```
                        {
                                $xmlstring .= "<$column>" . $dog_value[$column] . "</$column>\n";
                        }
                        $xmlstring .= "</$dogs>\n";
        }   }
$xmlstring .= "</dogs>\n";
$new_valid_data_file = preg_replace('/[0-9]+/', '', $this->dog_data_xml);
// remove the previous date and time if it exists
$oldxmldata = date('mdYhis') . $new_valid_data_file;
if (!rename($this->dog_data_xml, $oldxmldata)) {   throw new Exception("Backup file
$oldxmldata could not be created."); }
file_put_contents($new_valid_data_file,$xmlstring);
}
private function deleteRecord($recordNumber)
{
foreach ($this->dogs_array as $dogs=>&$dogs_value) {
                for($J=$recordNumber; $J < count($dogs_value) -1; $J++) {
                        foreach ($dogs_value[$J] as $column => $column_value)
                        {
                                $dogs_value[$J][$column] = $dogs_value[$J + 1][$column];
                        }
}
        unset ($dogs_value[count($dogs_value) -1]);
        }
        $change_string = date('mdYhis') . " | Delete | " . $recordNumber . "\n";
        $chge_log_file = date('mdYhis') . $this->change_log_file;
        error_log($change_string,3,$chge_log_file); // might exceed 120 chars
 }
private function readRecords($recordNumber)
 {
        if($recordNumber === "ALL") {
                return $this->dogs_array["dog"];
        } else {
        return $this->dogs_array["dog"][$recordNumber];
        }
}
private function insertRecords($records_array)
{
        $dogs_array_size = count($this->dogs_array["dog"]);
        for($I=0;$I< count($records_array);$I++) {
                $this->dogs_array["dog"][$dogs_array_size + $I] = $records_array[$I];
        }
        $change_string = date('mdYhis') . " | Insert | " . serialize($records_array) . "\n";
        $chge_log_file = date('mdYhis') . $this->change_log_file;
        error_log($change_string,3,$chge_log_file); // might exceed 120 chars
}
private function updateRecords($records_array)
{
        foreach ($records_array as $records=>$records_value)
        {
                foreach ($records_value as $record => $record_value)
```