What's New! ProAdmin®

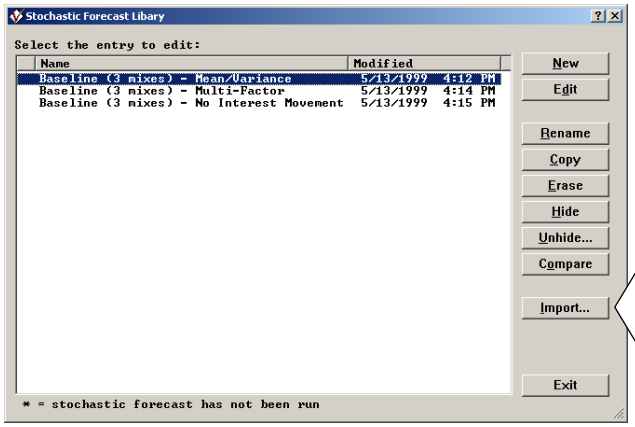
ProAdmin version 2.01 **December 2005**

ProAdmin version 2.01 introduces a robust **Import from Client** feature, **"if then else"** for ProAdmin expressions, the addition of the Service Definition Set calculations to the summary output and the audit report, a *rewrite of the **Service Definition** screens, and enhancements to the **Projection Assumptions** salary increases*. You'll find details about these and other enhancements below.

Please note: Items in *italics* are anticipated, but have not been included in the 2.01 base release. These will be released as updates to the 2.01 version. You will automatically be notified by email when updates are made available.

System

- Import from Client lets you import any type of item from another client, automatically including referenced items. For example, if you import a Plan Definition, ProAdmin will import the Plan Definition plus any referenced Benefit Definitions, Benefit Formula Components, Payment Forms, etc. Look for the Import button in each library. Import from Client replaces the (rather limited) Client to Client Copy command found in previous versions of ProAdmin.

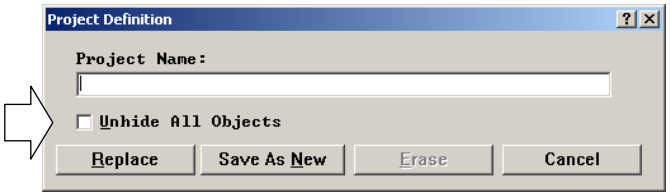


See Import from Client, Page 6

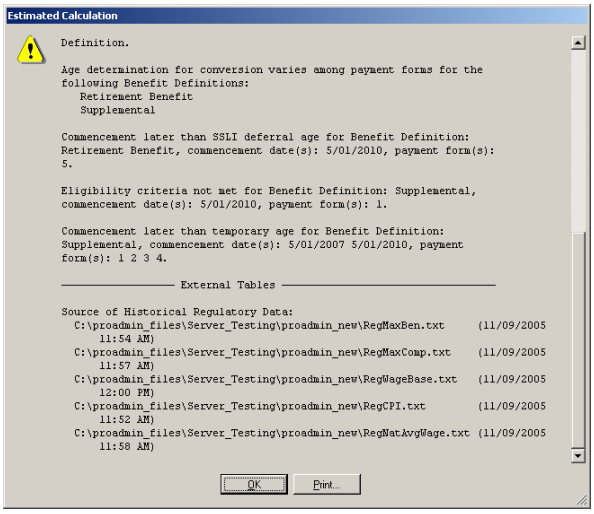
- ProAdmin now uses dynamic memory management to virtually eliminate WSFULL errors. The WSSize parameter in provalw.ini is obsolete.
- All clients, regardless of size, now open instantly.
- "If then else" syntax is now available to write ProAdmin expressions. For example, "[(division=1)*100]+[(division=2)*150]" can be written as "#If Division=1 #Then 100 #ElseIf Division=2 #Then 150 #EndIf"

See If-Then-Else, Page 7

- New projects can now be created with all objects unhidden.



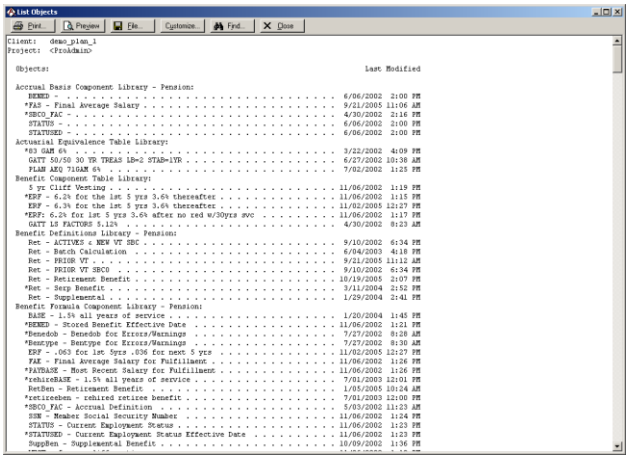
- A new system message has been added that displays the path, name, and date time stamp for all Historical Interest Rate tables and Regulatory Data files used in the calculation. These are shown in a new External Tables message at the end of the ProAdmin processing messages. (Please see illustration on the top of the next page.)



The regulatory data and interest rate table information has also been added as comments in the top of the XML return document from ProAdmin Server, as shown below.

```
<?xml version="1.0"?>
<!-- Created by ProAdmin (Winklevoss Technologies, LLC) -->
<!-- Calculation date: December 6, 2005  4:35 PM -->
<!-- System Plan: INDIVIDUAL CALCS - ESTIMATES -->
<!-- System Plan last updated: December 2, 2005  1:22 PM -->
<!-- System Plan update level date: December 1, 2005  11:37 AM -->
<!-- Repository file: d:\proadmin_files\Server_Testing\repository\repos_next_rel.sf -->
<!-- Repository file last updated: December 2, 2005  2:50 PM -->
<!-- ProAdmin engine effective date: December 6, 2005  4:34 PM -->
<!-- External Tables: -->
<!-- Source of Historical Regulatory Data: -->
<!-- d:\proadmin_files\Server_Testing\proadmin_new\RegMaxBen.txt  (11/09/2005 11:54 AM) -->
<!-- d:\proadmin_files\Server_Testing\proadmin_new\RegMaxComp.txt (11/09/2005 11:57 AM) -->
<!-- Source of Interest Rate Tables: -->
<!-- Directory:  d:\proadmin_files\government -->
<!-- 30_Year_Treas_Rate.csv ( 5/04/2005 10:23 AM) -->
```

- List Objects is a new command that allows you to view or print a list of all objects within a ProAdmin client. The list is similar to what's available when you view the inputs for an Estimate or Final calculation.



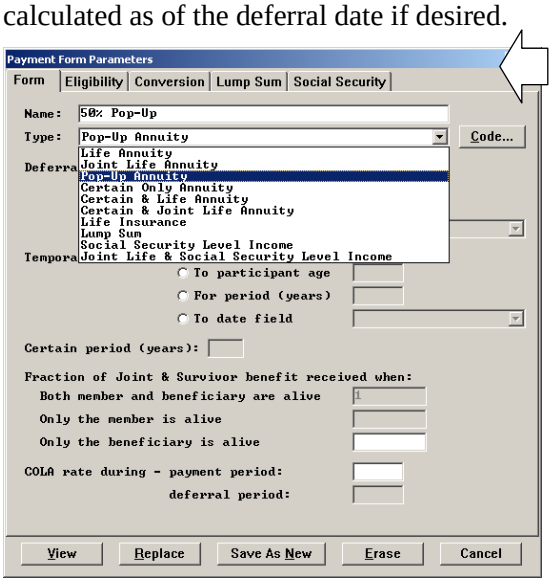
Database Linkage

- ProAdmin database linkage now allows special characters in table or field names.

See Database Linkage, Page 10

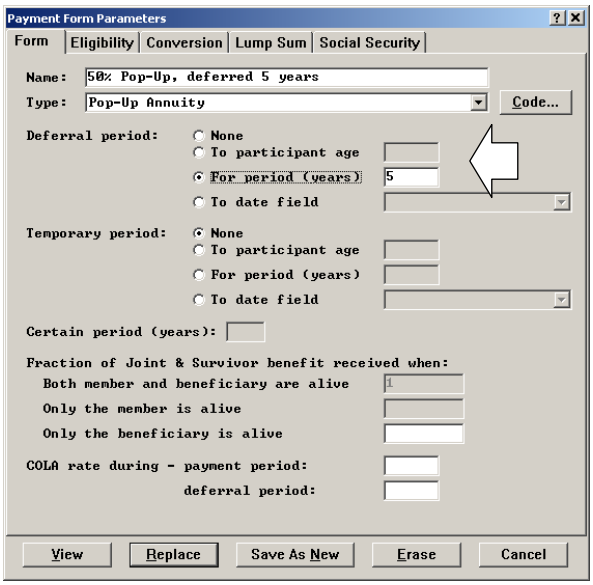
Payment Forms

- Pop-up annuity has been added as an available payment form. Pop-up annuities can be parameterized as immediate, temporary or deferred, and actuarial equivalence can be calculated as of the deferral date if desired.



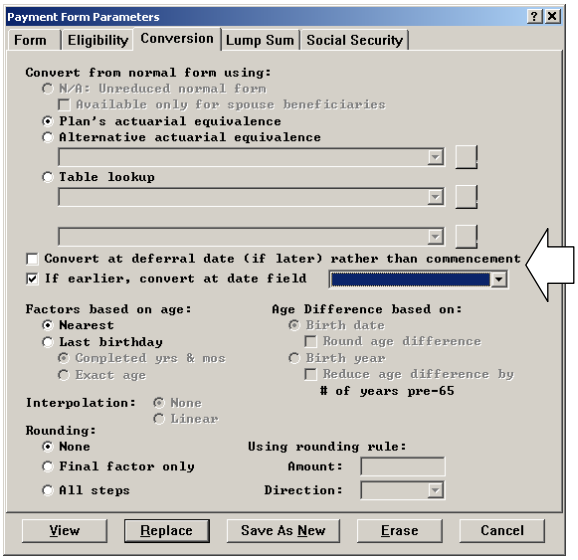
See Pop-up Annuity Payment Forms, Page 11

- A payment form can now be deferred for a specific number of years in addition to the previous options of deferring to an age or a date field. This is useful, for example, if the normal form of payment is 5 years certain and life and all optional form conversions are made after the initial 5 years of payment.



- Actuarial equivalence can now be calculated as of the deferral date for forms that are deferred to a date and for forms that are deferred for a specified number of years. Previously this option was only available for forms deferred to an age.

- A date that is the latest date for actuarial equivalence calculations can now be specified. This is useful for plans that calculate actuarial equivalence no later than at NRD, and also for specialized calculations based on a prior election by the participant.

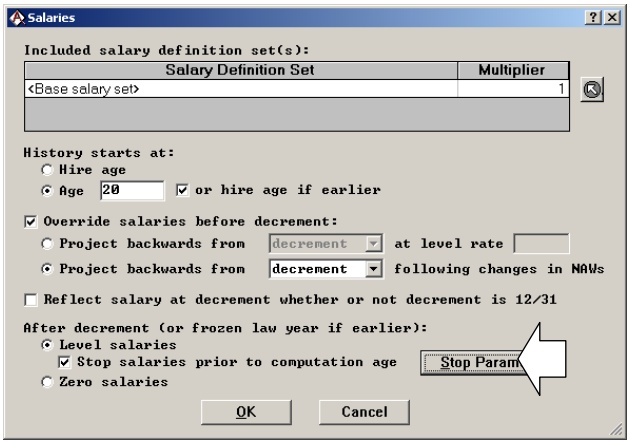


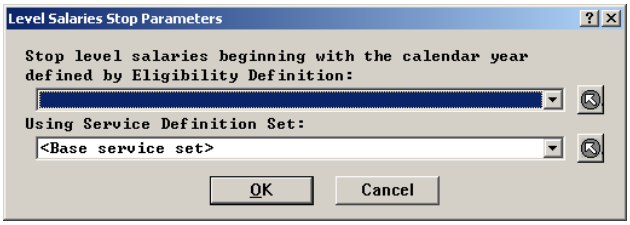
Benefit Formula Components

- Coded fields are now included among the choices for "database fields" in Benefit Formula and Accrual Basis Components. The mapping between codes and labels for these components appears in the expression help (F1).

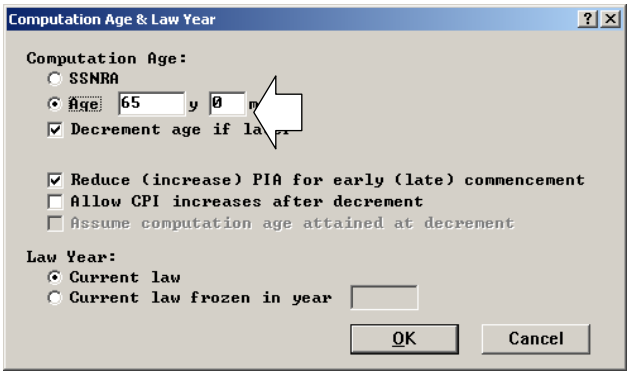
PIA Calculations

- Level salary projections under PIA Custom Operators can now be stopped prior to the computation age based on an eligibility definition.





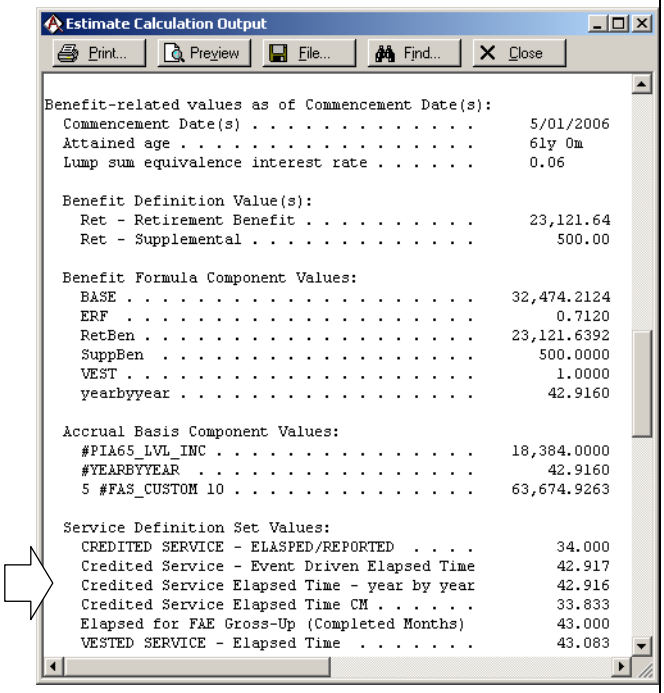
- The PIA Custom Operator computation age can now be specified in years and months rather than just years.



- The final monthly PIA amount is now rounded to lowest whole dollar. This is the same methodology used by the Social Security Administration and ProAdmin's Tools > PIA Calculations command.

Output

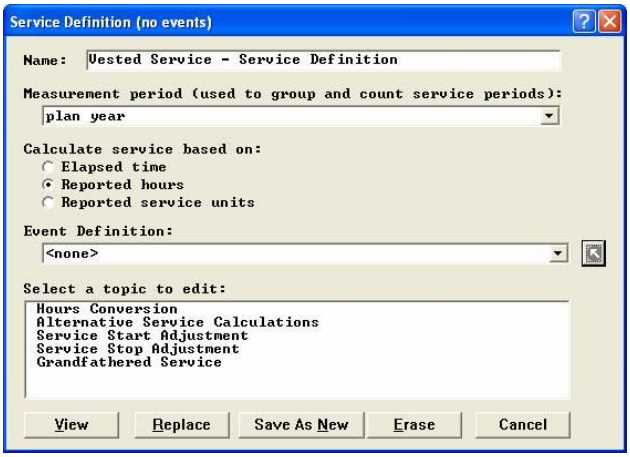
- ProAdmin now displays the results of the relevant Service Definition Set calculations in the Summary Results of an Estimate or Final Calculation. This information is also provided in the audit report from ProAdmin Server. Similarly to other summary output, the results are provided as of the requested decrement and commencement dates.



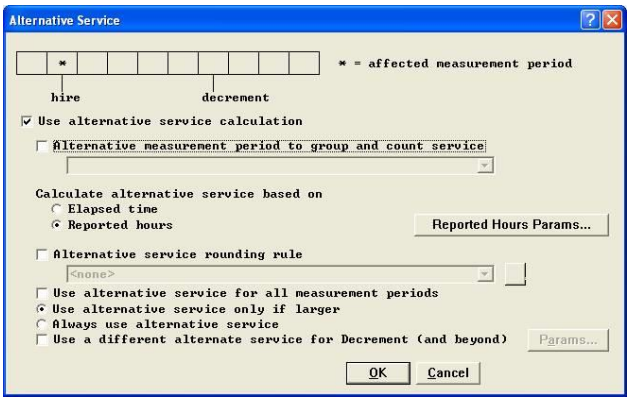
Service Definitions

- *The Service Definitions topic has been rewritten to present the user with a more concise and logical representation of the information required for a service calculation. It has been converted from a tab to a topic approach to make it easier to understand the information that must and/or can be parameterized.*

  Below is a sample of the new main service definition screen for service based on reported hours. Note that the list of available topics to edit will change as the service calculation approach changes. Note also that, while previously the main dialog asked if service was based on elapsed time or reported hours/service, now there is a distinct choice for reported hours versus reported service so that the other service topic choices will be more clearly labeled.



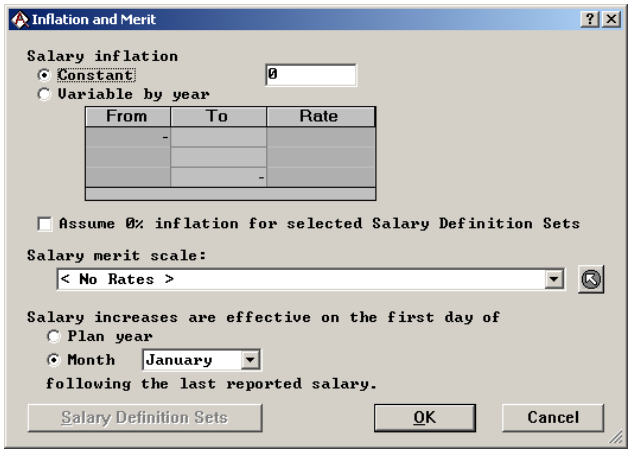
Below is a sample of the new Alternative Service Calculation screen. Note that graphics have been added to the dialog to provide a visual reminder of the period of service that is being parameterized.



Projection Assumptions

- *Several new options have been added regarding salary projections. To clarify their presentation, the projection assumption for salary increases is now split into two (2) topics: (1) Inflation and Merit and (2) Salary to Project.*

- *The Inflation and Merit topic sets the increase percentage and includes a new feature that allows you to specify the month that the increase is effective. Previously ProAdmin would only apply the increase on the first day of the plan year.*



- *The new Salary to Project topic defines the salary that will be used in the projection. A key enhancement in this area is that a database field, totally independent of any Salary Definition Sets, can now be specified as the basis of the salary projection.*
- *If the salary projection is not to be based on a database field, there is now much more flexibility as to how the Salary Definitions should be projected. Previously there were three (3) choices:*

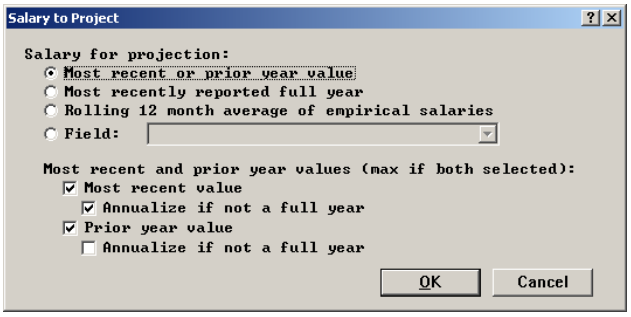
  1) Most recent available salary (annualized,)

  2) Greater of most recent (annualized) & prior year salaries (not annualized), or

  3) Rolling 12 month average of empirical salaries.

  Now there are more choices and there are variations within each choice. Specifically, you can choose:

  - *Most recent available salary, either annualized or not annualized.,*
  - *Prior year salary , either annualized or not annualized. (If there isn't a prior year salary, then the most recent salary is used.)*
  - *Greater of the most recent available salary and the prior year salary, where either or both can be annualized or not annualized.*
  - *Most recent fully reported year. (If there are no fully reported years, any of the preceding options can be chosen.)*
  - *Rolling 12 month average of empirical salaries.*



Changes Log

- Be sure to read the changes log (see What's New in Help or the ADMCHG.LOG file in the ProAdmin directory) about updates to certain calculations that may change results.

WinTech

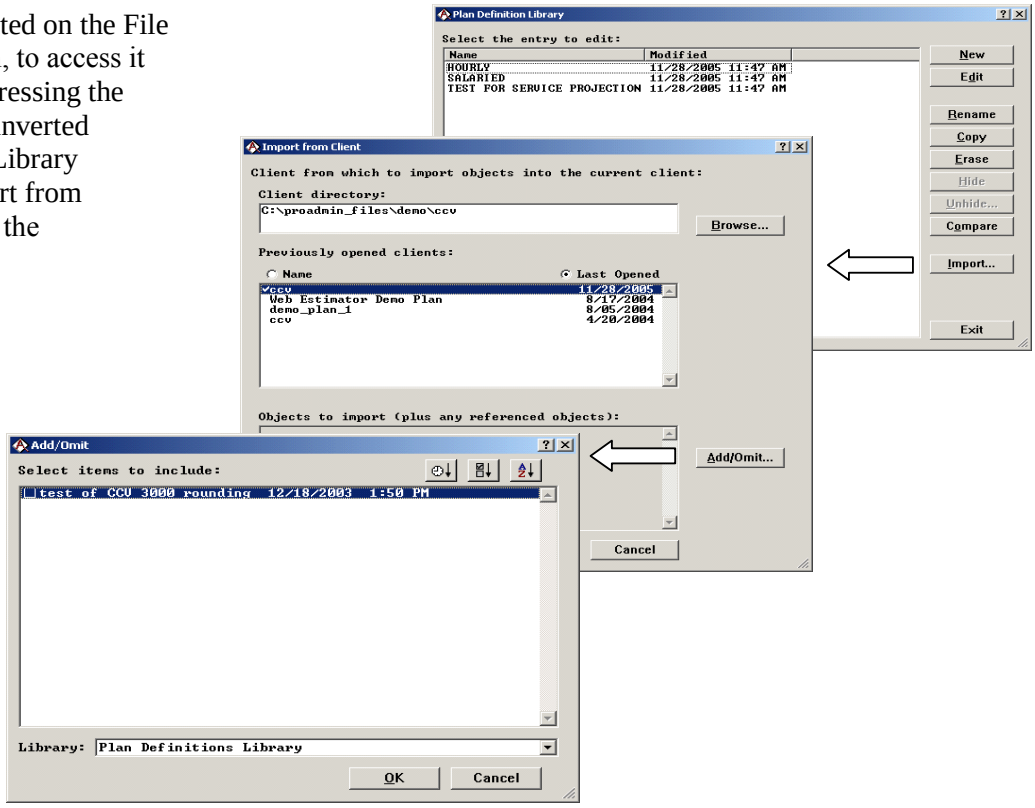
Import from Client

Import from Client lets you import any type of item from another client, automatically including referenced items. For example, if you import a Plan Definition, ProAdmin will import the Plan Definition plus any referenced Benefit Definitions, Benefit Formula Components, Payment Forms, etc. If ProAdmin encounters any naming conflicts, "#2" will be appended to the name of the imported item. If the identical item already exists (i.e., from previously using Import from Client), it will not be duplicated.

The possible uses for Import from Client are broad. Some examples include:

- Tables. Import tables from another client or a master client that you create.
- Fields. Import field names from a similar client or a template client that you create.
- Plan Definitions. Import a Plan Definition from a client with a similar design or from a template client that you create that contains sample plan definitions (e.g., a final average plan, a cash balance plan, etc.). A drawback of this approach is that Benefit Formula Components cannot easily be renamed.

Import from Client is located on the File menu. It's easiest, though, to access it from within a library by pressing the **Import…** button (or, for inverted libraries, by pressing the Library button and selecting Import from Client…). This will open the Import from Client command with the Mode and Library defaulted accordingly. You can **Add/Omit…** several items to import at once, from as many libraries as you wish.



Import from Client replaces the (rather limited) Client to Client Copy command found in prior versions of ProAdmin. It also replaces a lot of the functionality of Client Merge. Client Merge still exists, and is just a quick way of importing everything from one client into another. When the source client is a (revised) copy of the target client, Import from Client is often the better choice, since Client Merge will make extra copies of items with revised references in the source client.

If-Then-Else

Expressions are used within ProAdmin to:

- Create a data default;
- Define selection expressions; and
- Define benefit formulas.

ProAdmin version 2.01 introduces if-then-else operators to help you write clearer and more manageable expressions.

Multiplying by Boolean Values

Prior to version 2.01, in order to return conditional values (i.e., if x is true, then return value y), you had to multiply by Boolean values, as in:

```
[((PlanCode #in (1,2))* CAvgBft] + [(PlanCode=3) * FAvgBft]
```

This expression returns:

- `CAvgBft` for records with `PlanCode` equal to 1 or 2,
- `FAvgBft` for records with `PlanCode` equal to 3,
- `#MV` (missing value) for records with `PlanCode` equal to `#MV`, and
- 0 for records with `PlanCode` other than 1, 2, 3, or `#MV`.

If-Then-Else Operators

Now, you can use **`#IF`, `#THEN`, `#ELSEIF`, `#ELSE`,** and **`#ENDIF`** ("if-then-else") operators. The expression above can be rewritten as:

```
#IF PlanCode #in (1,2) #THEN CAvgBft #ELSEIF PlanCode=3 #THEN FAvgBft
#ENDIF
```

This expression returns:

- `CAvgBft` for records with `PlanCode` equal to 1 or 2,
- `FAvgBft` for records with `PlanCode` equal to 3, and
- `#MV` (missing value) for records with `PlanCode` other than 1, 2, or 3 (including `#MV`).

This expression is equivalent to the one above, except that it always returns `#MV` (missing value) for records with PlanCode other than 1, 2, or 3. It never returns 0. Barring this difference, the choice to multiply Boolean values or to use if-then-else operators is an aesthetic one. The aesthetic difference is especially noticeable when the `#IF` and `#ELSEIF` conditions are not mutually exclusive (see the example on the next page). In this case, if-then-else expresses the same logic in a more compact fashion. If-then-else, however, has little or no performance advantage.

When using if-then-else operators, line breaks and indentation can make it easier to read and provide space for comments, as in:

```
#IF PlanCode #in (1,2) #THEN
  CAvgBft                 ; career average benefit
#ELSEIF PlanCode=3 #THEN
  FAvgBft                 ; final average benefit
#ENDIF
```

The general syntax of if-then-else operators is:

```
#IF CondExpr1 #THEN
  ValueExpr1
#ELSEIF CondExpr2 #THEN
  ValueExpr2
  ...
#ELSE ValueExpr3
#ENDIF
```

Where:

- The if-then-else expression returns the `ValueExpr` corresponding to the first `#IF` or `#ELSEIF CondExpr` which is 1 (true).
- If no `CondExpr` is true, then the `ValueExpr` for `#ELSE` is returned. If no `#ELSE` is specified, then `#MV` (missing value) is returned. In some cases, such as benefit formulas, processing will halt rather than return a missing value.
- The `#ELSEIF` and `#ELSE` statements are optional. You may include as many `#ELSEIF` statements as desired, but only one `#ELSE` statement may be included.
- `CondExpr` must evaluate to 1, 0, or `#MV` (which is treated like 0). If not, the expression will result in a DOMAIN ERROR.
- `ValueExpr` may not be character.
- `CondExpr` and `ValueExpr` need not be surrounded by parentheses.
- Every `#IF` and `#ELSEIF` must be followed by `#THEN`.

If-then-else expressions can be nested, as in:

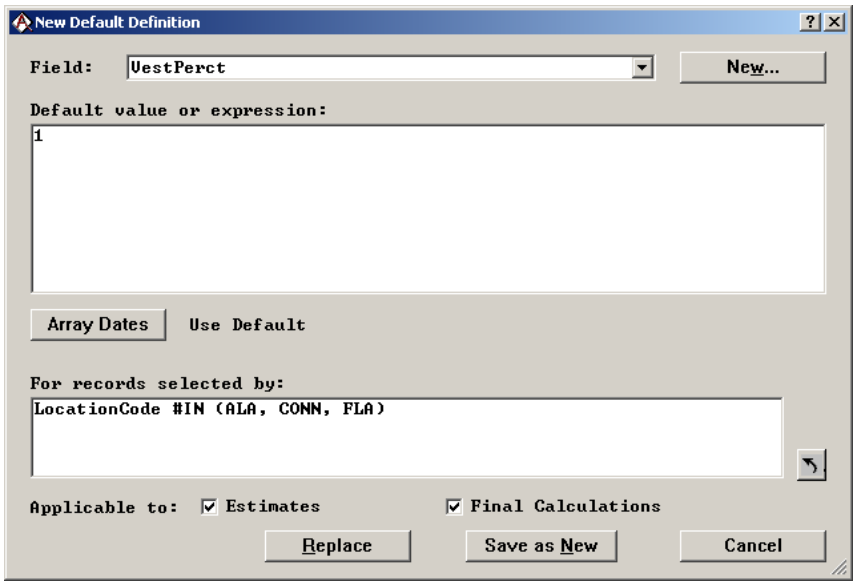
```
service := #DODEC #YEARDIF DOH  &
age     := #DODEC #YEARDIF DOB  &

#IF(plan=1) #THEN
  ; Early Retirement Date FOR SALARIED PARTICIPANTS
  #IF (age >= 55) #AND (service >= 5) #THEN
    2    ; 'UNRED'
  #ELSEIF (service >= 15) #AND (age + service >= 75) #THEN
    1    ; 'RED'
  #ELSEIF age >= 60 #THEN
    5    ; '5YRS'
  #ELSEIF (service >= 10) #AND (age + service >= 65) #THEN
    5    ; '5YRS'
  #ELSEIF age >= 55 #THEN
    10   ; '10YRS'
  #ELSE
    0    ;  'OTHER'
  #ENDIF
#ELSE
  ; RET ELIGIBILITY BUCKET FOR HOURLY PARTICIPANTS
  #IF age >= 60 #THEN
    2    ; 'UNRED'
  #ELSEIF (service >= 25) #AND (age >= 55) #THEN
    2    ; 'UNRED'
  #ELSEIF  service >= 15 #THEN
    5    ; 'RED'
  #ELSEIF (age >= 50) #OR (service >= 5) #THEN
    10   ; '10YRS'
```

```
    #ELSE
      0     ; 'OTHER'
    #ENDIF
  #ENDIF
```

While temporary assignment can be used in conjunction with if-then-else operators, as in the expression above, it is not allowed within a `CondExpr` or `ValueExpr`. For example, the expression "`#IF (plan=1) #THEN (age:= ValDate #yeardif DOB) #ENDIF`" is not allowed. Conceptually, `#IF` is an operator that returns a value (like Microsoft Excel's "IF" operator), rather than a control structure that determines which lines to execute as found in structured programming languages.

Selection Expressions

If-then-else operators are not a substitute for selection expressions, which select a subset of records to process. For example, if you want to default the `VestPerct` field for those members located in Alabama, Connecticut, and Florida to 100% because of a divestiture while leaving it unchanged for everyone else, use the expression "`1`" along with the Selection Expression "`LocationCode #IN(ALA, CONN, FLA)`".



By contrast, using the expression "`#IF LocationCode #IN(ALA, CONN, FLA) #THEN 1 #ENDIF`" with no selection expression will set the VestPerct field to 1 for members in Alabama, Connecticut, and Florida and `#MV` (missing value) for everyone else.

Database Linkage

ProAdmin database linkage now allows special characters and spaces in table or field names. For example, in MS Access you can now have tables named “Base Pay” or "Acc Ben @ (py end)" and fields named "Employee$" or "Pay 98" and ProAdmin will retrieve the data correctly. This change also allows you to use your EXECL workbooks as the data source.

When using Excel as a data source the following is some additional information to keep in mind.

1. If your worksheets have names on the tabs and your first row contains column headings you can select these with the look up arrows.
2. If the first row does not contain headers, you can specify HDR=NO in the extended properties of your connection string. If this is the case your fields are automatically named for you (where F1 would represent the first field, F2 would represent the second field, and so forth within the worksheet).
3. There is no direct way to specify the data types for columns in Excel the retrieval scans a limited number of rows in a column to "guess" the data type for the field. The number of rows to scan defaults to eight (8) rows; you can change the number of rows to scan by specifying a value between one (1) and sixteen (16) for the MAXSCANROWS setting in the extended properties of your connection string.

   This impacts where you are intending to use a data default to calculate a field. If the scan doesn’t find a data element to determine the data type it will default to a string. For example, if you are using an expression to calculate the date field DOP (date of participation) and in your Excel workbook this value is blank for all records, then ADO sets the data type to string when it reads the field in. In ProAdmin, a string is not a valid data type for a date field, so processing is aborted. To prevent this from happening, at least one of the records must have a valid date in it so ADO can determine that the correct data type is DATE (even though for a particular record the date may be empty or missing).

NOTE: When a WHERE statement is used in the Database Linkage regardless of the data source and one of the fields/tables in the where clause has special characters in the name, the user is responsible for adding square brackets around the names. For example, If you wanted to exclude base pay under $5,000 from the field you will now need to enter [Base Pay]>5,000 in the Where box.

Pop-Up Annuity Payment Forms

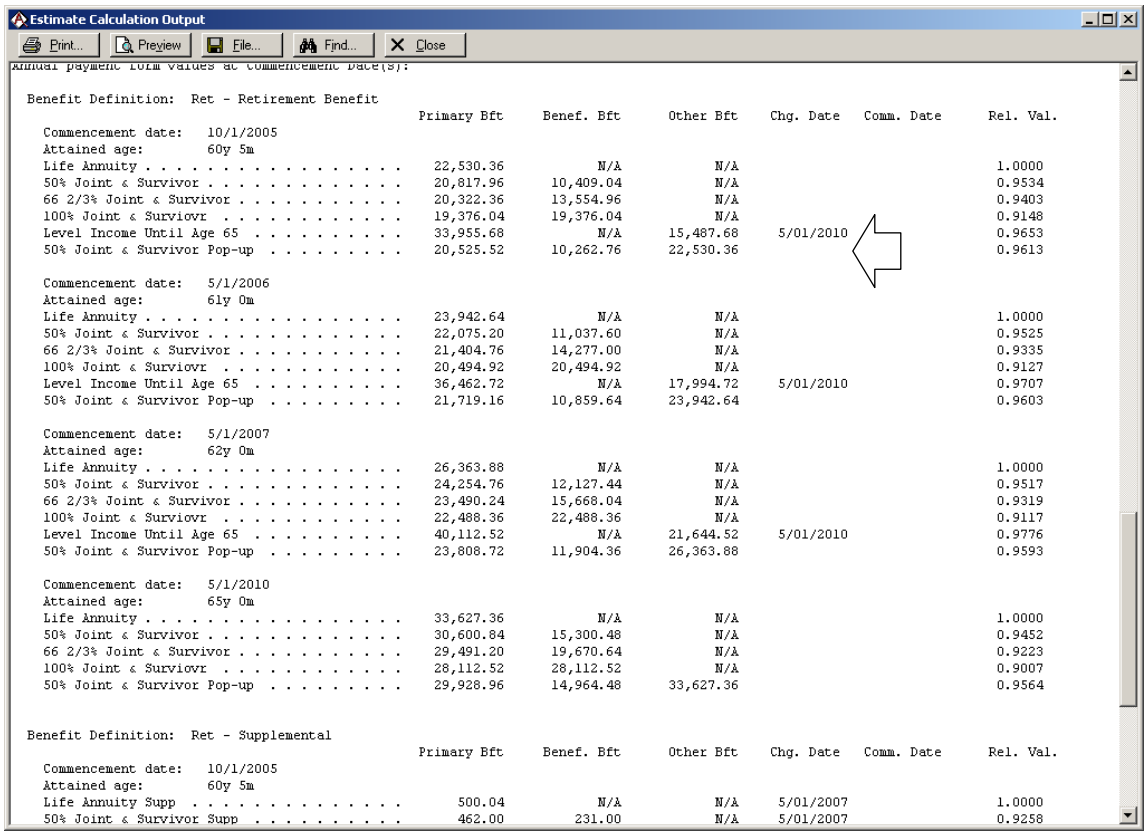
A pop-up type of annuity has been added to ProAdmin. This type of annuity provides a reduced benefit that is payable when both the member & beneficiary are alive and the beneficiary gets a specified percentage (usually 50%) of that amount if the member dies first, but the member's benefit pops up to the normal form amount if the beneficiary dies first.

Pop-up annuities can be converted based on either a table of factors or actuarial equivalence. Like Social Security Level Income forms, this form cannot be "unreduced".

ProAdmin is designed to handle pop-up annuities that are temporary or deferred, including the case where actuarial equivalence is calculated as of the deferral date. There is no provision for a certain period.

The pop-up annuity, when calculated, provides you with three (3) benefit pieces. The first is the benefit payable to the primary annuitant when both the primary annuitant and beneficiary alive. This benefit is reduced from the normal form based on the payment form parameters (e.g., beneficiary percentage, deferral, type of conversion, etc.). The second is the benefit payable to the beneficiary after the death of the primary annuitant. This benefit is equal to the beneficiary percentage of the primary benefit, possibly with an adjustment for monthly benefit rounding.. The third is the pop-up benefit and is equal to the normal form of benefit.

The screen snapshot below illustrates ProAdmin's summary results for a plan with a pop-up payment form.



A new optional field has been added to the XML Output Linkage to define where ProAdmin Server writes the pop-up benefit. A sample of this screen is shown below.