

```
<?php
// Import of XML data from XML files created by "MEDICS" offline EMR application
// development by mi-squared.com -2010 licences under GPL v2 or greater
//
// Provided as an example of how to import complete medical records into OpenEMR
// from an XML formatted file
//
// This is intended to be run by a automated command line process/cron job not a web request
// Based on OpenEMR 3.1 release
// See import_mi2xml-xpaths.txt for a field structure details

$ignoreAuth = true;
include_once("../interface/globals.php");
include_once("$srcdir/patient.inc");
include_once("$srcdir/forms.inc");
include_once("$srcdir/acl.inc");

// set name of XML file
if ( $argc < 1 ) {
    echo "No XML file path provided.\n";
    exit(-1);
}

array_shift($argv);
$file = $argv[0];

// load file
$medics = simplexml_load_file($file) or die ("Unable to load XML file!");

////////////////////////////////////
process_medics( $medics );
////////////////////////////////////

////////////////////////////////////
// FUNCTION START
////////////////////////////////////

function process_medics ( $medics ) {

    // add patient
    // - medical history
    // - family/social history
    {
        $errors = array();
        $msg = create_patient( $medics, $errors );
        $has_errors = sizeof($errors) > 0;
        if ( $has_errors ) {
            render_errors( $errors );
            return;
        } else {
            echo $msg;
        }
    }

    // add encounter:
    // - encounter form
    // - vitals
    {
        $errors = array();
        $msg = add_encounter( $medics, $errors );
        $has_errors = sizeof($errors) > 0;
        if ( $has_errors ) {
            render_errors( $errors );
            return;
        } else {
            echo $msg;
        }
    }
}
```

```
    }
}

function render_errors( &$errors ) {
    $first = true;
    foreach( $errors as $error ) {
        if ( !$first ) {
            echo "|";
        }
        echo "ERR[$error]";
        $first = false;
    }
}

function create_patient( &$medics, &$errors ) {

    $alertmsg = "";
    $patient_pid = get_patientid($medics);

    $pubpid = trim( $medics->pubpid );

    // ID must be valid or 'NEWPATIENT'
    if (empty($pubpid)) {
        array_push( $errors, "Patient ID '$pubpid' missing, patient skipped! ");
        return $alertmsg;
    }

    if ( $pubpid != 'NEWPATIENT' ) {

        // 1. validate patient
        $patient_pid = 0;
        $query = "SELECT pid FROM patient_data WHERE pubpid LIKE '$pubpid'";
        $res = sqlStatement($query);
        $row = sqlFetchArray($res);

        if ($row) {
            $patient_pid = $row['pid'];
            if (sqlFetchArray($res)) {
                array_push( $errors, "Patient ID '$pubpid' is ambiguous, patient skipped! ");
                return $alertmsg;
            } else {
                // array_push( $errors, "Patient ID '$pubpid' exists, updates/new encounters only. ");
            }
        }
    }

    // 2. validate insurance provider - REMOVED

    // 3. validate billing provider

    $tmp = array();
    $tmp = fetchProviderInfo($medics);
    if (!array($tmp)) {
        array_push( $errors, "Provider '$tmp' not found, patient skipped!");
        return $alertmsg;
    }

    $patient_provider_id = $tmp['id'];
    $patient_provider_name = $tmp['username'];
    $patient_provider_facility = $tmp['facility_id'];

    // 4. get facility from // Move to function
    $row = sqlQuery("SELECT id, name, pos_code FROM facility WHERE id = '$patient_provider_facility' ");

    if (!$row['id']) {
        array_push( $errors, "Facility '$tmp' not found, patient skipped! ");
    }
}
```

```

return $alertmsg;
}
$patient_facility_id = $row['id'];
$patient_facility_name = $row['name'];
$patient_facility_pos = $row['pos_code'];

// 5. insert patient data
if (!$patient_pid) {

// Insert into patient_data.
//
$row = sqlQuery("SELECT max(pid)+1 AS pid FROM patient_data");
$patient_pid = $row['pid'] ? $row['pid'] : 1;

// Combine street lines
$patient_street = $medics->street . ' ' . $medics->street2;
// Build array
newPatientData(
'', // id
'', // title
form2db($medics->fname), // fname
form2db($medics->lname), // lname
form2db($medics->mname), // mname
sex($medics->sex), // sex
form2db($medics->dob), // dob
form2db($patient_street), // street
'', // Dutch: nstreet
'', // Dutch: nnr
'', // Dutch: nadd
form2db($medics->zip), // zip
form2db($medics->city), // city
form2db($medics->state), // state
'', // country
'', // ssn
'', // occupation
form2db($medics->phone_home), // phone_home
form2db($medics->phone_alternate), // phone_biz
'', // phone_contact
'', // status
'', // contact_relationship
'', // referrer
'', // referrerID
'', // email
'', // language
form2db($medics->ethnicity), // ethnoracial
'', // interpreter
'', // migrantseasonal
'', // family_size
'', // monthly_income
'', // homeless
'0000-00-00 00:00:00', // financial_review
$patient_pid, // pubpid - use PID when NEWPATIENT
$patient_pid, // pid
'', // providerID
'', // genericname1
'', // genericval1
'', // genericname2
'', // genericval2
'', // phone_cell
form2db($medics->hippa_notice), // hipaa_mail
form2db($medics->hippa_notice), // hipaa_voice
'', // squad
'', // pharmacy_id
'', // drivers_license
form2db($medics->hippa_notice), // hipaa_notice
'', // $hipaa_message
$dos = fixDate($medics->fromdate) // regdate

```

```

);

// Insert dummy row for employer_data.
newEmployerData($patient_pid);

// Update or Instest subscriber ins data
if( ($medics->pubpid == 'NEWPATIENT') || (!empty($medics->policy_id) ) ) {
newInsuranceData(
  $patient_pid,
  'primary',
  $insurance_company_id,      // (insurance) provider
  form2db($medics->policy_id), // policy_number - same as pt identifier?
  '',                          // group_number - anything special here?
  '',                          // plan_name - anything special here?
  form2db($medics->lname),      // subscriber_lname
  form2db($medics->mname),      // subscriber_mname
  form2db($medics->fname),      // subscriber_fname
  'self',                      // subscriber_relationship
  '',                          // subscriber_ss
  fixDate($medics->dob),        // subscriber_DOB
  form2db($medics->street),      // subscriber_street
  form2db($medics->zip),        // subscriber_postal_code
  form2db($medics->city),       // subscriber_city
  form2db($medics->state),      // subscriber_state
  '',                          // subscriber_country
  form2db($medics->phone_home), // subscriber_phone
  '',                          // subscriber_employer
  '',                          // subscriber_employer_street
  '',                          // subscriber_employer_city
  '',                          // subscriber_employer_postal_code
  '',                          // subscriber_employer_state
  '',                          // subscriber_employer_country
  '',                          // copay
  sex($medics->sex),            // subscriber_sex
  fixDate($medics->eff_date)    // effective date
);
}

$tmp = $medics->lname . ',' . $medics->fname;
$alertmsg .= "New Patient Added: '$patient_pid' / '$tmp' <br>\n";
}

$medics->pid = $patient_pid;

$history = array(
  'history_father'           => form2db($medics->familyinformation->father),
  'history_mother'          => form2db($medics->familyinformation->mother),
  'history_spouse'          => form2db($medics->familyinformation->spouse),
  'history_siblings'        => form2db($medics->familyinformation->siblings),
  'history_offspring'       => form2db($medics->familyinformation->offspring),
  'relatives_cancer'        => form2db($medics->medical->relativesexperience->cancer),
  'relatives_tuberculosis'  => form2db($medics->medical->relativesexperience->tuberculosis),
  'relatives_diabetes'      => form2db($medics->medical->relativesexperience->diabetes),
  'relatives_high_blood_pressure' => form2db($medics->medical->relativesexperience-
>highbloodpressure),
  'relatives_heart_problems' => form2db($medics->medical->relativesexperience->heartproblems),
  'relatives_stroke'        => form2db($medics->medical->relativesexperience->stroke),
  'relatives_epilepsy'      => form2db($medics->medical->relativesexperience->epilepsy),
  'relatives_mental_illness' => form2db($medics->medical->relativesexperience->mentallillness),
  'relatives_suicide'       => form2db($medics->medical->relativesexperience->suicide),
  'usertext12'              => form2db($medics->medical->relativesexperience->other),
  'coffee'                  => form2db($medics->medical->lifestyleusage->coffee),
  'tobacco'                  => form2db($medics->medical->lifestyleusage->tobacco),
  'alcohol'                  => form2db($medics->medical->lifestyleusage->alcohol),
  'sleep_patterns'          => form2db($medics->medical->lifestyleusage->sleep),
  'exercise_patterns'       => form2db($medics->medical->lifestyleusage->exercise),
  'seatbelt_use'            => form2db($medics->medical->lifestyleusage->seatbelt),

```

```

'counseling'           => form2db($medics->medical->lifestyleusage->counseling),
'hazardous activities' => form2db($medics->medical->lifestyleusage->hazardactivities),
'usertext13'          => form2db($medics->medical->lifestyleusage->urinaryincontinence),
'usertext14'          => form2db($medics->medical->lifestyleusage->fallhistory),
'additional_history'  => form2db($medics->medical->lifestyleusage->other) . " " .
                        form2db($medics->medical->lifestyleusage->generalnotes)
);

// Insert/Update into history_data.
if ($medics->pubpid == 'NEWPATIENT') {
    newHistoryData($patient_pid, $history);
} else {
    updateHistoryData($patient_pid, $history);
}

// Add or Update History data
add_update_history($medics, $patient_pid, $errors);

// Create or update an issue for each historical medication.
//
foreach ($medics->medical->medications->medication as $medication) {

    if (isempty($medication->name)) continue;

    $meds = array();
    $meds['title']     = form2db($medication->name);
    $meds['dosage']    = form2db($medication->dosage);
    $meds['frequency'] = form2db($medication->frequency);
    $meds['duration']  = form2db($medication->duration); // TBD does not exist in MEDICS
    $meds['id']        = form2db($medication->id);

    if ( !isempty($meds['id']) ) {
        $row = sqlQuery("SELECT id FROM lists WHERE id = " . $meds['id'] );
        if (!$row) {
            create_issue($patient_pid, 'medication', $meds );
        } else {
            update_issue($patient_pid, 'medication', $meds );
        }
    } else {
        create_issue($patient_pid, 'medication', $meds );
    }
}

return $alertmsg;
}

// Create a new issue in the lists table.
//
function create_issue($pid, $type, $fields) {

    if ( !isempty( $fields['title'] ) ) {
        echo "\nAdding new issue '" . $fields['title'] . "'";

        sqlInsert("INSERT INTO lists SET " .
            "date = NOW(), " .
            "pid = '$pid', " .
            "type = '$type', " .
            "title = '" . $fields['title'] . "', " .
            "activity = 1, " .
            "user = '" . $_SESSION['authUser'] . "', " .
            "groupname = '" . $_SESSION['authProvider'] . "', " .
            "outcome = '', " .
            "destination = '', " .
            "frequency = '" . $fields['frequency'] . "', " .
            "dosage = '" . $fields['dosage'] . "', " .
            "duration = '" . $fields['duration'] . "'
        );
    }
}

```

```

    } else {
        echo "Skipping field creation with empty title.";
    }
}

// update issue in the lists table.
//
function update_issue($pid, $type, $fields) {

    echo "\nUpdating issue id " . $fields['id'] . " (" . $fields['title'] . ")";

    sqlInsert("update lists SET " .
        "date = NOW(), " .
        "pid = '$pid', " .
        "type = '$type', " .
        "title = '" . $fields['title'] . "', " .
        "activity = 1," .
        "user = '" . $$SESSION['authUser'] . "', " .
        "groupname = '" . $$SESSION['authProvider'] . "', " .
        "outcome = '', " .
        "destination = '', " .
        "frequency = '" . $fields['frequency'] . "', " .
        "dosage = '" . $fields['dosage'] . "', " .
        "duration = '" . $fields['duration'] . "' " .
        "WHERE id = " . $fields['id']
    );
}

function add_update_history ($medics, $patient_pid, &$errors) {
    $msg;
    $dos = fixDate($medics->fromdate);

    if (!isempty($medics->medical->medicalhistory->medicationnotes) ||
        !isempty($medics->medical->medicalhistory->allergies) ||
        !isempty($medics->medical->medicalhistory->history) ||
        !isempty($medics->medical->medicalhistory->surgicalhistory) ||
        !isempty($medics->medical->medicalhistory->preventative)) {

        $row = sqlQuery("SELECT pid FROM aperio_medical_history WHERE pid = '$patient_pid'");
        if (!$row) {
            sqlInsert("INSERT INTO aperio_medical_history SET " .
                "pid = '$patient_pid', " .
                "medication = '" . $medics->medical->medicalhistory->medicationnotes . "', " .
                "allergies = '" . $medics->medical->medicalhistory->allergies . "', " .
                "medical = '" . $medics->medical->medicalhistory->history . "', " .
                "surgical = '" . $medics->medical->medicalhistory->surgicalhistory . "', " .
                "preventative = '" . $medics->medical->medicalhistory->preventative . "' " .
                . "updated = '$dos'"
            );
            // return( "Created history form for Patient ID:" . $patient_pid);
        } else {
            sqlStatement("UPDATE aperio_medical_history SET " .
                "medication = '" . $medics->medical->medicalhistory->medicationnotes . "', " .
                "allergies = '" . $medics->medical->medicalhistory->allergies . "', " .
                "medical = '" . $medics->medical->medicalhistory->history . "', " .
                "surgical = '" . $medics->medical->medicalhistory->surgicalhistory . "', " .
                "preventative = '" . $medics->medical->medicalhistory->preventative . "' " .
                // "updated = '$dos'" .
                " WHERE pid = '$patient_pid'"
            );
            // return("Updated history form for Patient ID:" . $patient_pid);
        }
    }
}

```

```

    // return("No History form created for Patient ID:" . $patient_pid);
}

function add_encounter( $medics, &$errors ) {
    $msg;

    $patient_pid = $medics->pid;
    $dos = fixDate($medics->fromdate);
    $encounter_id = $GLOBALS['adodb']['db']->GenID('sequences');
    $encounter_reason = form2db($medics->chiefcomplaint);
    addForm($encounter_id, "New Patient Encounter",
        sqlInsert("INSERT INTO form_encounter SET " .
            "date = '$dos', " .
            "onset_date = '$dos', " .
            "reason = '$encounter_reason', " .
            "facility = '$patient_facility_name', " .
            "facility_id = '$patient_facility_id', " .
            "sensitivity = 'normal', " .
            "pid = '$patient_pid', " .
            "encounter = '$encounter_id'"
        ),
        "newpatient", $patient_pid, 1, $dos
    );

    $msg .= "Created new encounter for " . $medics->pubpid . ".<br>\n";

// Custom Forms

    $msg .= add_ros_subj ( $patient_pid, $dos, $encounter_id, $medics, $errors ); //ROS
    $msg .= add_phyexam_obj( $patient_pid, $dos, $encounter_id, $medics, $errors ); //PE
    $msg .= add_assessment_plan( $patient_pid, $dos, $encounter_id, $medics, $errors ); //Assessment/Plan
    $msg .= add_vitals( $patient_pid, $dos, $encounter_id, $medics, $errors ); //
    $msg .= add_other_dx( $patient_pid, $dos, $encounter_id, $medics, $errors ); //Other DX

// Fee Sheet
    $msg .= add_billing_records( $patient_pid, $dos, $encounter_id, $medics, $errors );

    return $msg;
}

function isempty( $str ) {
    return $str == null || empty($str);
}

function add_ros_subj( $patient_pid, $dos, $encounter_id, $medics, &$errors ) {

    if (
        !isempty($medics->medical->subjective->general) ||
        !isempty($medics->medical->subjective->neurological) ||
        !isempty($medics->medical->subjective->heent) ||
        !isempty($medics->medical->subjective->respiratory) ||
        !isempty($medics->medical->subjective->cardio) ||
        !isempty($medics->medical->subjective->gastro) ||
        !isempty($medics->medical->subjective->skin) ||
        !isempty($medics->medical->subjective->extremities) ||
        !isempty($medics->medical->subjective->subjective)
    ) {

        $row = fetchProviderInfo($medics);

        if ($row['id']) { // TBD array error check NEEDED
            $patient_provider_name = $row['username'];
        }

        addForm($encounter_id, "ROS",
            sqlInsert("INSERT INTO form_aperio_ros SET " .

```

```

        "date = '$dos', " .
        "pid = '$patient_pid', " .
        "authorized = 1, " .
        "activity = 1, " .
        "general = '" . form2db($medics->medical->subjective->general) . "', " .
        "neurological = '" . form2db($medics->medical->subjective->neurological) . "', " .
        "heent = '" . form2db($medics->medical->subjective->heent) . "', " .
        "respiratory = '" . form2db($medics->medical->subjective->respiratory) . "', " .
        "cardio = '" . form2db($medics->medical->subjective->cardio) . "', " .
        "gastro = '" . form2db($medics->medical->subjective->gastro) . "', " .
        "skin = '" . form2db($medics->medical->subjective->skin) . "', " .
        "extremities = '" . form2db($medics->medical->subjective->extremities) . "', " .
        "subjective = '" . form2db($medics->medical->subjective->subjective) . "'
    ),
    "aperio_ros", $patient_pid, 1, $dos, $patient_provider_name
);

    return "Added ROS for Patient ID " . $patient_pid . ".";
}
}

function add_phyexam_obj( $patient_pid, $dos, $encounter_id, $medics, &$errors ) {

    if (!isempty($medics->medical->objective->general) ||
        !isempty($medics->medical->objective->neurological) ||
        !isempty($medics->medical->objective->heent) ||
        !isempty($medics->medical->objective->respiratory) ||
        !isempty($medics->medical->objective->cardio) ||
        !isempty($medics->medical->objective->gastro) ||
        !isempty($medics->medical->objective->skin) ||
        !isempty($medics->medical->objective->extremities) ||
        !isempty($medics->medical->objective->objective))
    {

        $row = fetchProviderInfo($medics);
        if ($row['id']) { // TBD array error check NEEDED
            $patient_provider_name = $row['username'];
        }

        addForm($encounter_id, "Physical Exam",
        sqlInsert("INSERT INTO form_aperio_pe SET " .
            "date = '$dos', " .
            "pid = '$patient_pid', " .
            "authorized = 1, " .
            "activity = 1, " .
            "general = '" . form2db($medics->medical->objective->general) . "', " .
            "neurological = '" . form2db($medics->medical->objective->neurological) . "', " .
            "heent = '" . form2db($medics->medical->objective->heent) . "', " .
            "respiratory = '" . form2db($medics->medical->objective->respiratory) . "', " .
            "cardio = '" . form2db($medics->medical->objective->cardio) . "', " .
            "gastro = '" . form2db($medics->medical->objective->gastro) . "', " .
            "skin = '" . form2db($medics->medical->objective->skin) . "', " .
            "extremities = '" . form2db($medics->medical->objective->extremities) . "', " .
            "objective = '" . form2db($medics->medical->objective->objective) . "'
        ),
        "aperio_pe", $patient_pid, 1, $dos, $patient_provider_name
    );

        return "Added PE for Patient ID " . $patient_pid . ".";
    }
}

function add_assessment_plan( $patient_pid, $dos, $encounter_id, $medics, &$errors ) {

    if (!isempty($medics->medical->systems->assessment) ||
        !isempty($medics->medical->systems->plan))

```



```

{
    $row = fetchProviderInfo($medics);
    if ($row['id']) { // TBD array error check NEEDED
        $patient_provider_name = $row['username'];
    }

    addForm($encounter_id, "Assessment-Plan",
    sqlInsert("INSERT INTO form_aperio_ap SET " .
        "date = '$dos', " .
        "pid = '$patient_pid', " .
        "authorized = 1, " .
        "activity = 1, " .
        "assessment = '" . form2db($medics->medical->systems->assessment) . "', " .
        "plan = '" . form2db($medics->medical->systems->plan) . "'
    ),
    "aperio_ap", $patient_pid, 1, $dos, $patient_provider_name
    );

    return "Added Assessment for Patient ID " . $patient_pid . ".";
}
}

function add_other_dx( $patient_pid, $dos, $encounter_id, $medics, &$errors ) {

    if (!isempty($medics->medical->systems->otherdx)) {
        $row = fetchProviderInfo($medics);
        if ($row['id']) { // TBD array error check NEEDED
            $patient_provider_name = $row['username'];
        }

        addForm($encounter_id, "Other DX",
        sqlInsert("INSERT INTO form_aperio_opd SET " .
            "date = '$dos', " .
            "pid = '$patient_pid', " .
            "authorized = 1, " .
            "activity = 1, " .
            "opd = '" . form2db($medics->medical->systems->otherdx) . "'
        ),
        "aperio_opd", $patient_pid, 1, $dos, $patient_provider_name
        );

        return "Added OtherDX for Patient ID " . $patient_pid . ".";
    }
}

function add_vitals( $patient_pid, $dos, $encounter_id, $medics, &$errors ) {
    $bmi;
    $bmi_status;

    // Create Vitals form.
    if (!empty($medics->medical->physicalexamsvitals->bps)) {

/* DISABLED
// Calculate BMI_status
if ($medics->medical->physicalexamsvitals->weight > 0 &&
    $medics->medical->physicalexamsvitals->height > 0) {
    $bmi = ($weight/$height/$height)*703;
    if ( $bmi > 42 )      $bmi_status = 'Obesity III';
    elseif ( $bmi > 34 ) $bmi_status = 'Obesity II';
    elseif ( $bmi > 30 ) $bmi_status = 'Obesity I';
    elseif ( $bmi > 27 ) $bmi_status = 'Overweight';
    elseif ( $bmi > 25 ) $bmi_status = 'Normal BL';
    elseif ( $bmi > 18.5 ) $bmi_status = 'Normal';
    elseif ( $bmi > 10 ) $bmi_status = 'Underweight';
    }
*/
}
}

```

```

addForm($encounter_id, "Vitals",
  sqlInsert("INSERT INTO form_vitals SET " .
    "date = '$dos', " .
    "pid = '$patient_pid', " .
    "user = '" . form2db($patient_provider_name) . "', " .
    "groupname = '" .
    "authorized = 1, " .
    "activity = 1, " .
    "mentalstate = '" . form2db($medics->medical->physicalexamsvitals->mentalstatus) . "', " .
    "bps = '" . form2db($medics->medical->physicalexamsvitals->bps) . "', " .
    "bpd = '" . form2db($medics->medical->physicalexamsvitals->bpd) . "', " .
    "weight = '" . form2db($medics->medical->physicalexamsvitals->weight) . "', " .
    "height = '" . form2db($medics->medical->physicalexamsvitals->height) . "', " .
    "temperature = '" . form2db($medics->medical->physicalexamsvitals->temperature) . "', " .
    "temp_method = '" . form2db($medics->medical->physicalexamsvitals->tempmethods) . "', " .
    "pulse = '" . form2db($medics->medical->physicalexamsvitals->pulse) . "', " .
    "respiration = '" . form2db($medics->medical->physicalexamsvitals->respiration) . "', " .
    "BMI = '" . form2db($medics->medical->physicalexamsvitals->bmi) . "', " .
    "BMI_status = '$bmi_status', " .
    "waist_circ = '" . form2db($medics->medical->physicalexamsvitals->waistcirc) . "', " .
    "head_circ = '" . form2db($medics->medical->physicalexamsvitals->headcirc) . "', " .
    "oxygen_saturation = '" . form2db($medics->medical->physicalexamsvitals->o2) . "'
  ),
  "vitals", $patient_pid, 1, $dos
);

return "Created vitals form for Patient ID " . $patient_pid . ".\n";
}

return "No vitals form created for Patient ID " . $patient_pid . ".\n";
}

function add_billing_records( $patient_pid, $dos, $encounter_id, $medics, &$errors ) {
  $row = fetchProviderInfo($medics);
  $patient_provider_id = $row['id'];

  $diags = array();

  for ( $diag = 0; $diag < 15; $diag++ ) {

    $_code = trim($medics->medical->systems->diagnosis[$diag]->code);
    $_codenote = trim($medics->medical->systems->diagnosis[$diag]->codenote);
    $_codestatus = trim($medics->medical->systems->diagnosis[$diag]->codestatus);

    if (empty($_code)) { continue; }
    $diags[] = $_code;

    echo "\n--> :: $_code $_codenote $_codestatus\n";
    add_billing(
      $patient_pid,
      $encounter_id,
      $patient_provider_id,
      'ICD9',
      $_code,
      $_codenote . ' ( ' . $_codestatus . ' )'
    );
  }

  if (!empty($medics->medical->billable)) {

    $cptCode = $medics->medical->billable->service[0]->code;
    $codeNote= $medics->medical->billable->service[0]->codenote;

    for ($justify = '', $j = 0; $j < 4 && $j < count($diags); ++$j) {
      $justify .= $diags[$j] . ':';
    }
  }
}

```

```

    }
    add_billing($patient_pid, $encounter_id, $patient_provider_id,
        'CPT4', $cptCode, $codeNote, $justify);
}
}

// Write a row to the billing table.
//
function add_billing($pid, $encounter, $provider, $codetype, $code,
    $description, $justify='', $modifier='', $units=1) {

    // echo "\n ADD BILLING: ---> $pid $encounter $provider $codetype $code $description $justify $modifier
    $units\n";

    // Get the fee from the codes table.
    $fee = 0;
    if ($codetype == 'CPT4') {
        $query = "SELECT fee FROM codes WHERE code_type = 1 AND code = '$code' AND ";
        if (empty($modifier))
            $query .= "( modifier IS NULL OR modifier = '')";
        else
            $query .= "modifier = '$modifier'";
        $row = sqlQuery($query);
        if ($row['fee']) $fee = $row['fee'] * $units;
    }

    sqlInsert("INSERT INTO billing ( " .
        "date, code_type, code, pid, provider_id, user, groupname, authorized, " .
        "encounter, code_text, billed, activity, payer_id, modifier, units, " .
        "fee, justify " .
        ") VALUES ( " .
        "NOW(), " .
        "'$codetype', " .
        "'$code', " .
        "'$pid', " .
        "'$provider', " .
        "' " . $$ _SESSION['authUser'] . "', " .
        "' " . $$ _SESSION['authProvider'] . "', " .
        "1, " .
        "'$encounter', " .
        "'$description', " .
        "0, " .
        "1, " .
        "'$insurance_company_id', " .
        "'$modifier', " .
        "'$units', " .
        "'$fee', " .
        "'$justify'" .
        ")");
}

function get_patientid( $medics ) {

    $pubpid = trim( $medics->pubpid );
    $patient_pid = 0;
    $query = "SELECT pid FROM patient_data WHERE pubpid LIKE '$pubpid'";
    $res = sqlStatement($query);
    $row = sqlFetchArray($res);
    if ($row) {
        $patient_pid = $row['pid'];
    }

    return $patient_pid;
}

```

```
}

// Encode a string from a form field for database writing.
//
function form2db($fldval) {
    $fldval = trim($fldval);
    if (!get_magic_quotes_gpc()) $fldval = addslashes($fldval);
    return $fldval;
}

// Encode sex for OpenEMR compatibility.
//
function sex($insex) {
    if (!empty($insex)) {
        $insex = strtoupper(substr($insex, 0, 1));
        if ($insex == 'M') return 'Male';
        if ($insex == 'F') return 'Female';
    }
    return '';
}

function fetchProviderInfo($medics) {
    $query = "SELECT id, username, facility_id FROM users WHERE npi = '" . $medics->provider_npi . "'";
    $row = sqlQuery($query);
    if (!$row['id']) {
        array_push($errors, "Provider '" . $medics->provider_npi . "' not found");
        return $alertmsg;
    }
    return ($row);
}

?>

<?php

/*

# XPATHS used by import_mi2xml.php scripts
# Includes customized encounter forms (see the code) provided for example
#
# Some xpaths can be repeated many times, see comments below
#
# Provided by Medical Information Integration, LLC
# www.mi-squared.com
#
patient/pubpid
patient/insurance_id
patient/policy_id
patient/eff_date
patient/lname
patient/fname
patient/mname
patient/sex
patient/dob
patient/ethnicity
patient/street
patient/street2
patient/city
patient/state
patient/zip
patient/phone_home
patient/phone_alternate
patient/hippa_notice
patient/provider_npi
patient/provider_name
patient/fromdate
patient/chiefcomplaint
```

patient/familyinformation/father  
patient/familyinformation/mother  
patient/familyinformation/spouse  
patient/familyinformation/siblings  
patient/familyinformation/offspring

patient/medical/relativeexperience/cancer  
patient/medical/relativeexperience/tuberculosis  
patient/medical/relativeexperience/diabetes  
patient/medical/relativeexperience/highbloodpressure  
patient/medical/relativeexperience/heartproblems  
patient/medical/relativeexperience/stroke  
patient/medical/relativeexperience/epilepsy  
patient/medical/relativeexperience/mentalillness  
patient/medical/relativeexperience/suicide  
patient/medical/relativeexperience/other  
patient/medical/lifestyleusage/coffee  
patient/medical/lifestyleusage/tobacco  
patient/medical/lifestyleusage/alcohol  
patient/medical/lifestyleusage/sleep  
patient/medical/lifestyleusage/exercise  
patient/medical/lifestyleusage/seatbelt  
patient/medical/lifestyleusage/counseling  
patient/medical/lifestyleusage/hazardactivities  
patient/medical/lifestyleusage/urinaryincontinence  
patient/medical/lifestyleusage/fallhistory  
patient/medical/lifestyleusage/other  
patient/medical/lifestyleusage/generalnotes

patient/medical/medicalhistory/medicationnotes  
patient/medical/medicalhistory/allergies  
patient/medical/medicalhistory/history  
patient/medical/medicalhistory/surgicalhistory  
patient/medical/medicalhistory/preventative

# Multiple segments supported in current script

patient/medical/medications/medication/name  
patient/medical/medications/medication/dosage  
patient/medical/medications/medication/frequency  
patient/medical/medications/medication/id

# Custom Form

patient/medical/subjective/general  
patient/medical/subjective/neurological  
patient/medical/subjective/heent  
patient/medical/subjective/respiratory  
patient/medical/subjective/cardio  
patient/medical/subjective/gastro  
patient/medical/subjective/skin  
patient/medical/subjective/extremities  
patient/medical/subjective/subjective

# Custom Form

patient/medical/objective/general  
patient/medical/objective/neurological  
patient/medical/objective/heent  
patient/medical/objective/respiratory  
patient/medical/objective/cardio  
patient/medical/objective/gastro  
patient/medical/objective/skin  
patient/medical/objective/extremities  
patient/medical/objective/objective

# Custom Form

patient/medical/physicalexamsvitals/mentalstatus

```
patient/medical/physicalexamsvitals/bps
patient/medical/physicalexamsvitals/bpd
patient/medical/physicalexamsvitals/weight
patient/medical/physicalexamsvitals/height
patient/medical/physicalexamsvitals/temperature
patient/medical/physicalexamsvitals/tempmethods
patient/medical/physicalexamsvitals/pulse
patient/medical/physicalexamsvitals/respiration
patient/medical/physicalexamsvitals/bmi
patient/medical/physicalexamsvitals/waistcirc
patient/medical/physicalexamsvitals/headcirc
patient/medical/physicalexamsvitals/o2
```

```
# Custom Form
```

```
patient/medical/systems/assessment
patient/medical/systems/plan
```

```
# Multiple segments supported in current script
```

```
patient/medical/systems/diagnosis/code
patient/medical/systems/diagnosis/codenote
patient/medical/systems/diagnosis/codestatus
```

```
# Custom Form
```

```
patient/medical/systems/otherdx
```

```
# Multiple segments supported in current script
```

```
patient/medical/billable/service[1]/code
patient/medical/billable/service[1]/codenote
patient/medical/billable/service[1]/modifier
```

```
*/
```

```
?>
```