

CONTENTS INCLUDE:

- Introduction
- Accessing Exposed Web Services
- Discovering Exposed Services from PHP
- Generating Documents
- Manipulating PDF Files
- Manipulating Images and more...

Adobe® ColdFusion Web Services for PHP Programmers

By Marco Tabini

INTRODUCTION

Adobe® ColdFusion 9 (CF9) is a technology platform that allows developers to create web-based applications of any complexity using a set of easy-to-use tools.

In addition to being a standalone server-side technology, ColdFusion also provides a number of facilities that allow you to perform tasks like manipulating PDF or Microsoft Office files, sending e-mail, creating charts, and much more—all tasks that often require expensive dedicated solutions that must be developed in-house.

Because ColdFusion services can be deployed using a variety of standard protocols like SOAP or plain XML, you can easily take advantage of them from your PHP 5 scripts using nothing more than the standard web service facilities of the language. In fact, you can take advantage of CF9's "Exposed Services" without even writing a single line of CFML code!

Obtaining and Installing CF9

Much like PHP, ColdFusion is a cross-platform product that is capable of running on a variety of platforms, including Windows, Mac OS X and any variety of Linux on which you can install a Java runtime environment.

CF9 comes in two different editions: Standard and Enterprise. The latter provides a number of scalability and monitoring facilities designed for large deployments, while the former is targeted at simpler needs. Outside of these differences, however, both editions offer a comparable set of core features, which means that your CF9 server setup can grow organically without having to sacrifice any functionality—which CF9 edition you choose will be primarily dictated by the amount of traffic that you need to serve, rather than by the type of task that you need to perform.

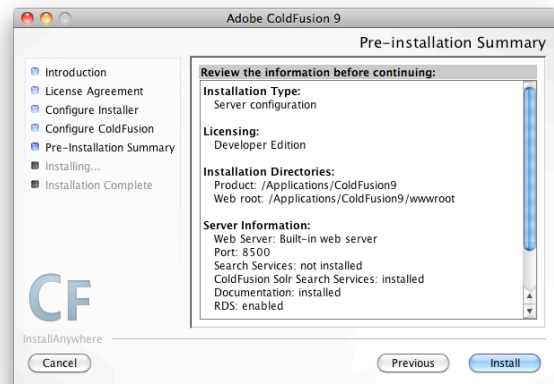


You can find a feature-by-feature comparison of each CF9 edition at http://www.adobe.com/products/coldfusion/pdfs/cf9_feature_comparison_matrix_ue.pdf

While CF9 is a commercial product that requires the purchase of a license in order to be used in a production environment, Adobe makes a Developer Edition of the platform available free of charge. Its only limitation is that it allows no more than two concurrent connections—perfect for developing locally on your machine.

You can download the Developer Edition of CF9 directly from the Adobe website at <http://www.adobe.com/products/coldfusion/>; like the commercial versions of CF9, the Developer Edition is available for every platform supported by the product. Once

you have downloaded and launched the installer, the setup process will ask you a number of questions, including whether you want to install OpenOffice, which CF9 uses to manipulate Microsoft Office documents, and whether to enable the Remote Development Service, which you should do in a development environment if you plan on taking advantage of CF9's debugging features.



At the end of the setup process, the installer will launch the Configuration and Settings Wizard, a web-based application that completes your server's configuration and eventually grants you access to the ColdFusion Administrator, from which you can control and manipulate all your server-side settings.

ACCESSING EXPOSED WEB SERVICES

You don't need to know how to develop ColdFusion applications in order to take advantage of the Exposed Services that are part of the platform.

ADOBE® COLD FUSION® 9



Less code. Less time.
More efficient. More features.
See why less is more.



ADOBE® FLASH® PLATFORM

Try it today.
www.adobe.com/go/trycoldfusion

In fact, CF9 makes a number of its features available directly through a SOAP-based interface, allowing you to:

- Generate and manipulate PDF files
- Manipulate images
- Generate Microsoft Office documents
- Send and receive e-mail

Access Control

In order to access any of the exposed CF9 services, you need to first define the appropriate set of access controls. You can do so by creating a user by opening the Administrator and clicking on the "User Manager" item under the "Security" menu. Click on the "Add User" button, and then fill the user authentication form with the appropriate information.

Add and manage users. Users can be granted access to roles and sandboxes

User Authentication (Add Mode)

*User name: (Minimum 5 characters)

*Password: (Minimum 5 characters)

*Confirm Password:

Description:

At this point, you can define which Exposed Services the user has access to by selecting them in the "Prohibited Services" list and transferring them to the "Allowed Services" list.

Exposed Services

Allowed Services:

Prohibited Services:

>> <<

To finish, click on the "Add User" button and you're all set!

Enabling Client Access

By default, to ensure the security of your server, CF9 precludes any client from accessing Exposed Services.

You can change this setting by clicking on the "Allowed IP Addresses" item under the "Security Menu," where you can add one or more IPs (or IP ranges) to the server's whitelist for Exposed Services.

Security > Allowed IP Addresses

Specify client IP addresses that should be allowed to invoke exposed services. This can be individual IP addresses, IP address ranges of the form 10-30, or * wild cards. Both IPv4 and IPv6 addresses are supported. To include an IP address in the list, enter the address and click Add. To delete an IP address from the list, select the address and click Remove Selected. When no IP addresses are selected, no user can invoke exposed services

Select Allowed IP Addresses for Exposed Services

IP Address

Add

View / Remove Selected IP Addresses for Exposed Services

Remove Selected

Hot Tip

The examples in this Refcard assume that you are running the Developer Edition of CF9 on your local machine. For deployment, you will need to alter your settings according to the make-up of your specific production environment.

Configuring Services

Some services must be configured before they can be used. For example, the Mail service requires you to specify an SMTP server before it can be used to send messages, while the Document service requires you to specify the location of OpenOffice, and so on.

In all cases, you can view and modify the settings for the services exposed by your server directly from the Administrator.

DISCOVERING EXPOSED SERVICES FROM PHP

Exposed Services are made available through a specific location on your server; typically, if you have installed your copy of CF9 using the standard settings, this will be at <http://127.0.0.1:8500/CFIDE/services/>. If you point your browser to that URL, you will be greeted with a list of all the service entry points available.

Clicking on a specific file will cause CF9 to output a help page that describes all the functionality provided by the service and all the available remote method calls.

CFIDE.services.mail Component mail

hierarchy:	WEB-INF.cftags.component CFIDE.services_base CFIDE.services.mail
path:	/Applications/ColdFusion9/wwwroot/CFIDE/services/mail.cfc
serializable:	Yes
properties:	
methods:	send
inherited methods:	convertArrayToMap , convertStructToMap , convertURLtoPath , getHttpUrl , getTempFilePath , isAllowed , isAllowedIP , readFileFromURL

* - private method

```
send
remote void send ( string serviceusername, string servicepassword, string server, string port, string username, string password, string from, string to, string bcc, string cc, string subject, string content, string type, string charset, string failto, string mailerid, string mimeattach, string priority, string replyto, string timeout, string useSSL, string useTLS, string wraptext, mailparam[] attachments, mailpart[] mailparts )
```

Hot Tip

Complete documentation for all the Exposed Services can be found on the Adobe Help site at http://help.adobe.com/en_US/ColdFusion/9.0/CFMLRef/index.html both in a "live" web-based version and in PDF format.

Accessing a Service's WSDL Specification

Since your PHP scripts will be accessing Exposed Services through SOAP, you need access to the WSDL specification for the services you want to interface with.

Luckily, CF9 supports the built-in generation of WSDL documents, so all you need to do is add a special query string to the end of the URL of its entry point. For example, the WSDL for the Mail service can be accessed at this URL:

<http://127.0.0.1:8500/CFIDE/services/mail.cfc?WSDL>

Using SOAP

PHP 5's web service libraries make accessing web services of any kind very easy.

Once you have access to the WSDL, you can call any of the services directly from a PHP 5.x script by using the SOAPClient class. SOAPClient encapsulates a SOAP service and provides the means for introspecting its methods and data types, as well as performing calls against it.

For example, having obtained the URL of the WSDL document for the Mail service, you can find out which method calls it supports by running this simple script:

```
$service = new SOAPClient('http://127.0.0.1:8500/CFIDE/services/mail.cfc?WSDL');
var_dump($service->__getFunctions());
```

This will produce the following output:

```
array(1) {
  [0] =>
  string(35) "sendResponse send(send $parameters)"
}
```

This indicates that the service supports a method called send() that takes a single parameter of type "send" in input and returns a value of type "sendResponse" in output.

Much like the __getFunctions() method of SOAPClient can be used to retrieve the names of all the remote procedure calls provided by a service, __getTypes() can be used to retrieve a description of all the compound data types defined by it. In this case, the output will contain a description of both the "send" and "sendResponse" data types, which we can then use to issue a call to the service:

```
<?php
$service = new SOAPClient('http://127.0.0.1:8500/CFIDE/services/mail.cfc?WSDL');

$payload = array(
    'serviceusername' => 'johndoe',
    'servicepassword' => 'johndoepassword',
    'server' => 'smtp.example.com',
    'port' => '25',
    'username' => 'smtuser',
    'password' => 'smtpassword',
    'from' => 'marco@example.com',
    'to' => 'marco@example.com',
    'bcc' => '',
    'cc' => '',
    'subject' => 'Test',
    'content' => 'This is a test e-mail!',
    'type' => '',
    'charset' => '',
    'failto' => '',
    'mailerid' => '',
    'mimeattach' => '',
    'priority' => '',
    'replyto' => '',
    'timeout' => '',
    'useSSL' => '',
    'useTLS' => '',
    'wraptext' => '',
    'attachments' => '',
    'mailparts' => ''
);
```

As you can see, you can leave those fields that are not required blank by setting them to an empty string; however, you must include them in the payload, and they cannot be set to NULL.

If you use an IDE to edit your PHP code, you may be able to take advantage of advanced SOAP functionality such as code completion or service proxying. Check your IDE's documentation for more information.

You have probably noticed the serviceusername and servicepassword properties at the beginning of the payload: these are the credentials of the user that you defined earlier in the Administrator.

Dealing with Failure

Occasionally, any code can fail. It is, therefore, essential to be able to manage failure in a predictable and safe way.

When a service call fails, SOAPClient throws an exception of the appropriate kind, which, generally, includes some information about the error and provides useful clues on how to address it. For example:

```
PHP Fatal error: Uncaught SoapFault exception: [soapenv:Server.userException] coldfusion.xml.rpc.CFCInvocationException: [coldfusion.service.layer.ServiceLayerExceptions$UserNotAuthorizedException : User johndoe is not authorized to invoke mail service]
Stack trace:
[...]
```

In this particular case, you can plainly see that the failure is caused by the fact that the password provided for the service user was incorrect.

In a production environment, you should always encapsulate your service calls in a try... catch code block and handle failures appropriately.

GENERATING DOCUMENTS

The Document Exposed Service can be used to generate documents based on existing templates, or even on the contents of a web page.

You can find the WSDL for the Document service at this URL: <http://127.0.0.1:8500/CFIDE/services/document.cfc?wsdl>

As you can see by introspecting the WSDL descriptor or reading the documentation for the service, Document supports a method called generate() that requires a long list of parameters. However, you only need a couple of them to create a PDF out of a web page:

```
<?php
$service = new SOAPClient('http://127.0.0.1:8500/CFIDE/services/document.cfc?wsdl');

$payload = array(
    'serviceusername' => 'johndoe',
    'servicepassword' => 'johndoepassword',
    'format' => 'pdf',
    'content' => '',
    'authpassword' => '',
    'authuser' => '',
    'backgroundvisible' => '',
    'bookmark' => '',
    'encryption' => '',
    'fontembed' => '',
    'marginbottom' => '',
    'marginleft' => '',
    'marginright' => '',
    'margintop' => '',
    'mimetype' => '',
    'orientation' => '',
    'ownerpassword' => '',
    'pageheight' => '',
    'pagetype' => '',
    'pagewidth' => '',
    'permissions' => '',
    'proxyhost' => '',
    'proxypassword' => '',
    'proxyport' => '',
    'proxyuser' => '',
    'src' => 'http://google.com',
    'scale' => '',
    'unit' => '',
    'useragent' => '',
    'userpassword' => '',
    'documentsection' => '',
    'documentitem' => ''
);

$result = $service->generate($payload);
$pdfUrl = $result->generateReturn;
```

As you can see, the service returns a structure that contains the URL to the PDF file that was generated from the URL you provided in input. You can either redirect your client to that URL, or read the file and output it directly from your PHP file if the CF9 server is not directly reachable to the client that invoked your PHP script.

The Trial and Developer edition of CF9 will place a watermark on all generated files. Licensed versions of CF9 do not have this limitation.

While the example above is a bit contrived, there are many practical scenarios in which being able to create PDF files can be useful to a web project—think, for example, of e-commerce systems that must deliver receipts to their customers.

MANIPULATING PDF FILES

The PDF service, as its name implies, makes it possible for you to manipulate PDF files. You can perform tasks like add watermarks, add, delete and extract pages from a document, merge documents together and modify their security features. The PDF service can be accessed through this WSDL file:

<http://127.0.0.1:8500/CFIDE/services/pdf.cfc?WSDL>

Retrieving Information About a PDF File

The `getinfo()` method of the service allows you to retrieve a wealth of information from a PDF file—from the name of the application that was used to create it to the number of pages it contains:

```
<?php
$service = new SOAPClient('http://127.0.0.1:8500/CFIDE/services/pdf.cfc?wsdl');

// var_dump($service->__getTypes());

$payload = array(
    'serviceusername' => 'johndoe',
    'servicepassword' => 'johndoepassword',
    'source' => 'http://example.com/example.pdf',
    'password' => '',
);

$result = $service->getinfo($payload);

echo $result->application;
echo $result->totalPages;
```

As you can see, the service expects you to pass a URL to the PDF file that needs to be inspected—clearly, this location must be accessible to your CF9 installation.

You can examine and manipulate protected PDF files by providing a value for the password parameter of most PDF service calls. Note that using the document's user password will limit you to the functionality that has been enabled by the creator. For more information on the PDF security model, visit http://createpdf.adobe.com/cgi-feeder.pl/help_security

Generating Thumbnail Views

Another useful feature of the PDF service is the ability to generate thumbnail images of a document's pages:

```
<?php
$service = new SOAPClient('http://127.0.0.1:8500/CFIDE/services/pdf.cfc?wsdl');

$payload = array(
    'serviceusername' => 'johndoe',
    'servicepassword' => 'johndoepassword',
    'source' => 'http://example.com/example.pdf',
    'format' => 'jpg',
    'imageprefix' => 'img',
    'password' => '',
    'pages' => '',
    'resolution' => 'high',
    'scale' => '',
    'transparent' => '',
);

$result = $service->thumbnail($payload);

foreach ($result->thumbnailReturn as $thumbnail) {
    echo $thumbnail->value;
}
```

Upon a successful execution, this method returns a collection of objects that provide URLs for each of the thumbnails generated. As you can see, you can specify a number of different options: the output format (JPEG in this case, but PNG and TIFF are also supported), the resolution (high or low), the scale (a percentage between 1 and 100), and so on.

Password-protect and Encrypt PDF Documents

It is sometimes convenient to be able to dynamically password-protect a PDF document—for example to provide each user with a customized version that can only be opened using his or her password.

This can be easily accomplished by calling the `protect()` method:

```
<?php
$service = new SOAPClient('http://127.0.0.1:8500/CFIDE/services/pdf.cfc?wsdl');

$payload = array(
    'serviceusername' => 'johndoe',
    'servicepassword' => 'johndoepassword',
    'source' => 'http://example.com/example.pdf',
    'newUserPassword' => 'userpass',
    'newOwnerPassword' => 'ownerpass',
    'permissions' => 'AllowPrinting,AllowScreenReaders',
    'encrypt' => 'RC4_128M',
    'password' => '',
);

$result = $service->protect($payload);

echo $result->protectReturn;
```

The result of the call will contain the URL to the newly protected document in a property called `protectedReturn`.

Note that the service call allows you to specify both an owner and client password (http://createpdf.adobe.com/cgi-feeder.pl/help_security for more information on the Adobe PDF security model) as well as specific permissions that users are allowed to take advantage of. All these options are explained in the CF9 documentation.

Merging PDF Files

The ability to merge one or more PDF files can be handy in a number of different scenarios. Imagine, for example, compiling several reports into a single document, or dynamically creating PDF books for your customers based on specific chapters that they decide to buy.

CF9 makes this task very easy to accomplish:

```
<?php
error_reporting(E_ALL | E_STRICT);

$service = new SOAPClient('http://127.0.0.1:8500/CFIDE/services/pdf.cfc?wsdl');

$payload = array(
    'serviceusername' => 'johndoe',
    'servicepassword' => 'johndoepassword',
    'source' => 'http://example.com/example.pdf, http://example.com/example2.pdf',
    'keepbookmark' => '1'
);

$result = $service->mergeFiles($payload);

echo $result->mergeFilesReturn;
```

The merge operation can be performed on an arbitrary number of files (greater than or equal to two). The `keepbookmark` parameter allows you to maintain all the existing bookmarks from your source PDFs into the final result.

MANIPULATING IMAGES

While PHP has some powerful image manipulation functions provided by the GD Library, the latter does have some

limitations. It is sometimes difficult to install, and the process of creating and manipulating the images can require multiple complex steps.

If you happen to also have access to an instance of CF9, however, you can take advantage of its Image services for a set of easy-to-use manipulation functions that, while not as rich as the ones provided by GD, require much less effort, especially for simple tasks.

The Image service's WSDL can be accessed at this URL:

<http://127.0.0.1:8500/CFIDE/services/image.cfc?WSDL>

Obtaining Information About an Image

For example, you can obtain a wealth of information on an image through a simple call to the Info method of the Image service:

```
<?php
$service = new SOAPClient('http://127.0.0.1:8500/CFIDE/services/image.cfc?wsdl');

$payload = array(
    'serviceusername' => 'johndoe',
    'servicepassword' => 'johndoepassword',
    'source' => 'http://example.com/logo.png',
);

$result = $service->info($payload);

foreach ($result->InfoReturn as $info) {
    echo $info->key, ' => ', $info->value, "\n";
}
```

The result of the method call is a long list of data on the image:

```
source => cf5341945594648670180.png
height => 126
width => 103
colormodel.pixel_size => 32
colormodel.bits_component_1 => 8
colormodel.bits_component_2 => 8
colormodel.alpha_channel_support => YES
colormodel.colormodel_type => ComponentColorModel
colormodel.colorspace => Any of the family of RGB color spaces
colormodel.alpha_premultiplied => NO
colormodel.transparency => TRANSLUCENT
colormodel.num_color_components => 3
colormodel.bits_component_3 => 8
colormodel.num_components => 4
colormodel.bits_component_4 => 8
```

As you can see, we can now determine the width and height of the image, as well as gather lots of information about its color model.

Resizing an Image

Another typical operation that is performed on images is resizing them; in this case, the functionality provided by CF9 is much simpler than its GD equivalent:

```
<?php
$service = new SOAPClient('http://127.0.0.1:8500/CFIDE/services/image.cfc?wsdl');

$payload = array(
    'serviceusername' => 'johndoe',
    'servicepassword' => 'johndoepassword',
    'source' => 'http://example.com/logo.png',
    'width' => 100,
    'height' => 100,
    'interpolation' => 'highestquality',
    'blurfactor' => 1,
);

$result = $service->resize($payload)->ResizeReturn;
```

As you can see, the process is fairly simple. All you need to do is specify a location for the image, its new dimensions, an interpolation algorithm and a blur factor; the latter is used to reduce resizing artifacts when the final image has a larger size than the original.

Compared to the GD method of resizing images, the CF9 approach has several advantages. First, it's a one-step

operation, whereas GD requires multiple separate steps. Second, CF9 takes care of caching the output image for you, thus making caching repeated resize operations easier.

CHARTING

While there are numerous open source libraries and free web services that can be used to generate charts, CF9 provides a truly powerful service whose functionality is hard to beat.

For example, in addition to creating charts in a multitude of different varieties, the Chart service allows you to output in several formats, including as an interactive file in the form of an Adobe® Flash® movie that features mouse-over effects and animations.

The WSDL for the Chart service can be found at this URL:

<http://127.0.0.1:8500/CFIDE/services/chart.cfc?WSDL>

Generating Charts

The functionality provided by the Chart service is rich and complex, but can all be accessed through a single call to the generate() method. As a result, the latter has a signature that looks a little intimidating at first, but that is, in reality, very easy to grasp once you get through the basics. Here's an example:

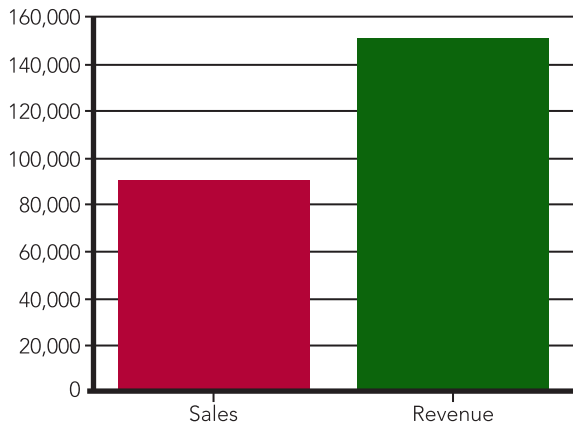
```
<?php
$service = new SOAPClient('http://127.0.0.1:8500/CFIDE/services/chart.cfc?wsdl');

$payload = array(
    'serviceusername' => 'johndoe',
    'servicepassword' => 'johndoepassword',
    'backgroundcolor' => '',
    'chartheight' => '',
    'chartwidth' => '',
    'databackgroundcolor' => '',
    'font' => '',
    'fontbold' => '',
    'fontitalic' => '',
    'fontsize' => '',
    'foregroundcolor' => '',
    'format' => 'flash',
    'gridlines' => '',
    'labelformat' => '',
    'markersize' => '',
    'pieslicestyle' => '',
    'scalefrom' => '',
    'scaleto' => '',
    'seriesplacement' => '',
    'show3d' => '',
    'showborder' => '',
    'showlegend' => '',
    'showmarkers' => '',
    'showxgridlines' => '',
    'showygridlines' => '',
    'sortxaxis' => '',
    'tipbgcolor' => '',
    'tipstyle' => '',
    'title' => '',
    'url' => '',
    'xaxistitle' => '',
    'xaxistype' => '',
    'xoffset' => '',
    'yaxistitle' => '',
    'yaxistype' => '',
    'yoffset' => '',
    'chartseries' => array (
        'chartdata' => array (
            0 => array (
                'item' => 'Sales',
                'value' => '90000'
            ),
            1 => array (
                'item' => 'Revenue',
                'value' => '150000'
            )
        ),
        'colorlist' => 'red,green',
        'datalabelstyle' => '',
        'paintstyle' => '',
        'seriescolor' => '',
        'serieslabel' => '',
        'type' => 'bar',
    )
);

$result = $service->generate($payload)->generateReturn;
```

As you can see, despite the long list of parameters, all you really need to provide is one or more "chartseries" values,

which are nothing more than a collection of data points in item/value format and a few specs on how those data points should be represented. The remaining parameters can be used to further tweak the chart's appearance and manipulate items like its title, font characteristics, the visibility of grid-lines, and so on.



This last example barely scratches the surface of CF9's charting service has to offer. For example, you can actually create 3-D charts, or combine multiple graphs into a single image. Depending on the complexity of your project, CF9 can greatly enhance the look of your application, and provide your users with helpful visuals.

LEARN MORE ABOUT CF9 WEB SERVICES

There is a lot more to CF9's Web Services offerings than what can be covered in this Refcard — for example, there is an entire service dedicated to e-mail. Luckily, Adobe maintains a comprehensive set of documentation for developers at http://help.adobe.com/en_US/ColdFusion/9.0/Developing/index.html.

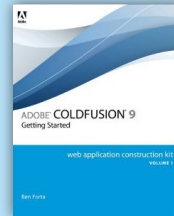
ABOUT THE AUTHOR



Marco Tabini is the co-founder, with Arbi Arzoumani, of php|architect, the world's largest PHP magazine in the English language, currently distributed in over 145 countries. He is also the co-founder, with Keith Casey, of Blue Parabola, LLC, a consulting firm that specializes in information architecture, code and security auditing, large-scale deployments and optimization.

An accomplished author on the subject of PHP and the business of web development, Marco is also a frequent speaker at PHP and OSS conferences throughout the world.

RECOMMENDED BOOKS



Written by the best known and most trusted name in the ColdFusion community, Ben Forta, The ColdFusion Web Application Construction Kit is the best-selling ColdFusion series of all time - the books that most ColdFusion developers used to learn the product. This Getting Started volume starts with Web and Internet fundamentals and database concepts and design, and then progresses to topics including creating data-driven pages, building complete applications, implementing security mechanisms, integrating with e-mail, building reusable functions and components, generating data-driven reports and graphs, building Ajax-powered user interfaces, and much more. Complete coverage of ColdFusion 9 continues in "Volume 2, Application Development" (ISBN:0321679199) and "Volume 3, Advanced Application Development" (ISBN:0321679202).

BUY NOW

books.dzone.com/books/adobe-coldfusion-9

Browse our collection of over 100 Free Cheat Sheets



Free PDF

Upcoming Refcardz
 Network Security
 Hadoop
 Solr
 Subversion



DZone communities deliver over 6 million pages each month to more than 3.3 million software developers, architects and decision makers. DZone offers something for everyone, including news, tutorials, cheatsheets, blogs, feature articles, source code and more. **"DZone is a developer's dream,"** says PC Magazine.

DZone, Inc.
 140 Preston Executive Dr.
 Suite 100
 Cary, NC 27513
 888.678.0399
 919.678.0300

Refcardz Feedback Welcome
refcardz@dzone.com
Sponsorship Opportunities
sales@dzone.com

