

Storing User Credentials in Configuration Files

The `<credentials>` section is rarely used in real-world applications because it's highly impractical to store user data in `web.config`. Worse, you surely noticed that passwords are stored in clear text, which makes them visible to anyone who has read access to the `web.config` file. Most of the time, you don't have to worry about this detail because you'll often prefer to store user credentials in a database, but it's good to know that you can encrypt passwords in the `web.config` file:

```
<credentials passwordFormat="SHA1">
  <user name="JoeDoe" password="7C9690380BB6A10B886AEFF2202F94C5C8FFCB92" />
  <user name="AnnSmith"
    password="D69DCC3EABAF925EF64BD625B0F045EACE4F0478" />
</credentials>
```

The `passwordFormat` attribute specifies the encryption algorithm used to encode the password and can be either SHA1, MD5, or Clear. SHA1 is more secure than MD5, but it's slower and produces longer passwords. Of course, you must encode the passwords before storing them in `web.config`. You do the encryption with the `HashPasswordForStoringInConfigFile` shared method of the `System.Web.Security.FormsAuthentication` class. The name of this method says it all: you pass the password and a string indicating the encryption method, and the method returns the encrypted password:

```
' The txtPassword control contains the password as entered by the user.
Dim encryptedPassword As String = FormsAuthentication. _
    HashPasswordForStoringInConfigFile(txtPassword.Text, "SHA1")
```

The value of the `passwordFormat` attribute is inherited from `machine.config` if you omit it. You might believe that it defaults to Clear and be tempted to omit it when you're storing passwords in clear text; unfortunately, `machine.config` sets SHA1 as the default password format, so you must specify this attribute when you're using unencrypted passwords.