Memorandum

To: Tony Marshall

cc: SIS Group, G. Petencin, P. Murphy, J. Webber

From: J. Effland

Date: 2002-11-15

Subject: System to Track the Progress of Jobs in the CDL's Machine Shop

1 Introduction

Tracking the progress of jobs in the CDL's machine shop could improve communication between the shop and it's customers, where the term "customer" here means anyone who submits a job to the shop. A tracking system that doesn't unduly burden the shop staff with reports should improve the CDL's overall efficiency because the CDL staff will be more aware if the status of their jobs.

A web-based tracking system, based on a similar design for tracking effort[1], has been designed to minimize the effort required for the machine shop staff to report on a job's status. This memo provides instructions for using the system along with design methodology and code listings.

2 History

Initial attempts to track the status of shop jobs used a spreadsheet to list job names, the customer's name, the machinist assigned to the work, expected completion dates, *etc*. The spreadsheet approach proved cumbersome for several reasons:

a) The shop foreman, Tony Marshall, spent too much time entering data into the spreadsheets.

b) Only rudimentary status information (start date and completion date) about the job was available.

c) Viewing of the spreadsheet by customers was awkward, because Excel was required on the customer's computer.

d) Simultaneous viewing of the spreadsheet by more than one person was cumbersome, and would occasionally lock Tony Marshall out of his own spreadsheet.

That system was abandoned in early September, 2002.

[1] "Proposed System to Track Effort Expended for ALMA Band 6 Cartridge Construction," Internal NRAO memo by J. Effland, 2002-07-31.

3 Proposed System

The new web-based system has been designed to manage the shop's work flow and to provide a simple means for customer's to view the status of their jobs. This system has the following features:

a) All jobs are visible to anyone with a web browser (Figure 1).

b) The customer enters a new job using a web form, by typing a job description and notes (see Figure 2).

c) Events for a job selected in the main screen are listed by clicking on the job number (see Figure 3)

d) Anyone can enter a new event for a selected job (see Figure 4)

The system is intended to minimize the amount of data entry by Tony Marshall, because the customer is responsible for filling out a new job form prior to submitting drawings to the shop.

4 Web Forms

The initial web form that displays the status of all shop jobs in progress, as shown in Figure 1, is available at

http://www.cv.nrao.edu/~jeffland/nrao-only/Progs/Shop/Jobs1.php3

The contents of the form are stored in a database (see Appendix for details) and a listing of completed jobs is available from a hyperlink on the page.

4.1 Shop Job Listing

From the screen shown in Figure 1, the user can either enter a new job by clicking the appropriate button, see events for a particular job by clicking on the job number, or view a listing of all completed jobs.

There are two important limitations that will be addressed in future software builds:

a) There is no way for the user to change any of the entries once they are entered into the database (although I can change them using other programs to access the database), and

b) the table can't be searched or sorted by, for example, the customer's name.

4.2 Input Form for New Jobs

Customers enter new jobs for the shop from the simple form shown in Figure 2, which is available by clicking a button on the shop "job listing" screen. This form provides default entries for the date and time fields. The **Date Entered** field defaults to the present time, and the **Date Required** field defaults to a date that is 28 days from the present date. The user can change these dates if desired. Limited validation of the entries is provided in this early software release:

c) The **Job Description** and **Your Name** fields can't be left blank.

d) A standard library function checks the validity of the date fields, but is of limited usefulness. For example, entering "11/30/62" brings up the "illegal format" screen, but entering "11-30-62" results in the system accepting the date but a "zero date" value is stored in the field.

e) Time can be omitted from the date fields and will default to midnight.

4.3 Job Event Listing

Each job has certain events associated with it, and the "Job Event Listing" tabulates events for a particular job, such as:

a) The date that the drawings are received by the shop,

b) the date drawings are checked by the shop,

c) the estimated completion date for the job, as assigned by Tony Marshall,

d) the name of the machinist assigned to the job,

e) the hours worked by the machinist on the job,

f) the actual completion date for the job, and

g) notes from the customer or machinist.

A listing of these events is obtained from the initial shop job screen by clicking on a job number shown in Figure 1, which brings up the "Event Listing" screen shown in Figure 3. Events for the job are sorted by date with the most recent entry at the top of the screen.

4.4 Input Form for New Events

New events for a particular job are entered using the screen shown in Figure 4, which is available by clicking on a button in the "Event Listing" screen. A single event can be entered, such as when the customer enters a note about the job, or multiple events can be entered simultaneously. Multiple events might include documenting that the drawings were both received and checked.

A number of features are incorporated in this input form to minimize data entry effort:

a) Checkboxes are used where possible.

b) The present time is entered for the appropriate date/time fields,

c) Tony's name is entered for the **Drawings checked by** field,

d) The **Estimated completion date** field has a number of predefined selections such as **Tomorrow, 1 Week** (from the present date), and **Other**, which includes a default date 4 weeks from the present.

Events cannot be changed once they are entered into the system, which admittedly is a serious limitation that will be addressed in future software versions. The work-around is to simply enter another event that includes the corrections.

5 Future Enhancements

The system can be enhanced in many ways, including:

a) Sorting the data shown on the "job listing" screen by clicking on any of the headings.

b) More extensive data validation of input fields.

c) Providing a way to search for a particular job.

d) Allowing modification of entries already in the database.

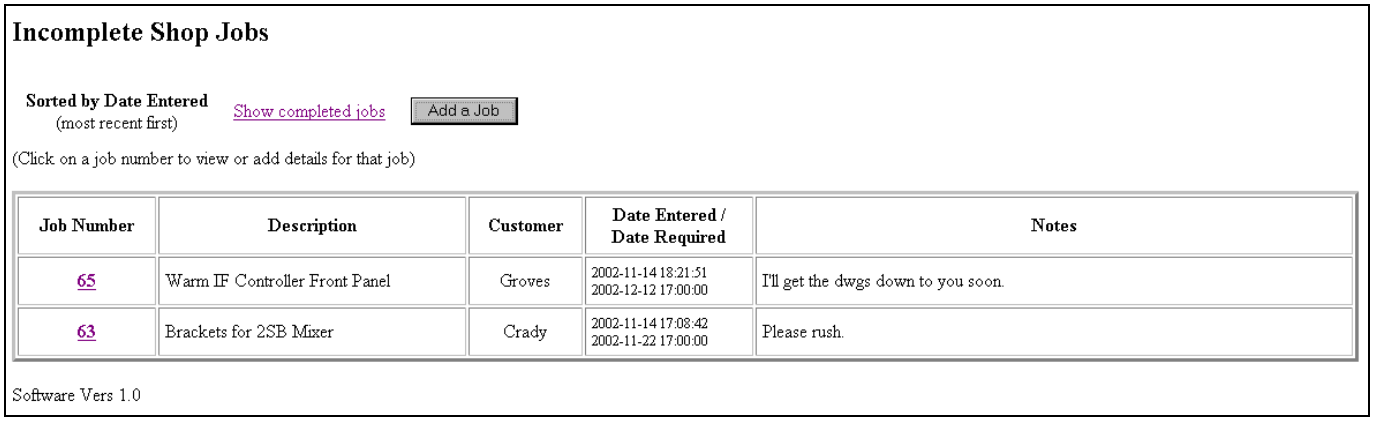
e) The use of cookies so users would have their name automatically entered into appropriate fields.

f) Additional security to:

   i) authenticate users and prevent them from entering data or from modifying certain fields.

g) Reporting screens to list job, machinist assigned, and hours expended.

An essential enhancement is to restructure the PHP code for more optimal reuse, such as moving all database statements into a PHP class. This enhancement, while invisible to the user, is required before the code grows much larger.



Incomplete Shop Jobs

Sorted by Date Entered
(most recent first)

Show completed jobs

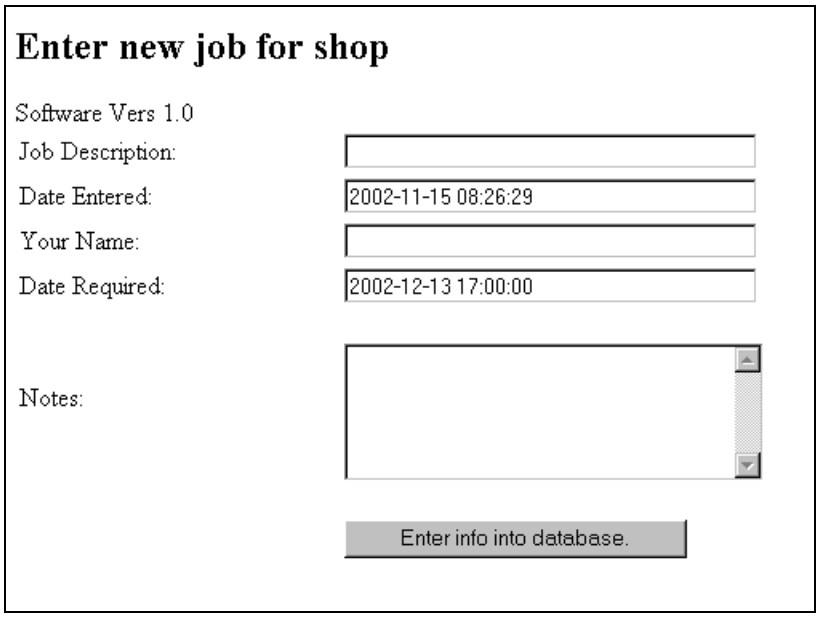
Add a Job

(Click on a job number to view or add details for that job)

Job Number	Description	Customer	Date Entered / Date Required	Notes
65	Warm IF Controller Front Panel	Groves	2002-11-14 18:21:51 2002-12-12 17:00:00	I'll get the dwgs down to you soon.
63	Brackets for 2SB Mixer	Crady	2002-11-14 17:08:42 2002-11-22 17:00:00	Please rush.

Software Vers 1.0

Figure 1: Listing of shop jobs to be completed



Enter new job for shop

Software Vers 1.0

Job Description:

Date Entered: 2002-11-15 08:26:29

Your Name:

Date Required: 2002-12-13 17:00:00

Notes:

Enter info into database.

Figure 2: Job Input Form

Event listing for shop job Add an event Return to main form

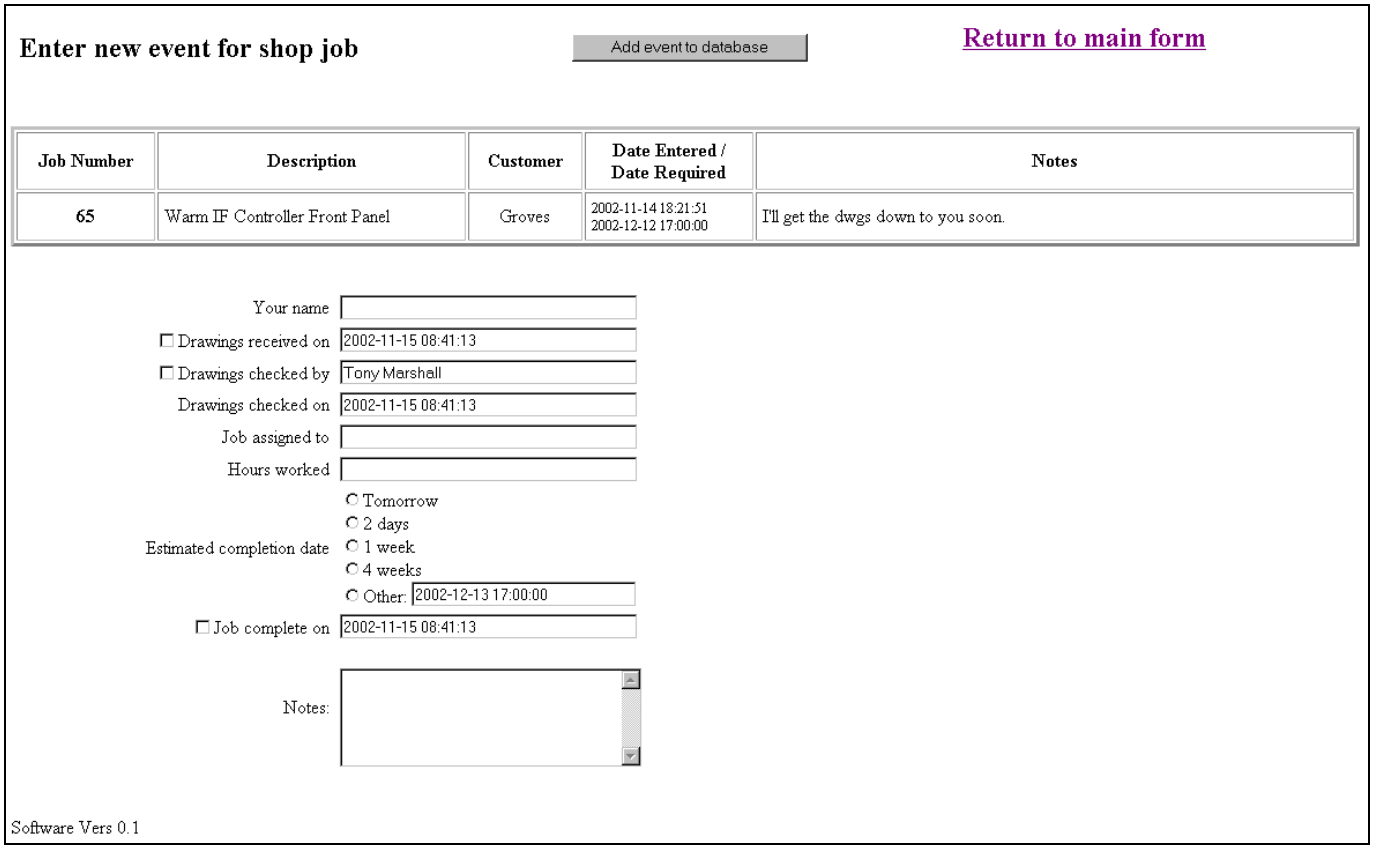
Job Number	Description	Customer	Date Entered / Date Required	Notes
65	Warm IF Controller Front Panel	Groves	2002-11-14 18:21:51 2002-12-12 17:00:00	I'll get the dwgs down to you soon.

Events:

Date Item Entered	Item	Notes
2002-11-14 18:24:24	Job **assigned to Vince** by Wharam	
2002-11-14 18:24:02	Wharam: **4 hrs of effort.**	
2002-11-14 18:23:45	Marshall: job should **complete on 2002-11-16 17:00:00**.	
2002-11-14 18:23:27	Job **assigned to Wharam** by Marshall	
2002-11-14 18:22:55	Tony Marshall: **checked dwgs.**	
2002-11-14 18:22:40	Marshall: **received dwgs**.	

Software Vers 2.1

Figure 3: Listing of events for the selected job



Enter new event for shop job Add event to database Return to main form

Job Number	Description	Customer	Date Entered / Date Required	Notes
65	Warm IF Controller Front Panel	Groves	2002-11-14 18:21:51 2002-12-12 17:00:00	I'll get the dwgs down to you soon.

Your name
☐ Drawings received on 2002-11-15 08:41:13
☐ Drawings checked by Tony Marshall
Drawings checked on 2002-11-15 08:41:13
Job assigned to
Hours worked
Estimated completion date ○ Tomorrow ○ 2 days ○ 1 week ○ 4 weeks ○ Other: 2002-12-13 17:00:00
☐ Job complete on 2002-11-15 08:41:13
Notes:

Software Vers 0.1

Figure 4: Input screen for adding events to the selected job

6 Appendix: Software and Database Design

6.1 Design Philosophy

A database serves to hold the job and event information. Records in the database are displayed using web pages and updated with web-based input forms. This allows the job information to be changed by simply adding or modifying records in the database.

6.2 Database Schema

The open source program MySQL is used as the database server because it has been tightly integrated with the Apache web server program and is presently supported by the NRAO for simple access from web pages. The MySQL server is located at `sql.cv.nrao.edu`. Pat Murphy has reluctantly volunteered to be the MySQL database administrator and as usual, is very responsive to questions.

MySQL lacks many of the advanced features of commercially available databases, such as facilities to graphically display database schemas, so schemas for the tables discussed below were generated manually. I used the free program "MySQL-Front" (http://www.anse.de/mysqlfront/) to manage the database tables. Microsoft's Access database program is also used because is provides a simple query interface and easy modification of database records.

Two database tables described next, `tblShopJobs` and `tblShopEvents`, store the data. The foreign-key field `fkShopJobs` in table `tblShopEvents` links multiple event records with a single shop job. Note that the completion date is entered in both tables because its useful to include it as an event for the job, but the query for completed jobs is simplified when it's included in the `tblShopJobs` table.

Table 1 :Description of table `tblShopJobs`

Purpose	**Each record provides information on a particular job in the machine shop**	
Field Name	**Field Type**	**Comments**
`keyShopJobs`	`int(3) unsigned`	Auto-Incremented key field
`DateAssigned`	`datetime`	Initial date of job entry
`DateRequired`	`datetime`	Desired date from customer
`DateCompleted`	`datetime`	Actual date that job was completed
`Task`	`varchar(50)`	Description of job
`Customer`	`varchar(50)`	Customer name
`notes`	`text`	Details about this job

Table 2: Description of table `tblShopEvents`		
Purpose	**Each record provides information about events associated with a particular job.**	
Field Name	**Field Type**	**Comments**
keyShopEvents	Int(3) unsigned	Auto-Incremented key field
fkShopJobs	Int(10) unsigned	Foreign key linked to table tblShopJobs
DateUpdated	datetime	Date this entry was made
EntryBy	varchar(50)	Name of person making the event entry
AssignedTo	varchar(50)	Name of machinist assigned to job
DateDwgsRec	datetime	Date drawings were received in the shop
DateDwgsChecked	datetime	Date drawings were checked by someone in the shop
DwgsCheckedBy	varchar(50)	Name of person that checked the drawings
DateEstCompletion	datetime	Estimated date for finishing the job
DateCompleted	datetime	Actual date that the job was completed
Effort	float	Hours required to complete the job
Notes	text	General purpose notes field

6.3 Web Page Software

The web page and database access software was written using the scripting language PHP. There is an incredible number of scripting languages available for dynamic web page construction, but Pat Murphy recommended PHP. The free program PHPEdit Version 0.6 (http://www.phpedit.com/) was used to write and edit most of the PHP code. I couldn't find any objective reviews of free PHP editors, so PHPEdit was selected as an impulse purchase. The editor is OK but the undocumented debugger is incredibly crude and buggy, and even highlights incorrect line numbers when flagging syntax errors. PHPEdit works well enough to be usable, at least until we purchase a real PHP editor.

When a URL pointing to a PHP file is entered using a browser, the web server first executes the PHP statements in the file, and the PHP program then outputs HTML to the browser. An notable limitation is that PHP programs can be executed from the Window's explorer because that program somehow circumvents the HTTP protocol. If the HTML code associated with the web forms shown here is examined using a browser's source button, the PHP listings shown below are not visible, but rather just the HTML output from those programs is shown.

Listing 1 is the software for the "job listing" form. This software first connects to the database using parameters stored in an include file that uses the php3 extension. This provides rudimentary security because web users can *execute,* but not view, php3-type files. Additional security results because all the code resides behind NRAO's firewall and hence is only accessible from the internal network.

Most of the PHP files are reentrant in that they can be executed multiple times and program flow changes depending on the state of certain variables. PHP builds the HTML statement

```
echo "<form method=\"post\" action=\"$ThisCode\">";
```

which causes the same code to reload when the appropriate submit button is pressed on the form. PHP sets the variable $btnNewJob to TRUE when the **Add A Job** button is pressed, as defined in HTML's submit statement constructed from the PHP echo command in Listing 1:

```
echo "<TD><input type=\"Submit\" name=\"btnNewJob\" value=\"Add a Job\"><BR></TD>";
```

Redirection to the "new job input" form is obtained from the statement

```
echo "<META HTTP-EQUIV=\"refresh\" CONTENT=\"0;URL=$ROOT_PATH$lnkSHOP_NEW_TASK\">";
```

where the constants `$ROOT_PATH` and `$lnkSHOP_NEW_TASK`, defined in the include file `shopconst.php3`, provide the path to the "new job input" form.

Listing 2 contains the PHP code that builds the screen to list events for a particular job. This code contains a large number of SQL SELECT statements to query the table `tblShopEvents` and returns various event records for the selected job. The selected job is identified by the variable `$JobNum` that is passed with the URL as a parameter to this page. An internal PHP array (`$rsEvents`) holds the results of each query to provide a mechanism to sort all events by date using the PHP statement `rsort($rsEvents)`.

Listing 3 is the PHP code that generates the "new job" input form and then validates the entered data. This code is reentrant and runs a second time after the user presses the **Enter info into database** button, which sets `TRUE` the variable `$btnAddNewTask`. The values of fields that are input by the user are retained when the code runs a second time by the PHP line that builds hidden variable HTML statements:

```
echo "<input type=hidden name=\"JobDescription\" value=\"$JobDescription\">";
```

Upon detecting that the variable `$btnAddNewTask` is `TRUE`, the user input fields are validated and then a new record is added to the `tblShopJobs` table.

Listing 4 holds the PHP code for generating and validating data entered on the "new event" input form. This code uses the same principles as the "new job" input form of Listing 3 because it displays the input form when first executed and then validates the entered data and adds a new record to the `tblShopEvents` table in the database.

The include file, Listing 5, contains common code and constants used by the other PHP files.

Listing 1: PHP code for main input form (Jobs1.php3)

```
<HTML>

<?php
/*
Version 1.0  -- 2002-10-31 jee Task Log for machine shop
                2002-11-11 jee more includes
                2002-11-13 jee removed task description from param list when calling
                               events form.
                2002-11-14 jee more work
*/

$version = "1.0";

echo "<HEAD>";
echo "<TITLE>Band 6 Effort</TITLE>";
echo "</HEAD>";
echo "<body>";

// can't get the restricted include file to work'
require '../database.php3';
require '../shopconst.php3';

// define some constants for refilling this page
define("SORT_DATE_ENTERED",0,TRUE);
define("SORT_NAME_CUSTOMER",1,TRUE);
define("SORT_JOBS_COMPLETE",2,TRUE);
define("SORT_JOBS_INCOMPLETE",3,TRUE);

// setup the action for the post method of this form
echo "<form method=\"post\" action=\"$ThisCode\">";

// Turn off all error reporting
error_reporting(0);

$db = mysql_connect($DatabaseServer, $UserID, $Password)
      or die("Connection to MySQL database on server \"$DatabaseServer\" failed!<P>Program ends.");

// report all errors except notices
error_reporting(E_ALL ^ E_NOTICE);

mysql_select_db("dbCDL",$db);


if ($btnNewJob)
{
        // show the new job form
        echo "<META HTTP-EQUIV=\"refresh\" CONTENT=\"0;URL=$ROOT_PATH$lnkSHOP_NEW_TASK\">";
}
elseif (!$sortdef OR $sortdef == SORT_JOBS_INCOMPLETE)
{
```

```
        $result = mysql_query("SELECT DISTINCT keyShopJobs, Task, Customer, DATE_FORMAT(DateAssigned, '%Y-%m-%d %H:%i:%s') AS DateAssigned,
DATE_FORMAT(DateRequired, '%Y-%m-%d %H:%i:%s') AS DateRequired, Notes FROM tblShopJobs WHERE (DateCompleted IS NULL) ORDER BY DateAssigned
DESC",$db);
        $JobDescription = "Incomplete Shop Jobs";
        $LinkDescription = "Show completed jobs";
        $SortType = SORT_JOBS_COMPLETE;
}
elseif ($sortdef == SORT_JOBS_COMPLETE)
{
        // show just the completed jobs
        $result = mysql_query("SELECT DISTINCT keyShopJobs, Task, Customer, DATE_FORMAT(DateAssigned, '%Y-%m-%d %H:%i:%s') AS DateAssigned,
DATE_FORMAT(DateRequired, '%Y-%m-%d %H:%i:%s') AS DateRequired, Notes FROM tblShopJobs WHERE (DateCompleted IS NOT NULL) ORDER BY DateAssigned
DESC",$db);
        $JobDescription = "Completed Shop Jobs";
        $LinkDescription = "Show incomplete jobs";
        $SortType = SORT_JOBS_INCOMPLETE;
}
else
{
        echo "Program error in $ThisCode: Fell through 'new job' and 'sort def' if statement!";
}


echo "<H2>$JobDescription</H2>";

// put the buttons in a hidden table
echo ("<TABLE BORDER=0 CELLPADDING = 10 BGCOLOR=WHITE>");
echo "<TR></TR>";
echo "<TR ALIGN=center>";
echo "<TR ALIGN=center><TD><B>Sorted by Date Entered</B><BR>(most recent first)</TD>";
echo "<TD><A HREF=\"$ThisCode?sortdef=$SortType\">$LinkDescription</A></TD>";
// button to add another job
echo "<TD><input type=\"Submit\" name=\"btnNewJob\" value=\"Add a Job\"><BR></TD>";
echo "</TABLE>";

echo "(Click on a job number to view or add details for that job)";
echo "<P>";

// show the jobs
echo ("<TABLE BORDER=3 CELLPADDING = 5 WIDTH=\"95%\" BGCOLOR=WHITE>");
echo ("<TR ALIGN=center><TH><B>Job Number</B><TD><B>Description</B></TH><TD><B>Customer</B><TD><B>Date Entered /<BR>Date Required</B><TD
WIDTH=\"45%\"><B>Notes</B></TH>");

while ($myrow = mysql_fetch_array($result))
{
  printf("<TR><TH><A href=\"%s?JobNum=%s\">%s</A><TD>%s<TD ALIGN=center>%s<TD><FONT SIZE=2>%s<BR>%s</FONT><TD>%s", $lnkSHOP_EVENTS,
$myrow["keyShopJobs"], $myrow["keyShopJobs"], $myrow["Task"], $myrow["Customer"],  $myrow["DateAssigned"], $myrow["DateRequired"],
$myrow["Notes"]);
}
echo ("</TABLE>");

echo "<P>Software Vers $version <br>";

echo "</form>";
```

```
echo "</body>";
?>
</html>
```

Listing 2: PHP code for events listing form (Events2.php3)

```
<HTML>

<?php
/*
Version 1.0  -- 2002-11-01 Job events screen
Version 2.0  -- 2002-11-01 Includes dates fields
                                        -- 2002-11-12 Continued development
*/
$version = "2.1";

echo "<HEAD>";
echo "<TITLE>Shop Details</TITLE>";
echo "</HEAD>";
echo "<body>";

// can't get the restricted include file to work'
require '../database.inc';
require '../shopconst.php3';

if ($JobNum == false)
{
        echo "Error in routine Events Version $version:<BR>";
        echo "No task number passed to this routine!";
}

// setup the action for the post method of this form
echo "<form method=\"post\" action=\"$lnkSHOP_NEW_EVENT?JobNum=$JobNum\">";

// Turn off all error reporting
error_reporting(0);

$db = mysql_connect($DatabaseServer, $UserID, $Password)
      or die("Connection to MySQL database on server \"$DatabaseServer\" failed!<P>Program ends.");

// report all errors except notices
error_reporting(E_ALL ^ E_NOTICE);

// select the appropriate database
mysql_select_db("dbCDL",$db);

// repeat a number of queries to determine what information was entered for the job
// Notes comment
$result = mysql_query("SELECT DATE_FORMAT(DateUpdated, '%Y-%m-%d %H:%i:%s') AS Date, EntryBy, Notes FROM tblShopEvents WHERE fkShopJobs =
$JobNum AND AssignedTo IS NULL AND DateDwgsRec IS NULL AND DateDwgsChecked IS NULL AND DwgsCheckedBy IS NULL AND DateEstCompletion IS NULL AND
DateCompleted IS NULL AND Effort IS NULL",$db);
if ($result != false)
{
        while ($myrow = mysql_fetch_array($result))
        {
                $rsEvents[] = array
```

```
          (
                        'date' => $myrow["Date"],
                        'item' => "<B>Notes</B> from " . $myrow["EntryBy"],
                        'notes' => $myrow["Notes"]
          );
        }
}

// job assignment
$result = mysql_query("SELECT DATE_FORMAT(DateUpdated, '%Y-%m-%d %H:%i:%s') AS Date, EntryBy, AssignedTo, Notes FROM tblShopEvents WHERE
fkShopJobs = $JobNum AND AssignedTo IS NOT NULL",$db);
if ($result != false)
{
        while ($myrow = mysql_fetch_array($result))
        {
                $rsEvents[] = array
          (
                        'date' => $myrow["Date"],
                        'item' => "Job <B>assigned to " . $myrow["AssignedTo"] . "</B> by  " . $myrow["EntryBy"],
                        'notes' => $myrow["Notes"]
          );
        }
}

// drawings received
$result = mysql_query("SELECT EntryBy, DATE_FORMAT(DateDwgsRec, '%Y-%m-%d %H:%i:%s') AS DateDwgsRecd, Notes FROM tblShopEvents WHERE fkShopJobs
= '$JobNum' AND DateDwgsRec IS NOT NULL",$db);

if ($result != false)
{
        while ($myrow = mysql_fetch_array($result))
        {
                $rsEvents[] = array
          (
                        'date' => $myrow["DateDwgsRecd"],
                        'item' => $myrow["EntryBy"] . ": <B>received dwgs</B>.",
                        'notes' => $myrow["Notes"]
          );
        }
}
// drawings checked
$result = mysql_query("SELECT DATE_FORMAT(DateDwgsChecked, '%Y-%m-%d %H:%i:%s') AS DateChecked, DwgsCheckedBy, Notes FROM tblShopEvents WHERE
fkShopJobs = $JobNum AND DwgsCheckedBy IS NOT NULL",$db);
if ($result != false)
{
        while ($myrow = mysql_fetch_array($result))
        {
                $rsEvents[] = array
          (
                        'date' => $myrow["DateChecked"],
                        'item' => $myrow["DwgsCheckedBy"] .": <B>checked dwgs.</B>",
                        'notes' => $myrow["Notes"]
          );
        }
```

```
}

// Effort entered
$result = mysql_query("SELECT DATE_FORMAT(DateUpdated, '%Y-%m-%d %H:%i:%s') AS DateUpdated, EntryBy, Effort, Notes FROM tblShopEvents WHERE
fkShopJobs = $JobNum AND Effort IS NOT NULL",$db);
if ($result != false)
{
        while ($myrow = mysql_fetch_array($result))
        {
                $rsEvents[] = array
          (
                        'date' => $myrow["DateUpdated"],
                        'item' => $myrow["EntryBy"] .": <B>" . $myrow["Effort"]. " hrs of effort.</B>",
                        'notes' => $myrow["Notes"]
          );
        }
}

// estimated completion date
$result = mysql_query("SELECT DATE_FORMAT(DateUpdated, '%Y-%m-%d %H:%i:%s') AS DateUpdated, DATE_FORMAT(DateEstCompletion, '%Y-%m-%d %H:%i:%s')
AS DateToComplete, EntryBy, Notes FROM tblShopEvents WHERE fkShopJobs = '$JobNum' AND DateEstCompletion IS NOT NULL",$db);
if ($result != false)
{
        while ($myrow = mysql_fetch_array($result))
        {
    $rsEvents[] = array
          (
                            'date' => $myrow["DateUpdated"],
                            'item' => $myrow["EntryBy"] . ": job should <B>complete on " . $myrow["DateToComplete"] . "</B>.",
                            'notes' => $myrow["Notes"]
          );
        }
}

// Date Completed
$result = mysql_query("SELECT DISTINCT DATE_FORMAT(DateCompleted, '%Y-%m-%d %H:%i:%s') AS DateComplete, EntryBy, Notes FROM tblShopEvents WHERE
fkShopJobs = $JobNum AND DateCompleted IS NOT NULL",$db);
if ($result != false)
{
        while ($myrow = mysql_fetch_array($result))
        {
                $rsEvents[] = array
                (
                        'date' => $myrow["DateComplete"],
                        'item' => "<B>Completed</B> (entered by " . $myrow["EntryBy"] . ")",
                        'notes' => $myrow["Notes"]
                );
        }
}
        // initial heading for page
        echo ("<TABLE BORDER=0 CELLPADDING = 5 WIDTH=\"95%\" BGCOLOR=WHITE>");
        echo ("<TD><H2>Event listing for shop job</TD>");
        echo "<TD><input type=\"Submit\" name=\"new\" value=\"Add an event\"></TD>";       // Build the button to add another event
        echo "<TD><A HREF=\"$lnkSHOP_MAIN\"><H2>Return to main form</H2></A></TD>";               // hypertext to allow return to main form
```

```
        echo "</TABLE><BR>";
        echo "<BR>";

        // generate the header information from the task number
        echo ("<TABLE BORDER=3 CELLPADDING = 5 WIDTH=\"95%\" BGCOLOR=WHITE>");
        echo ("<TR ALIGN=center><TH><B>Job Number</B><TD><B>Description</B></TH><TD><B>Customer</B><TD><B>Date Entered /<BR>Date Required</B><TD
WIDTH=\"45%\"><B>Notes</B></TH>");
        $result = mysql_query("SELECT Task, Customer, DATE_FORMAT(DateAssigned, '%Y-%m-%d %H:%i:%s') AS DateAssigned, DATE_FORMAT(DateRequired,
'%Y-%m-%d %H:%i:%s') AS DateRequired, Notes FROM tblShopJobs WHERE keyShopJobs = $JobNum",$db);
        $myrow = mysql_fetch_array($result);
  printf("<TR><TH>%s<TD>%s<TD ALIGN=center>%s<TD><FONT SIZE=2>%s<BR>%s</FONT><TD>%s", $JobNum, $myrow["Task"], $myrow["Customer"],
$myrow["DateAssigned"], $myrow["DateRequired"], $myrow["Notes"]);
        echo ("</TABLE>");


if (count($rsEvents) >= 1)
{
        // sort the output table by date with most recent at beginning
        rsort($rsEvents);

        echo "<P>";
        echo "<DIV ALIGN = \"CENTER\"><H2>Events:</H2></DIV><BR><BR>";

        // build the events table
        echo ("<TABLE BORDER=0 CELLPADDING = 5 WIDTH=\"95%\" BGCOLOR=WHITE>");
        echo ("<TR ALIGN=center><TH><B>Date Item Entered</B><TD><B>Item</B><TD><TD WIDTH=\"45%\"><B>Notes</B></TH>");

        // print out the table rows
        foreach ($rsEvents as $rsOut)
        {
          printf("<TR><TH><FONT SIZE=2>%s</FONT><TD>%s<TD><TD>%s", $rsOut['date'], $rsOut['item'], $rsOut['notes']);
        }
        echo ("</TABLE>");
}
else
{
        printf("<H2>No events found for this job!</H2>");
        echo "<P>";
}

echo "<P>Software Vers $version<br>";
echo "</form>";
echo "</body>";
?>
</html>
```

Listing 3: PHP code for entering new shop jobs (NewJob1.php3)

```
<HTML>

<?php
// Form to enter new job information
// Version 1.0  -- 2002-10-31 jee
// TODO -- CHECK TIME FIELDS FOR ERRORS PRIOR TO UPDATING DATABASE TABLE.

$version = "1.0";

echo "<HEAD>";
echo "<TITLE>Shop Task Management Program</TITLE>";
echo "</HEAD>";
echo "<body>";

// can't get the restricted include file to work'
require '../database.php3';
require '../shopconst.php3';

$LEN_OF_JOB_DESCRIPTION_FIELD = 50;

// Turn off all error reporting
error_reporting(0);

$db = mysql_connect($DatabaseServer, $UserID, $Password)
      or die("Connection to MySQL database on server \"$DatabaseServer\" failed!<P>Program ends.");

// report all errors except notices
error_reporting(E_ALL ^ E_NOTICE);

mysql_select_db("dbCDL",$db);

if ($btnAddNewTask)
{
        // this runs after the code below, and this block checks the fields and then
        // adds a new task to the database
        if (strlen($CustomerName) == 0)
        {
    echo "<H2>Field containing your name must be filled in.</H2><br>";
                echo "Use your browser's Back button to correct<br>";
                exit;
        }
        if (strlen($JobDescription) == 0)
        {
    echo "<H2>Job description field must be filled in.</H2><br>";
                echo "Use your browser's Back button to correct<br>";
                exit;
        }
        elseif (strlen($JobDescription) > $LEN_OF_JOB_DESCRIPTION_FIELD)
        {
                echo "<H2>Job Description field must be less than $LEN_OF_JOB_DESCRIPTION_FIELD characters.</H2><br>";
                echo "Use your browser's Back button to correct<br>";
```

```
                exit;
        }
        if (strlen($DateEntered) == 0)
        {
                // Generate the task date
                $DateEntered = date("Y-m-d H:i:s");
        }
        elseif (ConvertDate($DateEntered) == -1)
        {
                echo "<H2>Bad date format for 'Date Entered' ($DateEntered)</H2><BR>";
                echo constant("DATE_ERROR_STRING");
                echo "<BR>Use your browser's Back button to correct<br>";
                exit;
        }
        if (strlen($DateRequired) == 0)
        {
                echo "<H2>Required date field is blank</H2><BR>";
                echo "Use your browser's Back button to correct<br>";
                exit;
        }
        elseif (ConvertDate($DateRequired) == -1)
        {
                echo "<H2>Bad date format for 'Date Required' ($DateRequired)</H2><BR>";
                echo constant("DATE_ERROR_STRING");
                echo "<BR>Use your browser's Back button to correct<br>";
                exit;
        }
  $sql = "INSERT INTO tblShopJobs SET DateAssigned='$DateEntered', DateRequired='$DateRequired', ";
  $sql .= "Task='$JobDescription', Customer='$CustomerName', Notes='$Notes'";

        // run SQL against the DB
  $result = mysql_query($sql);
  echo (mysql_affected_rows() ? "<P>Database successfully updated!" : "<P>Database update failed!");

        //reload the first page
        echo "<P>Task table will reload in 2 seconds...<P> TO <BR>$ROOT_PATH$lnkSHOP_MAIN";
        echo "<META http-equiv=\"refresh\" CONTENT=\"2; URL=$ROOT_PATH$lnkSHOP_MAIN\">";
}
else
{
        // user has pressed a Add New Task button from task status listing page, so present new status input page

        // hide the following variables, which are needed by the input form the second time through this code
        echo "<input type=hidden name=\"JobDescription\" value=\"$JobDescription\">";
        echo "<input type=hidden name=\"DateEntered\" value=\"$DateEntered\">";
        echo "<input type=hidden name=\"DateRequired\" value=\"$DateRequired\">";
        echo "<input type=hidden name=\"CustomerName\" value=\"$CustomerName\">";
        echo "<input type=hidden name=\"TaskNotes\" value=\"$Notes\">";

        // setup the action for the post method of this form
        echo "<form method=\"post\" action=\"$PHP_SELF\">";

        echo "<H2>Enter new job for shop </H2><P>";
   echo "Software Vers $version <br>";
```

```
   $DateEntered = date("Y-m-d H:i:s");

        // add four weeks to the current date
  $DateRequired = AddToDate(0,28,0);
  // now build the input boxes using an invisible table
  echo ("<TABLE BORDER=0 WIDTH=60% BGCOLOR=WHITE>");
  echo "<TR><TD>Job Description:<TD><input SIZE = 35 type=\"Text\" name=\"JobDescription\" value=\"$JobDescription\"</TR>";
  echo "<TR><TD>Date Entered:<TD><input SIZE = 35 type=\"Text\" name=\"DateEntered\" value=\"$DateEntered\"</TR>";
  echo "<TR><TD>Your Name:<TD><input SIZE = 35 type=\"Text\" name= \"CustomerName\" value=\"$CustomerName\"</TR>";
  echo "<TR><TD>Date Required:<TD><input SIZE = 35 type=\"Text\" name=\"DateRequired\" value=\"$DateRequired\"</TR>";
  echo "<TR><TD>Notes:<TD><br><textarea name=\"Notes\" rows=\"5\" cols=\"30\"> $Notes  </textarea></TR>";

        // Build the submit button in the second table column
  echo "<TR><TD><TD><br><input type=\"Submit\" name=\"btnAddNewTask\" value=\"Enter info into database.\"><br>";
  echo ("</TABLE>");
}
echo "</form>";
echo "</body>";
?>
</html>
```

Listing 4: PHP code for entering new events (NewJob1.php3)

```
<HTML>

<?php
/*
        Form to enter new event information for Shop tracking program
        Version 1.0  -- 2002-11-01 jee
        TODO -- CHECK TIME FIELDS FOR ERRORS PRIOR TO UPDATING DATABASE TABLE.
        2002-11-13 jee
*/

$version = "1.0";

echo "<HEAD>";
echo "<TITLE>Shop Task Management Program</TITLE>";
echo "</HEAD>";
echo "<body>";

// can't get the restricted include file to work'
require '../database.php3';
require '../shopconst.php3';

// private constants for this module
$LEN_OF_TASK_DESCRIPTION_FIELD = 50;
$RB_ONE_DAY = 1;
$RB_TWO_DAYS = 2;
$RB_ONE_WEEK = 7;
$RB_FOUR_WEEKS = 28;
$RB_OTHER = -1;

// Turn off all error reporting
error_reporting(0);

$db = mysql_connect($DatabaseServer, $UserID, $Password)
      or die("Connection to MySQL database on server \"$DatabaseServer\" failed!<P>Program ends.");

// report all errors except notices
error_reporting(E_ALL ^ E_NOTICE);

mysql_select_db("dbCDL",$db);

// setup the action for the post method of this form
//echo "<form method=\"post\" action=\"$btnAddNewEvent\">";

if ($btnAddNewEvent)
{
        // this block runs after the code below.
        // it checks the fields and then adds a new task to the database
        if (strlen($EntryBy) == 0)
        {
    echo "<H2>Field containing your name must be filled in.</H2><br>";
```

```
        echo "Use your browser's Back button to correct<br>";
        exit;
}
else
{
        // Generate the date updated
        $DateUpdated = date("Y-m-d H:i:s");

        // and generate the default SQL string
  $sql = "INSERT INTO tblShopEvents SET fkShopJobs='$JobNum', DateUpdated='$DateUpdated', EntryBy='$EntryBy', Notes='$Notes'";
}

if ($chkbDwgsRec == 'on')
{
        if (ConvertDate($DateReceived) == -1)
        {
                echo "<H2>The format for \"Date received\" ($DateReceived) is invalid.</H2><BR>";
                echo constant("DATE_ERROR_STRING");
                echo "Use your browser's Back button to correct<BR>";
                exit;
        }
        $sql .= ", DateDwgsRec='$DateReceived'";
}

if ($chkbDwgsChecked == 'on')
{
        if (ConvertDate($DateDwgsChecked) == -1)
        {
                echo "<H2>The format for \"Date drawings checked\" ($DateDwgsChecked) is invalid.</H2><BR>";
                echo constant("DATE_ERROR_STRING");
                echo "Use your browser's Back button to correct<br>";
                exit;
        }
        $sql .= ", DwgsCheckedBy='$DwgsCheckedBy', DateDwgsChecked='$DateDwgsChecked'";
}

if (strlen($JobAssignedTo) > 0)
{
        $sql .= ", AssignedTo='$JobAssignedTo'";
}

if (strlen($Effort) > 0)
{
        $sql .= ", Effort=$Effort";
}

if (!is_null($EstDate))
{
        // estimated completion date is filled in
        if ($EstDate > 0 )
        {
                // need to add an amount to the current date
                $DateEstCompletion = AddToDate(0,$EstDate,0);
        }
```

```
                else
                {
                        // estimated date is from input field
                        $DateEstCompletion= $DateEstCompletion;
                }
                $sql .= ", DateEstCompletion='$DateEstCompletion'";
        }

        if ($chkbJobComplete == 'on')
        {
                if (ConvertDate($DateCompleted) == -1)
                {
                        echo "<H2>The format for \"Date Completed\" ($DateCompleted) is invalid.</H2><BR>";
                        echo constant("DATE_ERROR_STRING");
                        echo "Use your browser's Back button to correct<br>";
                        exit;
                }
                // update two tables: tblShopJobs and tblEvents
                $sqlShopJob = "UPDATE tblShopJobs SET DateCompleted='$DateCompleted' WHERE keyShopJobs = $JobNum";
                $sql .= ", DateCompleted='$DateCompleted'";

                // run SQL against the DB for tblShopJobs
                // (tblShopEvents will be updated later)

        $result = mysql_query($sqlShopJob);
        if (mysql_affected_rows() < 1)
                {
                        echo "<P>Database update failed!";
                        echo "<BR><BR> SQL statement is: <BR><BR> $sqlShopJob";
                }
        }

        // run SQL against the DB
  $result = mysql_query($sql);
  if (mysql_affected_rows() < 1)
        {
                echo "<P>Database update failed!";
                echo "<BR><BR> SQL statement is: <BR><BR> $sql";
        }
        else
        {
                echo "<P>Database successfully updated!";
        }

        //reload the first page
        echo "<P>Task table will reload in 2 seconds...<P> TO <BR>$ROOT_PATH$lnkSHOP_EVENTS";
        echo "<META http-equiv=\"refresh\" CONTENT=\"2; URL=$ROOT_PATH$lnkSHOP_EVENTS?JobNum=$JobNum\">";
}
else
{
        // this code runs the first time through this form
        // setup the action for the post method of this form
        ECHO "<FORM METHOD=\"POST\" ACTION=\"$ThisCode?$JobNum\">";
```

```
        // hide the following variables, which are needed by the input form the second time through this code
        echo "<input type=hidden name=\"chkbJobComplete\" value=\"$chkbJobComplete\">";
        echo "<input type=hidden name=\"chkbDwgsRec\" value=\"$chkbDwgsRec\">";
        echo "<input type=hidden name=\"chkbDwgsChecked\" value=\"$chkbDwgsChecked\">";
        echo "<input type=hidden name=\"DateReceived\" value=\"$DateReceived\">";
        echo "<input type=hidden name=\"DateUpdated\" value=\"$DateUpdated\">";
        echo "<input type=hidden name=\"DwgsCheckedBy\" value=\"$DwgsCheckedBy\">";
        echo "<input type=hidden name=\"DateDwgsChecked\" value=\"$DateDwgsChecked\">";
        echo "<input type=hidden name=\"Customer\" value=\"$EntryBy\">";
        echo "<input type=hidden name=\"Effort\" value=\"$Effort\">";
        echo "<input type=hidden name=\"DateEstCompletion\" value=\"$DateEstCompletion\">";
        echo "<input type=hidden name=\"DateCompleted\" value=\"$DateCompleted\">";
        echo "<input type=hidden name=\"TaskNotes\" value=\"$Notes\">";
        echo "<input type=hidden name=\"JobNum\" value=\"$JobNum\">";


        // initial heading for page
        echo ("<TABLE BORDER=0 CELLPADDING = 5 WIDTH=\"95%\" BGCOLOR=WHITE>");
        echo ("<TD><H2>Enter new event for shop job</TD>");
        echo ("<TD><input type=\"Submit\" name=\"btnAddNewEvent\" value=\"Add event to database\"><BR>");
        echo "<TD><A HREF=\"$lnkSHOP_MAIN\"><H2>Return to main form</H2></A></TD>";               // hypertext to allow return to main form
        echo "</TABLE><BR>";
        echo "<BR>";
        // generate the header information from the job number
        echo ("<TABLE BORDER=3 CELLPADDING = 5 WIDTH=\"95%\" BGCOLOR=WHITE>");
        echo ("<TR ALIGN=center><TH><B>Job Number</B><TD><B>Description</B></TH><TD><B>Customer</B><TD><B>Date Entered /<BR>Date Required</B><TD
WIDTH=\"45%\"><B>Notes</B></TH>");
        $result = mysql_query("SELECT Task, Customer, DATE_FORMAT(DateAssigned, '%Y-%m-%d %H:%i:%s') AS DateAssigned, DATE_FORMAT(DateRequired,
'%Y-%m-%d %H:%i:%s') AS DateRequired, Notes FROM tblShopJobs WHERE keyShopJobs = $JobNum",$db);
        $myrow = mysql_fetch_array($result);
  printf("<TR><TH>%s<TD>%s<TD ALIGN=center>%s<TD><FONT SIZE=2>%s<BR>%s</FONT><TD>%s", $JobNum, $myrow["Task"], $myrow["Customer"],
$myrow["DateAssigned"], $myrow["DateRequired"], $myrow["Notes"]);
        echo ("</TABLE><BR><BR>");

  $DateReceived = date("Y-m-d H:i:s");
        $DateCompleted = date("Y-m-d H:i:s");
        $DateDwgsChecked = date("Y-m-d H:i:s");
        $DwgsCheckedBy = "Tony Marshall";


        // add 28 days to the present day
        $DateRequired = AddToDate(0,28,0);
        $DateEstCompletion = $DateRequired;

  // now build the input boxes using an invisible table
        // spacing between lables and input cells is obtained from blank column
        $SPACING = "2%";
  echo ("<TABLE BORDER=0 WIDTH=60% BGCOLOR=WHITE>");
  echo "<TR><TD ALIGN=RIGHT>Your name<TD><TD SIZE=$SPACING><input SIZE = 35 type=\"Text\" name=\"EntryBy\" value=\"$EntryBy\"</TR>";
  echo "<TR><TD ALIGN=RIGHT><INPUT TYPE=\"CHECKBOX\" NAME=\"chkbDwgsRec\">Drawings received on <TD SIZE=$SPACING><TD><input SIZE = 35
type=\"Text\" NAME=\"DateReceived\" value=\"$DateReceived\"</TR>";
  echo "<TR><TD ALIGN=RIGHT><INPUT TYPE=\"CHECKBOX\" NAME=\"chkbDwgsChecked\">Drawings checked by  <TD SIZE=$SPACING><TD><input SIZE = 35
type=\"Text\" name= \"DwgsCheckedBy\" value=\"$DwgsCheckedBy\"</TR>";
```

```
  echo "<TR><TD ALIGN=RIGHT>Drawings checked on  <TD SIZE=$SPACING><TD><input SIZE = 35 type=\"Text\" name= \"DateDwgsChecked\"
value=\"$DateDwgsChecked\"</TR>";
  echo "<TR><TD ALIGN=RIGHT>Job assigned to<TD><TD SIZE=$SPACING><input SIZE = 35 type=\"Text\" name= \"JobAssignedTo\"
value=\"$JobAssignedTo\"</TR>";
  echo "<TR><TD ALIGN=RIGHT>Hours worked <TD SIZE=$SPACING><TD><input SIZE = 35 type=\"Text\" name= \"Effort\" value=\"$Effort\"</TR>";
  echo "<TR><TD ALIGN=RIGHT>Estimated completion date<TD SIZE=$SPACING><TD><INPUT TYPE=\"RADIO\" NAME=\"EstDate\"
VALUE=\"$RB_ONE_DAY\">Tomorrow<BR><INPUT TYPE=\"RADIO\" NAME=\"EstDate\" VALUE=\"$RB_TWO_DAYS\">2 days<BR><INPUT TYPE=\"RADIO\" NAME=\"EstDate\"
VALUE=\"$RB_ONE_WEEK\">1 week<BR><INPUT TYPE=\"RADIO\" NAME=\"EstDate\" VALUE=\"$RB_FOUR_WEEKS\">4 weeks<BR><INPUT TYPE=\"RADIO\"
NAME=\"EstDate\" VALUE=\"$RB_OTHER\">Other: <input SIZE = 26 type=\"Text\" name=\"DateEstCompletion\" value=\"$DateEstCompletion\"</TR>";
        echo "<TR><TD ALIGN=RIGHT><INPUT TYPE=\"CHECKBOX\" NAME=\"chkbJobComplete\" >Job complete on<TD SIZE=$SPACING><TD><input SIZE=35
type=\"Text\" name=\"DateCompleted\" value=\"$DateCompleted\"</TR>";
        echo "<TR><TD ALIGN=RIGHT>Notes:<TD SIZE=$SPACING><TD><br><textarea name=\"Notes\" rows=\"5\" cols=\"30\"> $Notes  </textarea></TR>";
  echo ("</TABLE>");

        echo "<BR><BR>Software Vers $version <BR><BR>";
}
echo "</form>";
echo "</body>";

?>
</html>
```

Listing 5: Include file for PHP code (shopconst.php3)

```
<?php
// Include file for the shop job software.
//
// 2002-11-11 jee
// 2002-11-14 jee added $lnkSHOP_MAIN
//
//require 'database.php3';
// used to call the top level of this form again
$ROOT_PATH = "http://www.cv.nrao.edu";
// http://www.cv.nrao.edu/~jeffland/nrao-only/Progs/Shop/NewTask1.php3

// location of associated code
$lnkSHOP_EVENTS = dirname($PHP_SELF) . "/Events2.php3";
$lnkSHOP_NEW_TASK = dirname($PHP_SELF) . "/NewJob1.php3";
$lnkSHOP_DETAILS = dirname($PHP_SELF) . "/Details1.php3";
$lnkSHOP_NEW_EVENT = dirname($PHP_SELF) . "/NewEvent1.php3";
$lnkSHOP_MAIN = dirname($PHP_SELF) . "/Jobs1.php3";

define("DATE_ERROR_STRING", "Most date formats work (with or without the time), but <B>YYYY-MM-DD HH:MM:SS</B> is the ISO 8601
standard.<BR>Forms of <B>MM/DD/YYYY</B> also work, but <B>MM/DD/YY</B> works only for some years.<BR>The form MM-DD-YY converts to a completely
wrong year!");

$ThisCode = $PHP_SELF;
//*************************************************************************
//*************************************************************************
function AddToDate($pMonth, $pDay, $pYear)
{
        // returns a date as a string after adding the appropriate units to the present time
        //
        // 2002-11-12 jee
        //
        $timestamp =  time();
        $date_time_array =  getdate($timestamp);

        $month =  $date_time_array["mon"] + $pMonth;
        $day =  $date_time_array["mday"] + $pDay;
        $year =  $date_time_array["year"] + $pYear;

        // use mktime to recreate the unix timestamp
        // adding 4 weeks to $day and forcing the time to 5 PM
        $hours = 17;
        $minutes = 0;
        $seconds = 0;
        $timestamp =  mktime($hours, $minutes, $seconds , $month, $day, $year);
        return date("Y-m-d H:00:00", $timestamp);
}
//*************************************************************************
function ConvertDate($pDate)
{
        // checks a date and returns it as 'YYY-MM-DD HH:MM:SS'
```

```
        // Returns -1 if date can't be converted
        //
        // 2002-11-14 jee
        //

        $timestamp = strtotime($pDate);
        if ($timestamp == -1)
        {
                // bad format
                return -1;
        }
        return date("Y-m-d H:i:s", $timestamp);
}
?>
```