

Therefore, the contents of `$newupstring` (the new user info) are appended to `$output`. The `$output` string is then used to replace all contents of the user ID/password XML file. Once this information has been saved successfully, the user log is updated to indicate the creation of a new user ID, and the user is redirected to the login screen to sign in to the application with the new user ID and password.

Alternatively, the previous process of loading the file into an associative array, updating the associative array, and then loading the associate array back into the XML file could have been used. However, it would have taken more code and is unnecessary because this process is only used once in the application. The program will not make multiple attempts to update the user ID and password file with the same user.

Example 7-3. The registration.php file

```
<?php
session_start();
$user_log_file = "user.log";
try
{
if ((isset($_POST['username'])) || (isset($_POST['password'])))
{
$userid = $_POST['username'];
$password = $_POST['password'];
if (!(preg_match("/^.*(?:={8,})?(?=[a-z]*\d)(?=[a-z]*[A-Z]).*$/", $password)) ||
(!(strlen($userid) >= 8)))
{
throw new Exception("Invalid Userid and/or Password Format");
}
}
else
{
libxml:use_internal_errors(true);
$xmlDoc = new DOMDocument();
if ( file_exists("e7dog_applications.xml") )
{
$xmlDoc->load( 'e7dog_applications.xml' );
$searchNode = $xmlDoc->getElementsByTagName( "type" );
foreach( $searchNode as $searchNode )
{
$valueID = $searchNode->getAttribute('ID');

if($valueID == "UIDPASS")
{
$xmlLocation = $searchNode->getElementsByTagName( "location" );
$dog_data_xml = $xmlLocation->item(0)->nodeValue;
break;
}
}
}
else
{
throw new Exception("Dog applications xml file missing or corrupt");
}
} else {
throw new Exception("Dog applications xml file missing or corrupt");
}
}
```