

Подготовка рабочей станции к разработке в Joomla!

Разработка программ для Joomla! обычно выполняется на локальной рабочей станции. С этой целью следует установить программное обеспечение веб-сервера, необходимое для выполнения Joomla на локальной машине. Кроме того, нужно организовать редактирование исходных файлов PHP и прочих файлов, составляющих приложение Joomla.

Правда, все это программное обеспечение свободно доступно для загрузки и может быть установлено во всех распространенных операционных системах, включая Windows, Mac OS X и Linux. Кроме того, имеется немало других свободно доступных средств для правки программных файлов Joomla, как будет показано далее в этой главе, где поясняется также порядок настройки рабочей станции для выполнения Joomla, создания и правки файлов, необходимых для написания прикладных программ.

Если вы уже настроили свою рабочую станцию под Joomla и разработку средствами Joomla, можете пропустить раздел “Требования к выполнению Joomla”. А если вы уже пользуетесь такой интегрированной средой разработки, как Eclipse или NetBeans, то можете не читать и раздел “Рабочие инструменты”.

Требования к запуску Joomla

Как пояснялось в главе 1, для нормальной работы Joomla требуется веб-сервер (Apache или Microsoft Internet Information Services — IIS), язык программирования PHP и база данных (чаще всего MySQL). Каждый из этих программных компонентов может быть загружен и установлен отдельно от соответствующих веб-сайтов. Но, как правило, Apache, PHP и MySQL проще установить общим пакетом. А в Mac OS X и Linux эти программные компоненты могут быть уже установлены по умолчанию.

Следует иметь в виду, что независимо от способа установки (по отдельности или общим пакетом) конечный результат остается неизменным: Apache, PHP и MySQL устанавливаются на локальной машине. Единственное отличие состоит в том, что каждый пакет может немного отличаться своей комплектацией, именами папок и отдельными параметрами настройки по умолчанию.



Важное примечание по поводу безопасности

В этой главе предполагается, что для разработки и тестирования создается локальная среда, а тестируемые веб-сайты не связаны со Всемирной паутиной (или просто веб). Следовательно, вопросы безопасности веб-сервера не столь актуальны, как в том случае, когда веб-приложение размещается на реальном сервере. Это, разумеется, не означает, что не нужно уделять внимание защите самих программ, разрабатываемых для Joomla, от внешних атак.

Если вы работаете на удаленном сайте, доступном в веб, непременно организуйте его защиту — даже на время разработки. Это можно сделать самыми разными способами, в том числе используя файл с расширением `.htaccess` в Apache, требующий ввода пароля для доступа по указанному URL. В таком случае вам, возможно, потребуется консультация специалиста по безопасности веб вообще и Joomla в частности.

Папка DocumentRoot в Apache

Обо всех упомянутых выше программных компонентах и средах нужно знать следующее. Как пояснялось ранее, веб-сервер Apache представляет собой программное обеспечение связи браузера с веб-сайтом. Если веб-сервер Apache устанавливается локально, то URL по веб-адресу `http://localhost` сопоставляется с папкой на локальной рабочей станции. Как только этот URL будет введен в браузере, Apache произведет поиск файла `index.php` или `index.html` в папке DocumentRoot, находящейся на локальном компьютере. Если установлен пакет веб-сервера XAMPP, то папка DocumentRoot обычно находится по пути `C:\xampp\htdocs`, а если установлен пакет WampServer — то по пути `C:\Program Files\wamp\www`.

Папку DocumentRoot можно заменить на другую, внося соответствующие коррективы в файл `httpd.conf`, находящийся в папке `apache/conf`. Например, в пакете XAMPP этот файл находится по пути `c:\xampp\apache\conf`. А в пакете WampServer файл `httpd.conf` можно отредактировать в самом приложении, выбрав команду **Config Files (Конфигурационные файлы)** ⇒ `httpd.conf`.

Определившись с папкой DocumentRoot, вы можете затем выбрать место для установки файлов Joomla. Поиск подходящего места начнется с любого URL, введенного в браузере с префиксом `http://localhost`, т.е. с папки DocumentRoot, и следуя далее вниз по древовидной структуре папок. Так, если корневая папка для документов находится по пути `c:\xampp\htdocs`, то Joomla можно установить в папке `c:\xampp\htdocs\joomla_development\joomla_25`. В этом случае URL, который нужно ввести для доступа к административной части данной установки, будет следующим: `http://localhost/joomla_development/joomla_25/administrator`. Веб-сервер Apache будет искать файл `index.php` в папке `joomla_development/joomla_25/administrator`, вложенной в папку DocumentRoot.

Конечно, Joomla можно было бы установить и в самой папке DocumentRoot, но тогда в ней нельзя было бы установить и все остальное. А следуя приведенному выше примеру установки, можно одновременно начать несколько проектов в отдельных подпапках, вложенных в папку DocumentRoot. Итак, с учетом всего сказанного выше, перейдем к рассмотрению различных вариантов установки требующегося программного обеспечения на разных платформах.

Получение своевременных инструкций

Программы и комплекты, рассматриваемые в этом разделе, постоянно обновляются новыми версиями. Со временем выходят также новые версии Windows, Linux и Mac OS X. Поэтому материал этого раздела актуален лишь на момент написания книги. А впоследствии он может устареть или оказаться неполными в силу неизбежных изменений в программном обеспечении как самого веб-сервера, так и платформы, для которой он предназначен.

По этой причине для каждого комплекта здесь приводятся ссылки, по которым трудно получить своевременные инструкции. Помимо этих ссылок, здесь приводится также ссылка на статью, опубликованную на веб-сайте документации по Joomla, с подробными инструкциями по установке (http://docs.joomla.org/Setting_up_your_workstation_for_Joomla!_development). Эта статья обновляется по мере изменений в соответствующем программном обеспечении.

Платформа Windows

В различных версиях Windows (Windows 7, Windows Vista и Windows XP) программные компоненты Apache, PHP и MySQL не устанавливаются по умолчанию. Двумя наиболее распространенными стандартными пакетами для установки динамической веб-среды в Windows являются XAMPP и WAMP.

Наиболее распространенным в Windows считается пакет XAMPP (<http://www.apachefriends.org/en/xampp.html>). Он легко устанавливается и хорошо документирован. Кроме того, он доступен в Linux и Mac OS X. Подробные инструкции по его установке в Windows приведены по адресу <http://www.apachefriends.org/en/xampp-windows.html>.

Еще одним стандартным для Windows является пакет WampServer. К его преимуществам относится простота перехода к другим версиям Apache, PHP и MySQL. И это весьма значительное преимущество, если, например, требуется протестировать программу в разных версиях PHP. Пакет WampServer доступен по адресу <http://www.wampserver.com/en> или <http://www.wampserver.com/ru>. Он очень просто устанавливается в Windows и хорошо документирован на сайте по указанному выше адресу.

Примечание по поводу Microsoft IIS и SQL Server

Joomla может выполняться как на веб-сервере Apache, так и на веб-сервере Microsoft IIS. Но поскольку веб-сервер Apache имеет более долгую историю поддержки, а большинство веб-сайтов Joomla работают именно на нем, то разработчики в основном отдают предпочтение именно ему. По этим причинам в качестве сервера для разработки веб-сайтов рекомендуется Apache.

С версии 2.5 в Joomla поддерживается также база данных Microsoft SQL Server. Но в качестве основной базы данных для разработки веб-приложений в Joomla рекомендуется все же MySQL.

Платформа Mac OS X

Как упоминалось выше, версия пакета XAMPP имеется и для платформы Mac OS X. Ее можно загрузить по адресу <http://www.apachefriends.org/en/xampp-macosx.html>. В этом случае папка DocumentRoot по умолчанию находится по пути `/Applications/XAMPP/htdocs/`.

Имеется также пакет MAMP, предназначенный исключительно для платформы Mac OS X. Его можно загрузить по адресу <http://www.mamp.info/en/mamp/index.html>. В этом случае папка DocumentRoot по умолчанию находится по пути `/Applications/MAMP /htdocs/`.

В Mac OS X и Linux имеется также возможность воспользоваться версиями Apache, MySQL и PHP, распространяемыми как составная часть собственной операционной системы. А поскольку Mac OS X является производной от Linux, то эти программные компоненты могут работать и в Mac OS X как ее собственные.

Если у вас нет опыта работы с этими программными компонентами, вам, вероятно, будет проще выбрать один из существующих стандартных пакетов, чем пользоваться собственными их версиями. Кроме того, стандартные пакеты XAMPP и MAMP, как правило, содержат более актуальные версии Apache, MySQL и PHP, чем версии, входящие в комплект с операционной системой Mac OS X.

Важное примечание

Веб-сервер Apache может быть установлен по умолчанию в операционной системе Mac OS X. В этом случае он определяет порт на компьютере как “прослушивающий”. Если же затем установить вторую версию Apache из пакета XAMPP или MAMP, то и он попытается воспользоваться тем же самым портом, а следовательно, не сможет работать корректно.

Во избежание подобного усложнения придется изменить один из портов. Проще всего, вероятно, изменить порт в пакете XAMPP или MAMP на что-нибудь вроде **8080**. В пакете XAMPP для этого достаточно отредактировать файл `\Applications\XAMPP\xamppfiles\etc\httpd.conf`, изменив в нем строку `Listen 80` на `Listen 8080`. А в пакете MAMP порт можно изменить на панели глобальных параметров настройки, что, в свою очередь, внесет коррективы в файл `httpd.conf`. И в этом случае прослушивающий порт следует изменить на `Listen 8080`.

Платформа Linux

Все, что было сказано выше относительно платформы Mac OS X, справедливо в основном и для платформы Linux. И в этом случае имеется возможность выбрать пакет XAMPP или собственные версии Apache, MySQL и PHP. Начинаям проще установить пакет XAMPP. Сведения о загрузке пакета XAMPP для платформы Linux находятся по адресу <http://www.apachefriends.org/en/xampp-linux.html>.

По умолчанию папка `DocumentRoot` находится по пути `/opt/lampp/htdocs/`. Как пояснялось выше для платформы Mac OS X, вам, возможно, придется изменить используемый по умолчанию прослушивающий порт, если в вашей системе уже работает экземпляр веб-сервера Apache. В пакете XAMPP для этой цели придется внести соответствующие коррективы в файл `/opt/lampp/etc/httpd.conf`. А состоят эти коррективы в замене строки `Listen 80` строкой `Listen 8080`.

Владелец файлов и папок по умолчанию

Права доступа к файлам и папкам в Linux и Mac OS X по умолчанию могут привести к сложностям при попытке установить и отредактировать файлы Joomla. Поэтому следует убедиться в том, что соответствующему пользователю Linux и Mac OS X разрешено выполнять запись в ту папку, где устанавливается Joomla и размещаются файлы Joomla.

Владелец всех файлов, созданных веб-сервером Apache, устанавливается с помощью параметров `User` и `Group` в файле `httpd.conf`. По умолчанию оба эти параметра настроены на “демона” (службу, работающую в фоновом режиме). А это означает, что у пользователя нет полномочий на запись в файлы и папки Joomla. Если речь идет о локальном сайте, предназначенном в основном для целей разработки и тестирования, то параметры `User` и `Group` можно настроить на зарегистрированного в системе пользователя. Благодаря этому будут получены полномочия на создание и удаление файлов и папок Joomla.

Рабочие инструменты

Разработчики Joomla могут пользоваться самыми разными инструментами для создания и редактирования программных файлов Joomla. Так, если задать в дискуссионной группе разработчиков программного обеспечения следующий вопрос: какие инструментальные средства они предпочитают для написания программ, то ответов на него, скорее всего, будет столько же, сколько и мнений по поводу достоинств и недостатков этих самых инструментальных средств. Аналогичное разнообразие мнений сложилось и относительно инструментальных средств, применяемых разработчиками в Joomla. Не существует единого, самого лучшего инструмента для всех и на все случаи жизни. Поэтому в последующих разделах мы постараемся как можно более объективно проанализировать достоинства и недостатки различных инструментальных средства для разработки в Joomla.

Интегрированная среда разработки в сравнении с текстовым редактором

При установке Joomla весь исходный код программы загружается в систему в виде простых текстовых файлов. К числу основных типов файлов в базовой установке относятся PHP (с расширением `.php`), XML (с расширением `.xml`), JavaScript (с расширением `.js`), CSS (с расширением `.css`), а также SQL (с расширением `.sql`). Все подобные типы определяют простые текстовые файлы, а это означает, что они могут быть отредактированы в любом текстовом редакторе. Следовательно, у вас имеется возможность поправить уже имеющиеся файлы Joomla и создать новые без специального программного обеспечения.

В таком случае зачем обременять себя установкой и освоением нового программного обеспечения только для целей разработки в Joomla? Ответ на этот вопрос очевиден: производительность труда — по крайней мере, для многих разработчиков.

Назначение интегрированной среды разработки (IDE) в том и состоит, чтобы повысить производительность труда разработчиков. Такие интегрированные среды разработки, как, например, Eclipse или NetBeans, подробнее рассматриваемые далее в этой главе, предоставляют следующие функциональные возможности для упрощения процесса программирования на PHP.

- Обнаружение синтаксических ошибок в коде по мере его набора.
- Автоматическое завершение набранных частей кода на основании известного синтаксиса.
- Автоматическое отображение сведений о методах и полях редактируемого класса.
- Сравнение двух версий исходного файла (или двух разных исходных файлов) для выявления в них конкретных отличий.
- Упрощение поиска и замены во всей кодовой базе.
- Отладка программ на PHP.

Для выполнения всех перечисленных выше операций совсем не обязательно иметь IDE. Но все эти операции можно выполнять в IDE, как в едином инструментальном средстве.

Еще одно преимущество IDE состоит в том, что она сводит различные задачи разработки в единый проект. В частности, многие разработчики Joomla пользуются системой контроля версий типа Git или Subversion. К этим системам подключаются модули, работающие с IDE, что дает возможность пользоваться системой контроля версий непосредственно в IDE. Другим примером служит программа автоматического построения проектов типа Phing или Ant. И эти программы можно выполнять непосредственно в IDE, что намного упрощает труд многим разработчикам, особенно в крупных проектах.

Но некоторым применение IDE дается дорогой ценой, поскольку освоение самой среды требует дополнительного времени и труда. Кроме того, программные ошибки могут встречаться в IDE чаще, чем в текстовом редакторе, в силу ее большей сложности. И поэтому одни считают, что возникающая в связи с этим неудовлетворенность не стоит потенциального выигрыша в производительности труда. А другие легко мирятся с перспективой освоения нового программного обеспечения и возможными его недоработками, поскольку считают эти недостатки малой ценой, которую приходится платить за несомненные преимущества IDE.

Если вы собираетесь много заниматься разработкой веб-сайтов в Joomla, вам, вероятно, стоит уделить время хотя бы опробованию IDE, чтобы выяснить, насколько эта среда вам подходит. А если вы не собираетесь тратить время и силы на освоение IDE, но желаете сразу же приступить к написанию кода, то IDE вряд ли вам подойдет.

Варианты IDE с открытым кодом

Двумя самыми распространенными IDE для Joomla являются Eclipse и NetBeans. Обе IDE отлично справляются со своей задачей, помогая разработчику работать более продуктивно. В последующих подразделах они рассматриваются по очереди.

Eclipse

Несмотря на то что Eclipse давно уже стала самой распространенной IDE для разработки приложений на Java, в программировании на PHP она стала применяться относительно недавно. В 2007 году специально для разработчиков на PHP были выпущены инструментальные средства, которые так и называются: PHP Development Tools (PDT). Они представляют собой набор подключаемых модулей, адаптирующих Eclipse для работы с кодом PHP.

На момент написания этой книги самой последней считалась версия PDT 3.0. В рамках проекта Eclipse новая версия выпускается каждый год в июне, а вместе с ней зачастую и новая версия PDT. Пакет Eclipse можно свободно загрузить по адресу <http://eclipse.org/downloads>. Для разработки в Joomla обычно выбирается вариант Eclipse for PHP Developers (Eclipse для разработчиков программ на PHP) этого пакета. Подробные инструкции по установке и настройке Eclipse для разработки в Joomla можно найти на сайте документации по Joomla, перейдя по адресу http://docs.joomla.org/Setting_up_your_workstation_for_Joomla!_development.

IDE Eclipse написана на языке Java и работает на платформах Windows, Linux и Mac OS X. Это свободно доступное программное обеспечение с открытым кодом. И хотя не существует официальной статистики по поводу того, каким именно IDE отдают предпочтение разработчики Joomla, тем не менее Eclipse считается едва ли не самой распространенной IDE среди опытных разработчиков данной категории, и не в последнюю очередь потому, что она оказалась первой свободно доступной IDE для разработки на PHP.

Рассмотрим вкратце, чем же Eclipse может помочь в разработке веб-сайтов средствами Joomla. Прежде всего следует обратить внимание на ключевое понятие проекта для таких IDE, как Eclipse или NetBeans. В большинстве текстовых редакторов исходный файл правится постепенно, и самому редактору вообще ничего не известно о других файлах, связанных с исходным, да это ему и не нужно. А в IDE сначала создается проект, в котором хранится группа связанных вместе файлов. Так, если вы работаете над проектом Joomla, то все приложение Joomla содержится в одном проекте. Поначалу такой подход может показаться не совсем понятным, но он вполне оправдан с точки зрения разработки, поскольку разработчикам все время приходится работать с целыми группами связанных вместе файлов.

Создавая проект PHP в Eclipse, вы получаете представление под названием PHP Explorer (Проводник по PHP), где можете просматривать весь проект и открывать любой файл для редактирования. В представлении PHP Explorer, приведенном в качестве примера на рис. 2.1, один файл открыт для редактирования в левом верхнем углу.

А теперь рассмотрим в качестве примера некоторые из упоминавшихся ранее функциональных возможностей IDE. Начнем с выделения синтаксиса. На рис. 2.2 показана программа на PHP, открытая для редактирования в Eclipse. Различные

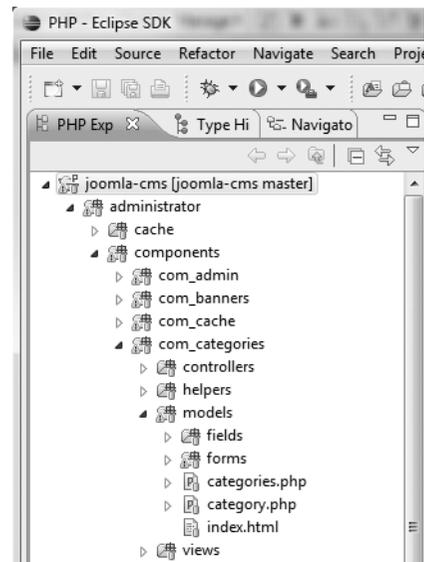


Рис. 2.1. Пример представления PHP Explorer в Eclipse

синтаксические конструкции выделены разным цветом для удобства чтения ее кода. Посмотрим далее, что произойдет, если сделать ошибку в коде. Как показано на рис. 2.3, после оператора `if` оставлены скобки.

Как видите, слева от строки с синтаксической ошибкой появляется знак **X**, а в самой строке обнаруженные ошибки подчеркнуты волнистой линией (на экране эти признаки ошибок отображаются красным цветом). Если навести курсор на знак **X**, то появится всплывающая текстовая подсказка, как показано на рис. 2.4. Эта подсказка вкратце поясняет характер ошибки и указывает возможный способ ее устранения.

Еще одним важным свойством IDE является автоматическое завершение кода. Оно представлено в разных формах. В простейшей форме символы предоставляются попарно. Например, при наборе исходного кода PHP вместе с левой круглой скобкой практически всегда требуется и правая скобка, поэтому редактор автоматически предоставляет обе скобки. Это же относится и к фигурным скобкам, кавычкам и элементам разметки в коде HTML.

```
/**
 * Class constructor.
 *
 * @param array $config A configuration arra
 *                session_name, client
 *
 * @since 11.1
 */
public function construct($config = array())
{
    jimport('joomla.utilities.utility');
    jimport('joomla.error.profiler');

    // Set the view name.
    $this->_name = $this->getName();

    // Only set the clientId if available.
    if (isset($config['clientId'])) {
        $this->_clientId = $config['clientId'];
    }

    // Enable sessions by default.
    if (!isset($config['session'])) {
        $config['session'] = true;
    }
}
```

Рис. 2.2. Выделение синтаксиса в редакторе исходного кода PHP, встроенном в Eclipse

```
98     $if $myVar != 1 {
99         echo 'something is wrong';
100     }
101
```

Рис. 2.3. Отображение синтаксической ошибки в редакторе исходного кода PHP, встроенном в Eclipse

```
97
98
99
100
101
```

Multiple annotations found at this line:
- syntax error, unexpected '['
- syntax error, unexpected '\$myVar'

Рис. 2.4. Всплывающая текстовая подсказка, раскрывающая характер ошибки

Еще одна форма завершения кода имеет место в том случае, если требуется набрать имя метода. В качестве примера на рис. 2.5 набрано выражение `$x = $this->` и затем нажата комбинация клавиш `<Ctrl+Пробел>`.

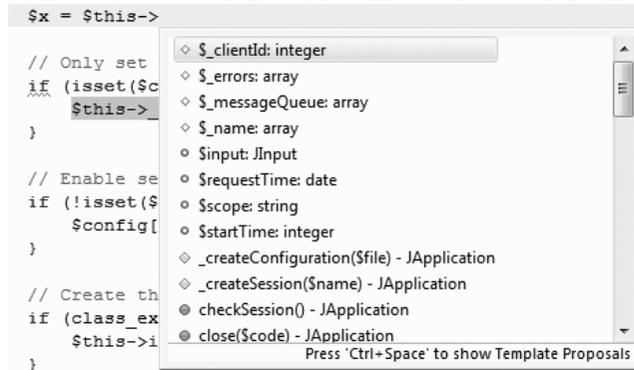


Рис. 2.5. Выбор подходящего метода в редакторе исходного кода PHP, встроенном в Eclipse

Редактору известно, что `$this` — это объект типа `JApplication`, и поэтому он отображает все поля и методы, доступные в данном классе. Для того чтобы выбрать подходящий метод или поле, достаточно щелкнуть кнопкой мыши на его имени или воспользоваться клавишами с изображением стрелок. Как только поле или метод будут выбраны, их имя автоматически вставится в исходный текст программы. Благодаря этому ускоряется набор исходного текста программы и уменьшается вероятность совершения опечаток. По мере набора исходного текста окно поиска автоматически обновляется, чтобы отображать только те имена, которые совпадают с набранным текстом.

Кроме того, в этом окне предоставляются дополнительные сведения о каждом методе или поле. В качестве примера на рис. 2.6, *слева*, выделен метод `redirect()`, а на рис. 2.6, *справа*, представлена форма объявления этого метода со всеми его параметрами и краткая документация на него. Это очень удобно для поиска нужного метода и делается быстрее, чем в отдельном файле или окне.

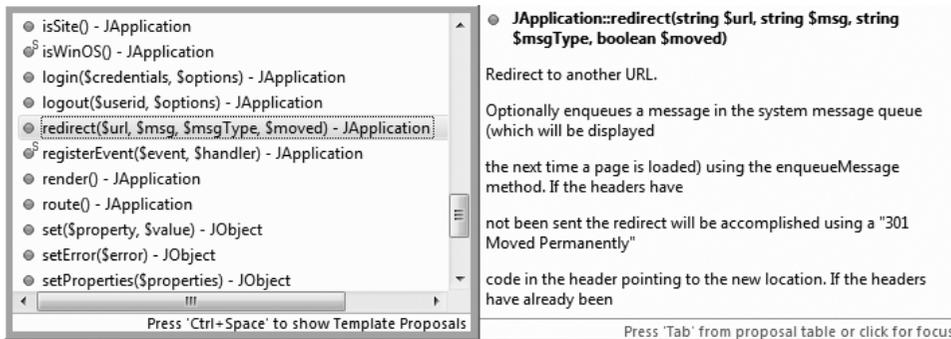


Рис. 2.6. Список параметров и документация на выбранный метод, предоставляемые в редакторе исходного кода PHP, встроенном в Eclipse

Когда правится исходный файл, в Eclipse предоставляется список полей и методов для данного файла в отдельном представлении. Так, на рис. 2.7, *слева*, показана область правки, а на рис. 2.7, *справа*, — соответствующее схематическое представление (Outline).

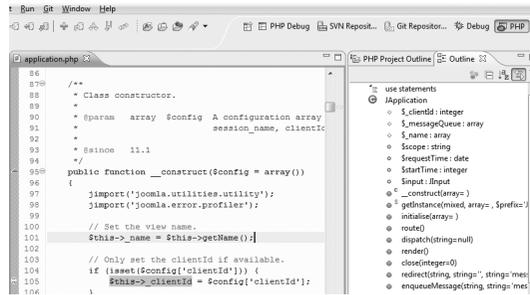


Рис. 2.7. Редактор исходного кода PHP и схематическое представление класса в Eclipse

В схематическом представлении приводятся поля и методы редактируемого класса. А в схематическом представлении проекта PHP (PHP Project Outline) можно посмотреть иерархическую структуру всего проекта, как показано на рис. 2.8. В этом представлении можно найти любой класс, доступный в Joomla, и посмотреть всего его поля и методы.

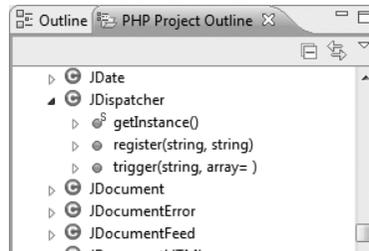


Рис. 2.8. Схематическое представление проекта PHP в Eclipse

Еще одним удобным свойством такой IDE, как Eclipse, является возможность быстро сравнивать две версии исходного файла, чтобы выявить конкретные изменения. Так, на рис. 2.9 представлено сравнение текущей версии файла с предыдущей после внесения самых последних правок.

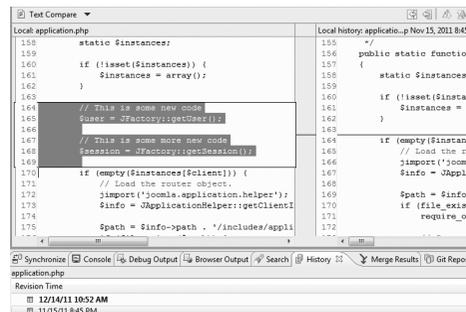


Рис. 2.9. Пример сравнения двух версий исходного файла в Eclipse

Этим свойством можно воспользоваться для сравнения различных версий одного и того же исходного файла. Их можно извлечь из местной предыстории, автоматически сохраняемой в Eclipse, или же из хранилища версии, если применяется система контроля версий.

При написании кода нередко требуется выполнить поиск в целом ряде программ или в проекте в целом, чтобы выяснить, где именно используется конкретная переменная или отдельный фрагмент кода. Для этой цели в Eclipse предоставляется очень простое и эффективное средство. В качестве примера на рис. 2.10 приведен результат поиска в проекте вхождений выражения `JFactory::getSession()`.

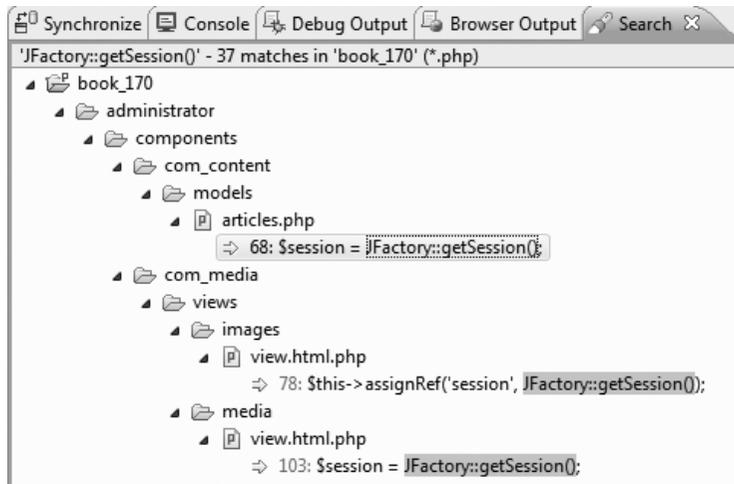


Рис. 2.10. Пример поиска выражения в проекте средствами Eclipse

Ради удобства результаты поиска привязываются к редактору исходного текста. Так, если дважды щелкнуть на одной из строк кода в результатах поиска, то в редакторе исходного текста откроется целевой файл, а курсор установится на номере той строки, где был найден искомый текст. В этом представлении можно выполнять не только поиск, но и замену найденного текста.

Для многих разработчиков очень важной является также возможность отладки программы в процессе ее выполнения. Следует, однако, иметь в виду, что для начинающих программировать отладка может оказаться непростым делом, и к тому же она не является совершенно необходимой при разработке программ для Joomla. Поэтому можете пропустить остальную часть этого подраздела и перейти к чтению следующего материала.

С другой стороны, если вы знакомы в общих чертах с процессом отладки, то сможете относительно легко и быстро освоить доступный в Eclipse отладчик кода PHP. В Eclipse применяется отладчик под названием *Xdebug*, позволяющий приостанавливать выполнение программы на PHP и выполнять ее далее в пошаговом режиме отладки. В качестве примера на рис. 2.11 показана приостановленная для отладки программа, разрабатываемая для Joomla.

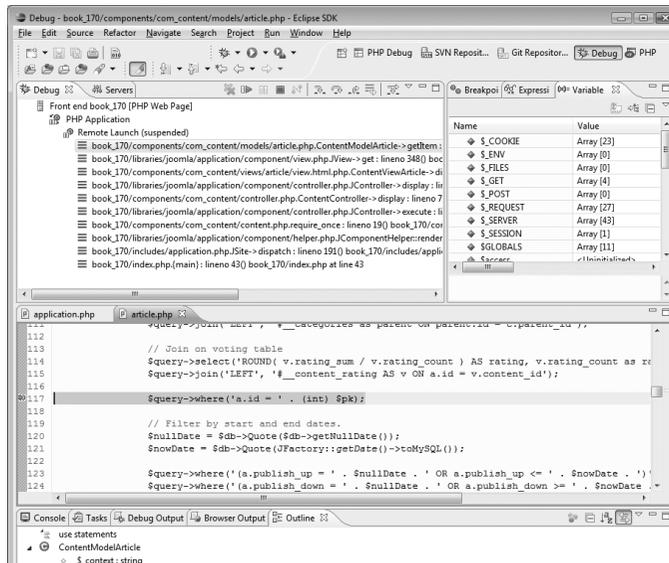


Рис. 2.11. Отладчик, доступный в Eclipse

Слева сверху на этом рисунке показана область, называемая *стеком*. Ее увеличенный вид приведен на рис. 2.12. В стеке показано, каким образом выполнение программы дошло до данной точки. В рассматриваемом здесь примере отладка программы из исходного файла `index.php` была начата со строки кода **43** (последней в стеке). В этой строке вызывается метод `JSite->dispatch`, который, в свою очередь, вызывает метод компонента `JComponentHelper::renderComponent` и так далее до самой вершины стека, где вызывается метод `ContentModelArticle->getItem`.

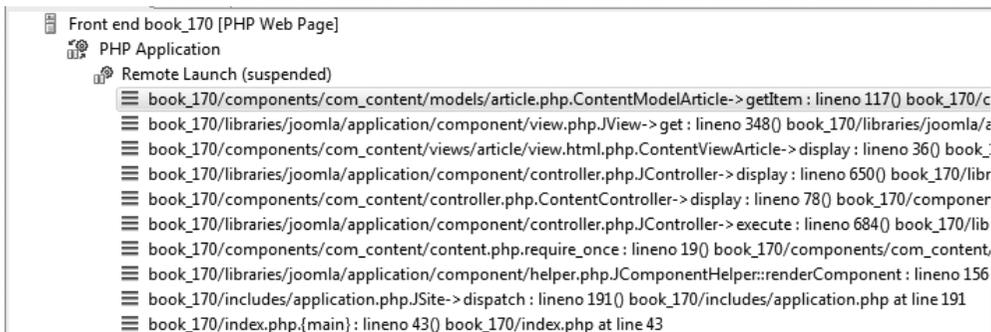


Рис. 2.12. Пример стека из отладчика кода PHP в Eclipse

Важное примечание

Если вы не знаете, что такое стек, — не отчаивайтесь. Принцип его организации и действия подробно рассматривается в главе 3.

В представлении переменных (Variables), приведенном на рис. 2.13, отображаются значения всех переменных из области действия в данной точке выполнения программы.

Name	Value
\$COOKIE	Array [23]
\$ENV	Array [0]
\$FILES	Array [0]
\$GET	Array [4]
\$POST	Array [0]
\$REQUEST	Array [27]
\$SERVER	Array [43]
\$SESSION	Array [1]
\$GLOBALS	Array [11]
\$access	<Uninitialized>
\$archived	<Uninitialized>
\$asset	<Uninitialized>
\$data	<Uninitialized>
\$db	JDatabaseMySQLi
\$e	<Uninitialized>
\$error	<Uninitialized>
\$groups	<Uninitialized>

Рис. 2.13. Представление переменных из отладчика кода PHP в Eclipse

В Eclipse имеется также немало других удобных средств, способствующих более эффективной разработке программ для Joomla. В частности, встроенная справочная система Eclipse устанавливается вместе с этой IDE и предоставляет подробную справку о ее различных функциональных возможностях. К слабым сторонам Eclipse в отношении программирования на PHP можно отнести недостаточное количество учебного материала и пользовательской документации, хотя на сайте документации по Joomla опубликован ряд полезных статей, посвященных Eclipse.

PHPEclipse

PHPEclipse (<http://www.phpeclipse.com>) — это еще одна IDE для программирования на PHP с функциональными возможностями, аналогичными программе PDT для Eclipse. Она поддерживается вне рамок официального проекта семейства Eclipse. На момент написания этой книги активная разработка PHPEclipse, по-видимому, была приостановлена, хотя эта IDE по-прежнему доступна для загрузки.

NetBeans

Другой IDE с открытым кодом, считающейся основной для разработки программ на PHP, является NetBeans. Она свободно доступна для загрузки по адресу <http://netbeans.org/downloads/index.html>. Аналогично Eclipse, IDE NetBeans была первоначально разработана для программирования на Java и теперь поддерживает разработку приложений на других языках программирования.

К главным преимуществам NetBeans относится отличная документация для разработки программ на PHP, включая и учебный видеоматериал. Именно по этой причине NetBeans является отличным выбором для тех, кто только начинает осваивать IDE.

NetBeans обладает такими же функциональными возможностями, как и Eclipse, включая выделение синтаксиса, автоматическое завершение кода, сравнение исходных файлов, глобальный поиск и отладку. Как и в Eclipse, работа в NetBeans организуется по отдельным проектам, что дает возможность без особого труда исследовать проект и находить в нем нужные файлы. В качестве примера на рис. 2.14 приведено представление проекта (Project) в NetBeans.

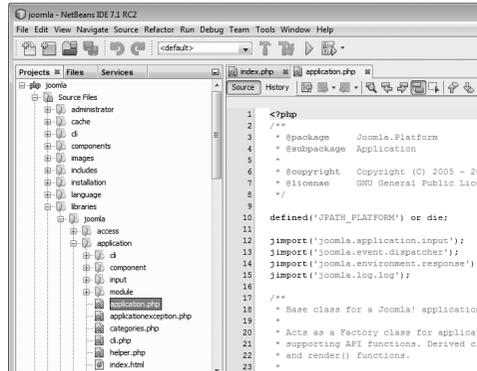


Рис. 2.14. Пример представления проекта в NetBeans

Для разработки программ на PHP в NetBeans поддерживается выделение синтаксиса, выдача сообщений об ошибках и автоматическое завершение кода. В качестве примера на рис. 2.15 показано, каким образом в NetBeans осуществляется выделение синтаксиса, а на рис. 2.16 — сообщается об ошибке в коде PHP. Если навести курсор на маркер ошибки, появятся дополнительные сведения об ошибке, как показано на рис. 2.17.

```
/**
 * Class constructor.
 *
 * @param array $config A configuration array
 *                  session_name, clientId
 *
 * @since 11.1
 */
public function construct($config = array())
{
    jimport('joomla.utilities.utility');
    jimport('joomla.error.profiler');

    // Set the view name.
    $this->_name = $this->getName();

    // Only set the clientId if available.
    if (isset($config['clientId'])) {
        $this->_clientId = $config['clientId'];
    }

    // Enable sessions by default.
    if (!isset($config['session'])) {
        $config['session'] = true;
    }
}
```

Рис. 2.15. Выделение синтаксиса в NetBeans

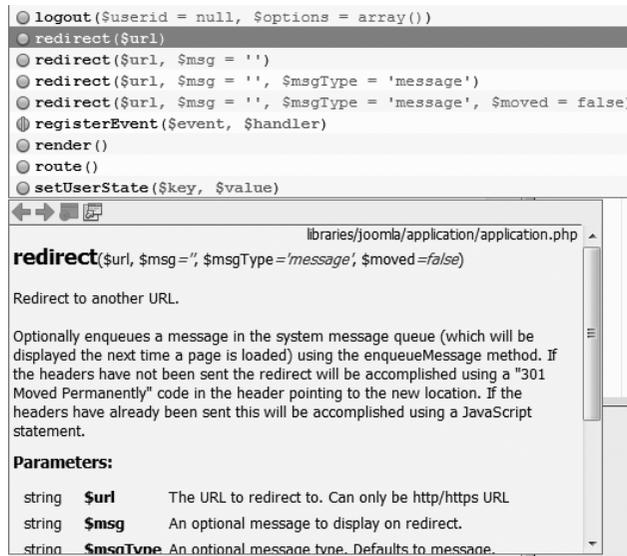


Рис. 2.19. Краткая документация на метод, появляющаяся при автоматическом завершении кода в NetBeans

В схематическом представлении редактируемого класса в NetBeans отображаются методы и поля этого класса. Соответствующий пример приведен на рис. 2.20. В NetBeans поддерживается также сравнение текущей и предыдущей версий исходного файла с использованием предыстории (локальной или контроля версий). Пример сравнения с предыдущей версией исходного файла приведен на рис. 2.21. Кроме того, в NetBeans допускается поиск по проекту в целом. Пример такого поиска приведен на рис. 2.22.

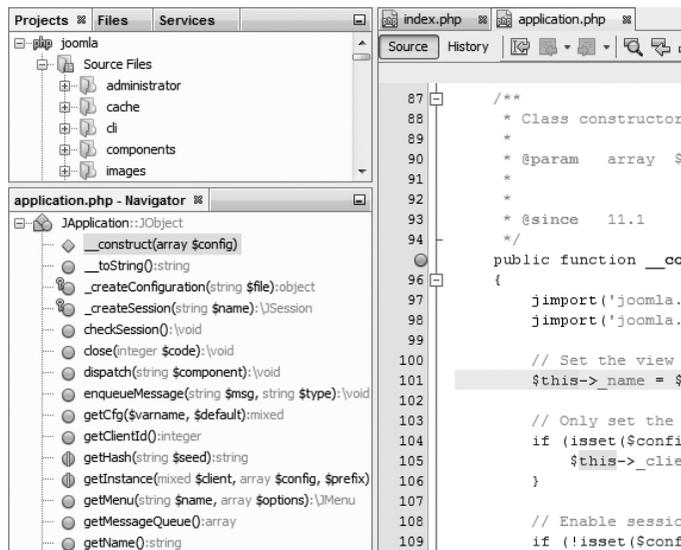


Рис. 2.20. Навигатор NetBeans, отображающий методы редактируемого класса

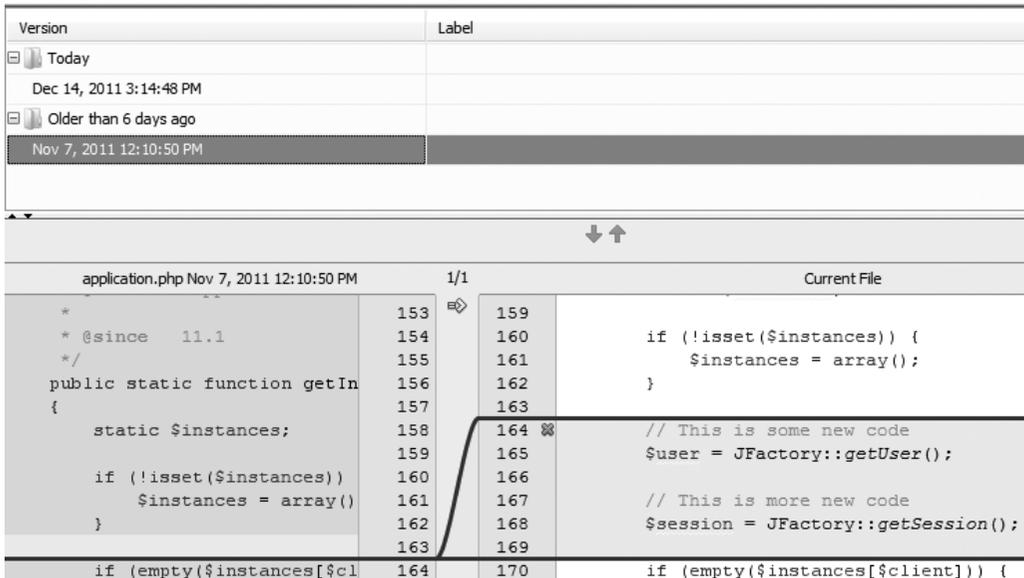


Рис. 2.21. Редактор сравнения версий исходного файла в NetBeans

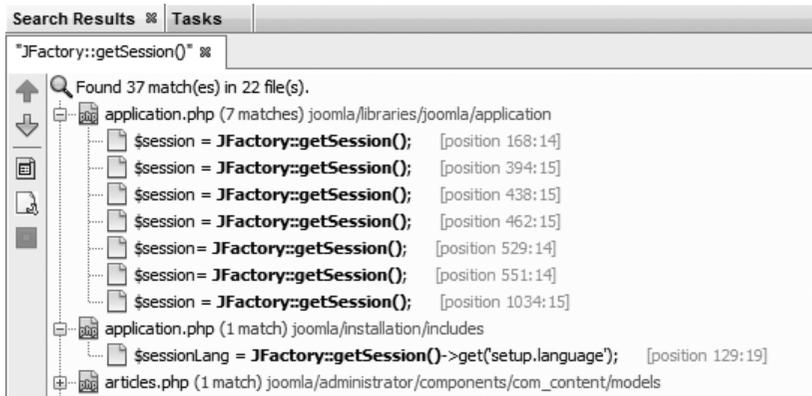


Рис. 2.22. Пример поиска в NetBeans

Как и в Eclipse, результаты поиска в NetBeans привязываются к редактору исходного текста. Так, если дважды щелкнуть на любой строке кода в представлении результатов поиска, соответствующий файл откроется для редактирования, а курсор установится на месте искомого фрагмента кода. В состав NetBeans входит и встроенный отладчик, также называемый Xdebug. В качестве примера на рис. 2.23 приведена программа, приостановленная во время сеанса отладки в NetBeans.



Рис. 2.23. Пример отладки исходного кода программы в NetBeans

В представлении переменных (Variables), доступном в NetBeans, отображаются значения всех переменных из области действия в той точке, где выполнение программы было приостановлено, как показано на рис. 2.24. А в представлении стека вызовов (Call Stack), также доступном в NetBeans, отображается содержимое стека вызовов, как показано на рис. 2.25. Как и в Eclipse, в стеке вызовов перечисляются все вызовы методов, приведшие в данную точку выполнения программы.

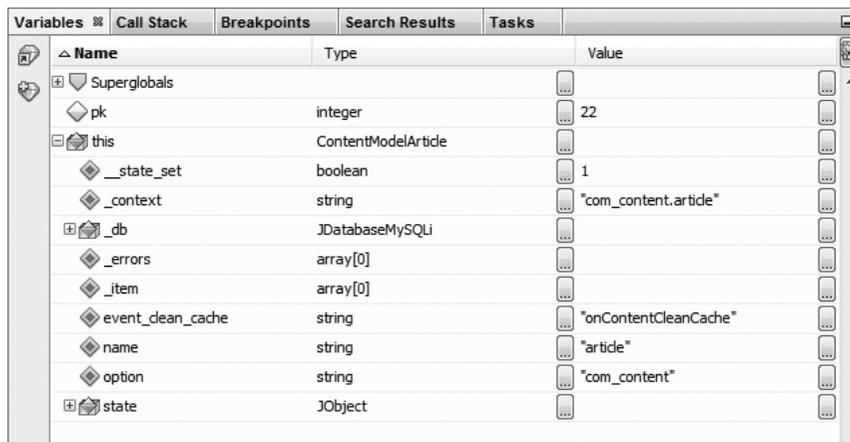


Рис. 2.24. Представление переменных при отладке в NetBeans

Если вы все еще не решились опробовать одну из этих сред, опробуйте обе, а затем выберите из них наиболее подходящую.

Коммерческие IDE

На рынке предлагается немало коммерчески доступных IDE для разработки программ на PHP. Ниже перечислены лишь некоторые из них.

- ZendStudio (<http://www.zend.com/en/products/studio>) — распространенная коммерческая IDE для разработки программ на PHP, основанная на Eclipse.
- Aptana Studio Pro (<http://www.aptana.com>) — другая коммерческая IDE для разработки программ на PHP, настоятельно рекомендуемая некоторыми разработчиками и также основанная на Eclipse.
- Komodo IDE (<http://www.activestate.com/komodo-ide>) — еще одна коммерческая IDE для разработки программ на PHP.
- PhpED от компании Nusphere (<http://www.nusphere.com/products/phped.htm>) — эта среда предоставляет встроенный отладчик и профилировщик, а также функции развертывания готового программного продукта.

Текстовые редакторы

Если материал предыдущего раздела не убедил вас остановить свой выбор на IDE или хотя бы опробовать ее, то в этом разделе рассматриваются более простые инструментальные средства для разработки программ на PHP.

Notepad++

Notepad++ представляет собой простой в использовании, но весьма эффективный открытый редактор исходного текста, предназначенный для Windows. В нем поддерживается выделение синтаксиса и отчасти завершение кода для целого ряда языков программирования, включая и PHP. Этот редактор исходного текста можно свободно загрузить по адресу <http://notepad-plus-plus.org>.

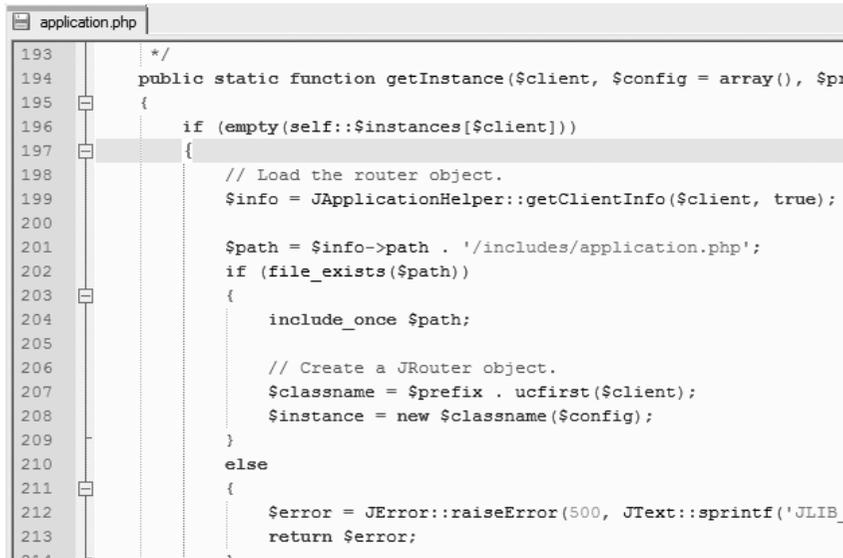
В качестве примера на рис. 2.26 приведен открытый в Notepad++ исходный файл программы на PHP, написанной для Joomla. Как следует из данного примера, в Notepad++ выделение ошибок не поддерживается.

Для пользователей Mac OS X имеется простой и удобный редактор исходного текста Smultron (<http://smultron.en.softonic.com/mac>), обеспечивающий выделение синтаксиса в коде PHP. А для пользователей Linux имеется стандартный системный редактор gedit, также обеспечивающий выделение синтаксиса в коде PHP.

Для написания программ на PHP можно также воспользоваться стандартным текстовым редактором вроде Notepad или vi, но рассчитывать на автоматическое выделение синтаксиса в таком редакторе не приходится. Любой редактор простого текста вполне подойдет для правки всех файлов Joomla. Очевидно, что чем дальше вы отступаете от функциональных возможностей IDE, тем труднее вам будет визуально проверять набранный исходный текст программы на наличие ошибок и опечаток.

Не менее важно понять, что для правки одного и того же набора файлов Joomla можно пользоваться самыми разными программными средствами. Даже если файл находится в

проекте, определенном в IDE, он по-прежнему доступен на уровне файловой системы для редактирования за пределами IDE. Таким образом, для решения различных задач разработки можно при желании пользоваться разными программными средствами.



```
application.php
193  */
194  public static function getInstance($client, $config = array(), $pre
195  {
196      if (empty(self::$instances[$client]))
197      {
198          // Load the router object.
199          $info = JApplicationHelper::getClientInfo($client, true);
200
201          $path = $info->path . '/includes/application.php';
202          if (file_exists($path))
203          {
204              include_once $path;
205
206              // Create a JRouter object.
207              $classname = $prefix . ucfirst($client);
208              $instance = new $classname($config);
209          }
210          else
211          {
212              $error = JError::raiseError(500, JText::sprintf('JLIB_
213              return $error;
214  }
```

Рис. 2.26. Редактор исходного текста Notepad++

Другие инструментальные средства

Для разработки в Joomla! могут пригодиться и другие программные средства. Ниже они рассматриваются вкратце.

Программное обеспечение контроля версий

Это программное обеспечение считается очень важным и просто необходимым для большинства программных проектов, включая и Joomla!. Программное обеспечение контроля версий автоматически отслеживает каждое изменение в кодовой базе проекта, позволяя группе разработчиков вносить изменения в кодовую базу в организованном порядке. Если же изменение, внесенное в код, вызовет какие-нибудь затруднения, оно может быть отменено с целью устранить эти затруднения. Даже если вы разрабатываете проект в одиночку, программное обеспечение контроля версий все равно принесет вам огромную пользу. Обнаружив в проекте какую-нибудь ошибку, вы можете всегда вернуться назад, чтобы выяснить, где именно вкралась ошибка. Это нередко помогает сузить поиск ошибок и недостатков в проекте.

В проектах, разрабатываемых в системе управления контентом (CMS) и на платформе Joomla!, для контроля версий применяется программа Git (<http://git-scm.com>). Подобные проекты размещаются на веб-сайте Github (<https://github.com/joomla>). К числу других распространенных программ контроля версий относятся Subversion (<http://subversion.apache.org>) и Mercurial (<http://mercurial.selenic.com>).

Если вы пользуетесь IDE, то средства контроля версий обычно включаются в саму IDE. Например, в Eclipse и NetBeans поддерживаются программы GIT, Subversion и Mercurial.

Программное обеспечение автоматического построения проектов (Ant и Phing)

Исходные файлы программ на PHP выполняются непосредственно как сценарии. Следовательно, программы на PHP не нужно компилировать, как этого требуют программы на Java или C. Разработчики, программирующие на языках, требующих компиляции или построения готовых программных продуктов, нередко пользуются программами, автоматизирующими эти процессы. К числу таких программ относятся Ant и Phing. Если Ant первоначально разработана для работы с программами на C и Java, то Phing выполняет операции построения проектов на PHP в пакетном режиме.

Если построение программ на PHP не требуется, то зачем вообще нужны такие программные средства, как Ant или Phing? Дело в том, что часто повторяющиеся пакетные задания иногда оказывается удобно автоматизировать. Так, если требуется построить расширение Joomla, то организовать работу можно, например, разделив ее на два проекта в IDE. Один проект будет содержать только код расширения и может быть связан с хранилищем для контроля версий, а другой позволит протестировать расширение в пределах установленных компонентов Joomla.

В данном примере расширение будет тестироваться в проекте тестирования, а изменения в коде — отслеживаться в проекте расширения. Возможно, было бы удобнее сначала изменить код в исходных файлах проекта тестирования, а после проверки на работоспособность скопировать файлы из проекта тестирования в проект расширения. С помощью программы Ant или Phing процесс копирования файлов из одного проекта в другой можно было бы автоматизировать. А кроме того, можно автоматизировать процесс создания архивов для упаковки расширений.

Учебный материал по применению Phing с подобной целью доступен на сайте документации на Joomla по адресу http://docs.joomla.org/Setting_up_your_workstation_for_extension_development.

Программы автоматизации тестирования

В разработке программного обеспечения наметилась устойчивая тенденция к автоматизации процесса тестирования программ. Автоматические тесты могут быть написаны один раз, а затем выполнены неоднократно для обнаружения ошибок, вкравшихся в кодую базу при внесении в нее изменений.

В проекте Joomla в настоящее время применяются две разновидности автоматического тестирования: блочное и системное. Блочное тестирование служит для тестирования платформы Joomla, а для выполнения блочных тестов — программа на PHP под названием PHPUnit. Это дает возможность писать тесты, непосредственно снабжающие данными метод в коде PHP, и сравнивать полученные результаты с предполагаемыми. Если они отличаются, выдается сообщение об ошибке.

Системное тестирование служит для проверки работоспособности системы CMS с точки зрения пользователя. С этой целью система CMS выполняется в сеансе работы

с браузером под управлением тестовой программы. А в браузере выполняется последовательность команд ввода данных, включая операции с мышью и ввода с клавиатуры, имитирующие действия пользователя тестируемого программного обеспечения. Документ в окне браузера может быть автоматически проверен в системном тесте на предмет появления некоторой информации на веб-странице. Если эта информация не появляется на веб-странице, выдается сообщение об ошибке. Для создания и выполнения системных тестов в Joomla применяется программа Selenium.

Выполнение подобных автоматических тестов является важной стадией процесса разработки в Joomla. При внесении изменений в виде устраняемых программных ошибок или усовершенствований кода автоматические тесты могут быть выполнены с целью упростить проверку каких-либо нарушений, вызванных внесенными изменениями.

Резюме

В этой главе рассмотрены вопросы подготовки рабочей станции к написанию программ для Joomla. Сначала в ней обсуждались варианты установки программного обеспечения веб-сервера на разных платформах, включая отдельные пакеты для различных операционных систем. Затем были вкратце описаны две IDE, наиболее распространенные для программирования на PHP: Eclipse и NetBeans, а также представлены примеры, показывающие, насколько они позволяют повысить производительность труда разработчиков прикладных программ для Joomla.

Затем были вкратце рассмотрены разные варианты редакторов исходного текста для тех, кто не желает пользоваться IDE. И в конце главы было кратко описано программное обеспечение контроля версий (вроде Subversion и Mercurial), автоматического построения проектов (типа Phing и Ant), а также автоматического тестирования.

Итак, рабочую станцию можно считать подготовленной и настроенной, и теперь можно приступить непосредственно к написанию прикладных программ для Joomla.

