The following loop tests if the random number has been previously generated (by checking for its presence in an array). This ensures that a random number is not used more than once.

Also tests if record ID generated is a valid metadata record and contains entries in specific elements

Next line checks if record is found by checking for existence of error message

The following loop tests if the random number has been previously generated (by checking for its presence in an array). This ensures that a random number is not used more than once.

AND also tests if record ID generated is a valid metadata record and contains entries in specific elements.

The complete metadata record is extracted and sent to an output file.

Results are sent to 2 output files; one for human translators, one for machine translator.

The random number array is updated to include the new random number.

The following loop tests if the random number has been previously generated (by checking for its presence in an array). This ensures that a random number is not used more than once.

AND also tests if record ID generated is a valid metadata record and contains entries in specific elements.

Tests if record ID generated is a valid metadata record and contains entries in specific elements.

Also tests if record ID generated is a valid metadata record and contains entries in specific elements.

Next line checks if record is found by checking for existence of error message

The following loop tests if the random number has been previously generated (by checking for its presence in an array). This ensures that a random number is not used more than once.

AND also tests if record ID generated is a valid metadata record and contains entries in specific elements.

The complete metadata record is extracted and sent to an output file.

Results are sent to 2 output files; one for human translators, one for machine translator.

The random number array is updated to include the new random number.

The following loop tests if the random number has been previously generated (by checking for its presence in an array). This ensures that a random number is not used more than once.

AND also tests if record ID generated is a valid metadata record and contains entries in specific elements.

The complete metadata record is extracted and sent to an output file.

Results are sent to 2 output files; one for human translators, one for machine translator.
mysql_query("INSERT INTO MDR_DET2B (MDR_ID, ELEMENT, VALUE)
VALUES ('$mdr_id', '$srchstring[$i]','$stringoutdb')"); // Update MDR_DET Table with extracted field element data - 5 elements per record.

fwrite ($outtmp2,$out2);

fwrite ($outtmp2, "</metadata-ID: $name$header$num>
" );

if (count($rand) < 10) {
    fwrite ($outtmp, "$rand
" );
}

fwrite ($outtmp, "###
"

if ($key < $value) {
    fwrite ($outattrib,
"
"
    foreach($recattrib as $key => $value){
        for ($i = 0; $i <= 7; $i++) {
            ${$tmpout.$i} = $recattrib[$key][$i];
        }
        mysql_query("INSERT INTO SAMPLE_FEATURES2B (TotRecSize, WrdCntTitle, WrdCntCreator, WrdCntPublisher, WrdCntDescription, WrdCntSubject, WrdCntCoverage, MDR_ID)
VALUES ($recattrib1, $recattrib2, $recattrib3, $recattrib4, $recattrib5, $recattrib6, $recattrib7, '$recattrib0')");
    }
}

$endtime = date("m.d.y").', T:'.date("H:i:s"); // Time script run ended
$output = "Start: "$starttime."n"; // send start time of run to output
fwrite ($outtmp, $strtout );
fwrite ($outtmp, $endout );
/* This function takes an entire metadata xml record as stored in $content and extracts a specific metadata element field based on parameters passed on to it */
function extractfields($wd,$content,$outtmp2,$outtmp3,&$wrdcount,&$stringout){
$srchtxt = '#<dc:'.$wd.'>.*?.</dc:'.$wd.'>#' ;
$numof = 'numof';
$arrcnt = 'arrcnt';
$field = 'field';
$matches = 'matches';
${$numof.$wd} = preg_match_all($srchtxt, $content, ${$matches.$wd});
${$arrcnt.$wd} = 0;
${$wd.$field} = null;
/* The following loop checks for the presence of multiple lines of the same field, extracts and concatenates them into one string delimited by the ## accordingly */
while (${$arrcnt.$wd} < ${$numof.$wd}) {
    ${$wd.$field} .= trim(strip_tags($matches.$wd)[0][${$arrcnt.$wd}]);
    ${$arrcnt.$wd}++;
    if ((${$numof.$wd} > 1) AND (${$arrcnt.$wd} != ${$numof.$wd})) {
        ${$wd.$field} .= " # # ";
    }
}
$stringout = ${$wd.$field};
/* The next set of statements counts the number of words in the field and sends both the number and the words to an output file */
$allowedchar = '$0123456789.àáâäèéêëìíîïòóôöùúûü'; // includes characters as valid when found in a word
$wrdcount = str_word_count($stringout,0,$allowedchar);
fwrite ($outtmp3,'Word count for '.$wd.': '.$wrdcount."n" );
$arrayout = str_word_count($stringout,1,$allowedchar);
if ($wrdcount != 0) {
    fwrite ($outtmp3,'Words are: ');$key = 0;
    $value = count($arrayout);
    for ($l = 0; $l < $value; $l++) {
        fwrite ($outtmp3, $arrayout[$l]." ");
    }
    fwrite ($outtmp3,"n" );
}
}