a category item called "Destinations" and then assign the appropriate news and fact items with this category. Finally on your Destinations page you would use the News Rule filtered by the Destinations category and the similarly for the Random Fact Rule to generate only Destination related information.

The category tree can be hierarchical and categories can be moved within the hierarchy. Categories can also have an alias assigned, allowing for developers to easily reference a category's objected via the application.catid structure.

#### 4.4.4 Shared Containers

Shared Containers allow a content editor to re-user a publishing rule across multiple containers, so that if a change is made to one container/publishing rule that change flows through to all 'shared containers' automatically.

#### 4.4.5 Quick Site Builder

This feature is a handy utility for quickly generating a navigation structure within your project and is often used during the initial site development stage. Users are able to choose:

- where in the tree the structure should be built
- what status the items should be given
- an option to create dmHTML items with the same title beneath each navigation node and the option to choose which display template to use for the dmHTML items
- an option to create nav Aliases for navigation nodes to a specified level down
- a text are box where users enter the new navigation structure with each line representing a new item and using as a level token to indicate the item should be at a lower level.

Eg

Item 1 -Item 1.2 --Item 1.2.1 -Item 1.3 Item 2 -Item 2.1 --Item 2.2 Item 3

#### 4.4.6 Category Tree Quick Builder

The Category Tree Quick Builder is a similar utility to the Quick Site Builder but used for the category tree.

#### 4.4.7 Bulk Image Upload

This feature allows users to quickly insert a number of images into the site structure. Users are able to:

- Choose where in the site the images should be loaded to (defaults to the global image library)
- Allows a user to upload a zip file containing all the images

#### 4.4.8 Bulk File Upload

A similar utility to the "Bulk Image Upload" but for file objects.

#### 4.4.9 Archive Utilities

An overview of all archived content within the current project.

## 4.5 FARCRY CMS PLUGIN

#### 4.5.1 News

Clicking the "News" link will show a listing of News (dmNews) objects coming from the "farcrycms" plugin. The dashboard overview page is common to Dynamic Content types in FarCry and is called "Object Admin". Users may add/edit/delete content objects and are provided a set of tools to search for a particular object to perform an action on.

farcry tell it to sol	meone who	cares						FarCry	Logged in: (Logout   <u>View</u> )
Overview Site	Conten	nt Admi	n Rep	orting Security					
Dynamic Content CONTENT TYPE: * News * Events * Links * Eactor	_	News Listing Filt Add Dele	Admin er: <u>set</u> tte Reques	istration	ft Properties Unlock				
7 1000		Displayin	g 1-1 of 1 r	esults					
		Select	Status	Action	Title	News Category	Publish Date	Last Upda	ted
					<b>v</b>		desc 💌		<b>~</b>
		1	draft	– action – 💌	Latest News Item		18-Feb-07	18-Feb-0	7
		Displayir	g 1-1 of 1 r	esults					

These filter tools limit the results shown by selectively 'searching' on a field type like the object title, or body content. Users are also able to order the results by any record in either descending or ascending order on the title, publish date or date last updated.

The results are shown by default with the following fields:

- Select (a checkbox used to select the object and perform a function like Delete)
- Status (an indication of the status for that particular content object)
  - Action (a dropdown box to perform certain actions for a content object)
    - Overview (a link to overview page for that content object)
    - Edit (a link to edit the object. This link is only active if the object is unlocked and the user has permission to edit the object)
    - View (a link to preview the object in a new window)
    - Request Approval
    - Approval
- Title (the object's title as entered by a content editor)
- News Category (only displays if a news item has been categorised)
- Publish Date (when the news item was published)

• Last Updated (shows the date the object was last updated)

Functions users are able to perform from Object Admin:

- Add
- Delete
- Request Approval
- Approve
- Send to Draft
- Properties
- Dump

These buttons are displayed only if the user has appropriate security permissions.

#### 4.5.2 Events

All the functionality as described for News, but lists dmEvent objects.

## 4.5.3 Facts

All the functionality as described for News, but lists dmFact objects

## 4.5.4 Links

All the functionality as described for News, but lists dmLink objects.

## 4.6 ADMIN TAB

The admin tab is used to perform various administration tasks from modifying a sites configuration, clearing cached objects to deploying/updating system types and rules. The Admin tab is by default only available to Site Administrators.

daemon serious web s	olutions				FarCry	Logged in: Daemon (Logout   View)
Overview Site Co	ntent Admin	Reporting	g Security			
Ceneral CONFIGURATION * Config Files • Custor Config • Dump Config CUSTOM ADMIN XML	Schedul Filter by: Cateo Displaying 1-2 Add Delete F	ed Task arry Keywor of 2 results. Properties Unit	ks Adminis d <u>Date range</u> No Filters Applie ock	tration d		
Dump Custom Admin XML	Select	Edit	Run Task	Label	Last Updated	Ву
SCOPE DUMP		1		Verity Content Indexing	8/08/2005	daemon
CACHE MANAGEMENT	Add Delete F	Properties Unio	Dock 2	Empty Trash	8/08/2005	daemon
<ul> <li>SCHEDULED TASKS</li> <li>Scheduled Tasks</li> </ul>						
MESSAGE CENTRE Messages						
MARAGE FRIENDLY URLS * Manage Friendly URLs * Rebuild Friendly URLs						
					Copyright © Daem	on 1997-2006, FarCry 3.0.2RC

#### 4.6.1 Config Files

This screen lists all configuration files within the system, both default and custom defined. Clicking a config file will take you to the edit method for that config, and allows you to update the various elements of the config. Clicking the "Update Config" button saves the configuration to the database and updates the values in the application scope. All config values are accessible via the application.config.[configName] structure eg application.config.general.siteTitle by will return "FarCry" after a new install. The default configs installed by FarCry are:

- FU Settings (Friendly URL settings including domains and URL patter)
- · General (general elements for the site include title, admin email and which text editor to use)
- overviewTree (configurable html strings for insert statements from the tree, eg insert Image)
- plugins (define which FarCry plug-ins the site uses eg Friendly URLs)

#### 4.6.2 Custom Config

Lists all custom config files and has the option to deploy the config if not already deployed, otherwise the option to re-deploy (restoring to its default values) is displayed.

#### 4.6.3 Dump Config

Primarily used by developers who need to quickly see the contents of all config files. Each config is presented as a separate cfDump structure.

## 4.6.4 Dump Custom Admin XML

Primarily used by developers who need to quickly see the xml structure of the Custom Admin definition file.

#### 4.6.5 Scope Dump

A Flash utility that allows users to dump the contents of four ColdFusion scopes:

- Application
- Server
- Request
- Session

#### 4.6.6 Cache Summary

The Cache Summary screen displays a listing of all cache blocks within the system. Caching is used to enhance performance of a website and each cache is as assigned to a cache block to help with the management of caching.

Each cache block is displayed with the following information:

- The name of the cache block which is linked through to a cache detail screen which lists each individual cache within that block
- Number of caches (displays the number of caches belonging to the cache block)
- Expired caches (displays the number of caches within the cache block that have passed their expiry date)
- Clean (a checkbox that when selected and the clean function run will remove all the expired caches within the block)
- Flush (a checkbox that when selected and the flush function is run will remove all caches from within the block)

The cache details screen presents the following information:

- The name of the individual cache. When clicked the contents of the cache will be displayed in a new window
- Timeout Period (how long the cache is set to last for (days:hours:minutes:seconds)
- Will Expire (the date/time at which the cache is due to expire)
- Flush (a checkbox that when selected and the flush function is run will remove selected cache)

#### 4.6.7 Auto Cache

The Auto Cache functional is way to automatically cache all pages within your site. A cache is only created when a page is first viewed so what this function does Is start at your home page and then progressing down the tree views each page in turn, therefore creating caches as it goes. When the utility is finished you will see all your cache blocks and individual caches in the Cache Summary screen.

#### 4.6.8 Scheduled Tasks

The Scheduled Tasks screen is a way to generate ColdFusion scheduled task from within FarCry. Users can choose from a number of default scheduled tasks or can create their own tasks by saving a file under the <a price the complexity of the comple

The default scheduled tasks are:

- Empty Trash (deletes all items under the Trash node)
- Statistics Overview Report (emails the Statistics overview report to a specified email address)
- Unlock Objects (unlocks any locked items)
- Verity Update (updates all verity collections)
- XML Feed Update (recompiles all RSS Feeds)

#### 4.6.9 Rebuild Friendly URLs

If the Friendly URL plug-in has been set up, users can reset all Friendly URLs using this tool. The function first deletes all existing Friendly URLs and then goes through the navigation tree structure, starting from the home node and working down, creating a Friendly URL for each node.

#### 4.6.10 Manage Friendly URLs

Clicking the Manage Friendly URLs link produces a list of all Friendly URLs defined in the system. The list can be filtered by either the alias name or an objectid. Users can delete individual Friendly URLs by selecting the URL's checkbox and clicking on the "Delete" button at the bottom of the screen.

# 4.7 FARCRY VERITY PLUGIN

## 4.7.1 Manage Collections

The Manage Collections link produces a list of verity collections in the site. Each collection has the following information/links:

- Collection Name
- Last Updated date
- Update link to update the collection's index
- Optimise link to optimise the collection's index
- Delete to remove the collection from the server

# 4.7.2 Build/Update All Collections

This utility loops through the verity config file and produces a collection for each object type that has been selected for indexing. If the collection already exists the function will update the existing collection.

## 4.7.3 Optimise All Collections

This utility optimises all verity collections within the site.

## 4.7.4 Verity Config

Displays the verity configuration file (same screen as config)

## 4.8 COAPI ADMIN

#### 4.8.1 Type Classes

The Type Classes screen is a very powerful utility used by System Administrators to perform the following tasks without the need to edit the database manually:

- Deploy new types to the system (creates the database table and updates the application scope)
- Deploys new properties to an existing type (adds new field to the database table and updates the application scope)
- Renames existing properties (renames the field in the database and updates the application scope)
- Deletes old properties (removes the field from the database)
- Changes property types eg from string to text (changes the field type in the database and updates the application scope)

#### 4.8.2 Rule Classes

Similar to the Types Classes screen but Rules based.

## 4.8.3 COAPI Metadata

Dumps the application.types and application.rules structures. Displays the following information:

- Type/Rule general information such as name, hint
- All the type/rule methods and arguments
- All the type/rule properties, including name, type and required information

#### 4.8.4 COAPI Schema

A graphical layout of all database tables in the system and their field information (Internet Explorer only)

## 4.8.5 Orphaned Nodes

A utility that searches for objects within the nested tree structure that have no parent objects. If objects are found it provides the user with an option to re-attach the objects to the tree in a specified location.

## 4.8.6 Fix Tree Levels

This utility is used to ensure that the nested tree structure is valid and all tree values including left, right and level are accurate.

## 4.8.7 Rebuild Tree

Rebuild Tree searches the nested tree for objects that have references to objects that no longer exist. If objects are found they can be removed from the database.

## 4.9 SECURITY TAB

The Security area is used by Site Administrators to perform various security related tasks such as creating/updating users, creating new policy groups, mapping policy groups to user groups, add new permissions, assigning permissions to policy groups and to view the permissions applied to the site navigation tree.

The FarCry security system works with:

- At least one User Directory.
- A User Directory contains Users and User Groups
- Users belong to User Groups
- User Groups are mapped to Policy Groups
- Permissions are set against Policy Groups.

This allows for multiple User Directories within the one FarCry site eg the FarCry default user system and an active directory setup. Each User Directory can have their own set of User Groups but they all map to the one set of Policy Groups within FarCry, meaning that permissions only have to be set at one level.

There are two types of permission within FarCry by default, Policy Group and Navigation. Policy Group permissions are static permissions that are mainly used in the administration interface. Eg whether a user can view the Admin tab or can work with dynamic content. Navigation permissions on the other hand are directly related to the navigation tree. The navigation tree has permissions such as view/edit/approve. Navigation nodes inherit permissions from their parent node unless otherwise set.

daemon serious web sol	olizions	FarCry (Locout   View)
Overview Site Cont	ntent Admin Reporting Security	
User Management	User Search	
Test Security Setup      USERS     Search for User     Create a User	Select a user directory:	
GROUPS + Search for Group + Create Group	Fragment of the userLogin: Starts With 💌	
	Search	

#### 4.9.1 Test Security Setup

Performs a test to ensure the correct security database tables are in place.

#### 4.9.2 Search for User

This page is used by administrators to search for existing FarCry users. Once a user has been found their details can be updated:

- change password
- change user status (active/not active)
- update user notes
- manage the user's groups (assign to new groups or remove existing assignments)
- delete the user

#### 4.9.3 Create a User

An administrator can create a new user using this screen by entering a user login and user password. Once created the same options are available as for search for user.

### 4.9.4 Search for Group

A search utility for FarCry Groups. Once found, groups can either be updated or deleted from the system.

#### 4.9.5 Create Group

Creates a new User Group with FarCry.

#### 4.9.6 Test Policy Setup

A utility that test that the needed Policy Store tables are in the database.

## 4.9.7 Policy Groups

A listing of all FarCry Policy Groups and their descriptions.

#### 4.9.8 Policy Group Permissions

This screen is where permissions are set for each Policy Group within the system. There is a drop down box at the top of the screen with all the Policy Groups. Each permission is listed and its current state for the selected Policy Group shown (either Yes/No). Permissions states can be changed and then saved by clicking the "Update" button.

#### 4.9.9 Create Policy Group

A form that allows administrators to add new Policy Groups to the system.

## 4.9.10 Copy a Policy Group

Administrators can use this tool to copy an existing Policy Group, including all its permission values, into a new Policy Group.

#### 4.9.11 Show Policy Group Mappings

A listing of relationships between FarCry Policy Groups and User Groups.

#### 4.9.12 Map Policy Group

A utility for administrators to map a User Group from a list of User Directories to a FarCry Policy Group.

#### 4.9.13 Permissions

A listing of permissions based on permission type (dmNavigation or PolicyGroup). Clicking the permission name will display an edit form where the permission name and notes can be updated or the permission deleted from the system.

#### 4.9.14 Create Permission

A form that allows administrators to create new permissions. Values required are permission name and permission type.

#### 4.9.15 Permissions Map

Displays the navigation structure with all places where a permission setting has been set manually (i.e. not inherited) linked to the edit navigation node permissions screen where permissions can be modified.

# 4.10 REPORTING TAB

The reporting section within FarCry presents a wide range of statistics on site views, sessions and searches. An entry is logged into the stats system each time a user requests a page with the following information stored:

- Pageid (ObjectId of the page viewed)
- Navid (ObjectId of the navigation parent of the page viewed)
- UserId (if the user is logged in then the FarCry username is stored, otherwise "anonymous")
- Remotelp (the IP address where the user is browsing from)
- logDateTime (the time and date the page was viewed)
- sessionId (the unique sessionid of the user)
- browser (the type of browser the user working with)
- referrer (the page referrer)
- locale (the user's locale code)
- OS (the user's operating system type)



#### 4.10.1 Overview Report

Provides a quick statistical overview in the time frame selected. Users can choose from the following time periods:

- Today
- Last Week
- Last Month
- Last Quarter
- Last Year

The total number of user sessions during the specified period is shown at the top of the screen and then the top 10 results of the following categories are shown:

- Pages
- Locales
- Browsers
- Operating Systems
- Referrers
- Searches

#### 4.10.2 View Summary

The view summary allows users to view the most popular objects in a defined period. The search can be filtered by:

- object type (defaults to all types)
- date (today, last week, last month, last quarter, last year, all)
- Rows (number of results returned (default to 20)

Click the "More Details" link produces the standard three-graph representation:

- Views per hour of the last 3 days
- Views per day of the last 4 weeks
- Views per day between a start and end date (user entered)

#### 4.10.3 Referrer Summary

Lists the most popular referrers with the option to filter by date, number of rows returned and remove internal site references.

#### 4.10.4 Locale Summary

Lists the most popular locales by country name and language with the option to filter by date and number of rows returned. A map of the world is also displayed with locales highlighted and number of sessions displayed in a rollover window.

#### 4.10.5 Operating System Summary

Lists the most popular operating systems with the option to filter by date and number of rows returned.

### 4.10.6 Browser Summary

Lists the most popular browsers with the option to filter by date and number of rows returned.

#### 4.10.7 Session Summary

Displays the standard 3-graph representation of sessions.

#### 4.10.8 Session Paths

Displays a listing of recent visitors that can be filtered by date, number of rows returned and IP address. Information displayed:

- IP address
- Date viewed
- Number of pages viewed
- Link to "View Path" (displays a listing of all pages user has viewed in that session with the amount of time spent on each page also shown.)

#### 4.10.9 Who's on Now

A list of all current users on the site. Information displayed:

- IP address
- Locale
- Been active for (time in minutes)
- Last Activity (time in minutes)
- Pages viewed
- Link to "View Path" (same screen as described in Session Paths)

#### 4.10.10 Google Key Words

A listing of the most popular Google keyword searches with the option to filter by date and the number of rows returned.

#### 4.10.11 Recent Searches

A listing of recent in-site searches with the option to filter by date and the number of rows to return. Results are displayed as:

- Search String
- Results (number of results returned)
- Date/Time
- Locale
- Remote IP

#### 4.10.12 Site Searches Returning No Results

A listing of in-site search strings that returned no results with the option to filter by date and the number of rows returned.

#### 4.10.13 Most Popular Site Searches

A listing of the most popular in-site search strings with the option to filter by date and the number of rows returned.

#### 4.10.14 Clear Stats Log

Deletes all stats related information from the database.

#### 4.10.15 All Logins

Shows details for FarCry logins. Details displayed are:

- Date
- Location (IP address)
- Note
- User

## 4.10.16 Failed Logins

A listing of all failed login attempts. Details displayed are:

- Date
- Location (IP address)
- Note
- User

## 4.10.17 Daily User Login Activity

A graph displaying the number of logins per hour for the last 3 days.

## 4.10.18 Weekly User Login Activity

A graph showing the number of logins per day of the last 4 weeks.

## 4.10.19 User Activity

A utility that lists all database activity by FarCry users. Results can be filtered by user, activity type (eg login, delete) and the number of rows returned. The results are shown as:

- Date
- Activity
- Object (linked to a preview of the object)
- Note
- User
- Location (IP address)

# 5 Webskins

A webskin is the design template used to display an object. It is the presentation layer of our application. Each FarCry type can have one or more webskins to allow for multiple ways to view the same data. A webskin is usually a combination of HTML/CSS for presentation and ColdFusion for any logic and outputting of dynamic values. Each webskin template has the stObj ColdFusion structure available to it. This structure contains all the values from the database for the object being viewed.

In this unit we will create our own set of webskins for the Solar Safaris website. We will design the following:

- Home Page
- Standard Page
- News Page

At the end of the unit we will have designed a site that looks like:



🗀 C:\farcry\solar_sa	afaris\webskin		<u>_ 🗆 ×</u>
File Edit View F	avorites Tools Help		27
😋 Back 🔹 💮 🗸	🏂 🔎 Search 🔀 Folders 🕼	s 迩 🗙 🍤 🖪	<b>.</b> -
Address 🛅 C:\farcry\	solar_safaris\webskin		💌 🄁 Go
Name 🔺	Size Type	Date Modified	
🛅 dmEvent	File Folder	14/01/2005 3:20 PM	
🛅 dmFacts	File Folder	14/01/2005 3:20 PM	
🛅 dmHTML	File Folder	14/01/2005 3:20 PM	
🗀 dmInclude	File Folder	14/01/2005 3:20 PM	
🛅 dmLink	File Folder	14/01/2005 3:20 PM	
🗀 dmNavigation	File Folder	14/01/2005 3:20 PM	
🗀 dmNews	File Folder	14/01/2005 3:20 PM	
🗀 includes	File Folder	14/01/2005 3:20 PM	
8 objects		0 bytes 🛛 😼 My Co	mputer //.
5.1 MOLLIO			

Mollio is a sample application that comes with the FarCry installation package. It has its own set of webskins that are cleverly designed using HTML and CSS. We could use the default Mollio templates and our site would look great and conform to a number of web standards (try browsing the site now http://localhost/solar) but often designers want a completely custom design which means starting from scratch. That is what we are going to do now.

Open Windows Explorer and browse to C:\farcry\projects\solar\_safaris\webskin

You will notice there is a directory for each of FarCry's content types. Open the dmHTML directory and you will see the following files.

😂 dmHTML					
<u>File E</u> dit <u>V</u> iew F <u>a</u> vorites <u>T</u> ools	Help				<b>.</b>
🚱 Back 🔹 🕥 🕤 🏂 🔎 Se	earch 😥 Folders 🛄 🗸				
Address 🛅 D:\farcry\trunk\projects\sola	ar_safaris\webskin\dmHTML				🗸 🔁 Go
	Name 🔺	Size	Туре	Date Modified	
File and Folder Tasks 🛛 🍣	🖬 displayPageTypeA.cfm	2 KB	CFM File	18/02/2007 12:06 PM	
	🔟 displayPageTypeB.cfm	2 KB	CFM File	18/02/2007 12:06 PM	
Other Places 🛛 😵	displayPageTypeC.cfm	2 KB	CFM File	18/02/2007 12:06 PM	
	displayPageTypeD.cfm	2 KB	CFM File	18/02/2007 12:06 PM	
Details 🛛 😵	displayPageTypeE.cfm	2 KB	CFM File	18/02/2007 12:06 PM	
	displayPageTypeF.cfm	2 KB	CFM File	18/02/2007 12:06 PM	
	displayTeaserTypeA.cfm	2 KB	CFM File	14/02/2007 11:21 PM	
	displayTeaserTypeB.cfm	2 KB	CFM File	14/02/2007 11:21 PM	

There are two types of files in this directory.

- 1. displayPage\* (these are the primary webskin templates and are used when viewing the full object, either via a preview or when clicking a link to the object within the site)
- 2. displayTeaser\* (these are webskin templates used in containers. They usually display just a short amount of information about the object with a link through to the full display)

To start working with a clean slate we need to delete all the Mollio webskins. To do this, go back up to the webskin directory. Select all directories and delete them. There should now be nothing in your webskin directory.

🚞 C:\farcry\solar_sa	afaris\webskin			
File Edit View F	avorites Tools I	Help		20
🕞 Back 🝷 💮 🝷	🏂 🔎 Search	n 😥 Folders 🛛 😭	s 🎯 🗙 🍤	•
Address 🛅 C:\farcry\;	solar_safaris\webski	n		💌 🄁 Go
Name 🔺	Size	Туре	Date Modified	
0 objects			0 bytes 🛛 😼 My	Computer

## 5.2 HOME PAGE

The home page for Solar Safaris is a dmHTML object. This means we need to create a webskin for the dmHTML type. To do this create a new file in your preferred code editor and save it as C:\farcry\projects\solar\_safaris\webskin\dmHTML\displayPageHome.cfm

#### 5.2.1 Hello World

Add the following code to the page to produce a standard "Hello World" page.

<cfoutput></cfoutput>
HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
<html></html>
<head></head>
<title>Untitled</title>
<body></body>
<h1>Hello World</h1>

Save the file.

If you try and view the site now you will get an error telling you that the template was not found. This is due to the homepage being set to use the displayPageTypeA.cfm file by the install process. We need to log in and change which template our home will use.

Perform the following tasks:

- Login to FarCry
- Click on the "Site" tab

- Select the page under the "Home" node "farcry open source"
- Click the "Send object back to draft" link in the right hand pain (you can't edit an approved object)
- Add a brief comment such as "home page re-design" and click the "Submit" button
- You are now back the object overview page and you will notice there is now the option to edit the page. Click the "Edit this content item" link.
- You will notice for "Display Method" there is a message "No Content Templates Available". This is because FarCry looks for a special line in each webskin template, which gives a short title for the template.
- Add the following line to your displayPageHome.cfm file (at the top of the page is best)

## <!--- @@displayname: Home Page --->

- Save the file and then refresh the edit frame in FarCry (you may need to updateApp)
- Select "Home Page" from the "Display Method" drop down list.
- Change the title to "Welcome"
- Click "Save"

Start	Ne	xt 🔎
- General Details	Start Body	
Title :       Welcome         Review Date :       ✓ remove         18 Feb 2050 12:00 AM       III         Display Method :       Home Page ♥	Categorisation	
Metadata	Cancel	I
Meta Keywords : Extended Metadata : show		
	Ne	xt 🔎

Refresh the homepage of your site (http://localhost/solar)



This is a very simple webskin. Obviously not much use to us if we want to display dynamic content but it is a basic form of a webskin. Now let's go ahead and add some dynamic output to the webskin.
# FarCry Developer Course

# 5.2.2 Dynamic Content

Add the following line code to your displayPageHome.cfm file just under the <h1> tags

<cfdump var="#stObj#">

Refresh your home page. The dump shows all the values for our "Welcome" page:

🖉 Untitled - Microsoft Inte	ernet Explorer
File Edit View Favorite	s Tools Help 🔐
🕞 Back 🔻 🛞 👻 💌	😰 🏠 🔎 Search 🤺 Favorites 🤣 🔗 - 🍃 📧 🛛 🕁 🎇
Address 🤄 http://localhost/s	iolar/ 🔽 ラ Go
Hello Wor	ld
struct	
AOBJECTIDS	array [empty]
ARELATEDIDS	array [empty]
BODY	<p>Built from the ground up on the revolutionary ColdFusion MX server platform, <strong>farcry</strong> is the affordable, powerful site management solution that is quick to implement, and intuitive to use.</p>
COMMENTLOG	farcry(14/01/2005 15:48:54): Status changed: approved -> draft home page re-design
CREATEDBY	farcry
DATETIMECREATED	2005-01-13 14:09:18.0
DATETIMELASTUPDATED	2005-01-14 15:48:54.0
DISPLAYMETHOD	displayPageHome
EXTENDEDMETADATA	[empty string]
LABEL	Welcome
LASTUPDATEDBY	farcry
LOCKED	0
LOCKEDBY	[empty string]
METAKEYWORDS	[empty string]
OBJECTID	6A0BE034-20E0-64AB-AF0181341662AF7A
E Done	Local intranet

We can reference any of these values via stObj.fieldName eg stObj.title. Change your displayPageHome.cfm file to now output the page title and body.

Your file should now look something like this:

The resulting home page should look like:



#### 5.2.3 Basic Design

Ok now let's give the page a bit of a facelift by adding some basic styles and a header image.

First go to the C:\farcry\projects\solar\_safaris\www\css folder and delete the 3 style sheets in that directory. Also delete all files under C:\farcry\projects\solar\_safaris\www\wsimages (wsimages is short for "webskin images". This is where images used in the template design are stored. Images that are added to the system through FarCry are saved under the www\images folder).

Copy the "images\header.gif" file from the training CD into the C:\farcry\projects\solar\_safaris\www\wsimages directory. Copy the "style sheets\styles.css" file from the training CD into the "C:\farcry\projects\solar\_safaris\www\css" directory.

Then make the following changes to your displayPageHome.cfm file:

```
<!--- @@displayname: Home Page --->
<cfoutput>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
      <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
      <!--- import style sheet --->
      k rel="stylesheet" type="text/css" href="#application.url.webroot#/css/styles.css"
media="screen" />
      <!--- page title taken from current object's label --->
      <title>#application.config.general.sitetitle#: #request.stObj.label#</title>
</head>
<body>
<!--- set up main layout table --->
<!--- header image linked back to homepage --->
      <a href="#application.url.webroot#"><img
src="#application.url.webroot#/wsimages/header.gif" border="0"
alt="#application.config.general.siteTitle#"></a>
<!--- primary navigation code --->
                             Navigation to go here
                     <!--- search form --->
                             Search form to go here
                     <h1>#stObj.label#</h1>
      <!--- display body content --->
      #stObj.body#
      <!--- copyright info --->
             © #year(now())# Solar Safaris
      </body>
</html>
</cfoutput>
```

The above changes import a style sheet, add a header image and a placeholder for navigation and search functions and also add a footer displaying a copyright message.

Refresh your homepage, it should now look like:



#### 5.2.4 Site Title

You will notice the page title is "farcry: farcry – open source". By default the installer sets the general config item siteTitle to "farcry". To update this perform the following tasks:

- Login to FarCry
- click the "Admin" tab
- click the "Config Files" link
- select the "General" config file
- In the "siteTitle" field change "farcry" to "Solar Safaris" and change the siteTagLine to "For an adventure out of this world"
- Click "Update Config"

Solar Safaris For an adve	inture out of this world		FarCry Compared In michael
Overview Site Content	Admin Reporting	Security Safaris	
General	LOCALE	an All	<u> </u>
CONFIGURATION	LOGINATTEMPTSALLOWED	3	
Contig Files     Custons Contig     Dunic Contig	LOGINATTEMPTSTIMEOUT	10	
CUSTOM ADMIN XML	LOGSTATS	Yes	
+ Dunp Custon Admin XML	NEWSEXPIRYTYPE	d	
Application Dump	RICHTEXTEDITOR	FCKeditor	
+ Rule Dump + Type Dump	SESSIONTIMEOUT	60	
+ Utilty vs Core Dunp	SHOWFORGOTPASSWORD	Yes	
+ scope cump	SITELOGOPATH		
CACHE MANAGEMENT     Cache Summary	SITETAGLINE	For an adventure out of this world	
+ Auto Cache	SITETITLE	Solar Safaris	
SCHEDULED TASKS	TEASERLIMIT	255	
+ Scheskiled Tasks		C & Run 4/servers/cfusion/cfusion-earlcfusion-war/WEB-INF/c	
MESSAGE CENTRE + Messages	VERITYSTORAGEPATH		
MANAGE FRIENDLY URLS + Nanage Friendly LRLs + Rebuild Friendly URLs	Update Config		Copyright @Deemon 1987-2006, ParCry 3.0.2RC

Refreshing the homepage should now show the updated page title:



#### 5.2.5 Primary Navigation

FarCry uses the Nested Tree Model (NTL) for its navigation hierarchy. There are a number of functions available to the developer to return navigation elements very quickly with very little code. Some of the functions in the core/packages/tree.cfc include:

- getDescendants (returns all objects below the passed in object)
- getAncestors (returns all objects above the passed in object)
- getChildren (returns objects one level down from the passed in object)
- getParentID (returns the object directly above the passed in object)
- getSiblings (returns all objects with the same parent as the passed in object)
- getNode (gets all details of the object passed in)

To produce our primary navigation we will use the getDescendants method of the tree.cfc. As the tree component is often used it has been placed in the application.factory.oTree scope so that you don't have to create the object each time you want to call a method.

Add the following lines of code to your displayPageHome.cfm file at about line 26 (under the primary nav comment):

<!--- set up where clause filter ---> <cfset navFilter=arrayNew(1)> <cfset navfilter[1]="status IN (#listQualify(request.mode.lvalidstatus, """)#)"> <!--- get navigation nodes ---> <cfset qNav = application.factory.oTree.getDescendants(objectid=application.navid.home, depth=1, afilter=navFilter ,blncludeSelf=1)> <!--- dump results ---> <cfdump var="#qNav#">

The above code first sets up an array to pass into the getDescendants method as a filter. This filter is basically a "where" clause in a SQL statement. This particular filter makes sure that only objects that are in the current view mode are returned.

By default only "Approved" objects are shown. If you are logged in you can use the widget menu and select the "Show Draft" link to allow draft objects to be displayed in your site (only you will see them).



A quick tip for development is to set the viewing mode to draft manually so that you don't have to be logged in to see your site whilst it is under development. Add the following line to C:\farcry\projects\solar\_safaris\config\\_serverSpecificRequestScope.cfm

request.mode.IValidStatus = "draft,pending,approved";

"\_serverSpecificRequestScope.cfm" is a project config file enabling developers to both override the default FarCry request scope variables and add custom project request variables which will be available through the application.

The remaining code above is the actual call to the getDescendants method. The arguments being passed in are:

objectId (The objectid of the node you want to get the descendants for. In this case we are
passing in application.navid.home, which is equal to the objectid column in the navigation node
"home". This is a nav alias, and can be set when editing navigation objects)

Solar Safaris For an advecture out of t	his world	FarCry Q
Overview Site Content Admin	Reporting Security Safaris	
Hore 💌 🕸 💧 🖍 Navigati	ion Node details: Home	*Required
Home V Title:"	Home	
a Selloma Advanc	ed Options	
Symbolic Lin	k: None 💌	
B - Destination Nav Aliases:	home	
a - Mews		
Friendly URL:		
	OK Cancel	
	these these the	

- depth (how many levels down to you want to return. In this case we only want one level but if you were using some sort of drop down structure you would return multiple levels)
- aFilter (an array of where clauses)
- blncludeSelf (Boolean flag if you want to include the object you are passing in as part of the returned results)

The getDescendants method will return a query object, which we have called qNav. If you reload the home page you will see a dump of the returned query.

qı	query						
	NLEFT	NLEVEL	NRIGHT	OBJECTID	OBJECTNAME	PARENTID	TYPENAME
1	2	1	5	6A0BE41D-20E0-64AB- AFC2460FC222D91B	Home	6A0BE413-20E0-64AB- AF0DDC6CF2845B63	dmNavigation
2	3	2	4	6A0BE427-20E0-64AB- AF60FD20F2ACD451	Support	6A0BE41D-20E0-64AB- AFC2460FC222D91B	dmNavigation

You can see this query has everything we need to build a navigation menu. It has returned the results in order from the highest point in the tree down and has also returned the objectids and the objectnames for use to build a link structure.

Remove the cfdump code and add the following loop to display the navigation menu:

loop over navigation and display each node	
<cfloop query="qNav"></cfloop>	
<a href="#application.url.conjurer#?objectid=#qNav.objectid#">#qNav.objectname#</a>	
check if object is not the last returned object	
<cfif neq="" qnav.currentrow="" qnav.recordcount="">   </cfif>	

The above code simply loops over the returned query and creates a link for each record. The link points to the objectid of the record and displays the objectName field as the link text. A simple if statement is added to only add the | if it's not the last record returned.

Refresh your homepage. It should now look like:



#### 5.2.6 Quick Site Builder

Let's go ahead now and add some navigational structure to our site. We don't really need the "Support" page that was added during the install process so log into FarCry, click the site tab and delete the navigation node called "Support".

Solar Safaris	For an adventure out of this world		FarCry Colleged in michae
Overview Site	Content Admin Repo	nting Security Safaris	
Hone     Image: Constraint of the second of th	APPROVED	Itp://solar.localhost         Image: Are you sure you with to delete this content item(s)?         Image: Circle         Image: Circle         90000         Unlocked         90000         farcey         approved         B733P1E8-D565-5B5E-E2A981054AB4PE7A	Edit this content lien Send this content Rem back to drint Delete Send to trash Create Navigation Create Navigation Create Navigation Create HTML Page Create Flash Create Statu Create Statu Create Statu Create Statu Create Statu Create Statu Create Statu

Before we go and add some new navigation items we should create a new dmHTML webskin for a standard page. For the moment just copy the existing displayPageHome.cfm into a new file called displayPageStandard.cfm (we'll modify this webskin later). The only line you need to change is

<!--- @@displayname: Home Page --->

Change it to:

<!--- @@displayname: Standard Page --->

Save the file.

To build a navigation structure we could add a Navigation node to the tree under the home node and then add a HTML page beneath that and repeat the process for all our new pages, or we could use the "Quick Site Builder" utility under the admin node. This is an excellent tool when in development as it quickly builds a site structure without having to do each page manually. To create our navigation structure perform the following tasks:

- Login to FarCry
- Click the "Content" tab
- Choose "Content Utilities" from the sub-section drop down menu

- Click the "Quick Site Builder" link in the left hand menu
- Ensure "Home" is selected in the "Create Structure Within" drop down
- Set status to "Draft" so that we can edit the objects later and add some content
- Check the "Create dmHTML items..." checkbox
- Select "Standard Page" as our "Display Method"
- Uncheck the "Create nav aliases" checkbox (we don't need them at this stage)
- Enter the following structure into the textarea:

bout Us	
estinations	
afaris	
ews	
ontact Us	

• Click the "Build Site Structure" button.

This will automatically create the 5 navigation nodes and associated html pages for you. Clicking on the "Site" tab should show you the updated hierarchy:



Solar Safaris For an adventure out of this world

Repeat the process now for the "Footer" navigation node (under the "Utility" node). Add the following structure using the quick site builder:

Disclaimer			
Contact Us			
Site Map			



Refreshing the home page should now look like:

## 5.2.7 Footer Navigation

Go ahead and create a footer menu using the same principles as the primary navigation. The objectid you want to pass in is #application.navid.footer#. The result should look like:



Try clicking on the navigation links. You will notice that clicking on the footer navigation items won't show the footer navigation. This is because we added it to the displayPageHome webskin and didn't update the displayPageStandard webskin. We could copy the code between the two webskins but the better way is to include the header and footer code as custom tags so you only have to update the code once.

# 5.3 HEADER

Create a new file called "header.cfm" and save it in C:\farcry\projects\solar\_safaris\webskin\includes.

We can copy from the displayPageHome.cfm file everything above the #stObj.body# tag except for the @@displayname line, and paste it into our header file. The only changes we then need to make is change the <h1>#stObj.label#</h1> to <h1>#request.stObj.label#</h1> and add a closing </cfoutput> tag to the bottom of your file. Your header.cfm file should look like this:

html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"<br "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> <html xml:lang="en" xmlns="http://www.w3.org/1999/xhtml"> <head></head></html>
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> <html xml:lang="en" xmlns="http://www.w3.org/1999/xhtml"> <head></head></html>
<html xml:lang="en" xmlns="http://www.w3.org/1999/xhtml"> <head></head></html>
<head></head>
<meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
import style sheet
<link <="" href="#application.url.webroot#/css/styles.css" rel="stylesheet" td="" type="text/css"/>
media="screen" />
page title taken from current object's label
<title>#application.config.general.sitetitle#: #request.stObj.label#</title>
<body></body>
set up main layout table
header image linked back to homepage
<a href="#application.url.webroot#"><img< td=""></img<></a>
src="#application.url.webroot#/wsimages/header.git" border="0"
alt="#application.config.general.siteTitle#">
<(1)>
<()>
<pre><!-- Set up where clause littler--> </pre>
<creat ""\#\"="" novfilter[1]="retatue IN&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;/#listQualify/request mode lvalidetatus "></creat>
(#iistQuaiiiy(request.mode.ivaliustatus, j#) >
get navigation nodes
<cfset gnav="&lt;/td"></cfset>
request.factory.oTree.getDescendants(objectid=application.navid.home.depth=1.
afilter=navFilter,blncludeSelf=1)>
loop over navigation and display each node
<cfloop guery="gNav"></cfloop>
<a< td=""></a<>
href="#application.url.conjurer#?objectid=#qNav.objectid#">#qNav.objectname#
check if object is not the last returned object
<cfif neg="" qnav.currentrow="" qnav.recordcount=""><cfoutput>  </cfoutput></cfif>
search form



We now need to include this header file in our displayPageHome webskin. Add the following line:

<!--- include header code --->
<cfmodule template="/farcry/projects/#application.applicationname#/webskin/includes/header.cfm">

And add an opening <cfoutput> tag beneath it. Your displayPageHome file should now look like:

```
<!--- @@displayname: Home Page --->
<!--- include header code --->
<cfmodule template="/farcry/projects/#application.applicationname#/webskin/includes/header.cfm">
<cfoutput>
       <!--- display body content --->
       #stObj.body#
       <!--- set up where clause filter --->
               <cfset navFilter=arrayNew(1)>
               <cfset navfilter[1]="status IN (#listQualify(request.mode.lvalidstatus, """)#)">
               <!--- get navigation nodes --->
               <cfset gFooter =
request.factory.oTree.getDescendants(objectid=application.navid.footer, depth=1, afilter=navFilter)>
                --- loop over navigation and display each node --->
               <cfloop query="qFooter">
                        <a
href="#application.url.conjurer#?objectid=#qFooter.objectid#">#qFooter.objectiname#</a> |
               </cfloop>
               <!--- copyright info --->
               © #year(now())# Solar Safaris
       </body>
</html>
</cfoutput>
```

## 5.4 FOOTER

Following the same principles as the header include file, go ahead and create a "footer" include file. You will need to perform the following tasks:

- Create a new file called footer.cfm in the C:\farcry\projects\solar\_safaris\webskin\includes folder
- Cut the footer content out of the displayPageHome webskin and paste it into the footer file
- Add an include call in the displayPageHome webskin to include the new footer code

Refreshing the homepage should again show no changes.

## 5.5 STANDARD PAGE

Update the displayPageStandard webskin to make use of the new header and footer files.

Now we'll add a breadcrumb to the top of our displayPageStandard webskin to show us where we are in the site hierarchy. The breadcrumb code is very simple as FarCry provides us with a breadcrumb generator. First we need to include the webskin custom tag library. Add the following code to the top of your displayPageStandard webskin:

```
<!--- import webskin tag library --->
<cfimport taglib="/farcry/core/tags/webskin" prefix="skin">
```

To display the breadcrumb we need to call the custom tag. Place the following code just below the header include code:

```
<!--- breadcrumb trail --->
<div id="breadcrumb">
<skin:breadcrumb separator=" / " here="#stobj.Title#">
</div>
```

Save the file and refresh the "About Us" page. You'll see the breadcrumb trail below the heading with a link back to the home page.



The full webskin code for displayPageStandard should now look something like:

Ok we now have most of the functionality we require for the time being so let's go back and add some content to the "Welcome", "About Us" and "Disclaimer" pages. If you are not feeling very creative you can copy the text from the training CD in the "content" folder. Also modify the displayPageHome webskin so that it displays two columns, the left column with the welcome message the right column blank for the time being.

The home page should now look something like:





The About Us page should look something like:

To add an image to an HTML page you'll need to first add it to an image library. There are a couple of ways to achieve this task; we'll look at adding the image to the global Media Library first.

In the webtop make your way to "Media Library Image" within the Media Library section of the Content tab.

Solar Safaris	For an adventure out of this world
Overview Site	Content Admin Reporting Security Safaris
Media Library 💌	Image Administration
MEDIA LIBRARY CONTENT Media Library Image Media Library File Media Library Flash	Filter by: <u>Category Keyword Date range</u> Displaying 1-0 of 0 results. No Filters Applied
MEDIA LIBRARY MAINTENANCE Mage Resize	Aud         Delete         Request Approval         Approval         Select         Edit         View         Label         Last Updated
<ul> <li>Files Without Content</li> <li>Content Without Files</li> <li>Content Without Relationship</li> </ul>	Add         Delete         Request Approval         Approve         Send to Draft         Properties         Unlock

Solar Safaris For	an adventure out of this world
Overview Site (	Content Admin Reporting Security Safaris
Media Library MEDIA LIBRARY CONTENT * Media Library Image * Media Library File * Media Library Filesh MEDIA LIBRARY MAINTENANCE * Image Resize * Files Without Content	General Details     Image Title :     Alternative Text :     Add to Library : ☑
Content Without Files     Content Without Relationship	- Image Files
	Source Image : Browse FitInside
	Mid Size Image : width:400 height:1000 FitInside
	Thumbnail Image : width:80 height:80 Padding image with #000000
	- Categorisation
	Category :
	Save Cancel

Click the "Add" button to be presented with a form to upload an image from your computer.

Enter "Solar Safaris Crew" as the Image Title and Alternate Text

Make sure the "Add to Library" checkbox is checked

Select the "crew.jpg" image to upload which can be found in the "content" folder of the CD.

We won't be generating a mid-size or thumbnail image so make sure those checkboxes are un-ticked.

Click "save" to upload the image.

Solar Safaris For	an adventure out of this world
Overview Site 0	Content Admin Reporting Security Safaris
Media Library MEDIA LIBRARY CONTENT * Media Library Image * Media Library File * Media Library Flash MEDIA LIBRARY MAINTENANCE * Image Resize	General Details     Image Title : Solar Safaris Crew     Alternative Text : Solar Safaris Crew     Add to Library :
<ul> <li>Files without Content</li> <li>Content Without Files</li> <li>Content Without Relationship</li> </ul>	<ul> <li>Image Files</li> </ul>
	Source Image : C:\Documents and Settin Browse FitInside
	Mid Size Image : generate based on "Source Image" Browse width:400 height:1000 FitInside
	Thumbnail Image : generate based on "Source Image" Browse width:80 height:80 Padding image with #000000
	- Categorisation
	Category : 🛛 root
	Save

The image will have been uploaded and now exists in your projects Media Library from where you can attach it to almost any content type.

Navigate to the "About Us" page in the Site tab and create an editable draft object. On the "body" step you will see a "Relationships" area under the body section.

The Relationships area allows a user to attach Associated Media or Associated Content to their HTML page. In the Associated Media library you can attach an image, file and flash movie. Click the "Open Library" button which will launch a new "Library" window.

In this window you will see your "Solar Safaris Crew" image in the right hand pane. Drag it across to the left hand pane with your mouse to attach it to your HTML page, and then click close.



#### Image Library...

Change To: Image  ATTACH ADD NEW	
Selected	Drag To Select
	Displaying 1-1 of 1 results
Solar Safaris	s Crew

You can now see that the "Solar Safaris Crew" image is sitting within the Associated Media library on your HTML page.

To place this image in the body content of your "body" content you can use the toolbar of the richtext editor. On the 2nd bar of the richtext editor there is button which is modelled after the farcry logo.

Clicking this button will launch a popup providing a tab for all the content types you can associate with your dmHTML object.

Related Content	
Image File Flash Navigation HTML Page Generic Templates	
Item Solar Safaris Crew 💟	
Template Source Image 💌	
preview	
Preview	
THE REAL PROPERTY AND A DESCRIPTION OF THE PARTY OF THE P	-
TO THE TOTAL COUNTY AND	
But the training manual of the	
	~
Insert	
Cancel	_

- 1. Select the "Item" you want
- 2. Select "Source Image" from the list of relevant "Templates"
- 3. Press [preview]
- 4. Press [insert]

The image will now be sitting within your body content where you can move it with your mouse to wherever you want it.



Solar Safaris is an organisation that wants you to experience places not previously possible to experience. They want to take you out of this world. They want to send you deep into the Solar System to experience life in out space. Blast off on the worlds most powerful rocket and take a heading to any of our local planets. Orbit the planets for some of the most stunning views in the Solar System or take an "Up Close and Personal" tour of Mars.

All safaris are designed with you the traveller in mind and we want to blow your mind!

Travel will never be the same again...



A downside to this approach is that you need to upload your image to the Media Library *before* creating/editing your HTML page. This can be troublesome as often a user may be editing their HTML page and then need to navigate to the Media Library before coming back to attach the image.

Luckily there is a shortcut you can take advantage of which is simply allows the user to upload the image(s) while editing an HTML page.

To do this open the Associate Media library (Open Library), then click the "Add New" tab.

ATTACH	ADD NEW	
– General	Details	
General	Details	
	Image Title :	
Alte	ernative Text :	
Ad	dd to Library :	
- Image Fi	iles	
2	Source Image :	Browse
		FitInside
Mi	d Size Image :	
	d Size image .	generate based on "Source Image" width:400
		height:1000
		FitInside
Thur	mbnail Image :	
	nonan inage i	vidth:80
		height80
		Padding image with #000000
- Categori	sation	
outegon	Sation	
	Category :	
Attach Close		

From this screen you can upload an image just as you would do in the Media Library. When you are finished click "Attach" which will upload your image and attach it to your Associated Media library.

## 5.6 NEWS PAGE

#### 5.6.1 Standard Webskin

Before you can view news items on our site we need to create a webskin for the News type. Create a displayPageStandard webskin and save it under C:\farcry\projects\solar\_safaris\webskin\dmNews. This webskin will be almost identical to the dmHTML version except that you may wish to display date and time information for the News version (you may want to dump stObj structure again to see available fields).

Add at least 2 news items to the system via the "Content" tab in FarCry. There are some more example news items on your training CD under the "content" directory.

If you were to preview a news item from within FarCry is should display something like:



#### 5.6.2 Container

We now need to get these news stories listed on the "News" page. To do this we are going to add a container to the dmHTML displayPageStandard webskin so that we can publish the news rule on the page. You can think of containers as a placeholder for content and rules as the content providers for the containers.

To add a container to the dmHTML displayPageStandard webskin you first need to import the container custom tag library and then need to add the actual container following code below the body output so that we can display standard page content first if we need to.



<!--- page based container --->
<con:container label="#stobj.objectID# bottom">

#### 5.6.3 Teaser webskin

Rules display information via teaser webskins. These are a short version of a webskin don't usually contain header or footer code as more than one object is likely to be displayed by each rule. Before we can publish our news items we need to create a teaser webskin for dmNews. Create a new file called displayTeaserStandard.cfm and save it under C:\farcry\projects\solar\_safaris\webskin\dmNews. All we really want to display is the title, date published and teaser text so add the following code to the webskin:

```
<!--- @@displayname: Standard Teaser --->
<cfoutput>
<!--- Show heading and link it to full news item --->
<h3><a href="#application.url.conjurer#?objectid=#stObj.objectid#">#stObj.label#</a></h3>
<!--- show date last updated --->
[#dateFormat(stObj.dateTimeLastUpdated, "dd-mmm-yyyy")#]
<!--- show teaser content --->
#stObj.teaser#
</cfoutput>
```

#### 5.6.4 News Rule

To add the rule to the "News" page browse to the page and then choose "Show Design" from the widget menu. Design mode displays all containers on the page and allows you to edit each container's contents.



Solar Safaris: News - Microsoft Internet Explorer			
File Edit View Favorites Tools Help			
🔇 Back 🔹 🕥 🖌 📓 🐔 🔎 Search 🤺 Favorites 🥝	🛛 😂 🗟 • 🥃 🕯	🗶 🥸 👘	
Address Addres	P9A0F6A8129418designmode=1		💌 🔁 Go
SOLAR SAFARIS		27	×
Home   About Us   Destinations   Safaris   News   Conta	ct Us	Searc	h form to go here
News			
Home / News			
☉ Container Label : bottom			
Edit Container Content D	isclaimer   Contact Us	Site Map   ©	2005 Solar Safaris
javascript:void(0)			Sucal intranet

The page should then refresh and the container displayed:

Click the Edit Widget on the left hand side of the container to bring up Container Management screen.

oottom 🎸			
CONFIGURE RULES	CONTAINER CONTENT	REFLECTIONS	
Available Rule Types:	Child Links Rule Event Calendar Rule Event Rule External Links Rule Handpicked Rule News Rule		Move to Active Rules
Active Rules			<u>Move up</u> Move down Delete Rule
	Commit Changes		

As there are no existing rules set for this container we need to select a rule from the "Available Rule Types" and add it to the "Active Rules" list and "Commit Changes". Select the "News Rule" from the list and click the "Commit Changes" button to add it to the active rules list.

oottom 🍋	
CONFIGURE RULES	CONTAINER CONTENT REFLECTIONS
Available Rule Types:	Child Links Rule Event Rule Event Calendar Rule Handpicked Rule External Links Rule News Rule
Active Rules	News Rule       Move up         Image: State of the stat

You should now be presented with the News Rule edit handler where you can select display method, restrict by categories and other filters. Ensure that our new teaser display webskin for dmNews is listed in the "Display Method" drop down.

oottom 💊			
CONFIGURE RULES	CONTAINER CONTENT	REFLECTIONS	
Active Rules For This Container:	[1] News Rule 💌		
Display Method:	Teaser Standard 💌		
Intro Text			
Suffix			
# Items per page	5.0		

The default values are ok for the moment so click "go" to save the settings. If you now refresh your "News" page you will see your recently add news items listed using the new teaser webskin method. You can now hide the container by choosing "Hide Design from the widget menu.



Try clicking on the news heading to ensure they display the full news item.

#### 5.6.5 Additional Teaser webskins

Try adding another teaser display method for dmNews and modify the News Rule to use the new display method. This time you may want to show the author of the news item and the date and time it was last updated.

#### 5.6.6 Breadcrumbs and Dynamic Pages

The last thing required for the news webskins is to fix the breadcrumb when viewing a news item. Eg when viewing the "New Website Live" news item the breadcrumb is shown as "Home / New Website Live" where it should be shown as "Home / News / New Website Live". This is because the news item doesn't fit into the navigation hierarchy like a normal page, it's known as dynamic content. The same news item could appear in many places throughout the site. To fix this we need to add some code that tells the breadcrumb where in the site want to show the news item as being. Change the existing breadcrumb code to the following in the displayPageStandard webskin for dmNews :

<pre><!-- set navid to news nav node--> <cfparam default="#application.navid.home#" name="application.navid.news"></cfparam></pre>	
<cfoutput> <!-- breadcrumb trail--> <div id="breadcrumb"></div></cfoutput>	

The above code sets the current navid manually to the value of the news nav alias. We need to set this by editing the "News" node in FarCry and adding an alias of "news".

Solar Safaris	For an adventur	e out of this	world			
Overview Site	Content	Admin	Reporting	Security	Safaris	
Home 💟 🔅 Quick Zoom:	Na	vigatio	n Node de	etails: Ne	ws	
Home	Title	*	1	lews		
■-= Home		Advanced	Options			
Vvelcome	Sym	bolic Link:	[	None 🛛 🗸	Ī	
<ul> <li>Destination</li> <li>Safaris</li> <li>Nervision</li> </ul>	Nav	Aliases:	ľ	news		
🖻 - 🚰 Contact Us	E O Frier	ndly URL:	[			
			[	ОКС	ancel	

Now when you view a news item the breadcrumb should display correctly.



# 5.7 UNIT REVIEW

In this unit you have learnt:

- The concept of webskins as display templates
- How to generate multiple webskins for each content type
- How to call tree methods via the request factory scope
- How to produce dynamic navigation menus
- How to add breadcrumbs to a webskin
- How to add containers to a webskin
- How to set and reference Navigation Aliases
- How to add a rule to a container
- How to use teaser webskin methods

# 6 Include Objects

Include Objects are a way to insert custom code within FarCry. It can be as simple as a self-posting form or a complex page referencing external systems. They fit right into the navigation hierarchy tree and can quickly add new functionality to your site.

# 6.1 SITEMAP

The first Include Object we will add to the Solar Safaris website is a site map that will show the site hierarchy on a single page.

## 6.1.1 Inserting Include Object

Include Objects are stored as .cfm files under the <appName>/includedObj directory. If you look at the C:\farcry\projects\solar\_safaris\includedObj directory you will see that FarCry comes pre-installed with two include objects, a site map and search (ignore the \_doNotDelete file, this is a blank file for cvs).

🚞 C:\farcry\solar_sa	afaris\includedOb	j		_ 🗆 🗵
File Edit View F	avorites Tools I	Help		<b></b>
😋 Back 🝷 💮 🝷	🏂 🔎 Search	n 🕟 Folders 🛛 🕼	۵ 🗙 😒	•
Address 🛅 C:\farcry\;	solar_safaris\include	dObj		💌 🔁 Go
Name 🔺	Size	Туре	Date Modified	
adonotdelete.cfm	1 KB	Macromedia HomeSi	17/09/2003 5:53 PM	
👩 _search.cfm	6 KB	Macromedia HomeSi	3/03/2004 2:33 PM	
👩 _sitemap.cfm	1 KB	Macromedia HomeSi	6/01/2004 3:36 PM	
<u> </u>				
3 objects		5	.42 KB 🛛 😔 My Co	omputer //

To make use of these Include files we need to insert them into our site navigation hierarchy. Perform the following tasks to insert the Site Map include object into the Solar Safaris website:

- Login to FarCry
- Click the "Site" tab
- Click the "Utility" tab at the top of the left hand pane
- Expand the "Footer" navigation node
- Expand the "Site Map" navigation node
- Delete the "Site Map" html object
- Click the "Create Include" link

Solar Safaris For at
Overview Site Co
Utility 💽 🛇
Quick Zoom: Utility 🔽 💿
Utility Utility Footer Disclaimer Contact Us Site Map Edit
Сору
Cut
Move Create Status Propertie Delete Send to Zoom In. Mavigation HTML Page HTML Page Include CSS Link Image File
Destination

- Enter a title of "Site Map"
- Select "\_siteMap.cfm" from the "Include" drop down
- You will notice that there are no display methods available. We need to create one before we can continue.

title :		
teaser :		
include :	_sitemap 💙	
displayMethod :	No Content Templates Available	
catinclude :	🖃 root	

## 6.1.2 dmInclude webskin

The webskin for dmInclude is very simple. We just need to include our standard header and footer and another include call for the actual Included Object code. Create a new file called displayPageStandard.cfm and save it in the C:\farcry\projects\solar\_safaris\webskin\dmInclude directory. Then add the following code to the page:

<cfsetting enablecfoutputonly="true"></cfsetting> import tag libraries <cfimport prefix="skin" taglib="/farcry/core/tags/webskin"></cfimport>
<cfmodule <br="" template="/farcry/projects/#application.applicationname#/webskin/includes/dmHeader.cfm">pageTitle="#stObj.title#" /&gt;</cfmodule>
use skin:include tag to include the file <skin:include template="#stObj.include#"></skin:include>
<cfmodule template="/farcry/projects/#application.applicationname#/webskin/includes/dmFooter.cfm"></cfmodule>
<cfsetting enablecfoutputonly="false"></cfsetting>

Save the file and refresh the edit page in FarCry. You should now see our new display method listed. Go ahead and complete the edit form and click the "OK" button.

title :		]
teaser :		]
include :	_sitemap 💌	_
displayMethod :	Standard Page	*
catInclude :	🗆 root	

You can now either preview the page from FarCry or browse the Solar Safaris website to the "Site Map" page to see our Include Object. It should list all the pages under the homepage and create a link for each.



#### 6.1.3 Sitemap custom tag

Open the C:\farcry\projects\solar\_safaris\includedObj\\_siteMap.cfm file to look at how the Site Map is generated.

You will see there are only a few lines of code in the file, an import for the webskin tag library, a caching tag (more about this later) and a call to the sitemap custom tag. Take a look at this custom tag by opening the file C:\farcry\core\tags\webskin\siteMap.cfm. You will see that it uses very similar code to our navigation menus, primarily the getDescendants tree method. There are few attributes you can pass in to customise the results but the basic code is very simple. If this code doesn't produce a sitemap that meets your requirements you can always create your own custom sitemap in your include file, but you will find this custom tag method will meet your requirements for the vast majority of the time.

# 6.2 SEARCH

The other pre-installed Include Object is a search form. This is quite a powerful search utility that searches through the sites verity collections, collates the results and displays them with a ranking, linked title and view of the contents that matched the search criteria. There are a few things we need to configure before we can use the search Include Object.

# 6.2.1 Verity Config

•

FarCry has a configuration file for verity that lists all object types in the system with the option to select which properties of each type should be indexed and available to the verity search interface. You can also index an external directory if required.

Login to FarCry, click the "Admin" tab, select the "Verity Management" option from the sub-section dropdown menu and then click "Verity Config". If you click the "dmHTML" link you will see all the properties for the dmHTML type appear and see that "Body", "Title" and "Teaser" are already selected by default. The same fields are also selected for the dmNews type.

TARCRY tell it to someone who cares				
Overview Site	Content Admin Reporting Security			
Verity Management  FREE TEXT SEARCH Manage Collections Duild/Update All Collections Verity Config	Collection Settings for dmHTML Properties to Index hetaKeywords (HTML head section metakeywords.) habel (Object label or title.) datetimecreated (Timestamp for record creation.) extendedmetadata (HTML head section for extended keywords.) Title (Title of object. *perhaps this should be deprecated for object label*) Teaser (Teaser text.) reviewDate (The date for which the object will be reviewed) datetimelastupdated (Timestamp for record last modified.) Body (Main body of content.) catHTML (Topic.) Custom Fields CustoM3: - not used · · · Associated File Library FileCollectionProperty: - not used · · · Last Built To Date Commit Collection Settings			

As we only have news and html content at the moment these settings will do for now. As we didn't make any changes we don't need to update the config file. We will come back later and add some more selections to expand our search capabilities.

## 6.2.2 Build Collections

FarCry will only let verity index approved content items so before we can build the search collections we need to approve some content. Go ahead and approve the two news items, the "About Us", "Welcome" and "Disclaimer" pages.

## **News Administration**

#### Listing Filter: set

Add Delete Request Approval Approve Send to Draft Properties Unlock

Displaying 1-2 of 2 results						
Select	Status	Action	Title	News Category	Publish Date	Last Updated
			×		desc 💌	<b>~</b>
1	draft	action 🛛 👻	Mars Landing Safari coming soon!		18-Feb-07	19-Feb-07
2	draft	– action – 🛛 👻	Solar Safaris, the new way to travel		18-Feb-07	19-Feb-07

Displaying 1-2 of 2 results

Quick Zo	bom: Home	
- <b>-</b> -		
=	Welcome	
<b>.</b>	About Us	
About U	s	
Edit		
Сору		
Cut		
Preview		
Move	+	
Create		
Status	Approve	
Permiss	Approve Branc	:h
Propertie	Request	
Delete	Request Appro	val for
Send to	Branch	
Zoom In	Zoom Out	

Now that we have some approved content we can go ahead and build our collections. Click the "Admin" tab then the "Verity Management" option from the sub-section drop-down and then the "Build/Update All Collections" link. This will check if we already have a collection for both the dmHTML and dmNews types, if not it creates one for us. It then goes through all approved objects and indexes them in the appropriate collection. It also removes deleted pages from the collection.



## 6.2.3 Search Form

farenv

tall it to compoor

We now need to build a search interface so that users can search by keyword in the Solar Safaris website. We'll add a simple form to the header file that has a text input box and a submit button. Add the following code to the C:\farcry\projects\solar\_safaris\webskin\includes\header.cfm file replacing the "search form to go here" text:

<form action="#application.url.conjurer#?objectid=#application.navid.search#" method="post" style="display:inline;"> <input type="text" name="criteria"> <input type="submit" value="Search"> </form>

You will see that the action value for the form uses a "search" nav alias. During the install process a "Search" navigation node is created for us under the "Utility" node and a nav alias of "search" is set. An include object is also created for us using the "\_search" include code. The only problem we have is that this include object was set to use a display method called "displayDefault" and we called ours "displayPageStandard" so we need to log into FarCry and change this value now.

Tarciy ten to someone wi	10 cares	
Overview Site Conte	ent Admin Reporting Se	ecurity
Utility V Quick Zoom: Utility V C Utility Search C C Search C Search C Search C Search	title : teaser :	Search
	include :	_search 💙
	displayMethod :	Standard Page
	catInclude :	🗆 root
	Save cancel	
If you now refresh your homepage you will see the search form displayed in the header section of the page. Try searching for "Mars" and see if you get any results.



You will see the results are returned with a ranking, the title of the object shown and linked through to the full page and the keyword highlighted in context so you can get a feel for what the page is about. There is also the option to search by:

- Any of these words (an OR search)
- All of these words (an AND search)
- These words as a phrase (a "phrase" search)

#### 6.2.4 Include Object

Again if this doesn't quite meet your needs you can customise the include object but it is a very powerful search interface without having to write a line of code! If you open the C:\farcry\projects\solar\_safaris\includedObj\\_search.cfm file you will see the following logic:

- Get the active verity collections for the site (as specified in the verity configuration file)
- Prepares the search criteria keywords adding verity operators were appropriate

- Performs the search using the FarCry search component
- Displays a keyword search form
- Displays the search results
- Displays pagination navigation

## 6.3 CONTACT US

Let's add one more Include Object to the Solar Safaris site for now, an online contact form where users can enter their details and a short message and send it to the Solar Safaris contact team. To do this we will perform the following:

- Create a self posting form capturing name, email and message
- Add form validation
- Create a custom configuration file to store the address the message should be sent to
- Send the message using cfmail
- · Link to the one form from two locations within the site using Symbolic Links

### 6.3.1 Contact Form

Create a new file called \_contact.cfm and save it in the C:\farcry\projects\solar\_safaris\includedObj directory. We need to create a form with three fields:

- Name (the user's name)
- Email (the user's email address so they can be contacted in the future)
- Message (the message the user wants to send to Solar Safaris)

Add the following code to you \_contact file to produce the form:

<cfoutput> <!-- default values--> <cfset showform="1"> <cfparam form.er<="" name="form.us&lt;br&gt;&lt;cfparam name=" th=""><th>serName" default=""&gt; mailAddress" default=""&gt;</th></cfparam></cfset></cfoutput>	serName" default=""> mailAddress" default="">
<cfparam "="" name="form.m&lt;br&gt;&lt;cfset error="></cfparam>	essage" default="">
check if form has be</td <td>een submitted&gt;</td>	een submitted>
<ctif isdefined("form.subr<="" td=""><td>nit")&gt;</td></ctif>	nit")>
<cfset showform<="" td=""><td></td></cfset>	
add form ac</td <td>tion code&gt;</td>	tion code>
<cfif showform=""></cfif>	
display the f</td <td>orm&gt;</td>	orm>
<form <="" action="#a&lt;/td&gt;&lt;td&gt;pplication.url.conjurer#?objectid=#application.navid.contact#" method="post" td=""></form>	
name="contact">	
<table border="0&lt;/td&gt;&lt;td&gt;" celipadding="3" celispacing="3"></table>	
~u~	Please enter your name, email & message
below.	
	Name
	<input <="" name="userName" td="" type="text"/>
value= #form.userName	# />
	Email

	<input <="" name="emailAddress" td="" type="text"/>
value="#form.emailAddre	ess#"/>
	Message
	<textarea <="" name="message" rows="10" td=""></textarea>
cols="35">#form.messag	ge#
	<input <="" name="submit" td="" type="submit" value="Send Message"/>
class="submit"/>	

The above code is a simple self posting form that checks if the form has been submitted, if it has it sets a flag to not show the form and process any form actions, otherwise it displays a form with name, email and message fields. You will notice the form action uses a navigation alias called "contact". This doesn't exist yet so we need to set that now.

- Log into FarCry
- Click the "Site" tab
- Edit the "Contact Us" navigation node
- Click the "Advanced Options" button
- Add a "Nav Alias" of "contact"
- Click the "OK" button to save the details.

Utility 💟 🔅	Navigation Node	details: Contact Us
Quick Zoom: Utility	Title: <sup>*</sup> Advanced Options Symbolic Link: Nav Aliases:	Contact Us None  contact
	Friendly URL:	
		OK Cancel

While we are there we should delete the "Contact Us" html page and add an Include Object instead.

- Expand the "Contact Us" navigation node in the tree
- Select the "Contact Us" html page and select "Delete"
- Select the "Create Include" link
- Enter a title of "Contact Us"
- Set the "Include" drop down to "\_contact.cfm"
- Make sure the "Display Method" drop down is set to "displayPageStandard"
- Click "OK" to save

Utility V Quick Zoom: Utility V =	title teaser	Contact Us	
E Search E Footer Contact Us (incomplete) Site Map	include displayMethod	ContactUs V Standard Page	
	Save cancel	: 🖸 root	

You should now be able to browse to the "Contact Us" page in the primary navigation bar and see the following:



#### 6.3.2 qForms validation

FarCry comes with support for the qForms JavaScript validation library. Using qForms requires very little code and provides a consistent layer of validation across forms in your site. We are going to add validation to our contact form by ensuring the following:

- A name is given
- An email address is given and it's a valid format
- A message is given

First we need to load the qForms libraries. The best place to do this is in the header so we will an attribute to the header file that is a flag for whether or not qForms needs to be loaded. It won't be loaded by default.

Add the following code to the top of C:\farcry\projects\solar\_safaris\webskin\includes\header.cfm

<!--- set default values ---> <cfparam name="attributes.bHasqForms" default="0">

Then add the following in the head section:

qForms
<cfif attributes.bhasqforms=""></cfif>
// load the qForm JavaScript API //
<script src="js/lib/qforms.js" type="text/javascript"></script>
<script language="JavaScript" type="text/javascript"></td></tr><tr><td><!//</td></tr><tr><td>// specify the path where the "/qForms/" subfolder is located</td></tr><tr><td>qFormAPI.setLibraryPath("js/lib/");</td></tr><tr><td>// loads all default libraries</td></tr><tr><td>qFormAPI.include("*");</td></tr><tr><td>//></td></tr><tr><td></script>

The above code does a check if the bHasqForms attribute has been passed in and if it has and is set to true it will load the qForms API. So now we need to modify our displayPageStandard dmInclude webskin so that it passes the bHasqForms attribute to the header. This will allow all include objects to use the qForms API. Change the call to the header include to the following:

<!--- include header code ---> <cfmodule template="/farcry/projects/#application.applicationname#/webskin/includes/header.cfm" bHasqForms="1">

Now that qForms is available to us on the page, we just need to add the validation code to our form. The following code should be placed just before the closing </form> tag in your contact include file:

<!--- form validation ---> <SCRIPT LANGUAGE="JavaScript"> <!--// objForm = new qForm("contact"); objForm.userName.validateNotNull("Please enter your name"); objForm.emailAddress.validateNotNull("Please enter an email address"); objForm.message.validateEmail("Please enter a valid email address"); objForm.message.validateNotNull("Please enter a message"); //--> </SCRIPT> The above code first binds the form to qForms (you must name your form when using qForms) and then for each field that we want validated we add a line specifying the name of our form field (remember JavaScript is case sensitive), a validate function and an error message to display.

If you now reload your contact page and try to submit the form without entering any information you should see the following:

Contact Us				
	Mi	icrosoft I	nternet Explorer	×
Please enter your name, email	& message l		The following error(s) occ - Please enter your name - Please enter an email a	:urred: e iddress
Name			- Please enter a message	3
Email			ОК	
Message				
			<b>Y</b>	
Send Message				

### 6.3.3 Custom Config

Now that we have a validated form we will want to send the details passed in an email to one of the staff members at Solar Safaris. To avoid hard coding values into our include object we can create a custom config item to store the email address to use when sending the email. Custom configs are editable just like any config from the FarCry "Admin" tab.

To create a custom config we need to create a new file called contact.cfm (this will be the name of the custom config) and save it in the C:\farcry\projects\solar\_safaris\system\dmConfig directory. Add the following code:

```
<cfset stConfig = structNew()>
<cfset stConfig.contactEmail = "brendan@daemon.com.au">
<cfset stConfig.thankyou = "Thank you for taking the time to contact us. We will get back to you
shortly">
```

Save the file and perform the following tasks:

- Login to FarCry
- Click on the "Admin" tab
- Click the "Custom Config" link

## Custom Config Types

Config	Deployed	Deploy
contact	×	Deploy

- Click the deploy link (this adds the config to the database and application scope)
- Click the "Config Files" link
- You should see "contact" listed. Click "contact"

## FarCry Internal Config Files

contact Config	
CONTACTEMAIL	brendan@daemon.com.au
THANKYOU	Thank you for taking the time to contact us. We will get back to you shortly

Update Config

You can now update the email address used and thank you message at any time simply by logging into FarCry and going to this edit screen – a much better approach than hard coding these values in the include object.

Each custom config can have as many values as you like, just a new structure element for each value. You can have as many custom config files as needed.

## 6.3.4 Form Action

The form action will be to email the message to the configured email address (you may need to configure ColdFusion to use a mail server) and then display a thank you message (also from our custom config).

Add the following code to your contact include object, replacing the <!--- add form action code ---> line.

<!--- send email ---> <cfmail to="#application.config.contact.contactEmail#" from="#form.emailAddress#" subject="Website Contact Form"> A message has been submitted from the website: From: #form.userName# Email: #form.emailAddress# Message: #form.message# </cfmail> <!--- display thank you message ---> #application.config.contact.thankyou#</P> <a href="#application.url.webroot#">Return to home page</a>

Now try submitting your name, email and message to the contact form. You should see your thank you message appear and an email in your inbox.





#### 6.3.5 Symbolic Links

Now that we have our "Contact Us" form functional we need to link the footer "Contact Us" link to the same page. We could create another Include Object in the navigation tree but the best practice is to make the "Contact Us" navigation node in the footer menu a Symbolic Link of the main "Contact Us" navigation node. What this means is that when we click the footer link we are actually viewing the page from the home branch.

To set the Symbolic Link we need to perform the following tasks:

- Login into FarCry
- Click on the "Site" tab •
- Click on the "Utility" sub tab"
- Expand the "Footer" navigation node"
- Click on the "Contact Us" navigation node
- Click the "Edit" link

farcrv

- Click the "Advanced Options" button
- Select "contact" from the "Symbolic Link" drop down list
- Click the "OK" button to save the changes

farcry tell it to someone wi	ho cares	
Overview Site Cont	ent Admin Repo	rting Security
Utility 💽 🔹	Navigation Not	de details: Contact Us
otinty of	Title:*	Contact Us
■ 🚰 Utility □ 🚰 Search	Advanced Option	s
⊨ Footer	Symbolic Link:	None 🔽
Contact Us	Nav Aliases:	None contact
a Sine wah		disclaimer fileroot
	Friendly URL:	footer
		home
		imageroot
		news
		rubhish
		search
		site_map

If you now click on the "Contact Us" link in the footer navigation of the Solar Safaris website you will see the same form that we set for the primary navigation version. FarCry ignores the html page that is still set under the "Contact Us" footer node because we've set the navigation node as a symbolic link. We can now safely delete the "Contact Us" html page from the footer "Contact Us" navigation node.

FarCry Developer Course

## 6.4 UNIT SUMMARY

In this unit you have learnt:

- The concept of Include Objects
- How to set an Include Object in the navigation tree
- How to use the sitemap custom tag
- How to setup the verity configuration file
- How to build/update verity collections
- How to use the default search interface
- How to build a custom include object
- How to setup and use the qForms API
- How to create a custom configuration file
- How to create a symbolic link in the navigation tree

# 7 Custom Content Types

In this unit you will learn how to add new Content Types to FarCry, giving you greater flexibility in the way your content is stored and presented.

FarCry 4.0 has launched a revolutionary web development tool to assist developers in not only creating custom types, but also in creating and maintaining web forms to handle the adding, editing and deleting requirements of that type. We will be looking at this new feature (called "FormTools") when we create our custom types for the Solar Safaris site.

You can create Custom Types as either Tree types or Dynamic types. Tree types allow you to use your new Custom Type in the main navigation tree with all of the default tree behaviours. Dynamic types are used in a similar way to dmNews and dmEvents and require a custom tab for listing and editing. They also require a Publishing Rule to display their contents (we will cover Custom Rules in the next unit).

For the Solar Safaris website we will create two Custom Types:

- Destination
- Safari

We will first work with the Destination type, which will be used in the main navigation tree and capture information on all the Solar Safaris destinations such as title, description, attractions and an image of the destination.

The Safari custom type will later be developed as a Dynamic type which will be used to define the Safaris that Solar Safaris offers, which will be a involve multiple destinations, dates and costs.

## 7.1 COMPONENT DEFINITION

The first step in creating a Content Type is the component definition. Create a new file called destination.cfc and save it in the C:\farcry\projects\solar\_safaris\packages\types folder. Add the following code.

<cfcomponent< th=""></cfcomponent<>
name="Destination"
displayname="Destination"
extends="farcry.core.packages.types.types"
hint="Custom Type for Destination content"
bUseInTree="true"
bObjectBroker="true"
objectBrokerMaxObjects="1000">

The most important line above is the "extends" attribute. By extending the core **types.cfc** means that our Destination custom type will inherit all the default type properties and methods. The default type properties are:

- ObjectId
- Label
- DateTimeCreated
- DateTimeLastUpdated
- CreatedBy
- LastUpdatedBy
- Locked
- LockedBy

Some of the default type methods include:

- createData
- setData
- delete
- getData
- getView

The "bUseInTree" attribute sets this Custom Type to be used in the navigation tree. Once deployed there will be the option to create a "Destination" within the tree.

Another of FarCry 4.0's advanced features is a special server side caching mechanism called ObjectBroker. A well known fact of web development (regardless of language) is that the biggest bottleneck in an application is the database.

ObjectBroker effectively bypasses that bottleneck by storing content objects in server memory for incredibly fast data lookups without the need for a roundtrip call to the database server. This is nothing new as a concept, but it is the way in which the FarCry framework achieves this which is special.

To leverage ObjectBroker in your custom type you simply need to set the bObjectBroker flag to true and specify how many content objects (destinations in this example) you want to store in memory.

## 7.2 PROPERTIES

Destination requires a number of properties. Add the following code inside your <cfcomponent> tags:

property definitions
<cfproperty <="" ftseq="1" td=""></cfproperty>
ftFieldSet="Destination Details"
name="title"
type="string"
required="true"
hint="Title of a Destination"
ftLabel="Title"
ftType="string"
ftValidation="required" />
<cfproperty< td=""></cfproperty<>
ftSeq="2"
ftFieldSet="Destination Details"
name="teaser"
type="longchar"
required="false"
hint="Short description of a Destination"
ftLabel="Teaser"
ftType="longchar" />
<cfproperty< td=""></cfproperty<>
ftSeq="3"
ftFieldSet="Destination Details"
name="description"
type="longchar"
required="false"
hint="Long description"
ftLabel="Description"
ftType="richtext" />
<cfproperty< td=""></cfproperty<>
ftSeq="4"
ft-ieldSet="Attractions"
name="attractions"
type="longchar"
required="faise"
hint="Destination attractions"
ttLabel="Attractions"
ftType="longchar" />

Each <cfproperty> tag represents a field that is to be created within the database table for the Content Type. The "type" attribute determines the field type in the database table. The "required" attribute sets Null and Not Null definitions and if a "default" attribute is defined it sets the default value for the field in the database.

The supported property types are:

- UUID (creates a varchar or equivalent field)
- String (creates a varchar or equivalent field)
- LongChar (creates an nText or equivalent filed)
- Boolean (creates a Boolean or equivalent field)
- Date (creates a date/time field)
- Numeric (creates a numeric field)
- Array (creates a new table with objectid, data and sequence fields used to store relational information)

## 7.3 DEPLOYMENT

To deploy the Content Type into the system perform the following tasks:

- Login to FarCry
- Click the "Admin" tab
- Select "COAPI Management" from the sub-section drop-down
- Click on the "Type Classes" link
- You will see all the core types listed as deployed (they are all deployed as part of the install process) and your "Destination" Custom Type marked as un-deployed and a link to deploy it.

OAPI Management 💌	Custom C	ontent Types					
OAPITOOLS	Integrity	Component	Component		Deploy		Doc
Type Classes     Rule Classes     COAPI Metadata     COAPI Schema  DIAGNOSTICS     Rebuild Tree Metadata     Frk Refbylects Table     Reatlach Orphan Folders     Reatlach Orphan Categories	~	Facts	dmFacts		Scaffold		Doc
	~	Events	dmEvent		Scaffold		Doc
	<b>~</b>	Destination	destinatio	n	Deploy		Doc
	~	Link	dmLink		Scaffold		Doc
	Type Clas	News	dmNews		Scaffold		Doc
	Integrity	Component		Component		Deploy	Doc
	~	Flash		dmFlash		n/a	Doc
	~	Include		dminclude		n/a	Doc
	~	dmArchive handler		dmArchive		n/a	Doc
	✓	XML Export		dmXMLExport		n/a	Doc
		XML Export wizard		dmXMLExport dmWizard		n/a n/a	Doc Doc

 Click the "Deploy" link. This will create a database table called "destination" and fields as defined in with our <cfproperty> tags without requiring to open the database or run any custom SQL scripts

	annarannye	•	17.4
~	destination	~	n/a
1	desBrofile	1	

## 7.4 ADDING NEW PROPERTIES

The COAPI screen all allows for additional properties to be deployed at a later time. If your type requirements change you can simply add a new <cfproperty> tag to the component definition and then deploy the new property via the COAPI screen. Add the following property to your destination.cfc file:

<cfproperty name="newProperty" type="string" required="false" hint="a new property">

Now refresh the "Type Classes" screen under the "COAPI" tab in FarCry. You will see it is reporting a conflict with the "newProperty" field and has it marked as undeployed with the option to "Deploy Property". Go ahead and select "Deploy Property" and click the "Go" button.



The "newProperty" field has now been added to the database with no sql scripts or database manipulation required.

## 7.5 DELETING PROPERTIES

You can also delete properties from a Custom Type through the COAPI screen. Remove the "newProperty" <cfproperty> tag from your destination.cfc, save the file and then refresh the "Type Classes" screen under the "COAPI" tab.

•	anterine		•	17 8
🗙 See Below	destination		<ul> <li>✓</li> </ul>	n/a
The following database discrepencies exist :				
Property	Exists In CFC	xists In CFC Type		
newProperty	×	varchar	Do Nothing	<b>-</b> Go
· · · · ·			Do Nothing	
~	dmProfile		Delete Column Rename Column	n/a

Select "Delete Column" and click the "Go" button. This will remove the field from the database.

## 7.6 EDIT DESTINATIONS

Thanks to the magic of FarCry 4.0 FormTools, creating administration forms to add and edit destinations is as easy as 1-2-3. In fact because you are creating a tree type you don't need to do anything at all! The FarCry framework will automatically handle the building of your form to add and edit a destination.

## 7.7 ADDING CUSTOM TYPE TO THE NAVIGATION TREE

Now that we have the Destinations custom type defined and deployed (and FarCry is automatically creating our edit form) we can now add a Destination to the navigation tree. Perform the following tasks to add a Destination to the navigation tree:

- Login to FarCry
- Click on the "Site" tab
- Click on the "Destinations" navigation node"
- Click the "Create Navigation" link
- Give the new navigation node a title of "Mercury" and click the "OK" button
- You will now see "Destination" is listed under the "Create" menu. Click the "Create Destination" link

Home 💟 🛇	DRAFT		
Quick Zoom: Home Home Home Destinations Destinations Destinations Hercury Contact Us	Content Item Title: Content Item Type: Navigation Alias(es): Created by: Date Created: Locking: Last Updated: Last Updated By: Current Status: ObjectID Stats Auddt Properties	Mercury Navigation farcry 19/02/07 Unlocked 19/02/2007 farcry draft D56B7B8C-E480-BEFE-3CC7C1AC78A36344	Edit this content item Request Approval for this content item yourself Delete Send to trash Create Navigation Create Navigation Create HTML Page Create Include Create Include Create Include Create Destination Create Destination Create Link Preview

You will now see the edit form which FarCry (using FormTools) has created for you. Add some information about Mercury into the form (you can copy some from the "content" directory on the training CD).

The framework handles all the database processing for you, it even provides validation (try submitting the form with no title).

Go ahead and save the Mercury destination and add navigation and destination objects for the following planets:

- Venus
- Mars
- Jupiter
- Saturn
- Neptune
- Uranus

Your navigation tree should now look something like this:

SITE MANAGEMENT



#### 7.8 STATUS

Content Types can have a status setting just as the core types can. By simply adding a property to our custom type called status, FarCry automatically adds status functionality like request approval, approve and send to draft. Add the following <cfproperty> tags to destination.cfc and deploy the new properties via the COAPI screen.

<cfproperty name="status" type="string" required="true" default="draft" hint="status of the object"> <cfproperty name="commentlog" type="longchar" hint="Workflow comment log" required="no">

You will now be able to use the basic FarCry workflow for your destination objects.

DRAFT		
		Edit this content item Request Approval
Content Item Title:	Mercury	for this content item
Content Item Type:	Destination	Approve this item
Created by:	farcry	yourself
Date Created:	19/02/07	Delete
Locking:	Unlocked	Draview
Last Updated:	19/02/2007	FIGVIEW
Last Updated By:	farcry	
Current Status:	draft	
Template:	displayPageStandard	
Teaser		
ObjectID	D56D35FD-9C76-E3CC-2D2C2EFF6A488899	
<ul> <li>Stats</li> <li>Audit</li> <li>Add Comments</li> <li>Properties</li> <li>Show Archive</li> </ul>		

Overview Site Cont	ent Admin Reporting Security
Home 💙 🔇 Quick Zoom: Home 🔽 🐨	Set content item status to approved
Home     Welcome     About Us     Destinations     Destinations     Safaris     Safaris     Contact Us	
	Submit Cancel

## 7.9 VERSIONING

Content types can also make use of FarCry versioning. Versioning allows a draft object to be made of a current live object so that changes can be made without affecting the live site. Ones changes have been finalised the draft version can be sent live and the previous live version is archived.

To add versioning to a content type simply change the extends attribute of the component as follows:

<cfcomponent extends="farcry.core.packages.types.versions" displayname="Destination" ... />

Note you will need to update your application to apply the change.

If you now select an approved Destination object in your tree you will see you have a link called "Create an editable draft version". Click this link, and change the some of the content.

Other versioning options include:

- Restore live object over this draft
- Request approval
- Send object live
- Delete this draft version
- Send live object back to draft (deleting draft version)
- Archive dump
- Archive preview
- Archive rollback

farcry tell it to someone who cares							
Overview Site Conten	t Admin Repor	ting Security					
Home 💌 🕸	Archive						
Quick Zoom: Home 🔽 💿	Date	Label User					
	19 February 2007 02:22	Mercury farcry	Preview	Rollback			
	19 February 2007 02:22	Mercury farcry	Preview	Rollback			
About Us     About Us     Destinations     Mercury     Safaris     News     Contact Us	[Cancel]						

## 7.10 WEBSKIN

Now that we have our Custom Type fully functional from an admin perspective, we now want to be able to view it in our site. We can achieve this by allowing the user to choose a display method while editing a destination, much like you would do in an HTML object. Add the following <cfproperty> tag your destination.cfc and deploy it via the COAPI screen.

<cfproperty< th=""><th></th></cfproperty<>	
ftSeq="5"	
ftFieldSet="Display"	
name="displayMethod"	
type="string"	
required="false"	
hint="Display method template"	
ftLabel="Display Method"	
ftType="webskin"	
ftprefix="displayPage" />	

We are now ready to create our destination webskin.

Create a new file called displayPageStandard.cfm and save it under the C:\farcry\projects\solar\_safaris\webskin\destination directory. Add code to this template that identifies this template as "Standard Display" and includes the header/footer and display all the attributes of our destination object. Save your file and then go back and edit your "Mercury" destination page in FarCry (you may have to send it back to draft or create a draft version first). You should now see the "Display Methods" drop down box listing your new "Standard Display" webskin. Select this and click the "Save" button.

Previewing the "Mercury" destination from FarCry should now look something like:



## 7.11 CHILD LINKS RULE

To list all of our new Destination objects on the "Destinations" page in the Solar Safaris website we will use the ChildLinks rule. This rule looks down the tree from the current page and gets all the child objects (those which are one level below) and displays them using a common teaser display method.

We first need to create this common teaser display method across dmHTML, dmInclude and dmLink (the standard core types that are used with Child Links). Use the below code and save it as displayTeaserStandard.cfm in the dmHTML, dmInclude, dmLink and destination directories.

<cfoutput> <!--- Show heading and link it to full item ---> <h3><a href="#application.url.conjurer#?objectid=#stObj.objectid#">#stObj.label#</a></h3> <!--- show teaser content ---> #stObj.teaser# </cfoutput>

CONFIGURE RULES	CONTAINER CONTENT	REFLECTIONS
Active Rules For This Container:	[1] Child Links Rule 💌	
Select display method for this publishing rule:	displayTeaserStandard.cfm	1 💌
Intro Text		
	go	



## 7.12 DYNAMIC CUSTOM TYPE

The next Custom Type will be a "Safari" type which will store information on the safaris that the Solar Safaris company offers. A safari is a combination of destinations packaged together and will also have departure date, return date, cost, teaser and description. Because the safaris are date based it is regarded as Dynamic Content and does not fit into the navigation hierarchy. Therefore we will create a Custom Tab for this new type and provide a listing/search interface similar to that of News and Events.

We will create a multi-step edit handler or wizard to help with all the properties on the form, and finally we will create some webskin templates for the new type to be displayed in the website itself.

#### Create a new component called safari.cfc and save it under the

C:\farcry\projects\solar\_safaris\packages\types directory. Add the standard component attributes (this one doesn't require the bUseInTree attribute) and then add the following property definitions:

- title
- cost
- displayMethod
- departureDate
- returnDate
- teaser
- description
- aDestinations (type = array. This will be used to store the destination objectids)
- categories

#### <!--- property definitions --->

<cfproperty ftSeq="1" ftWizardStep="Start" ftFieldset="General Details" name="title" type="string" required="true" hint="Title of a Safari" ftLabel="Title" bLabel="true" ftType="string" ftValidation="required" />

<cfproperty ftSeq="2" ftWizardStep="Start" ftFieldset="General Details" name="cost" type="numeric" required="false" hint="Cost of a Safari" ftLabel="Cost" ftType="numeric" />

<cfproperty ftSeq="3" ftWizardStep="Start" ftFieldset="General Details" name="displayMethod" type="string" required="false" hint="The display method to use" ftLabel="Display Method" ftType="webskin" />

<cfproperty ftSeq="5" ftWizardStep="Start" ftFieldset="General Details" name="departureDate" type="date" required="false" hint="The departure date" ftLabel="Departure Date" ftType="datetime" />

<cfproperty ftSeq="6" ftWizardStep="Start" ftFieldset="General Details" name="returnDate" type="date" required="false" hint="The return safari date" ftLabel="Return Date" ftType="datetime" />

<cfproperty ftSeq="7" ftWizardStep="Start" ftFieldset="General Details" name="teaser" type="longchar" required="false" hint="Teaser of the safari" ftLabel="Teaser" ftType="longchar" />

<!--- step 2 --->

<cfproperty ftSeq="20" ftWizardStep="Description" ftFieldset="Description" name="description" type="longchar" required="false" hint="Description of the safari" ftLabel="Description" ftType="richtext" />

<!--- step 3 --->

<cfproperty ftSeq="30" ftWizardStep="Destinations" ftFieldset="Destinations" name="aDestinations" type="array" required="false" hint="Destinations of the safari" ftLabel="Destinations" ftType="array" ftJoin="destination" />

<!--- step 4 --->

<cfproperty ftSeq="40" ftWizardStep="Categories" ftFieldset="Categories" name="SafariCategories" type="string" required="false" hint="Categories of the safari" ftLabel="Categories" ftType="Category" ftAlias="Root" />

Deploy the new type via the COAPI screen. If you click on the "Site" tab and select a navigation node you will notice there is no option to "Create Safari" as we didn't set the type to use the tree. This means we need to administer it from a Custom Tab.

## 7.13 CUSTOM ADMIN

FarCry has the ability to define multiple Custom Tabs in the admin interface. This allows developers to add administration interfaces for custom types and other custom functionality. Each tab can have multiple sub tabs and all can be secured using Policy Group permissions.

#### 7.13.1 XML Schema

The Custom Tab structure is defined in an XML document called customadmin.xml in the <appName>\customadmin directory. This document defines each tab (called a section), sub sections, menus, menu items and permissions.

Use the following code for the C:\farcry\projects\solar\_safaris\customadmin\customadmin.xml file:

xml version="1.0" encoding="utf-8"?
<webtop></webtop>
<section <="" id="solarSafarisSection" mergetype="merge" p="" permission="MainNavContentTab"></section>
label="Solar Safaris" labelType="value">
<subsection <="" id="solarSafarisSubSection" label="Solar Safaris" mergetype="merge" td=""></subsection>
labelType="value">
<menu <="" id="solarSafarisMenu" label="Solar Safaris" mergetype="merge" td=""></menu>
labelType="value">
<menuitem <="" id="solarSafaris" label="Safari" mergetype="merge" td=""/>
link="/admin/customadmin.cfm?module=customLists/safari.cfm" />

The above schema creates one main tab called "Safaris" which is only visible to users with the MainNavAdminTab permission. A sub tab called "Safaris" is also created and uses the same permission structure. Finally the menu itself is created with a title of "Safaris" and a link to a safaris list page (yet to be created). To see this new tab structure log back into FarCry and append updateapp=1 to the url, eg <a href="http://localhost/farcry/?updateapp=1">http://localhost/farcry/?updateapp=1</a>. This will re-initialise the application scope. This is required as the customadmin schema is stored as an application variable.

farcry	tell it to son	neone who cai	res			
Overview	Site	Content	Admin	Reporting	Security	Safaris
Safaris 💌						
SAFARIS + Safaris						

#### 7.13.2 Object Admin

Previous versions of FarCry came with a standard interface for listing/searching dynamic content types called Type Admin. News, Events, Links and Facts made use (and still do) of Type Admin by default, but in FarCry 4.0 there has been an enhancement to Type Admin. The new version is called "Object Admin" which Custom Types can easily leverage to produce similar functionality to Type Admin (but with more power and flexability), which includes:

- Add new item
- Status changes (eg approve, send to draft, request approval)
- Delete
- Edit
- Preview
- Search
- Sorting of results
- Pagination
- Unlock

To include Object Admin functionality for our Safaris custom type we need to create a new file called safari.cfm and save it in the C:\farcry\projects\solar\_safaris\customadmin\customLists directory. Add the following code:

<cfsetting enablecfoutputonly="true"></cfsetting>
<cfimport prefix="admin" taglib="/farcry/core/tags/admin"></cfimport> <cfimport prefix="ft" taglib="/farcry/core/tags/formtools"></cfimport>
set up page header
<admin:header></admin:header>
<ft:objectadmin title="Safaris" typename="safari" ColumnList="label" SortableColumns="label" IFilterFields="label" sqlorderby="datetimelastUpdated desc" /&gt;</ft:objectadmin 
setup footer <admin:footer></admin:footer>

#### <cfsetting enablecfoutputonly="false" />

The above code imports the tag libraries required, the calls the standard admin header code and the calls the objectAdmin custom tag. Passed to this tag are:

- Typename of the Custom Type you want to manage
- · ColumnList which defines which properties are viewable on the Object Admin screen
- SortableColumns allows you to sort (asc or desc) your result set on a particular property
- IFilterFields defines which type properties you can use to search your result set on
- sqlorderby is the default sorting of the result set, displaying the most recently edited object first

If you now click on "Safaris" tab and then the "Safaris" link you will see the Object Admin interface listing your Safari objects. There should be none listed at this time.

farcry tell it to some	eone who cares		
Overview Site	Content Admin Report	ing Security Safaris	
Safaris 💌 SAFARIS	Safaris Listing Filter: set		
<ul> <li>Sataris</li> </ul>	Add Delete Properties	Unlock	
	Select	Action	label
	• 0 records found.		

Thanks to the magic of FormTools, this is all we need to do to get a multi-step edit hander for our Custom Type.

#### 7.13.3 Category Tree

The final step of our Wizard is a categorisation step that allows us to categorise our safari against the site category tree. First we need to add some categories to the site category tree. To add Categories perform the following tasks:

- Login to FarCry
- Click on the "Content" tab
- Select the "Content Utilities" option from the sub-section dropdown
- Click the "Manage Keywords" link
- Right-click on the "Root" category and click "Insert"

Solar Safaris For a	n adventure out of this world		FarCry	Logged in: Michael (Loqout   <u>View</u> )
Overview Site Co	ontent 🗍 Admin 🗍 Reporting 🗍 Security 🗍 Solar Safaris			
Content Utilities	Existing Hierarchies: Show All Hierarchies 💌			
EXPORT	<b>i</b> e root	Category Name:	root	update
<ul> <li>Export</li> </ul>	root			
CATEGORY MANAGEMENT	insert	Sort Children Asc	ending	
<ul> <li>Manage Keywords</li> <li>Manage keywords Flex beta</li> </ul>	Move +			

- Give the category a title of "Safaris"
- Insert another category under the "Safaris" category called "Inner Solar System"

Insert another category under the "Safaris" category called "Outer Solar System"



•

If you have a large category tree to build you can also use the "Quick Category Builder" under the "Admin" tab which functions in a similar way to the "Quick Site Builder" we used earlier.

## 7.14 WEBSKINS

You may have noticed that we have no Display Methods listed on the start step of our Wizard as we don't have any webskins created for the safari type.

Solar Safaris	For an adventure out of th	is world			F
Overview Site	Content Admin	Reporting	Security	Solar Safaris	
Solar Safaris 💉 SOLAR SAFARIS + Safari	(HTML) St	tart			
	- Genera	al Details —			
	©	<b>Title</b> Cost Display Method Departure Date Return Date Teaser	<ul> <li>The BIG :</li> <li>1,000.00</li> <li>No Conte</li> <li>23 Feb 2</li> <li>28 Feb 2</li> <li>This super have to b</li> </ul>	safari! nt Templates Available 007 11:34 AM 007 11:34 AM er dooper huge safari wil ve seen to be believed!	I be taking in sites which

Go ahead and create one now called "displayPageStandard.cfm" and save it in C:\farcry\projects\solar\_safaris\webskin\safaris. Make sure you updateApp unless you are in developer mode.

One thing that is different from other webskins we've created is the arrayProperties. Use the following code to display the destinations for each safari object:

<h3>Destinations</h3>
create the destination object
<cfset odestination='createObject("component",application.types.destination.typepath)'></cfset>
loop over and display teasers for all destinations
<cfloop from="1" index="dest" to="#arrayLen(stObj.aDestinations)#"></cfloop>
<cfset html="&lt;/td"></cfset>
oDestination.getView(objectid="#stObj.aDestinations[dest]#",template="displayTeaserStandard",
alternateHTML="Teaser not found") />
<cfoutput>#html#</cfoutput>



## 7.15 LAB

Now that you are familiar with Wizards go ahead and modify our destination type and change its cfproperties to make it a multi-step Wizard and use the richTextEditor for both the "description" and "attractions" steps.

#### 7.15.1 Permissions

It is best practice to define a new permission for each Custom Tab to allow for maximum flexibility when applying security to your site. Perform the following tasks to create a new permission, modify the xml schema and assign a policy group to use the new permission:

- Login to FarCry
- Click on the "Security" tab
- Choose "Policy" from the sub-section dropdown
- Click the "Create Permission" link
- Set "Permission Name" to be "MainNavSafarisTab"
- Set Permission Type to "PolicyGroup"
- Click the "Create Permission" button

farcry tell it to someone w	vho cares				
Overview Site Con	tent Admin	Reporting	Security	Safaris	
Policy	Create Pe	rmission			
SETUP * Test Policy Setup	Permission Nam	ne: M	lainNavSafaris	Tab	
POLICY GROUPS Policy Groups Policy Group Permissions Create Policy Group Copy a Policy Group Show Policy Group Mappings Map Policy Group	Permission Note	es: P cr e: P	ermission to vi ustom tab. olicyGroup	ew the Safaris	
PERMISSIONS Permissions Create Permission Rebuild Permissions Permissions Map Old Permission Map Webtop Permission Map	0		Create Perm	ission	

Change you customadmin.xml file to now use the permission "MainNavSafarisTab" instead of "MainNavAdminTab". Save the file and update your application scope to refresh the schema. You will now notice that our Custom Tab is no longer visible. This is because we have not assigned the new permission to our user's policy group.

farcry tell it to someone who cares				
Overview Site	Content Admin	Reporting Security	Safaris	
Policy				
		LinkApprove	Allows approval of dmLink objects	
SETUP		LinkCanApproveOwnContent	Allows user to approve their own dmLink objects	
<ul> <li>Test Policy Setup</li> </ul>		LinkCreate	Allows creation of dmLink objects	
POLICY GROUPS		LinkDelete	Allows deletion of dmLink objects	
Policy Groups     Delicy Croup Derminations		LinkEdit	Allows editing of dmLink objects	
Policy Group Permissions     Create Policy Group     Copy a Policy Group		LinkRequestApproval	Allows user to request approval for dmLink objects	
<ul> <li>Show Policy Group Mappings</li> </ul>		<u>MainNavAdminTab</u>	Access to the admin tab in the header	
Map Policy Group		MainNavContentTab	Access to the Content tab in the header	
PERMISSIONS		<u>MainNavHelpTab</u>	Allows access to the main help tab	
Permissions		MainNavMyFarcryTab	Access to the My Farcry tab in the header	
Create Permission     Rebuild Permissions     Permissions Map     Old Permission Map     Webtop Permission Map		MainNavReportingTab	Allows access to the main reporting tab	
		<u>MainNavSafarisTab</u>	Permission to view the Safaris custom tab.	
		MainNavSecurityTab	Access to the security tab in the header	
		MainNavSiteTab	Access to the site tab in the header	
		ModifyPermissions	N/A	

To assign the permission to a policy group perform the following tasks:

• Click on the "Security" tab

- Choose "Policy" from the sub-section dropdown
- Click on the "Policy Group Permissions" link
- Select "SysAdmin" from the "Policy Group" drop down list (SysAdmin is the default policy group for the farcry user)

## Permissions on PolicyGroup (PolicyGroup)

Switch policy groups:	SysAdmin 🔽
	Anonymous
	Contributors
	Member
<b>•</b> • • •	Publishers
Permission	SiteAdmin
Admin	SysAdmin

- Scroll down to the "MainNavSafarisTab" and click the "No" button to change it "Yes"
- Scroll back up and click the "Update" button

If you refresh the screen you will now see the "Safaris" Custom Tab re-appear. Permissions can be set against the Parent Tab, Sub Tab and Link in the customadmin.xml file.

## 7.16 UNIT REVIEW

In this unit you have learnt the following:

- How to define a Custom Type
- How to deploy a Custom Type
- How to add/remove Custom Type properties
- How to use a Custom Type in the navigation tree
- How to use a Dynamic Custom Type
- How to create Custom Tabs
- How to create new permissions
- How to assign permissions to Policy Groups
- How to define a Wizard
- How to incorporate a Rich Text Editor
- How to use the FarCry category tree

## 8 Custom Rules

We now have our dynamic Content Type functional from an administration perspective and we can preview the safari objects but we haven't as yet displayed it in the Solar Safaris website itself. To do this we are going to create a Custom Publishing Rule that will list all Safaris that have not yet departed and have a status equal to that of our current viewing mode.

Creating Custom Rules is a very similar process to creating Custom Types. First we need to define our Rule component and the properties it will have, then we deploy it via the COAPI screen. Also required are edit and execute methods, the edit method being the form we use in the Container Management Screen and the execute method being the code that publishes content to the container.

## 8.1 RULE DEFINITION

Create a new file called ruleSafaris.cfc and save it in the C:\farcry\projects\solar\_safaris\packages\rules directory. Add the following code to define our rule:

#### <cfcomponent

displayname="Safari Listing Rule" extends="farcry.core.packages.rules.rules" hint="Displays all safaris that haven't departed">

<cfproperty ftSeq="1" name="intro" type="longchar" hint="Intro text" required="false" ftType="longchar" ftLabel="Intro" />

<cfproperty ftSeq="2" name="displayMethod" type="string" hint="Display method to render objects with" required="true" ftLabel="Display Method" ftDefault="displayTeaserStandard" ftType="webskin" ftprefix="displayTeaser" ftTypename="Safari" />

</cfcomponent>

The above code first sets the custom rule to extend the core rules type which means it will inherit the following properties:

- ObjectId
- Label

The rules component also extends FourQ which means it inherits the same standard methods that the types object does, eg createData, setData etc.

### 8.2 DEPLOY

Save the file and deploy the rule via the COAPI screen. Instead of using the "Type Classes" link, use the "Rule Classes" link (the functionality is identical).

farcry tell it to someone wh	no cares	norting Security Safaris		FarCry Content (Logged in: (Logged in:
COAPI Management 💌	Rule Classes			
COAPITOOLS	Integrity	Component	Deployed	Deploy
<ul> <li>Type Classes</li> <li>Rule Classes</li> </ul>	~	ruleRandomFact	1	n/a
COAPI Metadata	~	ruleSafaris	×	Deploy
+ COAPISchema	~	ruleNews	1	n/a
DIAGNOSTICS	~	ruleChildLinks	✓	n/a
<ul> <li>Fix RefObjects Table</li> </ul>	~	ruleXMLFeed	✓	n/a
<ul> <li>Reattach Orphan Folders</li> <li>Reattach Orphan Categories</li> </ul>	~	ruleEvents	✓	n/a
	<ul> <li>✓</li> </ul>	ruleLinks	~	n/a
	1	ruleEventsCalendar	~	n/a
	✓	ruleText	×	n/a
	~	ruleHandpicked	×	n/a

The rule is now deployed in the database and metadata saved to the application scope. Browse to the "Safaris" page of the Solar Safaris website and then use the widget menu to view design mode to display the page containers. Edit the container on the page to bring up the Container Management Screen listing all the available rules. You should now see the new "Safari Listing Rule" displayed in the select box.

oottom			
CONFIGURE RULES	CONTAINER CONTENT	REFLECTIONS	
Available Rule Types:	Handpicked Rule News Rule Random Fact Rule Safari Listing Rule Text Rule XML Feed Publishing Rule		Move to Active Rules
Active Rules	Commit Changes		<u>Move up</u> <u>Move down</u> <u>Delete Rule</u>
Displays all safaris tha	it have not departed		

Once again thanks to the FarCry framework (and FormTools) we don't need to create a form to handle editing the DisplayMethod and Intro of our rule.

💫 bottom	
CONFIGURE RULES	CONTAINER CONTENT REFLECTIONS
Active Rules For This Container:	[1] Safari Listing Rule 💌
Display method:	
	Update

## 8.3 TEASER WEBSKIN

You will notice that there are no "Display Methods" listed in the update form. This is because we haven't created any teaser methods for the Safari custom type as yet. Go ahead and create a webskin called displayTeaserStandard for the Safari object. Refreshing the Container Management Screen you should now see your display method listed. Select this new display method and enter some intro text for your listing.

oottom 🍾	
CONFIGURE RULES	CONTAINER CONTENT REFLECTIONS
Active Rules For This Container:	[1] Safari Listing Rule 💌
Display method:	displayTeaserStandard.cfm 💌
Intro:	Listed below are our currently available Safaris.
	Update

Click the "Update" button to save the rule settings.

### 8.4 EXECUTE METHOD

If you now browse to your "Safaris" page in the Solar Safaris website you will see the following error:

": No execute method specified"

We have the ability (from FormTools) to save rule information but we haven't created the execute method that will actually determine what objects are to be displayed and display them. Add the following code to your safaris.cfc file:

```
<cffunction access="public" name="execute" output="false" returntype="void">
       <cfargument name="objectID" required="Yes" type="uuid" default="">
       <cfargument name="dsn" required="false" type="string" default="#application.dsn#">
       <cfargument name="dbowner" required="false" type="string" default="#application.dbowner#">
       <!--- set up vars --->
       <cfset var stObj = getData(arguments.objectid)>
       <cfset var gGetSafaris = ">
       <cfset var stlnvoke = structNew()>
       <!--- get safaris that haven't left yet and have status equal to current view mode --->
       <cfquery datasource="#arguments.dsn#" name="qGetSafaris">
                SELECT*
                FROM #arguments.dbowner#safari
                WHERE status IN ('#ListChangeDelims(request.mode.IValidStatus,"',",",")#')
                         departureDate > #now()#
                AND
                ORDER BY departureDate
       </cfquery>
       <!--- check if intro has been entered --->
       <cfif len(trim(stObj.intro)) AND gGetSafaris.recordCount>
                <!--- output intro --->
                <cfset arrayAppend(request.alnvocations,stObj.intro)>
       </cfif>
       <!--- loop over safaris --->
       <cfloop query="qGetSafaris">
                <!--- call display method for safari --->
                <cfset stInvoke = structNew()>
                <cfset stInvoke.objectID = qGetSafaris.objectID>
                <cfset stInvoke.typename = application.stCoapi.safari.packagePath>
                <cfset stInvoke.method = stObj.displayMethod>
                <cfset arrayAppend(request.alnvocations,stInvoke)>
       </cfloop>
</cffunction>
```

Viewing the "Safaris" page should now show the intro text and then list all your safari objects using the new teaser webskin we created.



## 8.5 LAB

Go ahead and create a second custom rule that will display a random destination on the home page. You will need to add a container to the home page webskin, create the rule, deploy it via the COAPI screen and assign it to the home page. The end product should look something like this:


# 8.6 UNIT REVIEW

In this unit you have learnt the following:

- How to define a Custom Rule
- How to deploy a Custom Rule
- How to assign a Custom Rule to a container
- How to define an execute method

# 9 Advanced Topics

There are numerous advanced features of FarCry that we haven't covered as yet and we won't be able to cover all of them in this course. We will however cover some of the more important and often-used features including:

- Caching
- Security
- Extending Core Types
- Friendly URLs

# 9.1 CACHING

Caching is used to reduce the loads on both the ColdFusion and database servers. Anything can be cached within your website, but it is a standard practice to cache things that are server intensive, things like navigation and containers. Caching can be set to expire in a matter of seconds or can be set to last for weeks, depending on how often your content is updated and the load on your server.

FarCry makes caching a very simple task by providing a cache custom tag. You simply wrap the code you want to be cached with the custom tag, set a few attributes and the next time someone views that page the cache is generated. FarCry also provides caching administration by allowing you to view current caches, flush (delete) caches, clean expired caches and auto cache the entire website. You are also able to select a view mode called "Show Latest" that flushes all caches on the page currently being viewed.

# 9.1.1 Cache custom tag

In the Solar Safaris website we will start by caching the primary navigation. To do this open the C:\farcry\projects\solar\_safaris\webskin\includes\header.cfm file.

Add the following import line to the top of the page.

```
<!--- import webskin tag library --->
<cfimport taglib="/farcry/core/tags/webskin" prefix="skin">
```

Then replace the primary navigation block with the below code. The only change is that it is now wrapped with <skin:cache> tags.

```
<!--- cache the primary nav for 7 days --->
<skin:cache cacheBlockName="Navigation" cacheName=" PrimaryNav" days="7">
       <!--- set up where clause filter --->
       <cfset navFilter=arrayNew(1)>
        <cfset navfilter[1]="status IN (#listQualify(request.mode.lvalidstatus, """)#)">
       <!--- get navigation nodes --->
       <cfset qNav = request.factory.oTree.getDescendants(objectid=application.navid.home, depth=1,
afilter=navFilter,blncludeSelf=1)>
        --- loop over navigation and display each node --->
       <cfloop query="qNav">
                 <a
href="#application.url.conjurer#?objectid=#qNav.objectid#">#qNav.objectname#</a>
                 <!--- check if object is not the last returned object --->
                 <cfif qNav.currentRow neq qNav.recordCount> | </cfif>
        </cfloop>
</skin:cache>
```

The <cache> tag takes the following attributes:

• cacheBlockName (name for a group of related caches)

- cacheName (name for this individual cache)
- days (number of days to cache for)
- hours (number of hours to cache for)
- minutes (number of minutes to cache for)
- seconds (number of seconds to cache for)

# 9.1.2 Show Latest

If you now add a new page under the home navigation node called "Travel Agents" and then refresh the home page you will that it doesn't appear in the menu. That is because the menu is cached. To clear the cache and make the newly added page appear use the widget menu and select the "View Latest" option.





To cache the navigation now again including the new "Travel Agents" page, select "Show Cached" from the widget menu.

## 9.1.3 Lab

Go ahead and cache the footer navigation menu and the containers on both the "Destinations" and "Safaris" pages. As the "Safaris" page has date based information you probably only want to cache this page for a maximum of 1 day. Your "Cache Summary" page under the "Admin" table now may display as:

#### 9.1.4 Advanced Webskin Caching

It is possible to cache an entire webskin, not just a small content section. FarCry can cache the webskin whether it is an entire page (including header and footer), or if it is just a teaser.

Do take advantage of webskin caching you can add the following to your cfcomponent tag for the type you want to cache.

<cfcomponent name=""

IObjectBrokerWebskins="displayPageHome:15,displayPageStandard,displayTeaser,displayPageLandin g" objectBrokerWebskinTimeOut="60">

objectBrokerWebskinTimeOut sets the default timeout (in minutes) for you type. You can however override the timeout for each webskin by passing a webskin:timeout pair.

One thing you need to remember with webskin caching is that when you make a change and "Save" a content object which uses that webskin, the cache will be cleared. This is to assist content editors in making sure the most up to date content is displayed.

Be careful however with any dynamic content which has been added to your page via a container. All content is cached in the webskin but if container data is modified it might not be cleared from the cache.

Block	Number of Caches	Expired Caches	Clean	Flush
Content	5	0		
Navigation	2	0		
Refresh			Clean	Flush

#### 9.2 SECURITY

Securing content on a website is a highly request feature. FarCry provides this functionality by assigning view permissions to each navigation node in the hierarchy. If the value is not set for a particular node, it inherits the value from its parent. By default all Policy Groups have view permission for the entire site.

#### 9.2.1 Set Anonymous View Permission

To secure a section of the Solar Safaris site, such as the "Travel Agents" page so that only registered travel agents could view the content we first need to turn off anonymous view access to the "Travel Agents" page.

- Login to FarCry
- Click on the "Site Tab"
- Right-click on the "Travel Agents" node

• Select the "Permissions" link

Solar Safaris	For an adventure out of this world				
Overview Site	Content Admin	Reporting Security	Solar Safaris		
Overview       Site         Home       Image: Contact Us         Hore       Image: Contact Us         Home       Image: Contact Us         Hore       Image: Contact Us         Hore <th>Content Admin  Content Iter Content Iter Content Iter Content Iter Navigation A Created by: Date Create Locking: Last Update Current Sta ObjectID  Stats Audit Propertie</th> <th>Reporting     Security       m Title:     m       m Type:     Alias(es):       Alias(es):    </th> <th>Solar Safaris Travel Agents Navigation farcry 2/22/07 Unlocked Feb 22, 2007 farcry draft E7A57C19-C560-3ED2-B41DE8C74E8359D7</th>	Content Admin  Content Iter Content Iter Content Iter Content Iter Navigation A Created by: Date Create Locking: Last Update Current Sta ObjectID  Stats Audit Propertie	Reporting     Security       m Title:     m       m Type:     Alias(es):       Alias(es):	Solar Safaris Travel Agents Navigation farcry 2/22/07 Unlocked Feb 22, 2007 farcry draft E7A57C19-C560-3ED2-B41DE8C74E8359D7		

- Select "Anonymous" from the "Policy Group" drop down
- Set the "View" permission to "No"
- Click the "Update" button

Policy	group: Anonymous 💌	Update	
	Permission	State	Inherited
	Approve	Inherit	No
	CanApproveOwnContent	Inherit	No
	ContainerManagement	Inherit	No
	Create	Inherit	No
	Delete	Inherit	No
	Edit	Inherit	No
	RequestApproval	Inherit	No
	SendToTrash	Inherit	No
	View 🤇	No	NO

PERMISSIONS ON TRAVEL AGENTS (DMNAVIGATION)

If you now log out of FarCry and try and view the "Travel Agents" page you will be prompted for a username and password. Using the default "farcry/farcry" combination will allow you view the page as the SysAdmin view permission is still set to "Yes".

# 9.2.2 Create New User

Let's now create a new user and make them part of the "member" policy group and check that they can log in and see the "Travel Agents" page.

- Login to FarCry
- Click on the "Security" tab
- Click the "Create User" link
- Add a new user with a login of "ABCTravel" and password of "abc"
- Click the "Create User" button
- The user is not a member of any Policy Groups at this stage so click the "Manage Groups" button
- Make the user a member of the "Member" group and click the "Update" button.

# MANAGE USER GROUPS (ABCTRAVEL)

mber of: ember	->>	Not a member of: Contributors News Contributor Publishers SiteAdmin SysAdmin
	Update	

#### 9.2.3 View the page

Now that we have a new user, let's now check we can log in and view the secured page as planned.

- Go back to the "Site" tab and ensure that the "Member" policy group has view permissions on the "Travel Agents" page.
- Log out of FarCry
- Browse to the "Travel Agents" page.
- Login using "ABCTravel/abc"
- Try browsing to the FarCry administration screen (you should not be able to do this a member. It will log you out and prompt you for a username/password)

#### 9.3 EXTENDING CORE TYPES

All core FarCry types and rules can be extended to allow for new properties and methods, and changes to existing methods. For the Solar Safaris website we want to display some trade event information, where Travel Agents would sign up to exhibitions/conferences regarding Solar Safaris. Solar Safaris would like to charge for these events. The core dmEvent object has all the properties we require except for a cost field. What we need to do is extend the dmEvent type, add a new property, deploy this new property and modify the edit handler to allow the new property information to be entered.

#### 9.3.1 Type definition

The first thing we need to do is define the dmEvent type as an extended type. This means the inheritance chain will now be:

Solar\_Safaris dmEvent -> Core dmEvent -> Core types -> FourQ

Create a new file called dmEvent.cfc and save it in the C:\farcry\projects\solar\_safaris\packages\system directory. Add the following code:

<cfcomponent< td=""></cfcomponent<>
extends="farcry.core.packages.types.dmEvent"
displayname="Events"
hint="Dynamic events data"
bSchedule="1">
<pre><cfproperty <="" ftfieldset="Event Overview" ftseq="6" ftwizardstep="General Details" name="Cost" pre=""></cfproperty></pre>
type="string" hint="Display method to render." required="false" ftType="string" ftLabel="Cost" />

</cfcomponent>

## 9.3.2 COAPI

Login to FarCry and deploy the new "cost" property via the Types COAPI screen.

~	dmLink	dmLink		✓		n/a
🗙 See Belo	w dmEver	dmEvent		×		n/a
The following CFC properties conflicts exist :						
Property Deployed Type Action				Action		
Cost	×	string	Do	Nothing	•	Go
		1	Do De	Nothing ploy Property		·

Now login in to FarCry and create a new Event object under the "Content" tab. You will see our new "Cost" field in the first step of the Wizard.

Solar Safaris	For an adventure out of this world
Overview Site	Content Admin Reporting Security Solar Safaris
Dynamic Content	General Details
<ul> <li>♦ News</li> <li>♦ Events</li> <li>♦ Links</li> <li>♦ Facts</li> </ul>	- Event Overview
	Title : Solar Safaris Expo
	Start Date : 22 Feb 2007 03:19 PM
	End Date : 27 Feb 2007 03:19 PM
	Location :
	Content Template : Standard Page 💙 Cost : 1000

#### 9.3.3 Webskin

Go ahead and create two webskin files for the dmEvent object, on called displayPageStandard and the other called diplayTeaserStandard. Then go and add the "Event Listing Rule" to the container on the "Travel Agents" page. The final result should look something like this:



You can easily extend any other core type or rule and add new methods where needed. One common extension is to add new properties to the dmProfile type to capture member details.

## 9.4 FRIENDLY URLS

The Friendly URL plugin was created by Stephen "Spike" Milligan and has been integrated into FarCry to allow links to be entered using the titles of the navigation pages instead of passing through the objectid. Eg:

http://localhost/go/destinations/venus

instead of

http://localhost/solar/index.cfm?objectid=7F2B7A9F-20E0-64AB-AFDCEC35ED29A43C

This has a number of benefits both to the user and site administrator:

- Better presentation, the url looks nicer
- User friendly, the user can tell where they are in the site from the url
- Search engine friendly, search engines will index deeper into your site if your url doesn't contain url parameters

To incorporate friendlyURLs into your site you need to download the plugin (follow the instructions at <a href="http://docs.farcrycms.org:8080/confluence/display/FCDEV30/Friendly+URL+Configuration">http://docs.farcrycms.org:8080/confluence/display/FCDEV30/Friendly+URL+Configuration</a>) and then set the plugins configuration file to use friendlyURLs.

Solar Safaris For an adventure out of this world				
Overview Site	Content Admin	Reporting Sec	urity Solar Safari	
General	FarCry Int	ernal Config Fi	les	
CONFIGURATION * Config Files * Custom Config Dump Config CUSTOM ADMIN XML * Dump Custom Admin XML	plugins Config FU GEOLOCATOR JTIDY	Yes No No		
SCOPE DUMP + Scope Dump	Update Cor	nfig		
CACHE MANAGEMENT		' <u>ro</u>		
SCHEDULED TASKS * Scheduled Tasks	<ul> <li>FUSettings</li> <li>general</li> </ul>			
MESSAGE CENTRE + Messages	<ul> <li>HIMLArea</li> <li>image</li> <li>overviewTre</li> </ul>	e		
MANAGE FRIENDLY URLS Manage Friendly URLs Rebuild Friendly URLs	<ul> <li>plugins</li> <li>soEditor</li> <li>soEditorPro</li> <li>tinyMCE</li> <li>verity</li> </ul>	2		

Once you update the config you will see two new menu items appear under the "Admin" tab.

Solar Safaris	For an advent	For an adventure out of this world			
Overview Site	Content	Admin	Reporting		
General 💌					
CONFIGURATION Config Files Custom Config Dump Config					
CUSTOM ADMIN XML	-				
SCOPE DUMP + Scope Dump					
CACHE MANAGEMENT <ul> <li>Object Broker Report</li> <li>Cache Summary</li> <li>Auto Cache</li> </ul>	۲				
SCHEDULED TASKS					
MESSAGE CENTRE Messages					
MANAGE FRIENDLY URLS Manage Friendly URLs Rebuild Friendly URLs					

The "Rebuild Friendly URLs" section provides you with all core content types to build friendly URLs for. If you choose dmNavigation, FarCry goes through the site tree from the Home node down and creates a friendly URL entry for each nav node. Note that depending on the size of your project (i.e. the amount of navigation items) this may take a while to process. Therefore care should be taken particularly when running on a production system.

To actually generate the links throughout your site as friendlyURLs you need to use the <buildLink> custom tag instead of hard coding <href> tags. You simply pass the objectid to the buildLink tag, it checks to see if friendlyURLs are installed and configured and then checks to see if the object passed in has a friendlyURL entry. If it does the friendlyURL is returned, if not, the standard objectid= url is returned.

Modify your header.cfm file to incorporate friendlyURLs. Make sure you import the webskin tag library, then replace the line that generates the <a href> tag with the following:

<skin:buildLink objectid="#qNav.objectid#">#qNav.objectName#</skin:buildLink>

If you have installed, configured and set friendlyURLs for your site the primary navigation will now show as friendly urls. If you haven't configured friendly URLs then the links will show as they did before, but as soon as you install and configure the plugin the links will change, without having to update the code. It is best practice to use the <buildLink> custom tag to generate navigation links even if you don't have friendlyURLs installed because it saves you a lot of code re-write if you do eventually install them.

# **10 ADDITIONAL ONLINE MATERIAL**

## 10.1 FARCRY CMS OPEN SOURCE COMMUNITY

The official website of the FarCry CMS development community. http://www.farcrycore.org/

#### **10.2 PUBLIC FORUMS**

## 10.2.1 FARCRY-DEV (PUBLIC)

farcry-dev@googlegroups.com

Aimed at managing support for FarCry open source developers. Anyone making enquiries about modifying or extending or deploying the code base should be referred to this list.

To join the mailing list, go to: <u>http://groups.google.com/group/farcry-dev/about</u> Old web based archive: <u>http://www.mail-archive.com/farcry-dev@lists.daemon.com.au/</u> Alternative web based interface: <u>http://www.nabble.com/FarCry-f621.html</u>

# 10.2.2 FARCRY-USER (PUBLIC)

#### farcry-user@googlegroups.com

Aimed at managing support for FarCry open source users. Anyone making enquiries about adding, editing or managing content should be referred to this list.

To join the mailing list, go to: <u>http://groups.google.com/group/farcry-user/about</u> Alternative web based interface: <u>http://www.nabble.com/FarCry-f621.html</u>

# 10.2.3 FARCRY-BETA (PUBLIC)

farcry-beta@googlegroups.com

The beta list is focused at providing a discussion environment and ad hoc support forum for developers getting to grips with and developing changes for the next generation of FarCry CMS.

To join the mailing list, go to: http://groups.google.com/group/farcry-beta/about

# 10.3 BUG TRACKING

Daemon manages and maintains an open and comprehensive issue tracking system for active releases of the FarCry CMS platform. This includes roadmaps for future development, and schedules for maintenance releases and hot fixes.

If you need to report an issue, check on the status of an issue, request a new feature or even submit a code change for review you'll need to visit http://bugs.farcrycms.org/

#### 10.4 FARCRY WIKI

FarCry WIKI is a community effort to generate much needed user and developer documentation for this fantastic platform. Sign-up and help out — even if its only to comment and ask questions! <u>http://docs.farcrycms.org/</u>



This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 2.5 License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-nd/2.5/au/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

This work is copyrighted Daemon Pty Limited 2007, http://www.daemon.com.au/