

AccuBuild Mobile - Job Number Lookup Option Setting ENHANCEMENT

- The setting for the Job Number Lookup Option in the Mobile Daily Report Settings group box has been enhanced to work with 3 different lookup controls in the AccuBuild Mobile program; Jobs, Employees, and Equipment. The Label on this screen has been changed to identify the new enhancement.

Old screen:



Mobile Daily Report Settings

Equipment Screen Format: Screen Format 2

Job Number Lookup Option: Drop Down Lookup List

New screen:



Mobile Daily Report Settings

Equipment Screen Format: DBIncCombo1

Job/Emp/Equip Lookup: DBIncCombo2

Purchase Order Screen - Order module takes considerable time to load BUG FIX

- The recent changes that expanded the vendor name in the Purchase Order Master screen caused a speed issue with loading and closing the screen. This feature has been removed in order to speed up the load time of the screen and will be re-addressed once a more robust solution can be developed.

ACH Vendor Payment - Balanced File BUG FIX

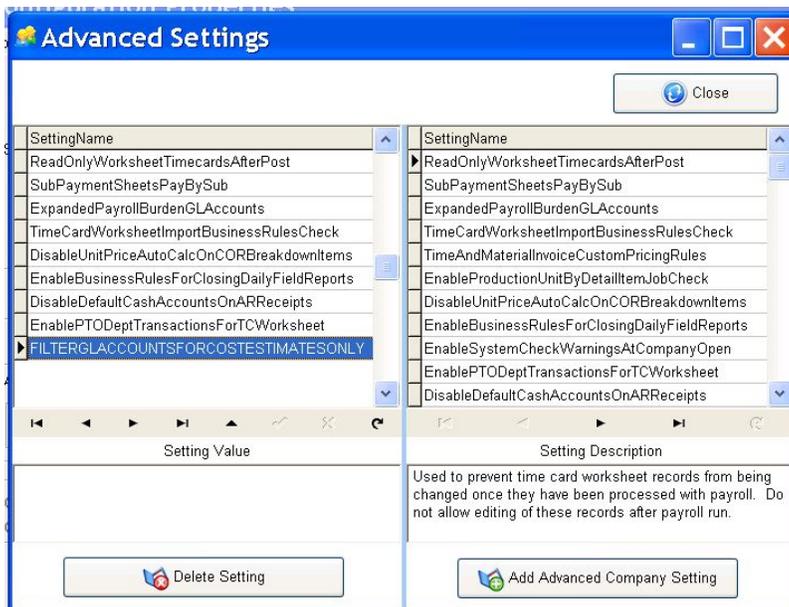
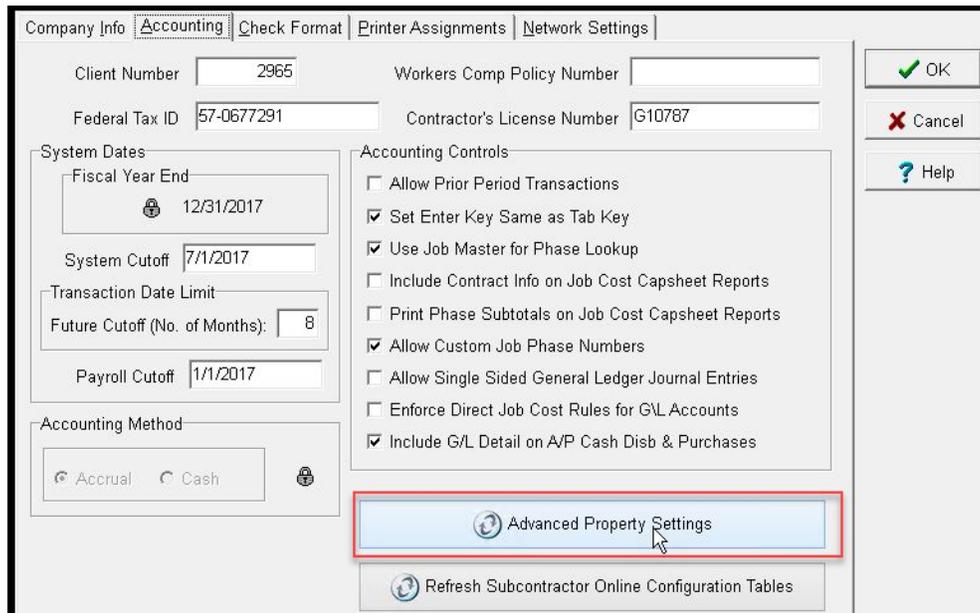
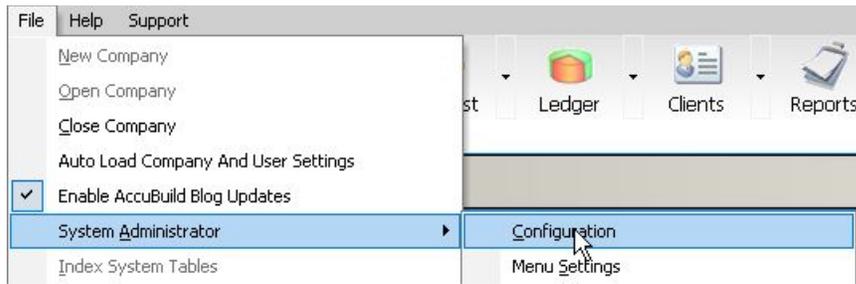
- The process for creating a balanced ACH File had an issue with the Credit Amount not being filled in on the batch control record - this issue has been corrected.

Doc Routing - PO Invoice Unit Price Rounding- BUG FIX

- On Purchase Order invoices that were entered via the invoice routing process, a rounding issue existed for items that had large quantities and fractions of cents on the item pricing. This issue was prohibiting the SAVE Button from being enabled after all of the entries were made. This issue has been corrected.

Accounts Payable - Enhancement to Prevent Coding to Job/ G/L Acct When There is No Estimate Record

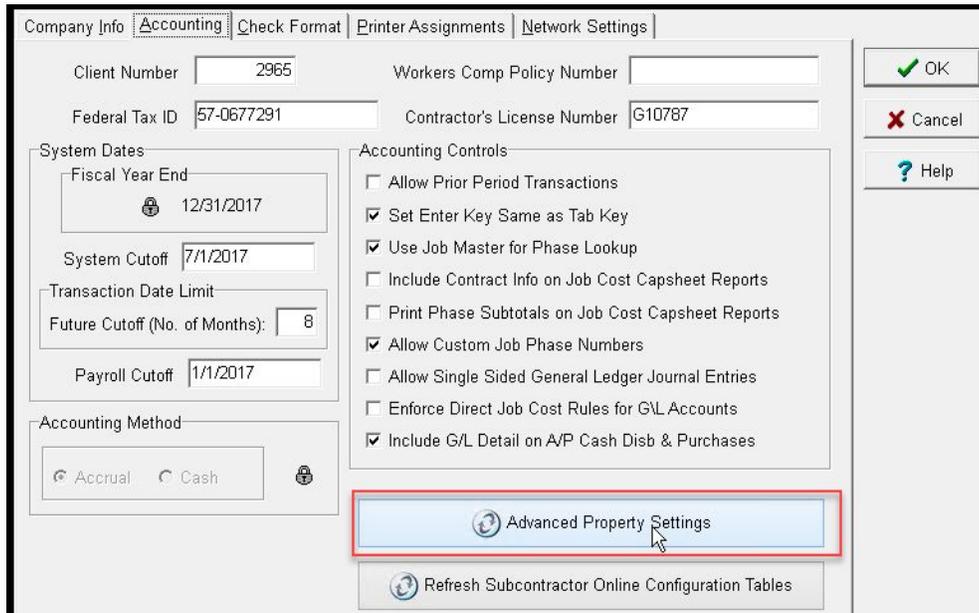
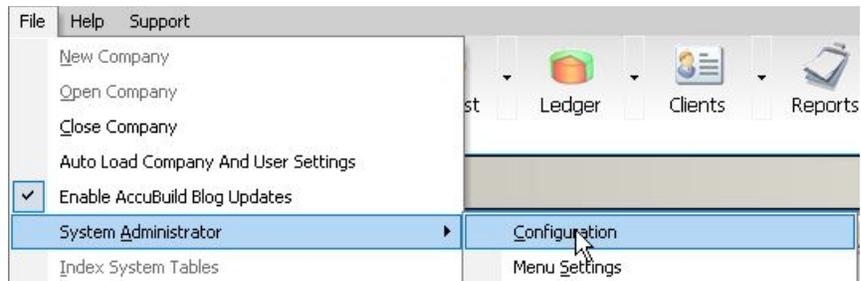
- Overview - A new Advanced setting option has been added entitled “**FILTERGLACCOUNTSFORCOSTESTIMATESONLY**”. Enabling this setting, limits the list of accounts in the G/L Account drop down for the accounts that are set up in the Estimate (ESTDETL table) for the corresponding job and phase.
- When this setting is enabled, it is applied to the following input screens:
 - Regular AP Invoices - non routing
 - Regular AP Invoices - routing
 - Purchase Order Creation - PO Items - Debit Account
 - Invoice Routing Settings Screen
- The setting does Not evaluate the estimated dollar amounts on the Estimate Detail Items for each G/L Account.
- Setup: This setting must be enabled in the Advanced Settings screen which is accessed via the System Administrator option on the File Menu:

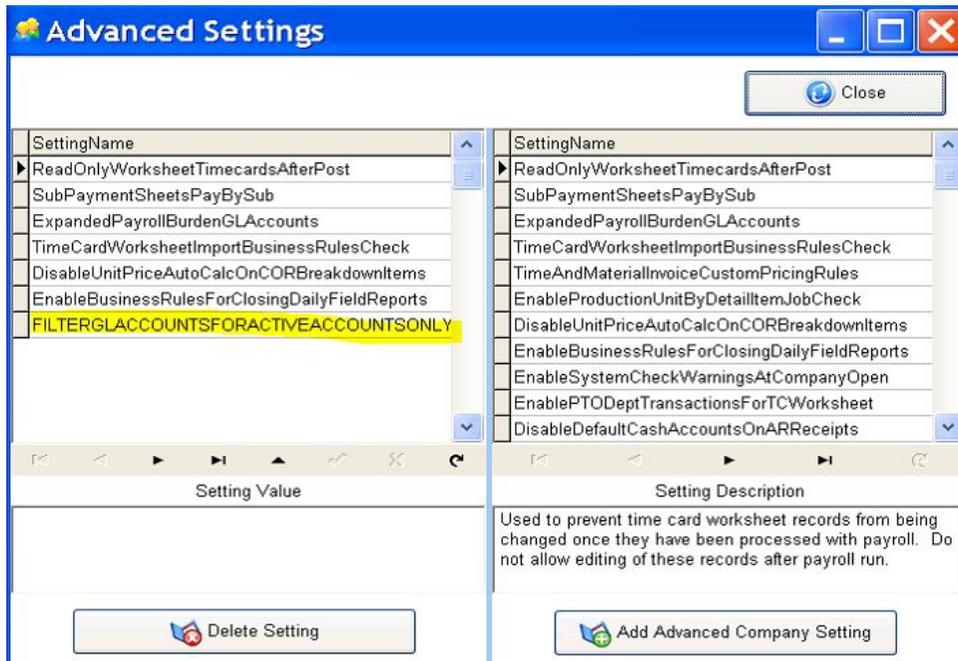


Accounts Payable - Enhancement to Mark GL Accounts as Inactive / Hide from Lists

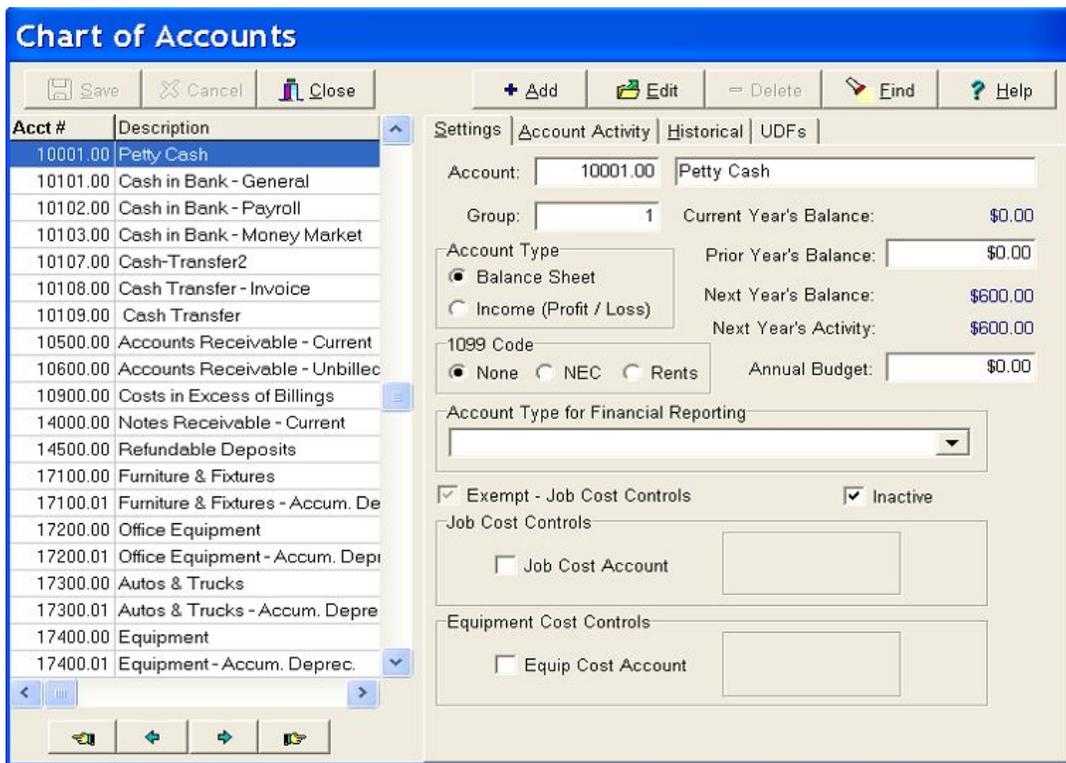
- Overview:** A new feature has been added to mark G/L Accounts as Inactive for Accounts Payable Invoice entry purposes. This feature requires a new Advanced Setting to be enabled along with one or more G/L Accounts being marked as Inactive in the Chart of Accounts Screen. In other words, an account on the Chart of Accounts can be marked as inactive, but this setting will only be respected by the Accounts Payable screens when the new Advanced Setting is enabled. See the Setup information below for more details.

- When this setting is enabled, it is applied to the following input screens:
 - Regular AP Invoices - non routing
 - Regular AP Invoices - routing
 - Purchase Order Creation - PO Items - Debit Account
 - Invoice Routing Settings Screen
- Setup: To Implement this feature you will need to do the following:
 - Set up the Advanced Setting Feature in the system configuration screen entitled **“FILTERGLACCOUNTSFORACTIVEACCOUNTSONLY”**:





- Mark one or more GL Accounts in the Chart of Accounts Screen as Inactive by checking the new Inactive Checkbox:



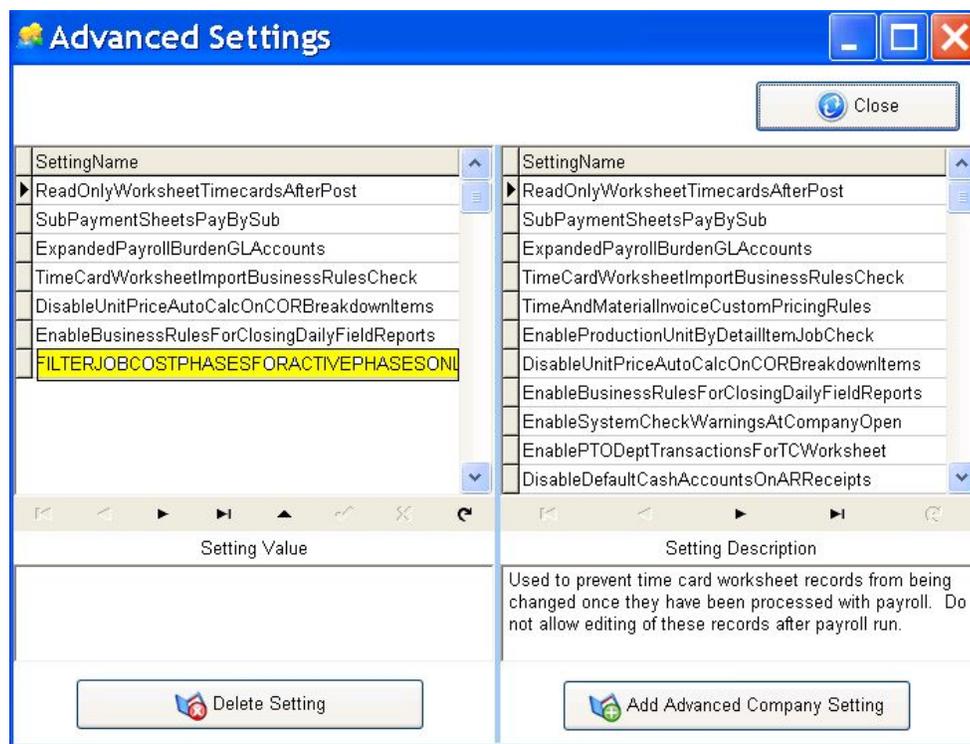
Any Inactive account will NOT be available in the designated A/P Entry Screens when this new Advanced Setting is enabled. You may still mark an account as inactive on the Chart of Accounts for internal purposes, but without the Advanced Setting enabled, the Inactive Checkbox is ignored. NOTE: The new Inactive setting is linked to the *glmaster.miscbool2* field.

Accounts Payable / Project Management Enhancements - Mark Cost Phases as Inactive / Hide from Lists

- Overview - A new Advanced setting has been added to allow cost phases on the job cost master (JOBCOST) table to be marked as inactive so that they cannot be selected for date input on certain screens.
- When this setting is enabled, it is applied to the following input screens:
 - Regular AP Invoices - non routing
 - Regular AP Invoices - routing
 - Purchase Order Creation - PO Items
 - Invoice Routing Settings Screen
 - Project Management Log - Daily Report screen - Personal Tab
 - Project Management Log - Daily Report screen - Equipment Tab

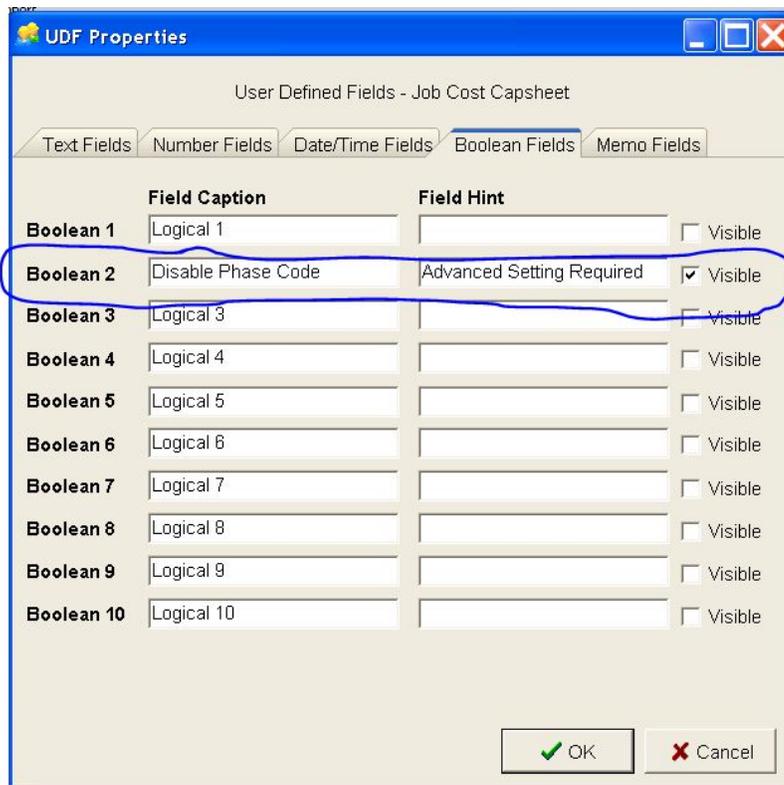
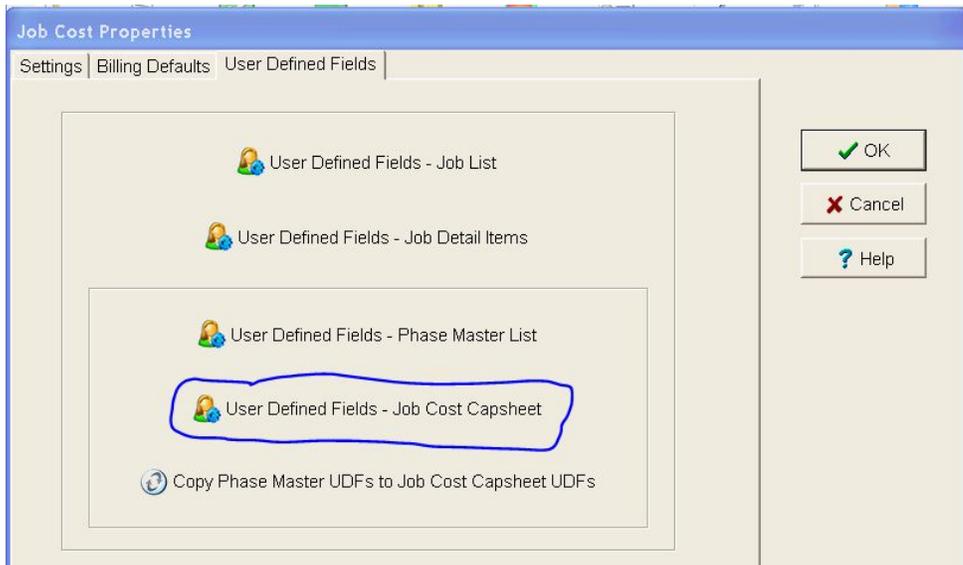
IMPORTANT NOTE: The Project Management Log Screen is dependent on the AccuBuild Mobile Stored Procedure Library for looking up the job cost phases. If the mobile stored procedures are set up for tailored solution to implement this setting, then the Daily Report screen will respond accordingly. See the notes below for the stored procedure changes that need to be implemented in order to activate the Cost Phase filters for Daily Reports.

- Setup: To implement this feature you will need to do the following:
 - Set up the Advanced Setting Feature in the system configuration screen entitled “**FILTERJOBCOSTPHASESFORACTIVEPHASESONLY**”:

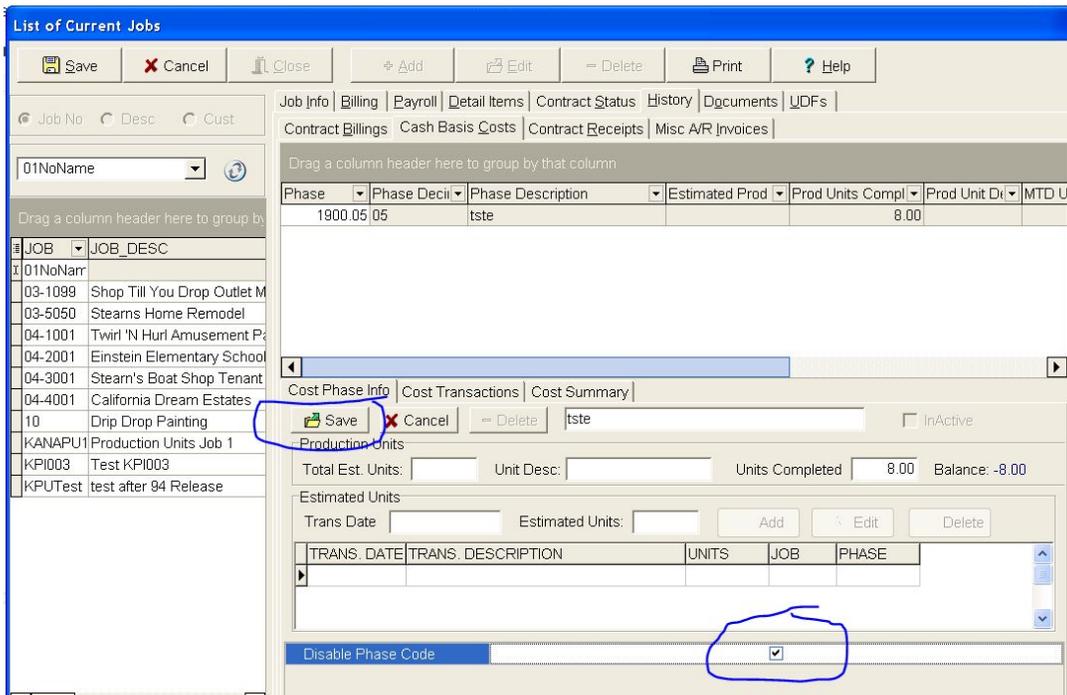


- When this Advanced Setting is enabled, the Phase drop down lists will display only phases from the JOBCOST table that are Active Phases.

- You will need to set up a UDF On the JOB COST Table for the UDFLogical2 Field in order to use this feature. Open the Job Cost Properties Screen and select the Job Cost Capsheet option from the User Defined Fields Tab:



- Enable the *Boolean2* Field by filling in the Field Caption, Field Hint (optional), and checking the Visible Checkbox.
- Mark the Phases to be InActive or Active by setting the new UDF field as set up in the previous step. Go to Job -> History -> Cash Basis Costs -> Edit. Locate the new UDF field at the bottom of the screen and check the box to Disable the Phase Code (Inactivate), or else uncheck the box to enable the phase code, When you change the setting on the checkbox, The Save and Cancel button will be enabled. Click the Save Button to save the new UDF Setting or Cancel to abort the change.



- If AccuBuild Mobile is being used or you are using daily reports in the project management system, and you want to push these changes to the mobile stored procedures, you will need to tailor the SQL for the following stored procedures. Note: Changes are indicated in **BLUE Text**.

- **abm_GetJobPhase_Lookup** - Library Script 7772000 Seq No 20:

```
//////// JOB PHASE LIST - JOB SPECIFIC
```

```
try
  drop procedure abm_GetJobPhase_Lookup2;
catch all
  // nothing to do - just ignore
end;
```

```
CREATE PROCEDURE abm_GetJobPhase_Lookup2
```

```
(
  UserID Integer,
  Job CHAR ( 8 ),
  Phase DOUBLE ( 15 ) OUTPUT,
  Phase_Desc CHAR ( 35 ) OUTPUT
)
```

```
BEGIN
```

```
Declare @COPath string;
Declare @COFreePath string;
Declare @UserID integer;
Declare @SQLStmt string;
Declare @JobNo string;
Declare @DecimalPhaseFilterCount integer;
Declare @C1 Cursor;
```

```
Declare @c3 cursor;
Declare @AdvancedProcedureName string;
Declare @PhaseInactiveSettingEnabled logical;
Declare @PhaseInactiveFilter string;
```

```
@UserID = (select UserID from __input);
```

```
@JobNo = (select Job from __Input);
```

```
@COPath = (select rtrim(companyuncpath)+'BASECOMP.ADD' as datapath from GLOBALCLOUDCOMPANYSETTINGS
where userno = @UserID);
```

```

@COFreePath = (select rtrim(companyuncpath) as datapath from GLOBALCLOUDCOMPANYSETTINGS where usern
= @UserID);

@AdvancedProcedureName = 'FILTERJOBCOSTPHASESFORACTIVEPHASESONLY';
try
  open @c3 as EXECUTE PROCEDURE abm_GetCompanyAdvancedSettingStatus( @UserID,
@AdvancedProcedureName);
  if fetch @c3 then
    @PhaseInactiveSettingEnabled = @c3.AdvancedSettingExists;
    @PhaseInactiveSettingEnabled = coalesce(@PhaseInactiveSettingEnabled,false);
  endif;
  close @c3;
catch all
  @PhaseInactiveSettingEnabled = false;
end;
If @PhaseInactiveSettingEnabled then
  @PhaseInactiveFilter = ' AND coalesce(UDFLogical2,false) = false ';
else
  @PhaseInactiveFilter = "";
endif;

try
  open @c1 as execute Procedure abm_GetPhaseDecimalFilterCount(@UserID);
  if fetch @c1 then
    @DecimalPhaseFilterCount = @c1.recordcount;
  else
    @DecimalPhaseFilterCount = 0;
  endif;
catch all
  @DecimalPhaseFilterCount = 0;
end;
close @c1;

if @DecimalPhaseFilterCount > 0 then
  @SQLStmt = 'insert into __output (Phase,Phase_desc) select phase,phase_desc from '+
    ""+@CoPath+"".JobCost where Job = ""+@JobNo+"" and phase_desc is NOT NULL '+@PhaseInactiveFilter+
    ' AND round((phase - truncate(phase,0))*100,0) in (select convert(PhaseDecimal,sql_integer) as PhaseCategory from
'+""+trim(@COFreePath)+'mobilephasedecimalsenabled" ) '+
    ' group by phase,phase_desc;';
else
  @SQLStmt = 'insert into __output (Phase,Phase_desc) select phase,phase_desc from '+
    ""+@CoPath+"".JobCost where Job = ""+@JobNo+"" and phase_desc is NOT NULL '+@PhaseInactiveFilter+' group
by phase,phase_desc;';
endif;
execute immediate @SQLSTMT;

END;

```

■ **abm_GetActiveJobCostCapsheetPhases_Lookup** - Library Script 7772000 Seq No 340:

```

try
  drop procedure abm_GetActiveJobCostCapsheetPhases_Lookup2;
catch all
  // nothing to do - just ignore
end;

CREATE PROCEDURE abm_GetActiveJobCostCapsheetPhases_Lookup2
(
  UserID Integer,
  Job CHAR ( 8 ) OUTPUT,
  Phase DOUBLE ( 15 ) OUTPUT,
  Phase_Desc CHAR ( 35 ) OUTPUT,
  ResultMessage CHAR ( 1500 ) OUTPUT,
  ResultStatus Integer OUTPUT
)
BEGIN
  Declare @COPath string;

```

```

Declare @UserID integer;
Declare @SQLStmt string;
Declare @UserNANO double;
Declare @UserJobSetting integer;
Declare @UserJobWhere string;
Declare @ResultMessage string;
Declare @ResultStatus integer;

Declare @c3 cursor;
Declare @AdvancedProcedureName string;
Declare @PhaseInactiveSettingEnabled logical;
Declare @PhaseInactiveFilter string;

@UserID = (select UserID from __input);
@CoPath = (select rtrim(companyuncpath)+'BASECOMP.ADD' as datapath from GLOBALCLOUDCOMPANYSETTINGS
where usermo = @UserID);

@AdvancedProcedureName = 'FILTERJOBCOSTPHASESFORACTIVEPHASESONLY';
try
    open @c3 as EXECUTE PROCEDURE abm_GetCompanyAdvancedSettingStatus( @UserID,
@AdvancedProcedureName);
    if fetch @c3 then
        @PhaseInactiveSettingEnabled = @c3.AdvancedSettingExists;
        @PhaseInactiveSettingEnabled = coalesce(@PhaseInactiveSettingEnabled,false);
    endif;
    close @c3;
catch all
    @PhaseInactiveSettingEnabled = false;
end;
If @PhaseInactiveSettingEnabled then
    @PhaseInactiveFilter = ' AND coalesce(UDFLogical2,false) = false ';
else
    @PhaseInactiveFilter = "";
endif;

@UserNANO = (select isnull(misc_float2,0.00) from users where userrecordid = @UserID);
@UserJobSetting = (select isnull(misc_int1,0) from users where userrecordid = @UserID);
@UserJobWhere = "";
if @UserJobSetting > 0 then
@UserJobWhere = (
select case @UserJobSetting
    when 1 then ' AND ProjectManagerNo = '+trim(convert(@UserNANO,sql_char))
    when 2 then ' AND SalesmanNo = '+trim(convert(@UserNANO,sql_char))
    when 3 then ' AND SupervisorNo = '+trim(convert(@UserNANO,sql_char))
    when 4 then ' AND (ProjectManagerNo = '+trim(convert(@UserNANO,sql_char))+ ' OR SalesmanNo =
'+trim(convert(@UserNANO,sql_char))+ ' OR SupervisorNo = '+trim(convert(@UserNANO,sql_char))+
        ' OR Miscalc3 = '+trim(convert(@UserNANO,sql_char)) +
        ' OR Overheadpercent = '+trim(convert(@UserNANO,sql_char))+ ' ) '
    when 5 then ' AND Miscalc3 = '+trim(convert(@UserNANO,sql_char))
    when 6 then ' AND Overheadpercent = '+trim(convert(@UserNANO,sql_char))
end as jobfilter from system.iota);
endif;

@SQLStmt = 'insert into __output (Job,Phase,Phase_desc) select Job,phase,phase_desc from '+
'''+@CoPath+'''.JobCost where job in ('+
        'select job from '+
'''+@CoPath+'''.JobList where isnull(complete,false) = false '+@UserJobWhere+') '+ @PhaseInactiveFilter +
        ' group by Job,Phase,Phase_desc;';

@ResultStatus = 0;
@ResultMessage = "";
try
    execute immediate @SQLSTMT;

```

```

catch all
  @ResultStatus = -1;
  @ResultMessage = __errText+' SQL Statement Failed: '+@SQLStmt;
end;
if @ResultStatus = -1 then
  try
    insert into __Output (ResultStatus,ResultMessage) VALUES (@ResultStatus,@ResultMessage);
  catch all
  end;
endif;

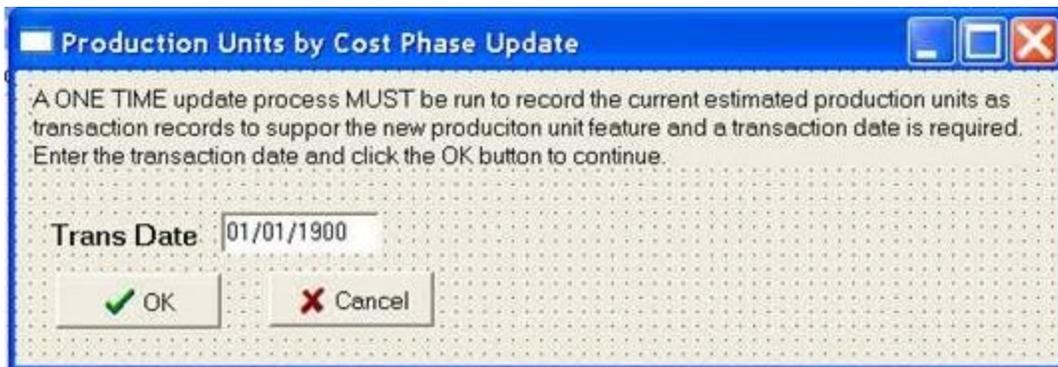
END;

```

Estimated Production Units Enhancement - transaction driven production unit estimates

- **Overview** - The Production Units by Cost Phase system has been updated to track the estimated production unit changes using transaction records. Previously, production units were handled as a static field updates where the estimated units were updated but no audit trail or dates were being tracked. The Job List Screen for tracking estimated production units has been updated with new controls to handle the new transaction based entries. A new transaction type of 19 will now be used in the JOBDETL table to track the estimated production unit changes.
- **Setup:** No setup is required to enable this feature as it is enabled automatically. However, if you have been using the Production Units by Cost Phase process, then the first time you open the job list screen, you will be prompted to supply a transaction date in order to record transaction records for the current estimated production unit balances. Although you may cancel out of this screen, the Job List will NOT be functional until this ONE TIME process is carried out. When you click OK on this screen, a new production unit transaction record will be created for every job cost phase that has an estimated production units balance. This process is necessary in order to switch the system to a transaction driven process without losing any current estimated balances.

NOTE: You may pick any transaction date including the current date. Just remember that this will be the starting date for when the production unit changes will be tracked with an actual transaction date. If you are in the practice of closing out your accounting periods (month end process), you may want to choose the month ending date of the last month that has been closed.



- **Process:** The following information explains how to use the new features and controls for updating estimated units for the production units by cost phase process. Since the production units are now tracked by transaction records, there are new buttons to Add new estimates, Edit existing estimate records, or Delete existing estimate records.

1. Select a Job and navigate to the History Tab -> Cash Basis Costs Tab. The 'Estimated Units' group with the transactions grid should be visible as shown below.

NOTE: You do NOT need to Edit the Job to use these new button for production.

List of Current Jobs

Save Cancel Close Add Edit Delete Print Help

Job No Desc Cust

01NoName

Drag a column header here to group by that column

Phase	Phase Decim	Phase Description	Estimated Prod U	Prod Units Comple	Pr
10.00	00	Materials	15.00	50.00	EA
20.10	10	Labor	50.00	.00	
40.00	00	Equipment Rentals - Vendors	50.00	.00	
60.00	00	Plans, Permits & Fees	.00	.00	
1900.00	00	Misc Job Expenses	.00	.00	
1900.01	01	Labor	.00	.00	

Cost Phase Info Cost Transactions Cost Summary

Save Cancel Delete Materials

Production Units

Total Est. Units: 15.00 Unit Desc: EA Units Completed 50.00 Balance:-35.00

Estimated Units

Trans Date 09/26/2017 Estimated Units: 15.00 Add Edit Delete

TRANS. DATE	TRANS. DESCRIPTION	UNITS	JOB	PHASE
09/26/2017	Initial Bal Prod Units - Cost Phase	15.00	03-5050	10

Text 1
Text 5
Text 6

Open Closed Both

- Select a phase and then use the *Add*, *Edit* or *Delete* buttons in the 'Estimated Units' group as necessary. If you are Adding or Editing a transaction, you will be able to update the transaction date and the estimated units field. For a new entry, the transaction date will default to the current date. Use the *Save* Button to save the changes or the *Cancel* Button to cancel the changes. If you are deleting a transaction, you will be prompted with a confirmation screen before the record is deleted. NOTE: If there are no existing transactions for the current phase, then only the *Add* Button will be visible.

Estimated Units

Trans Date 10/24/2017 Estimated Units: 0.00 Save Cancel

TRANS. DATE	TRANS. DESCRIPTION	UNITS	JOB	PHASE
09/26/2017	Initial Bal Prod Units - Cost Phase	15.00	03-5050	10
* 10/24/2017	Estimated Prod Units - Cost Phase	0.00	03-5050	10

- After any change to the transaction records, the production unit totals will be immediately updated on the screen with the new estimate and balance amounts.

Production Units

Total Est. Units: 2500.00 Unit Desc: CU Units Completed .00 Balance: 2500.00

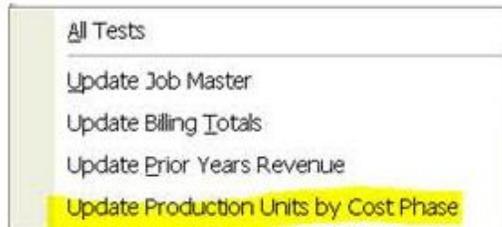
Estimated Units

Trans Date 09/30/2017 Estimated Units: 1500.00 Add Edit Delete

TRANS. DATE	TRANS. DESCRIPTION	UNITS	JOB	PHASE
09/30/2017	Estimated Prod Units - Cost Phase	1500.00	03-5050	2200
11/07/2017	Estimated Prod Units - Cost Phase	1000.00	03-5050	2200

Job Cost Integrity Tests - New Option for Estimated Production Units

- With the introduction of the new feature for transaction driven production units by cost phase, a new update feature was necessary for the Job Cost Integrity Tests menu. This new option entitled “Update Production Units by Cost Phase” can be used to update the running estimate totals for each cost phase from the new estimate transaction records.

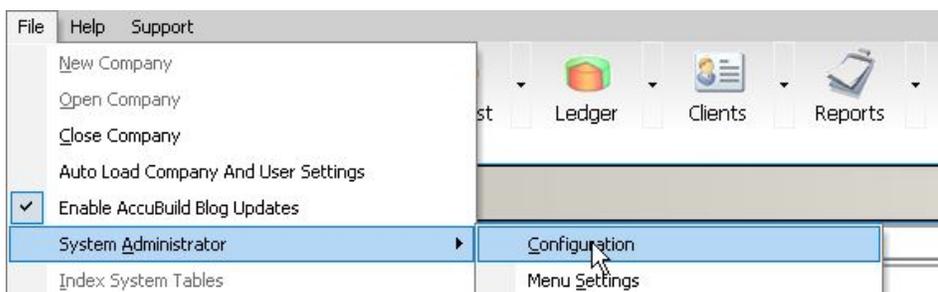


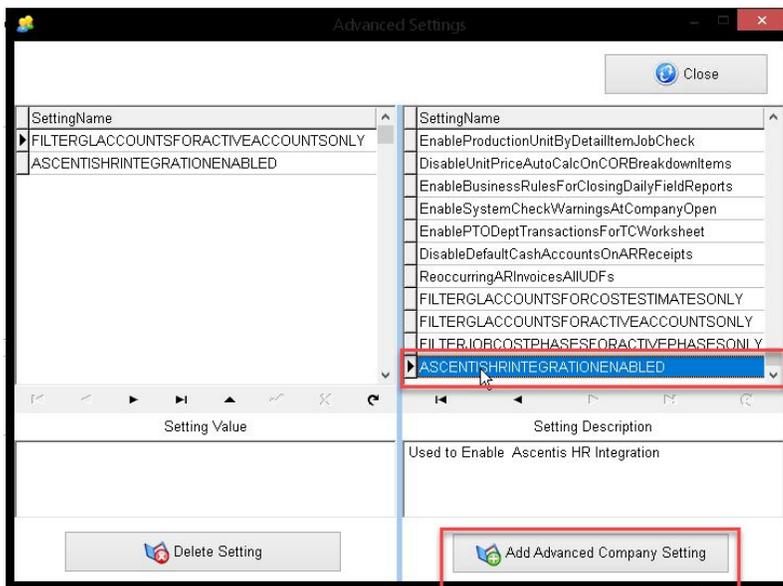
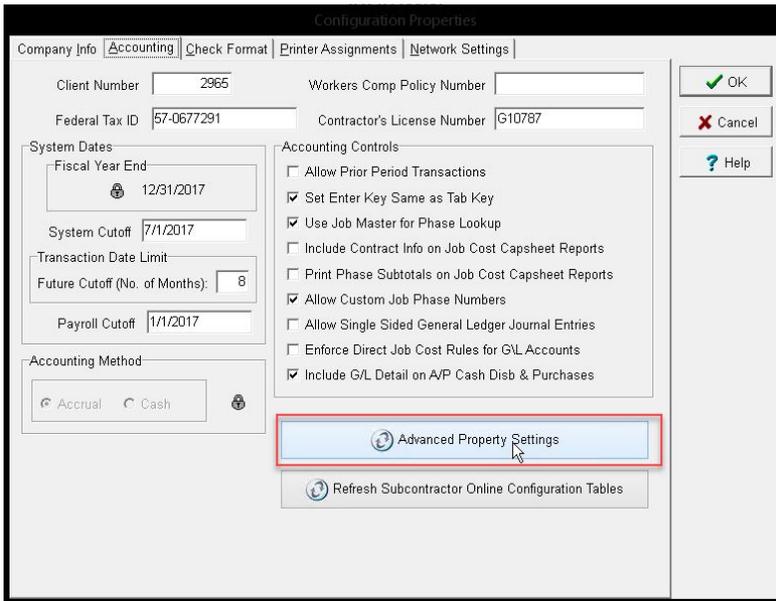
Payroll ENHANCEMENT - Integration with Ascentis HR Software

- Overview: A new integration product has been created to work with the Ascentis Human Resources Software Package. This integration is a custom tailored solution that synchronizes employee field changes between the Ascentis Database and the AccuBuild Employee Records. Changes from the Ascentis System are automatically pushed to the AccuBuild Employee List and selected AccuBuild Employee changes are pushed to the Ascentis System. This process is typically done once at the start of each day to update both systems with any changes from the previous day.
- Setup: The setup process for the Ascentis Integration involves several different systems which include:
 - The AccuBuild Program - Setting up the Advanced Setting so that employee changes in AccuBuild are flagged for updates to Ascentis.
 - The AccuBuild Database Tables - The Ascentis Integration configuration and transaction tables which track the changes and handles the business rule settings for the two systems.
 - The AccuBuild File Transfer Program Configuration - handles passing data between the AccuBuild Cloud and the Ascentis Cloud servers via ascii files.
 - The Automated Alerts System in AccuBuild - Needed to schedule the automatic synchronization process.

Separate documentation is available for the Ascentis Integration setup process and the following information only includes the setup process to enable the feature inside the AccuBuild Program using a new Advanced Setting:

- In order to activate the tracking of employee field changes for the Ascentis Integration, you will need to enable the new Advanced Setting entitled “ASCENTISHRINTEGRATIONENABLED”:





Production Units by Detail Item - Bug Fix

- The Stored Procedure that is used to compute the Production Units totals for cost phases, bid items, and tasks was not summarizing the totals correctly at the bid item level and changes were made to the stored procedure library to correct the issues:
 - abm_Get_DPU_BidItemMasterRecs_FullDetail_Lookup

Production Units by Detail Item - Enhancements

- A new stored procedure was added to the library for tracking any error conditions that violate the business rules for production unit record grouping for Bid Item and Task relationships. This procedure is being used to add new error reporting records in the production unit reports.
 - abm_Get_DPU_BidItemErrorRecords_Lookup

- The existing report classes for production unit reports and forms have been updated to include the new table for tracking bid item errors entitled "BidItemErrorRecords". These records are actually created from the new stored procedure *abm_Get_DPU_BidItemErrorRecords_Lookup* so any changes to the stored procedure will automatically be updated in the associated reports and forms for these classes:
 - Class 455 - Production Units by Detail Item Form
 - Class 839 - Production Units by Detail Item Report

- The production unit library reports have been updated to include the new error records in the report at the end of report in order to alert the user for any business rule violations.
 - PMREP17.839 - Production Units by Detail Item - REPORT
 - PMPDNUN1.455 - Production Units by Detail Item - FORM

MAR Views - BUG FIX

- In the previous release, a new enhancement was added to the MAR Views which was able to save the grid grouping and column filtering settings in addition to the other settings such as column selections and column order etc. These new changes also changed the names of the INI files that stored the grid settings and for MAR Views, this caused a bug where the user configuration settings were no longer remembered. This problem has been corrected in this release.