Memorandum

To:	K. Crady	G. Ediss
	N. Horner	A. R. Kerr
	D. Koller	G. Lauria
	S. -K. Pan	G. Petencin

cc: P. Murphy
J. Webber
C. Cunningham

From: J. Effland

Date: 31 July 2002

Subject: Proposed System to Track Effort Expended for ALMA Band 6 Cartridge Construction

1 Introduction

Tracking of the effort expended during construction of the Band 6 Cartridges is vitally important to provide timely warnings for cost overruns. It also has an added benefit of helping to maintain focus on those tasks funded by ALMA.

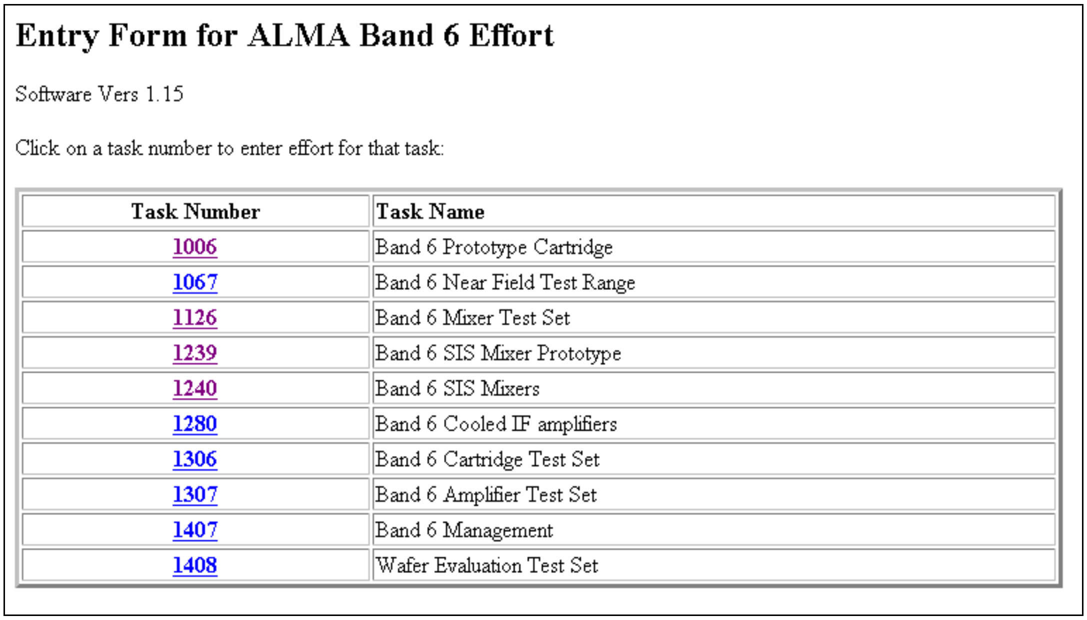
A simple but easy to use web-based system has been designed and is proposed to allow each person assigned to ALMA cartridge construction to input times expended on ALMA tasks. To be useful, everyone at the CDL working on ALMA cartridge construction would fill out this web form at least weekly and prior to the weekly coordination meetings. Progress and the time remaining to complete the tasks can then be reviewed at those meetings.

2 Web Forms

The initial web form, shown in Figure 1 is available at

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

and provides a list of all the tasks assigned to the Band 6 cartridge construction project. The contents of the form are generated from a database (see Appendix for details) so task names and numbers can easily be modified and enhanced.



Entry Form for ALMA Band 6 Effort

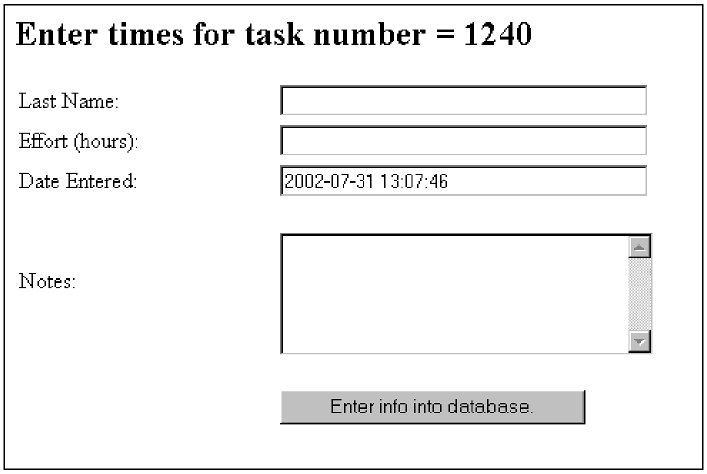
Software Vers 1.15

Click on a task number to enter effort for that task:

Task Number	Task Name
1006	Band 6 Prototype Cartridge
1067	Band 6 Near Field Test Range
1126	Band 6 Mixer Test Set
1239	Band 6 SIS Mixer Prototype
1240	Band 6 SIS Mixers
1280	Band 6 Cooled IF amplifiers
1306	Band 6 Cartridge Test Set
1307	Band 6 Amplifier Test Set
1407	Band 6 Management
1408	Wafer Evaluation Test Set

Figure 1: The initial form lists all assigned tasks for the Band 6 cartridge construction.

Clicking on one of the task number hyperlinks calls up the input form as shown in Figure 2. Here the user simply enters their name and the hours worked for the selected task. The `Date Entered` field, whose content is automatically filled from the server's system time, presently reports when the form is shown, but it could be used to time stamp when the effort occurred. The `notes` field provides a way of entering addition information. Pressing the button `Enter Info into Database` adds a record in the database for the selected task and returns the user to the initial input screen.



Enter times for task number = 1240

Last Name:	
Effort (hours):	
Date Entered:	2002-07-31 13:07:46
Notes:	

Figure 2: Clicking on the hyperlink "1240" in the task list brings up this input form

3 Future Enhancements

The web-based forms can be enhanced in many ways, including:

1) Provisions for entering effort for more detailed tasks using a hierarchical structure. A plus sign could be inserted next to each task number and when clicked would list subtasks under the desired tasks. Clicking on the subtask numbers would bring up the input form.

2) The use of cookies so users would only need to enter hours on the input form.
3) A drop down list of user names so each would be unique and consistent.
4) Enhanced security to:
   a) authenticate users to prevent others from entering data for someone else
   b) Hide the user ids and passwords in the PHP include file.

4 Appendix: Software and Database Design

4.1 Design Philosophy

A database serves to hold the task and effort information. Records in the database are displayed using web pages and updated with web based input forms. This allows the task information to be changed by simply adding or modifying records in the database. Also, reports of effort expended for each task can be created using a variety of commercial off-the-shelf database packages, including Microsoft Access and Excel.

4.2 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 amenities of my preferred database, Microsoft's SQL Server, but MySQL is the only database server available at the NRAO for web applications. In particular, MySQL lacks any facilities to graphically display database schemas, so the tables discussed below were generated manually. I use the program "MySQL-Front" (http://www.anse.de/mysqlfront/) to manage the database tables.

Two database tables, `Tasks` and `effort`, are used to store the data as described next.

Description of table `Tasks`		
Purpose	**Each record provides task information such the number and description of the task**	
Field Name	**Field Type**	**Comments**
`id`	`Int(3) Unsigned`	Auto-Incremented key field
`TaskNum`	`Varchar(30)`	Task number (*e.g.* 1006)
`TaskName`	`Varchar(30)`	Task description (*e.g.* Band 6 Prototype Cartridge)
`date`	`datetime`	Date that task was entered
`notes`	`text`	Allows for lengthy description of task

Description of table `effort`		
Purpose	**Each record provides information such as who and how much effort was expended for a particular task.**	
Field Name	**Field Type**	**Comments**
`TaskNum`	`Int(10) Unsigned`	Task number (*e.g.* 1006), that is linked to Tasks.TaskNum
`name`	`Varchar(30)`	Person's name who is expending the effort
`notes`	`text`	Allows for any notes about this particular record
`hours`	`float`	Effort expended in hours
`datetime`	`datetime`	Finish time when effort was expended

4.3 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 somewhat crude and has many undocumented features, but works well enough to be usable.

When a URL that points to a PHP file is clicked using a browser, the web server first executes the PHP statements in the file, then the HTML output from the PHP program is directed to the browser. If the HTML code is examined using a browser's source button, the PHP listing shown below is not visible, but rather just the HTML output from that program.

The software listing, shown in Listing 1, is different from conventional code because it executes again each time the browser generates a new page. PHP dynamically fills variables, so these can be tested to determine which page was the previous page to execute.

The production software is located at

```
http://www.cv.nrao.edu/~jeffland/nrao-only/Progs/EffortLog.php3
```

The software first connects to the database using parameters stored in an include file. This is supposed to provide a mechanism to hide database passwords and user ids, but the program won't work when the include file is restricted. Because all the code resides behind NRAO's firewall and hence is only accessible from the internal network, security is not too important (sorry Pat) but will be enhanced in the future.

The statement `if ($submit)` returns false until the user has entered a name and hours in the input data form. So when the URL is viewed for the first time with the browser, the script is directed to the associated `else` statement, which generates and displays a table containing task numbers and descriptions. Clicking on any of the task number hyperlinks runs the same form again because the variable `$PHP_SELF` contains the name of the presently displayed form *via* the PHP-generated HTML statement:

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

The second time the form is displayed, if the `$tasknum` variable holds a value, then it is known that one of the task number hyperlinks was selected, so the script generates the effort input form. After the `Enter Info into Database` button is pressed on the input form, the `$submit` variable contains a value so a new record is created with information from the input form.

Finally, the top-level form with the task table is shown again using the PHP-generated HTML statement:

```
echo "<META http-equiv=refresh CONTENT=\"2; $ThisCode\">";
```

```
<HTML>

<?php
// Version 1.14 -- 2002-07-30 jee changed from tasknum as key to task number.
// Version 1.15 -- 2002-07-31 jee reloads top page, further refinements.
$version = "1.15";

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

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

// used to call the top level of this form again
$ThisCode = $PHP_SELF;

// 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 ($submit)
{
  // only execute the next block if the submit button has been pressed.
  // here if no ID then adding else we're editing
       // echo "Submit pressed <br>";

       if (strlen($EffortName) > 0 and $EffortHours > 0)
       {
              if (strlen($EffortDate) == 0)
              {
                     $EffortDate = date("Y-m-d H:m:s");
              }


    $sql = "INSERT INTO effort SET TaskNum='$tasknum', name='$EffortName',
datetime='$EffortDate', hours='$EffortHours', notes='$EffortNotes'";

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

              //reload the first page
              echo "<P>Task table will reload in 2 seconds...<P>";
              echo "<META http-equiv=refresh CONTENT=\"2; $ThisCode\">";

       }
       else
  {
    echo "<H2>Name field must be filled in <br>and effort must have non-zero value.</H2><br>";
              echo "User your browser's BACK button to fix";
       }
}
else
{
  // this part happens the first time through this code
  // and'if we don't press submit or delete
```

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

  if ($tasknum)
  {
    $EffortDate = date("Y-m-d H:m:s");

    // keep a hidden type to hold the id for the next time through the form
    // then show the input fields
    echo "<input type=hidden name=\"tasknum\" value=\"$tasknum\">";

    // now build the input boxes using an invisible table
    echo "<H2>Enter times for task number = $tasknum</H2><P>";
    echo ("<TABLE BORDER=0 WIDTH=60% BGCOLOR=WHITE>");
    echo "<TR><TD>Last Name:<TD><input SIZE = 35 type=\"Text\" name=\"EffortName\"
value=\"$EffortName\"</TR>";
    echo "<TR><TD>Effort (hours):<TD><input SIZE = 35 type=\"Text\" name= \"EffortHours\"
value=\"$EffortHours\"</TR>";
    echo "<TR><TD>Date Entered:<TD><input SIZE = 35 type=\"Text\" name=\"EffortDate\"
value=\"$EffortDate\"</TR>";
    echo "<TR><TD>Notes:<TD><br><textarea name=\"EffortNotes\" rows=\"5\" cols=\"30\">
$EffortNotes  </textarea></TR>";
    // Build the submit button in the second table column
    echo "<TR><TD><TD><br><input type=\"Submit\" name=\"submit\" value=\"Enter info into
database.\"><br>";
    echo ("</TABLE>");
  }
       else
       {
    // no tasknum so show table of tasks
             echo "<H2>Entry Form for ALMA Band 6 Effort</H2><P>";
    echo "Software Vers $version <br>";
    //echo date("Y-m-d H:m:s");
    echo "<P>";
    echo "Click on a task number to enter effort for that task:";
    echo "<P>";
    echo ("<TABLE BORDER=3 WIDTH=60% BGCOLOR=WHITE>");
    echo ("<TR><TH><B>Task Number</B><TD><B>Task Name</B></TH>");
    // print the list if there is no editing
    $result = mysql_query("SELECT * FROM Tasks ORDER BY tasknum",$db);
    while ($myrow = mysql_fetch_array($result))
    {
        printf("<TR><TH><a href=\"%s?tasknum=%s\">%s<TD>%s", $PHP_SELF, $myrow["TaskNum"],
$myrow["TaskNum"], $myrow["TaskName"]);
    }
    ECHO ("</TABLE>");
       }
}
echo "</form>";
echo "</body>";
?>
</html>
```