The delete_Process method will place the change file records into $change_Array using the same change_Array as shown before. It will pass the record number to be deleted ($_GET['rn']), the number of rows in the array ($row_Count), and the array ($change_Array) into the deleteRecord method. The deleteRecords method will use the same logic as shown in the deleteRecord method from the readerrorlog (in Example 5-8) program. The delete_Process will then call the saveChanges method, passing in the row_count, change_Array, and change_File information.

```
function saveChanges($row_Count,$change_Array,$change_File)
{
        $changeFile = fopen($change_File, "w");
        for($I=0; $I < $row_Count; $I++)
        {
            $writeString = $change_Array[$I][0] . " | " . $change_Array[$I][1] . " | " .
            $change_Array[$I][2];
           fwrite($changeFile, $writeString);
        }
        fclose($changeFile);
}
```

The saveChanges method builds the date/time-changetype-changedata format, seen previously, from the change_Array. This information is saved in $writeString and is used to replace the change log file with the updated version (minus the record that was deleted).

The delete_Process method then recalls the displayRecords method (described earlier) to display the updated change log (minus the record deleted) and the data file drop-down list.

Once the user selects the data file to be changed, the update_XML_File_Process method is called.

```
function update_XML_File_Process()
{
$change_Array = load_Array();
require_once("dog_data.php");
$data_Changer = new dog_data();
$row_Count = count($change_Array) -1;
for($I=0;$I < $row_Count; $I++)
{
   if($change_Array[$I][1] != "Delete")
        {
                $temp = unserialize($change_Array[$I][2]);
        }
        else
        {
                $temp = (integer)$change_Array[$I][2];
        }
        $data_Changer->processRecords($change_Array[$I][1], $temp);
}
```

The method calls the load_Array method to return the changes into $change_Array. The dog_data file is imported into the method to prepare for the changes to the data file selected by the user. An instance of the data_data class is created ($data_Changer).