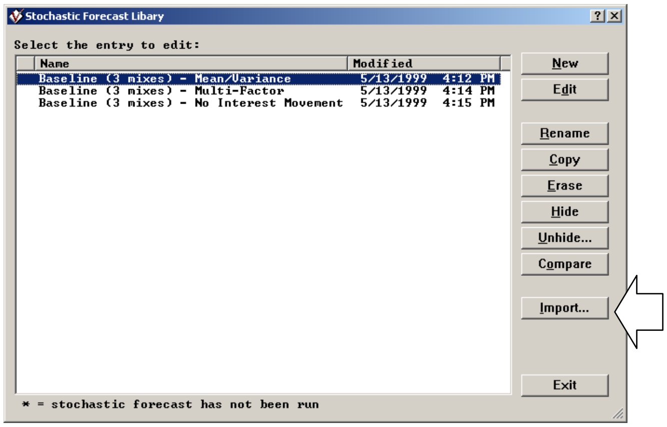
What's New! ProVal®

ProVal version 2.24 **November 2005**

ProVal version 2.24 introduces a robust **Import from Client** feature, **"if-then-else"** for ProVal expressions, and the ability to specify **service for decrement tables**. You'll find details about these and other enhancements below.

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, ProVal will import the Plan Definition plus any referenced Benefit Definitions, Benefit Formula Components, Payment Forms, etc. Look for the Import button in each library, or select Import from Client from the File menu. Import from Client replaces the (rather limited) Client to Client Copy command found in previous versions of ProVal.



Select the entry to edit:

Name	Modified
Baseline (3 mixes) - Mean/Variance	5/13/1999 4:12 PM
Baseline (3 mixes) - Multi-Factor	5/13/1999 4:14 PM
Baseline (3 mixes) - No Interest Movement	5/13/1999 4:15 PM

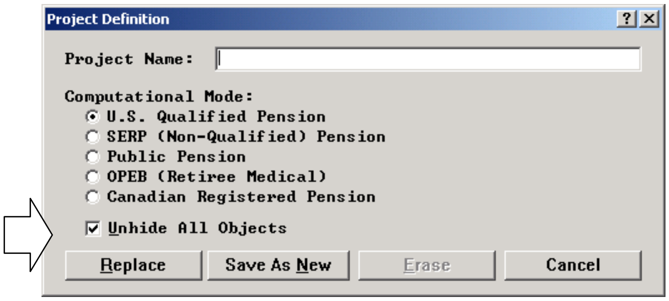
* = stochastic forecast has not been run

See Import from Client, Page 5

- ProVal now uses dynamic memory management to virtually eliminate WSFULL errors. The WSSize parameter in provalw.ini is now obsolete.
- All clients, even large ones, now open instantly.
- "If-then-else" syntax is now available to write ProVal 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 9

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



Project Name:

Computational Mode:
● U.S. Qualified Pension
○ SERP (Non-Qualified) Pension
○ Public Pension
○ OPEB (Retiree Medical)
○ Canadian Registered Pension

☑ Unhide All Objects

- The "RP-2000 Combined Mortality Table" has been added to the Mortality Rates library.
- The new ProVal Batch Server feature lets you execute runs (e.g., valuations, core projections, etc.) on other machines, freeing up your machine for other activities.

See Batch Server, Page 12

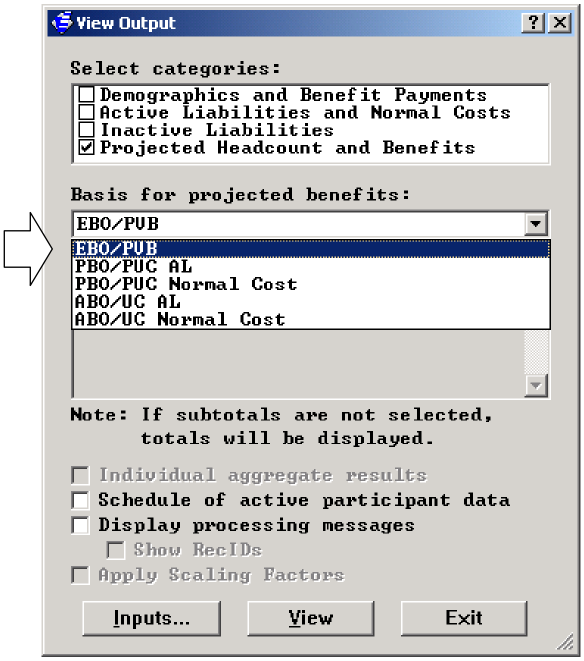
- In List Objects, client databases are now listed.

Pension Plans

- Projected benefit payments in a valuation can now be viewed on any of the following bases:
  - EBO / PVB (the previous behavior)
  - PBO / PUC liability
  - PBO / PUC normal cost
  - ABO / UC liability
  - ABO / UC normal cost

Page

(This was actually released as a special update to version 2.23 but is included here in case you missed it.)



View Output

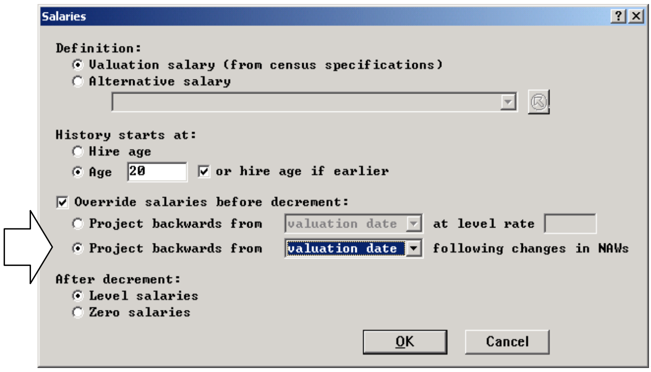
Select categories:
□ Demographics and Benefit Payments
□ Active Liabilities and Normal Costs
□ Inactive Liabilities
☑ Projected Headcount and Benefits

Basis for projected benefits:
EBO/PVB
EBO/PVB
PBO/PUC AL
PBO/PUC Normal Cost
ABO/UC AL
ABO/UC Normal Cost

Note: If subtotals are not selected, totals will be displayed.

□ Individual aggregate results
□ Schedule of active participant data
□ Display processing messages
□ Show RecIDs
□ Apply Scaling Factors

- There is a new option in the PIA custom operator and PIA tool to project salaries backwards from the *valuation date* following changes in the National Average Wage or at a level rate.



Salaries

Definition:
● Valuation salary (from census specifications)
○ Alternative salary

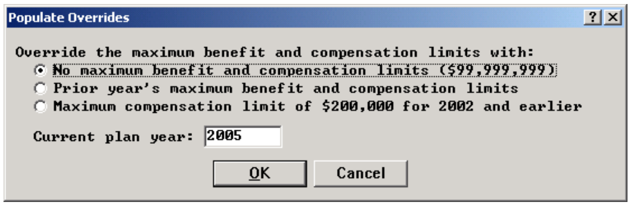
History starts at:
○ Hire age
● Age 20 ☑ or hire age if earlier

☑ Override salaries before decrement:
○ Project backwards from valuation date at level rate
● Project backwards from valuation date following changes in NAWs

After decrement:
● Level salaries
○ Zero salaries

.

- In Valuation Assumptions > Regulatory Data, a new Populate… button quickly sets overrides for common situations:
  - No maximum benefit and compensation limits.
  - Prior year's maximum benefit and compensation limits (for determining a plan change base).
  - Maximum compensation limit of $200,000 for 2002 and earlier.



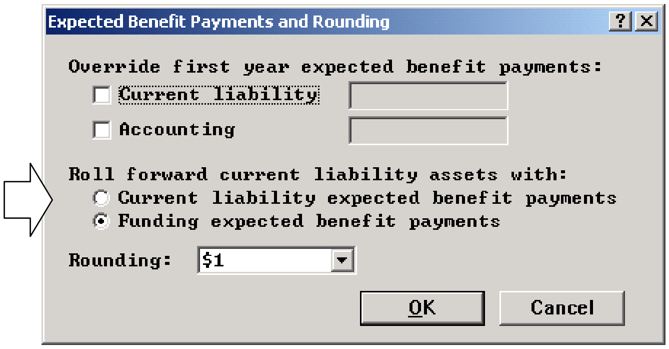
Populate Overrides

Override the maximum benefit and compensation limits with:
● No maximum benefit and compensation limits ($99,999,999)
○ Prior year's maximum benefit and compensation limits
○ Maximum compensation limit of $200,000 for 2002 and earlier

Current plan year: 2005

- In Asset & Funding Policies > Prior Year Values, a new option controls whether the in-payment/in-deferral splits for current liability and FAS 35 are based on status code (vested participants are considered in-deferral, all others are in-payment) or payment form commencement date.

- In Asset & Funding Policies > Expected Benefit Payments and Rounding, there is a new option to roll forward the assets used for the full funding limit calculations with funding expected benefit payments. If this option is selected, the same end of year asset value will be used for all full funding limit calculations.



Expected Benefit Payments and Rounding

Override first year expected benefit payments:
□ Current liability
□ Accounting

Roll forward current liability assets with:
○ Current liability expected benefit payments
● Funding expected benefit payments

Rounding: $1

OPEB Plans

- Service and Age-by-Service Benefit Component Tables can now be used in OPEB benefit formulas. The service lookup uses service at decrement (specified in census specifications for inactives) and the age lookup uses payment age.

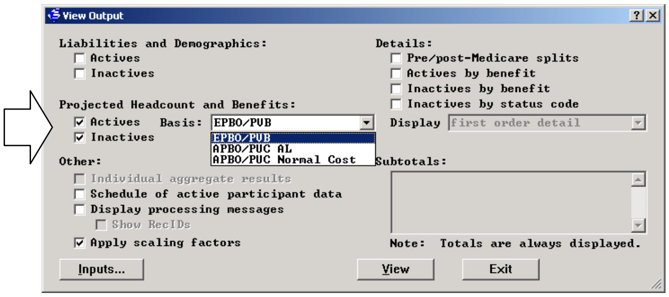
- In OPEB sample lives, payment form values can now be viewed for multiple ages at once.

- The Frozen Attained Age cost method has been added to OPEB mode with an underlying cost method of projected unit credit.

- Projected benefit payments in a valuation can now be viewed on any of the following bases:
  - EPBO / PVB (the previous behavior)
  - APBO / PUC liability
  - APBO / PUC normal cost

In addition, the projected headcount and benefits can now be viewed pre- and post-Medicare. When viewed by benefit, these splits are helpful in determining per-capita costs for Medicare Part D premiums.

(This was actually released as a special update to version 2.23 but is included here in case you missed it.)



View Output

Liabilities and Demographics:
- □ Actives
- □ Inactives

Projected Headcount and Benefits:
- ☑ Actives Basis: EPBO/PVB
- ☑ Inactives

Basis choices: EPBO/PVB, APBO/PUC AL, APBO/PUC Normal Cost

Other:
- □ Individual aggregate results
- □ Schedule of active participant data
- □ Display processing messages
  - □ Show RecIDs
- ☑ Apply scaling factors

Details:
- □ Pre/post-Medicare splits
- □ Actives by benefit
- □ Inactives by benefit
- □ Inactives by status code

Display: first order detail

Subtotals:

Note: Totals are always displayed.

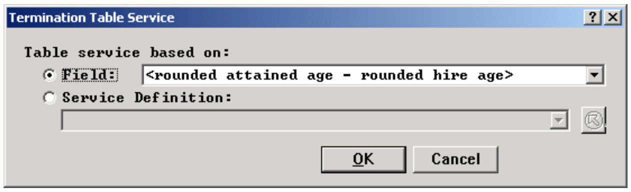
All Plans

- ♦ Scaling factors are now allowed to be any number, positive or negative. Also, new scaling factors have been added for number of actives, number of inactives, total salary, and the denominator of FAS average working lifetimes.

  (This was actually released as a special update to version 2.23 but is included here in case you missed it.)

  See Valuing Excess Plans, page 8

- ♦ The service used for decrement tables in Valuation and Projection Assumptions can now be specified. This might be useful, for example, if termination rates are age-by-service and termination service is not the same service as specified in the census specifications.



Termination Table Service

Table service based on:
- ● Field: <rounded attained age - rounded hire age>
- ○ Service Definition:

  See Service Options, page 6

- ♦ 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).

- ♦ In Valuation and Projection Assumptions, new "clear" buttons erase previous inputs for increase & crediting rates, post-decrement probabilities, and lump sum interest & mortality so they won't appear when viewing the assumptions.

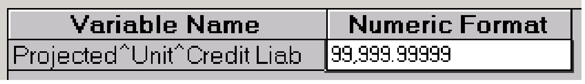
Valuation Sets, Deterministic & Stochastic Forecasts

- ♦ A new option in Asset & Funding Policies lets you establish 412 amortization bases based on "UAL + CB not less than zero".

Output & Reporting

- ♦ Valuations and Core Projections can now be subtotaled on more than one coded database field, such as by division and status. The output commands allow subtotals to be isolated from individual fields (e.g., just division) or viewed in combination (e.g., by status within division).

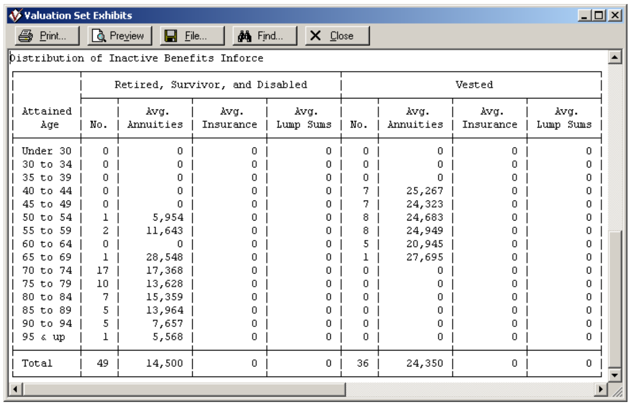
Did you know?
The number of decimal places displayed on output variables can be changed in the Format Options > Variables > Format Patterns screen. This might be useful, for example, to see why the sum of your subtotal results does not appear to (exactly) equal the total result.



Variable Name	Numeric Format
Projected^Unit^Credit Liab	99,999.99999

- ♦ The gateway funded current liability percentages for years -1, -2, and -3 have been added to the Additional Funding Requirement exhibit.

- ♦ In pension modes, a new Distribution of Inactive Benefits Inforce exhibit shows annuities, insurance, and lump sums inforce by age, with retired, survivor, and disabled inactives separated from vested inactives.



Valuation Set Exhibits

```
Distribution of Inactive Benefits Inforce

|          |      Retired, Survivor, and Disabled      |                 Vested                    |
|          |-------------------------------------------|-------------------------------------------|
| Attained |     |   Avg.    |   Avg.    |   Avg.    |     |   Avg.    |   Avg.    |   Avg.    |
|   Age    | No. | Annuities | Insurance | Lump Sums | No. | Annuities | Insurance | Lump Sums |
|----------|-----|-----------|-----------|-----------|-----|-----------|-----------|-----------|
| Under 30 |   0 |         0 |         0 |         0 |   0 |         0 |         0 |         0 |
| 30 to 34 |   0 |         0 |         0 |         0 |   0 |         0 |         0 |         0 |
| 35 to 39 |   0 |         0 |         0 |         0 |   0 |         0 |         0 |         0 |
| 40 to 44 |   0 |         0 |         0 |         0 |   7 |    25,267 |         0 |         0 |
| 45 to 49 |   0 |         0 |         0 |         0 |   7 |    24,323 |         0 |         0 |
| 50 to 54 |   1 |     5,954 |         0 |         0 |   8 |    24,683 |         0 |         0 |
| 55 to 59 |   2 |    11,643 |         0 |         0 |   8 |    24,949 |         0 |         0 |
| 60 to 64 |   0 |         0 |         0 |         0 |   5 |    20,945 |         0 |         0 |
| 65 to 69 |   1 |    28,548 |         0 |         0 |   1 |    27,695 |         0 |         0 |
| 70 to 74 |  17 |    17,368 |         0 |         0 |   0 |         0 |         0 |         0 |
| 75 to 79 |  10 |    13,628 |         0 |         0 |   0 |         0 |         0 |         0 |
| 80 to 84 |   7 |    15,359 |         0 |         0 |   0 |         0 |         0 |         0 |
| 85 to 89 |   5 |    13,964 |         0 |         0 |   0 |         0 |         0 |         0 |
| 90 to 94 |   5 |     7,657 |         0 |         0 |   0 |         0 |         0 |         0 |
| 95 & up  |   1 |     5,568 |         0 |         0 |   0 |         0 |         0 |         0 |
|----------|-----|-----------|-----------|-----------|-----|-----------|-----------|-----------|
| Total    |  49 |    14,500 |         0 |         0 |  36 |    24,350 |         0 |         0 |
```

- ♦ When saving exhibits to an Access database, data is now included for:
  - ○ Exhibits
    - ▪ Schedule of Employer Contributions

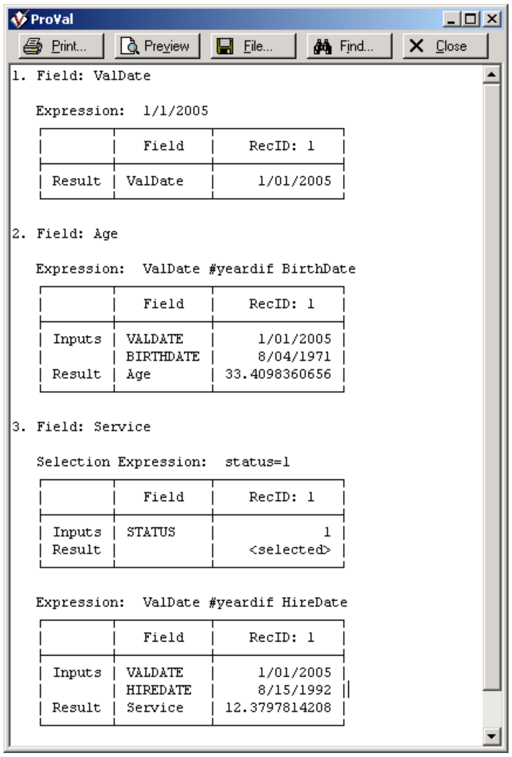
    - Interest on Late Quarterly Contributions
    - Schedule of Active Participant Data
    - Distribution of Inactive Benefits Inforce
  - Projected headcounts and benefits
  - Valuation interest rates

  For details, see Help > Help Topics > Command Reference > Output menu > Valuation Set Exhibits > Saving Exhibits to Access.

- The PBGC interest rate is now available under Output > Valuation Output > Inputs.

Census Data

- Sample lives are now available in Expression Sets so that you can check your formulas before running the Expression Set.



```
1. Field: ValDate

   Expression:  1/1/2005
   ┌────────┬────────────┬───────────────┐
   │        │   Field    │   RecID: 1    │
   ├────────┼────────────┼───────────────┤
   │ Result │ ValDate    │     1/01/2005 │
   └────────┴────────────┴───────────────┘

2. Field: Age

   Expression:  ValDate #yeardif BirthDate
   ┌────────┬────────────┬───────────────┐
   │        │   Field    │   RecID: 1    │
   ├────────┼────────────┼───────────────┤
   │ Inputs │ VALDATE    │     1/01/2005 │
   │        │ BIRTHDATE  │     8/04/1971 │
   │ Result │ Age        │ 33.4098360656 │
   └────────┴────────────┴───────────────┘

3. Field: Service

   Selection Expression:  status=1
   ┌────────┬────────────┬───────────────┐
   │        │   Field    │   RecID: 1    │
   ├────────┼────────────┼───────────────┤
   │ Inputs │ STATUS     │             1 │
   │ Result │            │    <selected> │
   └────────┴────────────┴───────────────┘

   Expression:  ValDate #yeardif HireDate
   ┌────────┬────────────┬───────────────┐
   │        │   Field    │   RecID: 1    │
   ├────────┼────────────┼───────────────┤
   │ Inputs │ VALDATE    │     1/01/2005 │
   │        │ HIREDATE   │     8/15/1992 │
   │ Result │ Service    │ 12.3797814208 │
   └────────┴────────────┴───────────────┘
```

- To assist in auditing database changes, the date and time of database changes made in Spreadsheet Edit or Individual Record Edit is now recorded (this was always recorded for other types of changes). The name of the user who made the change is also included.

- In Frequency Tables, scaled counts can now be seen for grouped data (i.e., headcounts as opposed to record counts).

Changes Log

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

WinTech

500 West Putnam Avenue
Greenwich, CT 06830

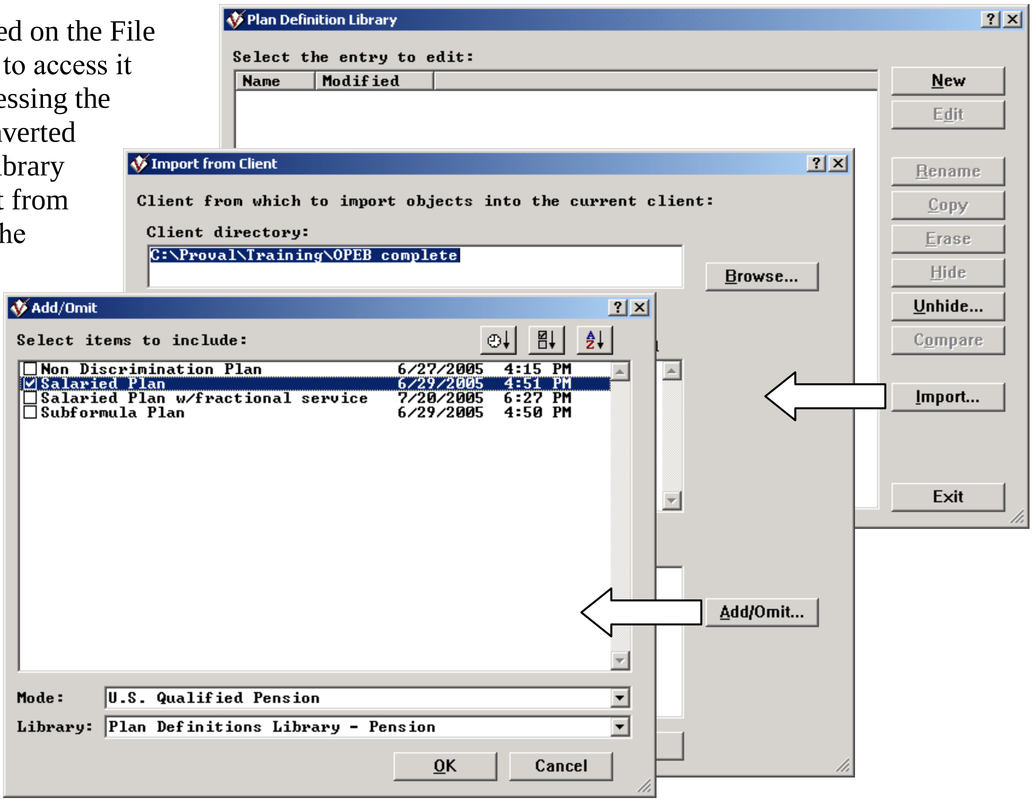
tel: (203) 861-5530
fax: (203) 861-5531
email: support@winklevoss.com
website: www.winklevoss.com

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, ProVal will import the Plan Definition plus any referenced Benefit Definitions, Benefit Formula Components, Payment Forms, etc. If ProVal 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.
- Distributed processing. Suppose you have 4 core projections for client X which each take 8 hours to run. In addition, suppose it is 5pm and you need the results by 9am. After making 3 additional copies of client X, you can set core #1 to run on computer #1, core #2 to run on computer #2, and so on. Then, you can use Import from Client (3 times) to import cores #2, #3, and #4 back into the original client.
- Approximating concurrent multi-user access. Suppose John and Mary are both working on client X. John is working on the OPEB Plan Definition and Valuation Assumptions; Mary is setting up the pension Plan Definition and Valuation Assumptions. John and Mary can work on separate copies of client X and then use Import from Client to put their work together.



Select the entry to edit:

Client from which to import objects into the current client:

Client directory: C:\Proval\Training\OPEB complete

Select items to include:

☐ Non Discrimination Plan	6/27/2005	4:15 PM
☑ Salaried Plan	6/29/2005	4:51 PM
☐ Salaried Plan w/fractional service	7/20/2005	6:27 PM
☐ Subformula Plan	6/29/2005	4:50 PM

Mode: U.S. Qualified Pension

Library: Plan Definitions Library - Pension

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 ProVal. 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 (e.g., extraneous output styles might be created).

Service Options

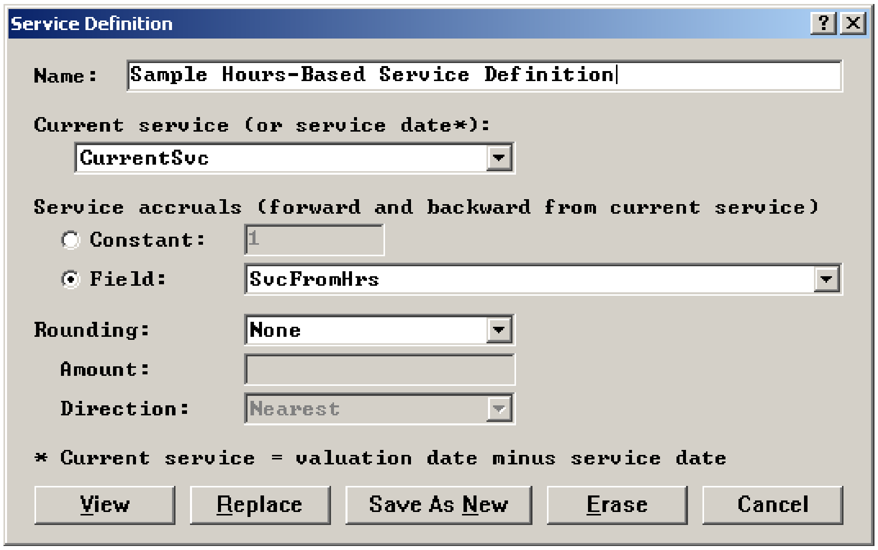
ProVal offers considerable flexibility in how service is measured for different purposes. If needed, you can create and reference distinct service fields or Service Definitions for different benefit eligibilities, accrual rates, attribution, or in almost any other place where service is used. Here, we point out some recent additions to the options for handling service in ProVal.

Service Definitions

Service Definitions allow you to measure yearly service accruals on a basis other than simple elapsed time. For example, in a plan with an hours-based service formula, some employees may earn less than one year of service in each plan year. Or, there may be rounding rules in the plan that limit the measurement of service to whole years or to integer months, quarters, or other fractions of a year. Service Definitions enable you to handle these cases.

Using a Service Definition is optional, and not all plan designs will call for one. Instead, service can be specified by referencing a database field containing a service amount or date (e.g. date of hire). ProVal will then count service at a rate of 1 year per year of elapsed time, with no rounding.

Service Definitions are accessed through Input > Census Specifications > Service Definitions. Within the definition, you will be asked to specify current service at the valuation date, the yearly rate of service accruals, and the rules for rounding.


Service Definition

Name: Sample Hours-Based Service Definition

Current service (or service date*):
CurrentSvc

Service accruals (forward and backward from current service)
○ Constant: 1
● Field: SvcFromHrs

Rounding: None
Amount:
Direction: Nearest

* Current service = valuation date minus service date

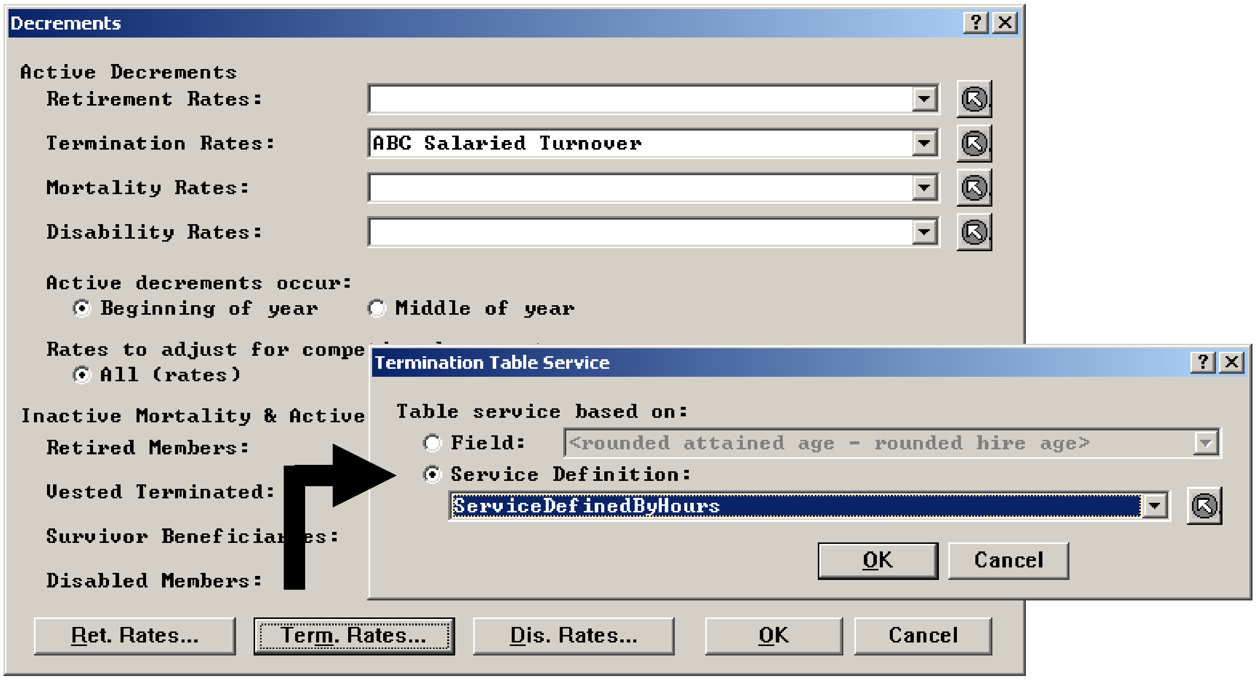
Service accruals (forward and backward from current service) defines, as a fraction of a year, how much service accrues each year. It may be specified as a constant that applies to all active participants or as a database field. The parameter value is often a constant of 1. However, in an hours-based plan or other plan where part-timers earn partial years of service, a database field for service accruals may be more appropriate, allowing you to specify a different rate of accruals for each participant. The rate must be expressed in terms of decimal years. So, if the database contained hours for each participant, you would need to convert hours to a yearly accrual fraction using Define Field by Expression or Census Data > Data Defaults.

Rounding allows you to handle plans in which service is recognized in fixed quantities such as whole years, months, or quarters. For example, in a plan which counts partial years as full years, you may want to create a Service Definition and set Rounding = Year, and Direction = Up.

Once you have created a Service Definition, it can be used almost anywhere ProVal asks for service, including benefit eligibilities, accrual rates, and attribution (e.g. for Projected Unit Credit). Now in ProVal 2.24, you can also select a Service Definition to specify the service to be counted for service-related decrements.

Service for Service-Related Decrements

New in version 2.24 is the ability to specify the service that is to be used for service-related decrements. Within Input > Valuation Assumptions > Decrements, a button for each of the Active Decrements is available at the bottom of the dialog box. Clicking one of the buttons brings you to another dialog that allows you to enter additional parameters. Here, you can indicate what service is to be considered in determining a participant's position on a decrement table. The service may be specified by a database field or by a Service Definition.



Decrements

Active Decrements

Retirement Rates:

Termination Rates: ABC Salaried Turnover

Mortality Rates:

Disability Rates:

Active decrements occur:
● Beginning of year ○ Middle of year

Rates to adjust for compe
● All (rates)

Inactive Mortality & Active

Retired Members:

Vested Terminated:

Survivor Beneficiaries:

Disabled Members:

Termination Table Service

Table service based on:
○ Field: <rounded attained age - rounded hire age>
● Service Definition: ServiceDefinedByHours

Valuing Excess Plans

There are three methods to value an Excess plan within ProVal, where an Excess Plan is defined as a plan that replaces benefits that are reduced by U.S. Qualified pay or benefit limits (e.g. unlimited benefits minus limited benefits). These plans are sometimes referred to as Restoration Plans. This article assumes that the U.S. Qualified plan is already set up in ProVal.

Method 1 – Code the excess benefit in SERP Mode (this is the preferred approach)

How To: Set up a Plan Definition in SERP mode that directly values the excess benefit. The operators #MAXBEN and #MAXSAL will be useful.

For example, suppose your plan's formula is 1.5% x (final average 5 of 10) x service. The steps to code the excess benefit in SERP mode are:

1. Create a custom operator, #FASLimited, that limits each salary to the IRC 401(a)(17) maximum compensation.
2. Create two accrual basis components, FAPUnlimited and FAPLimited, with rates of 1.5% for all years of service, and basis formulas of (5 #FAS 10) and (5 #FASLimited 10), respectively.
3. Write your benefit formula as the following where x is the age to which the benefit is deferred:
   FAPUnlimited #zminus (FAPLimited #min #MaxBen x)
4. Use this Plan Definition for your Valuation and/or Core Projection.

Limitations: None

Advantages: All other features in ProVal's SERP Mode are available and will produce correct results.

Method 2 - Subtraction outside of ProVal

How To:

1. Set up a valuation or core projection in ProVal's SERP mode with the same Plan Definition and assumptions as run in U.S. Qualified mode (Valuation and Projection Assumptions identical to those run in U.S. Qualified mode can be created by using the Populate button).
2. Run the valuation or core projection. The result will be the unlimited liabilities of the Plan.
3. Export these results and the results from the U.S. Qualified valuation or core to an outside source and subtract the liabilities.

Limitations: Since ProVal does not know who has a non-qualified benefit, the future working lifetime from both valuations will contain all participants. The subtracted liabilities are not available to use in any other feature in ProVal's SERP Mode (such as valuation sets, forecasts, or exhibits).

Advantages: Simple and almost no setup time required in SERP Mode.

Method 3 – Negative Scaling Factors

How To:

1. Create a copy of your U.S. Qualified mode Valuation Assumptions except override the historical limits with very large numbers (e.g. 99,999,999).
2. Set up and run a valuation or core projection, Run A, with all inputs identical to your U.S. Qualified valuation or core, except reference this new set of valuation assumptions. This run will produce unlimited liabilities.
3. Set up scaling factors where all items are -1 except the total salary, number of actives, number of inactives, and future working lifetime numerator and denominator, for which the scaling factors are 0.
4. Set up another valuation or core, Run B, that is identical to your U.S. Qualified valuation or core but references these scaling factors.
5. Aggregate Run A and Run B in Valuation Sets, Deterministic Forecasts, Stochastic Forecasts and the Output menu. This will subtract the limited liabilities from the unlimited liabilities.

Limitations: The resulting future working lifetime will be for the entire population, not just for those eligible for an excess benefit. This may or may not be material.

Advantages: Easy set up. All other features in ProVal's U.S. Qualified Mode are available.

If-Then-Else

Expressions are used within ProVal to:

- define new database fields;
- specify subsets of a database (selection expressions); and
- define benefit formulas.

ProVal version 2.24 introduces if-then-else operators to help you write clearer and more manageable expressions.

Multiplying by Boolean Values

Prior to version 2.24, 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:

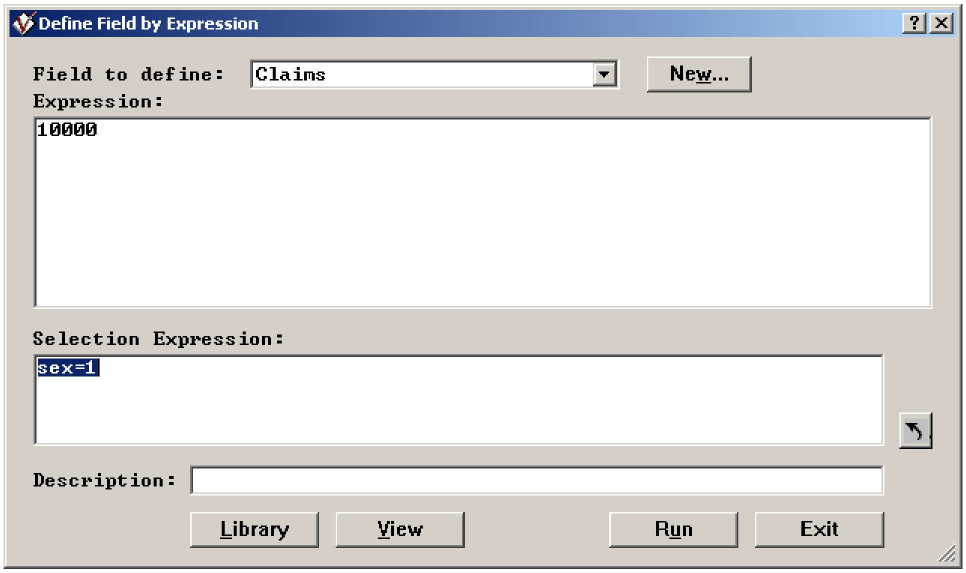
```
service := ValDate #YEARDIF DOH  &
age     := ValDate #YEARDIF DOB  &

#IF(plan=1) #THEN
  ; RET ELIGIBILITY BUCKET FOR SALARIED PARTICIPANTS
  #IF (age >= 65) #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 set the `Claims` field for males to 10000 while leaving it unchanged for females, use the expression "`10000`" along with the Selection Expression "`Sex=1`".



Field to define: Claims

Expression:

10000

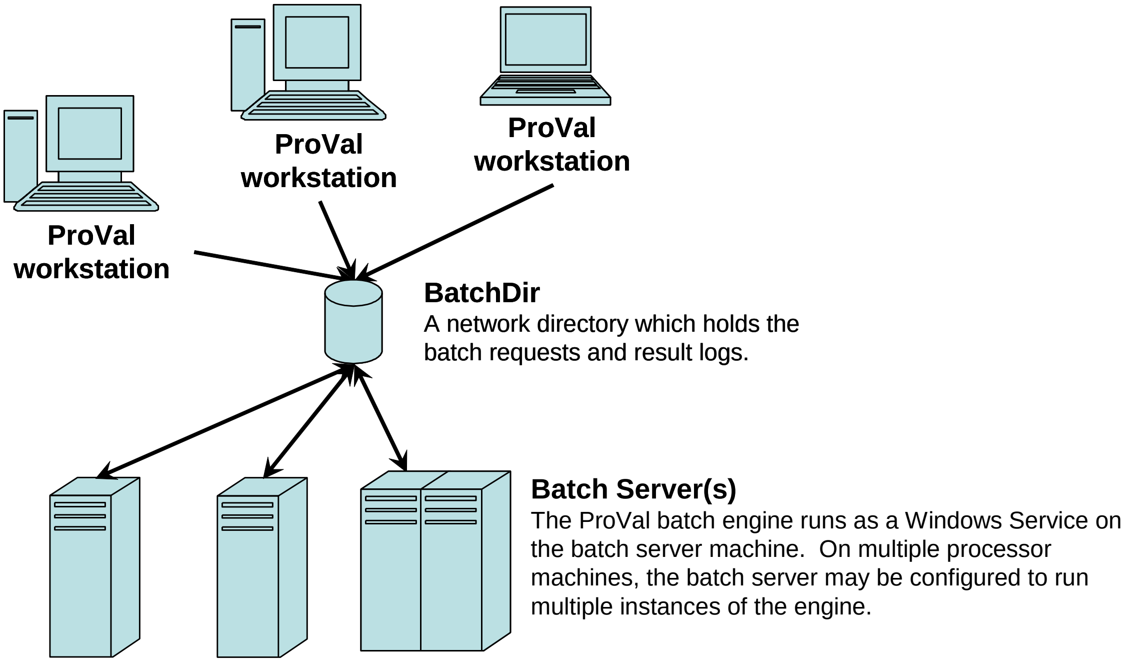
Selection Expression:

sex=1

Description:

By contrast, using the expression "`#IF sex=1 #THEN 10000 #ENDIF`" with no selection expression will set the Claims field to 10000 for males and `#MV` (missing value) for females – overriding any existing values.

Batch Server


ProVal workstation
ProVal workstation
ProVal workstation
BatchDir
A network directory which holds the batch requests and result logs.
Batch Server(s)
The ProVal batch engine runs as a Windows Service on the batch server machine. On multiple processor machines, the batch server may be configured to run multiple instances of the engine.

The new ProVal Batch Server feature lets you execute runs (e.g., valuations, core projections, etc.) on other machines, freeing up your machine for other activities. To submit runs to a ProVal Batch Server, select the runs in Batch Execution just as you always have, then check the "Submit to Batch Server" box and press the Run button. Also, exit the client (or open another client) so that a ProVal Batch Server can have access to your client.

Setting it Up

Each Batch Server machine needs to use a Windows 2000, XP, 2000 Server or 2003 Server operating system. A ProVal Batch Server runs as a Windows Service on the machine. This design improves the security, reliability and scalability of running ProVal on the machine. One advantage of being set up as a Windows Service is that the Batch Server can be automatically started when the machine is powered on without requiring a user logon.

The link between each ProVal workstation and the Batch Servers is a directory on a common file share. This directory is specified by a new BatchDir parameter in each workstation's and Batch Server's PROVALW.INI file. For details, see Batch Server Installation Guide.pdf in the ProVal folder.

How it Works

The ProVal workstations write each request for a batch job to the BatchDir directory. The servers scan the directory for pending requests. If a request is found, it is then passed to the first available ProVal engine. The ProVal batch engine opens the ProVal client and begins execution of the job. The ProVal engine must wait until it can get exclusive access to the client files.

Within each batch request file is the path to the ProVal client files. Paths using mapped drives will be converted to use UNC paths. This avoids the need for the Batch Server to maintain the same drive mappings as all the workstations. In addition, the request file will have a requested start time for each job, allowing jobs to be deferred (presumably to low volume times of the day).

Batch requests will be processed on a first come basis. If a job completes or aborts, the request file will be removed from the BatchDir and the processing messages will be written to a log file. If the client files needed for the request are currently in use by another user, the server will not delete the request but wait and try again later.