WinTech

ProVal®

Getting Started with ProVal:
How to Convert a Plan

Valuation and projection software for both pension & OPEB plans

Version 3.19

Version 3.19

CONTENT

INTRODUCTION

OVERVIEW

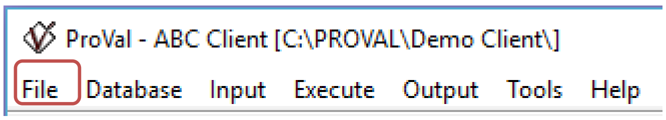
This book provides step-by-step instructions for "booting up" plans with ProVal. The focus is on annual valuations, starting with data and ending with contributions and expense. After running a valuation, see Appendix E: Forecasts for information on running forecasts. Both pension and retiree medical plans are covered.

The major steps in the process are:

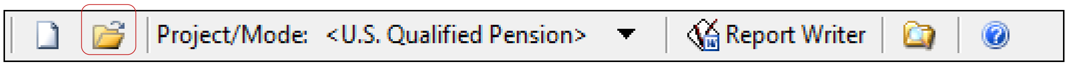
	1.	Client	*Open or create a ProVal client*
	2.	Census Data	*Import birthdates, benefits, etc.*
	3.	Plan Benefits	*Enter plan provisions*
+	4.	Valuation Assumptions	*Enter interest rates, salary scale, etc.*
	5.	Valuation	*Calculate liabilities and normal costs*
+	6.	Asset & Funding Policy	*Enter assets and amortization bases*
	7.	Valuation Set	*Calculate contributions and expense*

PERFORMING THE STEPS

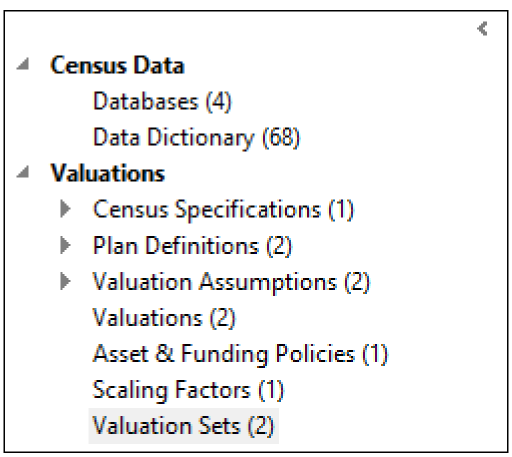
Step 1: Use the **File** command on the Menu bar to open a template client or create a new client.



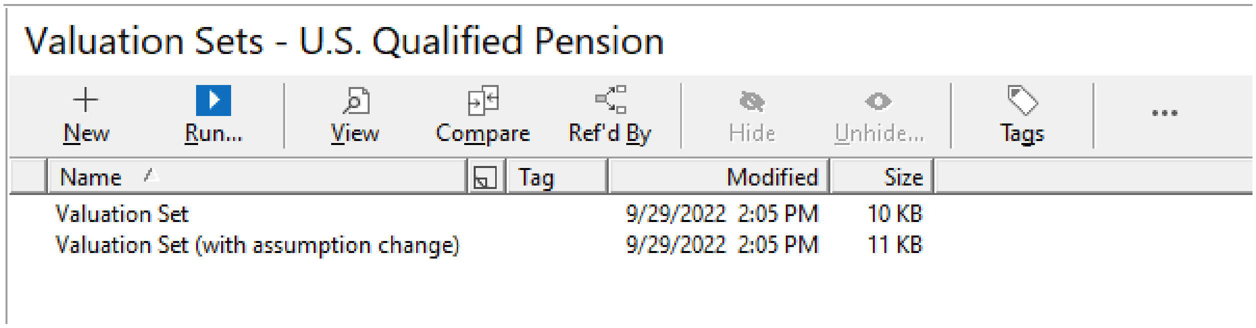
Or, alternatively, click the folder icon on the main toolbar to open a template client:



Steps 2-7 create objects which are saved in libraries. Each library can be accessed through a command on the menu bar (**Input**, **Execute,** or **Tools**). Libraries found under these menus can also be accessed through the **Shortcuts** pane on the left (the highlighted entry on the Shortcuts pane is the library which is currently open):

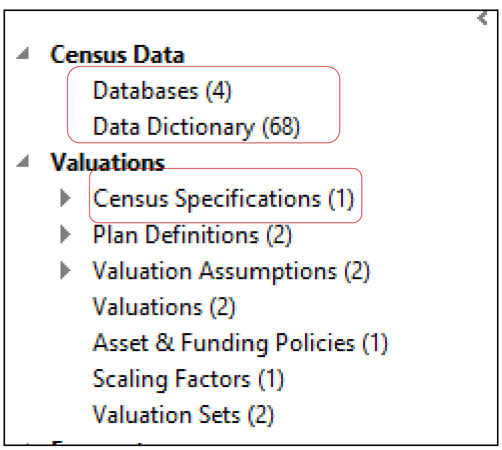


The **Entries** pane on the right displays the contents of the selected library. The command bar for the **Entries** pane contains buttons for several library entry operations, including New, Run, View, Compare, Ref'd by, Hide, Unhide and Tags. Additional tools may be found under the "…" button and the toolbar may be customized for the user's preferences.

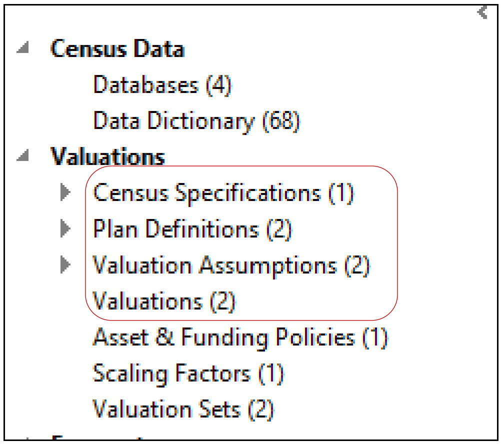


For more information regarding libraries and the Entries pane, see Appendix B: Libraries

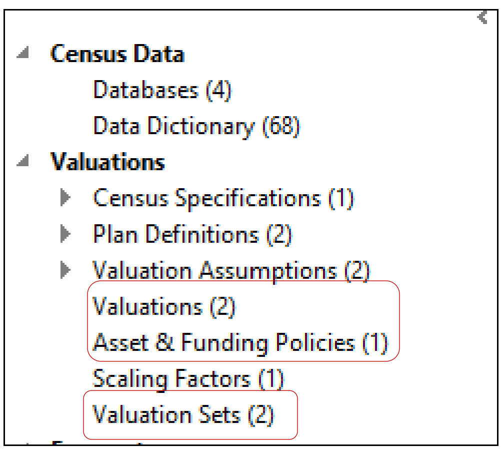
<u>Step 2</u>: Import data, define fields in the **Data Dictionary**, and create the **Census Specifications** by using the **Database** and **Input** commands on the Menu bar, or alternatively using the Shortcuts pane:



<u>Steps 3-5</u>: Create a **Plan Definition** and set of **Valuation Assumptions** that, when combined with the **Census Specifications** created in Step 2, are used to calculate liabilities and normal costs in a **Valuation**. These steps can be performed using the **Input** and **Execute** menus or, alternatively, using the Shortcuts pane:



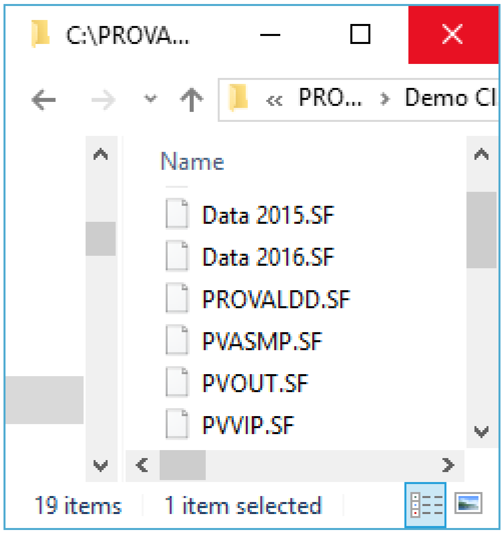
<u>Steps 6-7</u>: Create an **Asset & Funding Policy** that is used to calculate funding contributions and accounting expense in a **Valuation Set** when combined with the **Valuation** created in step 5. These steps can be performed using the **Input** and **Execute** menus or, alternatively, using the Shortcuts pane:



STEP 1: OPEN A CLIENT

Your work in ProVal is stored in “clients”. To open an existing client, use **File | Open Client**; to create a new client, use **File | New Client**.

A ProVal client is a directory, for example C:\PROVAL\Demo Client, containing:



Database files storing birthdates, salaries, benefits, etc. for the plan participants. The database file names are specified by the user, but always end with the extension .sf.

Library files storing the client’s benefit definitions, valuation assumptions, valuation results, etc. The file names are always provaldd.sf, pvasmp.sf, pvout.sf, and

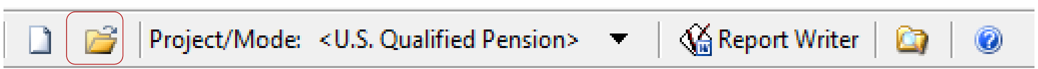
Note: Since the library filenames are always the same, each ProVal client must be kept in a separate directory.

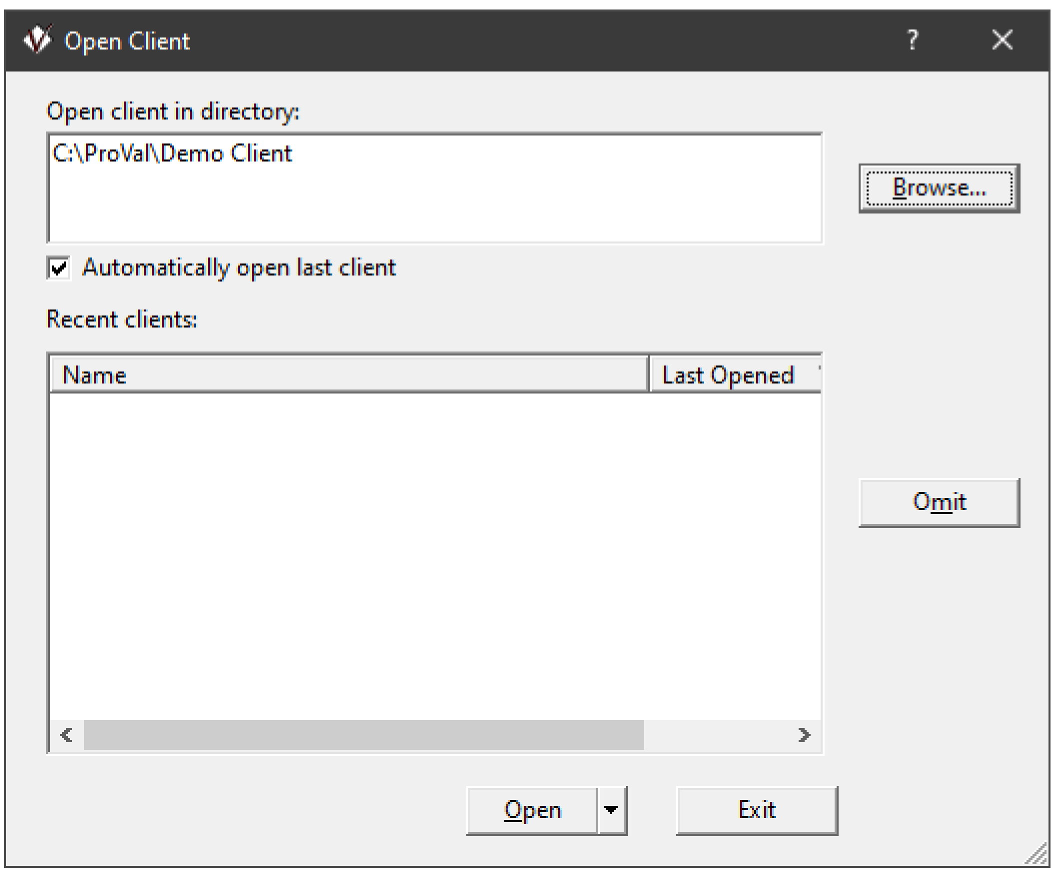
ALTERNATIVE 1: OPEN A TEMPLATE CLIENT

If you have a template client that contains standard field names, record layouts, tables, output styles, etc., follow these steps. Otherwise, follow the instructions below for Alternative 2: Create a New Client.

1. In Windows Explorer, make a copy of the template client folder. For example, copy C:\CLIENTS\TEMPLATE to C:\CLIENTS\ABC

2. From ProVal's **File** menu, choose **Open Client**. Alternately, click the folder icon on the main toolbar:

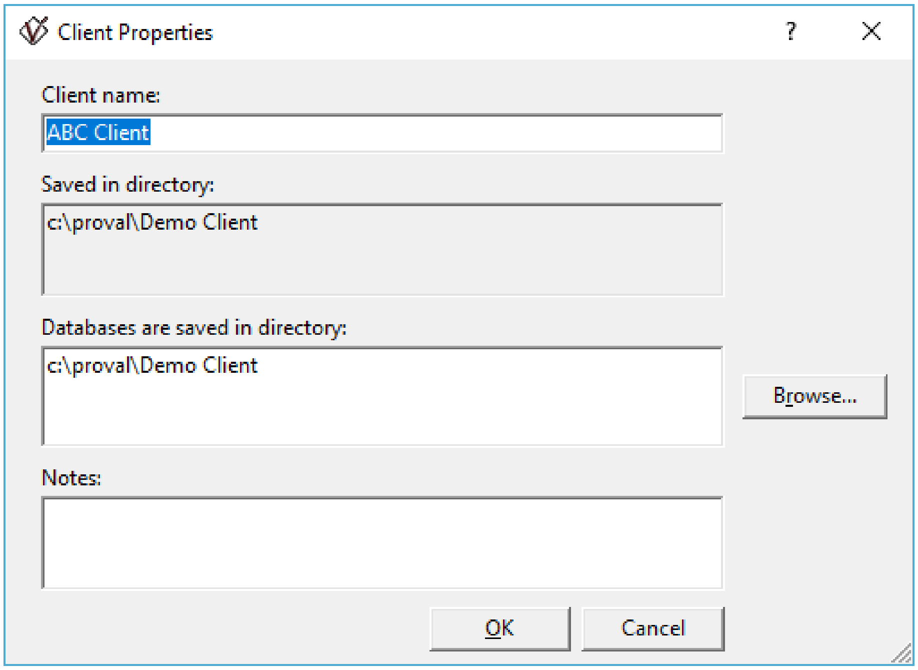




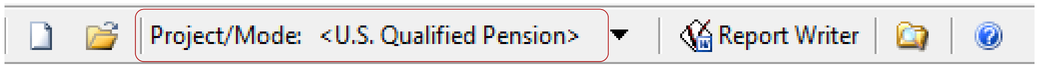
Browse to (or type in) the new client **directory** you created in step 1 and click **Open**.

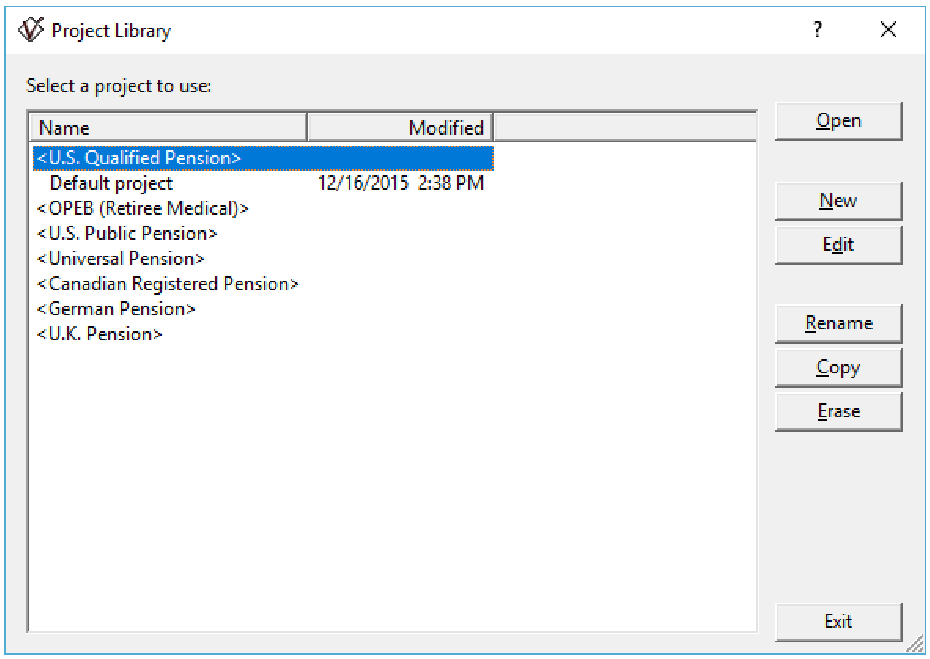
If a message appears saying this client “was developed under a previous version of ProVal,” click **Yes** to update the client.

3. From the **File** menu, choose **Properties**. Change the **Client name** and click **OK**.



4. From the **File** menu, choose **Change Project/Mode**. Alternately, click the arrow next to **Project/Mode** on the main toolbar.





ProVal can function in one of seven possible modes:

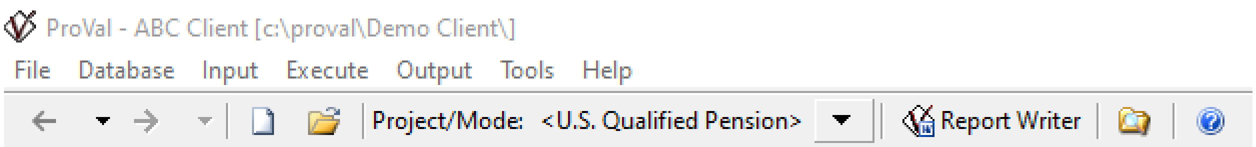
- U.S. Qualified Pension,
- Universal Pension,
- U.S. Public Pension,
- OPEB (Retiree Medical),
- Canadian Registered Pension,
- German Pension, and
- U.K. Pension.

Each pension mode specializes in a unique set of legislative topics for defined benefit pension plans. The one non-pension mode focuses on post-retirement medical and life insurance benefits. Menu and input items vary by the specific topics covered in each mode. Therefore, it is important to make sure that all work is completed in the appropriate mode.

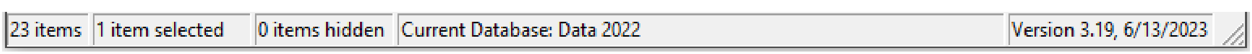
User defined **Projects** may be created within each mode. You may double-click (or highlight then click **Open**) to switch to a different mode or project.

For more information, see Appendix C: Projects.

5. The client name and folder should appear in the caption of the main ProVal window; while the project appears in the toolbar.

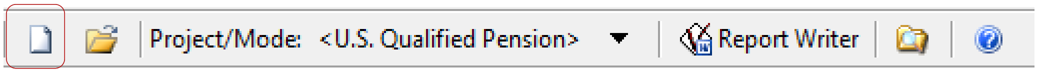


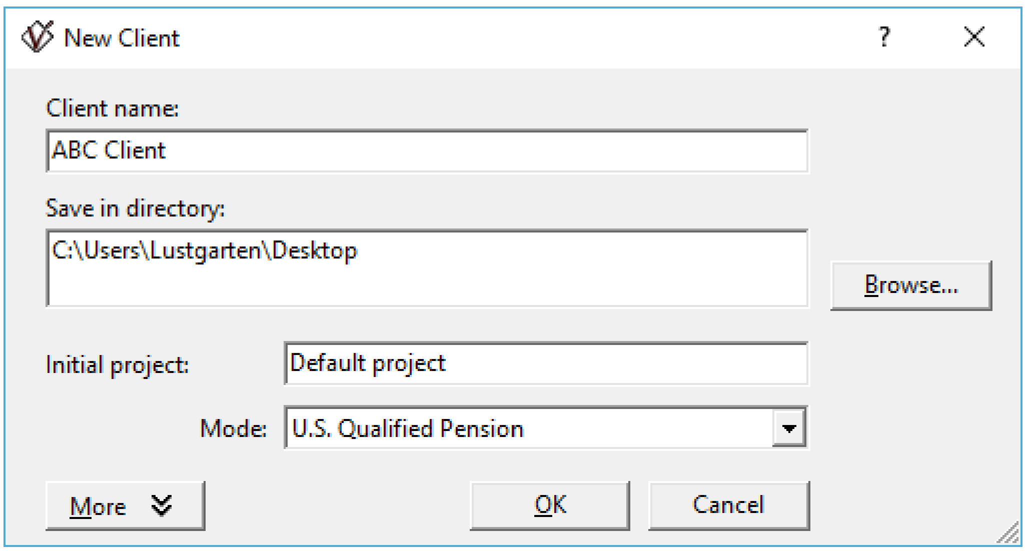
The current database, if any, version number and version date should appear in the status bar at the bottom of the ProVal window. You can begin to work and everything will be saved in your new client folder.



ALTERNATIVE 2: CREATE A NEW CLIENT

1. From ProVal's **File** menu, choose **New Client**. Alternately, click the folder icon on the main toolbar:





2. Provide a Name and Directory for the client.

   For example, "ABC Client" in directory C:\PROVAL\Demo Client. It is ok to provide a directory that does not exist; ProVal will create it for you.

   If you wish, you can store a client's associated databases in a different directory by pressing the **More** button. This is a matter of personal preference; there is no inherent advantage to doing this.

3. Provide a name and computational **Mode** for your **Initial project**. For example, you might create the "1/1/2022 Pension Valuation" project in U.S. Qualified Pension mode. ProVal can function in one of seven possible modes:

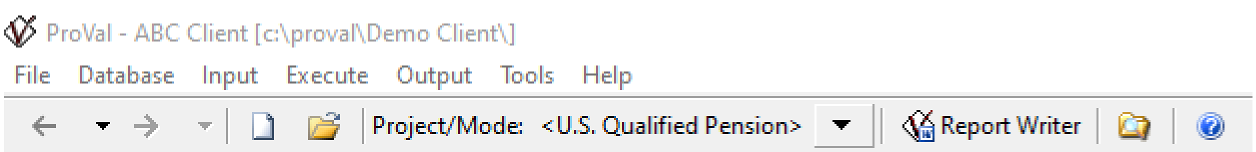
    - U.S. Qualified Pension,
    - Universal Pension,
    - U.S. Public Pension,
    - OPEB (Retiree Medical),
    - Canadian Registered Pension,
    - German Pension, and
    - U.K. Pension.

    Each pension mode specializes in a unique set of legislative topics for defined benefit pension plans. The one non-pension mode focuses on post-retirement medical and life insurance benefits. Menu and input items vary by the specific topics covered in each mode. Therefore, it is important to make sure that all work is completed in the appropriate mode.

    For more information, see Appendix C: Projects.

4. Click **OK**.

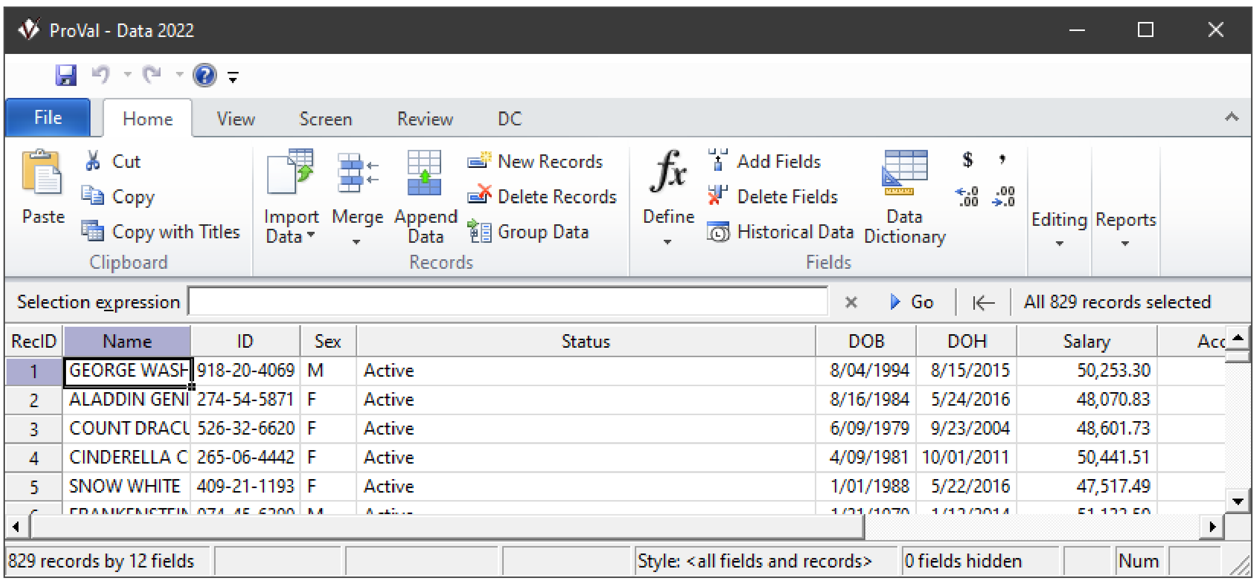
    Once the client file is initialized and opened, the name and folder of the client should appear in the caption of the main ProVal window, while the project appears in the toolbar:



The current database, if any, version number and version date should appear in the status bar at the bottom of the ProVal window. You can begin to work and everything will be saved in your new client folder.

STEP 2 CENSUS DATA

Conceptually, a ProVal database is a large table of values. Each row of the table is called a **record** and generally contains information for a particular person or group of persons. Each record is identified by a unique number called the **record ID** (referenced in ProVal as the field "RecID"), which is automatically generated at the time the data is read into ProVal. Each column of the table is called a **field** (or **database field**) and contains a specific kind of information, such as age, sex, or employee name. Each field is identified by a descriptive name given by the user. The database may have any number of records and fields.



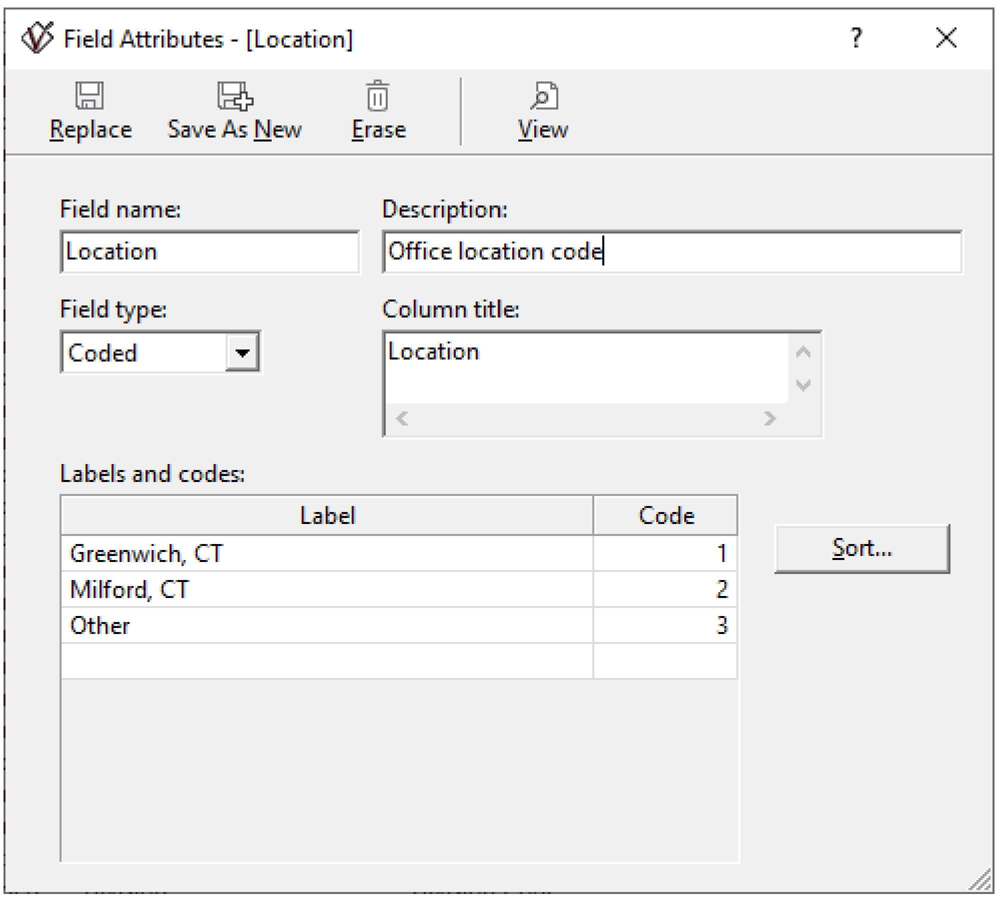
The database is stored in a file having a special format that is usable only by ProVal (however, the database can be easily copied to excel). You may have any number of database files. You may open a database file from the **Databases** command on the **Shortcut** pane by highlighting the desired database file and clicking **Edit**, or by double-clicking the desired file. Once a database file is open, you are viewing **Spreadsheet Edit** (as in the screen shot above). From within Spreadsheet Edit, you may switch between database files using the **File | Open** command. At any time, one file is selected as the current database; the name of this file is shown in the status bar at the bottom of the screen.

DATA DICTIONARY

For each database field, ProVal stores information about the field in what is known as the **data dictionary**.

The data dictionary contains the definition of every database field used by a ProVal client. Note that there is only *one* data dictionary used by *all* databases stored under a given *client*. This ensures that all databases within a client have a compatible field structure. If the data is saved as an Excel workbook and contains a header row in which each header conforms to ProVal's requirements for a field name, then ProVal can auto-create fields during import. Otherwise, before any data can be imported into a database, fields must be defined using the **Database | Data Dictionary** command or from **Spreadsheet Edit** using the **Home | Data Dictionary** command.

An example of a coded field type is illustrated below.



Each field in the data dictionary contains the following attributes:

- **Field Name**: The field name may contain letters, digits, and underscore (_) symbols, but it may not contain other symbols or blanks, and it must begin with a letter. Although field names are displayed in upper and lowercase letters, case is not significant when matching names.

- **Description**: In addition to the field name, you can provide a one-line description as a reminder as to what the field represents. The description may include any characters, including spaces.

- **Field Type**: ProVal allows for five field types:

1. **Numeric:** A single number, such as an employee's salary. This can be either an integer or decimal number.

2. **Character**: A character string, such as the employee's name. The string may have any number of characters and may include spaces.

3. **Coded**: A field with distinct values or categories. Each value is identified by both a character *label* and a numeric *code.* For example:

Sex

Label	**Code**
Male	1
Female	2

Location

Label	**Code**
Greenwich, CT	1
Milford, CT	2
Other	3

When a coded field is displayed, the character labels are shown. When writing an expression, e.g., “Location #in (1,3)”, the numeric codes are used.

Note that the same labels may not necessarily be used in the source data that you will import. For example, if your client provides location codes “G”, “M”, and “O” in the source data file, you may map them into the codes and labels shown above.

4. **Date:** A date field represents a year, month, and day, such as an employee's date of birth.

5. **Social Security Number:** A 9-digit integer representing a Social Security number (or other Identifier). ProVal stores the value imported but displays this type of field using a format you supply. You may use X in your supplied format to obscure some of the digits, as illustrated in the examples on the screen.

- **Column Title**: The title that will be used when the field is displayed in an output table. The default is field name.

- **Formatting Style**: For numeric fields, you can specify a format to tell ProVal how you want the number displayed. The format is an example of how the number should appear in output, such as “$9,999.99.” Your example is used to determine the following parameters:

  - How many columns to allow for display of the field
  - How many digits are displayed to the right of the decimal point
  - Whether or not to insert commas in the number
  - Whether to put a dollar sign to the left of the number

- **Display Width**: For character fields, you can specify how many characters should be used to display the value. More characters may be stored. This only affects the display on the screen.

ProVal has 36 standard fields:

- CAPctMale (used for grouped data)
- Count (used for grouped data)
- PctMale (used for grouped data)
- RecID (used in each database to track import order)
- yFcstYear (used for yield curve forecasting)
- yTrial (used for yield curve forecasting)
- yDur01 through yDur30 (used for yield curve forecasting)

Otherwise, ProVal’s database is free form. That is, you can give any name you wish to your data fields (e.g., DateOfBirth, BirthDate, Dob, etc.)

Listed below are sample field names that are commonly encountered. Keep in mind, ProVal databases are free form so the fields listed below are just samples. If you choose, you can establish your own naming conventions.

CASE 1: IMPORTING COMPLETE DATA

Use these steps to import complete data, that is, data that includes all plan participants. If instead you have data updates, i.e., just the changes from last year's population, follow the instructions below for Case 2: Importing Data Updates. Note: all of the following steps may also be done from **Database | Edit Data | Spreadsheet Edit** as shown below in parentheses.

1. If it isn't already, save the data as either an Excel file or a text file in a fixed width (fields are aligned in columns) or delimited format.

2. Create a new database using the **Database | Current Database | New** command (or from **Spreadsheet Edit** using **File | New**).

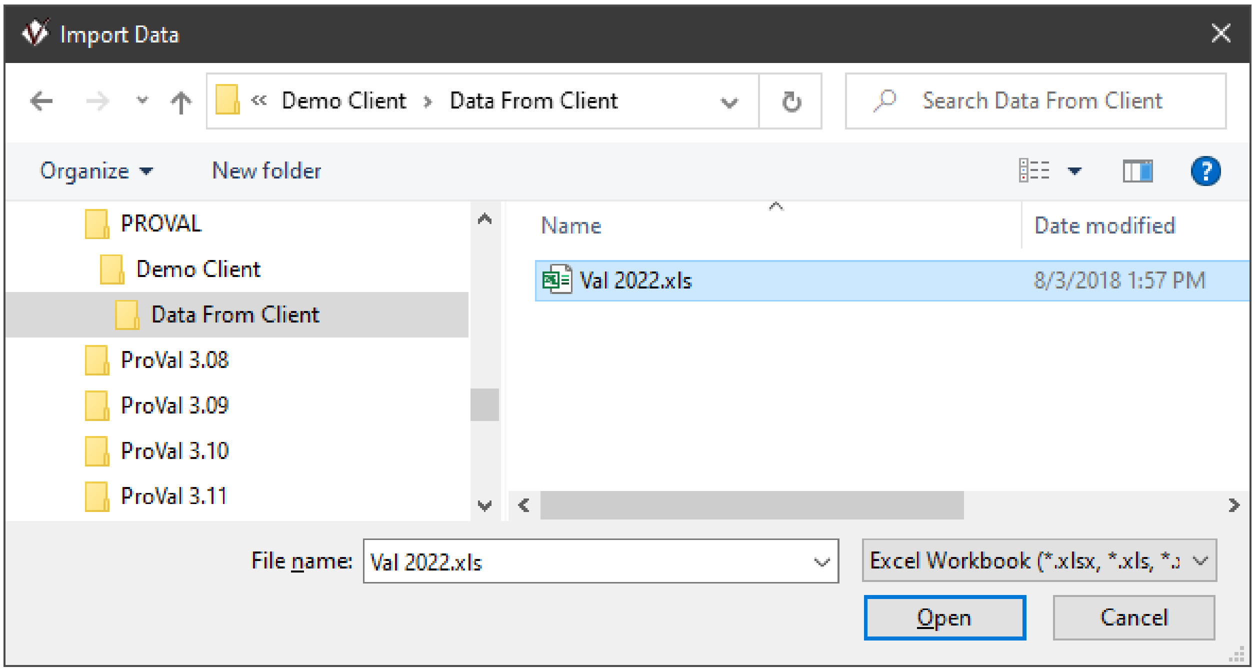
   Alternatively, you may open an existing database from the **Databases** command on the **Shortcut** pane by highlighting the desired database file and clicking **Edit**, or by double-clicking the desired file. From within **Spreadsheet Edit**, you may switch between database files using the **File | Open** command.

3. *Optional:* Create field names using the **Database | Data Dictionary** command (or from Spreadsheet Edit using **Tools | Data Dictionary**). Field names can also be defined as part of the import process.

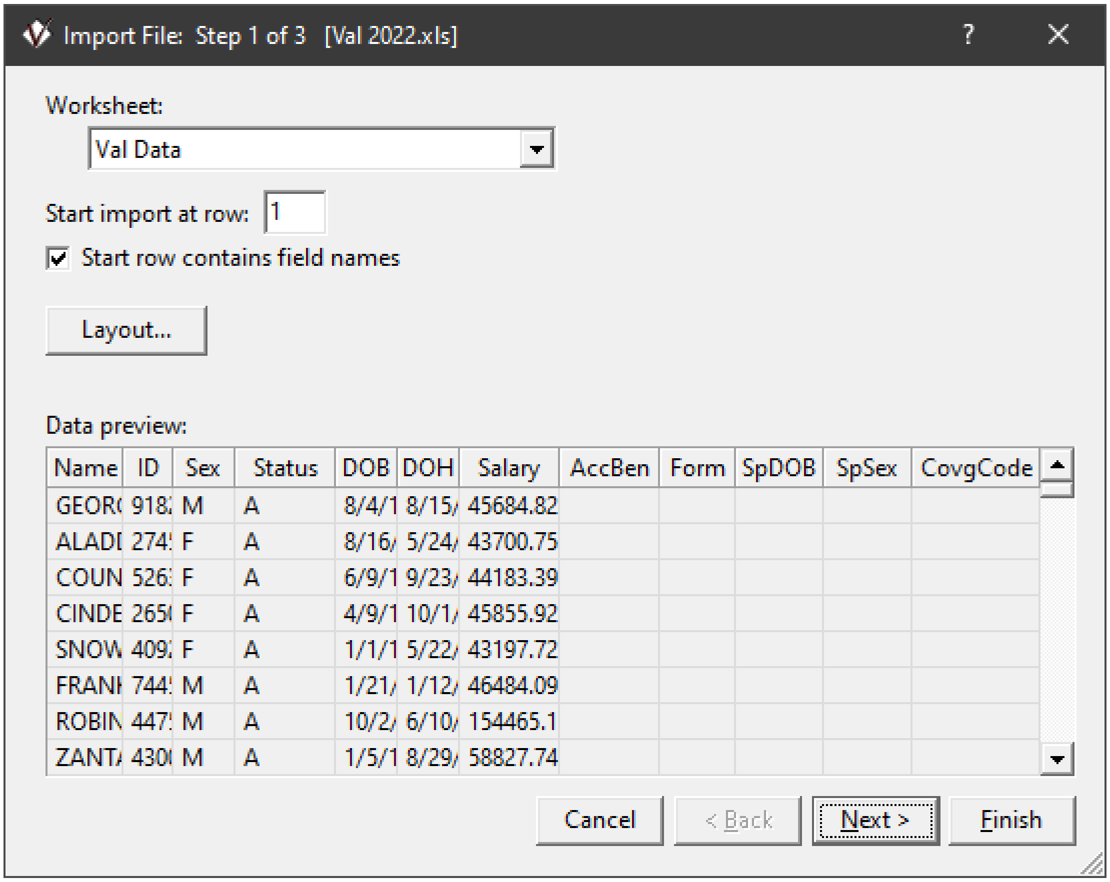
4. *Optional:* Create a Record Layout using the **Database | Import / Export Data | Record Layouts** command (or from **Spreadsheet Edit** using **File | Import / Export Data | Record Layouts**). During the import process, ProVal will automatically generate a record layout which can be modified.

   A Record Layout specifies the names and locations of fields in the file you're importing from, as well as additional information about certain types of data (date formats, code mappings, and scaling factors). This is useful if you receive data from your client in the same format every year.

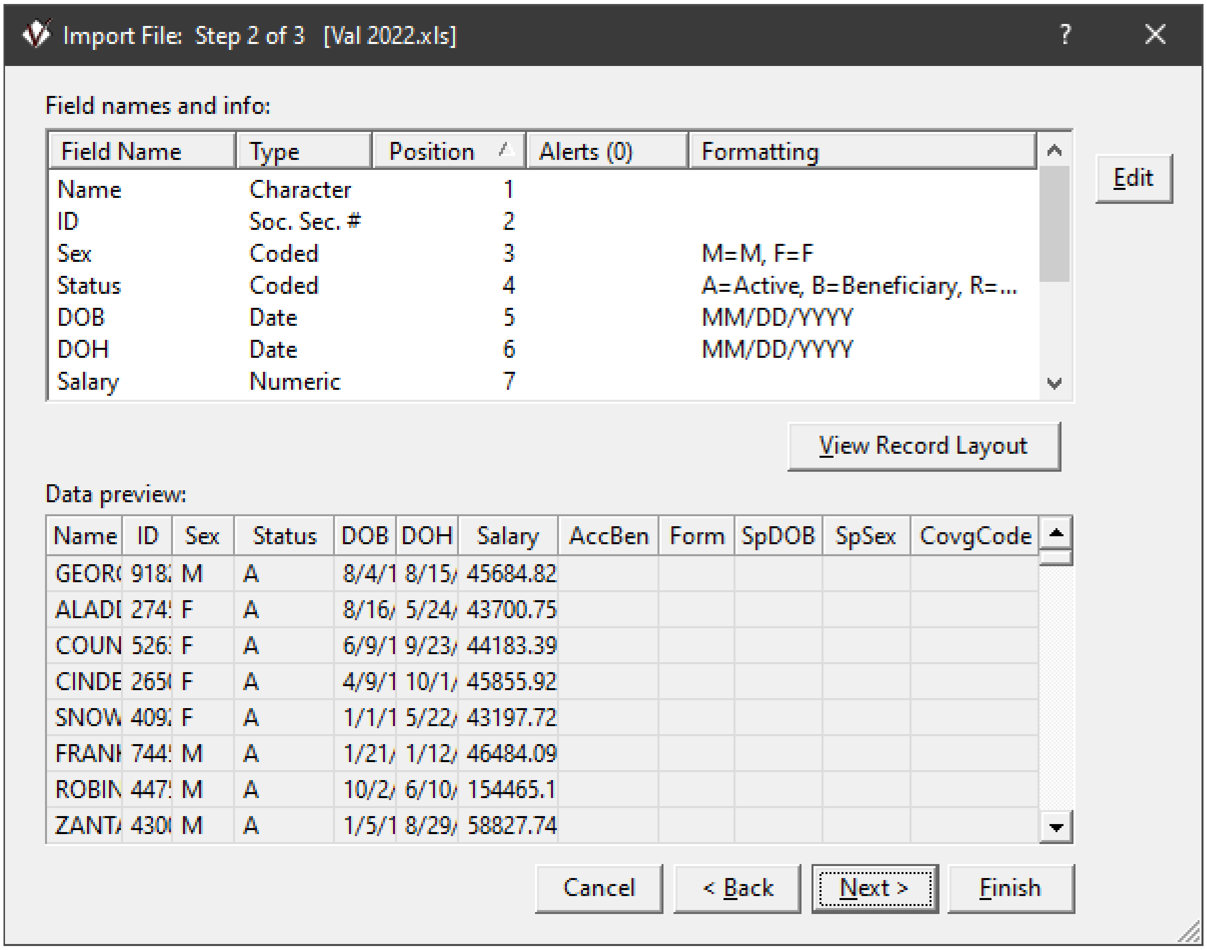
5. Import the data using the **Database | Import / Export Data | Import Data** command (or from Spreadsheet Edit using **File | Import / Export Data | Import Data**). Select the file that you wish to import and click **Open**.



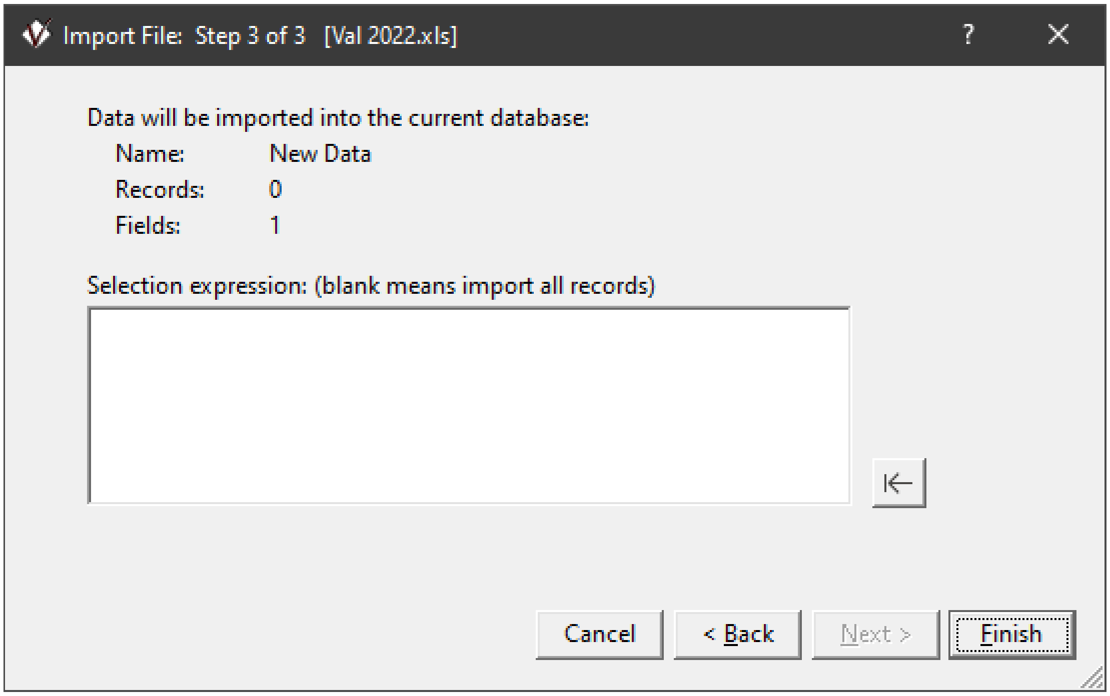
6. If you are importing from Excel, select the appropriate worksheet. Next, make sure the import is starting at the proper row. If you are importing from a delimited file or an Excel file and you have a header row, you may check the box that says **First row contains field names**. If you created a record layout, click the **Layout…** button and select it. The name of the Record Layout should appear next to the **Layout…** button. Click **Next**.



7. Double check that the **field names and info** are correct, especially for coded fields. Click on a field name in the top box to edit that field's properties. Click **Next**.



8. If you want to only import some of the records, you may type in a selection expression or choose one already saved in a selection expression library.



9. If the database already contains records, ProVal will ask whether to append imported records to the end of the database or replace the database.

10. To see the data that you imported, use the **Database | Edit Data | Spreadsheet Edit** command (not necessary when importing from **Spreadsheet Edit**).

11. Use the **Census Specifications** command to explain the structure of your data for running valuations. The **Census Specifications** library can be accessed **through Input | Census** on the menu or from the **Shortcuts** pane.

    For more information, see Census Specifications below.

CASE 2: IMPORTING DATA UPDATES

Use these steps to import data updates, i.e., just the changes from last year's population. We'll use the following sample database names throughout the steps:

Data 2022	ProVal database containing all data for last year's valuation.
Update23	ProVal database containing key field(s) (e.g., SSN), updated data for continuing participants (e.g., status, salary for continuing actives, benefits for new retirees, etc.), and complete data for new entrants.
Data 2023	ProVal database containing all data for this year's valuation.

Note: all of the following steps may also be done from **Database | Edit Data | Spreadsheet Edit** as shown below in parentheses.

1. Follow steps 1-9 above for Case 1: Importing Complete Data to import this year's updated data into the database Update23 (this is a temporary place for the data; we'll erase this database after we're done).

2. Make a copy of the database Data 2022 called Data2023 using the Copy command from the database library toolbar.

3. Open Data2023 from **Spreadsheet Edit** using the **File | Open** command, or from the **Databases** command in the **Shortcuts** pane.

4. Merge Update23 into Data 2023 using the Database | Import/Export Data | Merge Data command (or from Spreadsheet Edit using File | Import / Export Data | Merge Data).

5. If you wish, erase the database Update23 using the Erase command from the database library toolbar; this database is no longer needed.

6. Use the **Census Specifications** command to explain the structure of your data for running valuations. The **Census Specifications** library can be accessed through **Input | Census** on the menu or from the Shortcuts pane. For more information, see Census Specifications below.

CENSUS SPECIFICATIONS

The **Census Specifications** maps the data from the data dictionary you've defined into the basic data fields ProVal needs to calculate liabilities.

- **Status Code Mappings** tell ProVal how to treat each participant with respect to future service, decrements, and mortality. ProVal can treat each participant in seven different ways:

1. **Active**: Use for participants who are subject to probabilities of decrement (i.e., retirement, termination, death, and disability). This might include actives that are still working but no longer accruing benefits.

2-5. **Retired, Vested, Disabled, and Survivor**: Use for inactive participants. The only distinction among the inactive statuses is which mortality table applies from valuation assumptions.

6. **Non-participating:** Use for records which are excluded from all calculations (e.g. death, cash out, or non-vested termination).

7. **Vested valued through active*** (or **Terminated vested** in German mode) Use for inactive participants when benefits must be calculated at commencement, rather than termination (e.g. graded retirement rates or lump sums).

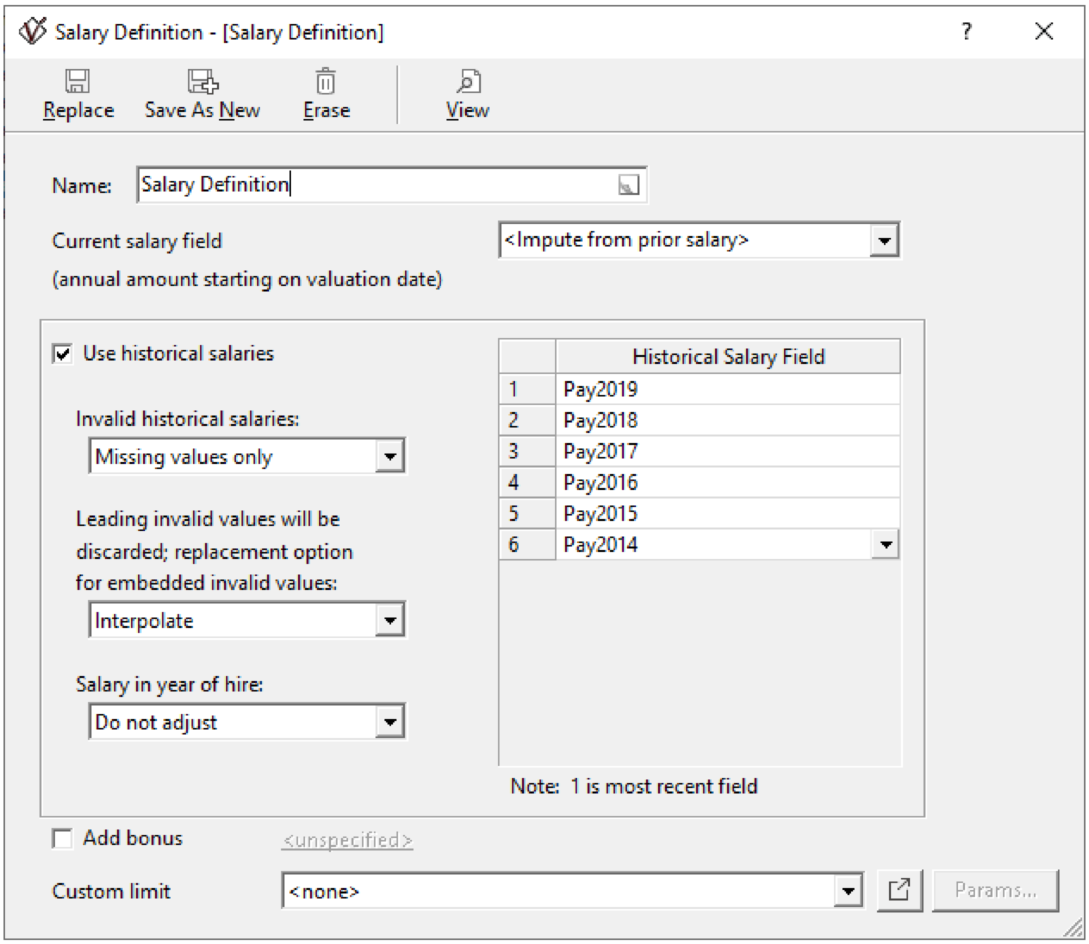
   Benefits for these participants are coded in the Plan Definition. Participants are exposed to retirement rates and vested mortality rates. Disability and termination decrements (and benefits) are ignored (except in German mode where only termination decrements are ignored).

   ** Not yet implemented for U.K. Mode*

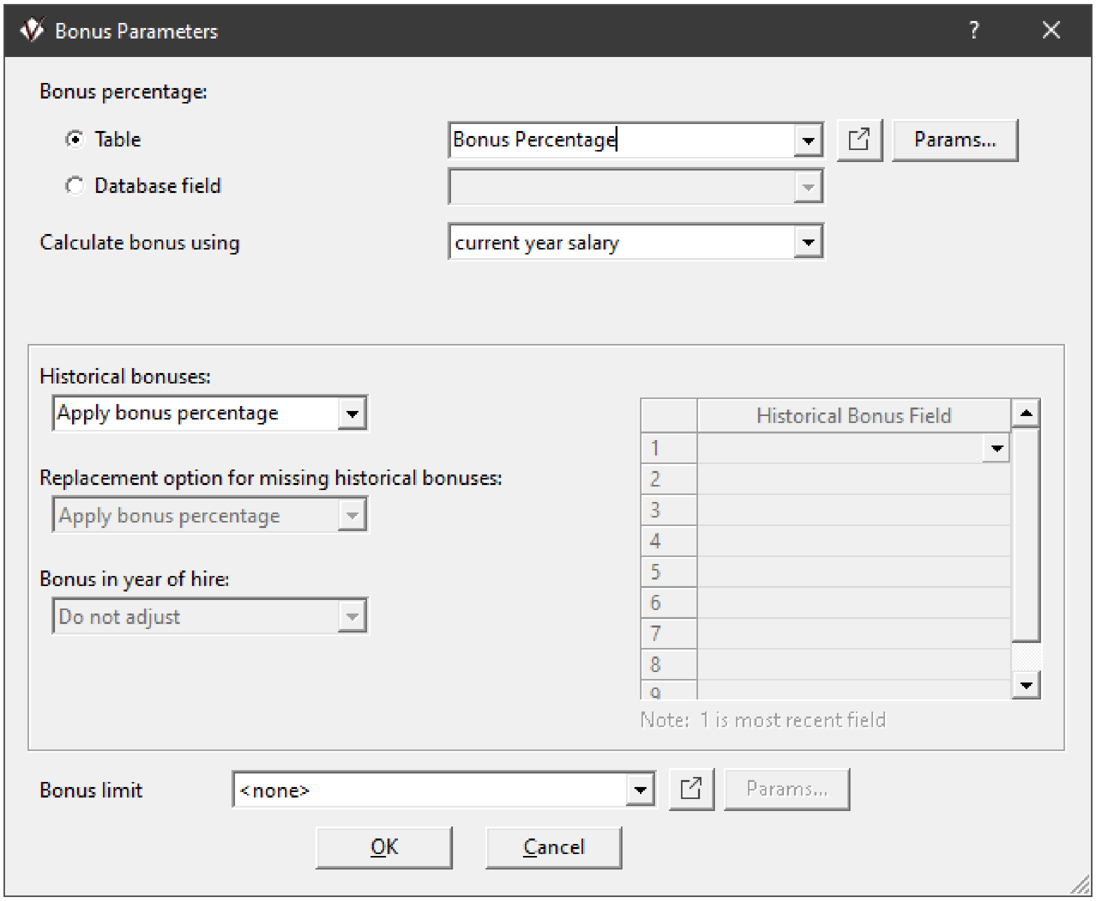
- **Active Data** determines the attained age, hire age, sex, and salaries for active participants.

  - A salary definition must be defined, as illustrated below.

    The “current salary” refers to salary *beginning* on the valuation date. If you wish to determine current salary by using the most recent historical salary increased with salary scale, choose “<Impute from prior salary>.”



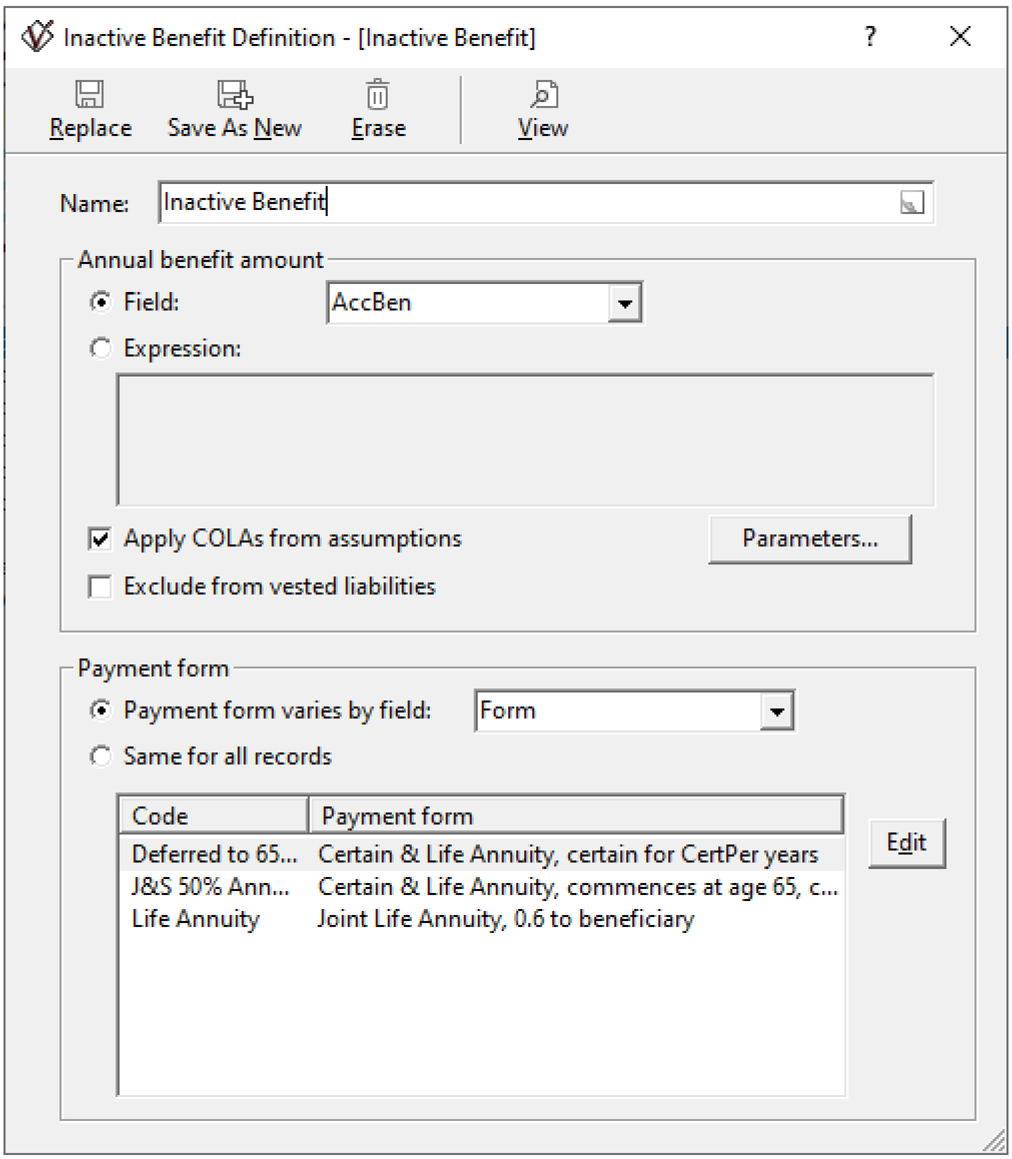
If you wish to apply a bonus percentage to your base salary, you may add a bonus definition by clicking **Add bonus**. Bonuses may be based on data or a percent of pay table.

- 

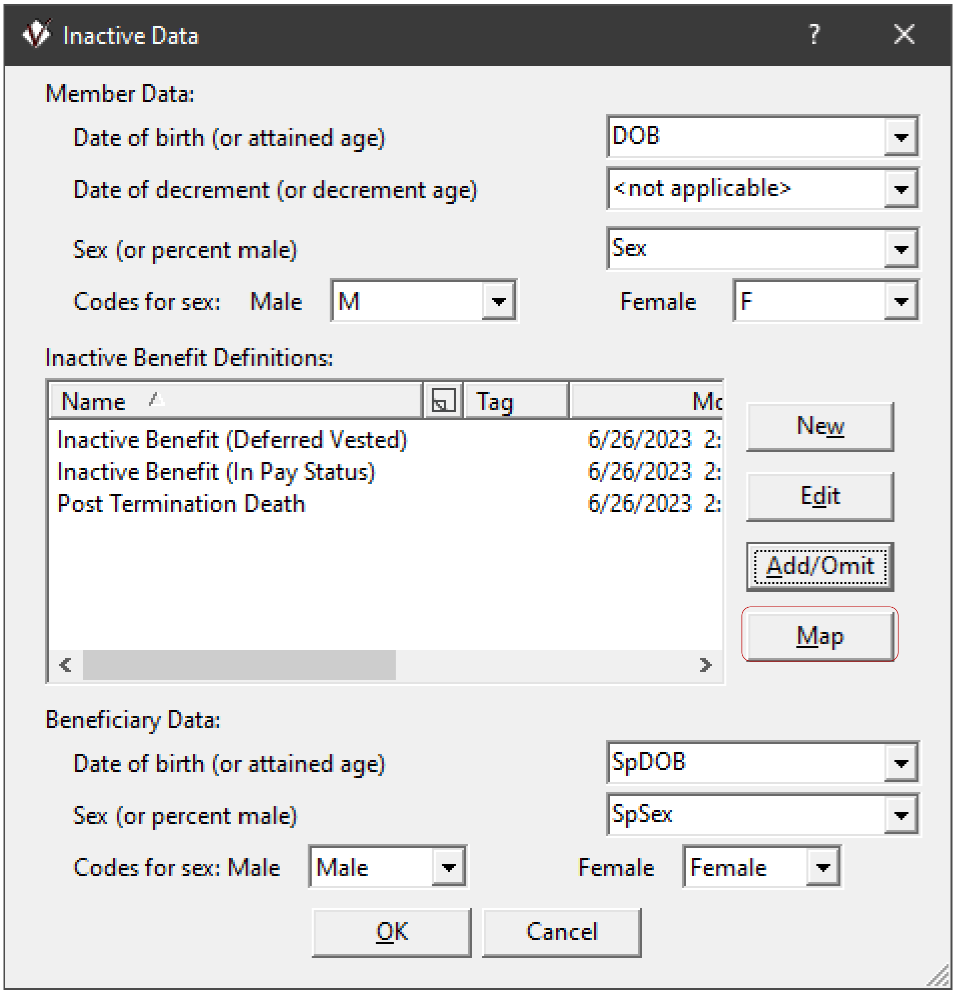
Inactive Data determines the attained age, sex, and benefits (in pension modes) for inactive participants and their spouses. In OPEB mode, it also determines which (or percentage of which) spouses are covered.

Inactive benefits (in pension modes), as illustrated below, define the annual benefit amount and payment form to be paid to each inactive participant. This includes whether the payment is deferred, in payment status, certain for a number of years, temporary for a number of years, etc. Once you select your payment form field, you will need to click on each individual payment form to define the payment form. You cannot leave any form as <unspecified>.

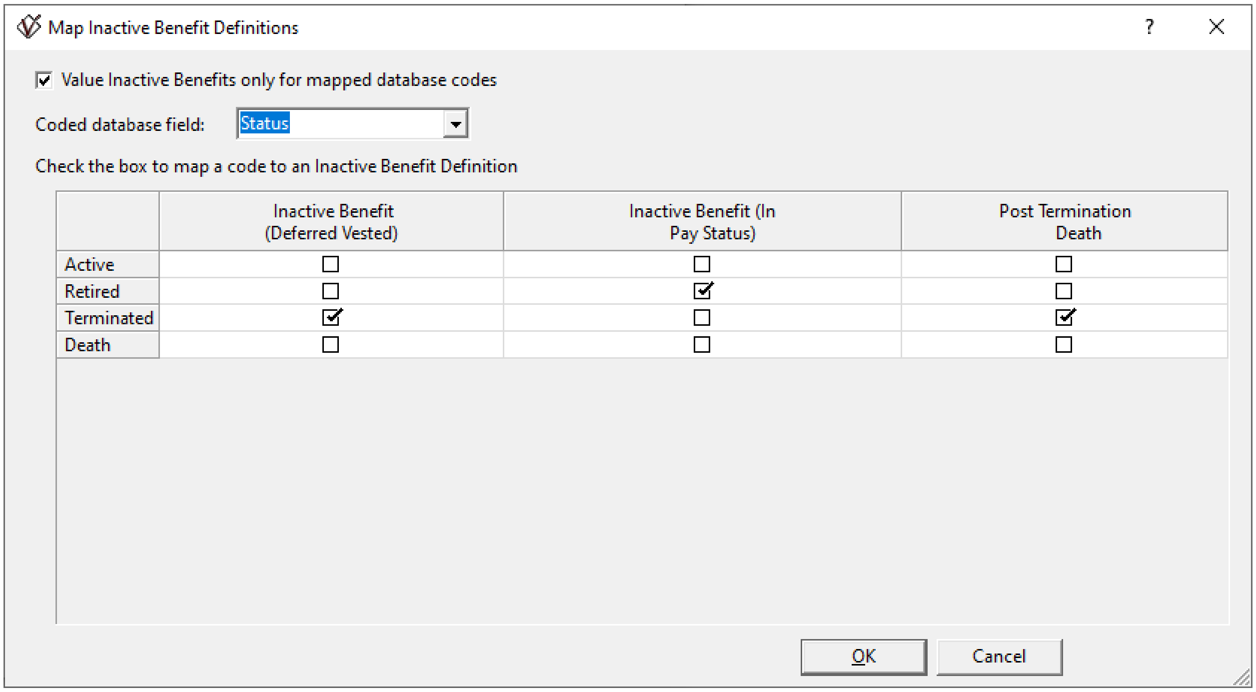
In the U.K pension mode, several service tranches can be specified within a single inactive benefit. For more information, see U.K. – Multiple Tranches below.



- If you have multiple inactive benefits for which different groups of participants are eligible, you may map inactive benefits using a coded field. Click the Map button on the Inactive Data Dialog Box.



Select the coded database field for which the mapping should apply and then map benefits as necessary for each code.



- **Data Defaults** define default values used for missing data (and optionally zero values) at the time a valuation or projection is executed. Note that these default values merely fill in missing field values or create new fields; therefore, existing database field values are not replaced. Data defaults are optional.

STEP 3: PLAN BENEFITS

The **Plan Definition** is where all of the active benefits are defined**.** Benefits within the plan definition are defined using a building block approach.

A Plan Definition is a collection of **Benefit Definitions**. Each Benefit Definition is paid upon a defined contingency (e.g. Retirement, Termination) and contains information about payment form, eligibility, plan formula, and other topics.

The benefit formula within a Benefit Definition is written using **Benefit Formula Components**. Each Benefit Formula Component defines one piece of the total formula (e.g. Final Average Pay formula, Early Retirement Factors, Minimum Benefit).

Note that when working in German Pension mode there are also benefit promises which are discussed in a separate section below.

Here's a sample plan that illustrates how these building blocks are put together:

Plan Definition

Benefit Definition

Contingency .... Retirement
Payment Form... Life annuity commencing immediately
Eligibility Conditions................................ Age 55 and 10 years of service
Benefit Formula (using **Components**)........ ERF * (BASE + EXCESS)

Benefit Definition

Contingency .... Termination
Payment Form... Life annuity deferred to 65
Eligibility Conditions................................ 5 years of service
Benefit Formula (using **Components**)........ BASE + EXCESS

Benefit Definition

Contingency .... Death
Payment Form... Lump Sum
Eligibility Conditions................................ 5 years of service
Benefit Formula (using **Components**)........ LSFACT * (BASE + EXCESS)

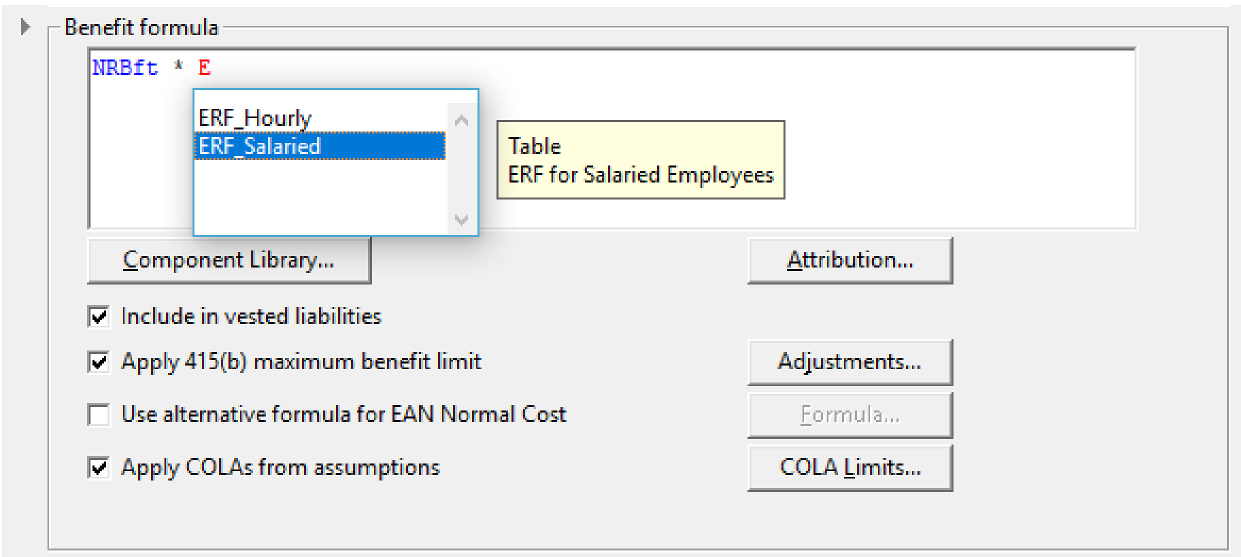
ProVal's benefit formula expression language is both flexible and concise. It accommodates pension and OPEB plans of all types: final average pay, career average pay, cash balance, pension equity, floor-offset, gross claims, net claims, etc. Below you'll find step-by-step instructions for the most common types of pension and OPEB formulas.

A few notes you might find helpful regarding expressions:

- ***ProVal starts at the left and works its way to the right, executing operators as they are encountered in the expression***. There is no precedence among operators.

- ProVal allows any of the following pairs of parenthetical symbols to be used in expressions: ( ), [ ], or { }. Judicious use of these alternative symbols can help clarify expressions containing many nested parentheses.

- When you are entering an expression, if you begin typing, predictive text will suggest possible components or operators which you may be looking for. Click once to see the description of the component and twice to select it. Click on a blue component to edit it.



- You may also press the **F1** to display an expression help window. The help screen describes the components and operators available for use in expressions. The cursor must be in an expression entry field for F1 to display this help message.

- For more information, see Appendix A: Expressions.

Benefit formula components can be one of ten different types:

1. **Accrual – Final Average** refers to a rate accrual by service, age or points (age + service). The sum of rates will be multiplied by a basis, such as dollar amount or salary.

2. **Accrual definition – Career average** refers to a rate accrual by service, age or points (age + service). The accrual in each year will be multiplied by the corresponding basis in each year.

3. **Accrual definition – Cash balance** refers to a rate of accrual by service, age or points (age + service). The accrual in each year will be multiplied by the corresponding basis in each year and the balance will be increased with interest.

4. **Accrual definition – Pension Equity** refers to a rate of accrual by service, age, or points (age + service). The sum of rates will be multiplied by a basis, such as dollar amount or salary. The benefit will then be converted from a lump sum to an annuity at normal retirement age.

5. **Accrual basis – Basis only** will return the value of the basis in each year (such as dollar amount or salary), without applying any rate accruals.

6. **Constant** is a flat constant value which may vary by coded field and/or have an increase assumption associated with it. If an increase rate isn't desired; the constant value (e.g., 10000) can be typed directly into the benefit formula.

7. **Database field** refers to the value in a field on the database, such as a grandfathered benefit amount, or an expression using a combination of fields. An increase rate may be applied to this component type.

8. **Late retirement** adjusts an existing component for late retirement. For ages after normal retirement age, it returns the greater of the component with projected accruals and an actuarial adjusted value at normal retirement age.

9. **Lump sum factor** calculates a lump sum conversion factor, or annuity factor, based on a mortality and interest basis which are specified in valuation assumptions.

10. **Table** looks up values from a table based on age and/or service, and may be sex distinct. A Table component may point to a single table or multiple tables which vary by coded database field and increase rates may be applied. Examples include: early retirement factors, claims tables, etc.

11. **Subformula** is just an expression combining the other types of components. Rather than repeating the expression in multiple benefit formulas, simply refer to the subformula component's name.

Plan Definition

The formulas on the following pages detail how to set up a Benefit Definition for the termination contingency (pension) or retirement (OPEB). You will also need to set up additional Benefit Definitions for other contingencies.

Once all benefits have been created, from the **Shortcuts** pane, click **Plan Definitions** and click **New**. Enter a **Plan name** (e.g. "ABC Company Pension Plan") and use the **Add/Omit** button to select the relevant benefits to be included in your plan.

PENSION – FINAL AVERAGE SALARY BENEFIT

The following pages will guide you through setting up a final average pay benefit formula using the general expression:

BASE + EXCESS #ZMINUS OFFSET

Before you start coding, gather the following information (each *item* in italics is referred to in the steps that follow):

Item	Example	Your plan
Salary definition	W2	
Final average years	5 out of the last 10	
% of all pay	1.5% per year of service	
Maximum service	30 years	
Integration level	Covered Compensation	
% of pay over integration level	0.67% per year of service	
Years to apply integration	30 years	
% of pay after integration years	0%	
SSPIA offset	PIA at SSNRA with salaries projected backwards at NAW	
% of SSPIA for offset	1.75% per year of service	
Max years to apply offset	30 years	

Benefit Formula Components

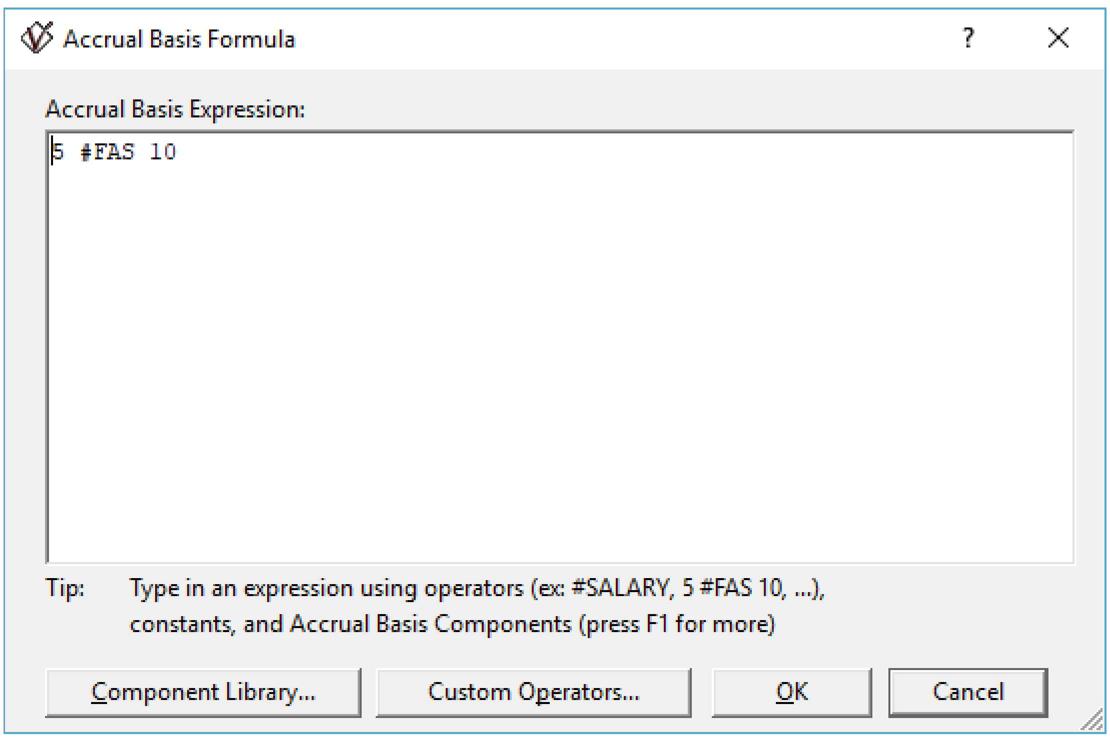
Create each necessary **Benefit Formula Component**: BASE, EXCESS, and OFFSET.

BASE

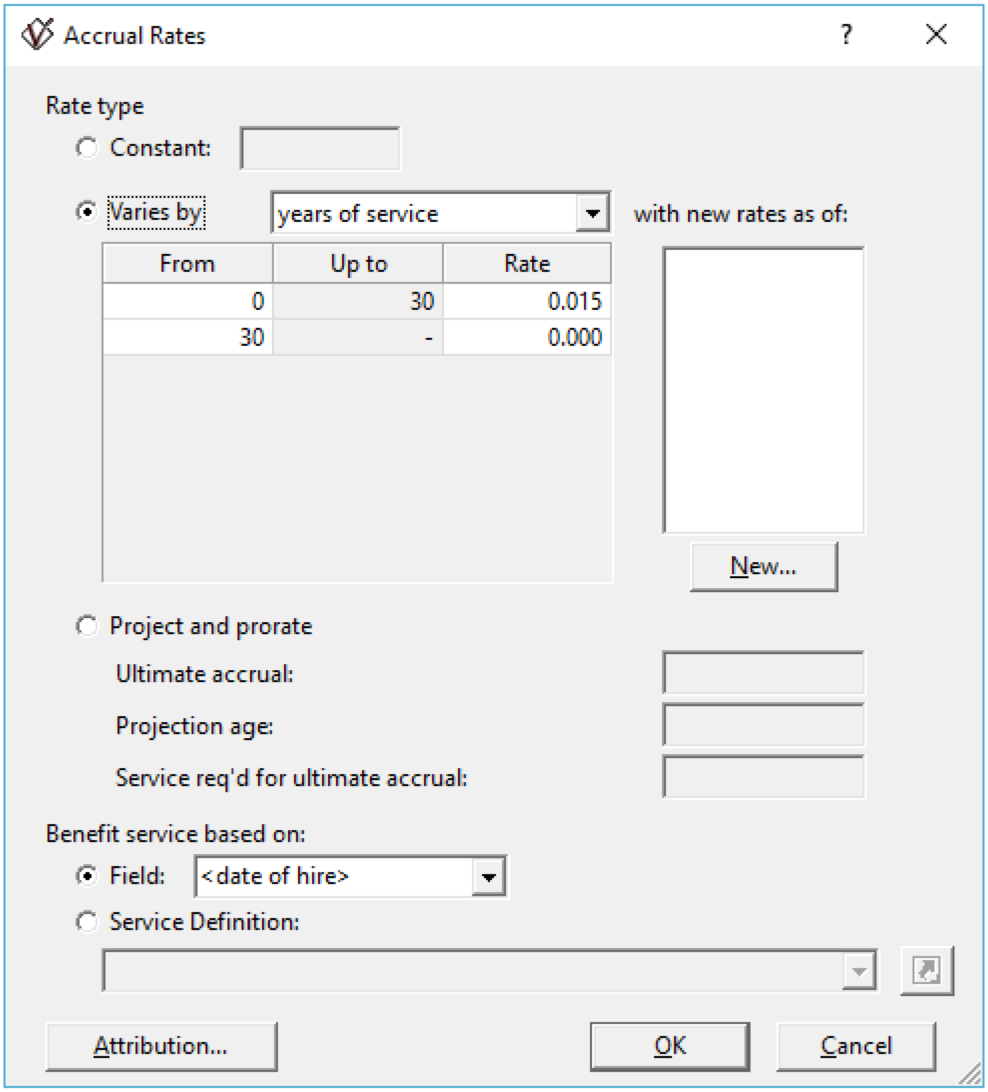
1. In the **Benefit Formula Component** library, click **New** and enter a **name** for the component, for example, BASE. Ensure that the component type is **Accrual definition** and the accrual format is **Final average.**

2. Click on **Basis Formula** and type in the *final average years* using the syntax "n #FAS m", for example "5 #FAS 10" for the highest consecutive 5-year average out of the last 10 years. Click **OK.**

   You will need to use a **Custom Operator** in place of #FAS if your final average salary includes nonconsecutive pays or if your *salary definition* differs from your valuation salary definition (set in Census Specifications).



3. Click on **Accrual Rates** and enter the *% of pay* (as a number between 0 and 1). The rate will change to 0 if your plan has a *maximum service*. The rate will also change if your plan has a *% of pay after integration years* once service has exceeded *years to apply integration*. Click **OK.**



4. Click **Save As New.**

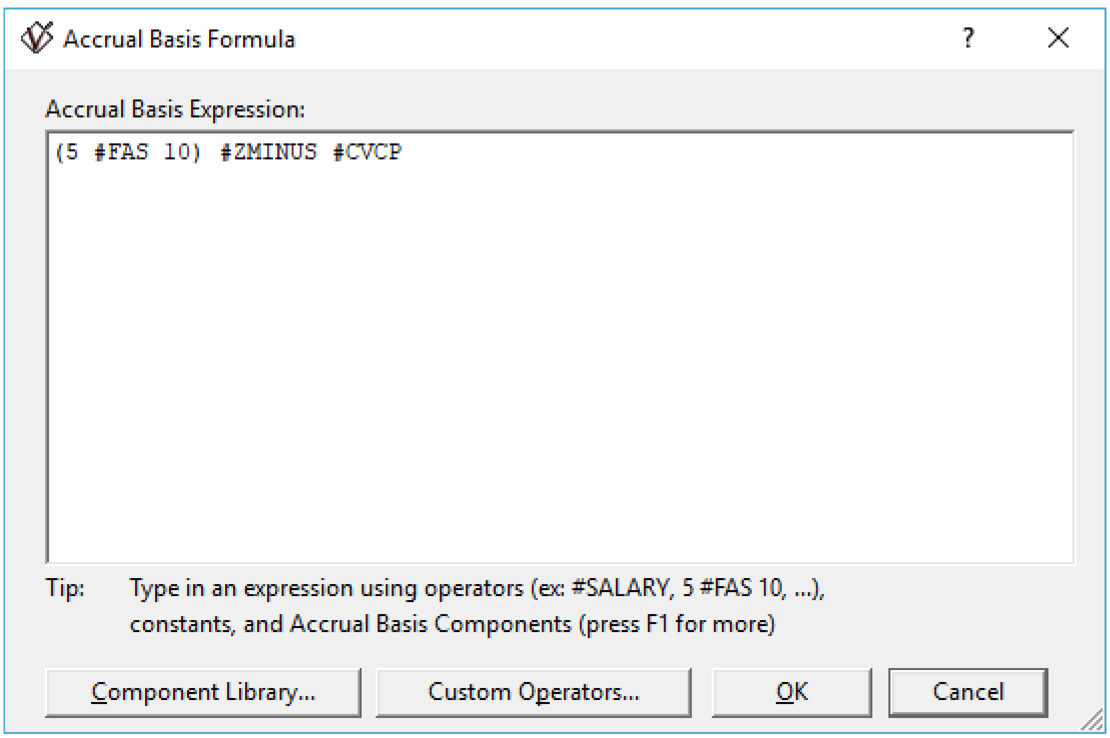
EXCESS

Skip this component if your plan doesn't have an *integration level*.

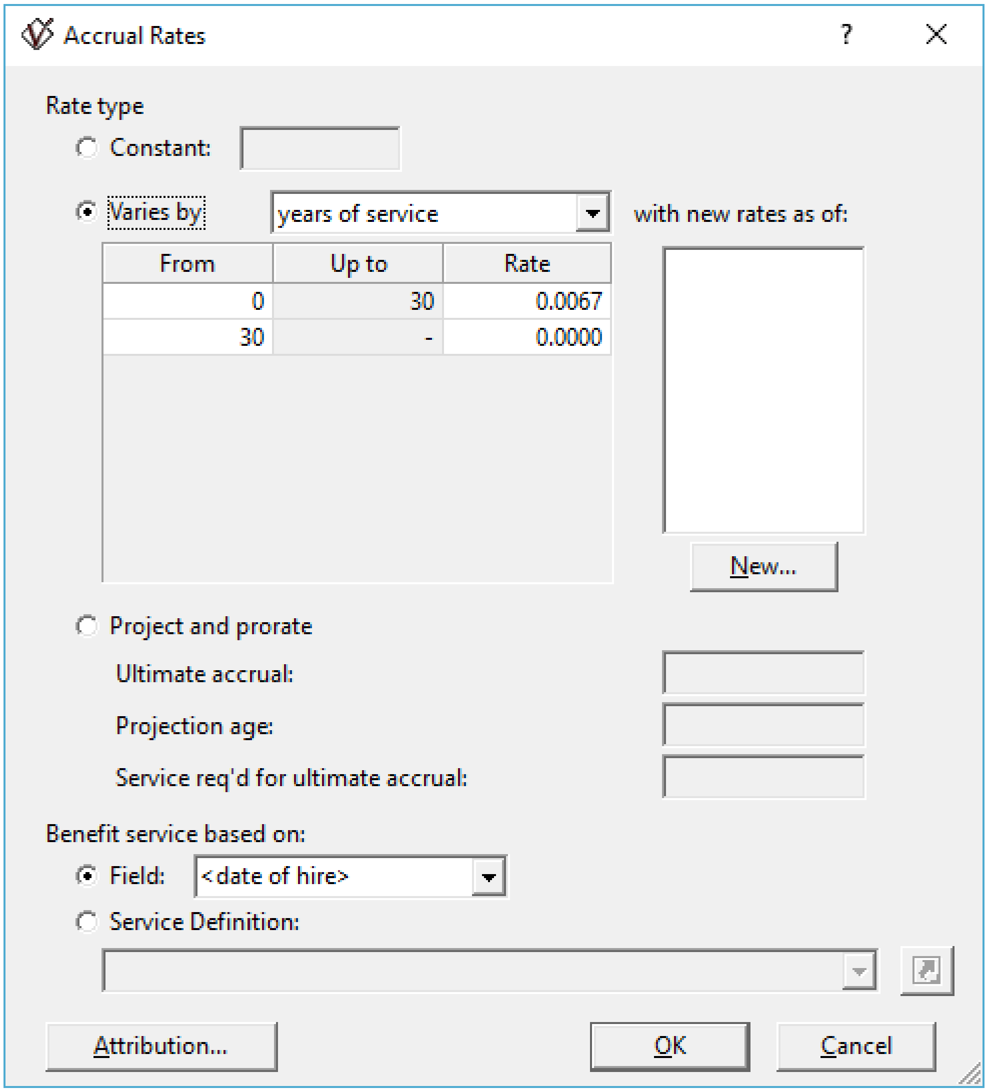
1. In the **Benefit Formula Component** library, click **New** and enter a **name** for the component, for example, EXCESS. Ensure that the component type is **Accrual definition** and the accrual format is **Final average**.

2. Click on **Basis Formula** and type in the *final average years* the same way you did for BASE. Then subtract the *integration level*, for example "#CVCP" for covered compensation. Click **OK**.

   You will need to use a Custom Operator in place of #CVCP if your plan's definition of covered compensation uses a different averaging period or wage base increases.



3. Click on **Accrual Rates** and enter the *% of pay over integration level*. The rate will change to 0 if your plan has a *years to apply integration* or *maximum service*. Click **OK**.



4. Click **Save As New.**

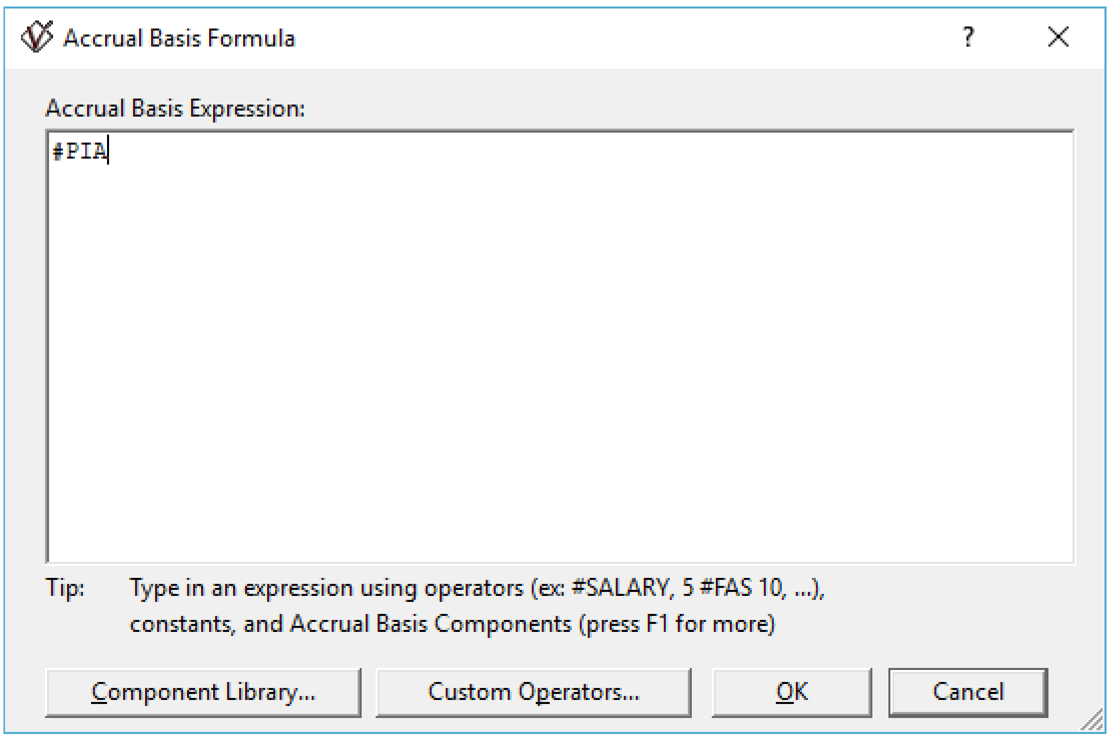
OFFSET

Skip this component if your plan doesn't have a *SSPIA offset*.

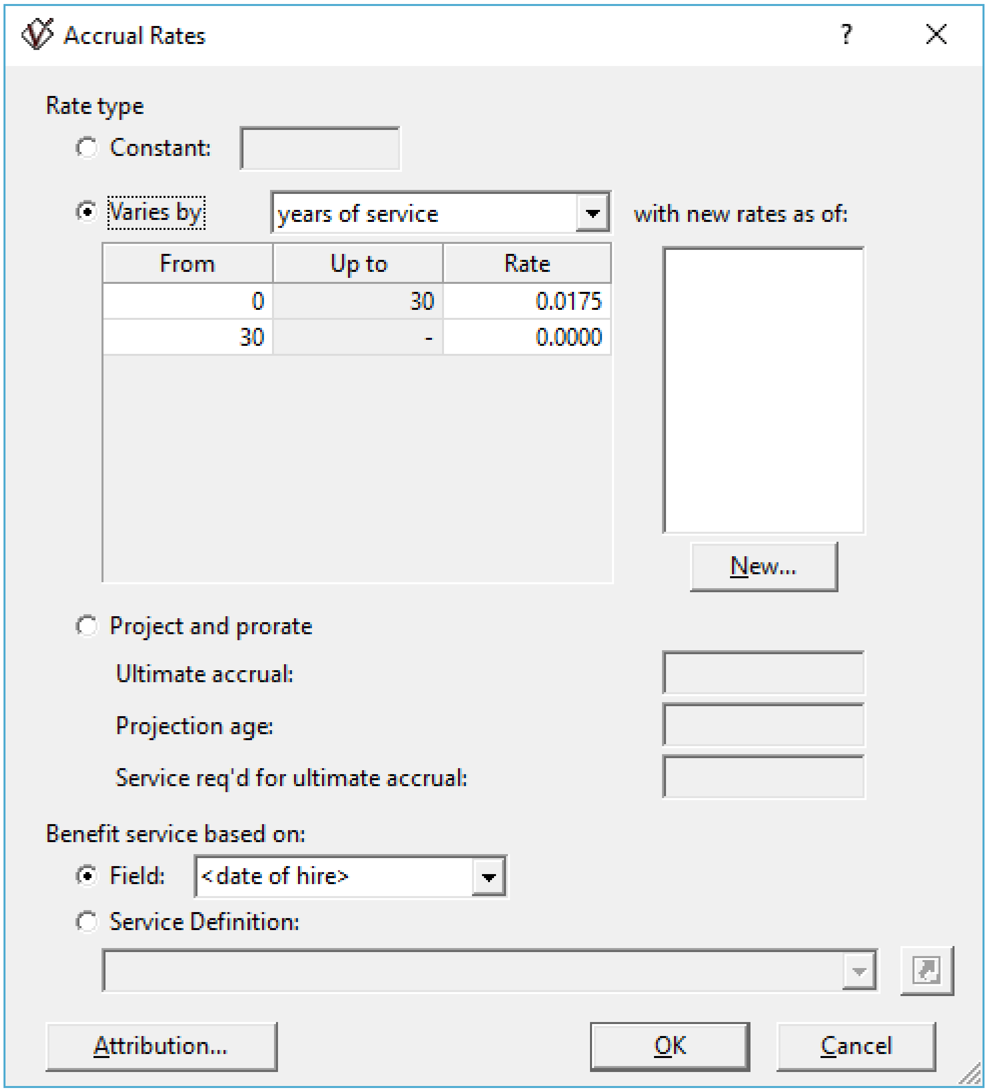
1. In the **Benefit Formula Component** library, click **New** and enter a **name** for the component, for example, OFFSET. Ensure that the component type is **Accrual definition** and the accrual format is **Final average**.

2. Click on **Basis Formula** and type in the *SSPIA offset*, for example "#PIA". Click **OK**.

   You will need to use a Custom Operator in place of #PIA if your plan's definition of PIA uses different salaries, computational ages, or law year.



3. Click on **Accrual Rates** and enter the *% of SSPIA for offset*. The rate will change to 0 if your plan has a *max years to apply offset*. Click **OK**.



4. Click **Save As New**.

Termination Benefit

1. From the **Shortcuts** pane, expand **Plan Definitions**, and then click **Benefit Definitions**.

2. Click **New** and enter a **name** for the benefit definition, for example, “Termination Benefit.” Choose Termination as the **Contingency initiating benefits**.

3. If you have **Election Probabilities**, check the box. These probabilities will be specified in Step 4: Valuation Assumptions.

4. Create **Payment forms** from within the **Benefit Definition** by clicking the “back door button”

    a. Create and select the **Normal form** of payment (e.g. a life annuity deferred to 65).

    b. Create any **Optional forms** you want to value, and check their boxes within the Optional forms topic. Conversion factors and probabilities of receipt will be specified in Step 4: Valuation Assumptions.

5. Enter **Eligibility criteria**, for example, 5 years of service.

6. Type in the **Benefit formula** using BASE, EXCESS, and OFFSET. If your plan has an *integration level* and a *SSPIA offset* your formula will be:

   BASE + EXCESS #ZMINUS OFFSET

   If your plan has an *integration level* but no *SSPIA offset* your formula will be:

   BASE + EXCESS

   Finally, if your plan has neither an *integration level* or *SSPIA offset* your formula will be:

   BASE

7. Click **Save as new**.

Benefits for the Retirement, Death, and Disability contingencies may be created with steps similar to those used for the Termination Benefit. Note that additional **Benefit Formula Components** may be required (e.g. an Early Retirement Factor table).

PENSION – CAREER AVERAGE SALARY BENEFIT

The following pages will guide you through setting up a career average pay benefit formula using the general expression:

BASE + EXCESS #ZMINUS OFFSET

Before you start coding, gather the following information (each *item* in italics is referred to in the steps that follow):

Item	Example	Your plan
Salary definition	W2	
% of all pay	1.5% per year of service	
Maximum service	30 years	
Integration level	Social Security Wage Base	
% of pay over integration level	0.67% per year of service	
Years to apply integration	30 years	
% of pay after integration years	0%	
SSPIA offset	PIA at SSNRA with salaries projected backwards at NAW	
% of SSPIA for offset	1.75% per year of service	
Max years to apply offset	30 years	

Benefit Formula Components

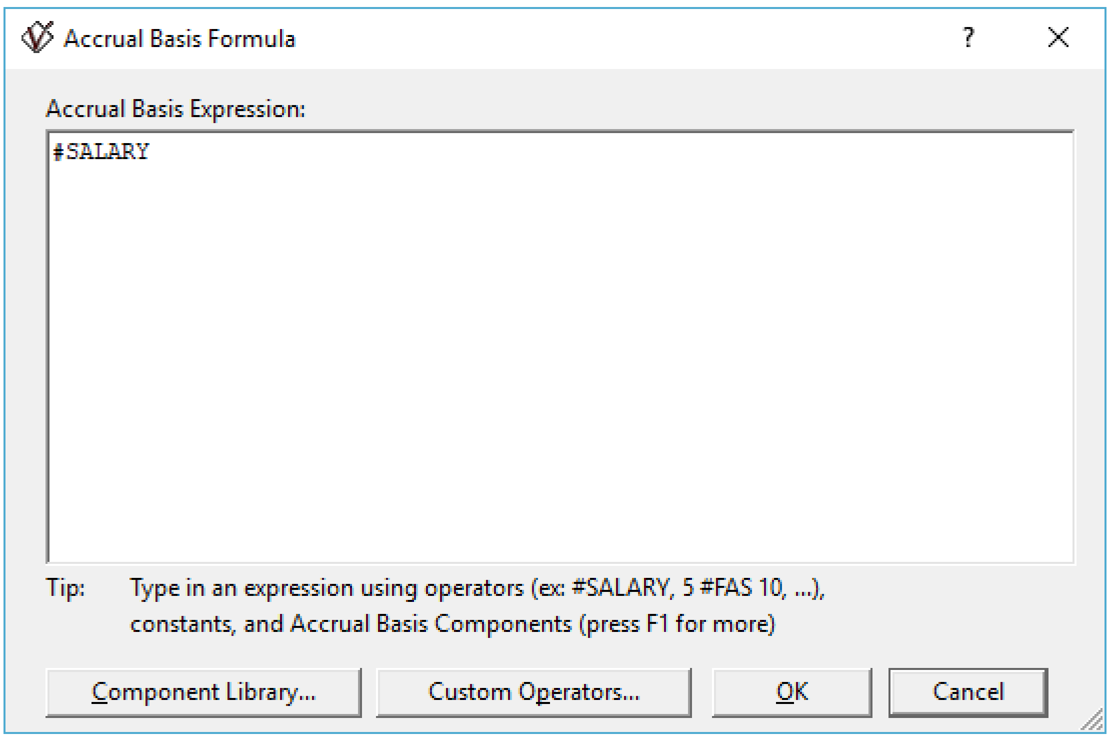
Create each necessary **Benefit Formula Component**: BASE, EXCESS, and OFFSET.

BASE

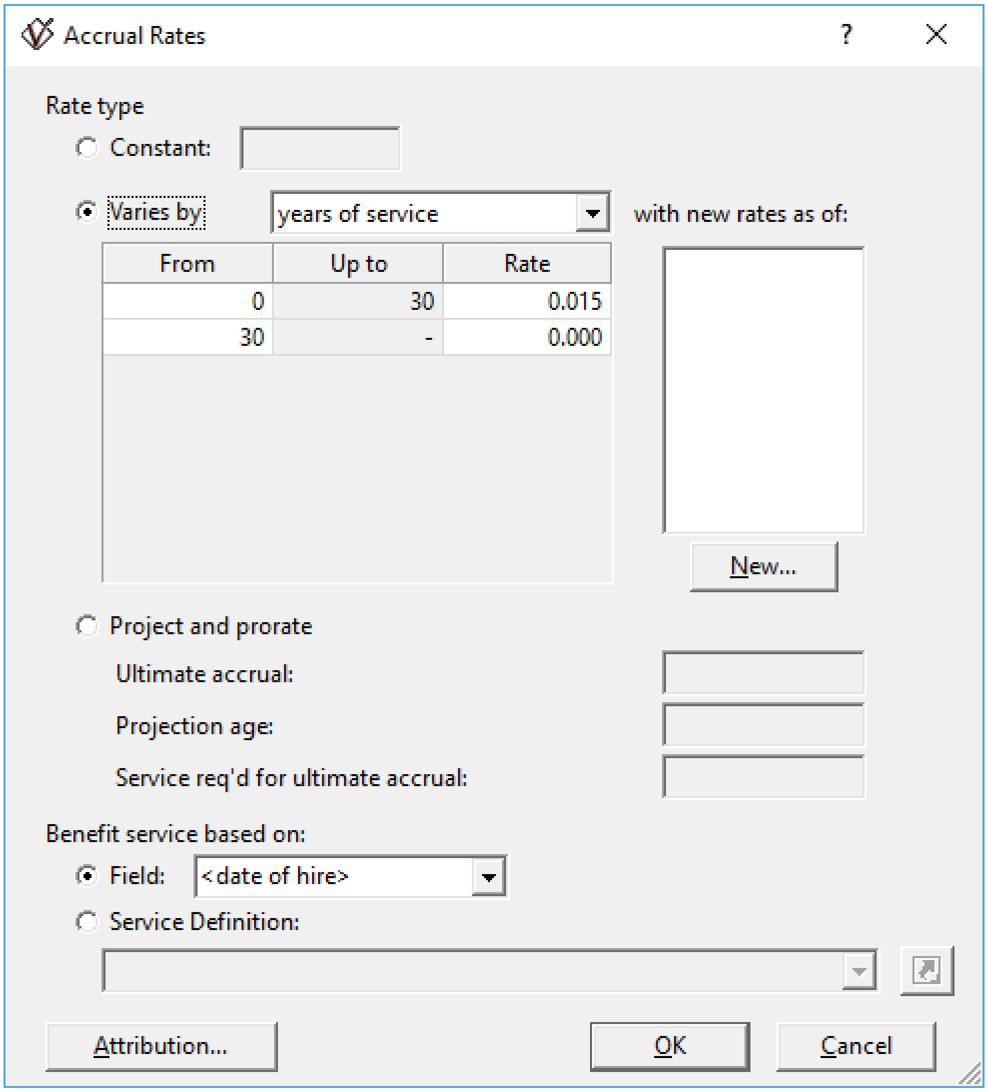
1. In the **Benefit Formula Component** library, click **New** and enter a **name** for the component, for example, BASE. Ensure that the component type is **Accrual definition** and the accrual format is **Career average.**

2. Click on **Basis Formula** and type in the *salary definition* using the operator #SALARY. Click **OK.**

   You will need to use a **Custom Operator** in place of #SALARY if your *salary definition* differs from your valuation salary definition (set in Census Specifications).



3. Click on **Accrual Rates** and enter the *% of pay* (as a number between 0 and 1). The rate will change to 0 if your plan has a *maximum service*. The rate will also change if your plan has a *% of pay after integration years* once service has exceeded *years to apply integration*. Click **OK.**



4. Click on **Accrued Benefit** and choose the **database field** containing the accrued benefit. If your plan does not store the accrued benefit, choose **expected value** to have ProVal estimate it for you. In German mode you will also have to specify crediting parameters.

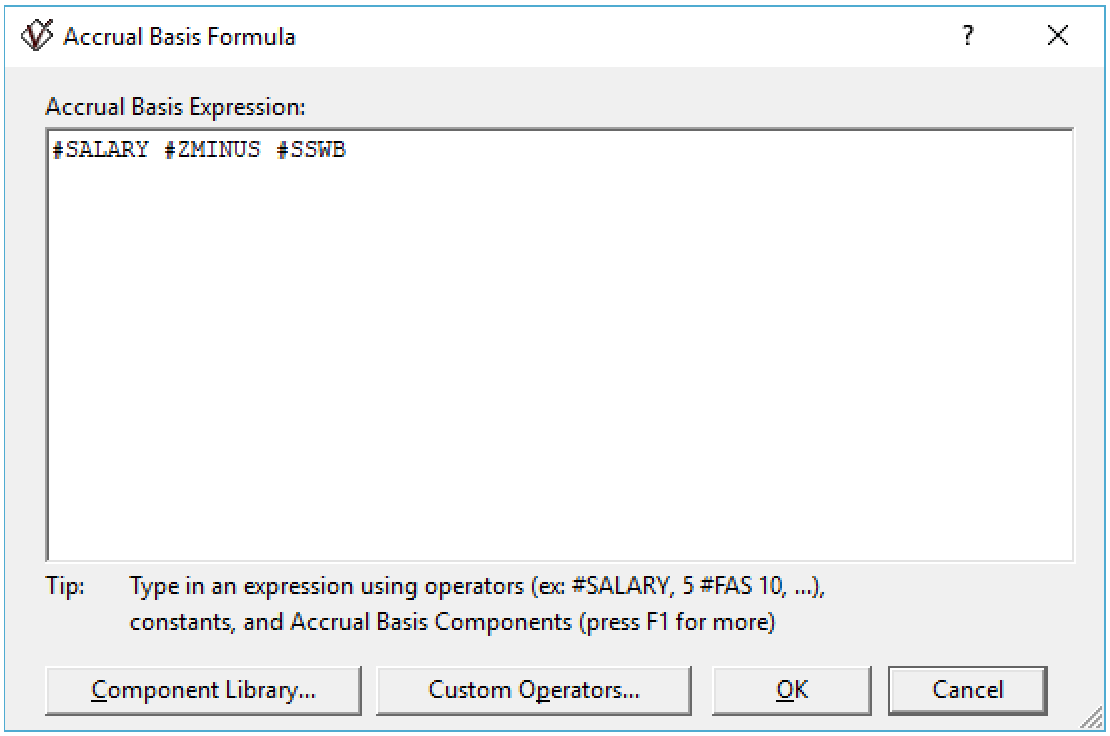
5. Click **Save As New.**

EXCESS

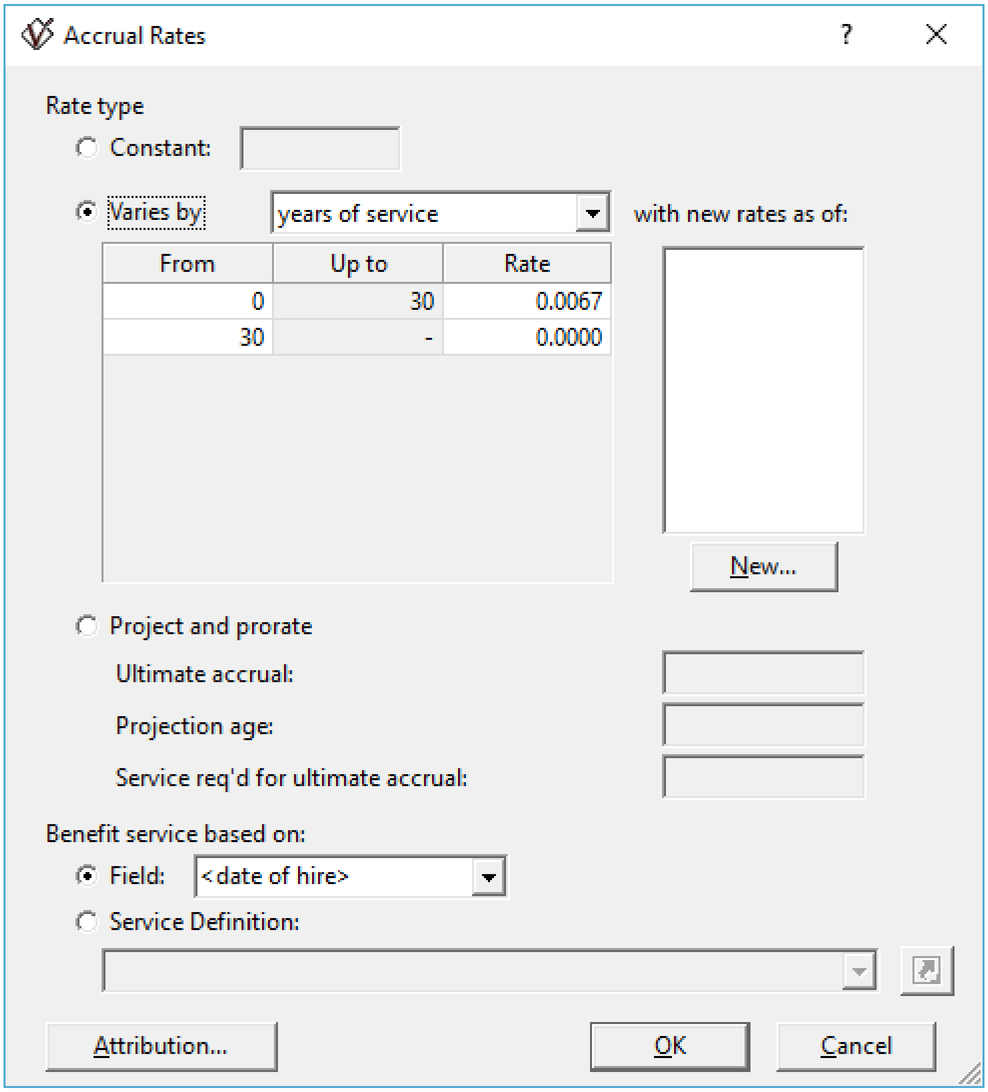
Skip this component if your plan doesn't have an *integration level*.

1. In the **Benefit Formula Component** library, click **New** and enter a **name** for the component, for example, EXCESS. Ensure that the component type is **Accrual definition** and the accrual format is **Career average**.

2. Click on **Basis Formula** and type in the *salary definition* the same way you did for BASE. Then subtract the *integration level*, for example "#SSWB" for Social Security wage base. Click **OK**.



3. Click on **Accrual Rates** and enter the *% of pay over integration level*. The rate will change to 0 if your plan has a *years to apply integration* or *maximum service*. Click **OK**.



4. Click **Save As New.**

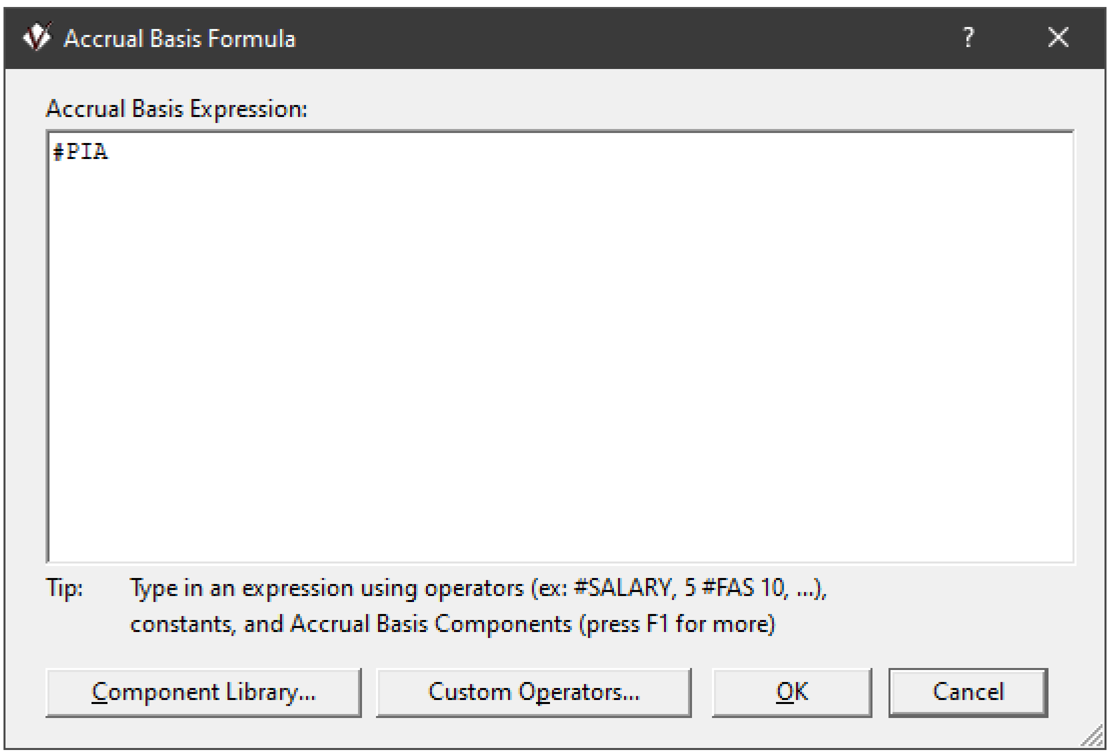
OFFSET

Skip this component if your plan doesn't have a *SSPIA offset*.

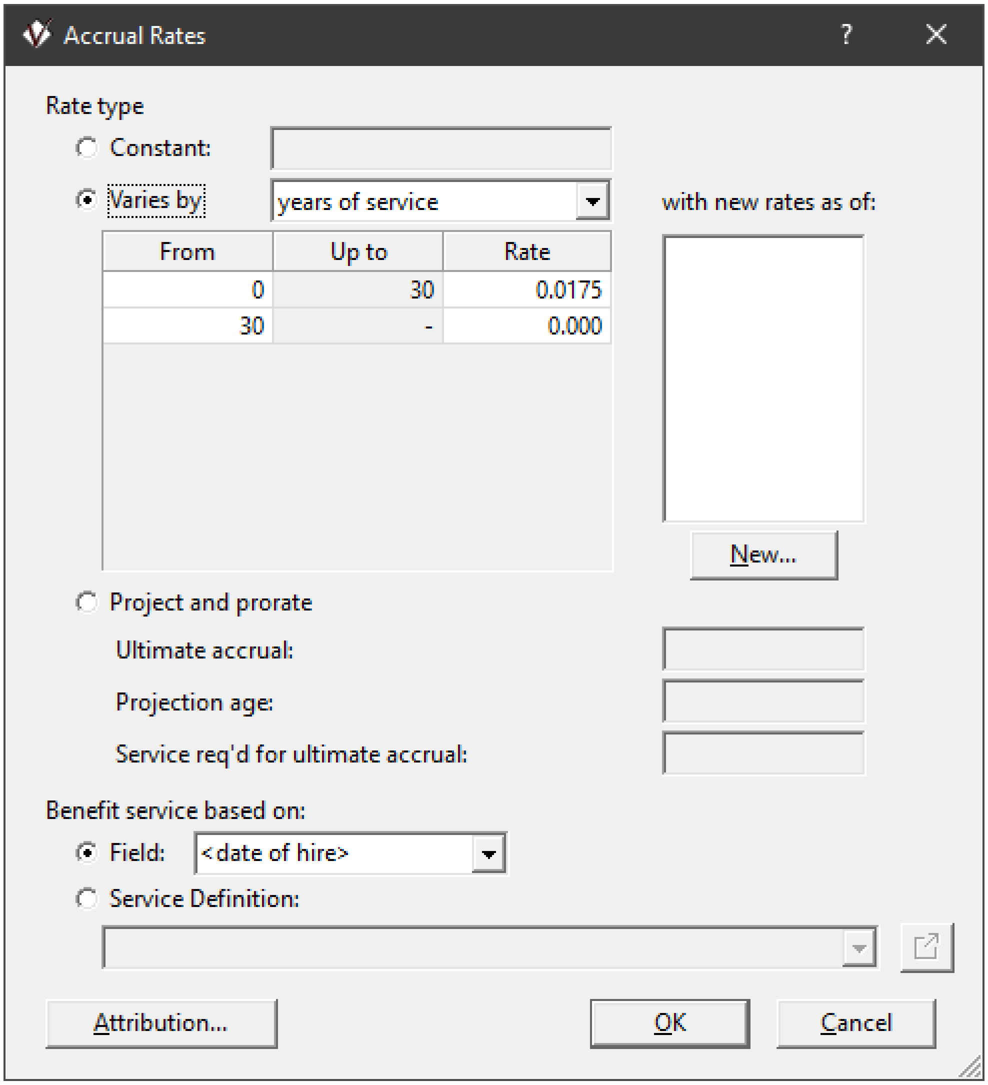
1. In the **Benefit Formula Component** library, click **New** and enter a **name** for the component, for example, OFFSET. Ensure that the component type is **Accrual definition** and the accrual format is **Final average**.

2. Click on **Basis Formula** and type in the *SSPIA offset*, for example "#PIA". Click **OK**.

   You will need to use a Custom Operator in place of #PIA if your plan's definition of PIA uses different salaries, computational ages, or law year.



3. Click on **Accrual Rates** and enter the *% of SSPIA for offset*. The rate will change to 0 if your plan has a *max years to apply offset*. Click **OK**.



4. Click **Save As New**.

Termination Benefit

1. From the **Shortcuts** pane, expand **Plan Definitions**, and then click **Benefit Definitions**.

2. Click **New** and enter a **name** for the benefit definition, for example, "Termination Benefit." Choose Termination as the **Contingency initiating benefits**.

3. If you have **Election Probabilities**, check the box. These probabilities will be specified in Step 4: Valuation Assumptions.

4. Create **Payment forms** from within the **Benefit Definition** by clicking the "back door button"

   a. Create and select the **Normal form** of payment (e.g. a life annuity deferred to 65).

   b. Create any **Optional forms** you want to value, and check their boxes within the Optional forms topic. Conversion factors and probabilities of receipt will be specified in Step 4: Valuation Assumptions.

5. Enter **Eligibility criteria**, for example, 5 years of service.

6. Type in the **Benefit formula** using BASE, EXCESS, and OFFSET. If your plan has an *integration level* and a *SSPIA offset* your formula will be:

   BASE + EXCESS #ZMINUS OFFSET

   If your plan has an *integration level* but no *SSPIA offset* your formula will be:

   BASE + EXCESS

   Finally, if your plan has neither an *integration level* or *SSPIA offset* your formula will be:

   BASE

7. Click **Save as new**.

Benefits for the Retirement, Death, and Disability contingencies may be created with steps similar to those used for the Termination Benefit. Note that additional **Benefit Formula Components** may be required (e.g. an Early Retirement Factor table).

PENSION – HOURLY BENEFIT

The following pages will guide you through setting up a final average pay benefit formula using the general expression:

HOURLYBFT

Before you start coding, gather the following information (each *item* in italics is referred to in the steps that follow):

Item	Example	Your plan
Dollar accruals	$40 / month (-1999) $45 / month (2000-2013) $50 / month (2014-2004) $55 / month (2015-)	
Accrued benefit	Stored on data (the alternative is to let ProVal calculate the accrued benefit)	
Maximum Service	30 years	

Benefit Formula Components

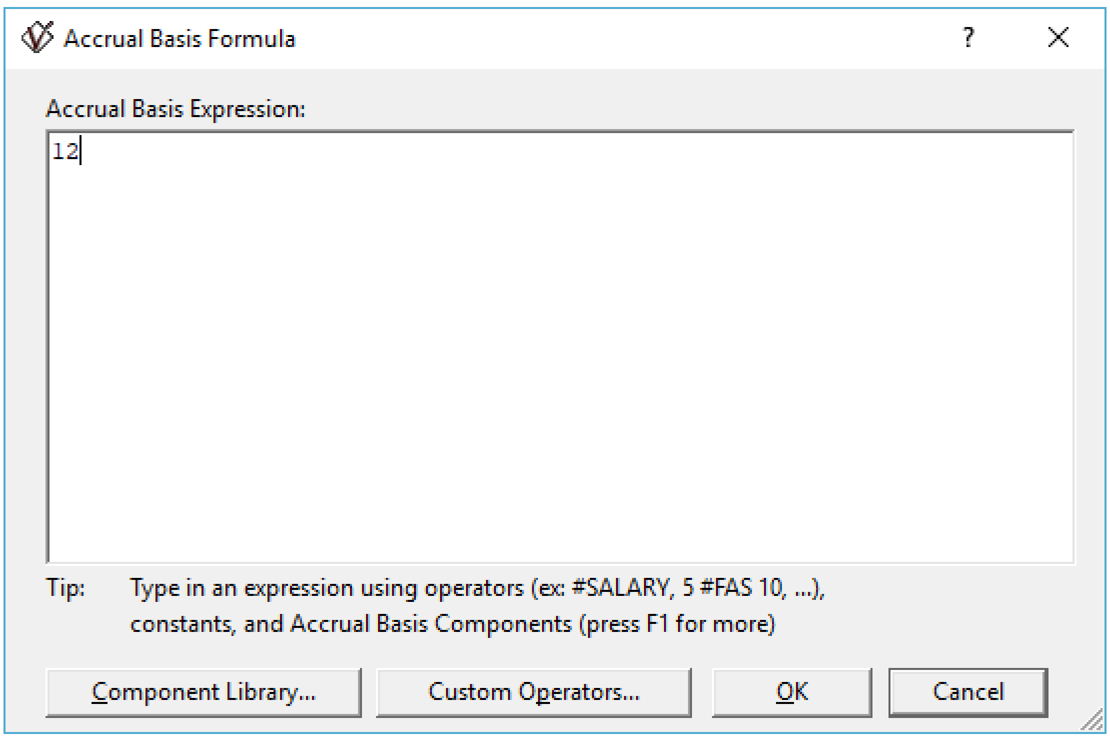
Create the necessary **Benefit Formula Component**: HOURLYBFT.

HOURLYBFT

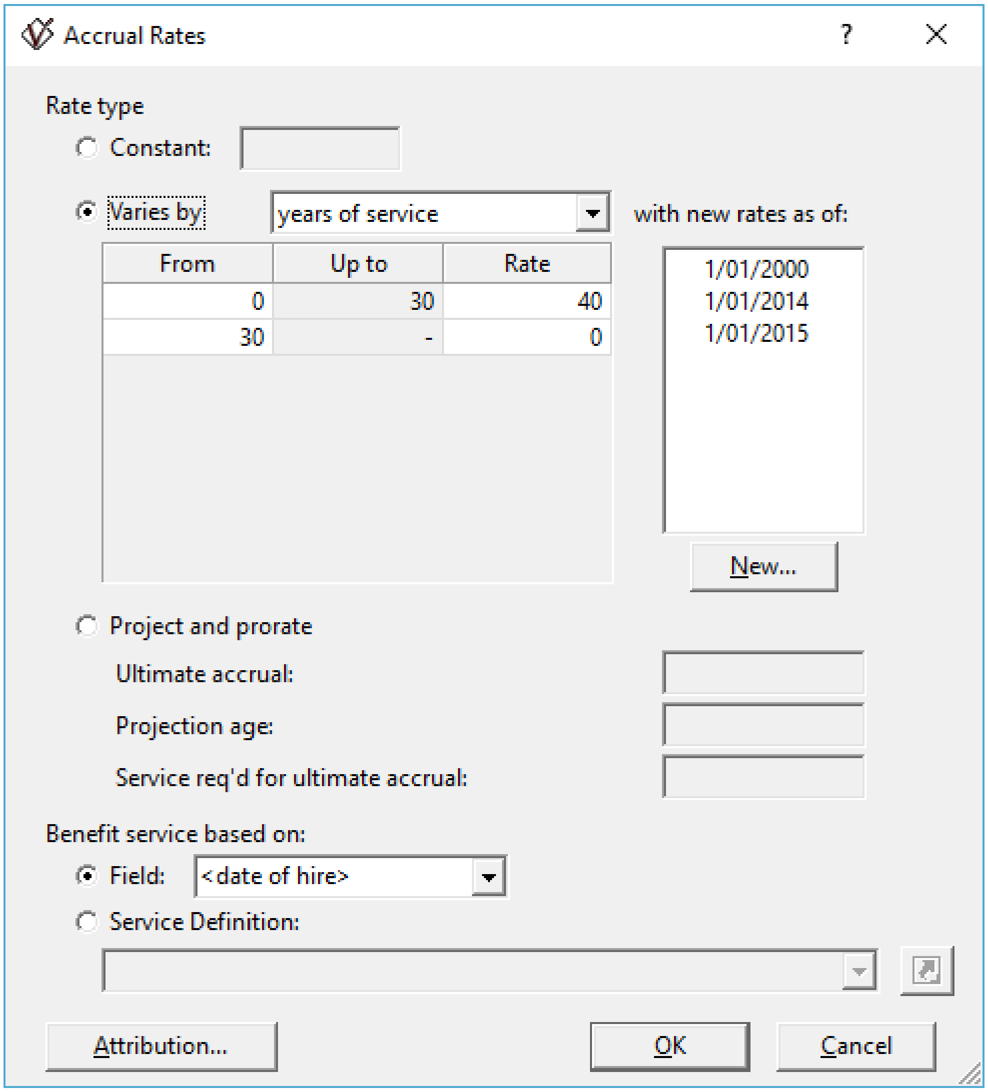
1. In the **Benefit Formula Component** library, click **New** and enter a **name** for the component, for example, HOURLYBFT. Ensure that the component type is **Accrual definition**.

   If you have the *accrued benefit* stored in the data, choose **Career average** as the accrual format. If not, choose **Final average** as the accrual format.

2. Click on **Basis Formula** and type in a factor to annualize the *dollar accruals*. This will generally be "1" if your accrual rates are annualized already or "12 if your accrual rates are monthly. Click **OK.**



3. Click on **Accrual Rates** and enter the *dollar accruals*. The rate will change to 0 if your plan has a *maximum service*. If the *dollar accruals* change over time, enter the initial amount on the rates screen. Enter subsequent amounts by clicking the **New** button and specifying whether the rates apply prospectively only (i.e., **years after the effective date)** or retroactively for **all years**. Click **OK.**



4. Click on **Accrued Benefit** and choose the **database field** containing the accrued benefit. Note: this only applies if you have an *accrued benefit* stored in the data.

5. Click **Save As New.**

TERMINATION BENEFIT

1. From the **Shortcuts** pane, expand **Plan Definitions**, and then click **Benefit Definitions**.

2. Click **New** and enter a **name** for the benefit definition, for example, “Termination Benefit.” Choose Termination as the **Contingency initiating benefits**.

3. If you have **Election Probabilities**, check the box. These probabilities will be specified in Step 4: Valuation Assumptions.

4. Create **Payment forms** from within the **Benefit Definition** by clicking the “back door button”

   a. Create and select the **Normal form** of payment, for example, a life annuity deferred to 65.

   b. Create any **Optional forms** you want to value, and check their boxes within the Optional forms topic. Conversion factors and probabilities of receipt will be specified in Step 4: Valuation Assumptions.

5. Enter **Eligibility criteria**, for example, 5 years of service.

6. Type in the **Benefit formula**:

   HOURLYBFT

7. Click **Save as new**.

Benefits for the Retirement, Death, and Disability contingencies may be created with steps similar to those used for the Termination Benefit. Note that additional **Benefit Formula Components** may be required (e.g. an Early Retirement Factor table).

PENSION – CASH BALANCE BENEFIT

The following pages will guide you through setting up a cash balance benefit formula using the general expression:

CASHBAL / ANNCONV

Before you start coding, gather the following information (each *item* in italics is referred to in the steps that follow):

Item	Example	Your plan
Salary definition	W2	
% of pay	1.5% per year of service up to 10 years, 2% for service thereafter	
Interest crediting rate	6% per year	
Maximum Service	30 years	

Benefit Formula Components

Create the necessary **Benefit Formula Component**: CASHBAL and ANNCONV.

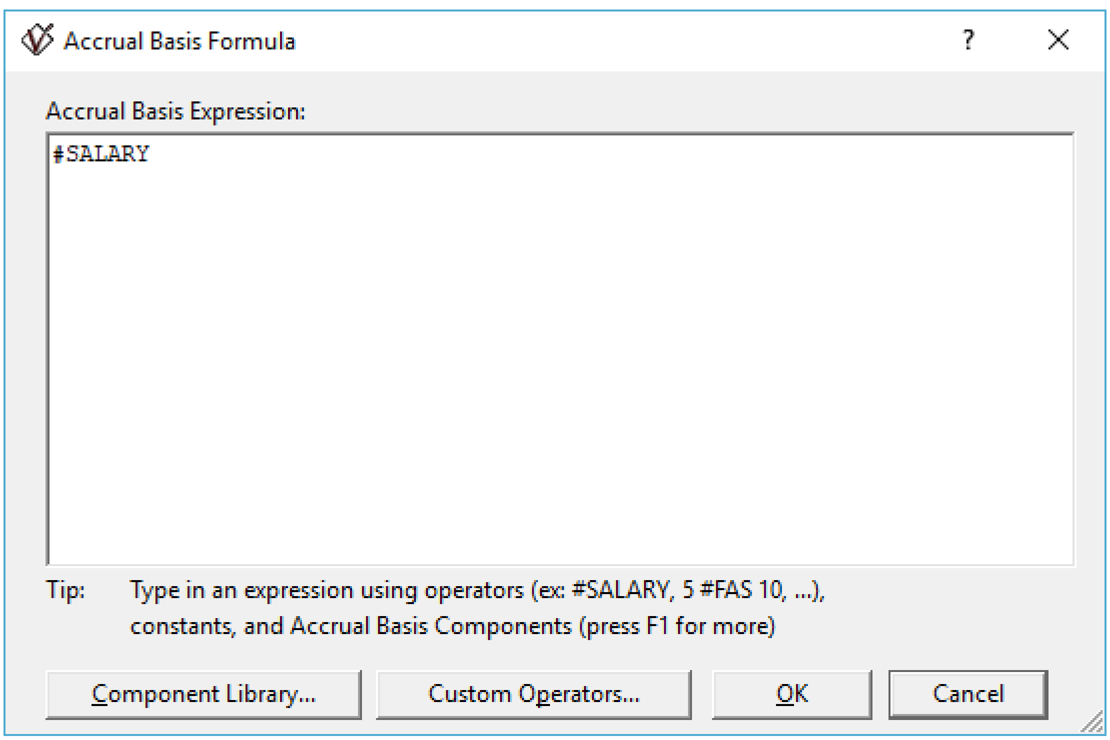
CASHBAL

1. In the **Benefit Formula Component** library, click **New** and enter a **name** for the component, for example, CASHBAL. Ensure that the component type is **Accrual definition** and the accrual format is **Cash Balance.**

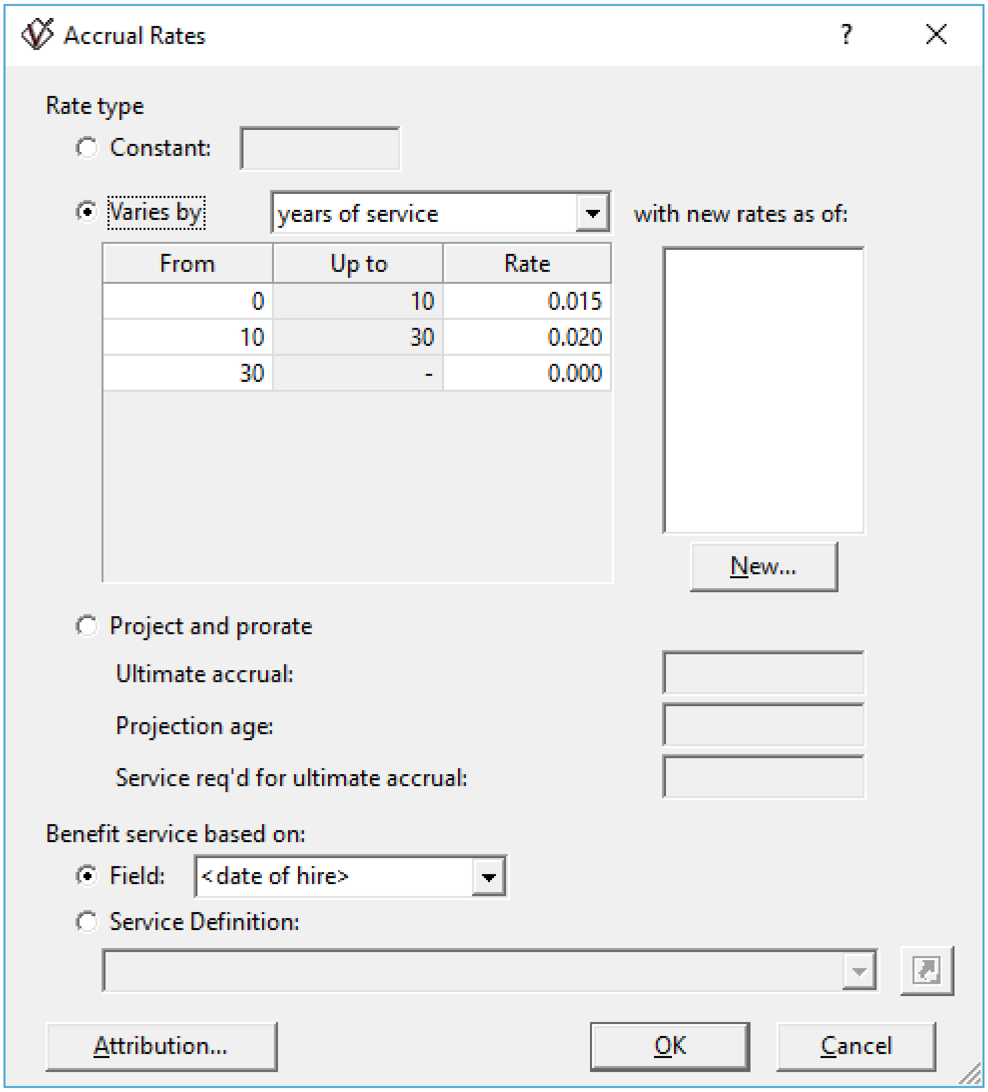
2. If the *interest crediting rate* is explicitly specified in the plan document and never changes, enter it directly here. Otherwise, choose **interest crediting rate specified in Valuation and Projection Assumptions**.

3. Click on **Basis Formula** and type in the *salary definition* using the operator #SALARY. Click **OK.**

   You will need to use a **Custom Operator** in place of #SALARY if your *salary definition* differs from your valuation salary definition (set in Census Specifications).



4. Click on **Accrual Rates** and enter the *% of pay* (as a number between 0 and 1). The rate will change to 0 if your plan has a *maximum service*. Click **OK.**



5. Click on **Accrued Benefit** and choose the **database field** containing the current cash balance on the valuation date. If your plan does not store the current cash balance, choose **expected value** to have ProVal estimate it for you. In German mode, you will also need to specify crediting parameters.

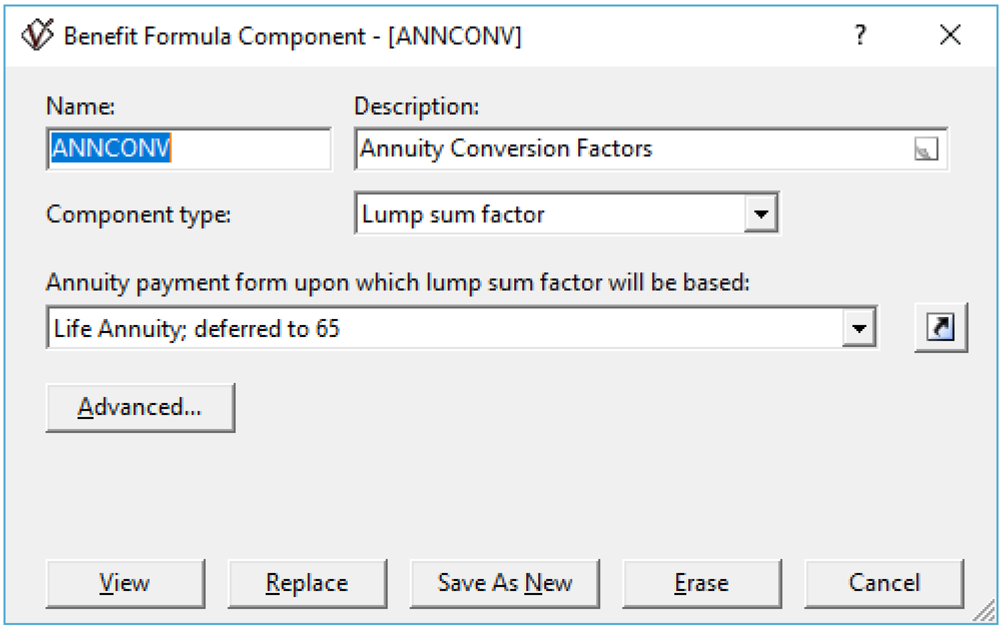
6. Click on **Projection & Freeze Ages, Crediting Frequency** if you want to change the crediting frequency from **annually** to **semi-annually**, **quarterly**, or **monthly**.

7. Click **Save As New.**

ANNCONV

Skip this component if you value the benefit paid as a lump sum rather than as an annuity.

1. In the **Benefit Formula Component** library, click **New** and enter a **name** for the component, for example, ANNCONV. Change the component type to **Lump Sum Factor.**



2. Click on the **Edit…** button next to the **annuity payment form**.

   Click the **New** button, define a payment form for a deferred-to-65 life annuity, and click **Save As New.**

3. Click **Save As New** to save the component.

The mortality and interest rate bases for ANNCONV will be specified in Step 4: Valuation Assumptions. If the conversion factors do not vary based on valuation assumptions, you can alternatively define ANNCONV using a **table** component rather than a **lump sum factor** component.

Termination Benefit

1. From the **Shortcuts** pane, expand **Plan Definitions**, and then click **Benefit Definitions**.

2. Click **New** and enter a **name** for the benefit definition, for example, “Termination Benefit.” Choose Termination as the **Contingency initiating benefits**.

3. If you have **Election Probabilities**, check the box. These probabilities will be specified in Step 4: Valuation Assumptions.

4. Create **Payment forms** from within the **Benefit Definition** by clicking the “back door button”

    a. Create and select the **Normal form** of payment, for example, a life annuity deferred to 65.

    b. Create any **Optional forms** you want to value, and check their boxes within the Optional forms topic. Conversion factors and probabilities of receipt will be specified in Step 4: Valuation Assumptions.

5. Enter **Eligibility criteria**, for example, 5 years of service.

6. Type in the **Benefit formula**:

   CASHBAL / ANNCONV

7. Click **Save as new**.

Benefits for the Retirement, Death, and Disability contingencies may be created with steps similar to those used for the Termination Benefit. Note that additional **Benefit Formula Components** may be required (e.g. an Early Retirement Factor table).

OPEB – NET CLAIMS, FOR ACTIVES OR INACTIVES

The following pages will guide you through setting up a retiree medical benefit formula using the general expression:

CLAIMS * AGING

If you prefer, the CLAIMS and AGING factors could be "pre-multiplied" outside of ProVal and entered as a single component.

Before you start coding, gather the following information (each *item* in italics is referred to in the steps that follow):

Item	Example	Your plan
Pre-Medicare claims	$5,200 / year for a 65-year-old	
Post-Medicare claims	$2,200 / year for a 65-year-old	
Aging (morbidity) adjustment	3.5% per year below 65, and 2.5% from 65 to 75, 1% from 76 to 80, and 0% after 80	

Benefit Formula Components

Create the necessary **Benefit Formula Component**: CLAIMS and AGING.

CLAIMS

1. In the **Benefit Formula Component** library, click **New** and enter a **name** for the component, for example, CLAIMS. Ensure that the component type is **Table.**

2. Select the option for **Table specified in Valuation Assumptions**. This option will allow you to specify a new claims table in each future valuation year.

3. Check **Apply increase rates to this component.**

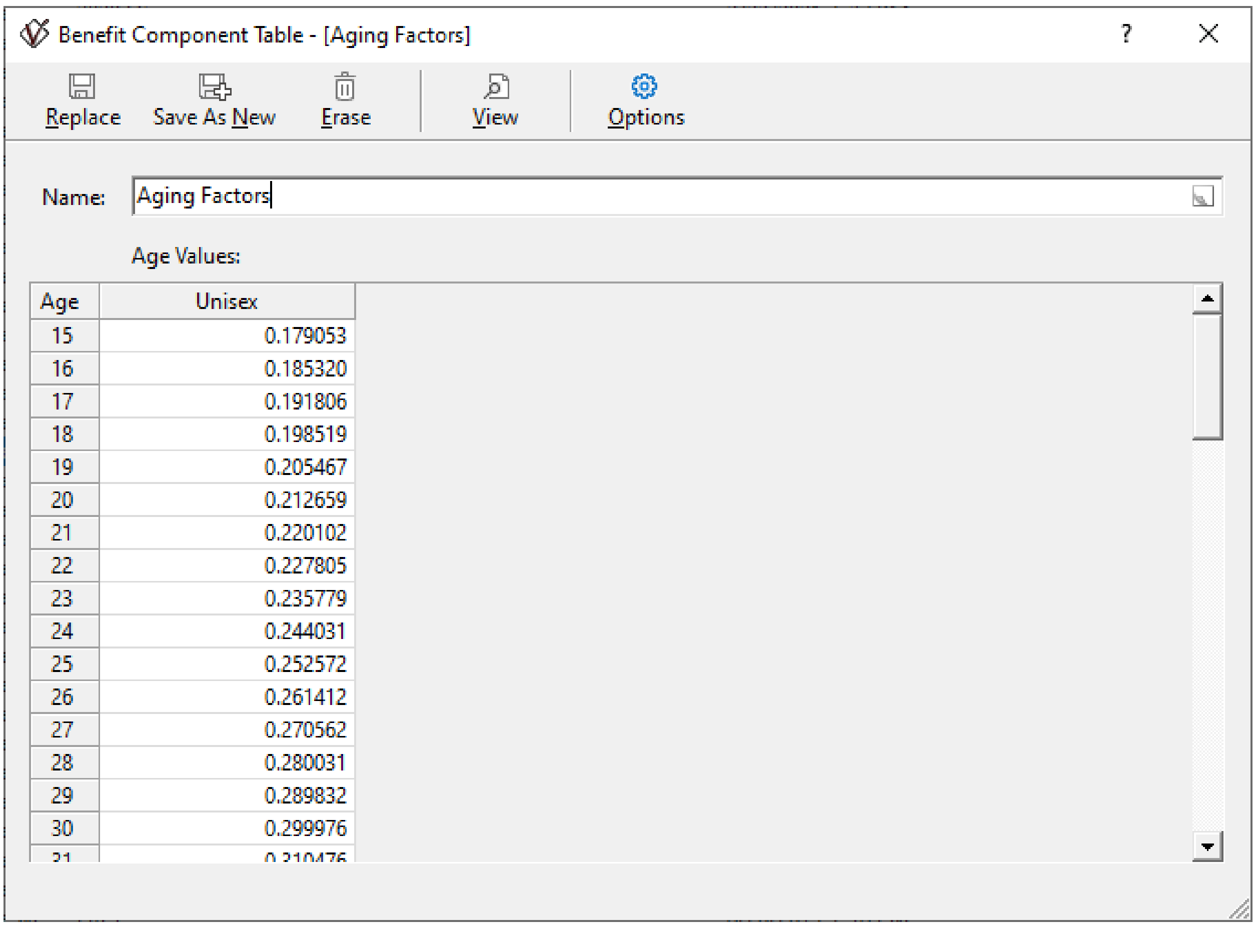
4. Click **Save As New.**

The values of for claims table will be specified in Step 4: Valuation Assumptions.

AGING

1. In the **Benefit Formula Component** library, click **New** and enter a **name** for the component, for example, AGING. Ensure that the component type is **Table.**

2. Create and select a **Table** with *Aging (morbidity) adjustment* factors. The factor should be 1 at age 65.



Age	Unisex
15	0.179053
16	0.185320
17	0.191806
18	0.198519
19	0.205467
20	0.212659
21	0.220102
22	0.227805
23	0.235779
24	0.244031
25	0.252572
26	0.261412
27	0.270562
28	0.280031
29	0.289832
30	0.299976

3. Ensure that **Apply increase rates to this component** is <u>NOT</u> checked**.**

4. Click **Save As New.**

Benefit for Future Retirees and Spouses (Current Actives)

1. From the **Shortcuts** pane, expand **Plan Definitions**, and then click **Benefit Definitions**.

2. Click **New** and enter a **name** for the benefit definition, for example, “Future Retirees and Spouses.”

3. Select that this benefit **Applies** to: Actives.

4. Select Retirement as the **Contingency**.

5. If you have **Election Probabilities**, check the box. These probabilities will be specified in Step 4: Valuation Assumptions.

6. Create and select the **Payment forms** which applies to the **Member** (for example a life annuity to member) from within the **Benefit Definition** by clicking the “back door button”

7. Create and select the Payment forms which applies to the Spouse (for example a life annuity to spouse) from within the Benefit Definition by clicking the “back door button”

8. Enter Eligibility criteria, for example, age 55 and 10 years of service.

9. Type in the **Gross Benefit Definition** formula:

   CLAIMS * AGING

10. Type in a **Participant Contribution** formula, if applicable. For example, if participant contributions are 30% of gross claims:

    0.3 * CLAIMS * AGING

11. Click **Save as new**.

Benefit for Current Retirees and Spouses

Same as for Future Retirees and Spouses above, with the following differences:

1. Click **New** and enter a **name** for the benefit definition, for example, "Current retirees and spouses."
2. Select that this benefit **Applies to**: Inactives.
3. Enter **Eligibility** requirements, if only some inactives are eligible. For example, if only division 3 and 4 retirees and spouses are eligible, enter a Selection Expression of "DIVISION #IN (3,4)." Or, if only those retirees who elected dental coverage (coverage code 3) are eligible, enter a selection expression of "COVCODE = 3."

Work smart: After saving your active benefit, make a **Copy** of the benefit, open the copy, make the few required changes for inactives, and **Replace.**

OPEB – GRADED INSURANCE

The following pages will guide you through setting up a graded life insurance formula of 2 x salary reduced 7.5% per year for 10 years after retirement.

We'll use the following benefit formula for actives:

2*SALARY*{1–[0.075*(10 #MIN [#PMTAGE–#DECAGE])]}

and this benefit formula for inactives:

2*INSURAMT*{1–[0.075*(10 #MIN [#PMTAGE–#DECAGE])]}

For inactives, #DECAGE refers to the decrement age specified in **Input | Census Specifications | Inactive Data.**

Benefit Formula Components

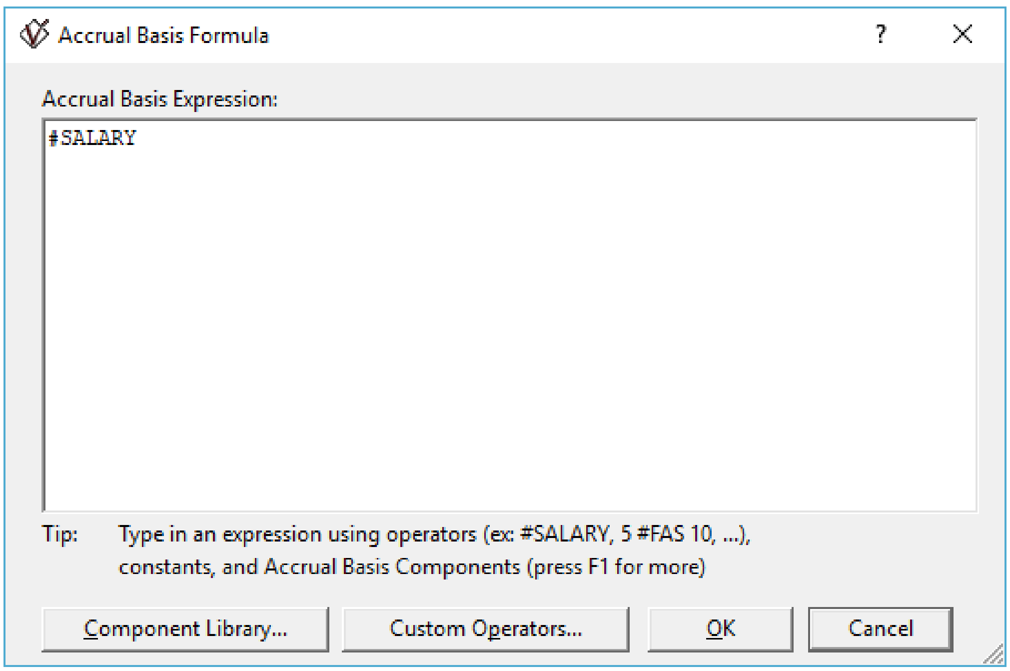
Create each necessary **Benefit Formula Component**: SALARY and INSURAMT.

SALARY

1. In the **Benefit Formula Component** library, click **New** and enter a **name** for the component, for example, SALARY. Change the component type to **Accrual Definition** and accrual format to **basis only.**

2. Click on **Basis Formula** and type in the employee's salary using the operator #SALARY. Click **OK.**

   You will need to use a **Custom Operator** in place of #SALARY if your salary definition differs from your valuation salary definition (in Census Specifications).



3. Click **Save As New.**

INSAMT

1. In the **Benefit Formula Component** library, click **New** and enter a **name** for the component, for example, INSAMT. Change the component type to **Database field.**

2. Select the **database field** which contains the original insurance amount, that is, the salary at decrement.

3. Click **Save As New.**

Benefit for Future Retirees and Spouses (Current Actives)

1. From the **Shortcuts** pane, expand **Plan Definitions**, and then click **Benefit Definitions**.

2. Click **New** and enter a **name** for the benefit definition, for example, “Future Retirees and Spouses.”

3. Select that this benefit **Applies** to: Actives.

4. Select Retirement as the **Contingency**.

5. If you have **Election Probabilities**, check the box. These probabilities will be specified in Step 4: Valuation Assumptions.

6. Create and select the **Payment forms** which applies to the **Member** (for example a life annuity to member) from within the **Benefit Definition** by clicking the “back door button”

7. Create and select the Payment forms which applies to the Spouse (for example a life annuity to spouse) from within the Benefit Definition by clicking the “back door button”

8. Enter Eligibility criteria, for example, age 55 and 10 years of service.

9. Type in the **Gross Benefit Definition** formula:

   2*SALARY*{1–[0.075*(10 #MIN [#PMTAGE–#DECAGE])]}

10. Click **Save as new**.

Benefit for Current Retirees and Spouses

Same as for Future Retirees and Spouses above, with the following differences:

1. Click **New** and enter a **name** for the benefit definition, for example, “Current retirees and spouses.”

2. Select that this benefit **Applies to**: Inactives.

3. Enter **Eligibility** requirements, if only some inactives are eligible. For example, if only division 3 and 4 retirees and spouses are eligible, enter a Selection Expression of “DIVISION #IN (3,4).” Or, if only those retirees who elected dental coverage (coverage code 3) are eligible, enter a selection expression of “COVCODE = 3.”

Work smart: After saving your active benefit, make a **Copy** of the benefit, open the copy, make the few required changes for inactives, and **Replace.**

OPEB – SEVERANCE PAY FOR ACTIVES

The following pages will guide you through setting up a severance pay formula for actives.

The severance plan in this example is 2 weeks of pay for each year of service, but not greater than 1.5 times pay. We'll use the following benefit formula:

SVRNCEBFT

BENEFIT FORMULA COMPONENTS

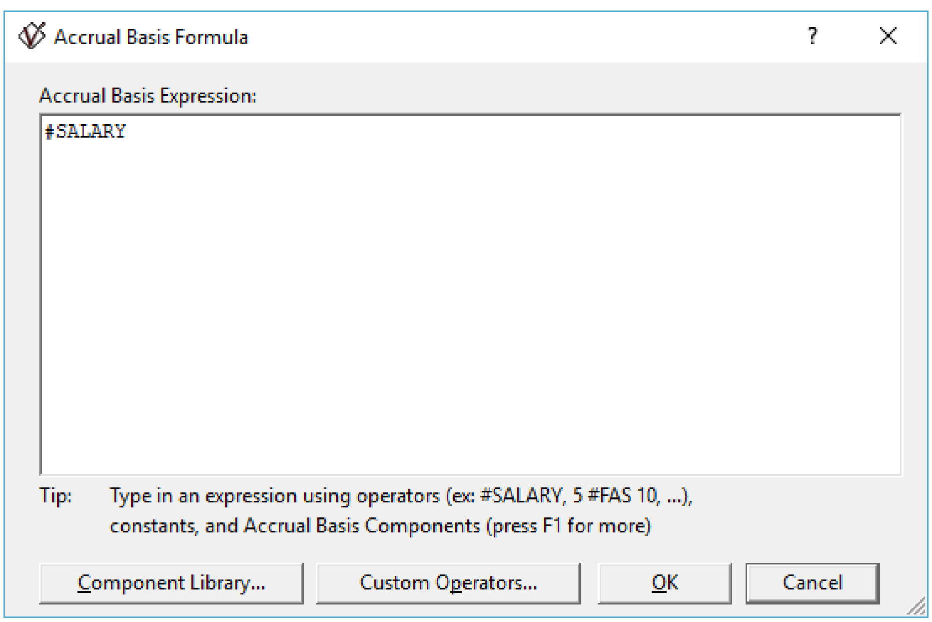
Create the necessary **Benefit Formula Component**: SVRNCEBFT.

SVRNCEBFT

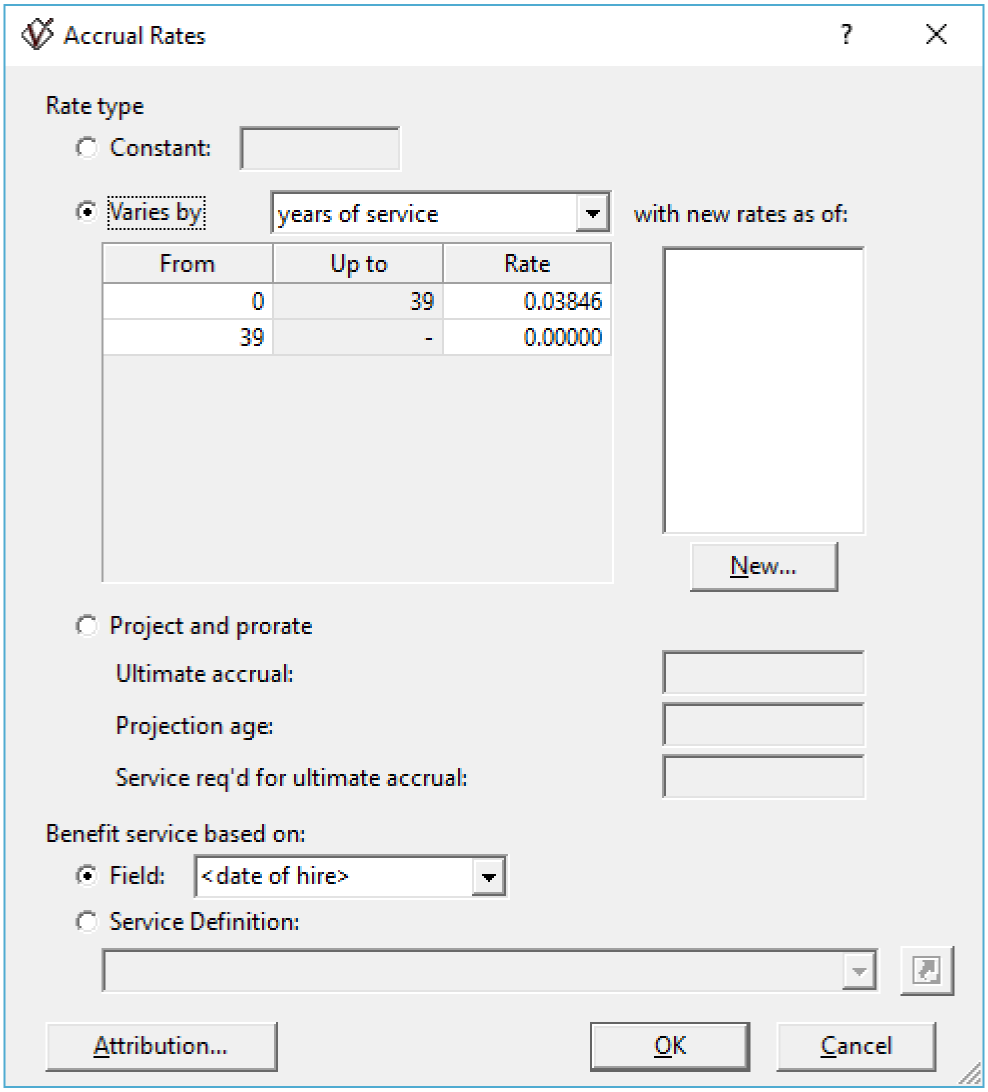
1. In the **Benefit Formula Component** library, click **New** and enter a **name** for the component, for example, SVRNCEBFT. Change the component type to **Accrual Definition** and accrual format to **Final average**.

2. Click on **Basis Formula** and type in the employee's salary using the operator #SALARY. Click OK.

   You will need to use a **Custom Operator** in place of #SALARY if your salary definition differs from your valuation salary definition (in Census Specifications).



3. Click on **Accrual Rates** and enter rates of 3.846154% (2 weeks/52 weeks per year) that go to zero after 39 years (1.5 / 3.846154% = 1.5 * 52 / 2).



4. Click **Save As New.**

Retirement Benefit For Actives

1. From the **Shortcuts** pane, expand **Plan Definitions**, and then click **Benefit Definitions**.

2. Click **New** and enter a **name** for the benefit definition, for example, "Severance benefit."

3. Select that this benefit **Applies** to: Actives.

4. Select Retirement as the **Contingency**.

5. If you have **Election Probabilities**, check the box. These probabilities will be specified in Step 4: Valuation Assumptions.

6. Create and select the **Payment forms** which applies to the **Member** (for example a life annuity to member) from within the **Benefit Definition** by clicking the "back door button"

7. Choose <none> for the **Payment form** which applies to the **Spouse**.

8. Enter **Eligibility criteria**, for example, age 55 and 10 years of service.

9. Type in the **Gross Benefit Definition** formula:

   SVRNCEBFT

10. Click **Save as new**.

GERMANY – BENEFIT PROMISES

In ProVal's German Pension mode, the plan benefits are defined using four principal building blocks: a **plan definition** ("plan"), **benefit promises** ("promises"), **benefit definitions** ("benefits"), and **benefit formula components** ("components"). A plan is a collection of promises and the promises are collections of benefit definitions. The benefit definitions specify formulas (among other things) which are written with benefit formula components.

Once you have set up **Benefit Definitions**, collect the benefits together in **Benefit Promises**. After creating the **Benefit Promises**, collect the promises together using the **Plan Definitions** command.

Note that Post-termination benefits for retirement, disability, and death are generated automatically whenever termination decrements are used. There is generally no need to include termination benefits in the **Benefit Promise**.

Here is a sample pension plan that illustrates how these building blocks are put together:

Plan Definition

Benefit Promise with Promise Type of Pension

Benefit Definition for old age

Contingency	Retirement
Payment Form	Life annuity
Eligibility Conditions	Age 55 & 10 years of service
Benefit Formula (using **Components**)	ERF * (BASE + EXCESS)

Benefit Definition for disability up to retirement age

Contingency	Disability
Payment Form	Life annuity temporary to retirement age[1]
Eligibility Conditions	5 years of service
Benefit Formula (using Components)	ERF * (BASE + EXCESS)

Benefit Definition for disability after retirement age[2]

Contingency	Disability
Payment Form	Life annuity deferred to retirement age
Eligibility Conditions	5 years of service
Benefit Formula (using Components)	BASE + EXCESS

Benefit Definition for death in service

Contingency	Death
Payment Form	Life annuity
Eligibility Conditions	5 years of service
Benefit Formula (using Components)	BASE + EXCESS

Benefit Definition for death in disability prior to retirement age[3]

Contingency	Disability
Payment Form	Joint life annuity, temporary to retirement age[4]
Eligibility Conditions	5 years of service
Benefit Formula (using Components)	ERF * SpRed * (BASE + EXCESS)

Benefit Definition for death in disability after retirement age[3,5]

Contingency	Disability
Payment Form	Joint life annuity deferred to retirement age[6]
Eligibility Conditions	5 years of service
Benefit Formula (using Components)	SpRed * (BASE + EXCESS)

Benefit Definition for death in old age[7]

Contingency	Retirement
Payment Form	Joint life annuity[6]
Eligibility Conditions	5 years of service
Benefit Formula (using Components)	ERF * SpRed * (BASE + EXCESS)

1. If the disability benefit after retirement age is the same as the disability benefit prior to retirement age then only one disability **Benefit Definition** is required and the **Payment Form** would generally be a life annuity with no temporary period.

2. This **Benefit Definition** is not required when the benefit prior to retirement age is the same as the disability after retirement age.

3. If the death in disability benefits are a simple percentage of the member benefit, the death in disability benefit may be combined with the disability benefit. In the **Payment Form** for the disability benefit reflect a joint life annuity where the fraction of the benefit payable when only the beneficiary is alive reflects the percentage of member benefit payable.

4. The joint life parameters of the **Payment Form** for this **Benefit Definition** should generally reflect that 0% of the benefit is payable unless only the beneficiary is alive, in which case 100% of the benefit is payable. If the death in disability benefit prior to retirement age is the same as the benefit after retirement age, then only one death in disability **Benefit Definition** is required and the payment form would generally be a life annuity with no temporary period.

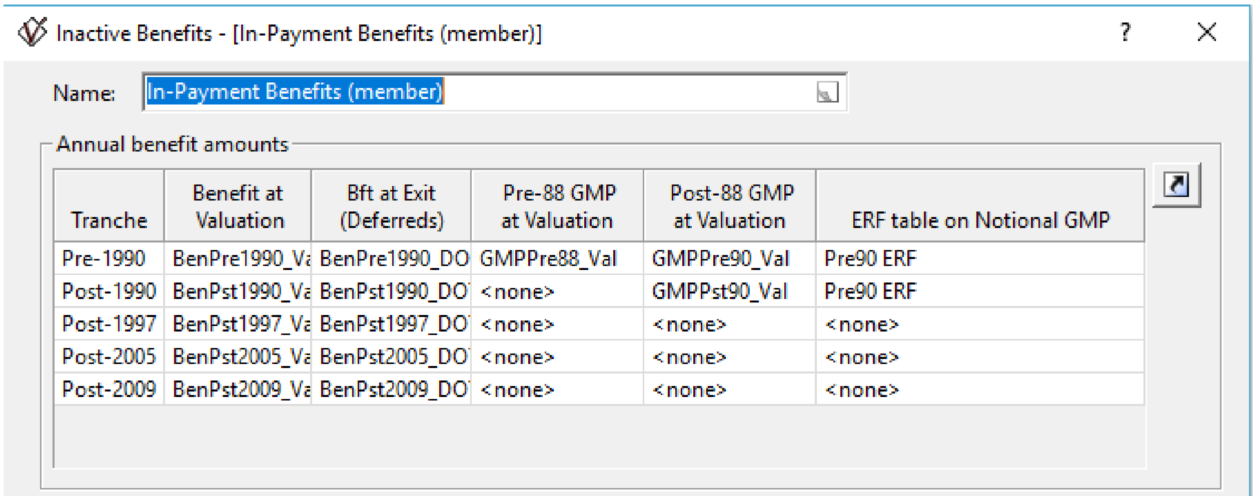
5. This **Benefit Definition** is not required when the death in disability benefit prior to retirement age is the same as the death in disability benefit after retirement age.

6. The joint life parameters of this **Payment Form** should generally reflect that 0% of the benefit is payable unless only the beneficiary is alive, in which case 100% of the benefit is payable.

7. If the death in old age benefit is a simple percentage of the member benefit, the death in old age benefit may be combined with the old age benefit. In this case, the **Payment Form** for the **Benefit Definition** for old age can reflect a joint life annuity where the fraction of the benefit payable when only the beneficiary is alive reflects the percentage of member benefit payable.

U.K. – MULTIPLE TRANCHES

In ProVal's U.K. Pension mode, the plan benefits are defined similar to other pension modes, except that users are allowed to specify multiple service tranches within each benefit definition in **census specifications** (fo inactives):


Inactive Benefits - [In-Payment Benefits (member)]

Name: In-Payment Benefits (member)

Annual benefit amounts

Tranche	Benefit at Valuation	Bft at Exit (Deferreds)	Pre-88 GMP at Valuation	Post-88 GMP at Valuation	ERF table on Notional GMP
Pre-1990	BenPre1990_Va	BenPre1990_DO	GMPPre88_Val	GMPPre90_Val	Pre90 ERF
Post-1990	BenPst1990_Va	BenPst1990_DO	<none>	GMPPst90_Val	Pre90 ERF
Post-1997	BenPst1997_Va	BenPst1997_DO	<none>	<none>	<none>
Post-2005	BenPst2005_Va	BenPst2005_DO	<none>	<none>	<none>
Post-2009	BenPst2009_Va	BenPst2009_DO	<none>	<none>	<none>

and **benefit definitions** (for actives):

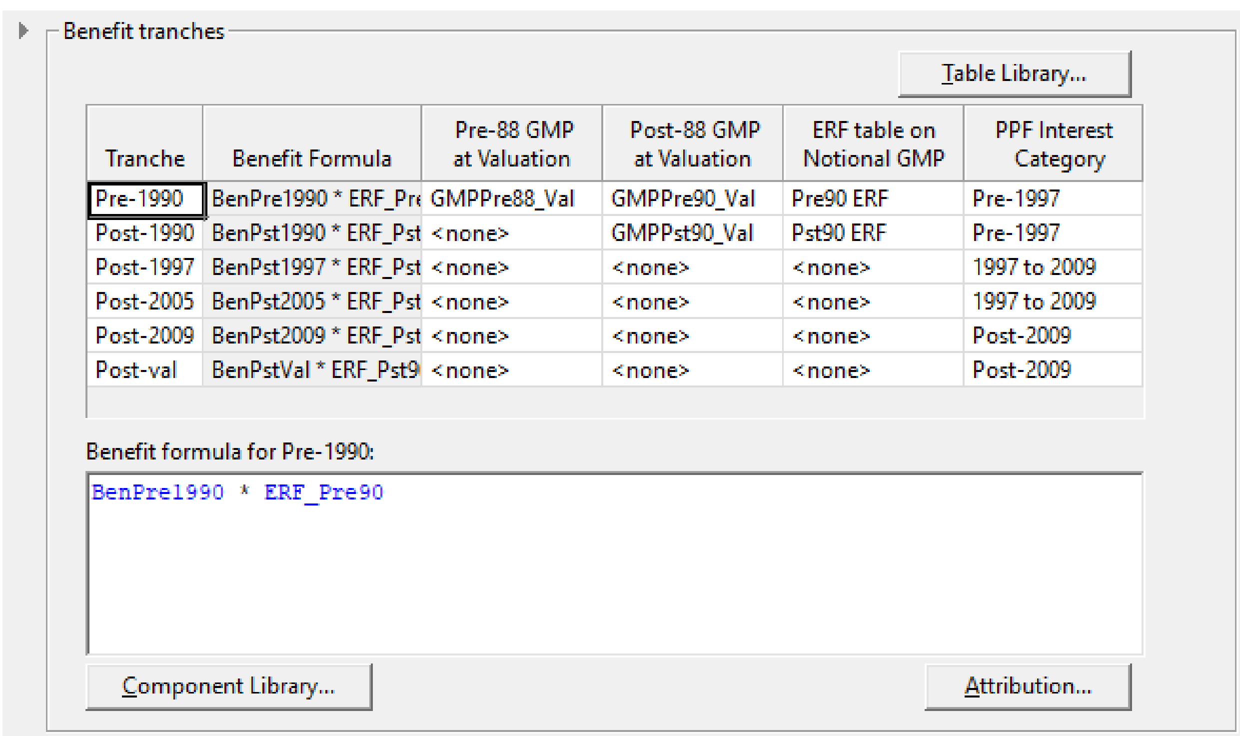

Benefit tranches

Table Library...

Tranche	Benefit Formula	Pre-88 GMP at Valuation	Post-88 GMP at Valuation	ERF table on Notional GMP	PPF Interest Category
Pre-1990	BenPre1990 * ERF_Pre	GMPPre88_Val	GMPPre90_Val	Pre90 ERF	Pre-1997
Post-1990	BenPst1990 * ERF_Pst	<none>	GMPPst90_Val	Pst90 ERF	Pre-1997
Post-1997	BenPst1997 * ERF_Pst	<none>	<none>	<none>	1997 to 2009
Post-2005	BenPst2005 * ERF_Pst	<none>	<none>	<none>	1997 to 2009
Post-2009	BenPst2009 * ERF_Pst	<none>	<none>	<none>	Post-2009
Post-val	BenPstVal * ERF_Pst9	<none>	<none>	<none>	Post-2009

Benefit formula for Pre-1990:

```
BenPre1990 * ERF_Pre90
```

Component Library... Attribution...

The examples above contain four tranches, but you may define additional (or fewer) tranches. Each service tranche corresponds to the benefit earned during a specific calendar period. While different benefit formulas, GMPs and pension increases might apply to benefits earned during those periods, they often share the same contingency, payment form and eligibility conditions. In sample lives, the service tranches within each benefit definition are split off into separate reports. For example, if four tranches are defined within a benefit definition then four reports will appear in the sample life exhibits.

STEP 4: VALUATION ASSUMPTIONS

ProVal has two types of valuation assumptions: funding and accounting. Enter them as distinct sets of assumptions. In the next step, they will be run together as a single valuation.

Please note that all of the following table libraries can be accessed from **Input | Reference Tables** on the menu or from the **Shortcuts** pane.

1. Enter your mortality and decrement tables using the **Input | Reference Tables** commands for Mortality Rates, Disability Rates, Termination Rates, and Retirement Rates.

   For mortality rates, the desired table generally exists. If it is hidden (i.e., is not displayed), simply **unhide** it from the Universe project.

2. If your salary scale varies by age, service, or sex, enter the merit scale (merit scale = [1+ total salary scale] / [1 + salary inflation] – 1) into a **Salary Merit Scale** table. If you have a scalar salary scale assumption (e.g., 5%), you do not need to create a table; you can enter it directly in valuation assumptions.

   a. If you wish to apply a second merit scale, you may do so by clicking the  next to the Salary Merit Scale Params… button. In this case, a second merit scale drop down will appear. The total increase will be (1+inflation)*(1+merit scale 1)*(1+merit scale 2).

3. If you assume rates of participation (common in OPEB plans) or election (e.g., annuity vs. lump sum, DROP vs. non-DROP) that vary by age, service, or sex, enter the rates into a **Post Decrement Probabilities** table. If you assume a scalar rate (e.g., 80%), you do not need to create a table; you can enter it directly in valuation assumptions.

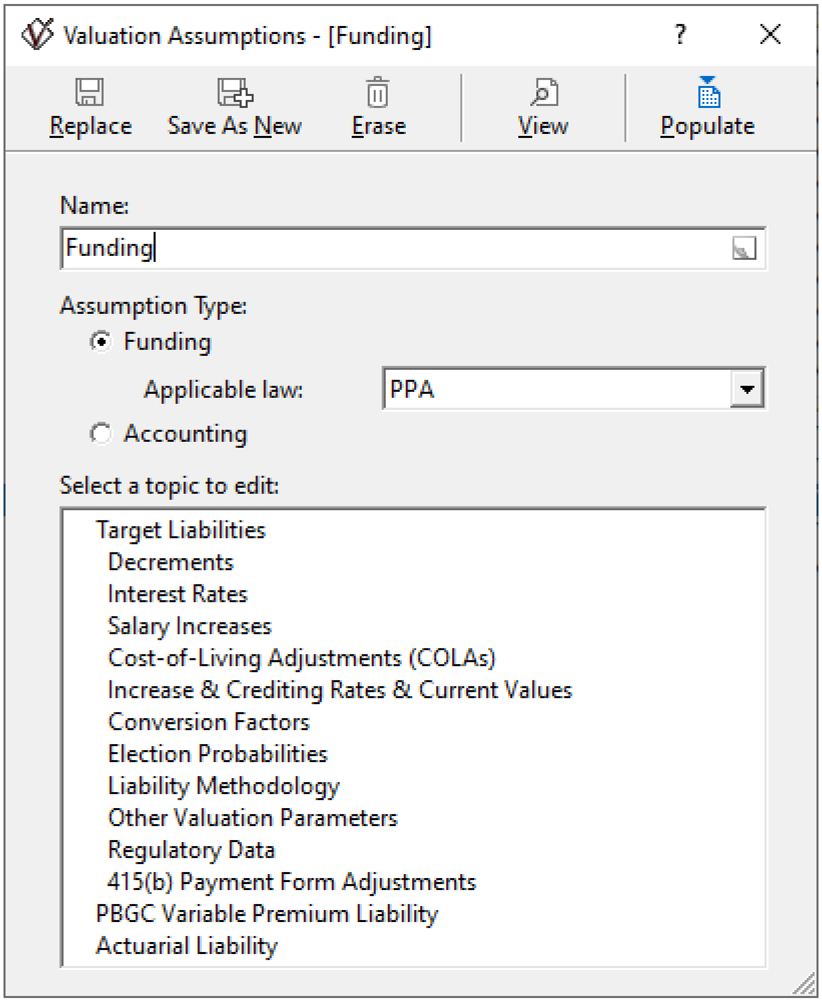
4. For OPEB plans, enter trend into an **Increase Rate Table**.

5. For OPEB plans with claims tables which are specified in the valuation assumptions, enter the pre- and post-Medicare claims into an age based **Benefit Component Table.**

6. If you assume COLAs vary by age or duration from commencement, enter them into a **COLA Rate Table**.

7. Either from the **Input** menu or **Shortcuts** pane**,** enter the **Valuation Assumptions** command to save separate assumptions for both funding and accounting, as applicable.

Work smart: After saving your funding assumptions, make a Copy of them, open the copy, make the few required changes for accounting, and **Replace.**



Fill out each of the topics. Here are a few notes about certain topics:

- **Applicable Law**. Applies to U.S. Qualified Pension and U.K. Pension modes only. Determines the legislated liabilities which apply to the valuation, such as, Current liabilities versus Target liabilities (for U.S.).

- **Decrements**. Refer to the decrement and mortality tables you specified above.

- **Interest Rates**. Enter the interest (a.k.a. discount) rate.

- **Salary Increases**. Enter the salary scale. Refer to the merit salary scale, if any, that you specified above. Note that the total salary scale = (1 + salary inflation) * (1 + salary merit scale).

- **Cost-of-Living Adjustments**. For pension plans, enter the assumed increase in benefits after decrement. Refer to the COLA Rate Tables, if any, that you specified above.

- **Increase & Crediting Rates**. For pension plans, enter the assumed increase in the maximum benefit limit, maximum compensation limit, social security wage base, and social security national average wage. For OPEB plans, enter the assumed increase in benefit formula components (i.e., trend). Refer to the Increase Rates Tables, if any, that you specified above. For OPEB plans where tables are specified in the valuation assumptions, refer to the Benefit Component Tables that you specified above. **Lump Sum & Optional Payment Forms**. If you are using **Lump Sum Factors** and/or **Optional Payment Forms** in your **Benefit Definitions**, you must fill in this topic. Enter interest and mortality assumptions for **Lump Sum Factors**. Enter conversion factors (or interest and mortality basis for conversion) for **Optional Payment Forms**.

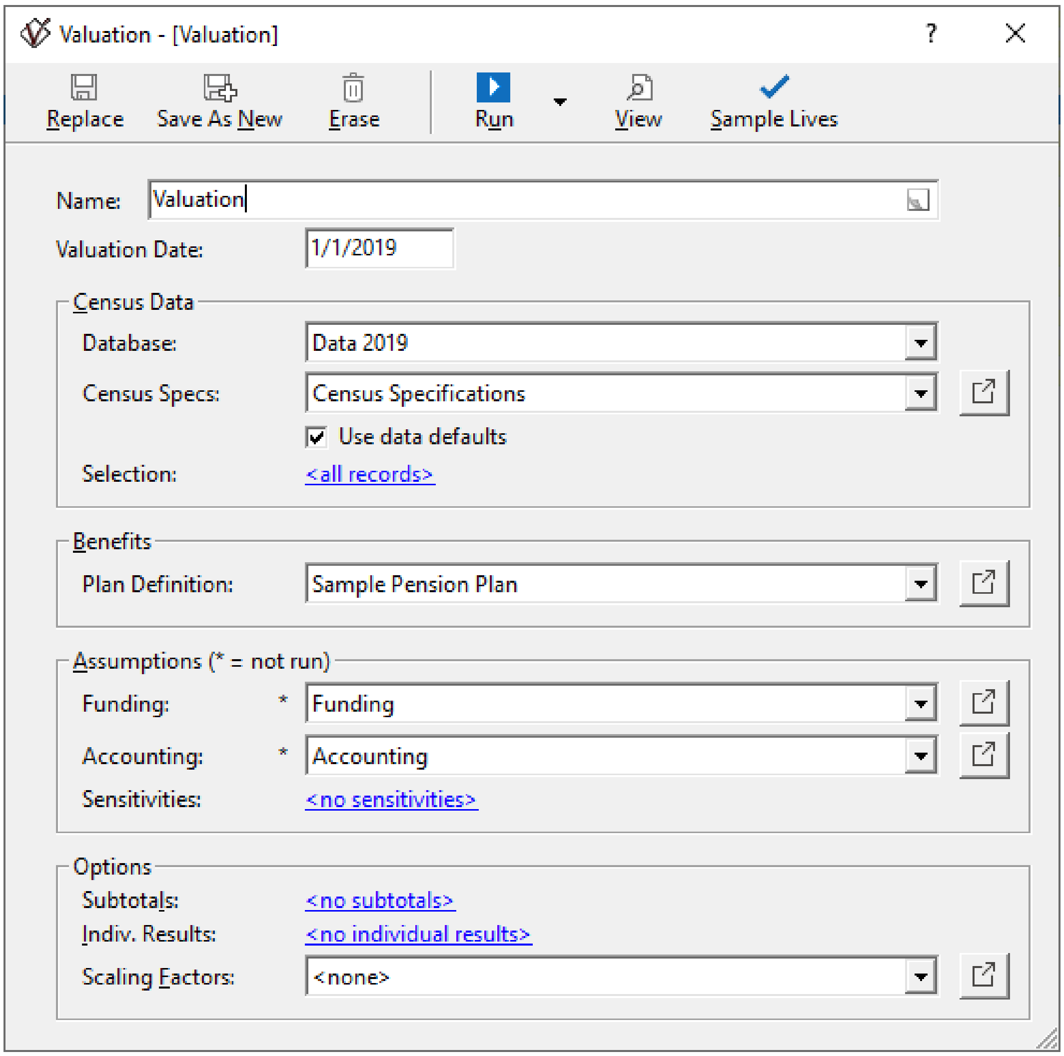
- **Election Probabilities**. Enter any probabilities of receipt for **Benefit Definitions** as well as for **Optional Payment Forms (e.g., 30% annuities/70% lump sums)**.

- **Liability Methods.** For funding assumptions, choose the family (or families) of cost methods your plan uses. The Entry Age Normal family includes the spread-gain cost methods (i.e., Aggregate, FIL with PUC, Frozen Attained Age, Frozen Entry Age (FIL), and Individual Aggregate) as well as Entry Age Normal. The cost method will be specified later in Step 6: Asset & Funding Policy.

STEP 5: VALUATION

A valuation calculates liabilities and normal costs based on the data, plan benefits, and valuation assumptions you specified in steps 2-4.

SET UP A VALUATION / SAMPLE LIFE



Using the Valuations command from either the Execute menu or the Shortcuts pane, specify the:

- **Valuation date**

- **Census data.** Specify the database and census specifications.

  If you wish to run only a subset of the population in the database, enter a selection expression. For example, “(Division #in (1,5,9)) #AND (Status = 1)” or “Hiredate <= 1/1/2000”.

- **Plan definition**. For pension plans, select “<Inactives only>” if there are no active participants which would require a plan definition.

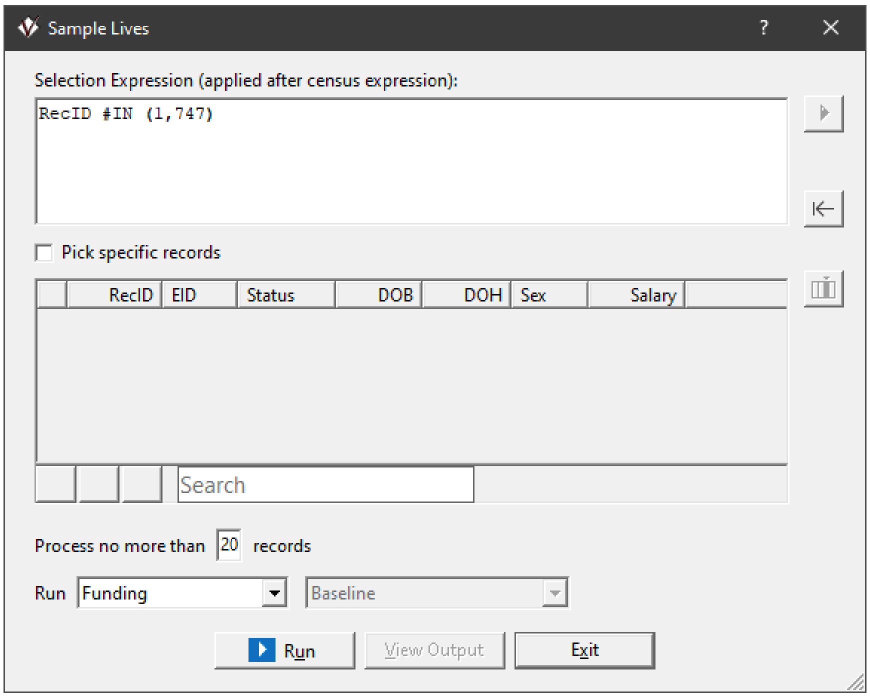
- **Valuation assumptions** for funding and accounting. Select “<none>” if not applicable (e.g., funding assumptions would typically not be applicable to an OPEB plan).

- If you wish to **subtotal** results by division, job class, etc., select a coded field(s) for subtotaling. Selecting a status field for subtotaling is generally not necessary since this is done automatically.

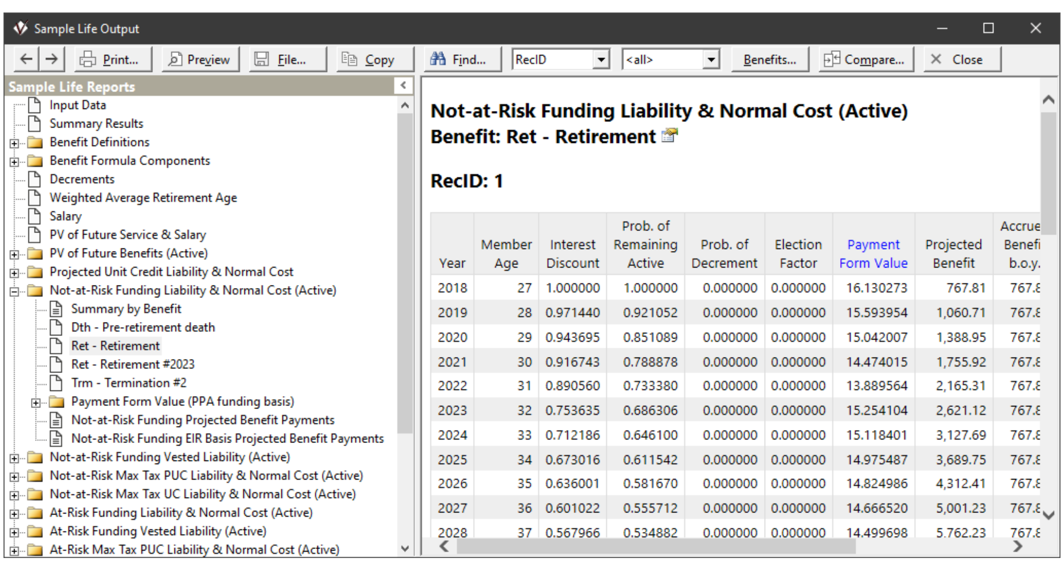
- If you wish, turn on **Individual Results.** This allows you to save selected liability and benefit results by participant back to a ProVal database.

INSPECT SAMPLE LIVES

Click the **Sample Lives** button and specify a selection expression that selects one or more records (ProVal will only run the first 20). For example, “SSN #IN (111-11-1111, 222-22-2222)” or “RecID = 1”. Click the **Run** button.



Once the sample life runs, click **View Output** to open the sample life tree viewer**.**



The pane on the left side of the screen displays a “tree” which contains single report labels, as well as folders containing additional report labels. Click on a folder’s “+” symbol to expand it or click on a report label to display the report. Use the ↑and ↓ arrow keys to move through tree. Use the → and ← arrow keys to expand and collapse individual folders. Right click on the tree to select the “Expand All” and “Collapse All” (folders) options.

From this screen you can:

- change the field used to identify records (RecID is the default),
- select an individual sample life to display or select all lives at once,
- use the Benefits button to select benefits for detailed output or decrement ages to display, or
- use the Print or File Button to print or save to file all of the tables.

Note: the sample life liability report options are dependent on the computational Mode (ie: U.S. Qualified Pension, OPEB, Canadian Registered Pension, etc.) as well as the Applicable Law (found in Valuation Assumptions in U.S. Qualified Pension mode).

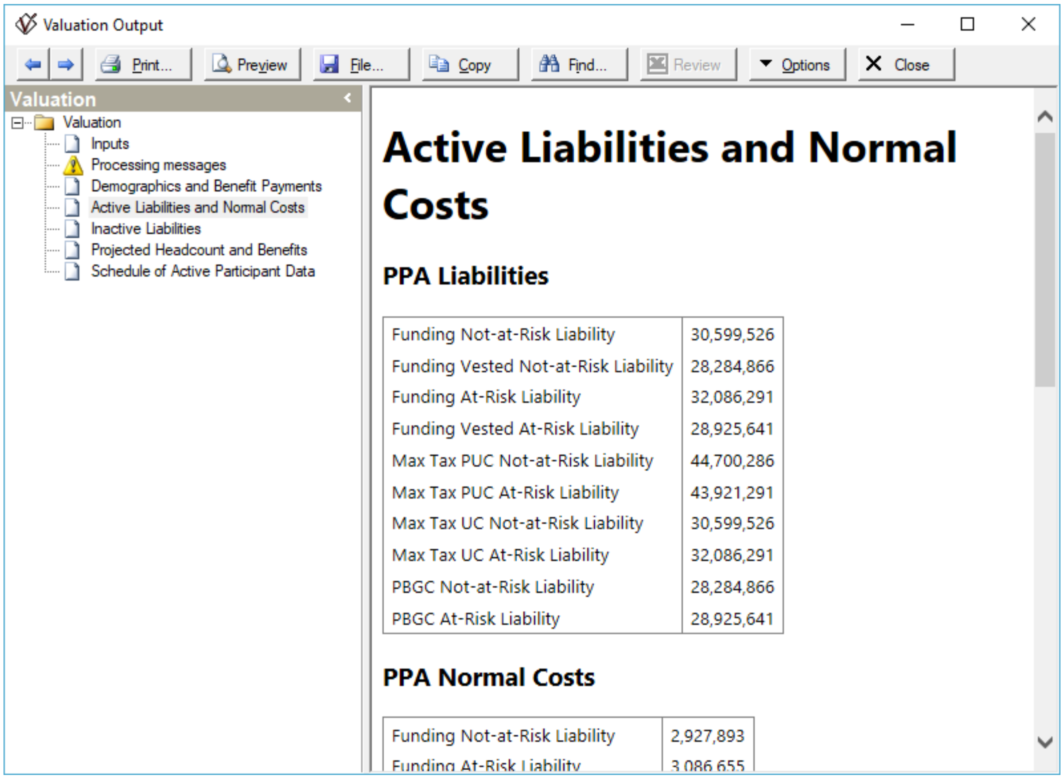
Here’s some advice on checking sample lives, particularly when comparing results to another system:

- Start with the Present Value of Future Benefits report (a.k.a. EBO or EPBO for accounting). Check each column. Where you find differences, dig deeper for details.
- Next check the report for your cost method (e.g., Projected Unit Credit or Entry Age Normal) for those things which are different from Present Value of Future Benefits. That is, attribution, funding period, etc.

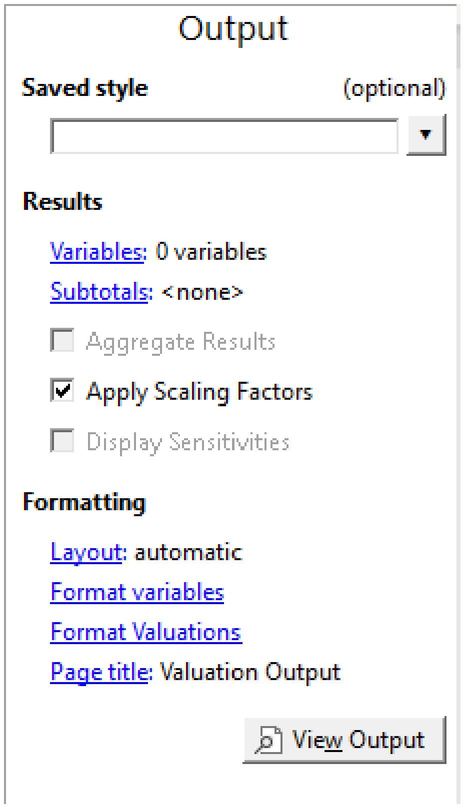
RUN A VALUATION

Click the **Run** button.

Once the valuation finishes running, click the **View** button to view the output.



You can view more detailed output, compare multiple runs, and more using the **Output** pane.

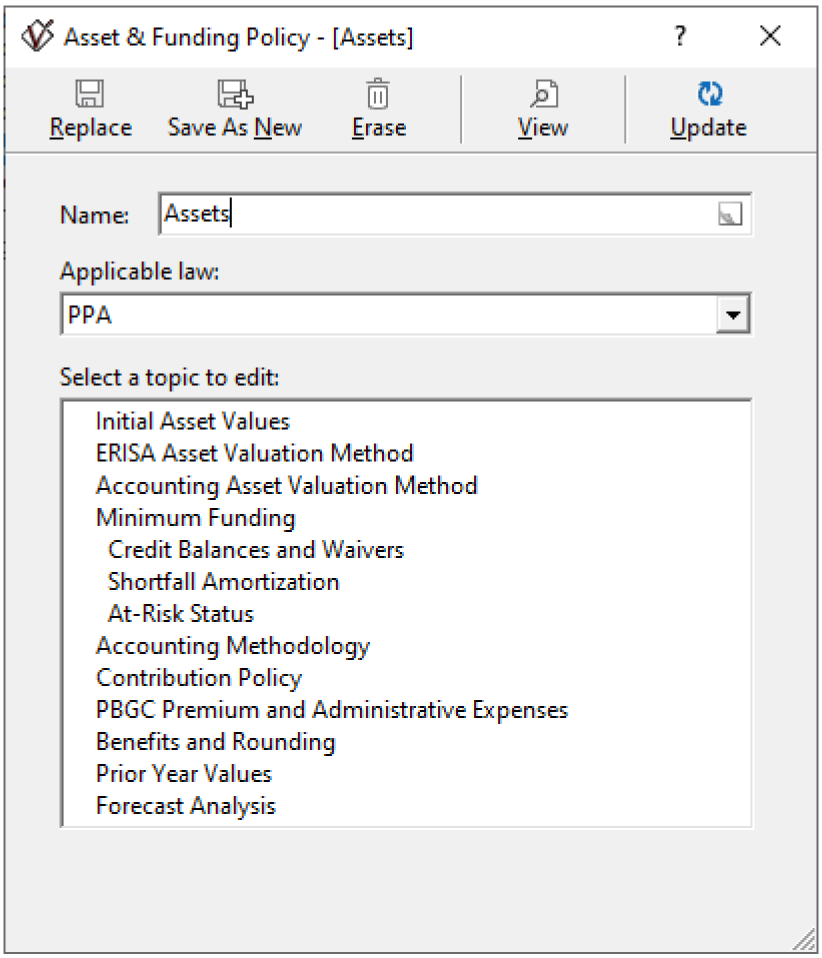


Click Variables to select the specific items (demographic, liability, projected benefits, etc.) that you want to view in the output. You may also elect to show Subtotals, Apply Scaling Factors, as well as, format the output using parameters in the **Output** pane. Click the **View Output** button to display the output.

STEP 6: ASSET & FUNDING POLICY

Additional information besides the valuation's liabilities – asset values, amortization bases, etc. – is needed to calculate contributions and expense.

Use the **Asset & Funding Policies** command to enter asset values, amortization bases, etc. The **Asset & Funding Policies** library can be accessed from either the **Input** menu or the **Shortcuts** pane.



Fill out each of the topics. Here are a few notes about certain topics:

- **Applicable Law**: Applies only to U.S. Qualified Pension mode. Determines the legislated funding rules for calculating contributions in a Valuation Set (or forecast).

- **Initial Asset Values.** For funding, enter the market value of assets as of the valuation date. For accounting, enter the assets as of the measurement date. If the measurement date of the assets does not match the valuation date for the liabilities (as run in the valuation), ProVal will automatically roll forward the liabilities to the measurement date.

- **Amortization Bases,** including those specified under **Accounting Methodology**. Enter the prior year's amortization bases rolled forward to the valuation date. For example, don't enter this year's gain/loss base and don't re-amortize the bases if the interest rate will change this year; ProVal will automatically adjust for these in Step 7: Valuation Set. If this is the second year you're using ProVal, you can roll forward last year's bases automatically by clicking the **Update…** button.

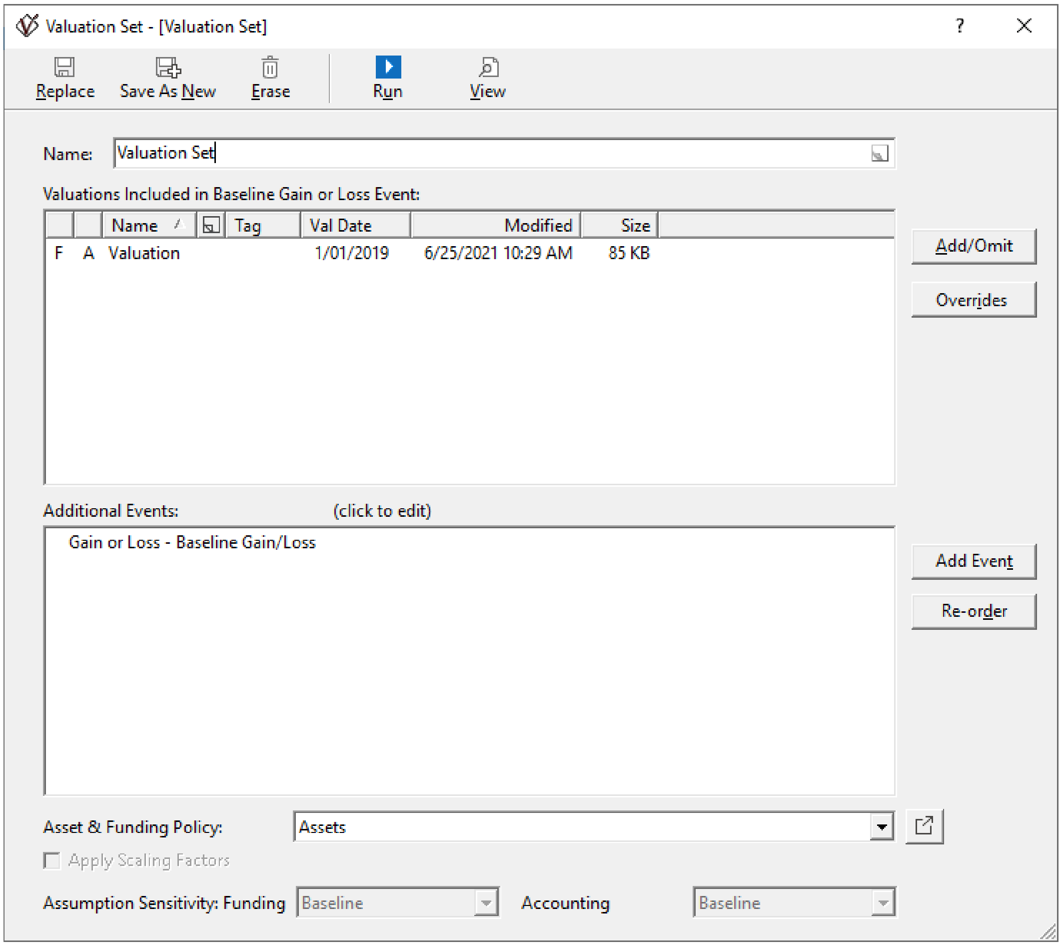
  Set the amortization schedule dates to the valuation date.

- **Contribution Policy.** Specify the actuarial cost method for determining the plan's normal cost.

  Specify when contributions will be made. To specify exact dates for contributions made to date, enter a contribution schedule.

STEP 7: VALUATION SET

A Valuation Set calculates funding contributions and accounting expense using the liabilities calculated in a Valuation and assets entered in an Asset & Funding Policy.



Using the **Valuation Sets** command from either the **Execute** menu or the **Shortcuts** pane, specify the:

- **Baseline valuation(s)**. If there is a plan change or assumption change this year, these valuations are the "old plan, old assumption" runs.

- **Additional events.** Add events if there are plan changes, assumption changes, or other changes from the prior year.

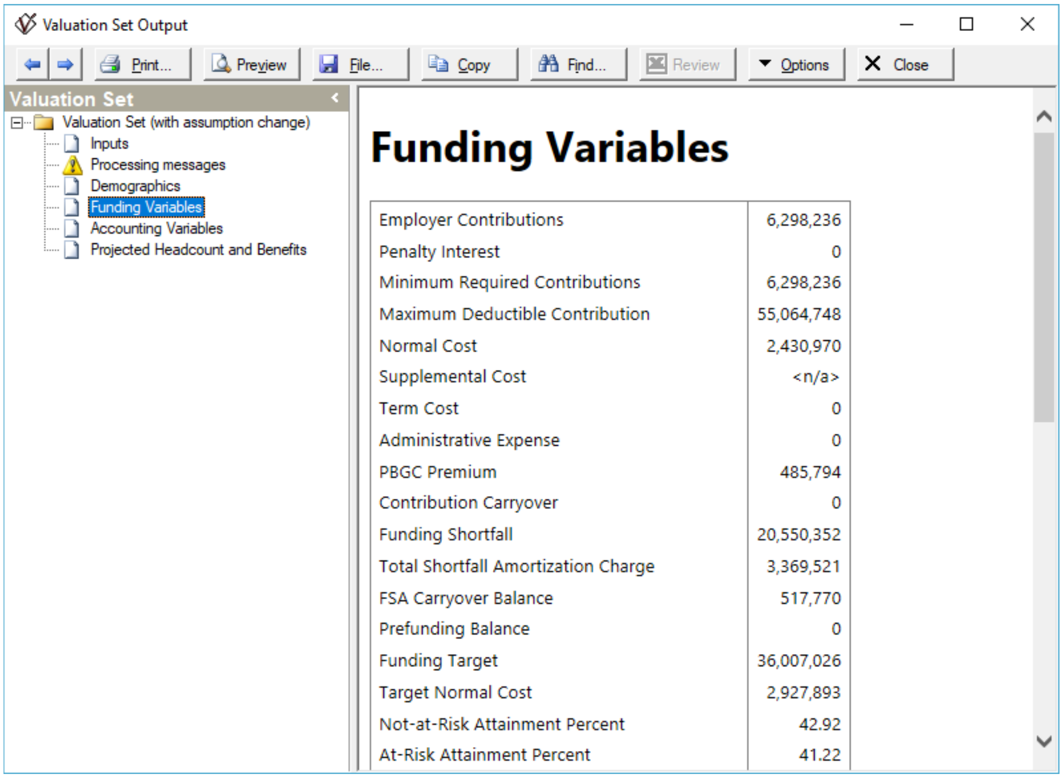
For the above, if you select more than one valuation, they will automatically be added together. Also, you can substitute, or override, certain liabilities with the results of separate valuations if desired.

- **Asset & Funding Policy**

Step 7: Valuation Set

Click the **Run** button.

Once the valuation set finishes running, click the **View** button to view the output.



You can view more detailed output, compare multiple runs, and more using the **Output** pane.

Detailed work papers are available through **Output | Valuation Set Exhibits** or from the **Valuation Set Exhibits** link found at the bottom of the **Output** pane.

APPENDIX A: EXPRESSIONS

Expressions are used within ProVal to:

- define new database fields;
- specify subsets of a database; and
- define benefit formulas.

This section describes the syntax of ProVal expressions.

EXPRESSION BASICS

Expressions may involve database field names, numbers, and a wide variety of operators, including the following standard arithmetic operators:

- + add
- * multiply
- - subtract
- / divide

For example, the expression:

(Salary1+Salary2)/2

can be used to define a new database field that is the average of the fields Salary1 and Salary2.

The expression is evaluated on a record-by-record basis and produces a result having a value for each record. For example, if the input fields have the values shown below, the result of the expression would be the values shown in the column labeled “Average.”

Record	**Salary1**	**Salary2**	**Average**
1	10,000	15,000	12,500.0
2	5,000	8,005	6,502.5
3	15,000	17,000	16,000.0
4	8,000	14,000	11,000.0
5	20,000	32,000	26,000.0

ProVal provides more than three dozen operators for use in expressions. Using a precedence scheme to decide which operators to execute first (as is done in many languages) would lead to a large and complicated set of rules. To avoid this problem, ProVal expressions use a simple, easily remembered rule: ***ProVal starts at the left and works its way to the right, executing operators as they are encountered in the expression***. There is no precedence among operators. Thus, the expression above would produce the same result if it were written as:

Salary1+Salary2/2

However, a right argument that is itself an expression (as opposed to a name or constant) *must* be enclosed in parentheses. For example, the expression:

Salary2-Salary1/Date2-Date1

is evaluated as:

((Salary2-Salary1)/Date2)-Date1

If the right argument to the division (/) is supposed to be Date2-Date1, it needs to be enclosed in parentheses, as in:

(Salary2-Salary1)/(Date2-Date1)

Left arguments don't need surrounding parentheses, but including them sometimes helps clarify the meaning of the expression (as done with Salary2-Salary1 above).

ProVal allows any of the following pairs of parenthetical symbols to be used in expressions: ( ), [ ], or { }. Judicious use of these alternative symbols can help clarify expressions containing many nested parentheses.

Only a few of the operators available in ProVal have symbolic names (like + and *); most have word names. All non-symbolic operator names are preceded by a pound sign (#), as in #MAX. This helps you distinguish between operator names and field names in your expressions, and it avoids requiring operator names to be reserved words that you cannot use as field names.

Unlike many languages, ProVal uses an "infix" notation for all operators, even those which have non-symbolic names. For example, in most languages you would compute the larger of two numbers using Max(Salary1, Salary2). In ProVal this is written as:

Salary1 #MAX Salary2

Although this may appear strange at first, it is really quite simple: named operators are used in exactly the same way as symbolic operators. The operator name is written *between* the arguments. You may wonder how this notation permits you to compute the maximum of three numbers, something conventionally written as Max(x,y,z). The answer is, the same way as you add three numbers:

sum:	x	+	y	+	z
maximum:	x	#MAX	y	#MAX	z

You can insert spaces in ProVal expressions freely without changing the meaning of the expression. On the other hand, you *must* put a space between adjacent names (whether operator names or field names) to prevent ProVal from interpreting the result as a single name.

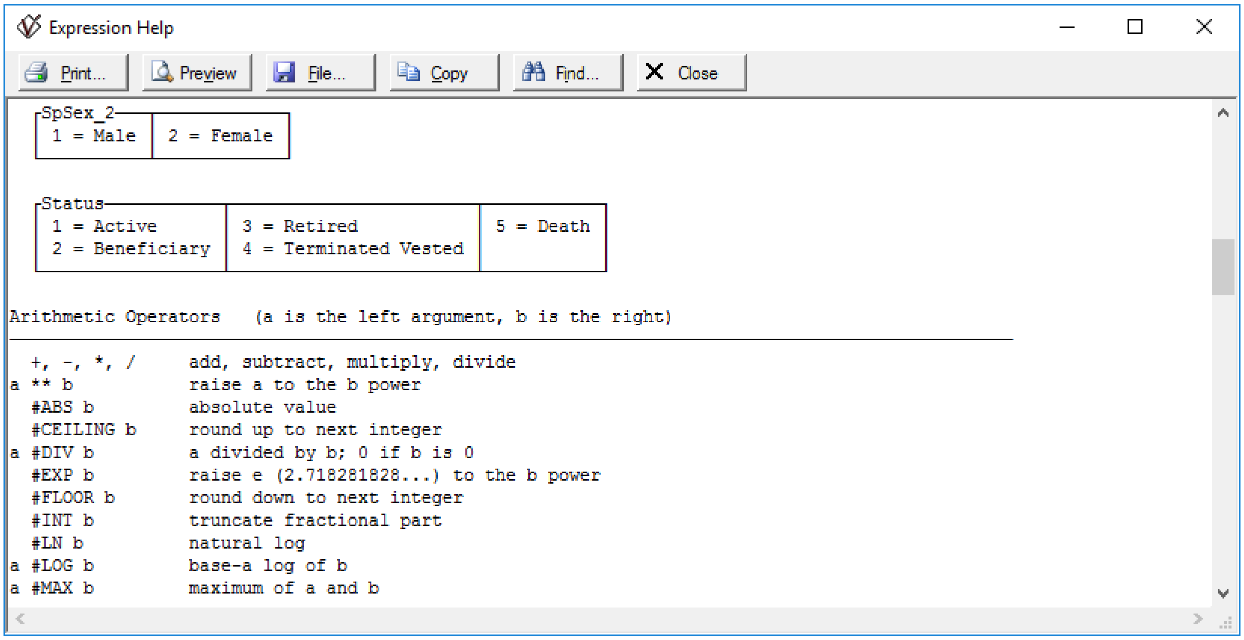
Note also that ProVal is *not case-sensitive*. In this manual, operator names are written in uppercase and field names in upper- and lowercase, but you can use whatever case you like when entering expressions.

The following is a partial list of the most common arithmetic operators used in ProVal.

Operator	Description
x + y	adds x and y
x – y	subtracts y from x
x * y	multiplies x and y
x / y	divides x by y
x ** y	raises x to the power of y
#ROUND y	rounds y to the nearest integer
#INT y	rounds y down to an integer (truncate)
x #MAX y	maximum (larger) of x and y
x #MIN y	minimum (smaller) of x and y
x #ZMINUS y	x minus y, not less than 0

EXPRESSION HELP

When you are entering an expression, the F1 key can be used to display an expression help window, as illustrated below.



This help screen

- lists the fields defined in the database;
- shows the numeric codes that the user specified to represent coded fields; and
- describes the operators available for use in expressions.

The cursor must be in an expression entry field for F1 to display this help message.

MISSING VALUES

Arithmetic operators treat missing values as a special case. If either or both of the arguments are missing, the result of the operation is missing. This processing is done on a record-by-record basis. For example, if fields A and B have the following values (with "*" representing missing values), the sum of A and B would be computed as follows:

A	B	A+B
3	4	7
*	4	*
3	*	*
*	*	*

Missing values may occur in database and selection expressions. However, they are not encountered in benefit formulas; records that would otherwise have missing values in a benefit formula are excluded from the valuation run (with appropriate notification to the user).

DATE ARITHMETIC

In ProVal, dates are represented as integers giving the number of days from January 1, 1900 to the date in question. For example, January 2, 1901 is represented by the number 366, and May 14, 1993 is represented by 34,101 (these numbers are never seen by the user, but are described here for pedagogical purposes). In expressions, dates are treated as ordinary integers, and you can perform various types of date transformations using the standard arithmetic operators. For example “Date+7” adds seven days to a date; “Date1-Date2” returns the number of days between two dates, and the expression

#ROUND[(Date1+Date2)/2]

computes the date halfway between two dates.

In an expression, dates can come from the value in a database field, or they can be written literally. Within ProVal, a **date constant** has the form mm/dd/yyyy, as in 12/31/1991.A constant of this form is converted to the integer that represents the date, and the integer is substituted for the date in the expression. For example, the expression

12/31/1991 - Date

computes the number of days from “Date” to December 31, 1991. (The result will be negative for dates after December 31, 1991.) The expression

1/1/1993 - 1/1/1992

computes the number of days between the two dates (which in this case, gives the number of days in 1992).

The system distinguishes between date constants and division of integers by the presence of two slashes within a single “token” (block of characters without a space). For example, 3/4 is interpreted as three divided by four, but 3/4/1992 is a date. If you actually want to divide three-quarters by 1992, use parentheses, as in (3/4)/1992.

Note that if you specify a year using two digits, it is converted automatically to a four-digit year falling in a “sliding window” range such as 1930 to 2029. For example, 1/1/45 becomes 1/1/1945 and 1/1/01 becomes 1/1/2001. The window extends roughly 30 years into the future and 70 years into the past. It remains fixed for 10 years, then jumps ahead a decade. Currently, the window is from 1940 to 2039.To specify a date outside the “sliding window” range, provide the full four digits of the year, as in 1/1/1898.

Two operators provide additional flexibility in comparing dates. The **#YEARDIF** operator subtracts two dates and returns the difference in decimal years. The **#MONTHDIF** operator is similar, but it returns the difference in decimal months. These operators are similar to subtracting dates using minus (-) and dividing by 365 or 30, respectively, however they handle the variable number of days per year and month more precisely. The rounding functions #ROUND, #INT, and #CEILING can be used to convert the results of #YEARDIF and #MONTHDIF to integers. For example, the expression

#INT (1/1/1992 #YEARDIF Date_of_Birth)

computes the age of each person in whole years as of January 1, 1992.

At times you may need to add a number of months or years to a date, and it may not be convenient (or even possible) to express the duration as a number of days. For example, adding exactly ten years to a date requires that you add either 3652 or 3653 days, depending on how many leap years fall in the interval. Exact date-shifting can be performed using the **#DATEPLUS** operator, as in the following example:

Date #DATEPLUS 1y6m3d

This adds 1 year, 6 months, and 3 days to each date in the Date field. The right argument to #DATEPLUS is called a **duration**, and it specifies a time period in years, months, and days. The elements of the duration may be written in any order (e.g., 3d1y6m is the same as 1y6m3d), and elements which are zero may be omitted (e.g., 10y for 10 years). The #DATEPLUS operator adds the specified duration to the dates in the left argument. A similar operator, **#DATEMINUS**, subtracts a duration from dates.

Durations, like dates, are translated to a (fractional) number representing the approximate number of days in the time period. For example, the interval “1y” becomes the number 365.25. You can use durations with ordinary arithmetic operators if you like; for example, you can add or subtract durations or multiply a duration by a constant. However, if you add or subtract a date and a duration using + or -, the result will be the same as adding or subtracting a number of days, and this will not always be the same as the result produced by #DATEPLUS and #DATEMINUS. For example, the expression

Date + 1y

adds 365.25 days to the dates in the Date field. The result, when rounded to the nearest integer, will not be exactly one year later for dates falling in a leap year. Using the #DATEPLUS function instead of + would ensure that all dates are advanced by exactly one year.

RELATIONAL OPERATIONS

Relational operators are used to compare numbers in expressions. For example, the expression

Salary > 30000

compares each record of the Salary field with the number 30,000. The result is expressed using 1s and 0s, with 1 meaning true and 0 meaning false. The result of this expression is 1 in records where Salary is greater than 30,000 and is 0 in records where Salary is 30,000 or less. The result is called a **Boolean** value because it consists of only the values 0 and 1.

Some ProVal commands use Boolean values to select a subset of the database on which to operate. The expression that is used to define the subset is called the **Selection Expression**. Records for which the Selection Expression is true (i.e., returns 1) are selected for use; the remaining records are deselected and not used by the command.

Another use for Boolean values is in arithmetic expressions. The 0s and 1s returned by a relational expression can be used in the same way as any numeric values, allowing you to write a variety of useful expressions. For example, the following expression classifies salaries by range:

(Salary>10000) + (Salary>25000) + (Salary>55000)

The result of this expression is 0 for records where Salary is less than or equal to 10,000; 1 where Salary is between 10,000 and 25,000; 2 where Salary is between 25,000 and 55,000; and 3 where Salary is greater than 55,000.

Multiplication by a Boolean value is another useful technique. For example, setting the field Status using the expression:

[Status * (Status <> 12)] + [99 * (Status = 12)]

changes 12s in Status to 99s.

The following relational operators can be used in ProVal expressions:

= equal	<> not equal
< less than	<= less than or equal
> greater than	>= greater than or equal

These operators can be used with numeric fields, date fields, coded fields, or social security number fields, but not with character fields. (See below for how to compare character fields.) Coded fields are treated as if they were numeric fields, with the numbers being the user-defined internal codes used by ProVal to represent the values of the coded field. These codes are shown in the expression help screen displayed by the F1 function key.

Most relational operators treat **missing values** in the same way that arithmetic operators do: if either or both of the arguments are missing, the result is missing. However, equal (=) and not equal (<>) handle missing values differently. If one of the arguments is the name #MV (which stands for “missing value”), the result has no missing values, and it simply reports where the other argument is missing. Thus, the expression

Sex = #MV

identifies those records where the Sex field is missing. On the other hand, if neither argument is #MV, the equal and not equal operators behave in the same way as other operators, i.e. the result is missing if either or both arguments are missing.

One additional system-defined variable can be used in logical expressions: **#SUBSET**. This Boolean variable has 1s marking records that were selected by the Database | Subset Selection command, which must be executed before using **#SUBSET**.

SEARCHING CHARACTER DATA

The relational operators described above can be used only with numeric, date, and coded fields. To compare character values you must use the **#IN** and **#NOTIN** operators. The **#IN** operator asks if the left argument (a database field) has any of the values listed in the right argument. The right argument is a quoted string (text surrounded by single quotes) or a parenthesized list of quoted strings separated by commas. For example:

LastName #IN 'Smith'

LastName #IN ('Smith', 'Jones', 'Johnson')

The first expression selects values in which LastName is Smith; in other words, the expression returns 1 where LastName is Smith and 0 where it is not. The second expression selects values in which LastName is Smith, Jones, or Johnson. The **#NOTIN** function has similar syntax, but it returns 0 if the value occurs in the list and 1 if it does not. The search conducted by #IN and #NOTIN is not case-sensitive; searching for “Smith” will select values in which the name is spelled Smith, smith, or SMITH.

If the right argument to #IN is empty (just two adjacent single quotes, ''), the result selects values that are blank. This allows you to select **missing values** of a character field. Note that the double quote character (") cannot be used as a substitute for two adjacent single quotes.

If a target string contains a quote, you must double it in the argument to #IN, as in 'That''s the ticket'. (Double-quote characters are discussed below; they actually have to be quadrupled in strings.)

The #IN and #NOTIN operators can be used with numeric and coded fields as well. In this case the argument should be a parenthesized list of numbers separated by commas. For coded fields, you must use the field’s numeric codes (which are shown in the F1 expression help) rather than the character labels for the field. For example:

Status #IN (1,5,7)

This selects records in which Status is either 1, 5, or 7.

PATTERNS

When searching for character values, the items in the right argument to #IN and #NOTIN may be either names or **patterns**. In a pattern, an asterisk (*) stands for zero or more arbitrary characters. For example, the pattern 'A*' matches any value that starts with A. The pattern 'Smi*' matches the names Smith, Smiley, and Smithers, but not Smuckers. A pattern may contain zero, one, or two asterisks. Here are some examples:

*ers	Values ending in “ers”
ch	Values containing “ch”
Jo*s	Values beginning with “Jo” and ending with “s”
J*son*	Values beginning with “J” and containing “son”

Spaces in a pattern are treated in the same way as non-blank characters—for the pattern to match, the value must contain spaces as specified in the pattern. For example, the pattern '*A B*' matches values containing the phrase “A <space> B”.

ADVANCED PATTERN FEATURES

You can also include question mark (?) symbols in a pattern. Each question mark matches exactly one arbitrary character. For example, the pattern '?BC' matches ABC and BBC, but not XYZBC. The pattern 'A??' matches all three-character values that start with A.

To select values that *don’t* match a pattern, put a tilde (~) in front of the pattern. For example, '~A*' selects values that don't begin with A.

The patterns in a list are processed sequentially, from first to last, and each pattern adds to or removes from the previous selection. This is significant for patterns that start with a tilde. For example, the list ('AB*','~??C*') first selects all values that begin with AB and then removes values that have C as the third character. This selects ABDEF, AB, and ABZ, but not ABC or ABCDE. If the first pattern in a list begins with a tilde, a tacit '*' pattern is added to the front of the list so the search starts by selecting everything, then removing items matching the first pattern.

To search for a value containing the symbols *, ?, ~, or " (which are "syntactic characters"), you must surround the symbol with double quotes ("). The quotes can go around the entire pattern or around the syntactic character itself. Here are some examples:

"*ABC"	Values equal to *ABC
"*"ABC	Same as above
"X?Y"*	Values that start with X?Y
""*	Values containing *

If a syntactic character is outside double-quotes, it has its usual syntactic meaning; if it is within double-quotes, it's treated as an ordinary character.

To search for a double-quote character, you must double the double-quote within the surrounding double-quotes, as in:

"D""Q"	Values equal to D"Q
D""""Q	Same as above

DATA OPERATORS

The following additional operators are available when searching data. These may also be used to define new fields by expression.

- a **#LEFT** b: returns a characters from the start of string b.

  For example: 10 **#LEFT** Name will return the first 10 characters of each record's name.

- a **RIGHT** b: returns a characters from the end of string b.

  For example: 8 **#RIGHT** Name will return the last 8 characters of each record's name.

- **#LEN** a: returns the number of characters in string a.

  For example, if your database contains three records, with the name field defined as: Washington, Adams, Jefferson, then **#LEN** Name would return 10,5, and 9, respectively.

- A **#FIND** b: returns the position of string a within string b.

For example, if your name field is written as Last, First and you wish to find where the comma is, use ‘,’ **#FIND** Name.

- a **#CONCAT** b: combines a and b into one character string.

LOGICAL OPERATIONS

By themselves, relational operators allow you to write simple, single-term expressions. To write more complex expressions, logical operators are used to join together individual comparisons. For example, the expression “male employees whose salary is more than $30,000” can be written as:

(Sex=1) #AND (Salary>30000)

In logic, two predicates connected by “and” form a true statement only if both predicates are true. The **#AND** operator returns 1 if both of its arguments are 1. If either or both arguments are 0, the result is 0. Like all operators, it works on a record-by-record basis. The #AND operator can be used to select records in which a numeric field falls within a particular range. The statement

(10000<=Salary) #AND (Salary<=20000)

selects records in which Salary is between $10,000 and $20,000, inclusive.

The other logical connective is **#OR**. Two predicates connected by “or” form a true statement if either or both predicates are true. The #OR operator returns 1 if either or both arguments are 1. The result is 0 only if both arguments are 0. To select employees who have more than 30 years of service or who are 65 or older, you could use the following statement:

(Service>30) #OR (Age>=65)

A third logical operator, **#NOT**, reverses logical values. It converts true to false and vice versa. Numerically, it converts 1 to 0 and 0 to 1. Thus, #NOT(Age<65) is the same as Age>=65.

MANAGING COMPLICATED EXPRESSIONS

As you develop longer and longer expressions, you may find that they become essentially unreadable. There are two ways to avoid this problem: by inserting line breaks and by using temporary variables.

When entering an expression, you can press **Ctrl+Enter** to start a new line. These line breaks do not affect the evaluation of the expression. For example, the following expression (entered as one long line)

(((OldMultBen #max (FrozBen – Offset)) + NewBenPost98) #max NewBenAllSvc) #max MinBen

could be written as:

```
{ [ (OldMultBen  #max (FrozBen - Offset))
          + NewBenPost98
  ]
          #max NewBenAllSvc
 }
          #max MinBen
```

Judicious use of line breaks and indentation can make it much easier to identify parenthesized terms and match up opening and closing parentheses.

Another way of avoiding unreadable long expressions is by using multiple statements and assigning intermediate results to temporary variables. Substatements in an expression are separated by ampersand (&), and assignment is denoted by colon-equals (:=). For example, the statement:

(A+B)/(1+A+B)

could be written as:

S:=A+B & S/(1+S)

The last substatement in an expression is the one that determines the overall result of the expression. The other substatements must define temporary variables.

Using substatements, assignment, and line breaks together offers many opportunities for writing clear expressions. The long expression shown above can be expressed as:

OldWearAway := OldMultBen #max (FrozBen – Offset) &

ExtWearAway := (OldWearAway + NewBenPost98)

#max NewBenAllSvc &

ExtWearAway #max MinBen

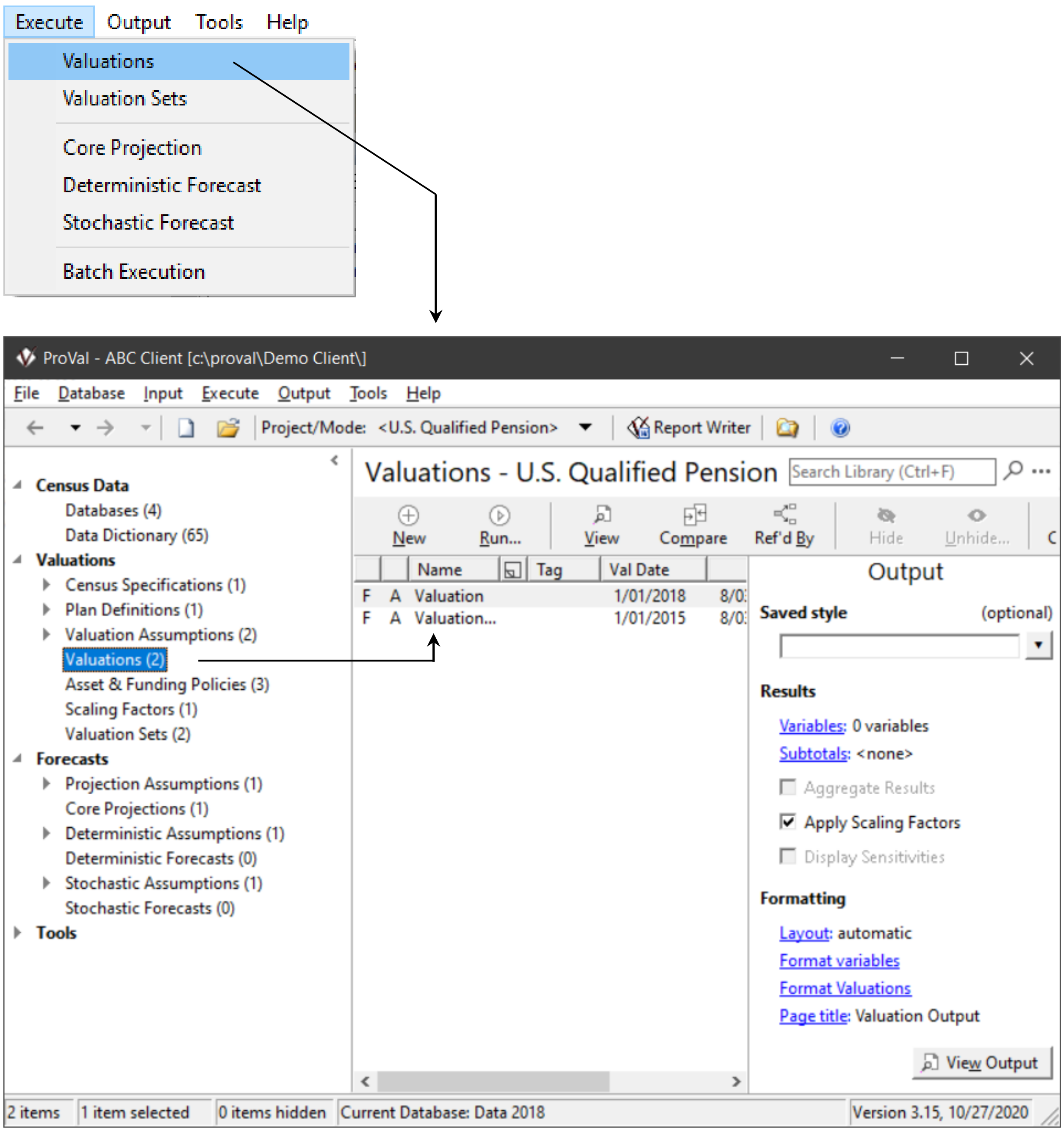
Note that even when line breaks are used, you still need to put an ampersand between statements.

APPENDIX B: LIBRARIES

LIBRARIES

Within each client, the different types of objects you create are stored in different ProVal libraries. For example, one library holds plan definitions, another valuation assumptions, another valuation runs, and so forth. Although the libraries hold very different types of data, certain operations and concepts are common to each of them.

Each library can be accessed through a command on the menu. Libraries found under the **Input**, **Execute** and **Tools** menus can also be accessed through the **Shortcuts** pane. For example, the library shown below can be accessed either through the **Execute | Valuations** command or through the **Shortcuts** pane. The **Entries** pane displays the contents of the selected library. A typical **Entries** pane is shown below.



The Entries pane has a command bar which may contain the following buttons:

- New – Creates a new library entry.
- Edit – Opens the highlighted library entry for editing.
- Rename – Allows you to enter a new name for the highlighted entry.
- Copy – Creates a copy of the highlighted entry which is identical to the original object, except it has a unique default name. Clicking Copy in the above example would create a copy of

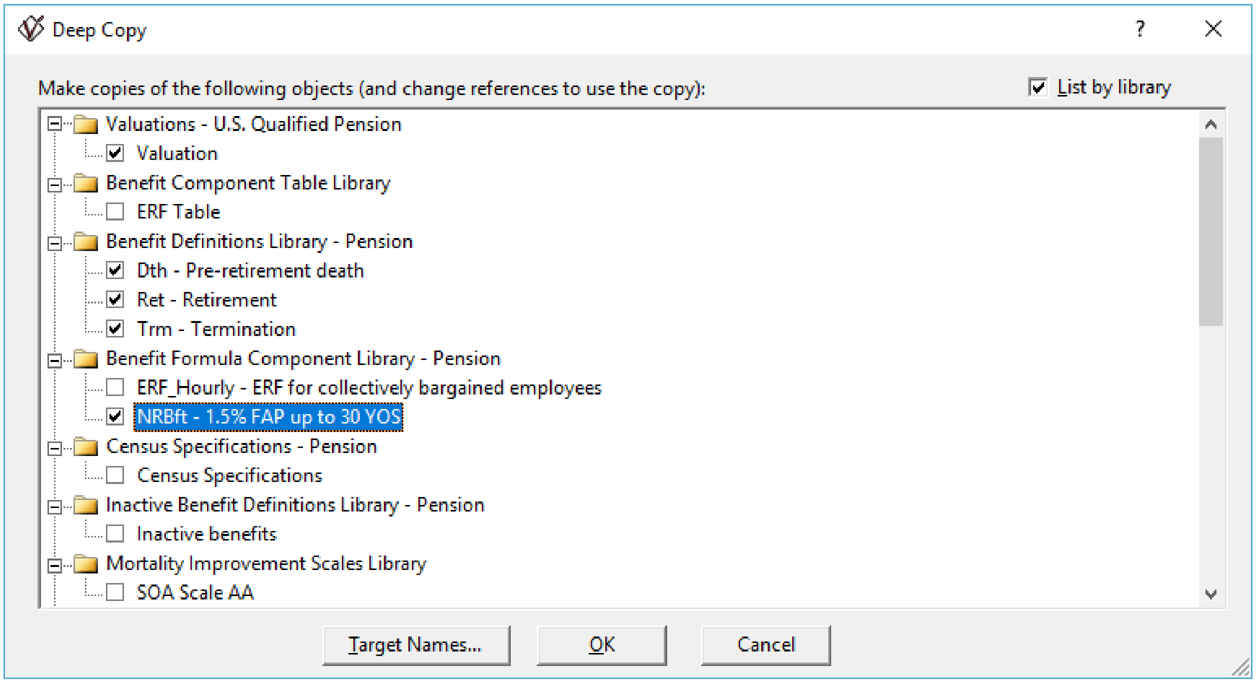
'Valuation,' except the new valuation would be called 'Valuation #2.' Clicking the down arrow directly to the right of the Copy button opens extra copy options:

Copy is identical to the standard copy described previously. Deep Copy allows you to optionally copy additional objects which are referenced by the original object. An example is described in step 4 below.

- Erase – Erases the highlighted entry.
- Hide – Removes the highlighted entry from view in the current project. Projects and their uses are described in detail in Appendix C: Projects.
- Unhide – Allows you to select and unhide an entry which is hidden from view in the current project, but exists in the universe project. The universe project, and projects in general, are discussed in Appendix C: Projects.
- Compare – Displays a text report of the differences between two highlighted objects within this library. This report may be printed or saved to a file.
- View – Displays a text report of the values for the parameters within the highlighted entry. This report may be printed or saved to a file.
- Run – Tells ProVal to perform the calculation parameterized in the highlighted entry. If multiple entries are highlighted, the selected entries are put into the Batch Execute command. Only objects within libraries found under the Execute menu, or have a gear icon on the Shortcuts pane, require the Run command.
- Import – Allows you to select and import objects from another ProVal client into the current ProVal client.
- Ref'd By – Lists any objects which reference the highlighted entry.
- History – Displays the Change History for the highlighted entry.

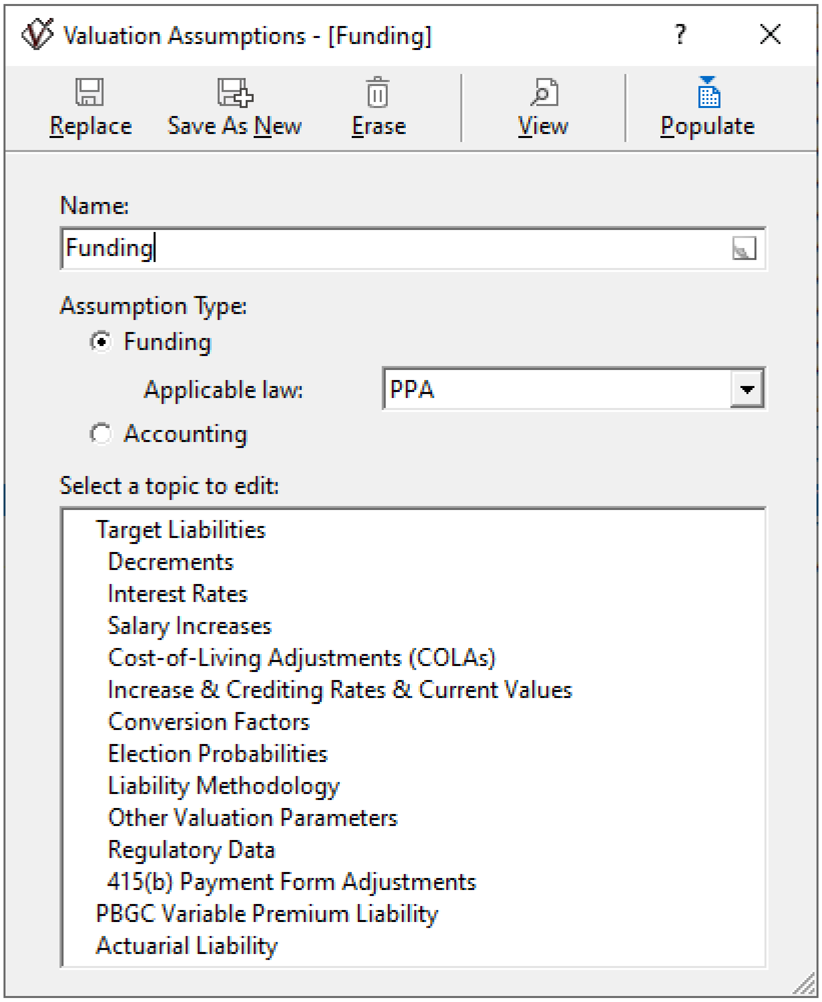
You can create a new library entry in four ways:

1. Click the **New** button found at the top left of the **Entries** pane, or
2. Select an existing entry, click the **Copy** button to make a copy, and **Edit** as necessary, or
3. Select an existing entry, click the **Edit** button, and then exit by clicking the **Save As New** button (see below).
4. Select an existing entry, click the arrow next to the **Copy** button and select **Deep Copy**. This opens a new dialog box which displays all objects referenced within that object from which you can select other objects to copy, specify **Target Names**, and update references:



This allows revision of a Plan Definition while keeping the original Plan Definition intact without manually creating duplicate benefits and components and updating all of the references.

Selecting one of the library entries (or clicking New) opens up a library **editing dialog box**, which displays the contents of the library entry. You can view the contents or modify them as you wish. For example:

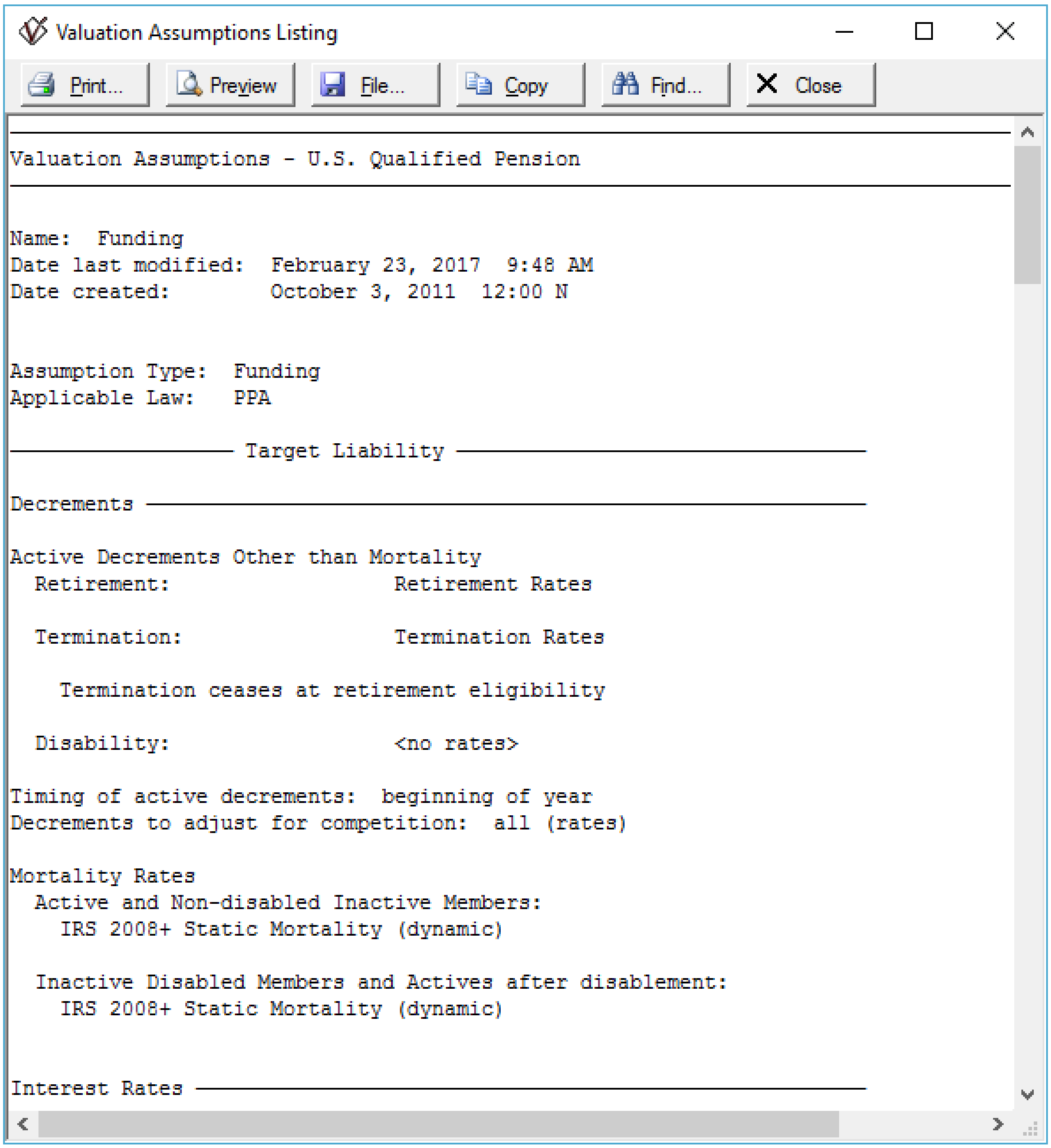


Along the top of the library editing dialog box are the following three buttons which can be used to exit the dialog box:

- **Replace** saves the entry back in the library, retaining any changes that you made and writing over the previous version of the entry.
- **Save As New** makes a copy of the entry you are editing and saves the copy in the library. The original library entry is not altered. You may want to change the name before clicking Save As New so the new entry will have a unique descriptive name. However, if you do not change the name, ProVal will suggest a unique name by adding, “#2” to the end of the original name.
- **Erase** erases the library entry. As described in Appendix C: Projects, this operation is not reversible; ProVal will issue a warning and get confirmation from you before erasing the entry.
- In addition, clicking the **X** on the top right corner exits the dialog box without saving any changes that you made. If you have made changes, ProVal will warn you of this and get confirmation before exiting.

When you exit the library editing dialog box, you are returned to the **Entries** pane where you can select a different library entry for editing.

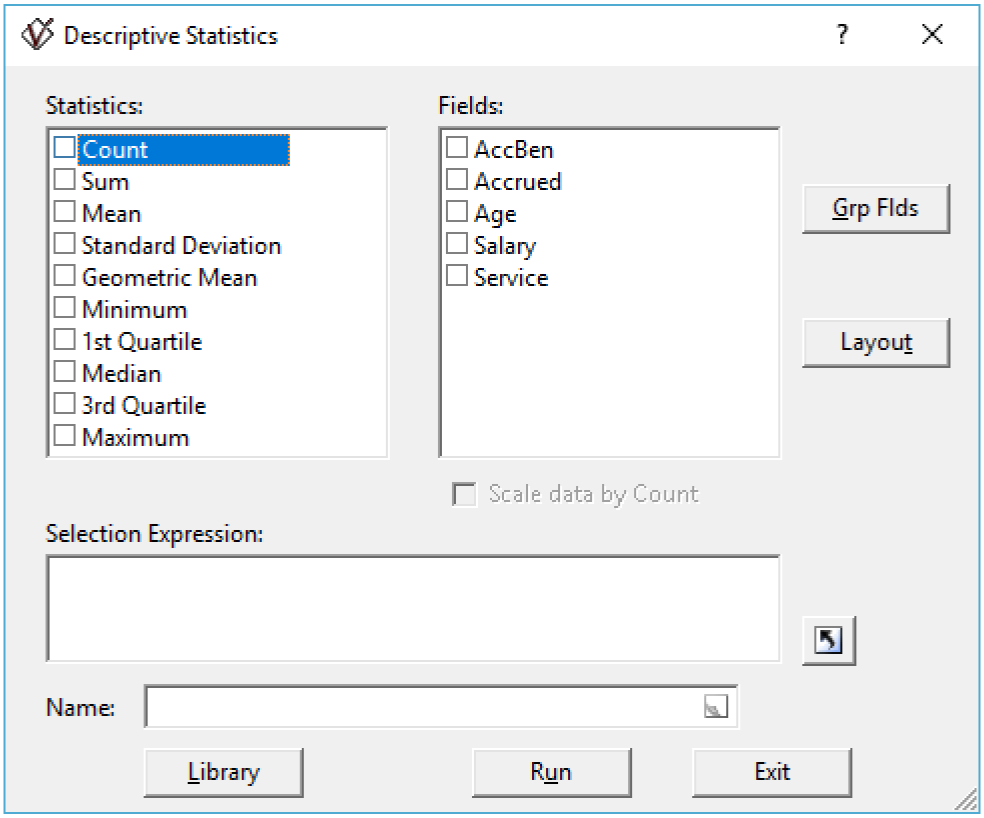
In addition to the cancel buttons, many library editing dialog boxes have a **View** button. Clicking the View button will produce a complete summary of the library entry (see sample below), which can be viewed, printed or saved to a file for reference and documentation.



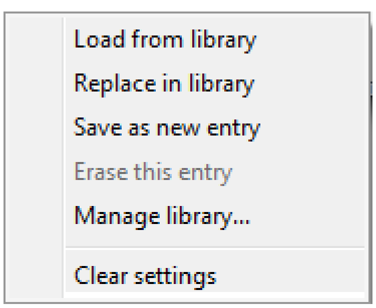
Inverted Libraries

The libraries described above are accessed through an Entries pane followed by a library editing dialog box. For some libraries, the editing dialog box is displayed first and an **entry selection dialog box** is accessed through a "Library" button. Such libraries are referred to as "inverted libraries."

The following is a sample of an inverted library from the **Edit Data | Descriptive Statistics** command (or alternately using **Edit Data | Spreadsheet Edit** under **Home |Descriptive Statistics**).



The **Library** button at the bottom of the library editing dialog box pops up a menu containing the following library operations:



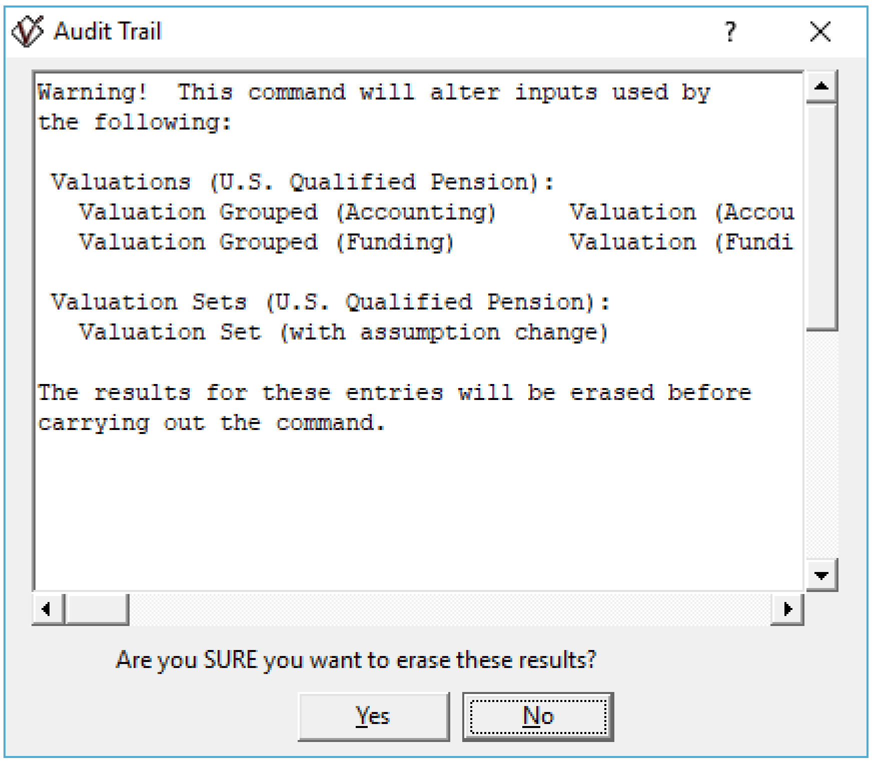
Libraries for the **Output** menu are also examples of inverted libraries; however, they are accessed from the **Output** pane through the back-door button for **Saved style**.

- **Load from library displays** the names of entries in the library and lets you select one to be copied into the library editing dialog box. You can revise it if you wish, but doing so will not affect the saved entry unless you explicitly replace it in the library.
- **Replace in library replaces** the current entry in the library.
- **Save as new entry** saves **the** current entry as a new entry in the library. Before using this operation, you must enter a name for the entry in the appropriate field of the library editing dialog box.
- **Erase this entry erases** the current entry from the library. As described in Appendix C: Projects, this operation is irreversible; ProVal will issue a warning and get confirmation from you before erasing the entry.
- **Manage Library… brings** up a dialog box which allows you to perform library operations on multiple library entries at once, including, **Copy**, **Erase**, **Hide**, **Unhide**. It also allows you to **Import** entries from another client.
- **Clear settings** removes all **selections**, including those made subsequent to loading an existing library entry. Note that when "Clear settings" is applied to a loaded library entry, the saved entry itself remains intact and can be reloaded.

AUDIT TRAIL

When you run a valuation or projection, ProVal makes note of the inputs that were used to generate the results, and it protects these inputs against later change. This is done so you can determine, with absolute certainty, what inputs were used to produce a given set of output. This protection of inputs is referred to as the **audit trail**. The audit trail protects all library inputs as well as any database fields that were used to compute the output.

In practice, you are not actually prevented from making changes to the inputs. Instead, when you attempt to alter an input, ProVal displays a warning message listing the outputs that will be lost if the associated input is changed and asks if you are sure that you want to modify the input.



If you answer:

- "**Yes**" to change the input, **ProVal** will erase all of the affected output. To produce the revised results, simply go back to the valuation command, for example, and rerun the valuations.
- "**No**" to retain the output, **you** may then choose to save the modified input item as a new entry in the library with a revised name.

OTHER CONSISTENCY CHECKS

Some library entries in ProVal refer to entries in other libraries. For example, a Plan Definition refers to Benefit Definitions; Benefit definitions include a formula that refers to Benefit Formula Components; and Benefit Formula Components may refer to Accrual Basis Components. Entries that are referenced by a higher-level entry (such as a Benefit Definition that is referenced by a Plan Definition) are protected against erasure. Before you can erase a referenced low-level entry, you must first remove the reference by the higher-level entry. There is, however, no protection against modification to lower-level entries that are not currently referenced in output; you can modify an object whether or not it is referenced.

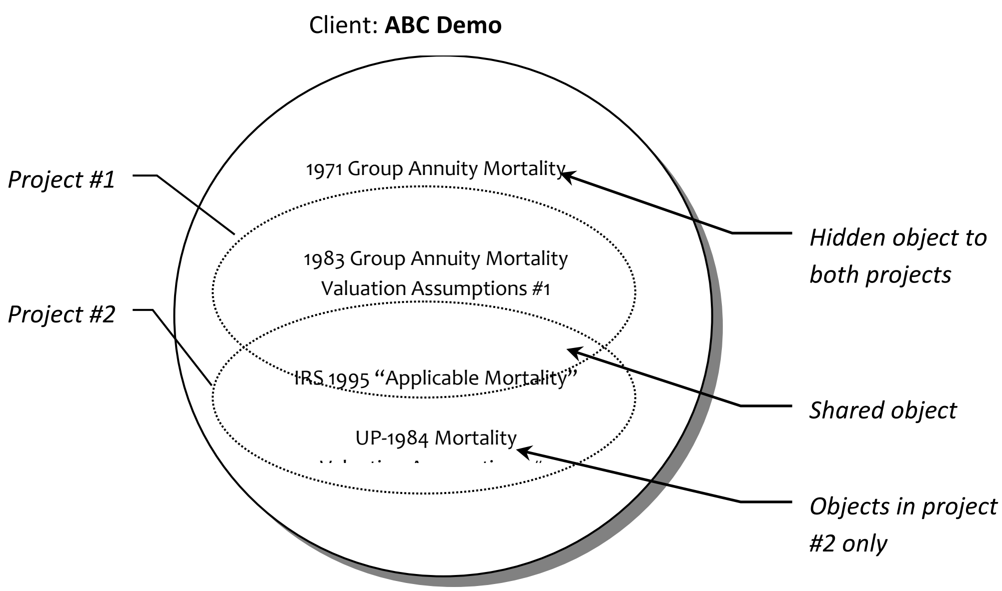
When you replace a library entry that is visible in more than one project, ProVal displays a message warning you that other projects will be affected. If you don't wish to alter the entry in the other projects, you can save the modified object as a new library entry. However, if you do so, beware that no higher-level objects will refer to the new object. If there are supposed to be references to it, you will have to manually change the high-level objects to reference the new entry. For example, if you change a benefit definition and save it as a new benefit, you will have to revise plans to include the new benefit.

APPENDIX C: PROJECTS

After working in ProVal for a while, you may find that your libraries are getting uncomfortably large. A large library means that you have to select from a long list to specify an item in the library. For some users, only a portion of the library may be relevant to any given task. For example, some library entries for the annual valuation may not be needed when working on a forecast for the same client. Similarly, library entries for the Salaried Plan may not be needed when working with the Hourly Plan. ProVal provides a way of hiding library entries so you can see just the entries that are relevant to your current task. The hiding is implemented by means of projects.

PROJECTS

You can define any number of projects within ProVal, and you can select a project to use as the **current project**. Within a project, you decide which library entries should be visible and which should be hidden. When you create a new library entry, it is automatically made visible in the current project and hidden in other projects. You can **unhide** an entry from another project in the current project, thereby making the entry visible in both projects. When you **erase** an entry, it is permanently erased from the library. There are two safeguards to this type of erasure. First, ProVal refuses to erase an object that is referenced by another library, and second, it warns you and gets a confirmation before erasing the object.



An object that is visible in more than one project is **shared** by those projects. Only one copy of the object is stored, however, and *if that copy is changed from within one project, it is changed in the other projects as well.* This behavior is especially useful for reference tables. A single copy of a table can be shared by all projects, and when a correction or update is needed, the change can be entered just once. However, for other types of objects this sharing may not be desirable. The **Save As New** button can be used to make an independent copy of an object so that changes in one project will not affect other projects. Whenever you replace a shared object in a library, ProVal warns you that the operation will affect projects other than the current project. You can then decide whether or not you really want a shared object.

MANAGING PROJECTS

You can use the **File | Change Project / Mode** command or Project / Mode dropdown on the tool bar to open, edit, rename or copy existing projects, create a new project, or erase projects. The name of the current project is displayed in the toolbar at the top of the screen

- **Open** facilitates switching between projects or modes. This operation opens the highlighted project/mode as the current project. Only objects that are unhidden for the opened (ie: current) project will be displayed for each library.
- **New** brings up a **dialog** box to create a new project. You must provide a Name and mode type (U.S. Qualified, OPEB, U.S. Public, Canadian Registered, Universal, German, U.K. Pension). Optionally, you may check the box to Unhide all objects. This option will display all library objects that can be utilized in the selected mode. Otherwise, all objects will be initially hidden in the new project, meaning all libraries will appear to be empty. As you edit the libraries, you can unhide objects as needed from other projects.
- **Edit** displays the library **entry** for the highlighted project. This operation allows you to edit the entry, create a new entry, or erase the entry. Edits can be made to the entry name, mode, or to unhide all objects. Replace will save the edits to the existing project. Save as new will create a new entry based on the edits. Erase will erase the project (however, no objects associated with that project will be erased). Universe projects cannot be edited.
- **Rename** allows you to edit the name of the highlighted project without entering the Edit dialog box. Universe projects cannot be renamed.
- **Copy** makes a copy **of** the highlighted project(s). A copy of a project will have the same library objects hidden/unhidden as the project from which it was copied. This operation produces the same result as highlighting a project, clicking Edit, making no changes and clicking Save as new. Highlighting multiple projects creates a single copy of each highlighted project. Universe projects cannot be copied.
- **Erase erases** the highlighted project(s). This operation produces the same result as highlighting a project, clicking Edit and clicking Erase. Multiple projects can be erased at once by highlighting multiple projects. Universe projects cannot be erased. Erasing a project or projects will not erase any objects which are associated with the erased project(s). This prevents other projects from being affected when a project is erased.

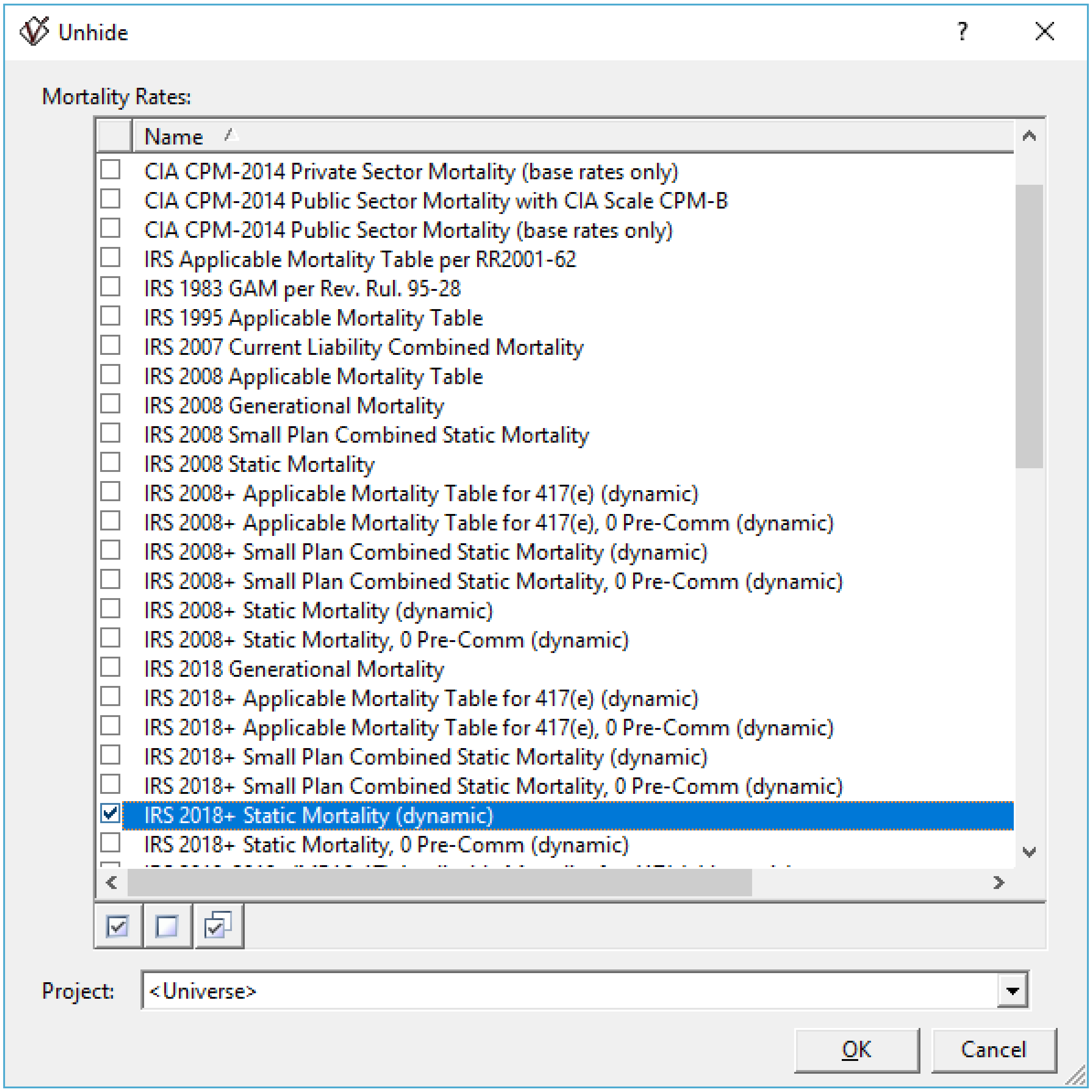
Note: After creating a new project, if you did not select to unhide all objects, the first thing you should do is use the **Data Dictionary** command to unhide database field definitions in the new project. The method for doing this is described below.

UNIVERSE PROJECT

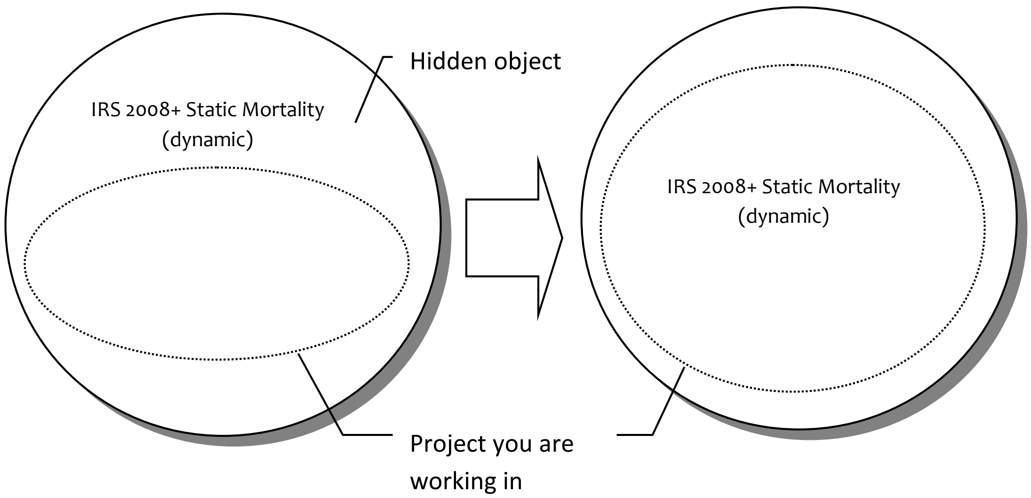
It is possible for an object to be visible in no projects (that is, be hidden in every project). This happens, for example, when an object belonging to a single project is hidden. The **Universe** project allows you to access such objects. The Universe project contains every object stored in the libraries, regardless of project membership. When you **hide** an object from a project, the object is still available in the Universe project; this allows you to "unhide" the object from the Universe project as described below. Also, you can select Universe as the current project using the File | Change Project / Mode command. This causes all objects to be visible when you edit a library.

UNHIDING OBJECTS

In most libraries, the entry selection dialog box has an **Unhide…** button at the bottom of the dialog box. When you click Unhide, ProVal displays an unhide dialog box, such as the one illustrated below from the Input | Reference Tables | Mortality Rates command.



You may select one or more objects, or you can click the All button to select all objects. When you click the OK button, the selected objects are unhidden in the library you were editing, and you are returned to the entry selection dialog box in which you started.



By default, all objects that can be unhidden are displayed (i.e. those in the Universe but not in the current project). You can shorten the list by choosing another project to unhide from. The choices include every project you have created plus the Universe and **Orphanage** projects. The **Orphanage** project contains objects that are not visible in any project other than the Universe. The Orphanage can't be selected as the current project, but it appears as a choice when you are unhiding objects.

To make an independent copy of an object (not shared with another project), first unhide the object as directed above and then edit the object. Enter a new descriptive name for the object and click the **Save As New** button. Then, once again edit the original object and click the **Hide** button to remove the shared copy from the project.

Note: It is better to make an independent copy of an object as soon as you unhide it from another project instead of waiting until you revise the object and then using Save As New to make the independent copy. If you wait until later and in the meantime create other library objects that refer to the original (shared) entry, you will have to change those references when you create the new, independent copy.

In an inverted library that starts with a library editing dialog box rather than an entry selection dialog box, an object is unhidden by clicking the **Library** button, selecting **Load from library** in the popup menu, and then clicking the **Unhide** button in the entry selection dialog box. From there, you proceed as directed above.

OBJECT DESCRIPTIONS

Objects in the libraries are identified by their descriptive names. Within a given library, each object must have a unique name. This applies even across projects: two objects may not have the same name even if the objects reside in different projects. When you save a new object, you must provide a name that does not yet exist in the library. If you attempt to save an object with the same name as another object in the library, ProVal displays a **Duplicate Name** warning and proposes a unique description. The proposal consists of the description you supplied followed by a numeric suffix (for example, “Disability benefit #2”). You can either accept the proposed description or change it to something else.

The formula components used in benefit formulas and accrual basis expressions are handled slightly differently. Each component has both a name (which is used in the formulas) and a description. The names must be unique, but the descriptions need not be unique. If you provide a name that is already in use, ProVal does not propose a unique name, but instead shows you the list of names already in use and asks you to provide a new name.

APPENDIX D: SHORTCUTS

This chapter describes some keyboard shortcuts to operate ProVal. Although ProVal operates in an intuitive way that is similar to other Windows-based software packages, ProVal has a number of specialized features. For this reason, even experienced software users will benefit from reading this chapter.

MNEMONICS

Mnemonics are the underlined letters in menus and buttons. For menus, entering this **letter** executes the menu command. For buttons, pressing **Alt+letter** pushes the button.

DIALOG BOX BASICS

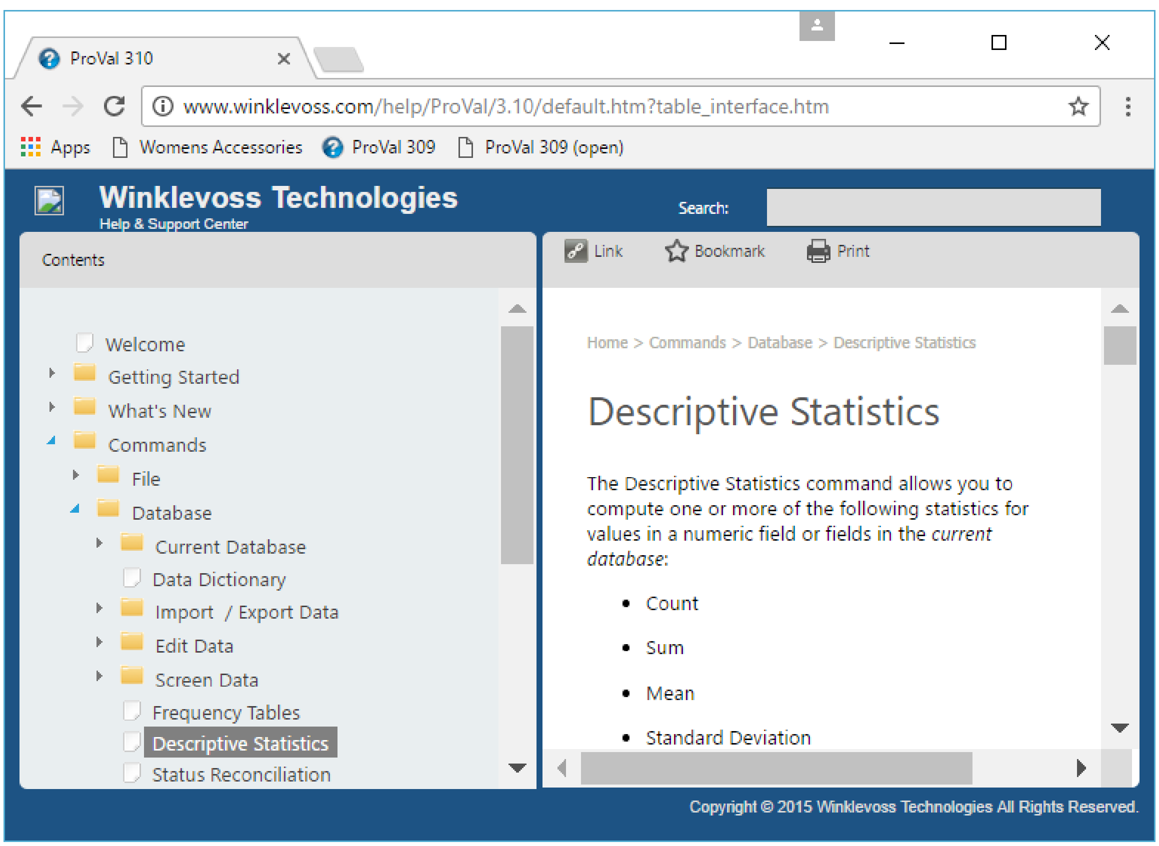
In any dialog box, you can use the **Tab** key to move your cursor from field to field. To go backwards, use the **Shift+Tab** key combination.

Clicking the button on the top of any dialog box has the same effect as clicking the Cancel or Exit button with the mouse or hitting the **Esc** key.

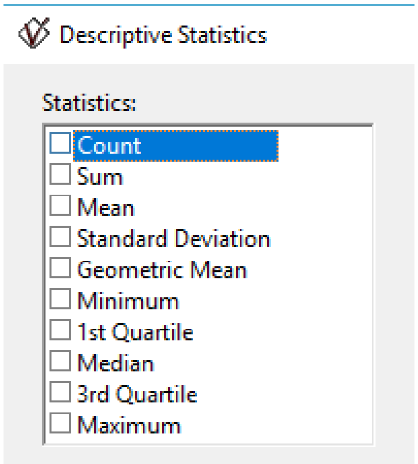
Clicking the button on the top of any dialog box and then clicking on a field brings up context-sensitive help to answer questions such as “What is this?” and “Why would I use it?” Alternatively, you can use the **Shift+F1** key combination.

> Allows you to specify the order of dimensions in the output
> More...

For more help click the More… link in context-sensitive help or press **F1.**

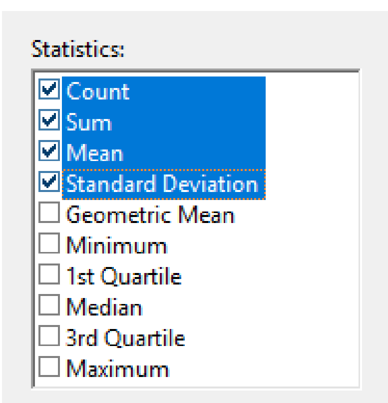


LIST BOXES



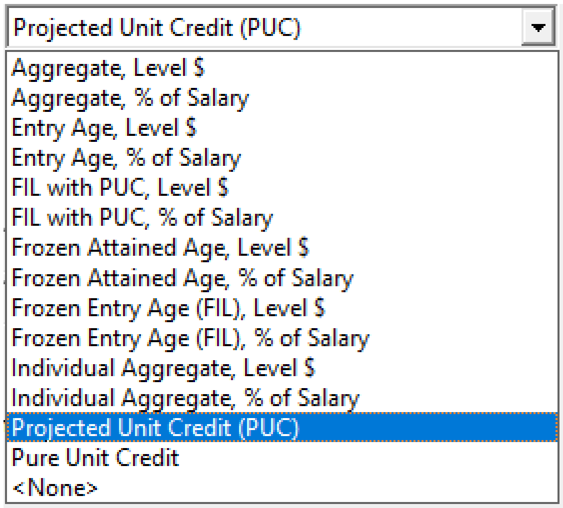
To select an item in a list box, press **Enter** or the **Spacebar** and a checkmark will appear. List boxes come in two varieties: **single selection** and **multiple selection.** Visually, you can recognize multiple selection lists by the checkbox next to each item (as shown in the figure above).

At times, you may want to select all items in a multiple selection list. To do this, simply press **Ctrl+A** (some dialog boxes have an “All” button that allows you to do this with the mouse). To clear all selections, press **Ctrl+N.** You can also select multiple items by clicking the first item and then holding down the shift key while clicking the last item.



To move the cursor using the keyboard, press the **Up** and **Down** arrows. Alternatively, press a letter key to move the cursor to the next item beginning with that letter (you would press the letter M to move to “Mean” in the example above). If several items in the list start with the same letter (“Mean”, “Minimum”, “Median”, and “Maximum” in the example above), you can either type the first few letters (without a pause) or type the first letter several times to reach the desired item.

DROP-DOWN LIST BOXES

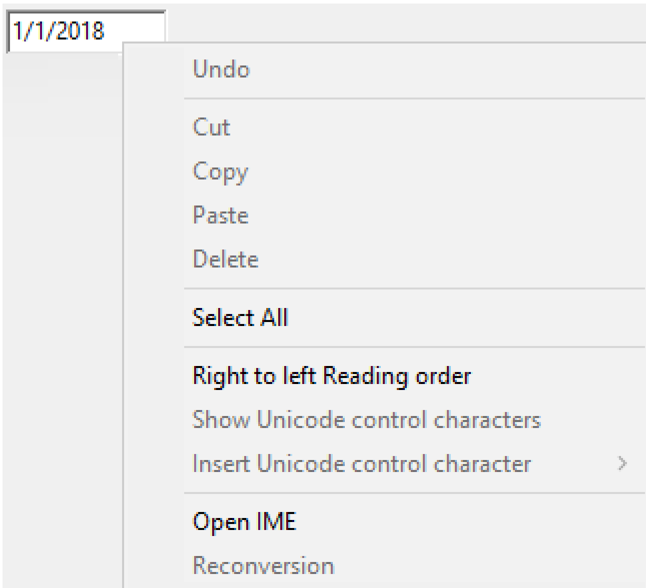


Like a single selection list box, a drop down list box allows you to choose a single item from a list. The difference is that the list is displayed upon demand.

When the list is closed, you can cycle through the choices using the **Up** and **Down** arrows. To display the list, press **F4**, **Alt+Down arrow**, or the **Spacebar**. Then, select an item by pressing **Enter**.

Alternatively, you can press a letter key to select the next item beginning with that letter (you would press the letter P to select "Pure Unit Credit" in the example above). If several items in the list start with the same letter, you can type the first few letters of the entry.

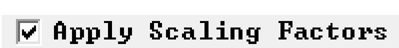
NUMBER, DATE, AND TEXT FIELDS



Several features are available when entering numbers, dates, and text. These are available by right clicking with the mouse or using the following keystrokes:

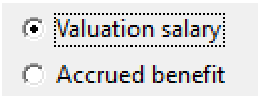
Ctrl+X	cut
Ctrl+C	copy
Ctrl+V	paste
Ctrl+Z or **Alt+Backspace**	undo

CHECK BOXES



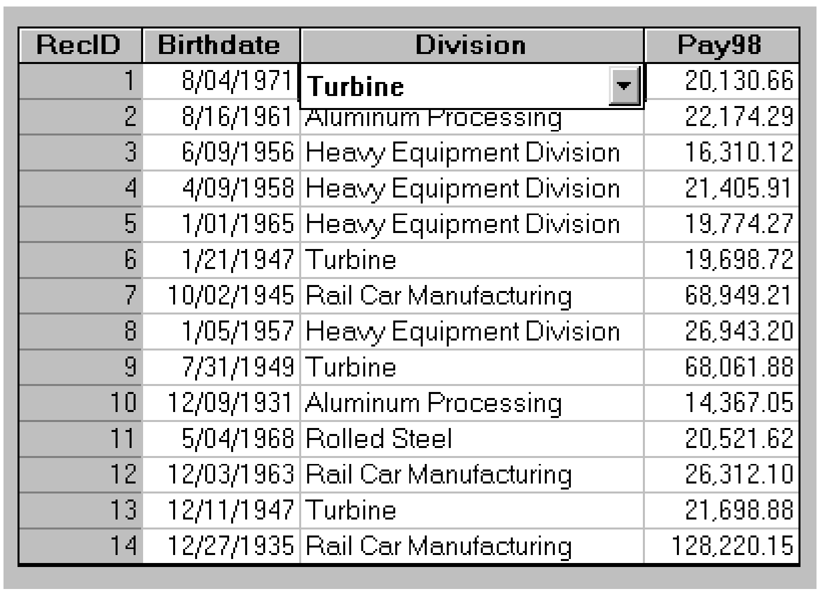
To toggle a check box with the keyboard, use the **Spacebar.**

RADIO BUTTONS



To change the setting of a radio button with the keyboard, use the **Spacebar** or the **arrow keys.**

SPREADSHEET FIELDS



RecID	Birthdate	Division	Pay98
1	8/04/1971	Turbine	20,130.66
2	8/16/1961	Aluminum Processing	22,174.29
3	6/09/1956	Heavy Equipment Division	16,310.12
4	4/09/1958	Heavy Equipment Division	21,405.91
5	1/01/1965	Heavy Equipment Division	19,774.27
6	1/21/1947	Turbine	19,698.72
7	10/02/1945	Rail Car Manufacturing	68,949.21
8	1/05/1957	Heavy Equipment Division	26,943.20
9	7/31/1949	Turbine	68,061.88
10	12/09/1931	Aluminum Processing	14,367.05
11	5/04/1968	Rolled Steel	20,521.62
12	12/03/1963	Rail Car Manufacturing	26,312.10
13	12/11/1947	Turbine	21,698.88
14	12/27/1935	Rail Car Manufacturing	128,220.15

A spreadsheet field is a grid of values. They are usually numbers, but may also include dates, characters, and drop-down lists. **Warning:** the shortcuts for drop-down list boxes cannot be used when they appear in spreadsheets!

To edit the value in a cell, press **F2** (or double click with the mouse). Alternatively, you can simply begin typing the replacement value. When you finish entering or revising the value, press **Enter** or **Tab**. The cursor will automatically advance to the next cell down (to the right, if you pressed Tab). Alternatively, you can press an **arrow key** to move in a specific direction.

In addition, several other features are available by right clicking with the mouse when you see the cursor or by using the following keystrokes:

Ctrl+Ins insert a row above the cursor (when available)

Alt+Ins insert a row below the cursor (when available)

Ctrl+Del delete the row the cursor is on (when available)

Ctrl+D duplicates the current cell down

Ctrl+C copy

Ctrl+V paste

Some of the columns of a spreadsheet may contain computed data or data that is displayed for reference purposes only. These columns cannot be changed, and they are displayed in gray.

FROM-TO TABLES

Several ProVal commands allow you to enter a quantity, such as an interest rate, that varies with time. These quantities are generally entered using what is called a **From-To Table**. An example is shown below.

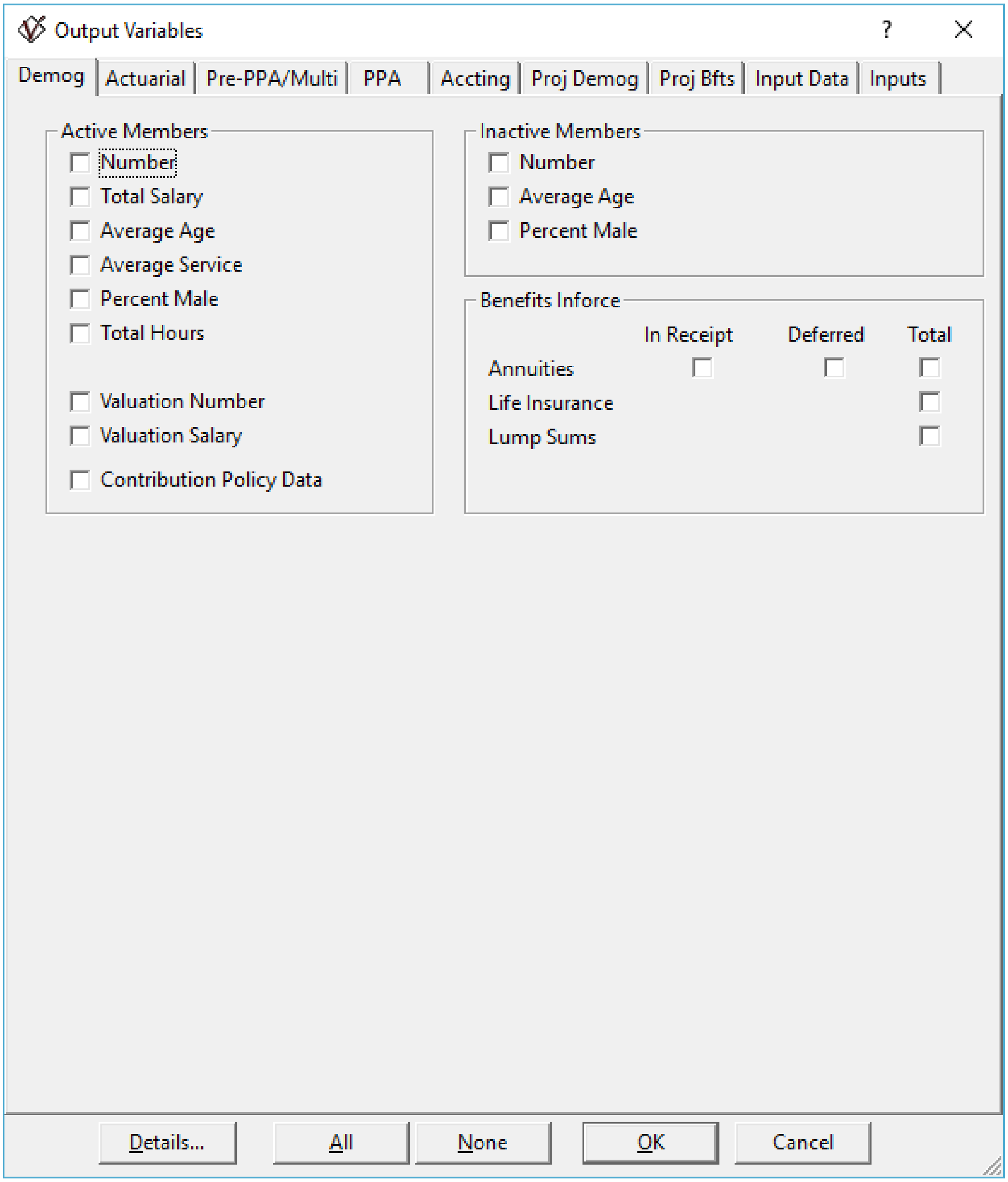
From	To	Rate
-	1998	0.085
1999	1999	0.080
2000	2004	0.070
2005	2009	0.075
2010	-	0.072

A from-to table is really just a special kind of spreadsheet. All the shortcuts that apply to spreadsheets also apply to from-to tables.

However, entering data in a from-to table is a little different. You enter data only in the first and last columns of the table. The middle “To” column is computed automatically based on your entries in the “From” column. In addition, the hyphens (-) in the first and last rows cannot be changed. The first hyphen (in the From column) represents the beginning of time. The second hyphen (in the To column) represents the end of time.

TABS

To move through a series of tabs, as in the example below:



Ctrl+Tab or **Ctrl+Page Down** move forward through tabs

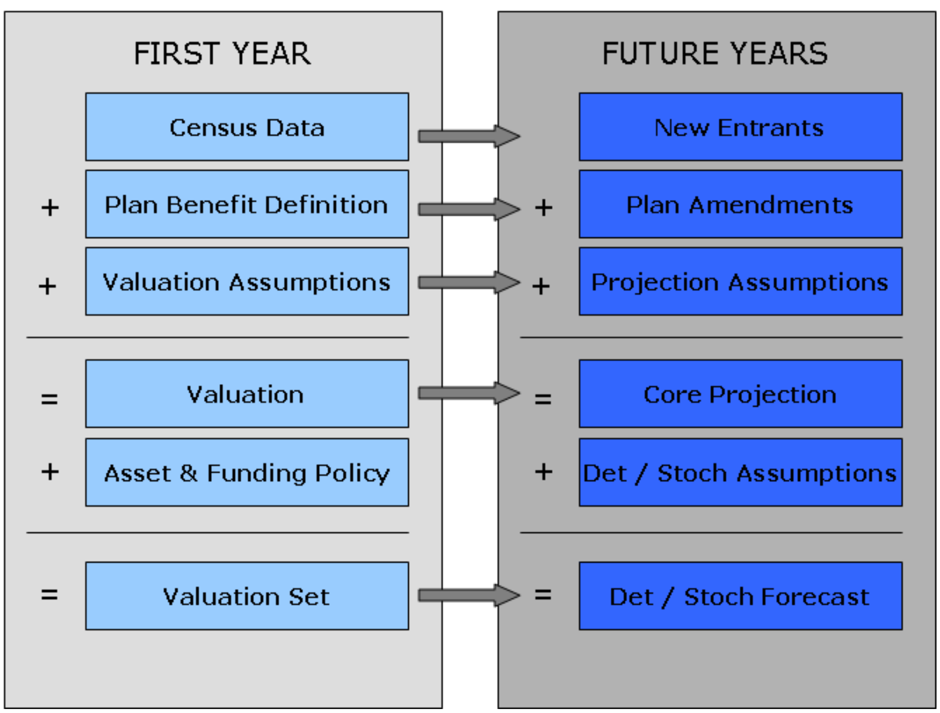
Ctrl+Shift+Tab or **Ctrl+Page Up** move backward through tabs

APPENDIX E: FORECASTS

Valuations are required to compute liabilities and costs for the current period, and the results are used to determine a sponsor's balance sheet and income statement entries, to determine cash contributions, and to calculate a host of other snapshot metrics used for keeping the plans in compliance.

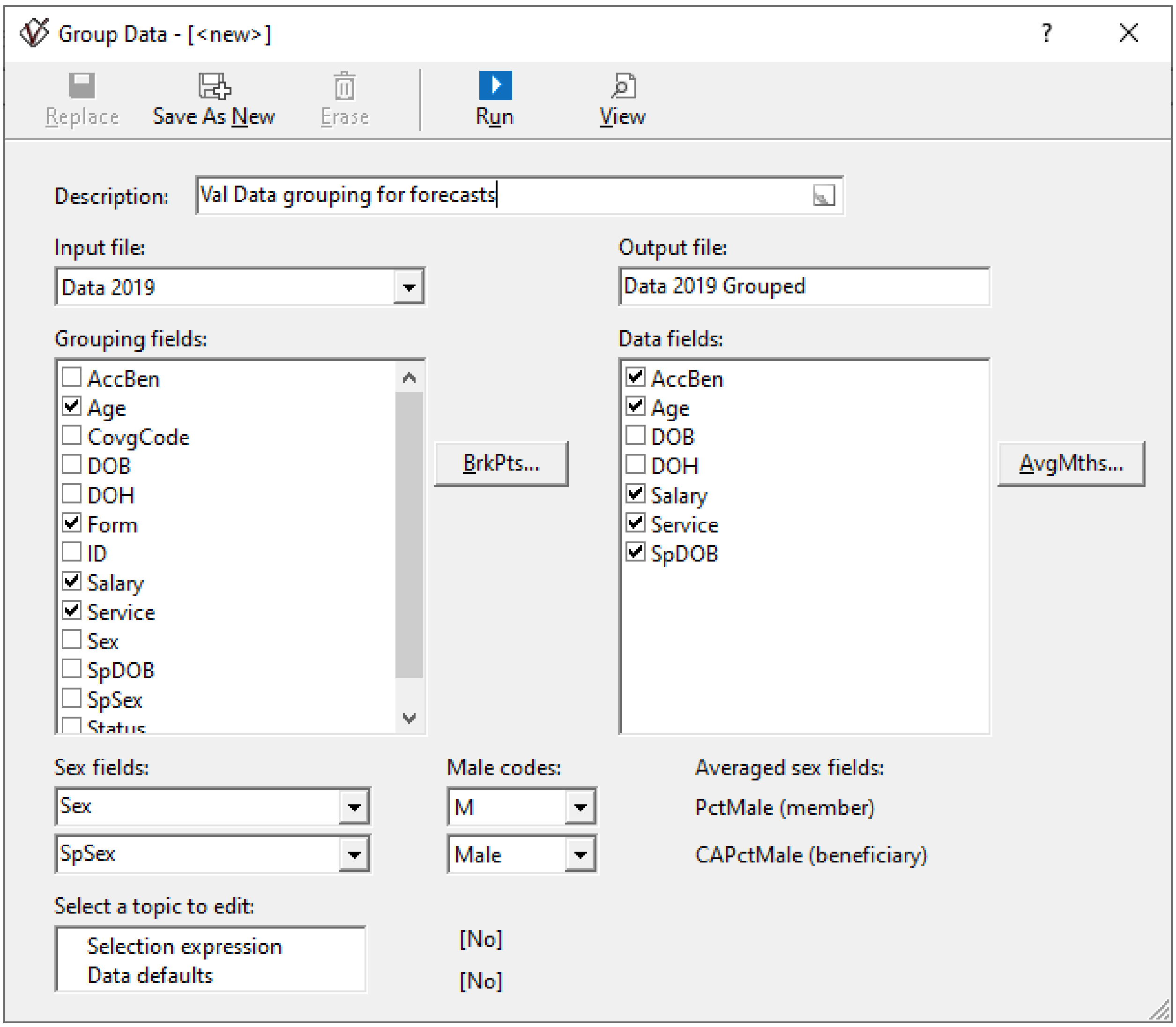
Forecasts are used by plan sponsors and the professionals that advise them to quantify potential valuation results for future periods. A deterministic forecast is a projection of valuation results assuming a given set of assumptions will be exactly realized. A stochastic forecast generates a distribution of projected valuation results, based on a quasi-random range of assumptions (a stochastic forecast with 2,000 trials is equivalent to 2,000 deterministic forecasts). Thus as an example, a deterministic forecast of pension expense for the year 2010 is a point estimate (e.g., $50,000,000) whereas the stochastic forecast for the same result is a distribution (e.g., 2,000 values that have a median value of $48,000,000 and a standard deviation of $10,000,000).

ProVal allows you to build on the valuation work and perform deterministic and stochastic forecasts. This chapter provides a conceptual overview and step-by-step instructions for this process. Here is an overall schematic:



GROUPING DATA

Since the Core Projections that are required for forecasting are extremely calculation intensive, it is usually necessary to use grouped (as opposed to seriatim) data for this purpose. Note the grouped data will require its own census specifications as well as its own scaling factors, both discussed below.



Using the **Database | Group Data** command, specify the:

- **Input file (**seriatim database) and **Output file** (grouped database)
- **Grouping fields** and **BrkPts.** A typical starting point might be:

Grouping field	Breakpoints	Comment
Age	From 20.5 to 100.5 step 1	If possible, group so that there are participants at every age. This keeps the data “smooth” and prevents “clumping” in a forecast. Alternatively, use date of birth with breakpoints spanning 100.5 to 20.5 years prior to the valuation date, e.g., “from 7/1/1914 to 7/1/1994 step 1y”.
Service	5 15 30	Generally, fewer service breaks are required. Keep in mind such things as a service cap for benefit accruals or accrual rates that vary by service. Alternatively, use date of hire with breakpoints equal to the valuation date minus 30, 15, and 5 years, e.g., “1/1/1985 1/1/2000 1/1/2010”.
Salary	200,000	Generally only 1 or 2 salary breaks are required, such as for the highly compensated and around the plan’s integration level.
Status	n/a for coded fields	
Form	n/a for coded fields	For inactives in pension modes

Be sure to include any coded fields used in the valuation, e.g., Division. Note selecting a coded field will create a separate bucket for each value; however, any coded fields not selected as a grouping field will not appear on the grouped database.

- Do not group on sex. Rather, have ProVal calculate the PctMale and CAPctMale **Sex Fields** as shown.
- Select all **Data fields.** These will be saved to the grouped database.

- Click the **Data defaults** topic to apply data defaults to the seriatim data before grouping it. If defaults were used in the valuation you're trying to replicate with grouped data, apply defaults here for best results.

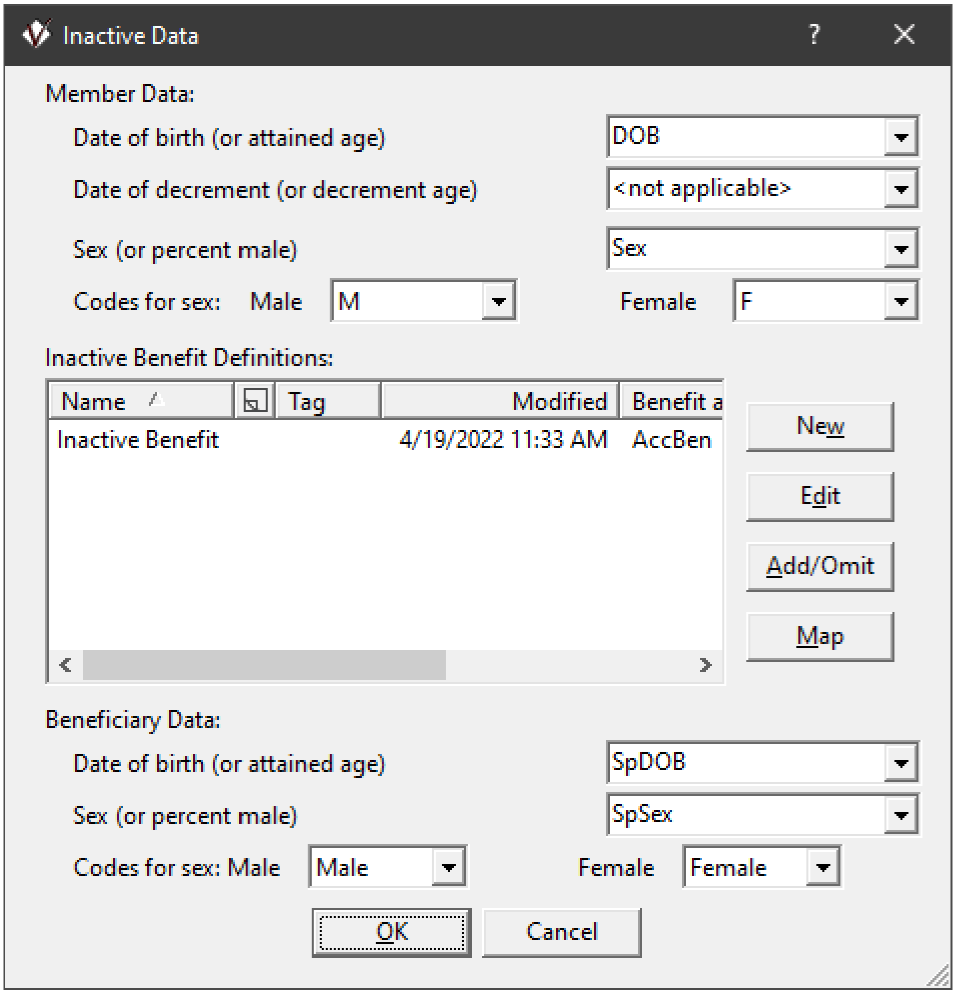
- **Run** the data grouping.

Save the Group Data entry for easy revision should your grouping need to be fine-tuned.

Think ahead: If you will be modeling an open group, you will also need a separate grouped database containing a profile for new entrants. For information on creating a new entrant database, see New Entrants below.

CENSUS SPECIFICATIONS

Next, go to the **Census Specifications** library from either the **Input** menu or the **Shortcuts** pane, **Copy** the existing Census Specifications associated with the seriatim valuation and modify the copy for the grouped data by changing the sex references for both Active Data (to PctMale) and Inactive Data (to PctMale for Members and to CAPctMale for beneficiaries).



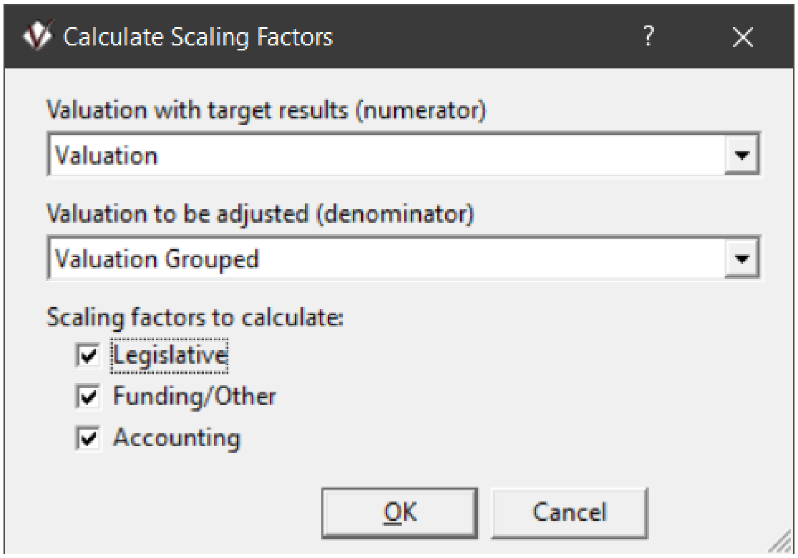
Scaling Factors

Finally, create a set of Scaling Factors which will correct for any small differences that arise when valuing the plan using grouped data.

Under **Valuations** (from the **Execute** menu or the **Shortcuts** pane)**, Copy** your existing Valuation and modify the copy by updating the Census Data Database File and Census Specifications to reflect the new grouped data. Run the grouped valuation and **Replace**.

Create a new library entry under **Scaling Factors** (from the **Input** menu or **Shortcuts** pane) to adjust your grouped data results to match your seriatim valuation results.

Use the **Calculate** utility (located on the top right) to derive the scaling factors. Specify the seriatim valuation as the target results (numerator) and the grouped valuation as the results to be adjusted (denominator).



Examine the scaling factors for reasonableness. Generally they should be within 1.5% for all liabilities.

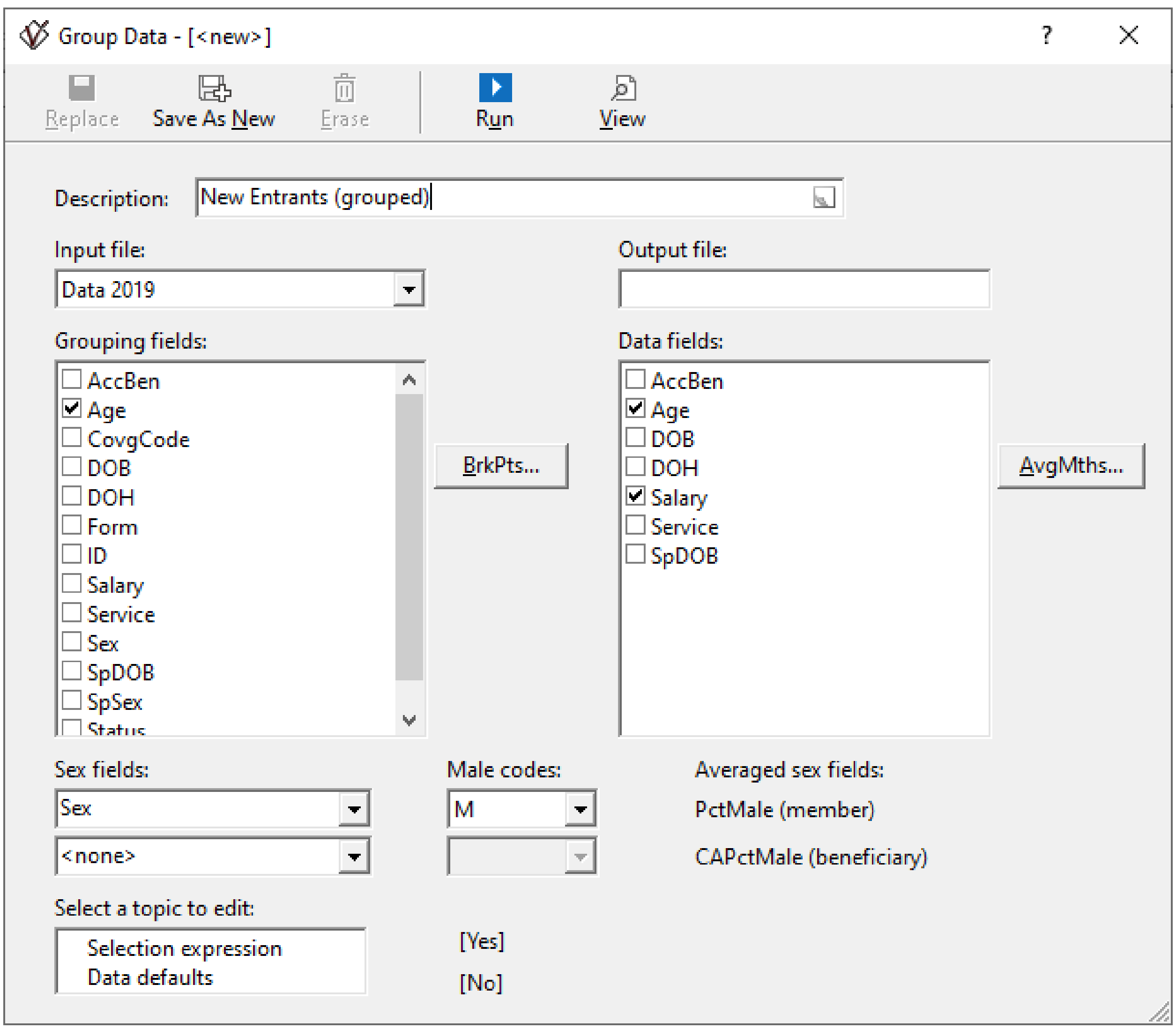
If your scaling factors seem too high, adjust your data grouping parameters under **Database | Group Data** to create a finer grouping and re-calculate scaling factors until the results are acceptable.

Make sure that the scaling factor for **Other Scaling Factors** is similar to (generally within 1.5% of) the scaling factors for active liabilities. Discrepancies will generate spurious gains/losses during the forecast as liabilities for inactives are materially different than expected when they were actives. If you are computing scaling factors by hand (e.g., to match another actuary's valuation results in the first year of a forecast), then simply set this scaling factor equal to the scaling factor used on the active liability of primary interest. It is rare, but possible, for discrepancies to occur when calculating scaling factors as the ratio of two ProVal valuations. If this is the case, you should refine the data grouping (e.g., by adding breakpoints at older ages for actives eligible to receive immediate benefits).

NEW ENTRANTS

Whereas valuations are almost always performed on a closed group basis, forecasts usually reflect the hiring of new entrants. (One notable exception would be a plan closed to new entrants.) New entrants can be introduced simply to replace those who decrement (i.e., a level active population = zero overall population growth) or they can be introduced at a rate to achieve an overall population growth rate specified in the Projection Assumptions (e.g., 5% for next three years, then level thereafter).

New entrants are stored in a database file that is separate from the valuation file. You may create this file "by hand", but it is typically created using the **Database | Group Data** command.



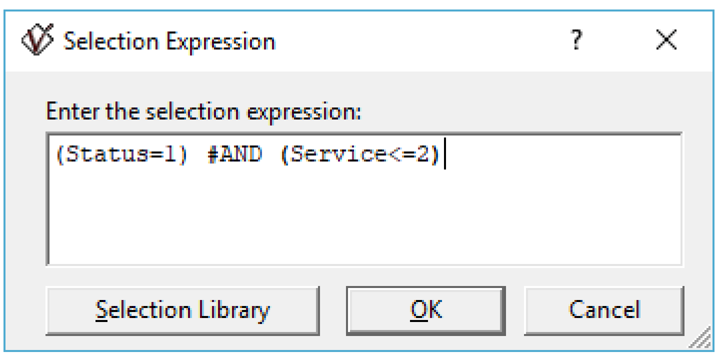
Using the **Database | Group Data** command, specify the:

- **Input file (**seriatim database) and **Output file** (new entrant database)
- **Grouping fields** and **BrkPts.** A typical starting point might be:

Grouping field	Breakpoints	Comment
Age	25 35 45 55	Alternatively, use date of birth with breakpoints equal to the valuation date minus 55, 45, 35, and 25 years, e.g., “1/1/1960 1/1/1970 1/1/1980 1/1/1990”

Also include any coded fields used in the valuation for actives, e.g., Division. It may also be appropriate to have a salary break for high and low paid new entrants.

- Do not group on sex. Rather, have ProVal calculate the PctMale **Sex Field** as shown. Creating the CAPctMale is not necessary for new entrants since it only applies to inactives.
- Select all **Data fields.** These will be saved to the grouped database.
- Select the **Selection expression** topic to limit the new entrant file to active participants hired within the last few years. This is predicated on the notion that future hires will have similar demographics to those most recently hired.



- Click the **Data defaults** topic to apply data defaults to the seriatim data before grouping it. If defaults were used in the valuation you’re trying to replicate with grouped data, apply defaults here for best results.
- **Run** the data grouping. It is generally sufficient to have no more than 5 to 10 new entrant records. Please note the length of time required to run a core projection increases dramatically as the number of new entrant records increases. Thus it is advisable to take the additional time to ensure the new entrant file does not contain superfluous records.

Save the Group Data entry for easy revision should your New Entrant grouping need to be fine-tuned.

The new entrant file is referenced via the **New Entrants…** button under the **Core Projection** command (**Execute** menu or **Shortcuts** pane). For more information, see Core Projections found below.

PLAN AMENDMENTS

You can model many kinds of plan amendments during a forecast. Usually you will use the Plan Amendments topic under **Projection Assumptions (Input** menu or **Shortcuts** pane). Here you need to specify a replacement Benefit Definition for each initial Benefit Definition in your Plan Definition that will be amended during the forecast.

Amendments can be either permanent (such as a new formula) or of a unit benefit or career average update type. If you need to make more than one permanent change, refer to ProVal's help under **Frequently Asked Questions | Projection Assumptions** for a discussion of how to do this.

Special treatment is necessary if your plan amendment changes retirement eligibility. ProVal won't allow such a change during a forecast because termination rates are affected by retirement eligibility. Thus, you need to start the projection with a dummy benefit at your desired eligibility.

Flat dollar plans should generally be set up for amendments during a forecast. This is done by using increase rates and a unit benefit update. Refer to ProVal's help under **Frequently Asked Questions | Projection Assumptions** for a discussion of how to do this. If a flat dollar plan amendment will increase all liabilities and normal costs uniformly, the amendment can most easily be handled without specifying amendments in the projection assumptions. Rather, a "benefit increase for actives" can be specified in deterministic and/or stochastic assumptions.

Special considerations apply if amendments differ for funding and accounting (typically because they have already been reflected for accounting but not yet for funding). Parallel but differently named benefit definitions must be used for (1) the accounting valuation and the (2) funding "replacement" benefit in the projection assumptions. Amendment benefit definitions get special treatment during a core projection and ProVal cannot treat the same benefit definition one way for the funding valuation and another for the accounting valuation.

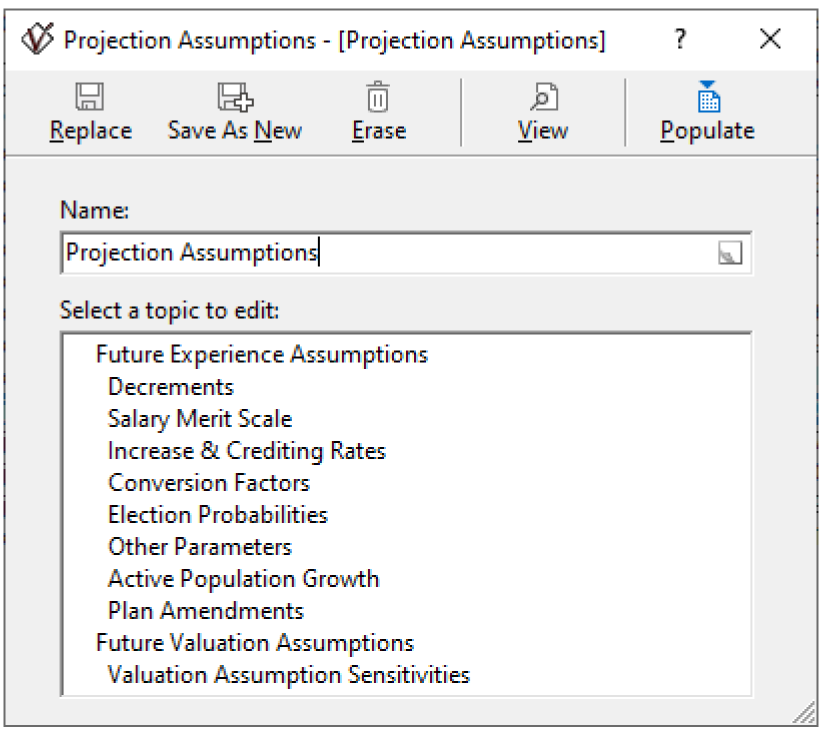
PROJECTION ASSUMPTIONS

Projection assumptions are used primarily to model the experience of the plan population during the forecast period. They project the plan participants from one year to the next by applying decrement assumptions, increasing salaries, crediting interest to cash balance benefits, adding new entrants, and valuing new plan benefits.

Another primary purpose of projection assumptions is to define the 11 “sensitivity forecasts” (i.e., interpolation anchor points) generally included in a core projection. These are 1 baseline forecast (using current valuation assumptions and medium/baseline for all anchor points), 2 inflation sensitivities (low and high), 2 interest sensitivities (low and high) and, if applicable, 2 lump sum experience interest rate sensitivities (low and high), 2 alternate benchmark sensitivities (low and high), and 2 valuation benchmark sensitivities (low and high). From this information, ProVal will be able to interpolate results very quickly for any scenario you may want to run in a deterministic or stochastic forecast.

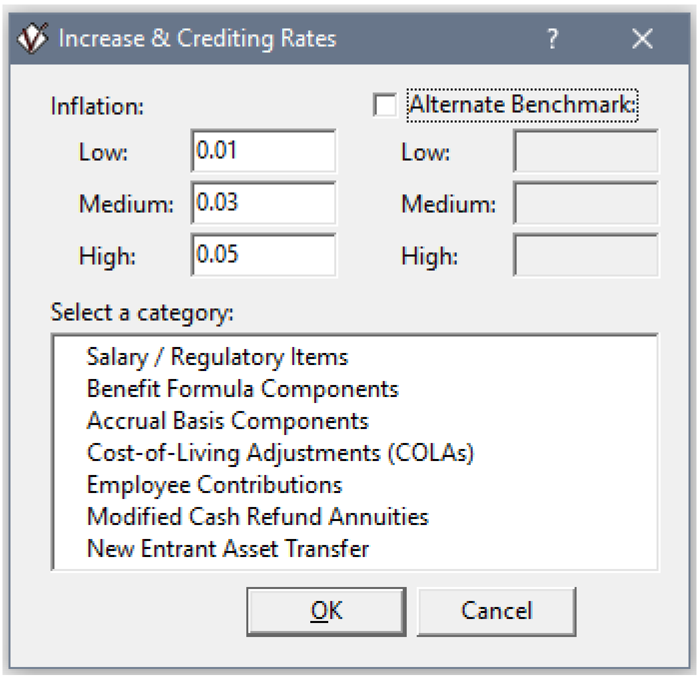
Projection assumptions, in large part, parallel valuation assumptions. Thus, you can explicitly specify experience decrements, mortality, salary scale, increase & crediting rates, etc. To the extent that they differ from your valuation assumptions, this will create gains and losses.

Use the **Projection Assumptions** command (**Input** menu or **Shortcuts** pane) to enter future experience assumptions and (sensitivities for) future valuation assumption assumptions.



Start by using the **Populate…** button to enter projection assumptions consistent with your valuation assumptions and create default sensitivities. You can then review each topic and make appropriate adjustments. Here are a few notes about certain topics:

- The **Increase & Crediting Rates** topic defines the 3 inflation sensitivities that will typically be modeled in the Core Projection. For example, you enter how much salaries, 415 limits, COLA's, etc. increase under each inflation environment. If desired, you can enter calendar year dependent increases to reflect known experience that will happen regardless of the economic environment.

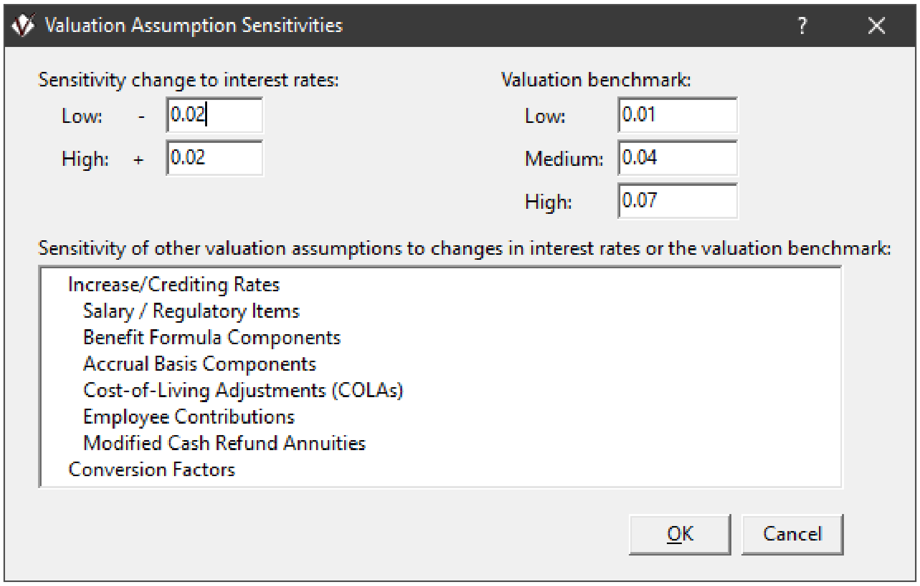


You may choose to model the sensitivity of **Benefit Formula Components, Cost-of-living Adjustments (COLAs)**, and **Employee Contributions** to an Alternate Benchmark, instead of to Inflation. To do this, check **Alternate Benchmark** and enter the three interpolation points. This may be useful, for example, if your Plan Definition includes a Cash Balance component with crediting rates specified in valuation assumptions.

- If you are modeling a plan with some closed and some open groups and want an overall **Active Population Growth** assumption of say 0 (i.e., level active population), it is necessary to project the closed groups first and determine (typically in an Excel worksheet) the growth rate needed on the open groups in order to achieve the total desired active population. Active population growth would then be entered as calendar-year dependent.

- If you are modeling a plan that is expected to have some future **Plan Amendments** during the forecast, you will need to specify replacement Benefit Definitions for each initial Benefit Definition in your Plan Definition. For more information, see Plan Amendments found above.

- **Valuation assumption sensitivities** specify the interest rates and valuation benchmark yields (if applicable) to be evaluated. You can also specify how various valuation assumptions vary as either the interest rate or valuation benchmark yield varies.



The **sensitivity change to interest rates** are “deltas” that determine the interpolation anchor points for the low and high valuation interest rate or valuation benchmark assumptions. That is, the amount that should be subtracted from or added to the valuation assumption in the low or high interest rate environment. If you assume that the interest rate is changing because underlying inflation is changing, you will set all of the values in the table to 1. This says to move other valuation assumptions 100% of the change in the interest rate. If you assume that the interest rate is changing because the risk premium is changing, you can set all of the values in the table to 0. You may choose any value between 0 and 1. Note that you may make different choices for funding and accounting assumptions.

You may choose to model the valuation assumption sensitivities by either tying them to the **Interest Rate** or **Valuation Benchmark**. The Valuation Benchmark may be useful, for example, if your Plan Definition includes a Cash Balance or lump sum component and you’d like to vary the valuation assumption independent of the valuation interest rates. It is also useful in situations when the sensitivity fraction cannot vary with interest rates, such as when the interest assumption is defined by a yield curve.

Valuation Assumptions vs. Experience Assumptions

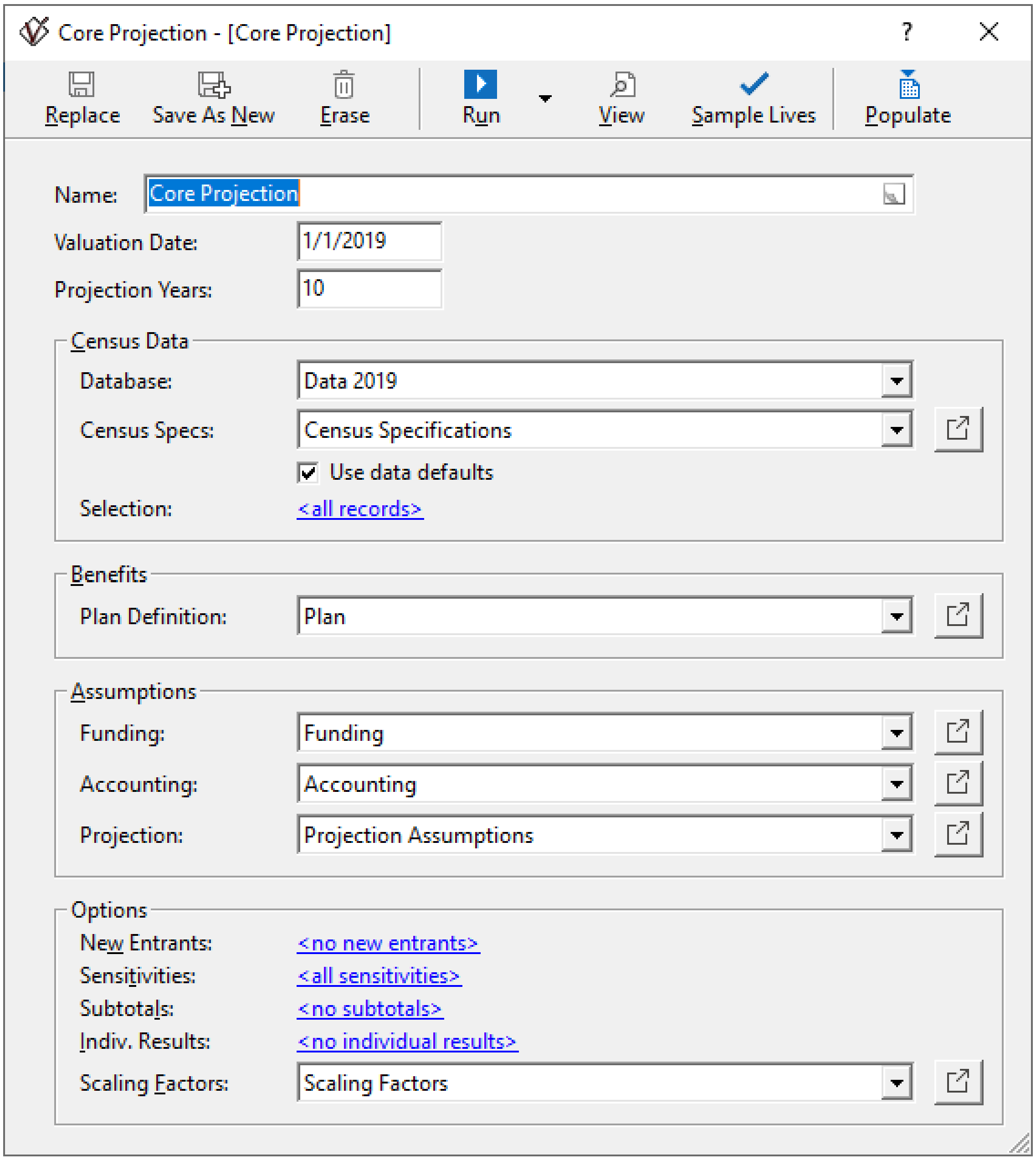
To understand the difference between valuation assumptions and experience assumptions, consider the following example. Suppose your valuation assumption for salary scale is 5%, but your experience assumption is 4% (you might justify this by saying your valuation assumption is a long-term expectation, but your experience assumption is a realistic expectation for the length of the forecast). Applied to a sample record who is age 35 as of 1/1/2016 with historical salaries of 10,000, 20,000, and 30,000:

	Valuation dates					
Age	**1/1/2016**		**1/1/2017**		**1/1/2018**	
33	10,000.00		10,000.00		10,000.00	
34	20,000.00		20,000.00		20,000.00	
35 (attained age)	30,000.00		30,000.00		30,000.00	
36	31,500.00	5%	31,200.00	4%	31,200.00	4%
37	33,075.00	5%	32,760.00	5%	32,448.00	4%
38	34,728.75	5%	34,398.00	5%	34,070.40	5%
38	36,465.19	5%	36,117.90	5%	35,773.92	5%

The shaded regions above represent “history” at each of the valuation dates. Experience assumptions are used to append additional years to history, whereas valuation assumptions are what, at each point in the future, the actuary assumes when calculating liabilities.

CORE PROJECTIONS

A **Core Projection** calculates liabilities and normal costs, just like a valuation. Unlike a valuation, however, a core projection repeats these calculations for future years. A core projection bases its calculations on the data described in a Census Specification, the plan described in a Plan Definition, a set of Valuation Assumptions, and the demographic experience from a set of Projection Assumptions.



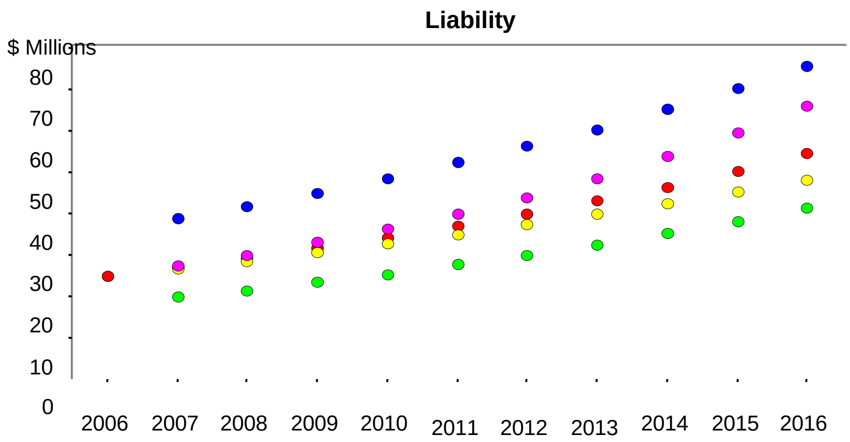
Forecasting is a two-step process. First you must run a **core projection** and then you can run either a **deterministic** or **stochastic forecast.** A core projection only calculates liabilities and normal costs; a deterministic or stochastic forecast uses the core projection's liabilities along with an Asset & Funding Policy to calculate funding contributions and accounting expense.

Forecasts are split into two steps because calculating projected liabilities is time consuming (it can take several hours). With the split, asset assumptions can be changed without having to rerun the liabilities. Moreover, ProVal's forecasting process has been constructed so that inflation, interest rates, lump sum rates (upon which the liabilities are based), alternate benchmark yields, and valuation benchmark yields can be changed without having to rerun the liabilities.

This flexibility is accomplished by producing 11 possible values (i.e., interpolation anchor points) for each liability in the core projection (these are called **sensitivities)** and interpolating among the sensitivities in the deterministic or stochastic forecast. These are 1 baseline forecast (using current valuation assumptions and medium/baseline for all anchor points), 2 inflation sensitivities (low and high), 2 interest sensitivities (low and high) and, if applicable, 2 lump sum experience interest rate sensitivities (low and high), 2 alternate benchmark sensitivities (low and high), and 2 valuation benchmark sensitivities (low and high)

Consider this sample liability projection with 5 sensitivity dimensions:

	Sensitivity Dimension				
Order of performance of scenarios ↓	Valuation Interest Rates	Experience Inflation Environment	Lump Sum Experience Interest Rate	Experience Alternate Benchmark	Valuation Benchmark
Baseline all	7.5%	4%	6.5%	7%	4%
Low interest rates	5.5%	4%	6.5%	7%	4%
High interest rates	9.5%	4%	6.5%	7%	4%
Low inflation	7.5%	2%	6.5%	7%	4%
High inflation	7.5%	6%	6.5%	7%	4%
Low lump sum experience interest	7.5%	4%	6.5%	7%	4%
High lump sum experience interest	7.5%	4%	6.5%	7%	4%
Low alternate benchmark	7.5%	4%	6.5%	5%	4%
High alternate benchmark	7.5%	4%	6.5%	9%	4%
Low valuation benchmark	7.5%	4%	6.5%	7%	2%
High valuation benchmark	7.5%	4%	6.5%	7%	6%



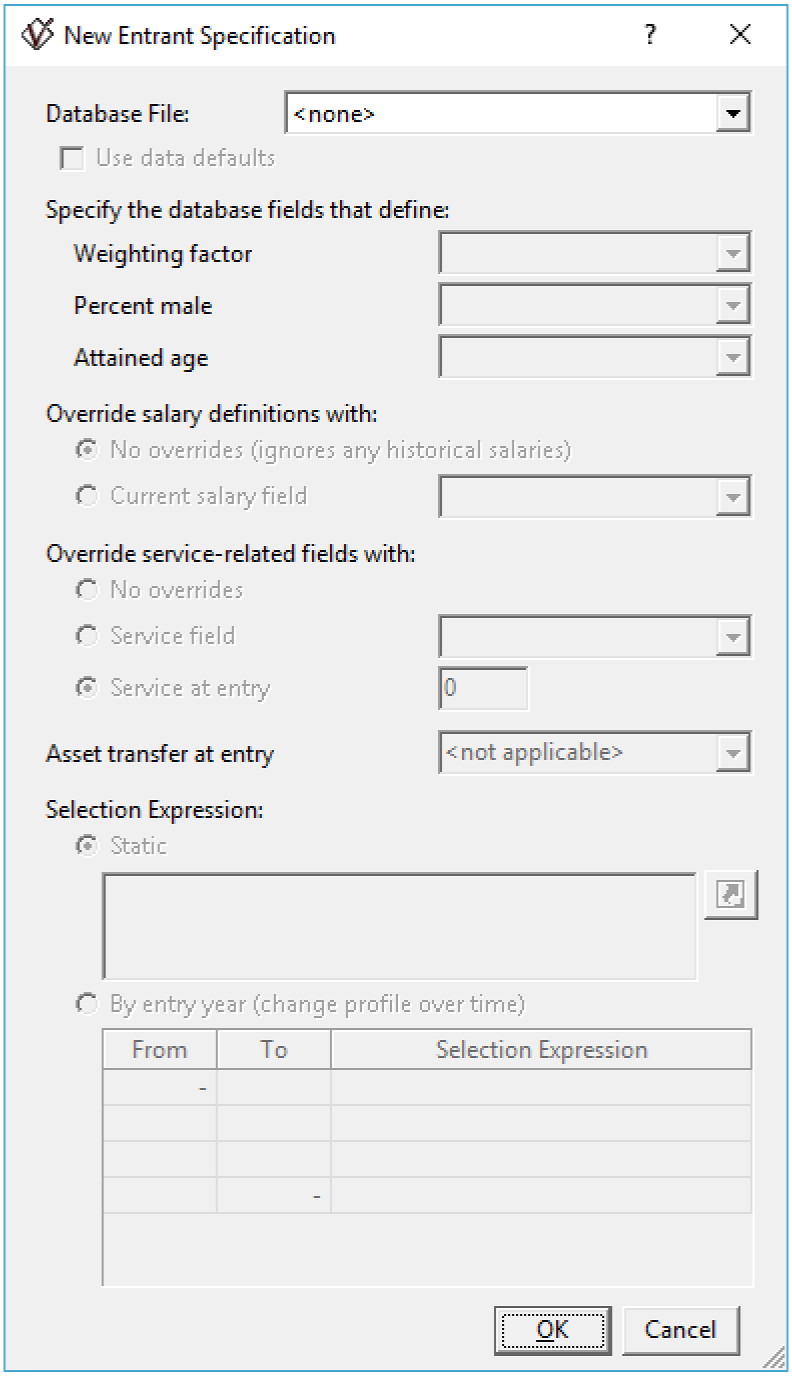
Another way to think about this is that a 10-year core projection consists of 51 valuations (1 + 5 x 10 years). If the plan included a lump sum factor component and lump sum sensitivities were run, along with alternate and valuation benchmark yields, analogous sensitivities would also be established for low and high sensitivity rates of each dimension, and would consist of 111 valuations (1 + 11 x 10 years).

Running a Core Projection

After you have set up your Projection Assumptions, you are ready to run a core projection. Go to **Core Projection** (**Execute** menu or **Shortcuts** pane) and fill in the valuation date, plan definition, valuation assumptions, projection assumptions and number of years for projection. Choose your grouped data and census specifications under Census Data. If running new entrants, select your new entrant data file under New Entrants (see below). Don't forget to apply your scaling factors. **Run** the core projection.

New Entrant Specifications

The new entrant file is referenced via the **New Entrants…** button under the **Core Projection** command (**Execute** menu or **Shortcuts** pane). For information on creating a new entrant database, see New Entrants above.



- ProVal's data grouping tool automatically calculates a Count field that should be referenced as the **Weighting factor**. The weighting factor tells ProVal what proportion of all new entrants in a given year should have specific characteristics. For example, if there are 2 new entrant records, a 25 year old with a Count of 40 and a 40 year old with a Count of 10, then 80% (40/50) of the new entrants each year will be 25 years old and 20% (10/50) will be 40 years old.

- The field referenced as the **Attained Age** should be age at hire not the current attained age. Thus, if your new entrants represent new hires over the last 4 years, you should calculate a

HireAge field for the database if one doesn't exist already rather than referencing the records' attained age or birth date field.

- You generally want to specify **service at entry** to be a value such as 0 or 0.5. This value will be used for all benefit and eligibility service. However, if it is appropriate for some "new hires" to enter with a material amount of service you can specify no overrides, and ProVal will use the service fields in your new entrant database. Note that if new entrant service does not accrue at 1 per year (and you are therefore using a Service Definition in your plan definition and/or valuation assumptions), you must choose no overrides.

Checking Results

- Review demographics (counts and average age, service and salary), including new entrant age and salary relative to initial averages. Typically, you would like to observe that average age and service will be relatively stable and average salary will increase a little faster than inflation, although this is not always automatically reasonable (think about a plan closed to new entrants, for example). Active count should be as desired.

- Consider experience decrements different from valuation decrements if participants' age and service is increasing too fast because they are not decrementing.

- Review projection salary assumptions relative to inflation. This should typically be consistent with the accounting assumptions. Note that new entrant salaries are increased with salary inflation but not the merit scale.

- Check new entrant salaries relative to current salaries. If new entrant salaries are considerably lower than the next service band you may wish to go back and adjust the new entrant file directly.

- If a declining active population is desired but not sufficiently achieved, calendar year-dependent retirement and termination rates can be used.

- Any irregularities may be investigated further using the Core Sample Lives.

ASSET & FUNDING POLICY

There are a few topics under **Asset & Funding Policy** (**Input** menu or **Shortcuts** pane under Valuations) which must be addressed in the context of forecasting.

The **Forecast Analysis** topic should be reviewed, especially if the plan has automatic COLAs. If the plan has automatic COLAs you will typically want to assume that COLAs vary as inflation changes during the forecast (as defined in **Projection Assumptions**). In this case ProVal's default treatment for amortizing COLAs and determining who gets them may not be correct.

Also within this topic is an option about Additional End of Year Contributions. Since the costs in any year depend on the next valuation date, care must be taken to ensure the final year values are correct. To do this, be sure to run your core projections (and later, your forecasts) for one year longer than you intend to show your client (e.g., 11 years for a "10 year forecast").

ProVal automatically calculates the present value of contributions and expense during a forecast. You may want to change the default interest rate for these calculations.

ProVal will calculate an Ultimate Cost output variable during a forecast. This is defined as the present value of the unfunded liability remaining in the reference year (based on market assets and the liability specified) plus the present value of contributions through the reference year. The present value is calculated using the same interest rate as used for the present value of contributions.

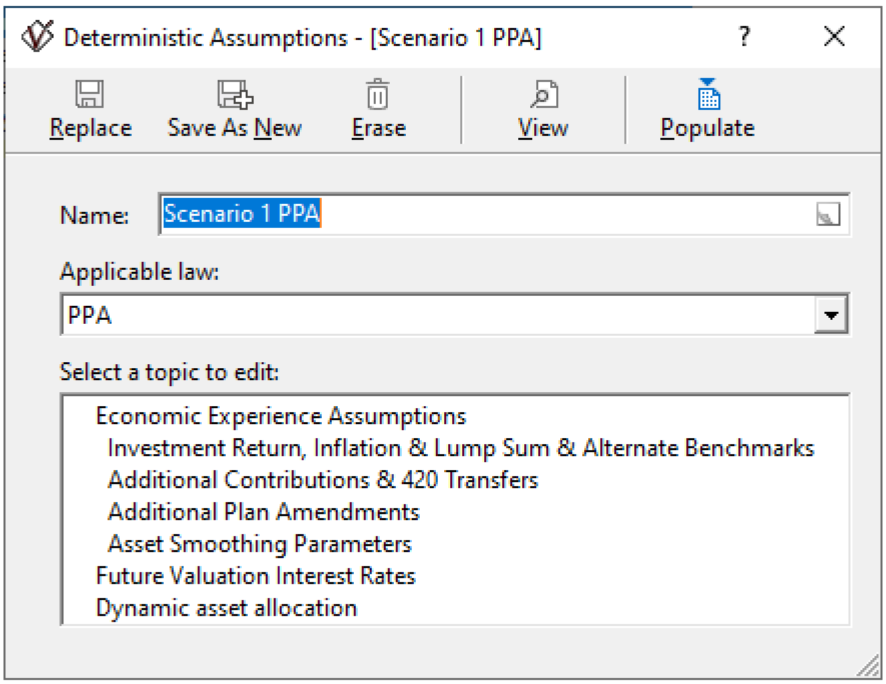
ProVal can calculate a Target Cost. The target cost is defined as the level annual contribution rate (as a percent of salary), which would drive assets to the target liability funded ratio (100% in the example shown) by the year for which results are examined. If desired, a variable contribution pattern can be selected, e.g., 0 contributions for the first n years. It is often helpful to consider target cost during an asset allocation analysis where the probability of meeting the funding target comes into play.

If data grouping, a curtailment or other reasons cause your first year expense to not be exactly correct, enter an Additional Current Year Expense to adjust for the difference (under the **Accounting Methodology** topic).

If you want to do the forecast assuming a particular contribution policy such as minimum contributions, but the first year contributions were other than the policy amount, enter an Additional Contribution to adjust for the difference in the first year's contribution (under the **Contribution Policy** topic).

DETERMINISTIC ASSUMPTIONS

To create a set of **Deterministic Assumptions** (**Input** menu or **Shortcuts** pane), you must specify the applicable law (if in U.S. Qualified Pension mode), investment return, inflation and any future contributions in addition to the expected contribution under your chosen contribution policy. You may wish to model COLAs or active benefit increases found under **Plan Amendments**. Future valuation interest rates default to those set in the valuation assumptions. However, if at the time of your forecast you already know future valuation interest rates, such as the 24-month average PPA segment rates for the following year, you may want model them in your forecast.

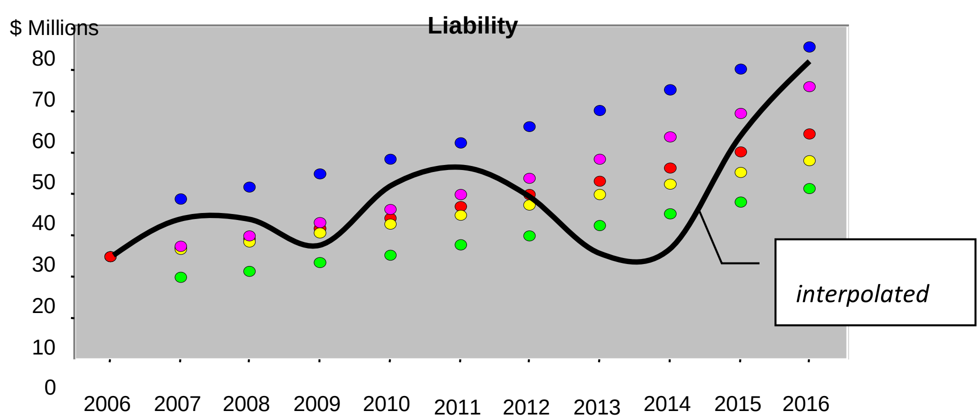


DETERMINISTIC FORECAST

A **Deterministic Forecast** (**Execute** menu or **Shortcuts** pane) calculates funding contributions and accounting expenses, just like a valuation set. Unlike a valuation set, however, a deterministic forecast repeats the calculations for future years. A deterministic forecast bases its calculations on the liabilities calculated in a **Core Projection**, assets entered in an **Asset & Funding Policy**, and future investment returns and interest rates entered in a set of **Deterministic Assumptions**. Future gain/loss events may also be modeled if there are changes in assumptions anticipated in the future.

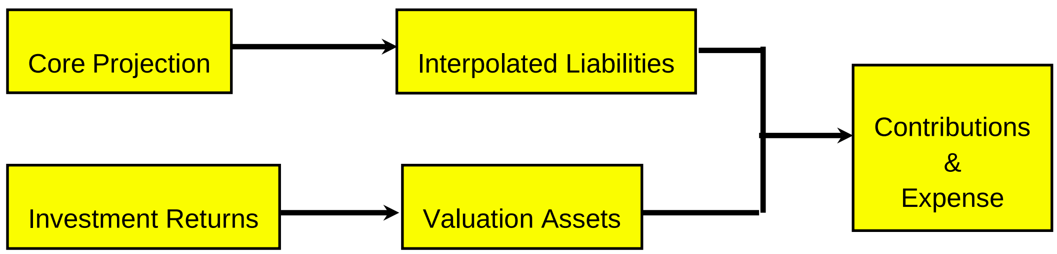
A **Deterministic Forecast** determines liabilities by interpolating among the sensitivities produced in the core projection.

For example:



In the first year (2006 above), the liabilities do not need to be interpolated. Funding contributions and expense can be determined as they would be in a valuation set.

For the next year, liabilities are determined by interpolating among the sensitivities in the core projection. [There is a detailed description of this process in ProVal Help under Technical Reference]. In addition, market assets are brought forward with investment return (plus last year's contribution less benefit payments, expenses, etc.), which determine valuation assets. Contributions and expense can then be calculated for the second year (2007 in our example).



The process repeats itself to get to the third year, fourth year, and so on. In the end, the forecast produces all of the demographic and financial information for the plan.

CAPITAL MARKET SIMULATION

In stochastic forecasts, many trials will be run, each capturing a distribution of desired variables (e.g., pension expense). Since a stochastic forecast bases these trials on quasi-random distribution of possible economic conditions, a capital market simulation is needed. A capital market simulation is simply a database of years and trials for each asset class, each bond yield, and inflation. An example: 2,000 values of inflation are generated randomly such that the mean and standard deviation of the 2,000 modeled trials equals the desired inflation assumptions.

ProVal offers three types of **Capital Market Simulations** (**Input** menu or **Shortcuts** pane under Stochastic Assumptions): **Classic Mean/Variance, Multi-Factor Term Structure**, and **Explicit Corporate Yield Curve** (or, alternatively, you may import your own **custom** simulation). The **Classic Mean/Variance** simulator is a straightforward simulator that is easy to use and explain. The **Multi-Factor Term Structure** simulator should be used when it is important to model changes in interest rates (following bond yields) consistent with changes in bond returns. The **Explicit Corporate Yield Curve** simulator is similar to Multi-Factor Term Structure model; however, it offers the capability to create a full corporate yield curve. A common use of the **Custom** simulation is to import previously exported results of one of the other two types of simulation, with small adjustments in return rates. If you have a custom simulator, you will also have the option to import up to four benchmark yields to use in your forecast (Government Bond, Corporate Bond, and up to 2 Custom Yields). A **Composite** simulation can be created by combining two existing capital market simulations.

MEAN/VARIANCE SIMULATOR

In order to run ProVal's **Classic Mean/Variance** capital market simulator, the following inputs are required:

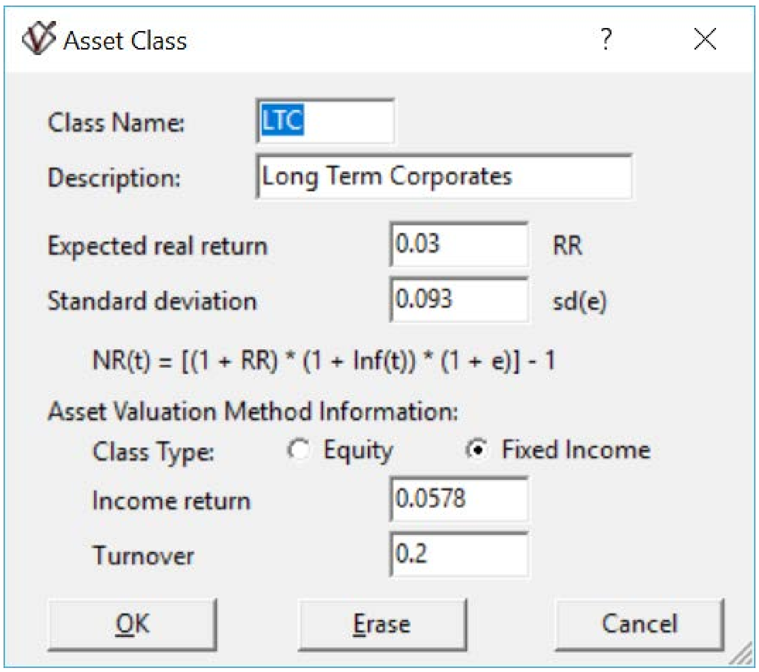
Inflation assumptions. ProVal simulates inflation independently of asset returns so it will be available to vary plan liabilities. The inflation equation assumes serial correlation (next year's inflation has some relationship to last year's inflation) and mean reversion (inflation trends to some long term mean over time):

$$Inf_t = w * Inf_{t-1} + (1-w) * Inf_{\infty} + e$$

The lag parameter (*w*) above is typically between 0.67 and 0.8, and is known as the serial correlation coefficient. The value (1-*w*) is sometimes referred to as the speed of mean reversion. *e* denotes surprise inflation, and the user must specify its standard deviation.

Typically the starting value for last year's inflation will be an empirical amount. Some consultants, however, choose to set it equal to long term inflation (Inf_∞) so as not to build a trend of inflation, returns and yields into their forecast.

Expected real return for each asset class. This is the long-term expected annualized compound return. That is, the geometric expected return rather than arithmetic expected returns. Expected returns should be readily available from the client or the asset consultant.



Standard deviation for each asset class. This is technically the real return standard deviation, although most people parameterize ProVal based on the expected nominal return standard deviation. The simulated nominal returns are a geometric combination of inflation, the real return and an error term,

$$NR_t^{AC1} = (1 + RR^{AC1})(1 + Inf\,)_t(1 + e_t^{AC1}) - 1$$

so they will reflect the standard deviations of both inflation and the real return error term. The standard deviation should be readily available from the client or the asset consultant.

Real Return Correlations, including the correlation of real returns to unexpected inflation and, if a benchmark yield is simulated, to the real yield. Correlations are a more difficult input. Typically nominal return correlations are readily available, but real return correlations are not. Correlations to unexpected inflation and real yields are not typically available. You may use ProVal's illustrative simulators for guidance to these inputs or you may use historical data to determine values for them. Note that ProVal provides the nominal and real correlations as part of the simulation results, so you can fine-tune your inputs as necessary to match the targeted nominal correlations.

When we use historical data we typically use annual data for as many years as possible (or maybe quarterly data if only a few years are available) and

(a) convert nominal returns (or yield) to real returns (or yield) by dividing out inflation:

$$RR_t^{AC1} = \left(\frac{1 + NR_t^{AC1}}{1 + Inf_t} \right) - 1$$

(b) convert inflation to unexpected inflation by taking the difference between the current and prior year's inflation:

$$UEInf_t = Inf_t - Inf_{t-1}$$

(c) use Excel's CORREL function to get correlations.

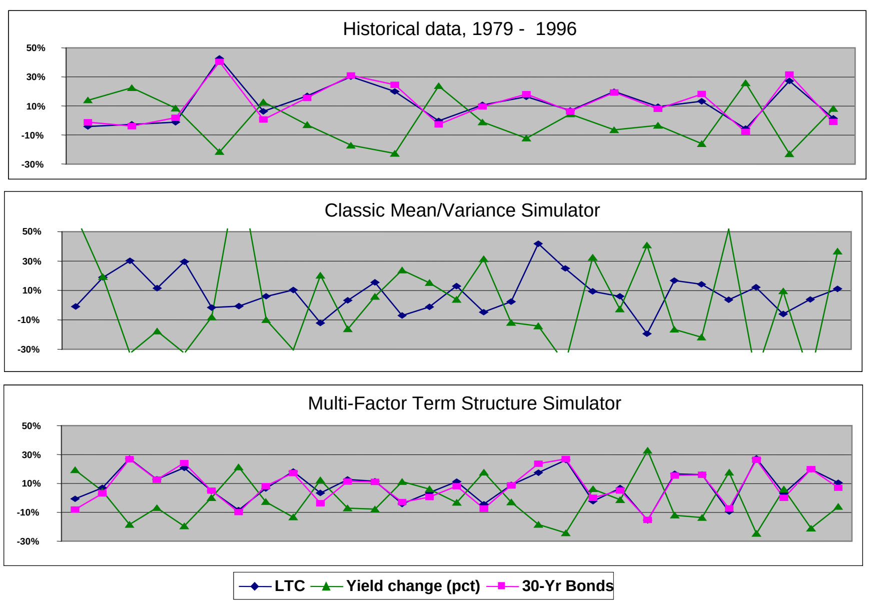
ProVal requires some other asset class parameters relative to asset class performance that would typically not be requested from the client, but are material to the forecast if certain types of asset valuation methods are to be used. These are the yield and turnover, and are labeled "Asset Valuation Method Information" on the screen (along with an indication of whether the class is an "equity" or "fixed income" type). The yield and turnover are used to decompose the total return into its various components: income, realized capital gains and unrealized capital gains. Technically, if the asset valuation method is not dependent on any of these components, e.g., it is based on straight Market Value or compares overall return to expected return, then these parameters are irrelevant and may be ignored. However, it is recommended that these parameters be set to a reasonable value because it would be very easy not to notice that they hadn't been appropriately set when trying to do a sensitivity on the asset smoothing method, or if one client's Capital Market Simulation is used for another client where these parameters would be pertinent.

The **yield** represents dividends for stocks and coupons for bonds. It is a nominal yield, so under a 3% inflationary environment, if bonds have a 4% expected real return they would be typically set to have a 7% yield. Except for fixed income classes derived directly from the multi-factor simulator yield curve (currently only automatic classes labeled _Tbill and _Tbond30), the yield is assumed to remain constant during the forecast and among the trials. Thus, if a particular asset in a particular year and trial has a 9% return and yield is specified as 2%, its return is assumed to be composed of 2% income and 7% capital gains. Income is an available forecasting output item, and is included directly in actuarial/market-related assets under many asset smoothing methods.

The **turnover** assumption is used to decompose the total capital gain into realized and unrealized components. If, for example, the asset class discussed above is specified as having 33% turnover, the 7% capital gain is assumed to be 2.31% realized (7% x .33) and 4.69% unrealized (7% x (1-.33)). Some asset smoothing methods take realized capital gains directly into actuarial/market-related assets while smoothing unrealized capital gains. In addition, some asset smoothing methods are partially based on the book value of assets, which grows with income and realized capital gains.

While the mean/variance simulator is easy to understand and parameterize, it may not produce a desirable simulation of 30-year bond yields (the 30-Year Treasury Yield Benchmark). This is definitely so if you wish to do any kind of asset/liability matching scenario where you expect bond yields and returns to be internally consistent. The mean/variance approach of simulating the bond yield the same way asset classes are simulated means that bond yields' (Note: Not *changes* in bond yields) only relationship to bond returns is through the correlation matrix – a tenuous relationship at best.

These charts illustrate the improvement of the multi-factor term structure model compared to the mean/variance in coordinating yields and returns:



In a non-asset/liability matching scenario, the variability of bond yields may still be a concern. By the nature of the formulas, mean/variance simulator bond yields will have an unreasonably high incidence of very low current liability rates, which in turn could create unreasonably high contributions. There is also the possibility of an unreasonable range of FAS discount rates. However, the user has direct control over the calculation of discount rates while the current liability rates are automatically determined directly from the simulated 30-year benchmark yield.

There are a few key issues to consider with respect to parameterizing bond yields:

In the mean/variance simulator, the specified **standard deviation of the real yield** should be very low (typically 0.0001). If you look at history and conclude that the standard deviation has historically been high, remember that in ProVal you will be simulating a fairly level or directly trending bond yield (within a normal distribution), while historical yields have been cyclical.

Your mean simulated bond yields should produce reasonable mean current liability rates. It is more important that the specified real yield produce a reasonable current liability rate than that it be historically reasonable. Compare the first year of your simulation to current rates to ensure reasonability.

You may want to use the flexibility in ProVal's stochastic assumptions to control the degree of variability in the FAS discount rate. Controlling discount rates is an iterative process where you do a stochastic forecast, view the discount and current liability rate distributions, and then change your capital market and stochastic assumption parameters as appropriate to produce more desirable results.

MULTI-FACTOR TERM STRUCTURE SIMULATOR

The **Multi-Factor Term Structure** simulator starts with a simulation of the yield curve and then determines asset class performance based on the change in the yield curve. <u>You should use the multi-factor simulator, or the Explicit Corporate Yield Simulator, whenever simulated interest rates are relevant.</u> You must use one of these simulators to get reasonable results for any kind of asset/liability matching analysis.

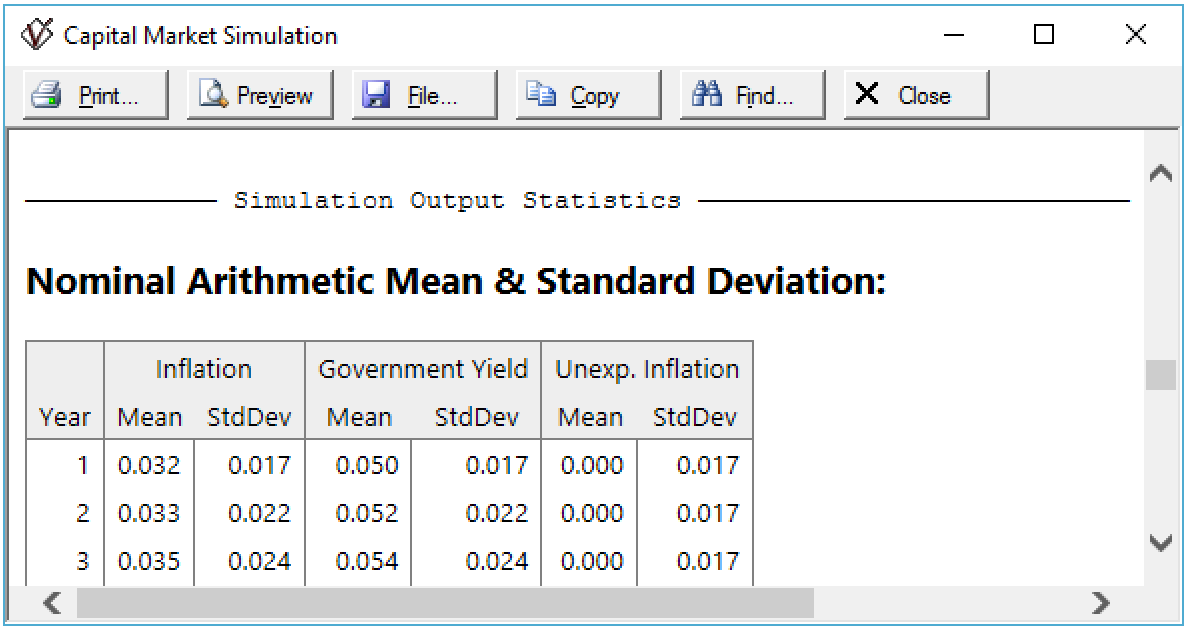
The first step with the multi-factor simulator is to specify the inflation, real rate of interest and term premium parameters. These parameters (there are 10 in total) are used to construct the yield curve.

The **inflation** parameters should be the same as would be used for the mean/variance simulator. For the real rate of interest and the term premium parameter, you would generally start with the parameters in ProVal's illustrative simulation. The **real rate** of interest (on a 1-year T-Bill assuming continuous compounding of interest) is modeled with the same equation as inflation:

$$RR_t = w * RR_{t-1} + (1-w) * RR_\infty + e$$

The **term premium** parameter is the expected excess of the 30-year bond yield over a 1-year T-Bill return. While yield curves are sometimes flat or inverted, typically they are upward sloping and this parameter controls, in essence, the average slope of the curve. When the simulation is run, ProVal will calculate the actual parameter required to coerce the starting yield curve to have the specified relationship between 1 year and 30-year yields. This internally calculated parameter (which is disclosed under the View button) is then part of the simulation of each future year's and trial's yield curve to produce the average slope.

Once you have set the inflation and real rate parameters you can view the initial yield curve if desired. **You should also run the simulation and examine the benchmark yield statistics.**

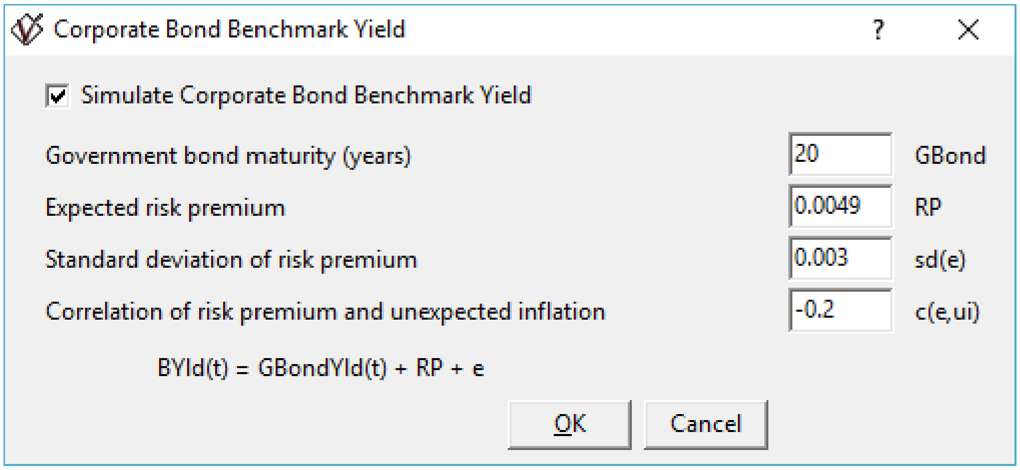


	Inflation		Government Yield		Unexp. Inflation	
Year	Mean	StdDev	Mean	StdDev	Mean	StdDev
1	0.032	0.017	0.050	0.017	0.000	0.017
2	0.033	0.022	0.052	0.022	0.000	0.017
3	0.035	0.024	0.054	0.024	0.000	0.017

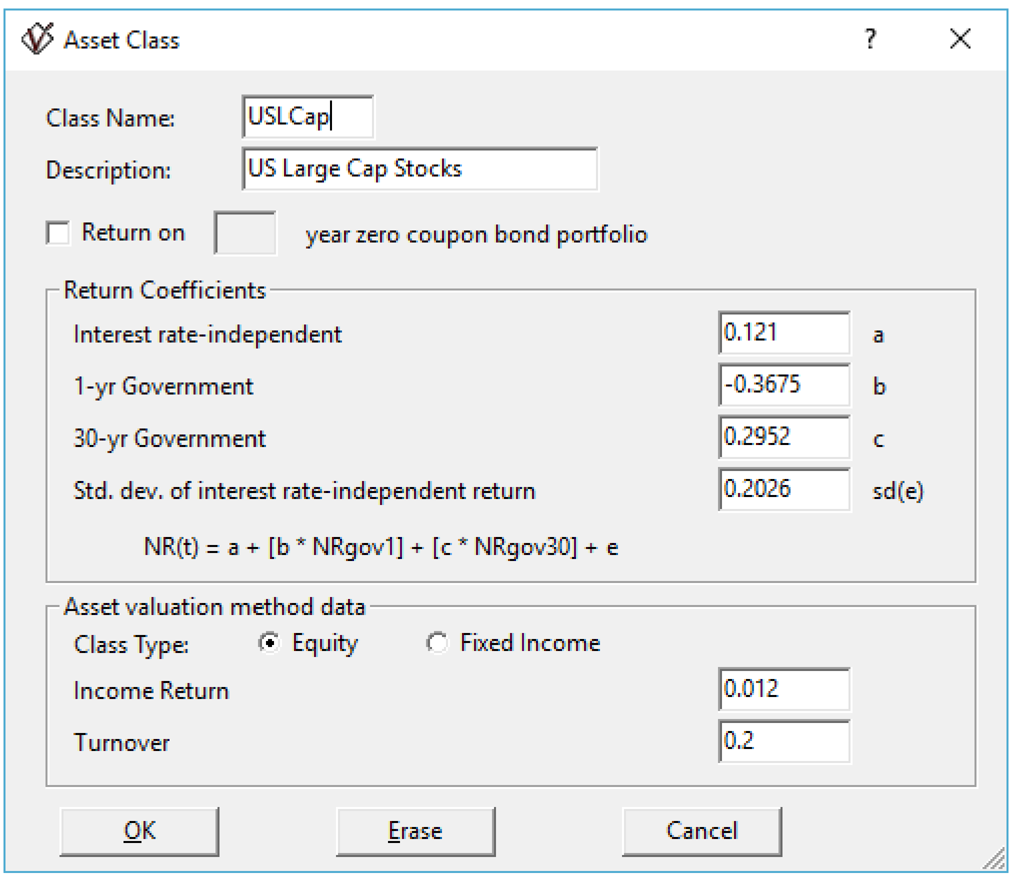
If the yield is trending in an undesirable fashion, you may want to adjust the real return parameters (particularly the relative value of last year's to the long term real rate) until you get a more desirable progression. In this process, keep an eye on the standard deviation of the bond yield to make sure it doesn't get too low (a value of at least 0.8% is desirable, higher if possible). Note that the simulated real rates are important to the yield curve development but otherwise essentially irrelevant to the forecast, so they can be changed at will to affect the mean and standard deviation of the item that is of primary importance: the yield.

If the yield starts out too high or too low relative to current bond yields (remembering that this yield will be used directly for current liability rate calculations), adjust the term premium parameter accordingly.

The next topic – **Corporate Bond Benchmark Yield** – is required if you wish to simulate corporate bond yields for purposes of generating interest rates for PPA funding and/or ASC 715 discount rates.



In the example above, ProVal will extract the yield (from the treasury yield curve) for 20 year bonds and add 49 basis points, with 0.3% standard deviation on the 49 basis points. Again, it is essential that you review the simulation output to ensure that any discount rates tied to this simulation are reasonable compared to current, actual rates. As an example, if current liability rates are actually 5.5%, make sure your stochastic forecast has similar rates in the first year of the forecast and if not, adjust the parameters for corporate bond yields and/or the treasury yield curve and try again.



The general process for parameterizing asset classes when using this simulator is as follows:

- Determine asset class return and standard deviation expectations with the client.

- Set parameters b and c according to their expected performance, e.g. a long term bond asset class would have a dominant c parameter, e.g. b=0 and c=.8. Another example: Global equities have b=0 and c=0, to embed the view that equities have no predictable duration looking forward.

- Alternatively, gather historical data and regress (nominal) 1-year T-bill and 30-year Treasury constant maturity returns against historical returns for the asset classes to be modeled. This approach carries limitations and challenges, please call support if you wish to pursue.

  ProVal will model asset classes based on the following equation:

  $$a + (b \times NR_{TBill1}) + (c \times NR_{TBond30}) + e,$$

  where *a* is the regression residual, *b* and *c* are the regression coefficients for T-Bills and T-Bonds respectively, and *e* is a normally distributed error term with a zero mean and a standard deviation equal to the regression standard deviation of the residuals.

- Initially parameterize your asset classes in the multi-factor simulation. Run the simulation and save the results. You may find it easiest to save them to an Excel worksheet.

- Check the simulated nominal return correlations. If they are not as expected, this can usually be fixed by setting the correlation of the residuals equal to the expected correlation of the nominal returns.

- Compare the simulated returns and standard deviations for each of your asset classes to the desired expected values. Then, change the *a* and standard deviation of *e* terms above iteratively until the simulated results for each asset class are consistent with your expected values. To do the comparison of returns and standard deviations, you may want to focus on the last year simulated, or you may want to average all of the simulation years. These two approaches will give different results particularly if inflation is trending during the simulation.

- While you can compare the standard deviations directly, your expected returns are comparable to geometric mean returns while the simulation standard output produces arithmetic means. To do the return comparison, convert your expected returns to an arithmetic mean basis by adding the expected returns to ½ the variance for each asset class:

  $$Mean_{Arith} = Mean_{Geom} + .5(StdDev)^2$$

  (Note that if you use an average of all of the simulation years, ProVal can calculate the geometric mean return and standard deviation from within the Efficient Frontier command. Just select an asset-only Efficient Frontier and have ProVal "populate" it with the geometric average nominal returns from your saved Capital Market Simulation.)

Since the standard deviation affects the mean but not vice versa, focus first on changing the standard deviation of the error term parameter for each asset class until the simulated asset class standard deviation matches your target to a reasonable tolerance. It is usually possible to finalize the "calibration" of the multi-factor simulator in 3 to 4 iterations.

Explicit Corporate Yield Curve Simulator

The Explicit Corporate Yield Curve capital market simulator is built on top of the multi-factor term structure capital market simulator. The first steps in the simulation are the same as the Multi-Factor model: generate inflation and Treasury yields. See **Multi-Factor Term Structure Simulation Type**, above, for a description of these processes.

Next, enter the thirteen bond spread parameters discussed under the Explicit Corporate Bond Spreads topic. Under the hood, the general process is as follows:

- The long term (30 year) corporate spread is calculated first, based on the five long term spread parameters (mean, standard deviation and serial correlation of the spread plus its correlation with the long term Treasury yield and with unexpected inflation). The credit spread is assumed to be lognormally distributed.
- Next the short term (1 year) corporate spread is calculated, based on the assumed correlation of the short and long term spreads and the five short term spread parameters (mean, standard deviation and serial correlation of the spread plus its correlation with the short term Treasury yield and with unexpected inflation). As with the long term spread, the credit spread is assumed to be lognormally distributed.
- The corporate bond (spot) curve values are then calculated as the sum of the Treasury zero coupon bond curve value at a duration and the simulated credit spread at that duration.
- The corporate bond benchmark yield calculated by the simulator, and available for simulating pension liabilities (although available in OPEB mode also), is defined in each year as the current 30-year par coupon rate, the same coupon rate that will be used to determine the 30 year corporate bond return in the next simulation year.

This simulator will also calculate and store nominal returns based on changes in the 1-year and 30-year corporate bond yields. These reference returns may subsequently be used to define asset class returns in the same way that government bonds are currently available in the Multi-Factor simulator. This allows you to establish asset classes whose returns on consistent with the changes in corporate bond yields.

- The one year corporate bond return, labeled as the asset class _CBill in the simulation output, is defined as the T-Bill return (_TBill in the simulation output) plus the short term corporate spread.

- The 30 year corporate bond return, labeled as the asset class _CBond30 in the simulation output, is defined as the return on a 30 year bond (paying coupons semi-annually) purchased at par at year n-1 and then sold a year later (at year n). For this calculation, the credit spread used to determine the corporate bond spot rates at each of the 60 durations from 0.5 to 30 (half year intervals between durations) is determined by linearly interpolating between the short and long term spreads (using the short term spread for both durations 0.5 and 1).

Thus, for example, you can establish an asset class whose returns are defined by a mixture of government and corporate bond returns.

EFFICIENT FRONTIER

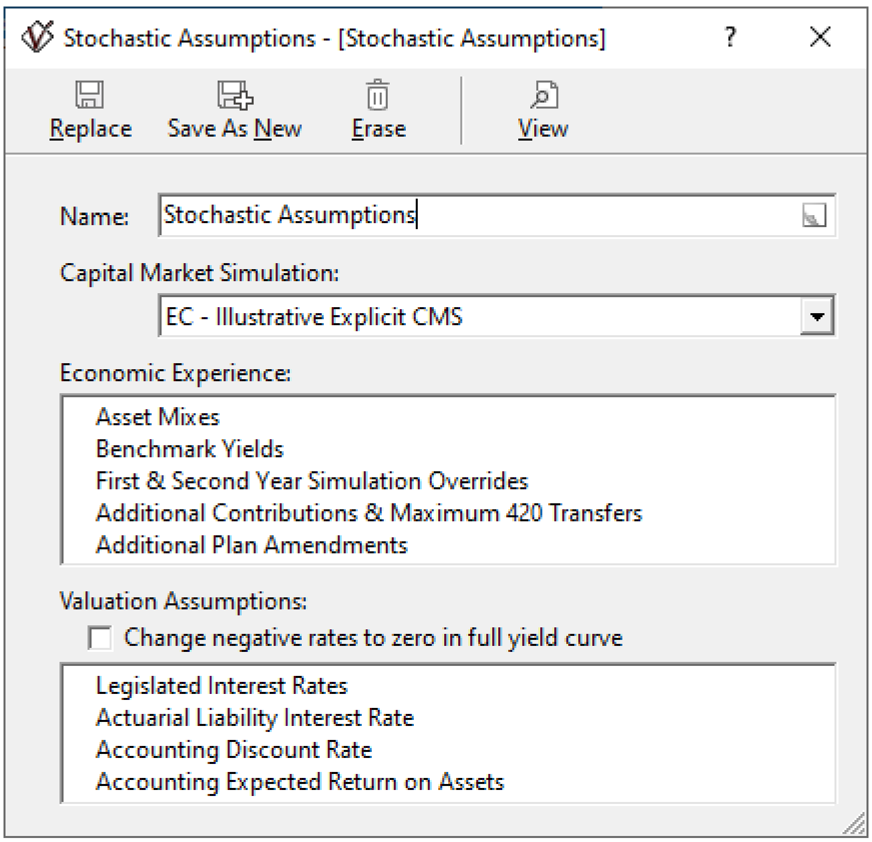
You may want to use ProVal's **Efficient Frontier** command (**Input** menu or **Shortcuts** pane under Stochastic Assumptions) to create an efficient frontier based on your client's asset assumptions including any constraints (e.g., no more than 5% cash). If you like the efficient frontier you can pick your mixes directly from it to set up your stochastic assumptions.

When you view the efficient frontier output, look for any asset classes shown as "0" with no decimals in their allocation. If there are any, then this asset class is not on the efficient frontier at all. In this case, review your asset assumptions for reasonableness (e.g., risk increasing with return).

STOCHASTIC ASSUMPTIONS

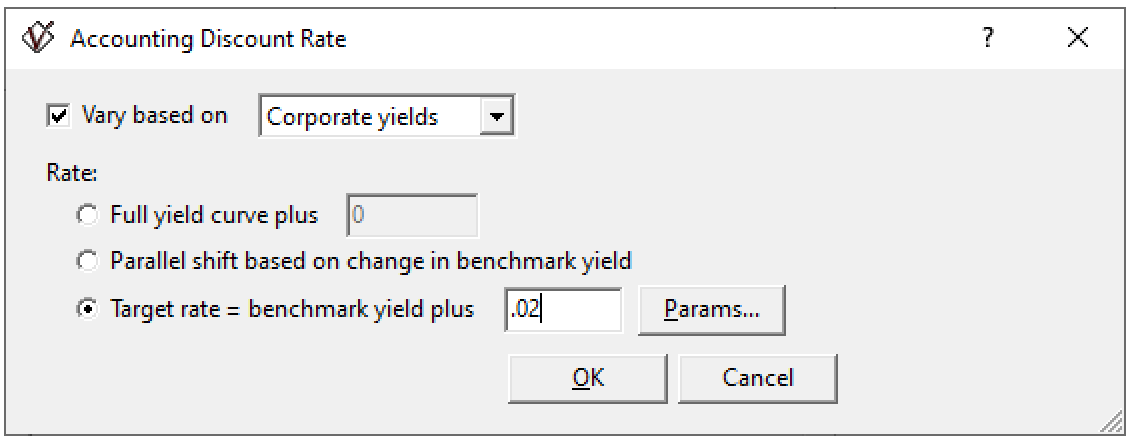
You can specify any number of **Asset Mixes** to be run in a single stochastic forecast. These mixes can be entered directly, or you can use the Efficient Frontier button to select them directly from the efficient frontier. Check the **Dynamic Asset Allocation based on** box to choose an asset mix based on funded ratio, interest rate environment, or calendar year.

You can select whether to have your annual asset mix returns calculated geometrically or arithmetically. ProVal's traditional geometric basis can be thought of as approximating continuous re-balancing, while the arithmetic basis is more in line with annual rebalancing. The arithmetic basis will produce higher returns. The compound returns over time will approach the geometric mean.



The **First & Second Year Simulation Overrides** topic is used when you wish to override modeled experience with actual, known experience. This can occur when your base year (Year 0, in ProVal terminology) is in the past. An example: doing a forecast in April 2023 based on the most recent valuation date of 1/1/2022. Although 1/1/2023 valuation is not complete, the 2022 return and many of the economic values for 2022 like inflation and benchmark yields are known. Without an override, ProVal would generate a distribution for these values, which would imply variability in those nowhistorical results where, in fact, those results are known. Check the **Use known asset values to override return** box to have ProVal project assets based on a known asset value at a specified point in time, up to 2 years, from the Valuation Date.

If your **Capital Market Simulation** includes bond yields, you will have the option of changing Valuation Assumptions for interest rate and expected return on assets during the forecast. Typically, you will want to vary the Legislated Interest Rates and the Accounting Discount Rate based on one of the available bond yields. For these two Valuation Assumptions, you can forecast to the full yield curve. For the Actuarial Liability Interest rate and/or the Accounting Expected Return on Assets (such as to correspond to the expected return of a new asset mix), you can set the future interest rates based on a parallel shift or target rate. These changes will all apply after the first year.



Note that you must be using an **Explicit Corporate Yield** Curve or **Custom** type capital market simulator in order to forecast to the full yield curve.

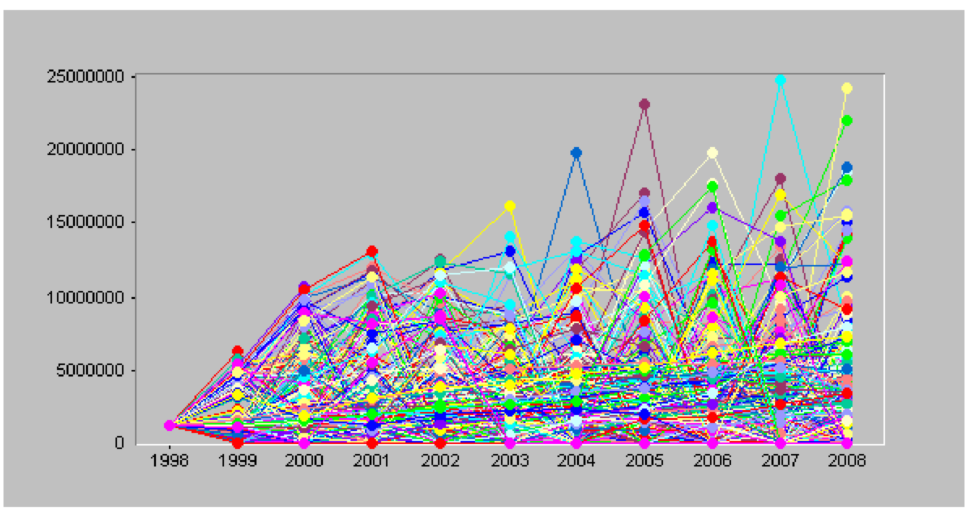
In **U.S. Qualified Mode**, you have limited control over the **Legislated Interest Rate** since its construction is prescribed by regulation.

STOCHASTIC FORECAST

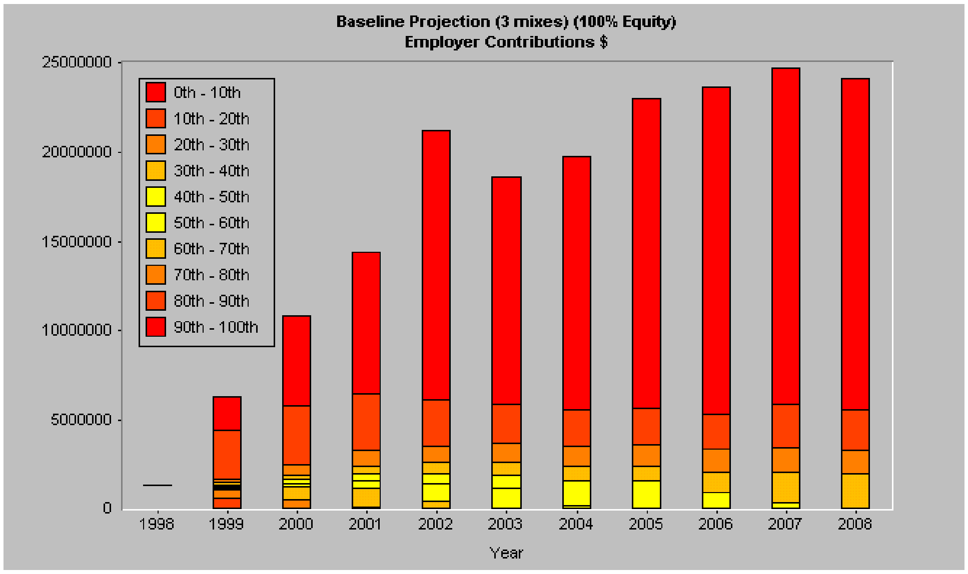
A **Stochastic Forecast** (**Execute** menu or **Shortcuts** pane) bases its calculations on the liabilities calculated in a **Core Projection**, asset and contribution policy details from an **Asset & Funding Policy**, and future investment returns and interest rates simulated and specified in a set of **Stochastic Assumptions**. Future gain/loss events may also be modeled if there are changes in assumptions anticipated in the future.

Conceptually, a stochastic forecast can be viewed as many (up to 25,000) deterministic forecasts, with each deterministic forecast being based on a random simulation of investment returns, inflation, benchmark yields, and valuation interest rates. This feature allows you to study the financial effects of fluctuations in the economic environment on the pension plan under analysis. It is also an ideal tool to study the financial implications of alternative asset mixes, ranging from low-return / low-risk portfolios to high-return / high-risk portfolios.

The stochastic forecast takes each of these deterministic “slices” and produces contribution and expense results in the same manner as a deterministic forecast. However, if we tried to graph contributions like we did for the deterministic forecast, we would end up with 2,000 lines of “spaghetti”:



For this reason, stochastic results are often presented as percentiles. For each year, ProVal will report the highest value, the lowest value, and every 5th percentile in between. In addition, the resulting mean and standard deviation are available. To graph these statistics, you might use a graph like this:


Baseline Projection (3 mixes) (100% Equity)
Employer Contributions $
25000000
20000000
15000000
10000000
5000000
0
0th - 10th
10th - 20th
20th - 30th
30th - 40th
40th - 50th
50th - 60th
60th - 70th
70th - 80th
80th - 90th
90th - 100th
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
Year