```
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);
}
$data_Changer->setChangeLogFile($_POST['data_File']);
$data_Changer = NULL;
echo "Changes completed";
}
// main section
if(isset($_POST['data_File'])) {
            update_XML_File_Process();
} else if(isset($_GET['rn'])) {
            delete_Process();
} else if(isset($_POST['change_file'])) {
            display_Process();
} else {
select_File_Process();
}
?>
```

JSON Backup and Recovery

What changes are needed to provide backup and recovery for JSON data instead of XML data? Actually, no changes at all. As long as the changes from the first section of this chapter are implemented, the `displaychangelog` program and the changes to the `dog_data` class will handle JSON in the same manner as XML data.

MySQL Backup and Recovery

As you might be guessing, as long as the changes from the second section of this chapter are implemented, no additional changes will be required for backup and recovery of MySQL data. However, you can take a moment to look at an alternative way of handling MySQL data.

It is a common practice to create a SQL script file to execute against a database. A script file contains all the SQL code necessary to update the database. Using this type of file will allow you to do proper `INSERT`, `UPDATE`, and `DELETE` SQL commands instead of only an `INSERT` as previously shown. The previous example required creating an `INSERT` command for every record in the associate array. This includes records that were not changed. This would be inefficient for medium to large databases. You only need to update the record that changed.

You can develop the scripting file from records in the associate array that have changed. You can use the change log as the script file, as the SQL script lists all changes that have been requested. It can be rerun to fix any corrupted data.