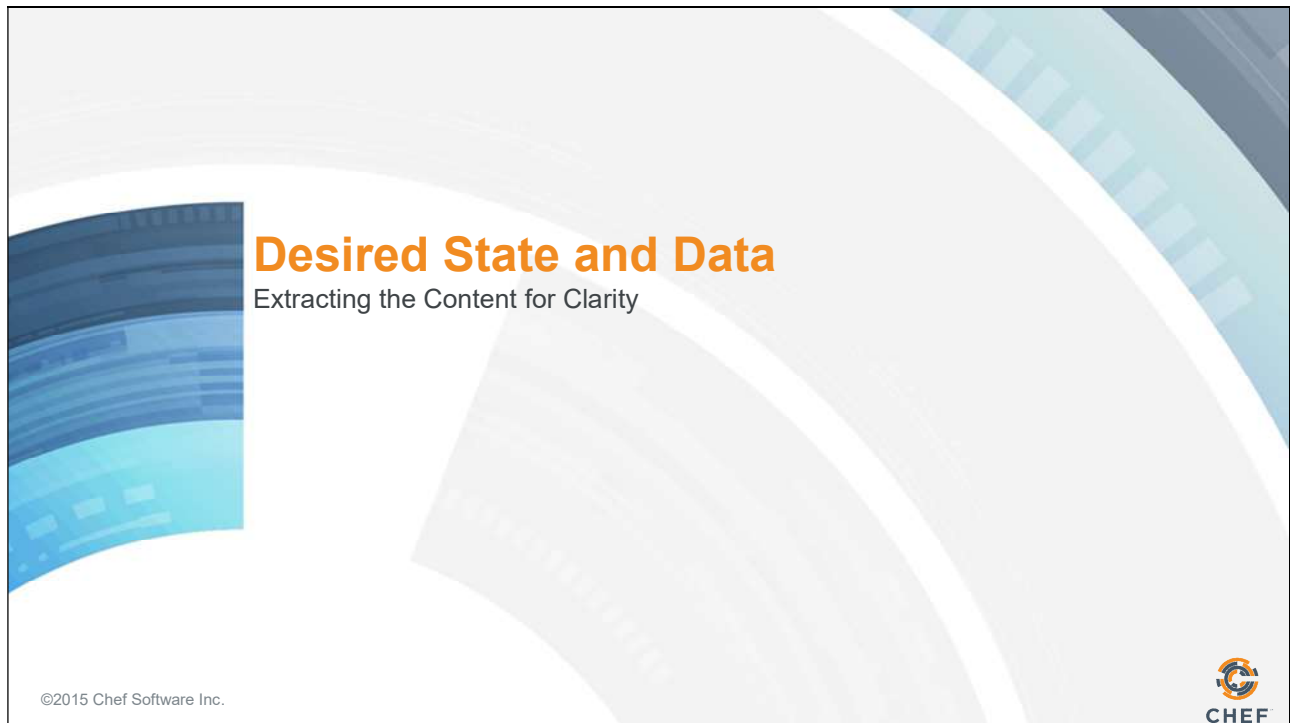


7: Desired State and Data



Slide 2

Objectives




After completing this module, you should be able to

- Explain when to use a template resource
- Create a template file
- Use ERB tags to display node data in a template
- Define a template resource

In this module you will learn how to understand when to use a template resource, create a template file, use ERB tags to display node data in a template, define a template resource.


Slide 3



Cleaner Recipes

In the last section we updated our webserver cookbook to display information about our node.

We added this content to the file resource in the server recipe.

©2015 Chef Software Inc. 7-3 

We were successful in displaying the details about the system instead of using hard-coded values. We added that content to the content attribute of the file resource that generates the Default page for the webserver.

Slide 4

Demo: The 'iis-demo' Server Recipe

```
~/cookbooks/iis-demo/recipes/server.rb
```

```
powershell_script "Install IIS" do
  code "add-windowsfeature Web-Server"
end

file 'c:\inetpub\wwwroot\Default.htm'
  content "<h1>Hello, world!</h1>
<h2>ipaddress: #{node['ipaddress']}</h2>
<h2>hostname: #{node['hostname']}</h2>
<h2>total memory: #{node['memory']['total']}</h2>
<h2>CPU Mhz: #{node['cpu']['0']['mhz']}</h2>"
end

service "w3svc" do
  action [ :enable, :start ]
end
```

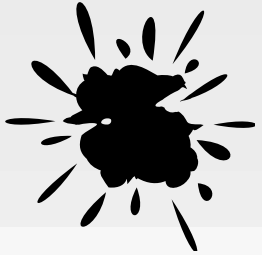
What if new changes are given to us for the website splash page?

For each new addition we would need to return to this recipe and carefully paste the contents of the new HTML into the string value of the content attribute.

Let's talk about some ways in which this method of keeping the content within the recipe will be problematic as we make more changes.

Slide 5


Double Quotes Close Double Quotes



Double quoted strings are terminated by double quotes.

```
"<h1 style="color: red;">Hello, World!</h1>"
```

The diagram shows the string `"<h1 style="color: red;">Hello, World!</h1>"`. Two red boxes highlight the opening double quote at the start and the closing double quote at the end. Two blue arrows point upwards from the bottom of the slide to these two quotes, illustrating that the string is terminated by double quotes.

©2015 Chef Software Inc. 7-5 

There are some things that you need to be careful of when working with double-quoted strings in Ruby: double-quoted strings are terminated by double-quotes.

If any of the text that we paste into this content field has double quotes it is going to have to be escaped. This can possibly become a problem if we wanted to add some additional style or design to our splash page.

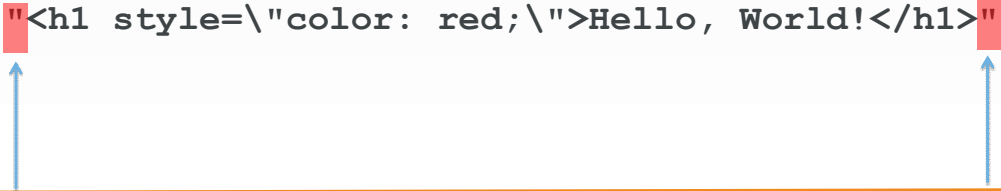
Slide 6

CONCEPT

Backslash


We can use double-quotes as long as we prefix them with a backslash.

```
"<h1 style=\"color: red;\">Hello, World!</h1>"
```



©2015 Chef Software Inc.

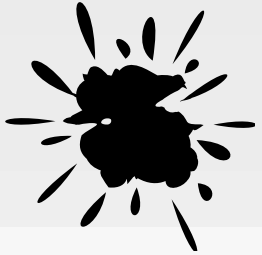
7-6



With Ruby strings you can use the backslash character as an escape character. In this case, if you wanted to have a double-quote inside a double-quoted string, you would need to place a backslash before the double-quote.

Here is the fixed version where all the double-quotes within the string are escaped to ensure that Ruby will read this as a single string of characters.


Slide 7




Backslash

Backslashes are reserved characters. So to use them you need to use a backslash.

```
"Root Path: C:\"
```



©2015 Chef Software Inc. 7-7 

That also brings up an issue with the backslash character. This is particularly important on Windows where the file separator is often represented with a backslash character. You will need to keep an eye out for backslash characters because backslash characters are now the escape character in a double-quoted string.

If you want to literally represent a backslash you'll need to use two-backslashes.

Slide 8

CONCEPT

Backslash

Backslashes are reserved characters. So to use them you need to use a backslash.

```
"Root Path: C:\\"
```



©2015 Chef Software Inc.7-8

So every time text you use a double-quoted string, you will need to find and replace all backslashes with double-backslashes and then replace all double-quotes with backslash double-quotes.

Slide 9

Unexpected Formatting

```
file '/etc/motd' do
  content 'This is the first line of the file.
          This is the second line. If I try and line it up...

Don't even think about pasting ASCII ART in here!
'
end
```

```
This is the first line of the file.
          This is the second line. If I try and line
it up...
```

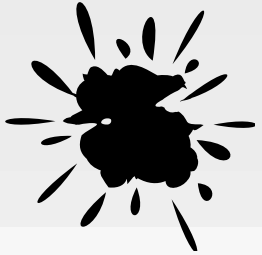
```
Don't even think about pasting ASCII ART in here!
```

It is important to note that the file content may have some important formatting that might be easily overlooked when working with the content in a recipe file.

Besides that, if the size of the string value of the content field grows, it will consume the recipe--making it difficult to understand what the desired state is and what the data is.


Instructor Note: The Message of the Day file is not a white-space important file. Other configuration files that could be managed with Chef may be white-space important.

Slide 10



Copy Paste

This process is definitely error prone. Especially because a human has to edit the file again before it is deployed.

©2015 Chef Software Inc. 7-10 

So if you are copying and pasting large amounts of text content into a double-quoted string it will be important to check that every time.

This sounds like a bug waiting to happen.

Any process that requires you to manually copy and paste values and then remember to escape out characters in a particular order, is likely going to lead to issues later when you deploy this recipe to production.



Slide 11

CONCEPT

What We Need

We need the ability to store the data in another file, which is in the native format of the file we are writing out but that still allows us to insert ruby code...

...specifically, the node attributes we have defined.




©2015 Chef Software Inc.

7-11

It is better to store this data in another file. The file would be native to whatever format is required so it you wouldn't need to escape any common characters.

But you still need a way to insert node attributes. So you really need a native file format that allows us to escape out to ruby.

Slide 12




GE: Cleaner Recipes

Adding the node attributes to our recipes did make it harder to read.

Objective:

- ☐ Decide which resource will help us address this issue

©2015 Chef Software Inc. 7-12 

To solve this problem, we need to read up on the file resource more or see if Chef provides alternatives.

Slide 13

DOCS



GE: Let's Check the Docs...


Use the `file` resource to manage files directly on a node.

Use the **`cookbook_file`** resource to copy a file from a cookbook's `/files` directory. Use the **`template`** resource to create a file based on a template in a cookbook's `/templates` directory. And use the **`remote_file`** resource to transfer a file to a node from a remote location.

https://docs.chef.io/resource_file.html

©2015 Chef Software Inc.

7-13




Let's start from what we know--the `file` resource. Open the documentation and see what it says and see if it gives us an clue to finding alternatives.

The `file` resource documentation suggests a couple of alternatives to using the `file` resource: `cookbook_file` resource; `template` resource; and `remote_file` resource.

Let's start with the `remote_file` resource.

Slide 14

DOCS




GE: remote_file

Use the **remote_file** resource to transfer a file from a remote location using file specificity. This resource is similar to the file resource.

https://docs.chef.io/resource_remote_file.html

©2015 Chef Software Inc.

7-14




Reading the documentation for `remote_file`, it seems that `remote_file` is similar to `file`. Except `remote_file` is used to specify a file at a remote location that is copied to a specified file path on the system.

So we could define our index file or message-of-the-day file on a remote system. But that does not allow us to insert attributes about the node we are currently on.

Slide 15

DOCS




GE: cookbook_file

Use the **cookbook_file** resource to transfer files from a sub-directory of COOKBOOK_NAME/files/ to a specified path located on a host that is running the chef-client.

https://docs.chef.io/resource_cookbook_file.html

©2015 Chef Software Inc.

7-15




Reading the documentation for `cookbook_file`, after the boiler-plate resource definition, it sounds as though a cookbook file is capable of...

Slide 16

Demo: cookbook_file's Source Match Up

```
$ tree /f cookbooks\iis-demo\files\default
Folder PATH listing
Volume serial number is B04A-119C
C:\USERS\ADMINISTRATOR\COOKBOOKS\IIS-DEMO\FILES\DEFAULT
    Default.htm
No subfolders exist
```



```
cookbook_file 'c:\inetpub\wwwroot\Default.htm' do
  source 'Default.htm'
end
```


...allowing us to store a file within our cookbook and then have that file transferred to a specified file path on the system.

While it sounds like it allows us to write a file in its native format, it does not sound as though the ability exists to escape out to access the node object and dynamically populate data.

Slide 17

DOCS


Template



A cookbook template is an Embedded Ruby (ERB) template that is used to generate files ... Templates may contain Ruby expressions and statements and are a great way to...

Use the template resource to add cookbook templates to recipes; place the corresponding Embedded Ruby (ERB) template in a cookbook's /templates directory.

https://docs.chef.io/resource_template.html

©2015 Chef Software Inc.7-17

Let's explore templates.


Reviewing the documentation, it seems as though it shares some similarities to the `cookbook_file` resource.

Slide 18

Demo: Template File's Source Matches Up

```
$ tree /f cookbooks\iis-demo\templates\default
Folder PATH listing
Volume serial number is B04A-119C
C:\USERS\ADMINISTRATOR\COOKBOOKS\IIS-DEMO\TEMPLATES\DEFAULT
    Default.htm.erb
No subfolders exist

template 'c:\inetpub\wwwroot\Default.htm' do
  source 'Default.htm.erb'
end
```




A template can be placed in a particular directory within the cookbook and it will be delivered to a specified file path on the system.

The biggest difference is that it says templates can contain ruby expressions and statements. This sounds like what we wanted: A native file format with the ability to insert information about our node.

Slide 19

DOCS

Template




To use a template, two things must happen:

1. A template resource must be added to a recipe
2. An Embedded Ruby (ERB) template must be added to a cookbook


https://docs.chef.io/resource_template.html#using-templates

©2015 Chef Software Inc.

7-19



And if we look at the bottom section about "Using Templates", we'll see more information about what is required and how we can use them to escape out to execute ruby code.



GE: Cleaner iis-demo Recipe

Adding the node attributes to the index page did make it harder to read the recipe.

Objective:

- ❑ Create a template with chef generate
- ❑ Define the contents of the ERB template
- ❑ Change the file resource to the template resource in the 'iis-demo' cookbook

©2015 Chef Software Inc. 7-20 

So our objective is clear. We need to use a template resource and create a template and then link them together.


Let's start by creating the actual template file and then we will update the recipe.

Slide 21

CONCEPT


What is chef?

An executable program that allows you generate cookbooks and cookbook components.



©2015 Chef Software Inc.

7-21



Remember that application Chef--the one that generated our cookbooks. Well it is able to generate cookbook components as well.

Templates and files (for `cookbook_files`) are a few of the other things it can generate for us.

GE: What Can chef Do?



```
$ chef --help
```

Usage:

```
chef -h/--help
chef -v/--version
chef command [arguments...] [options...]
```

Available Commands:

<code>exec</code>	Runs the command in context of the embedded ruby
<code>gem</code>	Runs the <code>`gem`</code> command in context of the embedded ruby
<code>generate</code>	Generate a new app, cookbook, or component
<code>shell-init</code>	Initialize your shell to use ChefDK as your primary ruby
<code>install</code>	Install cookbooks from a Policyfile and generate a locked cookbook set

Let's use help to review the command again.

GE: What Can chef generate Do?



```
$ chef generate --help
```

```
Usage: chef generate GENERATOR [options]
```

```
Available generators:
```

app	Generate an application repo
cookbook	Generate a single cookbook
recipe	Generate a new recipe
attribute	Generate an attributes file
template	Generate a file template
file	Generate a cookbook file
lwrp	Generate a lightweight resource/provider
repo	Generate a Chef policy repository
policyfile	Generate a Policyfile for use with the install/push commands (experimental)

And let's ask for help about the 'generate' subcommand.

Slide 24

GE: What Can chef generate template Do?



```
$ chef generate template --help
```

```
Usage: chef generate template [path/to/cookbook] NAME [options]
```

```
-C, --copyright COPYRIGHT      Name of the copyright holder - defaults  
to 'The Authors'
```

```
-m, --email EMAIL              Email address of the author - defaults to
```

```
...
```

```
-a, --generator-arg KEY=VALUE  Use to set arbitrary attribute KEY to  
VALUE in the
```

```
-I, --license LICENSE          all_rights, iis-demo2, mit, gplv2, gplv3  
- defaults to
```

```
-s, --source SOURCE_FILE       Copy content from SOURCE_FILE
```

```
-g GENERATOR_COOKBOOK_PATH,    Use GENERATOR_COOKBOOK_PATH for the  
code_generator
```

```
--generator-cookbook
```

Finally let's ask for help for generating templates.

The command requires two parameters--the path to where the cookbook is located and the name of the template to generate. There are some other additional options but these two seem like the most important.

GE: Use chef to Generate a Template



```
$ cd ~  
$ chef generate template cookbooks/iis-demo Default.htm
```

```
Compiling Cookbooks...
```

```
Recipe: code_generator::template
```

```
* directory[cookbooks/iis-demo/templates/default] action create  
  - create new directory cookbooks/iis-demo/templates/default  
* template[cookbooks/iis-demo/templates/default/ Default.htm.erb] action...  
  - create new file cookbooks/iis-demo/templates/default/Default.htm.erb  
  - update content in file cookbooks/iis-demo/templates/default/  
  Default.htm.erb from none to e3b0c4  
  (diff output suppressed by config)
```

Use '**chef generate template**' to create a template in the iis-demo cookbook found in the cookbooks/iis-demo directory and the file we want to create is named Default.htm.

Slide 26

GE: Lets Look at the Template File



```
$ tree /f cookbooks\iis-demo\templates
```

```
cookbooks/iis-demo/templates/
```

```
├─ default
```

```
├─ Default.htm.erb
```

```
1 directory, 1 file
```

That is the first step. Now that the template exists, we are ready to define the content within the template file.

Slide 27



Cleaner Recipes

Adding the node attributes to the default page did make it harder to read the recipe.

Objective:


- ✓ Create a template with chef generate
- ❑ Define the contents of the ERB template
- ❑ Change the file resource to the template resource in the 'iis-demo' cookbook

Now we need to understand what ERB means.

Slide 28

CONCEPT

ERB




An Embedded Ruby (ERB) template allows Ruby code to be embedded inside a text file within specially formatted tags.

Ruby code can be embedded using expressions and statements.

<https://docs.chef.io/templates.html#variables>

©2015 Chef Software Inc.

7-28



ERB template files are special files because they are the native file format we want to deploy but we are allowed to include special tags to execute ruby code to insert values or logically build the contents.

Slide 29

Text Within an ERB Template

```
<% if (50 + 50) == 100 %>
50 + 50 = <%= 50 + 50 %>
<% else %>
At some point all of MATH I learned in school changed.
<% end %>
```

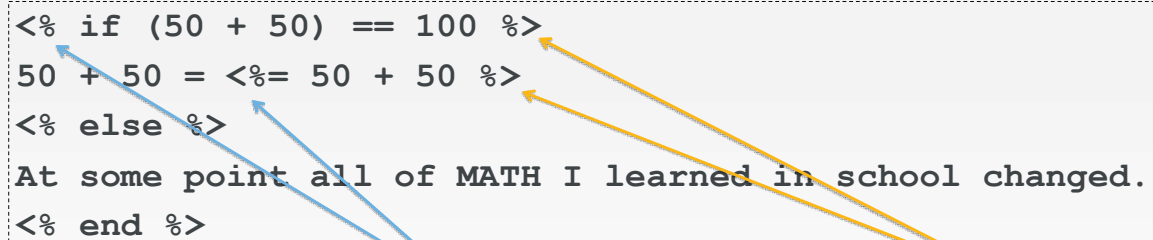
Each ERB tag has a beginning tag and a matched ending tag.

Here is an example of a text file that has several ERB tags defined in it.

Slide 30

Text Within an ERB Template

```
<% if (50 + 50) == 100 %>  
50 + 50 = <%= 50 + 50 %>  
<% else %>  
At some point all of MATH I learned in school changed.  
<% end %>
```

A diagram showing the structure of an ERB template. It contains a code block with an if-else statement. Blue arrows point from the opening tags to their corresponding closing tags: from the first opening tag to the first closing tag, from the opening tag of the else block to its closing tag, and from the opening tag of the if block to its closing tag. Yellow arrows point from the closing tags back to the opening tags: from the first closing tag to the first opening tag, from the closing tag of the else block to its opening tag, and from the closing tag of the if block to its opening tag.

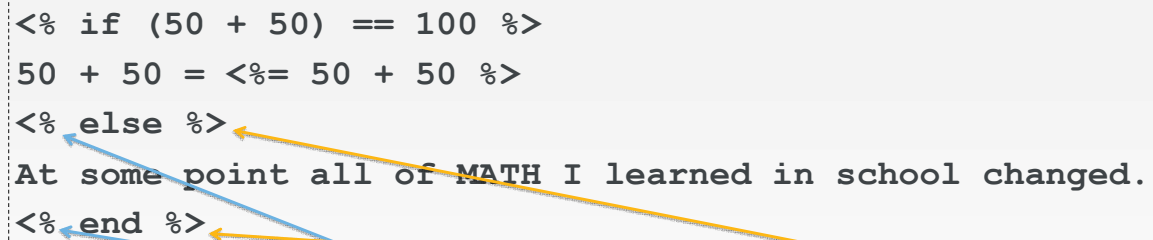
Each ERB tag has a beginning tag and a matched ending tag.

Each ERB tag has a beginning tag and an ending tag.

Slide 31

Text Within an ERB Template

```
<% if (50 + 50) == 100 %>  
50 + 50 = <%= 50 + 50 %>  
<% else %>  
At some point all of MATH I learned in school changed.  
<% end %>
```

A diagram illustrating the structure of ERB tags. It shows a code block with four lines. The first line is a beginning tag: <% if (50 + 50) == 100 %>. The second line is a text line: 50 + 50 = <%= 50 + 50 %>. The third line is another beginning tag: <% else %>. The fourth line is an ending tag: <% end %>. Blue arrows point from the opening <% of the first tag to the opening <% of the third tag, and from the closing %> of the third tag to the closing %> of the first tag. Yellow arrows point from the opening <% of the third tag to the opening <%= of the second line, and from the closing %> of the second line to the closing %> of the third tag.

Each ERB tag has a beginning tag and a matched ending tag.

The beginning tag is a less-than sign followed by a percent sign. The closing tag is a percent sign followed by a greater-than sign.

Text Within an ERB Template

```
<% if (50 + 50) == 100 %>
50 + 50 = <%= 50 + 50 %>
<% else %>
At some point all of MATH I learned in school changed.
<% end %>
```

Executes the ruby code within the brackets and do not display the result.

These tags are used to execute ruby but the results are not displayed.

Text Within an ERB Template

```
<% if (50 + 50) == 100 %>  
50 + 50 = <%= 50 + 50 %>  
<% else %>  
At some point all of MATH I learned in school changed.  
<% end %>
```


Executes the ruby code within the brackets and display the results.



ERB supports additional tags, one of those is one that allows you to output some variable or some ruby code. Here the example is going to display that 50 plus 50 equals the result of ruby calculating 50 plus 50 and then displaying the result.

Slide 34

CONCEPT

The Angry Squid





©2015 Chef Software Inc.

7-34

The starting tag is different. It has an equals sign. This means show the value stored in a variable or the result of some calculation.

We often refer to this opening tag that outputs the content as the Angry Squid. The less-than is its head, the percent sign as its eyes, and the equals sign its tentacles shooting away after blasting some ink.

GE: Move Our Source to the Template

```
~\cookbooks\iis-demo\templates\default\Default.htm.erb
```

```
<html>
  <body>
    <h1>Hello, world!</h1>
    <h2>ipaddress: #{node['ipaddress']}</h2>
    <h2>hostname: #{node['hostname']}</h2>
    <h2>total memory: #{node['memory']['total']}</h2>
    <h2>CPU Mhz: #{node['cpu']['0']['mhz']}</h2>
  </body>
</html>
```

With that in mind let's update the template with the current value of the file resource's content field.

Copying this literally into the file does not work because we no longer have the ability to use string interpolation within this html file. String interpolation only works within a ruby file between a double-quoted String.

GE: Replace String Interpolation with ERB

~\cookbooks\iis-demo\templates\default\Default.htm.erb

```
<html>
  <body>
    <h1>Hello, world!</h1>
    <h2>ipaddress: <%= node['ipaddress'] %></h2>
    <h2>hostname: <%= node['hostname'] %></h2>
    <h2>total memory: <%= node['memory']['total'] %></h2>
    <h2>CPU Mhz: <%= node['cpu']['0']['mhz'] %></h2>
  </body>
</html>
```

We are going to need to change string interpolation sequence with the ERB template syntax. And it seems for this content we want to display the output so we want to make sure that we are using ERB's angry squid opening tag.

Slide 37




Cleaner Recipes

Adding the node attributes to the default page did make it harder to read the recipe.

Objective:

- ✓ Create a template with chef generate
- ✓ Define the contents of the ERB template
- ❑ Change the file resource to the template resource in the 'iis-demo' cookbook

©2015 Chef Software Inc. 7-37 

The template is created and the contents are correctly defined. It is time to update the recipe.

Slide 38

GE: Remove the Existing Content Attribute

```
~\cookbooks\iis-demo\recipes\server.rb

# ... POWERSHELL_SCRIPT RESOURCE ...

file 'c:\inetpub\wwwroot\Default.htm' do
  content "<h1>Hello, world!</h1>
<h2>ipaddress: #{node['ipaddress']}</h2>
<h2>hostname: #{node['hostname']}</h2>
<h2>total memory: #{node['memory']['total']}</h2>
<h2>CPU Mhz: #{node['cpu']['0']['mhz']}</h2>"
end

# ... SERVICE RESOURCE ...
```

Let's open the iis-demo cookbook's recipe named 'server'.

We will want to remove the content attribute from the file resource. Because that content is now in the template. But only if we use a template resource.

GE: Change the File Resource to a Template

```
~\cookbooks\iis-demo\recipes\server.rb  
  
# ... POWERSHELL_SCRIPT RESOURCE ...  
  
template 'c:\inetpub\wwwroot\Default.htm' do  
  
end  
  
# ... SERVICE RESOURCE ...
```

So it's time to change the file resource to a template resource so that it can use the template file that we have defined.

What to Specify as the Source?

```
~\cookbooks\iis-demo\recipes\server.rb
```

```
# ... POWERSHELL_SCRIPT RESOURCE ...
```

```
template 'c:\inetpub\wwwroot\Default.htm' do
```

```
  source "?????????????"
```

```
end
```

```
# ... SERVICE RESOURCE ...
```

Lastly we need to specify a source attribute which contains that path to the template we generated. This path is relative starting from within the cookbook's template directory.

Slide 41

GE: Viewing the Partial Path to the Template



```
$ tree /f cookbooks\iis-demo\templates\default
```

```
Folder PATH listing
```

```
Volume serial number is B04A-119C
```

```
C:\USERS\ADMINISTRATOR\COOKBOOKS\IIS-DEMO\TEMPLATES\DEFAULT
```

```
Default.htm.erb
```

```
No subfolders exist
```

To visualize that with 'tree' we can run it with a path that places us right at the templates directory. So the results will be relative paths from the point specified.

And we see the filepath Default.htm.erb.

Instructor Note: The default folder denotes that we want to use this file for all platforms.

Slide 42

GE: Change the File Resource to a Template

```
~\cookbooks\iis-demo\recipes\server.rb  
  
template '/var/www/html/index.html' do  
  source 'Default.htm.erb'  
end
```

Now we have the path to our template so we can update the template resource's source attribute value.

Slide 43




Cleaner Recipes

Adding the node attributes to the default page did make it harder to read the recipe.

Objective:

- ✓ Create a template with chef generate
- ✓ Define the contents of the ERB template
- ✓ Change the file resource to the template resource in the 'iis-demo' cookbook

©2015 Chef Software Inc. 7-43 

We hopefully haven't changed the original goal of our recipe but we have made some changes.

Slide 44



Lab: Update the Version

- ☐ Use kitchen converge and kitchen verify on the "iis-demo" cookbook
- ☐ Use chef-client to apply the "iis-demo" cookbook's "default" recipe
- ☐ Update the "iis-demo" cookbook's version for this patch
- ☐ Commit the changes

In this lab, you will use 'kitchen' to verify the cookbook and use 'chef-client' to apply the cookbook. If everything is working then update the patch number and commit the changes to version control.

Instructor Note: Allow 8 minutes to complete this exercise.

Lab: Test the Cookbook



```
$ cd ~\cookbooks\iis-demo
$ kitchen converge
$ kitchen verify
```

```
-----> Starting Kitchen (v1.4.2)
-----> Setting up <default-windows-2012r2>...
        Finished setting up <default-windows-2012r2> (0m0.00s) .
-----> Verifying <default-windows-2012r2>...
..

Finished in 1.69 seconds (files took 1.01 seconds to load)
2 examples, 0 failures

        Finished verifying <default-windows-2012r2> (0m2.60s) .
```

Since kitchen is a cookbook testing tool, you need to move into the cookbook's directory.

Then run the 'kitchen converge' and 'kitchen verify' command, addressing any issues if they show up.

Slide 46

Lab: Change Directories and Apply the Cookbook



```
$ cd ~
$ chef-client --local-mode -r "recipe[iis-demo]"

[2015-09-16T14:18:05+00:00] WARN: No config file found or specified on command line,
using command line options.
Starting Chef Client, version 12.3.0
resolving cookbooks for run list: ["iis-demo"]
Synchronizing Cookbooks:
  - iis-demo
Compiling Cookbooks...
[2015-09-16T14:18:09+00:00] WARN: Cloning resource attributes for service[httpd]
from prior resource (CHEF-3694)
[2015-09-16T14:18:09+00:00] WARN: Previous service[httpd]: /root/.chef/local-mode-
cache/cache/cookbooks/iis-demo/recipes/server.rb:8:in `from_file'
[2015-09-16T14:18:09+00:00] WARN: Current service[httpd]: /root/.chef/local-mode-
cache/ ...
```

When all the tests pass, return to the home directory, so you can execute 'chef-client'.

And then apply the iis-demo cookbook's default recipe to the local system.

Slide 47

Lab: Update the Cookbook's Patch Number

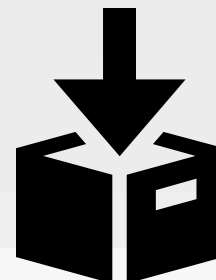
```
~\cookbooks\iis-demo\metadata.rb

name                'iis-demo'
maintainer           'The Authors'
maintainer_email     'you@example.com'
license              'all_rights'
description           'Installs/Configures iis-demo'
long_description     'Installs/Configures iis-demo'
version              '0.2.1'
```

If everything converges correctly, update the version number. As mentioned previously, this is a patch fix.

Slide 48

Lab: Commit the Changes




```
$ cd ~\cookbooks\iis-demo  
$ git add .  
$ git status  
$ git commit -m "Changed file resource to template  
resource and defined a template"
```

Return to the cookbook directory and add all the changed files and commit them with a message.

Slide 49

DISCUSSION




Discussion

What is the benefit of using a template over defining the content within a recipe? What are the drawbacks?

What do each of the ERB tags accomplish?

©2015 Chef Software Inc.

7-49




Answer these questions.

With your answers, turn to another person and alternate asking each other these questions and sharing your answers.

Slide 50

DISCUSSION




Q&A

What questions can we help you answer?

- Resources (file, cookbook_file, template, and remote_file)
- Templates
- ERB

©2015 Chef Software Inc.

7-50



What questions can we help you answer?

Generally or specifically about resources, templates, and ERB.

Slide 51

