9

WRITING FORMULAS

9

WRITING FORMULAS

USING FORMULAS IN PAYROLL

The Payroll system uses Formulas to calculate Deductions and Withholdings based on Earnings and Tables. These Formulas involve Variables, Constants, Operations, and Functions.

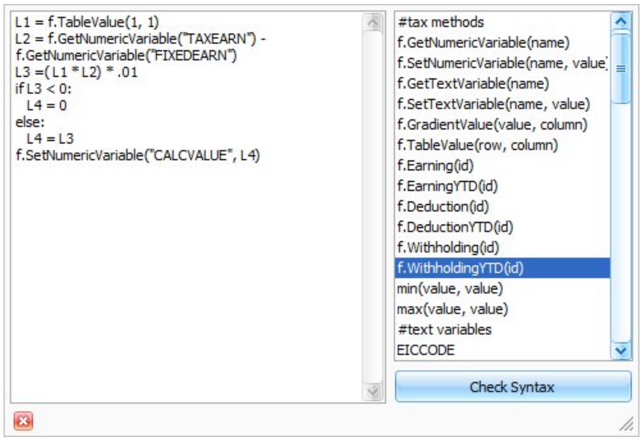
- **Commands** are functions that tell the Formula what to do to retrieve a value.
- **Variables** are an equivalent that stand for something which can change value as other actions occur.
- **Constants** are positive or negative numbers.
- **Factors** are a constant defined in a Formula or in an Employee or Employer Override Factor. You can set up to six Factors in each Formula. Employee and Employer Formulas have up to six Override Factors.
- **Functions** - you can use three types of Functions when constructing Formulas: Table Lookups, Conditionals, and MIN/MAX functions.
- **Operations** proceed according to standard mathematical rules. Multiplication and division are first performed. Addition and subtraction are performed afterwards. Functions within parentheses are performed before Operations outside of parenthesis.

The Payroll system uses Formulas to calculate Deductions and Withholdings based on Earnings and tables. You can use variables for numbers such as year-to-date amounts, gross earnings, and so forth, which you can manipulate using Operations and Functions, much like a spreadsheet program. You can also look up Tax Rates and other information in Formula Tables.

You can use positive or negative numbers (Constants) in Formulas. You can also use these variable which store payroll information values set by the Payroll system as it calculates the employee’s paycheck:

TRAVERSE uses Iron Python scripting for the scripting language to write the Formulas. There are many books available as references to use when writing scripts. A summary of valid Variables, Operations, and Functions follow. If you are new to Formulas or you need to review them, several examples of Formulas are at the end of this section.

Within the Formula box that is displayed when the combo box arrow is clicked you can return Commands and Variables into the Formula being generated by double-clicking on the selection you want brought into the Formula.



```
L1 = f.TableValue(1, 1)
L2 = f.GetNumericVariable("TAXEARN") -
f.GetNumericVariable("FIXEDEARN")
L3 =(L1 * L2) * .01
if L3 < 0:
  L4 = 0
else:
  L4 = L3
f.SetNumericVariable("CALCVALUE", L4)
```

When Variables and Factors are used the values must be enclosed within double quotes (“ “) within the parentheses.

Click on the Check Syntax button when finished entering the Formula to see if you have the syntax for the Formula correct.

Tax Methods

To retrieve a value to put into the Formula, you must use Methods to tell the Formula what value you want to retrieve. Commands are used along with the Variables to set the values to bring into the Formulas. The most commonly used Commands are f.GetNumericVariable and f.SetNumericVariable. You will see Commands used in many of the sections following this section.

The Methods are grouped in the Formula box by the heading **#METHODS**.

Below is a table with the commands and descriptions:

Command	Description
f.GetNumericVariable(name)	Bring a variable that contains a number value into the formula.
f.SetNumericVariable(name, value)	Set the numeric variable value that has been calculated to bring the value into the calculated check.

Command	Description
f.GetTextVariable(name)	Bring a variable that contains a text value into the formula.
f.SetTextVariable(name, value)	Set the text variable value that has been calculated to bring the text into the calculated check.
f.GradientValue(value, column)	Bring the value from a table that has graduated rows into the formula.
f.TableValue(row, column)	Bring the value from a table that does not have graduated rows into the formula.
f.Earning(id)	Bring in the amount calculated on the check for a specific earning code.
f.EarningMTD(id)	Bring in the amount month to date for a specific earning code.
f.EarningYTD(id)	Bring in the amount year to date for a specific earning code.
f.Deduction(id)	Bring in the amount calculated on the check for a specific deduction code.
f.DeductionMTD(id)	Bring in the amount year to date for a specific deduction code.
f.DeductionYTD(id)	Bring in the amount year to date for a specific deduction code.
f.Cost(id)	Bring in the amount calculated on the check for a specific employer cost.
f.CostMTD(id)	Bring in the amount month to date for a specific employer cost.
f.CostYTD(id)	Bring in the amount year to date for a specific employer cost.
f.Withholding(id, state, local)	Bring in the amount calculated on the check for a specific withholding code.

Command	Description
f.WithholdingMTD(id)	Bring in the amount month to date for a specific withholding code.
f.WithholdingYTD(id, state, local)	Bring in the amount year to date for a specific withholding code.
f.EmployeeTaxEarnings(id, state, local)	Bring in the taxable amount calculated on the check for a specific earning code.
f.EmployeeTaxEarningsMTD(id, state, local)	Bring in the taxable amount month to date for a specific earning code.
f.EmployeeTaxEarningsYTD(id, state, local)	Bring in the taxable amount year to date for a specific earning code.
f.Tax(id, state, local)	Bring in the amount calculated on the check for a specific tax code.
f.TaxMTD(id, state, local)	Bring in the amount month to date for employer tax.
f.TaxYTD(id, state, local)	Bring in the amount year to date for a specific tax code.
f.EmployerTaxEarnings(id, state, local)	Bring in the taxable amount calculated on the check for a specific earning code for employer tax calculation.
f.EmployerTaxEarningsMTD(id, state, local)	Bring in the taxable amount month to date for a specific earning code for employer tax calculation.
f.EmployerTaxEarningsYTD(id, state, local)	Bring in the taxable amount year to date for a specific earning code for employer tax calculation.

Numeric Variables

You can use positive or negative numbers (Constants) in Formulas. Numeric Variables are grouped, in the Formula box, by the heading **#NUMERIC VARIABLES**.

You can also use these variables in place of the numbers:

NOTE: All variables must be enclosed in quotes " ".

Variable	Description
ADJEARN*	Adjusted earnings for net pay deductions
ADJHR	Adjusted hours (minus exclusions).
ADJMIN	Adjust to minimum wage flag.
ADJSUIEARN	Total SUI earnings before the Self Adjust Month changed.
ADJSUIWITH	Total SUI withholdings before the Self Adjust Month changed.
DEDEXCL	Total amount of deductions exclusions.
DEDUCTIONTOT	Total deductions for the current check.
EARNEXCL	Total amount of earnings excluded.
EARNHOUREXCL	Total hours excluded.
EMPDEPT	Default department ID for the current employee
EMPFICAWH	Employee FICA Contribution (OASDI and Medicare).
EMPTYPE	Employee type (H or S)
EXEMPTIONS	Number of exemptions per employee.
EXTRAWH	Extra withholdings for the employee.
FC1 - FC20	Factor prompt values
FEDEXEMPT	Number of federal exemptions for the employee
FEDEXTRAWH	Extra federal withholding for the employee
FEDFIXEDWH	Fixed federal withholding amount for the employee

Variable	Description
FEDMARITSTATUS	Marital status of the employee for federal withholding.
FEDWITH	Total employee federal withholdings (including OASDI, Medicare, and EIC).
FIXEDEARN	Total fixed earnings for the employee.
FIXEDPCT	Fixed percent for the withholding.
FIXEDWH	Fixed withholding amount.
FWHWITH	Employee federal withholdings (not including OASDI, Medicare, and EIS).
GETDEDUCTION	Retrieve a deduction taken out of an employee's check. Need to list deduction code in ("xxx").
GRANDTOTEARN	Total gross earnings, not including earnings that are not included in net pay.
GRANDTOTGROSS	Total gross earnings, including all earnings.
GRANDTOTNET	Net pay for the employee
GROUPCODE	Local tax authority Group Code ID
HOURLYRATE	The employee’s hourly wage rate
HOURS	Total hours worked.
LOCALWITH	Total employee local withholdings.
MINWAGE	Minimum wage.
NETDEDTOT	Total deductions based on the net pay for the employee
PALOCAL	Local code if calculation local withholding
PASTATE	State code if calculating state withholding
PASTATUSCODE	Current withholding code

Variable	Description
PAYPERIODS	Total pay periods for the year; taken from the PAEGxxx file.
PAYEAR	The payroll year to bring data into the formula from.
PERIODCODE	Current deduction run code for the group code the current employee is in.
REGHRS	Regular hours worked, excluding leave hours.
SALARY	Employee’s pay period salary
STATEWITH	Total employee state withholdings.
STATUS	Marital status for current withholding.
TAXEARN	Taxable earnings per tax authority (total earnings minus all exclusions).
TAXHOURS	Taxable hours per tax authority (total hours minus all exclusions).
TAXMONTH	Tax month being processed (as set in the Business Rules)
TIPS	Tips accumulated for the employee.
TOTEARN	Total earnings.
UNCOLMED	Year-to-date uncollected Medicare.
UNCOLOASDI	Year-to-date uncollected OASDI.
WEEKSWORKED	Number of weeks worked for the employee
WEEKSWORKEDLIMIT	The value from the tax authorities setup in the weeks worked limit field.
WITHHOLDINGEARNINGS	Earnings used to calculate current withholding amount. Accounts for limits.
YTDEARNINGS	Year-to-date earnings for the withholding.

Variable	Description
YTDFICATIPS	Year-to-date FICA tips (used in employee OASDI to figure FICA tips).
YTDTIPS	Year to date tips accumulated for the employee.
YTDWITHHOLDINGS	Year to date withholdings accumulated for the employee.

*For Gross, **ADJEARN** is set to the same amount as **GRANDTOGROSS**, minus any Earning Code exclusions that may exist for that Deduction. For Net, **ADJEARN** is set to **GRANDTOTEARN**. Then **DEDUCTIONTOT**. **FEDWITH**, **STATEWITH**, and **LOCALWITH** are subtracted to account for all withholdings up to that point. So if you have two net pay deductions, the first one is calculated and that amount is added to **DEDUCTIONTOT**. The second deduction will follow the same process, with **DEDUCTIONTOT** having the updated deduction total from the previous net pay deduction.

When working with variables in a formula for TRAVERSE 11 you must first select the **f.GetNumericVariable** command from the list of selections. The variable must be enclosed in quote marks within the parentheses **f.GetNumericVariable("TAXEARN")**.

Text Variables

Text variables can be used to return a variable that has a text value. Text variables can be used to bring in things like statuses for tax purposes. The Formula might depend on a status to calculate a different amount based on the status for which an employee is set up.

Test variables are grouped, in the formula box, by the heading **#TEXT VARIABLES**.

Below is a table with the commands and descriptions:

Variable	Description
EICCODE	The EIC code selected in the employee information setup.
EMPDEPT	Employee department code
EMPLABORCLASS	Employee's labor class
EMPTYPE	Employee's wage type, Hourly or Salaried

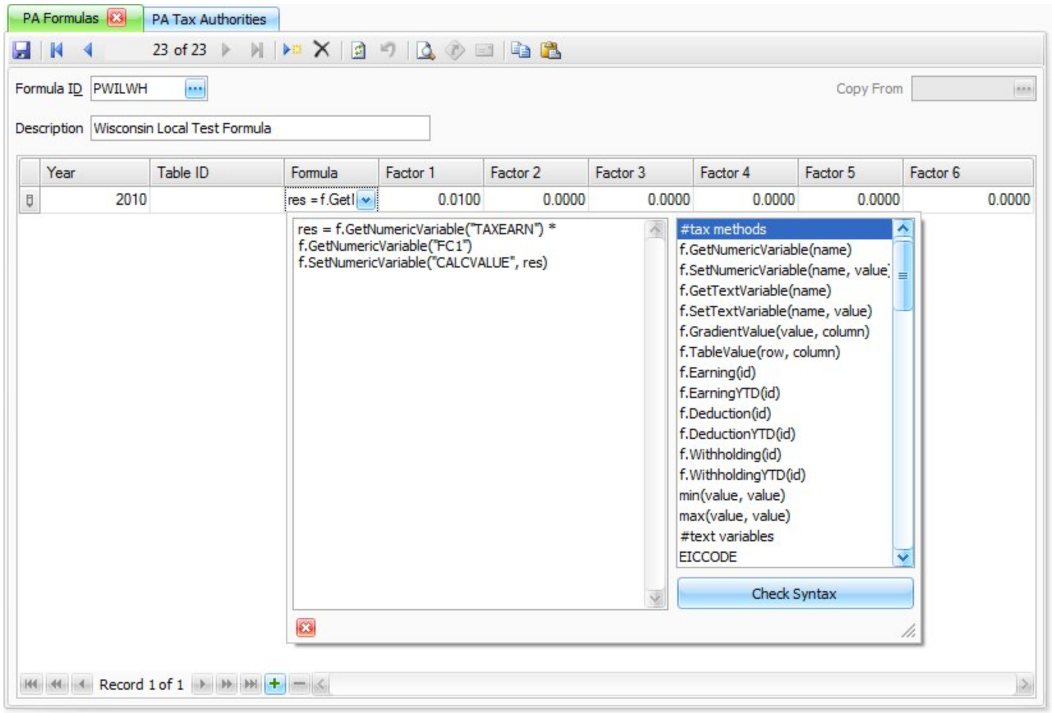
Variable	Description
FEDMARITSTATUS	The status selected on the federal tax tab in the employee information setup.
LOCALCODE	Local tax code
STATECODE	State tax code
STATUS	The status selected on the state and local tax tabs in the employee information setup.
TABLEID	The table ID used for the formula.

Formula factors

Formula factors are variables used to change the base rate in a Formula without changing the Formula. Each Formula can have six factors. Factors can be used in Formula lines by entering "FC*n*"***,*** where *n* is the number of the factor.

When working with variables in a Formula for TRAVERSE 11 you must first select the **f.GetNumericVariable** command from the list of selections. The variable must be enclosed in quote marks within the parentheses **f.GetNumericVariable(**"**FC1**"**)**.

Using Formula Factors



Formula ID PWILWH

Description Wisconsin Local Test Formula

Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2010		res = f.GetI	0.0100	0.0000	0.0000	0.0000	0.0000	0.0000

```
res = f.GetNumericVariable("TAXEARN") *
f.GetNumericVariable("FC1")
f.SetNumericVariable("CALCVALUE", res)
```

For example; you can set up a Formula that multiples taxable earning by a specific tax percentage. The formula itself can be entered in a single line:

f.GetNumericVariable(“FC1”) * .5

If you set Factor 1 to .05, the employee’s taxable earnings will be multiplied by five percent. Later, you can change the factor or override it for an individual employee.

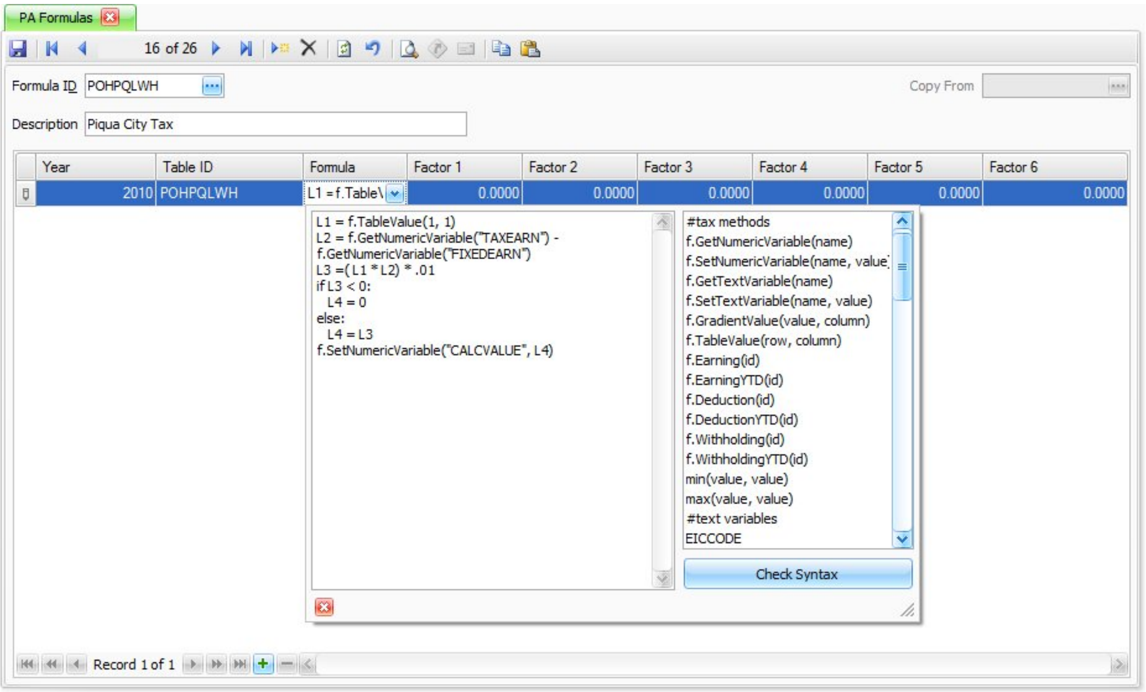
NOTE: Formula factors can be overridden for Deductions for an individual Employee on the Deductions tab of the Employee Information function on the Setup and Maintenance menu. Formula factors for Withholdings can be overridden for an individual Employee by selecting Formula and entering the values in the Formula Override fields.

Formula lines

Each line of a formula sets the value of a variable Ln (where *n* corresponds to the line number the value of the variable is calculated on). Lines are calculated in sequential order, line L1 is calculated before L2, and so on. You can use variables storing the results of previous lines with other variable in subsequent Formula lines. The result of the entire Formula is the result of the calculation on the last line of the Formula.

Line numbers can be defined many ways in the TRAVERSE Formulas. Most of the examples in this chapter use the L1, L2 convention for naming lines other examples could be y1, y2, y3 etc.... You can use any character reference prior to the equal sign = to define a line number. The important thing to remember is what you used to define the line number to use later in the formula.

Example: In line 3 (L3) of the formulas displayed on the following screen, L3 stores the amount calculated in line 2 L2 of the Formula. L1 is the amount retrieved for the table in line 1. (The table to be used with this Formula is assigned when the Withholding Code is set up when you set up the Tax Authority). These variables are used to calculate an amount in line 3 and then that amount is stored as the variable L3 and used in the if, then, else statement that gives us our tax amount.



Formula ID: POHPQLWH

Description: Piqua City Tax

Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2010	POHPQLWH	L1 = f.Table\	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

```
L1 = f.TableValue(1, 1)
L2 = f.GetNumericVariable("TAXEARN") -
f.GetNumericVariable("FIXEDEARN")
L3 =(L1 * L2) * .01
if L3 < 0:
  L4 = 0
else:
  L4 = L3
f.SetNumericVariable("CALCVALUE", L4)
```

```
#tax methods
f.GetNumericVariable(name)
f.SetNumericVariable(name, value)
f.GetTextVariable(name)
f.SetTextVariable(name, value)
f.GradientValue(value, column)
f.TableValue(row, column)
f.Earning(id)
f.EarningYTD(id)
f.Deduction(id)
f.DeductionYTD(id)
f.Withholding(id)
f.WithholdingYTD(id)
min(value, value)
max(value, value)
#text variables
EICCODE
```

NOTE: All line references do not need to be enclosed in brackets [ ].

Calculated variables

Use the calculated variables to bring in the calculated amount from predefined variables that are calculated within the Formula.

Below is a table with the commands and descriptions:

Variable	Description
CALC_COLUNCOLMED	Collected on Uncollected Medicare
CALC_COLUNCOLOASDI	Collected on Uncollected OASDI
CALC_FICATIPS	Recorded FICA tips
CALC_TAXABLE	Taxable earnings
CALC_TDW	Adjusted SUI withholdings
CALCVALUE	The final value calculated by the formula

Operations

Formulas are calculated line by line. Calculations in a line proceed according to the standard order of mathematical operations: numbers are multiplied, then divided, then added, and finally subtracted.

Below is a table of the valid operations and the order in which they are performed (1 = performed first, 5 = performed last).

Operator	Priority	Description
–	1	Negative Sign
^	2	Exponentiation
*****	3	Multiplication
/	3	Division
+	4	Addition
–	4	Subtraction

Operator	Priority	Description
=	5	Equal to
<>	5	Not equal to
>	5	Greater than
<	5	Less than
>=	5	Greater than or Equal to
<=	5	Less than or Equal to
==	5	Comparative value

Calculations are performed in the order listed in the table above, from left to right. To change the calculation order, use parentheses to group parts of the Formula together. Calculation is done from the innermost set of parentheses to the outermost set.

Using Formula Operation Commands

Formulas are calculated line by line. Calculations in a single line proceed according to the standard order of mathematical operations:

1. Numbers are multiplied
2. Numbers are divided
3. Numbers are added
4. Numbers are subtracted

Operations are performed in the order above from left to right in the Formula line. Calculations in parentheses are performed before all other calculations in the same order as listed earlier.

Using Formula Functions

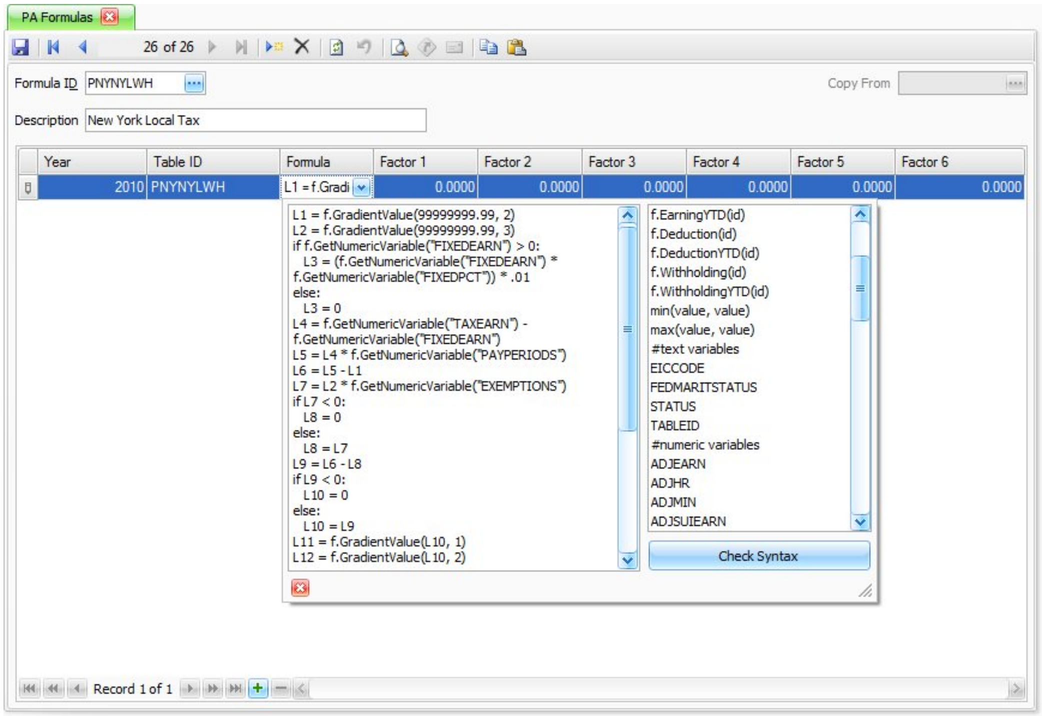
You can use three types of functions when constructing Formulas; the Tables Lookup function, the Conditionals function and MIN/MAX functions.

Tables Lookup Functions

Use the **f.TableValue(row, column)** and **f.GradientValue(value, column)** functions to look up values in Formula (tax) Tables. Use the **f.GradientValue** function to look up information in a graduated tax table that has multiple rows or lines in ascending order, like the FED__FWH and PMN__SWH tables. The **f.TableValue** function is used to look up information in a table that is not arranged with graduated multiple rows, like a PMN__SWH table. Use the **f.TableValue** and **f.GradientValue** functions to have the Formula look up values in Formula Tables that have been set up in the ST database (without the Show Company Specific Tax Tables box checked) and to have the Formula look up values in Formula Tables that have been set up in the company database (with the Show Company Specific Tax Tables box checked).

f.GradientValue Function

The format for the **f.GradientValue** command is **f.GradientValue(value, column)**; where value is the row (or line) in the tax table and column is the column number. The tax table read by the **f.GradientValue** function is specified in the Formula function on the Setup and Maintenance menu.



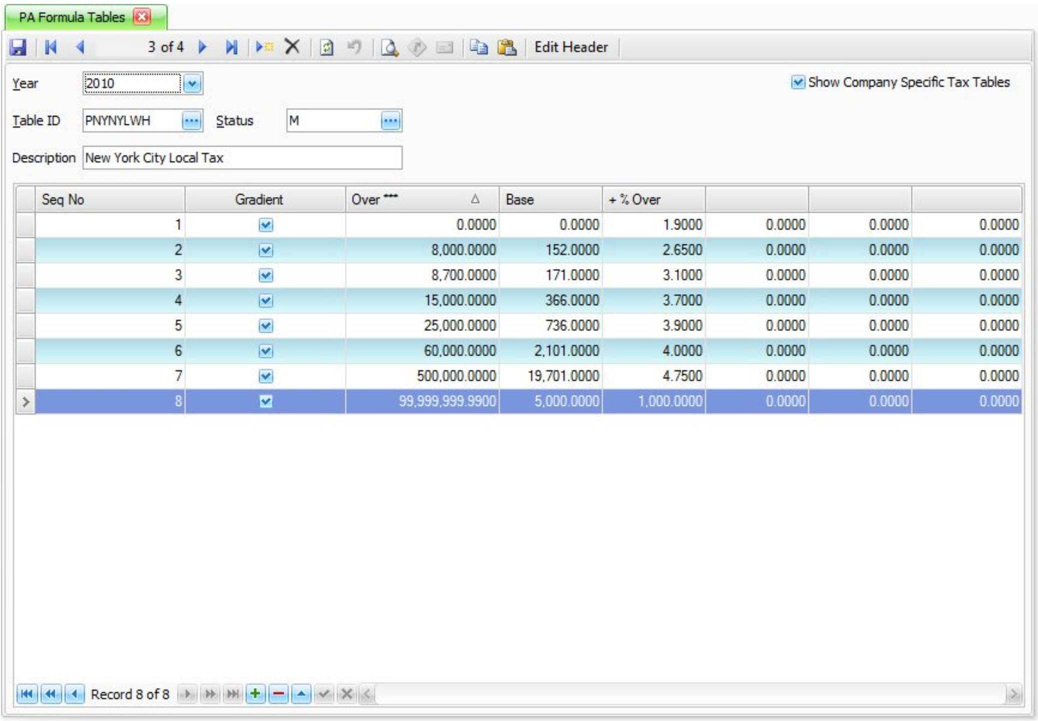
Formula ID PNYNYLWH

Description New York Local Tax

Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2010	PNYNYLWH	L1 = f.Gradi	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

```
L1 = f.GradientValue(99999999.99, 2)
L2 = f.GradientValue(99999999.99, 3)
if f.GetNumericVariable("FIXEDEARN") > 0:
  L3 = (f.GetNumericVariable("FIXEDEARN") *
f.GetNumericVariable("FIXEDPCT")) * .01
else:
  L3 = 0
L4 = f.GetNumericVariable("TAXEARN") -
f.GetNumericVariable("FIXEDEARN")
L5 = L4 * f.GetNumericVariable("PAYPERIODS")
L6 = L5 - L1
L7 = L2 * f.GetNumericVariable("EXEMPTIONS")
if L7 < 0:
  L8 = 0
else:
  L8 = L7
L9 = L6 - L8
if L9 < 0:
  L10 = 0
else:
  L10 = L9
L11 = f.GradientValue(L10, 1)
L12 = f.GradientValue(L10, 2)
```

Example: When the Withholding Code SWH for the state of Minnesota was set up in Codes Maintenance State Tax Setup. State and Federal Tax formulas in TRAVERSE are stored in the ST database. Do not modify Formulas in the ST database unless you are aware of what you are doing.



Year: 2010
☑ Show Company Specific Tax Tables
Table ID: PNYNYLWH
Status: M
Description: New York City Local Tax

Seq No	Gradient	Over ***	Base	+ % Over			
1	☑	0.0000	0.0000	1.9000	0.0000	0.0000	0.0000
2	☑	8,000.0000	152.0000	2.6500	0.0000	0.0000	0.0000
3	☑	8,700.0000	171.0000	3.1000	0.0000	0.0000	0.0000
4	☑	15,000.0000	366.0000	3.7000	0.0000	0.0000	0.0000
5	☑	25,000.0000	736.0000	3.9000	0.0000	0.0000	0.0000
6	☑	60,000.0000	2,101.0000	4.0000	0.0000	0.0000	0.0000
7	☑	500,000.0000	19,701.0000	4.7500	0.0000	0.0000	0.0000
8	☑	99,999,999.9900	5,000.0000	1,000.0000	0.0000	0.0000	0.0000

If the Formula used the command **f.GradientValue**(30000,2), the system looks at the first column of the specified tax table until it finds the first row with an amount greater than 30000. Then, it goes to the row immediately before that one and returns the value found in the second column of that row.

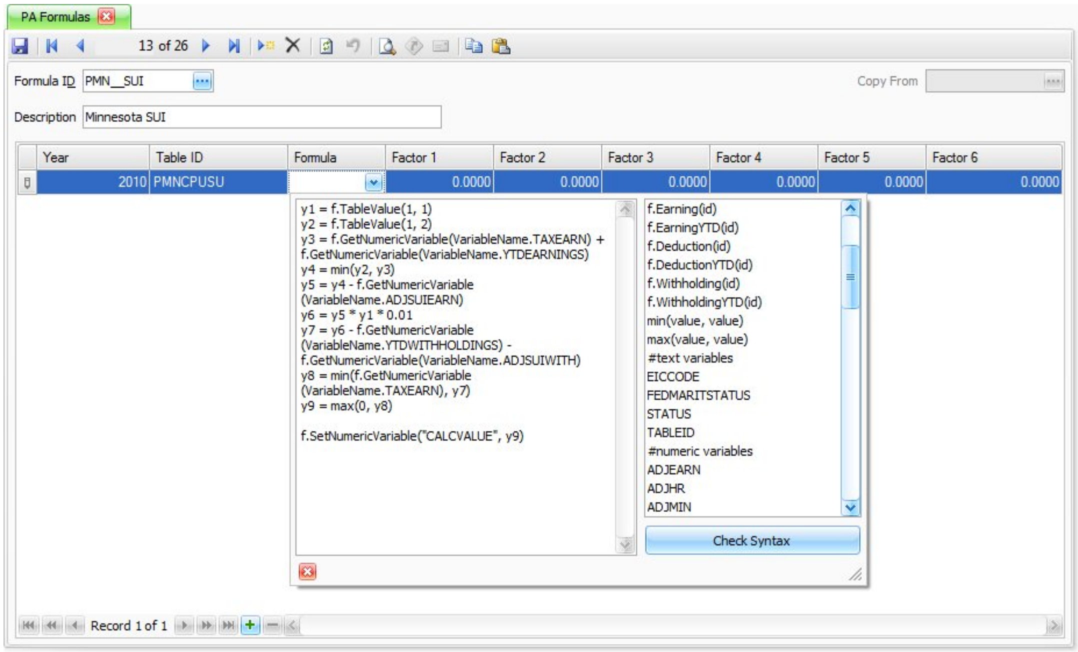
You can also use variable with the table lookup functions. You can, for example, use a variable calculated in a previous line of a Formula to find a row in the tax table. For example; in the command **f.GradientValue(L2,2)**, the system will read the tax table assigned to the Formula looking for the value greater than the amount stored in the variable L2.

NOTE: You can use only one f.GradientValue command per formula line.

f.TableValue Functions

Use the **f.TableValue** function to look up information in a tax table that is not arranged with graduated multiple rows, like the PMN__SUI table displayed on the screen.

The format for the **f.TableValue(row, column)**; where *row* is the row in the tax table and *column* is the column number.   The tax table to be used by the **f.TableValue** function is specified in the Formula function on the Setup and Maintenance menu when you setup the Formula ID.



Formula ID: PMN__SUI

Description: Minnesota SUI

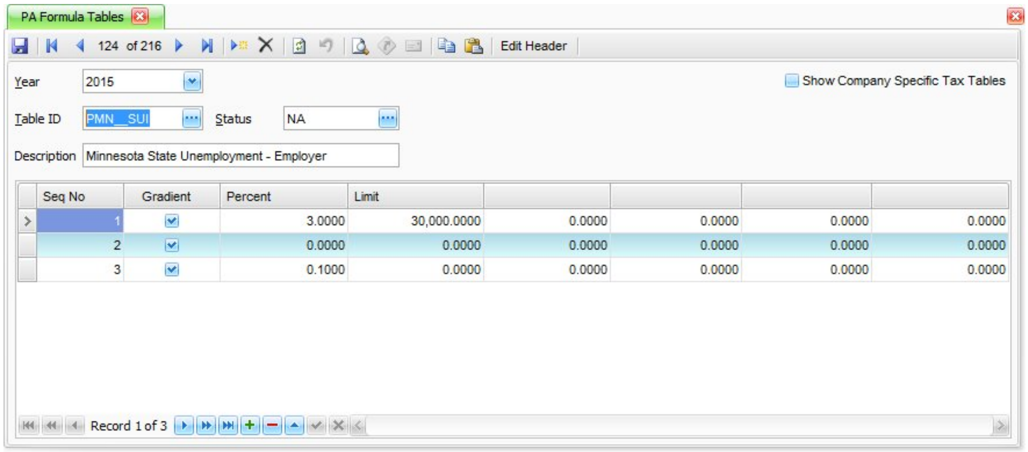
Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2010	PMNCPUSU		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

```
y1 = f.TableValue(1, 1)
y2 = f.TableValue(1, 2)
y3 = f.GetNumericVariable(VariableName.TAXEARN) +
f.GetNumericVariable(VariableName.YTDEARNINGS)
y4 = min(y2, y3)
y5 = y4 - f.GetNumericVariable
(VariableName.ADJSUIEARN)
y6 = y5 * y1 * 0.01
y7 = y6 - f.GetNumericVariable
(VariableName.YTDWITHHOLDINGS) -
f.GetNumericVariable(VariableName.ADJSUIWITH)
y8 = min(f.GetNumericVariable
(VariableName.TAXEARN), y7)
y9 = max(0, y8)

f.SetNumericVariable("CALCVALUE", y9)
```

```
f.Earning(id)
f.EarningYTD(id)
f.Deduction(id)
f.DeductionYTD(id)
f.Withholding(id)
f.WithholdingYTD(id)
min(value, value)
max(value, value)
#text variables
EICCODE
FEDMARITSTATUS
STATUS
TABLEID
#numeric variables
ADJEARN
ADJHR
ADJMIN
```

Example: When the Withholding Code SUI for the state of Minnesota was set up in the Formula function, the formula PMN__SUI and the table PMN__SUI were assigned to SUI. The command f.TableValue(1,2) is used in the Formula. The system returns 30,000, the value found in the first row, second column of the table PMN__SWH.



Year: 2015

☐ Show Company Specific Tax Tables

Table ID: PMN__SUI

Status: NA

Description: Minnesota State Unemployment - Employer

Seq No	Gradient	Percent	Limit				
1	☑	3.0000	30,000.0000	0.0000	0.0000	0.0000	0.0000
2	☑	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3	☑	0.1000	0.0000	0.0000	0.0000	0.0000	0.0000

NOTE: You can use only one f.TableValue command per formula line

NOTE: You must use the look function f.TableValue if you are adding a new Formula and Formula Table that currently is not in the included State and Federal tax tables and Formulas. (For example: Adding a Local Tax Formula or company specific SUI Formula and Table.)

Conditionals

You can use the **IF(*x, y, z*)** function to evaluate formulas conditionally. If condition *x* is true, then perform *y*, else perform *z*.

For example:

```
if L1 > f.GetNumericVariable(“FC1”):
   L3 = L2
else:
   L3 = 0.
```

If the value of L1 is greater than FC1, we return a value of L2; otherwise we return the value of 0.

NOTE: It is important to use a constant number of spaces as indentations for the lines following the if and else command lines. The example above uses 3 spaces as indentations. At the end of the if and else command lines you must use a colon: to indicate what the next condition is.

f.Deduction functions

If you want to bring in the value of a specific Deduction taken out of an Employee’s check you can use the f.Deduction functions. There are two f.Deduction functions.

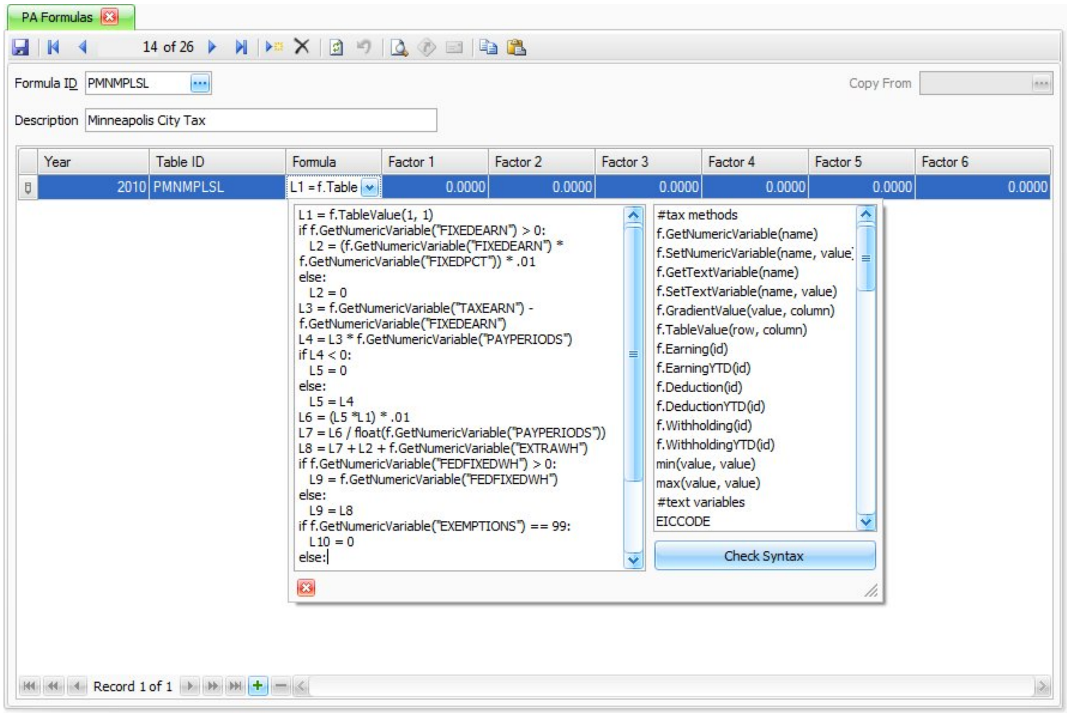
f.Deduction(dedID) - Returns the Deduction amount for the Deduction ID for the current check.

f.DeductionYTD(id) - Returns the year to date Deduction amount for the dedID.

When you are using the **f.Deduction** functions you do not put square brackets around the f.Deduction command. To return the value you must put the ID in parentheses and within quotes (“006”).

Using Conditional Functions

Use the If function, IF(x, y, z) to perform logical operations (x) and return (y) if (x) is true, or return (z) if (x) is false. You can also look at this as If (x) is true, then (y), else (z).



Formula ID: PMNMPLSL

Copy From:

Description: Minneapolis City Tax

Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2010	PMNMPLSL	L1 =f.Table	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

```
L1 = f.TableValue(1, 1)
if f.GetNumericVariable("FIXEDEARN") > 0:
  L2 = (f.GetNumericVariable("FIXEDEARN") *
f.GetNumericVariable("FIXEDPCT")) * .01
else:
  L2 = 0
L3 = f.GetNumericVariable("TAXEARN") -
f.GetNumericVariable("FIXEDEARN")
L4 = L3 * f.GetNumericVariable("PAYPERIODS")
if L4 < 0:
  L5 = 0
else:
  L5 = L4
L6 = (L5 *L1) * .01
L7 = L6 / float(f.GetNumericVariable("PAYPERIODS"))
L8 = L7 + L2 + f.GetNumericVariable("EXTRAWH")
if f.GetNumericVariable("FEDFIXEDWH") > 0:
  L9 = f.GetNumericVariable("FEDFIXEDWH")
else:
  L9 = L8
if f.GetNumericVariable("EXEMPTIONS") == 99:
  L10 = 0
else:
```

```
#tax methods
f.GetNumericVariable(name)
f.SetNumericVariable(name, value
f.GetTextVariable(name)
f.SetTextVariable(name, value)
f.GradientValue(value, column)
f.TableValue(row, column)
f.Earning(id)
f.EarningYTD(id)
f.Deduction(id)
f.DeductionYTD(id)
f.Withholding(id)
f.WithholdingYTD(id)
min(value, value)
max(value, value)
#text variables
EICCODE
```

MIN/MAX functions

If you wish to test the values of two variables, *a* and *b*, and have the larger value returned, you can use the function **MAX(*value*,*value*)**. Conversely, if you wish to test the values of two variables, *a* and *b*, and have the lesser value returned, you can use the function **MIN(*value*,*value*)**.

f.SetNumericVariable

To return a value from the Formula you must use the f.SetNumericVariable command as the last line of the Formula. The syntax for the f.SetNumericVariablef.SetNumericVariable(name, value), where name is the name of the value to return and value is the value you want to return, would be as follows:

f.SetNumericVariable("CALCVALUE", L4)

The command usually will include the variable CALCVALUE to return the value that has been calculated in the formula and the L4 which is the last line in the formula.

Formula Tips

- When setting up your Formulas, you don’t need to type all the commands and variables into the Formula. With the Formula box open you can double click on the command and variable in the right column to bring the value double clicked on into the Formula, and then add quotes and other text to the Formula manually. The value double clicked on will be brought into the Formula at the point of the cursor.
- If your Formula might result in a negative amount calculated use a line at the end of the Formula similar to this:

```
if L3 < 0:
   L4 = 0
else:
   L4 = L3
```

This formula line says if line 3 is less than zero then return 0 or else return the value from line 3.

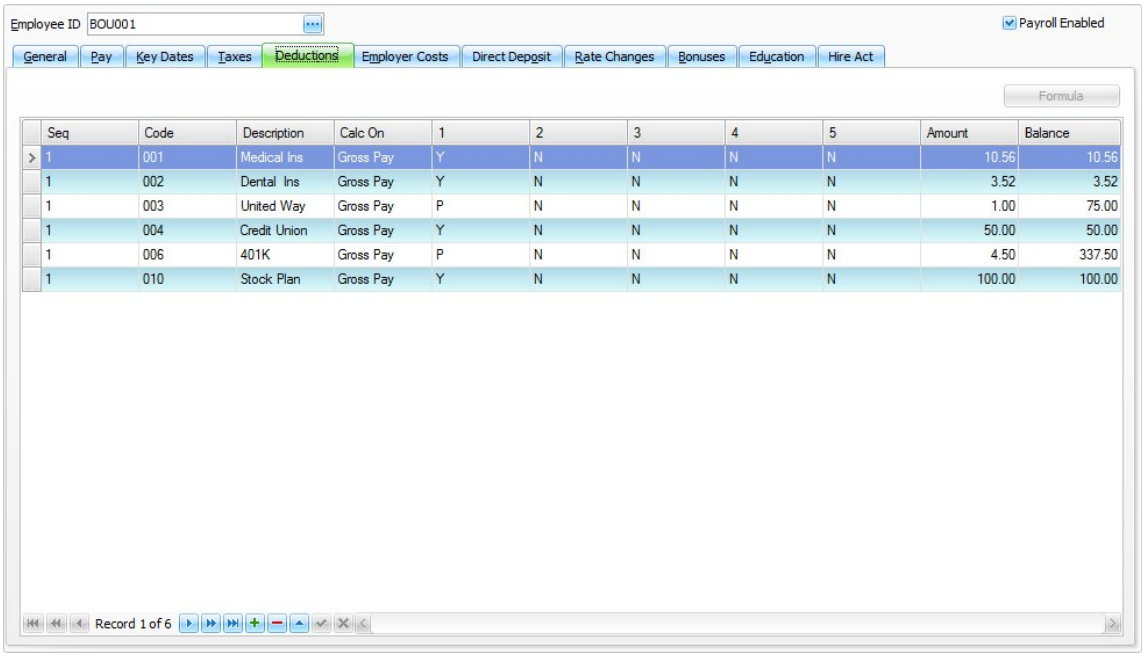
- If you are going to be using variable amounts that could change for each Employee, use the Factor fields and use the (“FCx”) function (x being the value 1 to 6 corresponding to the factor fields).
- When using the conditional if: else: make sure to put a colon after the if statement and after the else command. When defining the then, the line after the if, and when defining the else, the line after the else, be consistent with the number of spaces used to indent the conditional values. In this chapter 3 spaces are used for each of the if, then, else conditional statements.
- When using the #TAX METHODS commands the variable, and factors must be enclosed in quotation marks within the parentheses (“ADJEARN”).
- When bringing in line number values into another line the line numbers are not enclosed in parentheses or quotation marks. For example: L9 = L6 - L8
- Line numbers can be defined in many different ways within the Formulas. The line number is defined by putting an equal sign after the value you want to use for the line number. For example: **L1 =** would be defining line 1. **y1 =** would also be defining line 1 but using the y as a reference. When only one line is used you can use a character reference such as **res =** to define the line number as the result.
- The last line in each Formula must contain the **f.SetNumericVariable** command to return the value calculated. For example: **f.SetNumericVariable(“CALCVALUE”, L4)** is saying set the numeric variable to the calculated value from line 4, which is the last line defined in the Formula.

FORMULA EXAMPLES

Deduction Examples

The following examples demonstrate how to use different functions in the Payroll system to set up Deductions calculated by Formulas. There is more than one way to set up these Deductions to calculate correctly. The setups used in the examples are chosen to give you an overview of how the different functions in the Payroll system can work together to calculate Deductions.

When the system calculates Deductions, Deductions set up to be calculated on Gross Pay in the Deductions/Employer Costs function on the Code Maintenance menu. Deductions are calculated first in the order of the Seq number for the pay period in the Deductions tab in the Employee Information function.



Employee ID BOU001

☑ Payroll Enabled

Seq	Code	Description	Calc On	1	2	3	4	5	Amount	Balance
1	001	Medical Ins	Gross Pay	Y	N	N	N	N	10.56	10.56
1	002	Dental Ins	Gross Pay	Y	N	N	N	N	3.52	3.52
1	003	United Way	Gross Pay	P	N	N	N	N	1.00	75.00
1	004	Credit Union	Gross Pay	Y	N	N	N	N	50.00	50.00
1	006	401K	Gross Pay	P	N	N	N	N	4.50	337.50
1	010	Stock Plan	Gross Pay	Y	N	N	N	N	100.00	100.00

After the Gross Pay Deductions are calculated, the withholdings are calculated.

Deductions set up to be calculated on Net Pay are Deducted last and are Deducted in the reverse order of the sequence number on the Deductions tab. For Example: If the pay check was not large enough to have all the Withholdings and Deductions taken out and have an amount to print on the check, the Deduction calculation will look to the Deduction setup for the Employee and remove the Deduction with the largest sequence number. If the check now has enough to cover the Withholdings and Deductions it will print a check for 0 dollars and not include the Stock Plan as a Deduction.

If a garnished Deduction must be calculated after certain other Deductions are taken, then the Deductions that can be taken prior to the garnishment Deduction should be set up as Gross Pay Deductions. Set up the garnishment Deduction as a Net Pay Deduction.

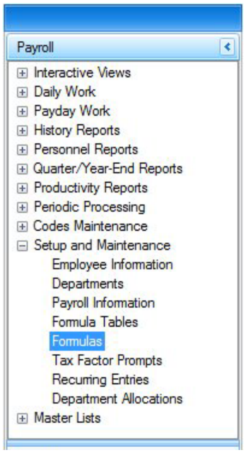
If there are Deductions that can be calculated *after* the garnishment is deducted, they should be set up as calculated on Net Pay and should have a sequence number *larger than* the garnishment Deduction in the Scheduled Deduction section of the Employee Information screen on the Deductions tab.

CREATING A MAXIMUM NET PAY GARNISHMENT DEDUCTION

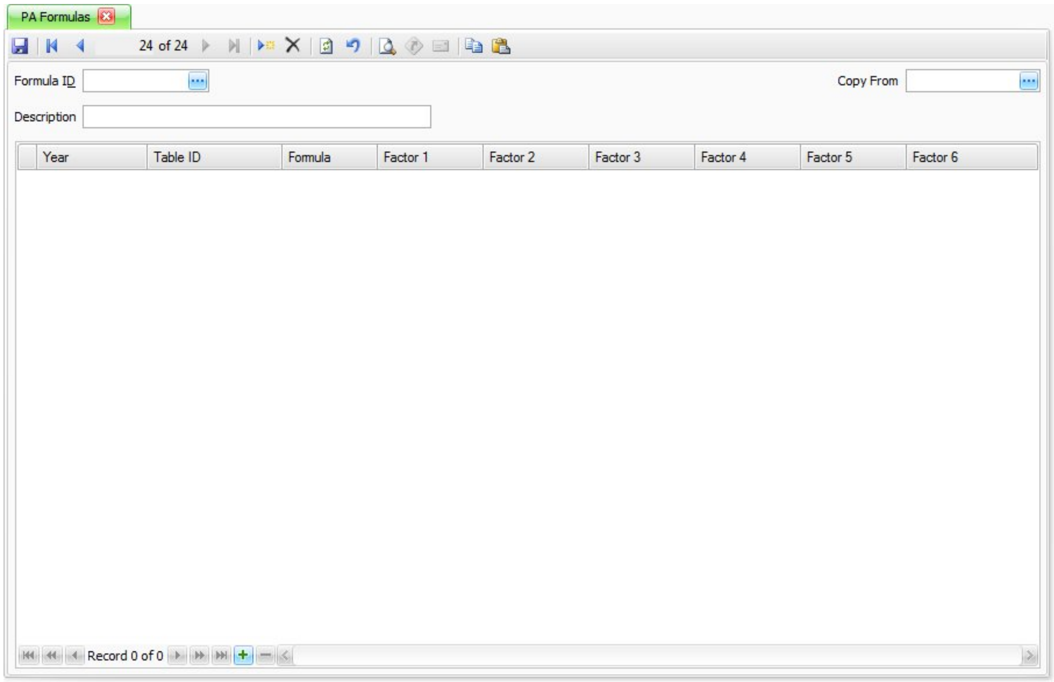
This garnishment Deduction should be calculated so that the maximum net pay the Employee ever receives is $300. All other scheduled Deductions for the Employee can be deducted prior to calculating this garnishment. The difference between the Gross Pay, less Withholdings and Deductions, and the end result of a $300 check is the amount of the garnishment Deduction. If the Gross Pay less Withholdings and Deductions is less than $300, then the garnishment Deduction is 0.

> **Example: For an Employee: if gross pay = $1,000, medical insurance Deduction = $20, State and Federal Withholdings = $80, then net pay = $900. This garnishment Deduction allows for the Employee's new pay to be a maximum of $300, so the amount of the garnishment Deduction would be $600 for this pay period.**

1. To set up the Deduction, enter the Formula for the Deduction, select **Formula** from the **Setup and Maintenance** menu.

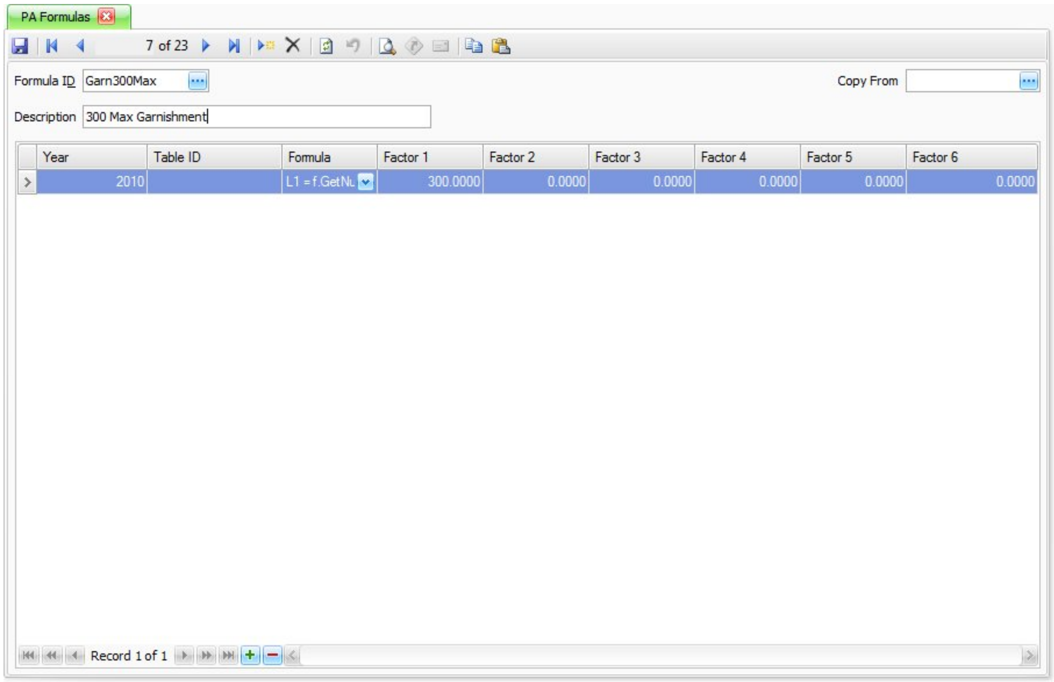


2. The Formulas screen is displayed. Select the **New Record** icon ▶. A blank Formulas screen is displayed.



3. Enter the **Formula ID**.

4. If you want to copy the Formula for this Deduction from an existing Formula you can enter the Formula ID or select the Formula ID from the Formulas window. We are not copying the Formula from an existing one, so tab to the **Description** field, and enter a Description for the Formula.



Formula ID Garn300Max

Copy From

Description 300 Max Garnishment

Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2010		L1 = f.GetNu	300.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Record 1 of 1

5. Click the green plus (+) sign at the bottom of the screen to add a new Formula for the current payroll Year.

6. To make the Formula as flexible as possible, use Factor 1 to store the value of the maximum amount of new pay, 300. Then, if we have more than one Employee subject to this type of garnishment or if the maximum net pay for an Employee changes, the net pay maximum (Factor 1) can be overridden on an individual Employee basis without changing the lines of the Formula.

7. Click the drop down list arrow in the **Formula** column to display the Formula box, into which to enter the formula.

8. We will use the Conditional function for this Formula. The variable ADJEARN stores the adjusted earnings for net pay Deductions (Gross Earnings minus Withholdings and gross pay Deductions at the time of calculating this net pay Deduction). Enter L1 as:

    L1 = f.GetNumericVariable("ADJEARN")

    This will bring in the adjusted earnings for the Employee.

NOTE: You can bring in the f.GetNumericVariable by double clicking on it in the window on the right side of the Formula box. You can then enter a quotation mark "within the parentheses and double click ADJEARN in the box on the right and close the quotation ".

9. Enter line 2 as:

   L2 = f.GetNumericVariable("ADJEARN") - f.GetNumericVariable("FC1")

   This will calculate the difference between the adjusted earnings and the Maximum amount the payroll check can be for this garnishment from Factor 1.

10. Enter the if condition as follows:

```
if L1 > f.GetNumericVariable("FC1"):
   L3 = L2
else:
   L3 = 0
```

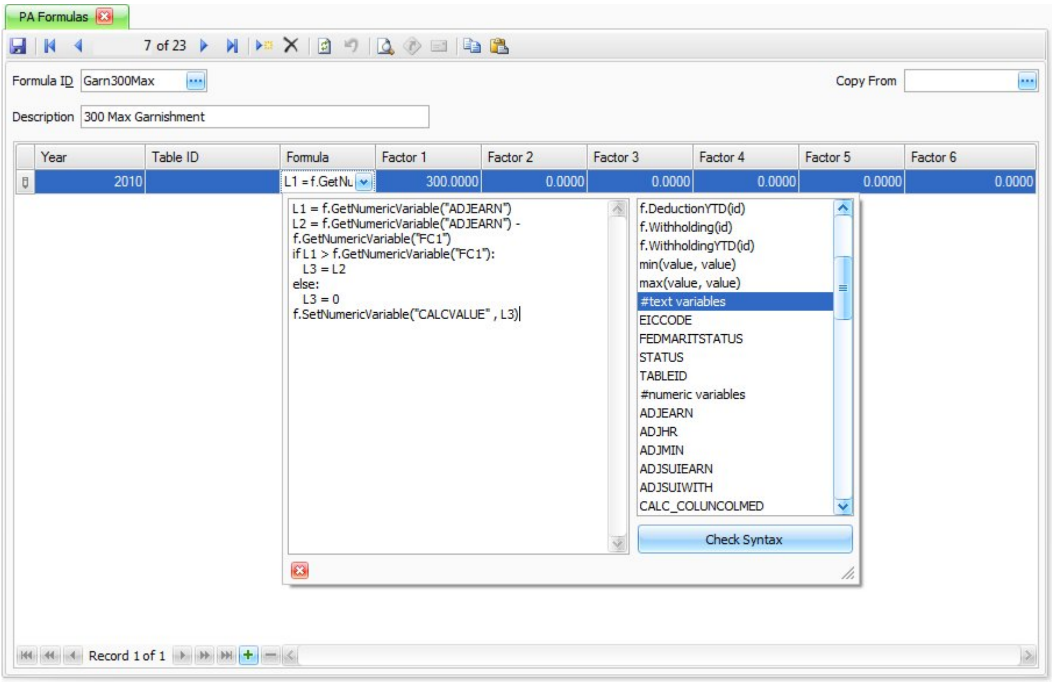
   This is saying if line 1 their adjusted Earnings is greater than the amount in Factor 1 use that amount, if that amount is greater than the Factor use the calculation in Line 2. Or else the value is 0 if Line 1 is less then Factor 1.

NOTE: It is important to use a constant number of spaces as indentations for the lines following the if and else command lines. The example above uses 3 spaces as indentations. At the end of the if and else command lines you must use a colon: to indicate what the next condition is.

11. Enter the following to return the calculated value:

    f.SetNumericVariable("CALCVALUE" , L3)

    This will return the value calculated in line 3, which is the if, then, else conditional statement.



Formula ID Garn300Max

Copy From

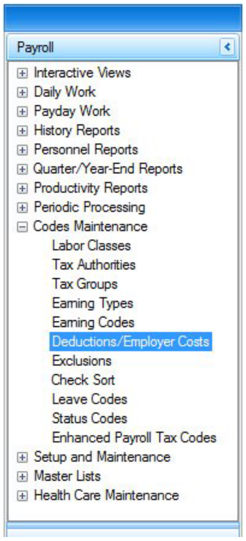
Description 300 Max Garnishment

Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2010		L1 = f.GetNu	300.0000	0.0000	0.0000	0.0000	0.0000	0.0000

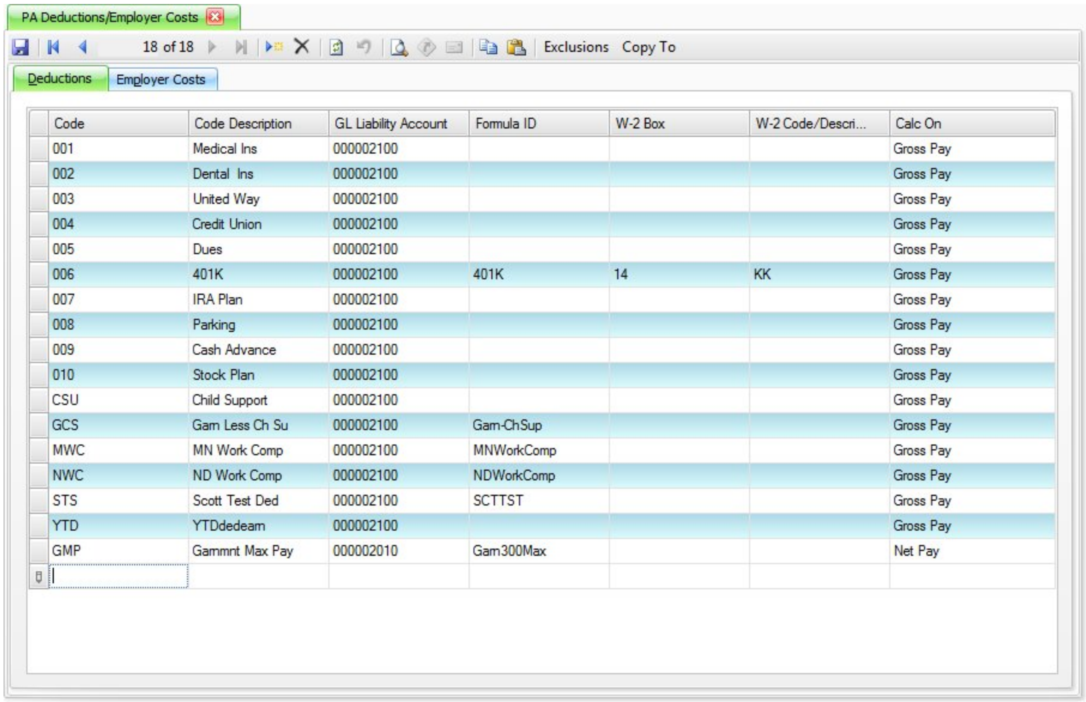
```
L1 = f.GetNumericVariable("ADJEARN")
L2 = f.GetNumericVariable("ADJEARN") -
f.GetNumericVariable("FC1")
if L1 > f.GetNumericVariable("FC1"):
  L3 = L2
else:
  L3 = 0
f.SetNumericVariable("CALCVALUE" , L3)
```

12. Click the **Save** button to save the Formula. Open the Formula entry box and click **Check Syntax** to verify the syntax is correct for this Formula.

13. Select the **Deductions/Employer Costs** function on the **Code Maintenance** menu.



14. The **Deductions/Employer Costs** screen is displayed.



Code	Code Description	GL Liability Account	Formula ID	W-2 Box	W-2 Code/Descri...	Calc On
001	Medical Ins	000002100				Gross Pay
002	Dental Ins	000002100				Gross Pay
003	United Way	000002100				Gross Pay
004	Credit Union	000002100				Gross Pay
005	Dues	000002100				Gross Pay
006	401K	000002100	401K	14	KK	Gross Pay
007	IRA Plan	000002100				Gross Pay
008	Parking	000002100				Gross Pay
009	Cash Advance	000002100				Gross Pay
010	Stock Plan	000002100				Gross Pay
CSU	Child Support	000002100				Gross Pay
GCS	Gam Less Ch Su	000002100	Gam-ChSup			Gross Pay
MWC	MN Work Comp	000002100	MNWorkComp			Gross Pay
NWC	ND Work Comp	000002100	NDWorkComp			Gross Pay
STS	Scott Test Ded	000002100	SCTTST			Gross Pay
YTD	YTDdedeam	000002100				Gross Pay
GMP	Gammnt Max Pay	000002010	Gam300Max			Net Pay

15. Select the **New Record** icon from the toolbar to add the Deduction to the Payroll system.

16. Enter the **Deduction Code** and a **Description** for the garnishment Deduction.

17. Enter the **Liability Account** used for this Deduction. This is the Account that will be credited when you post checks.

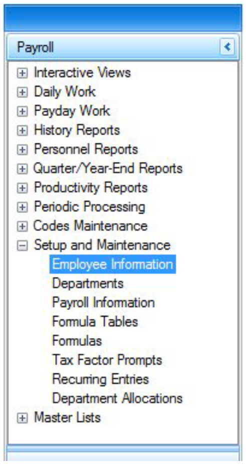
18. Select the **Formula ID** we just entered.

19. Select **Net Pay** on which to calculate the Deduction.

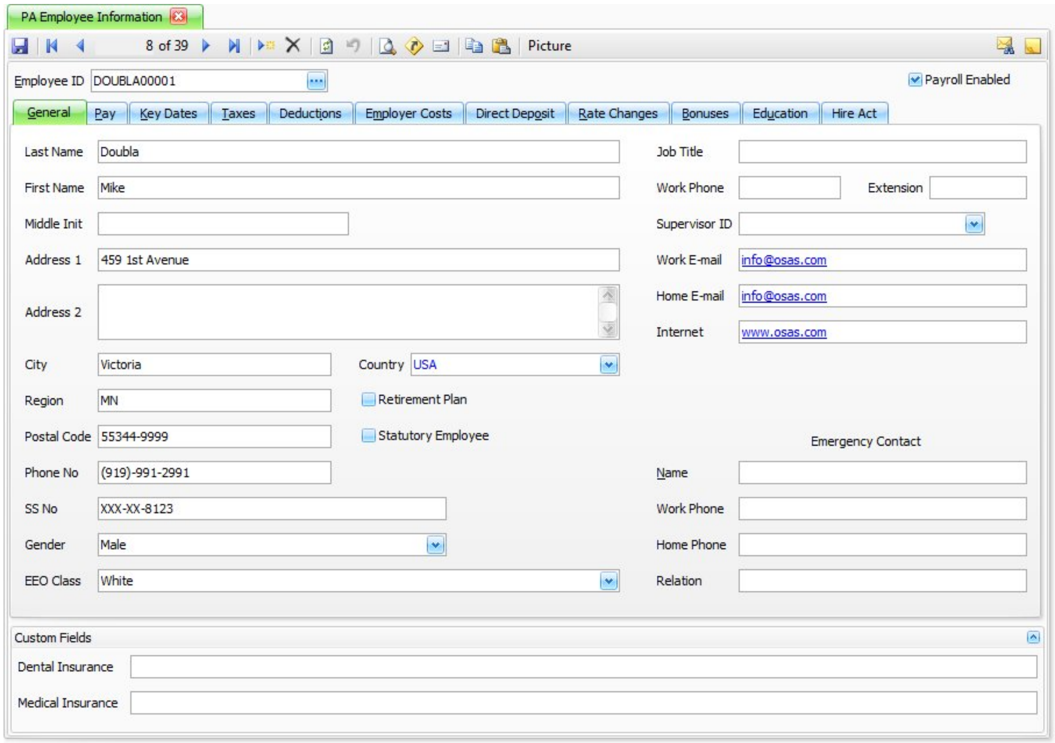
    This Deduction should not print on the W-2, so leave the **W-2** Field and **W-2 Code/Description** fields blank.

20. Click the **Save** button 💾 and close the screen when you are finished.

21. To add this Deduction for an Employee, select the **Employee Information** function on the **Setup and Maintenance** menu.



22. The **Employee Information** screen is displayed.



PA Employee Information

8 of 39 Picture

Employee ID DOUBLA00001

☑ Payroll Enabled

General | Pay | Key Dates | Taxes | Deductions | Employer Costs | Direct Deposit | Rate Changes | Bonuses | Education | Hire Act

Last Name	Doubla	Job Title	
First Name	Mike	Work Phone	Extension
Middle Init		Supervisor ID	
Address 1	459 1st Avenue	Work E-mail	info@osas.com
Address 2		Home E-mail	info@osas.com
		Internet	www.osas.com
City	Victoria	Country	USA
Region	MN	□ Retirement Plan	
Postal Code	55344-9999	□ Statutory Employee	
Phone No	(919)-991-2991		
SS No	XXX-XX-8123		
Gender	Male		
EEO Class	White		

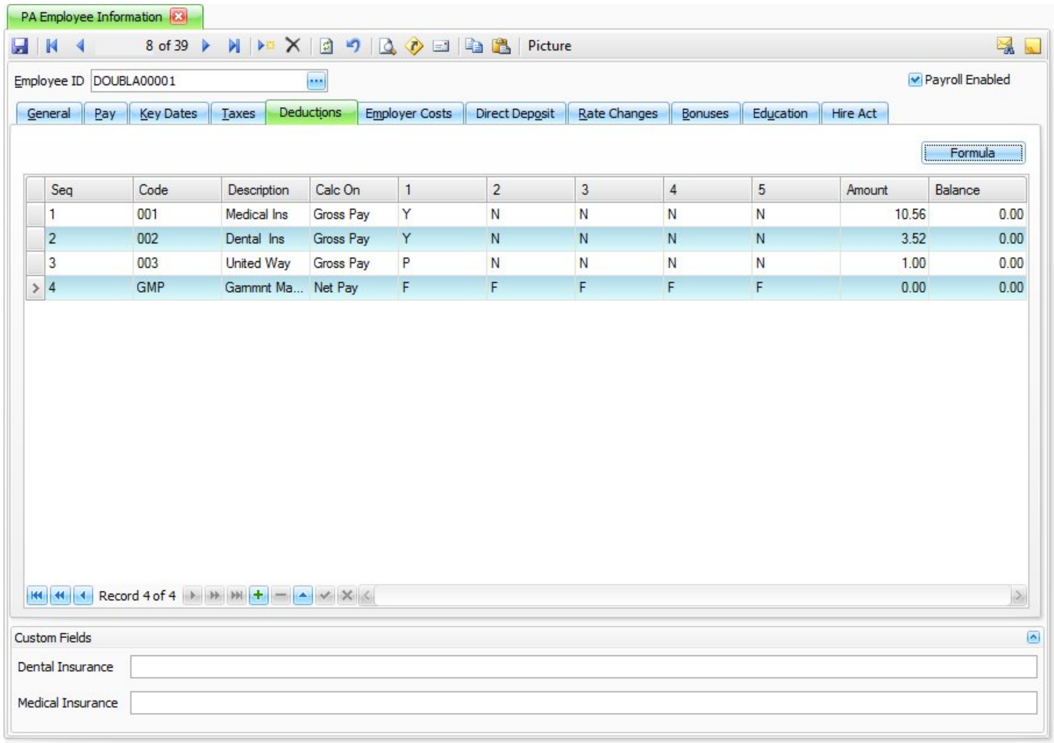
Emergency Contact

Name	
Work Phone	
Home Phone	
Relation	

Custom Fields

Dental Insurance	
Medical Insurance	

23. Enter the **Employee ID** of the Employee to which you want to add the deduction, and select the **Deductions** tab.



Employee ID DOUBLA00001

☑ Payroll Enabled

Seq	Code	Description	Calc On	1	2	3	4	5	Amount	Balance
1	001	Medical Ins	Gross Pay	Y	N	N	N	N	10.56	0.00
2	002	Dental Ins	Gross Pay	Y	N	N	N	N	3.52	0.00
3	003	United Way	Gross Pay	P	N	N	N	N	1.00	0.00
4	GMP	Gammnt Ma...	Net Pay	F	F	F	F	F	0.00	0.00

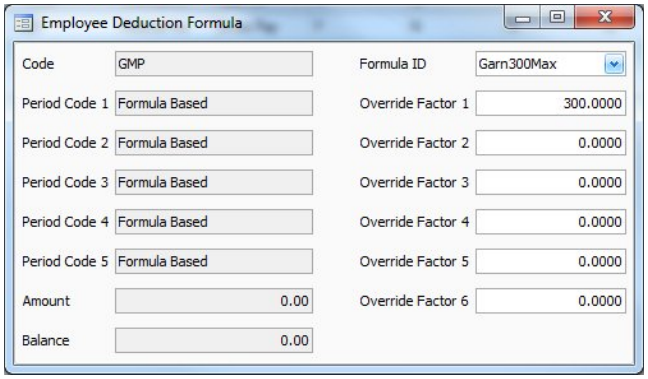
Dental Insurance

Medical Insurance

24. Use the mouse to scroll down to an empty record or click the green plus (+) sign at the bottom of the screen to add this garnishment Deduction to the Employee's scheduled Deductions. Since this net Deduction should be calculated after all other net Deductions, enter it as the last Deduction for the pay period. Select the **Code** for the Deduction.

25. The **Description** defaults from the Deduction Code. In the 1, 2, 3, 4, and 5 fields, enter an **F** for Formula in the pay period column you want to have the Deduction calculated. **N** will default in the other pay period columns. Leave the **Amount** and **Balance** fields zero.

If the garnishment for this Employee allows their net pay to be an amount other than $300, you will need to override Factor 1 for this Employee. In our example, Mike Doubla is subject to this garnishment Deduction but can receive a net pay amount not to exceed $450. To override the Factors for this Formula, select the **Formula** button and use the **Employee Deduction Formula** dialog box.



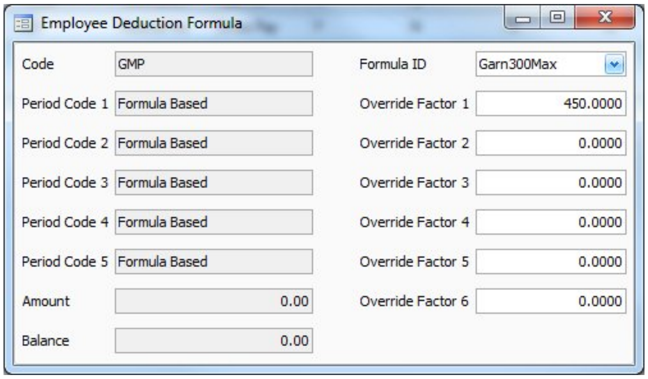
Employee Deduction Formula

Code	GMP	Formula ID	Garn300Max
Period Code 1	Formula Based	Override Factor 1	300.0000
Period Code 2	Formula Based	Override Factor 2	0.0000
Period Code 3	Formula Based	Override Factor 3	0.0000
Period Code 4	Formula Based	Override Factor 4	0.0000
Period Code 5	Formula Based	Override Factor 5	0.0000
Amount	0.00	Override Factor 6	0.0000
Balance	0.00		

The Factor Entry window appears. Select the **Formula ID** and the appropriate **Override Factors** for this Employee. Since the Formula for this deduction only uses **Factor 1** that is the only Factor we need to enter.

NOTE: If a Formula uses more than one factor but you only need to override one of them, you must still enter all of the Factor values in the Employee Deduction Formula dialog box. Otherwise the other Factors will be set to 0, the default amount.

26. Enter 450 for **Override Factor 1**. When this Deduction is calculated for Linda Bourne, the Formula will use 450 as the value of **Override Factor 1**. If this garnishment Deduction is used for other Employees, the value of Factor 1 for the Formula is still 300, unless an Override Factor is enter for that Employee.



Employee Deduction Formula

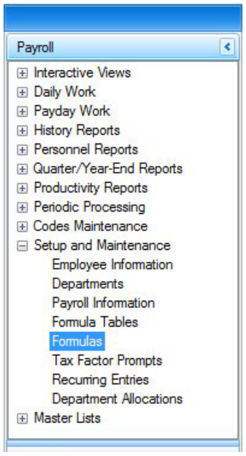
Code	GMP	Formula ID	Garn300Max
Period Code 1	Formula Based	Override Factor 1	450.0000
Period Code 2	Formula Based	Override Factor 2	0.0000
Period Code 3	Formula Based	Override Factor 3	0.0000
Period Code 4	Formula Based	Override Factor 4	0.0000
Period Code 5	Formula Based	Override Factor 5	0.0000
Amount	0.00	Override Factor 6	0.0000
Balance	0.00		

CREATING A POST-DEDUCTION GARNISHMENT DEDUCTION

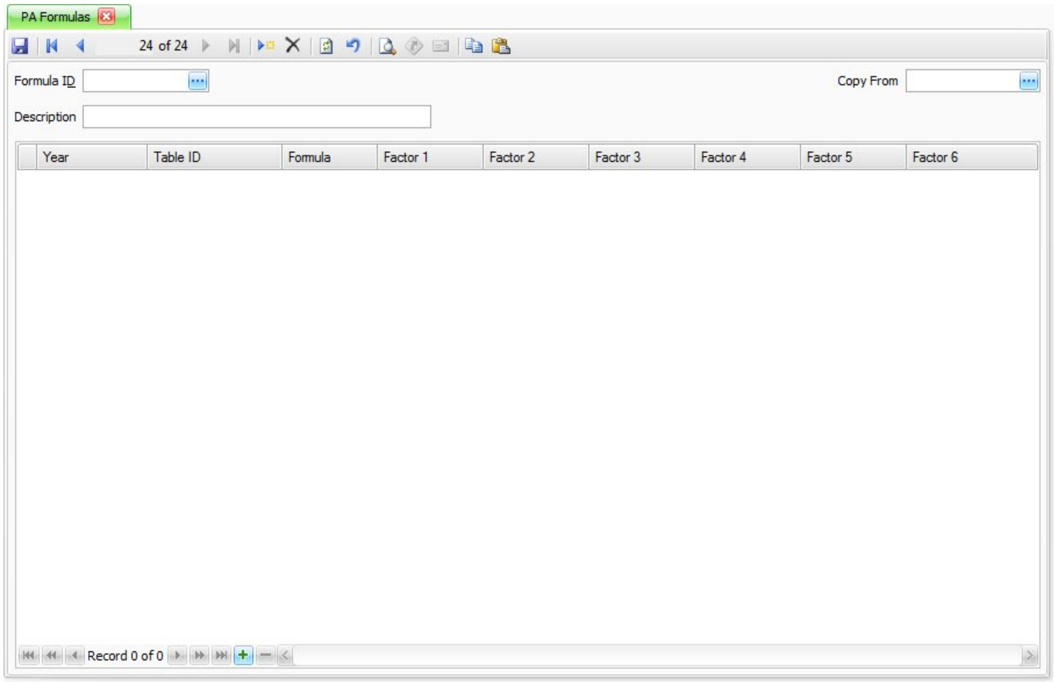
This garnishment Deduction is calculated after all other Deductions and Withholdings and requires that the maximum amount of the Deduction is $200, but the amount of the Deduction should never exceed more than 25% of the employee's net pay.

> **Example: For an employee: if gross pay = $1,000, medical insurance Deduction = $20, State and Federal Withholdings = $80, then net pay = $900. This garnishment Deduction allows the maximum amount of the Deduction to be $200, but should not exceed 25% of the net pay amount. 25% of $900 = $225, so the amount of this garnishment should be $200, the maximum.**

1. Select **Formulas** from the **Setup and Maintenance** menu to enter the Formula for this Deduction.

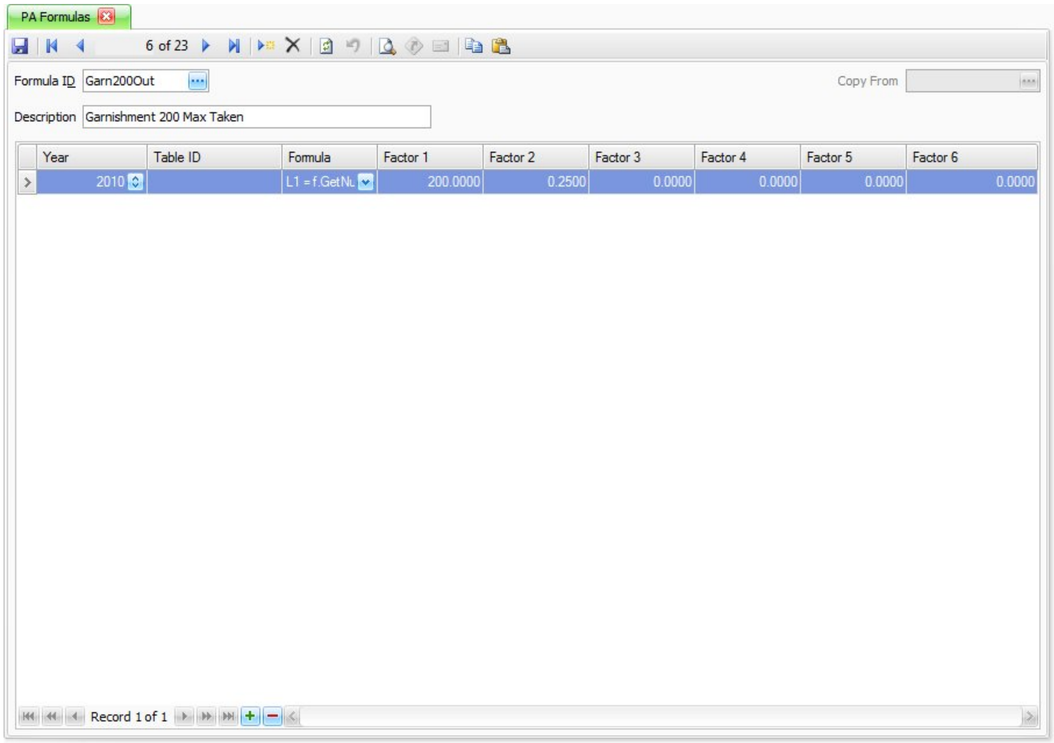


2. The **Formulas** screen is displayed



3. Select the **New Record** icon ▶ . A blank Formulas screen is displayed.
4. Enter the **Formula ID** for the new Formula we are entering.
5. Enter a **Description** for the Formula. We are not copying this Formula from an existing one.
6. Click the green plus (+) sign at the bottom of the screen to add a new Formula for the current payroll Year.

7. To make this Formula as flexible as possible, set up two Factors. **Factor 1** is set up equal to the maximum amount of the Deduction, $200. The maximum percent of net pay, in decimal format, allowed for the Deduction is the value of **Factor 2**, .25.



PA Formulas

6 of 23

Formula ID Garn200Out

Copy From

Description Garnishment 200 Max Taken

Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2010		L1 = f.GetNu	200.0000	0.2500	0.0000	0.0000	0.0000	0.0000

Record 1 of 1

8. Click the drop down list arrow in the **Formula** column to display the Formula box, into which to enter the Formula.

9. Enter line 1 as:

   L1 = f.GetNumericVariable(“ADJEARN”)

   This will bring in the adjusted earnings for the Employee.

NOTE: You can bring in the f.GetNumericVariable by double clicking on it in the window on the right side of the Formula box. You can then enter a quotation mark “ within the parentheses and double click ADJEARN in the box on the right and close the quotation “.

10. Enter line 2 as:

    L2 = f.GetNumericVariable(“FC1”)

    This will bring the value for Factor 1 into the Formula.

11. Enter line 3 as:

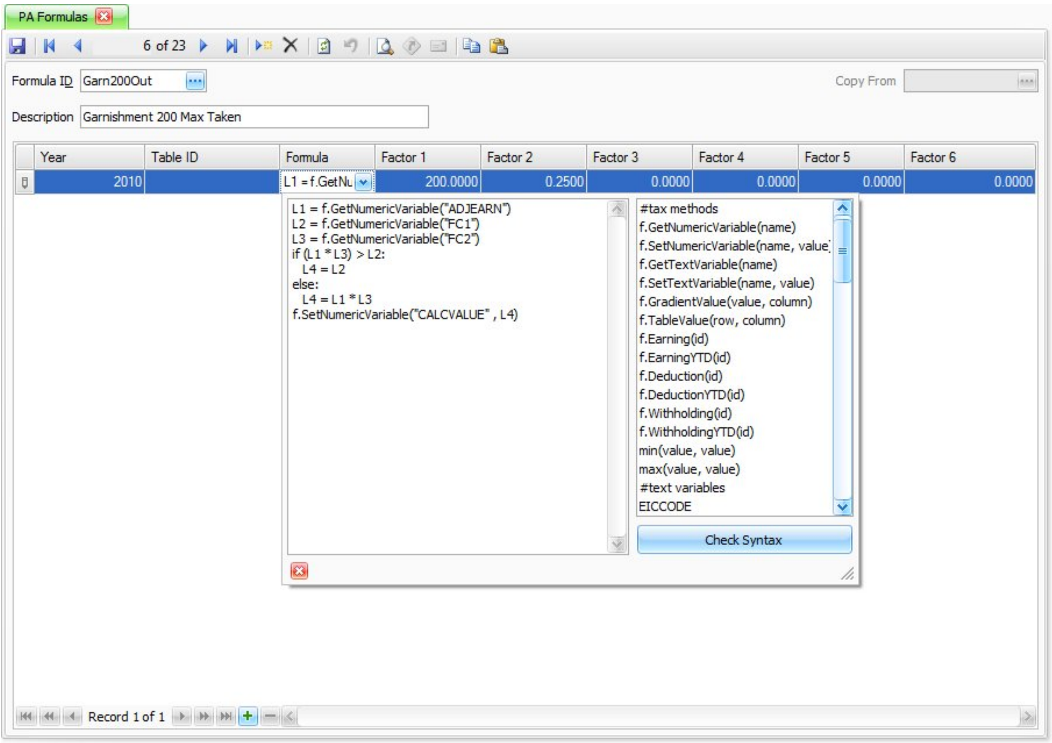
    L3 = f.GetNumericVariable(“FC2”)

    This will bring the value for Factor 2 into the Formula.

12. Now we will do the conditional statement. Enter the next section which will be line 4 as follows:

```
if (L1 * L3) > L2:
  L4 = L2
else:
  L4 = L1 * L3
```

If 25% (L3) of ADJEARN (Gross earnings minus Withholdings and gross pay Deductions at the time this Deduction is calculated) is greater than $200 (L2), then the Deduction amount will be $200 (L2). If 25% (L3) of ADJEARN is less than $200 (L2), then the amount of the Deduction is 25% (L3) of ADJEARN.



Formula ID: Garn200Out

Copy From:

Description: Garnishment 200 Max Taken

Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2010		L1 =f.GetNu	200.0000	0.2500	0.0000	0.0000	0.0000	0.0000

```
L1 = f.GetNumericVariable("ADJEARN")
L2 = f.GetNumericVariable("FC1")
L3 = f.GetNumericVariable("FC2")
if (L1 * L3) > L2:
  L4 = L2
else:
  L4 = L1 * L3
f.SetNumericVariable("CALCVALUE" , L4)
```

13. Enter the following to return the calculated value:

    f.SetNumericVariable(“CALCVALUE” , L4)

14. Set up the Deduction using the **Deduction/Employer Costs** function on the **Codes Maintenance** menu. This is not an Employer Cost paid deduction or deferred compensation. This garnishment Deduction is calculated on net pay.

15. To add the Deduction for an Employee, use the **Employee Information** function on the **Setup and Maintenance** menu. Enter the **Employee ID** and select the **Deductions** tab. Add this Deduction to the Employee's scheduled Deductions. If you need to override either or both of the Formula factors, select **Formula**.

NOTE: Remember that, if you need to override any of the factors for an Employee, both factors must be entered in the Employee Deduction Formula window.

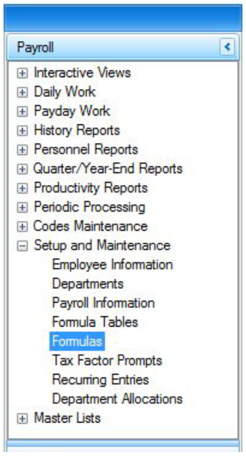
For details on setting up the Deduction for Employees, see the Creating a Maximum Net Pay Garnishment Formula section (page 9-25).

CREATING A 20% OF NET PAY GARNISHMENT

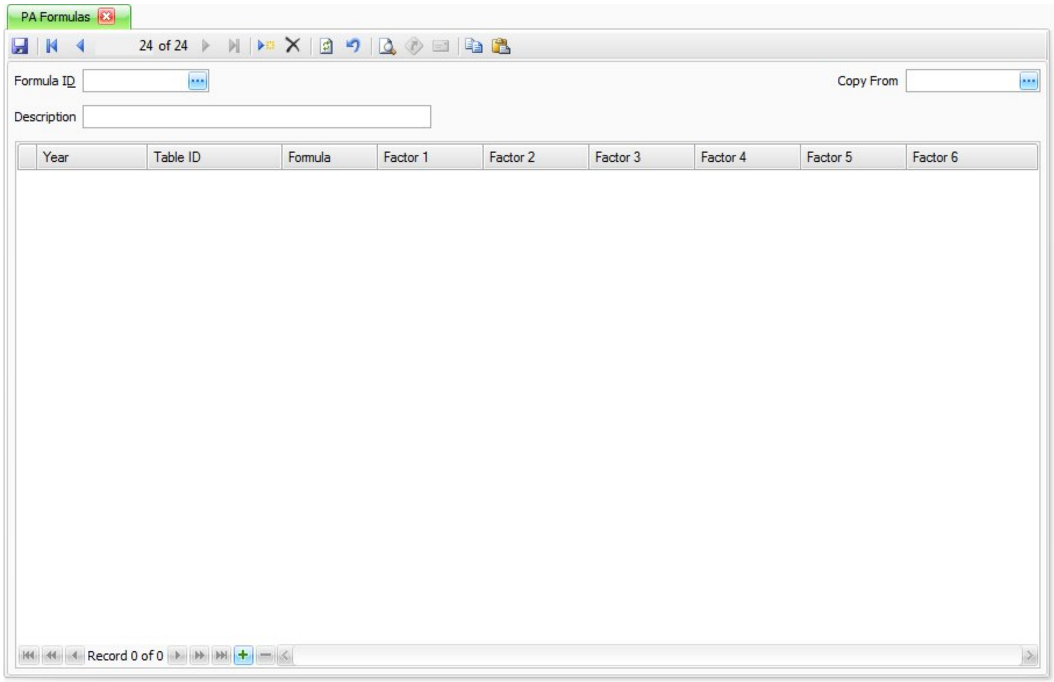
This garnishment Deduction is 20% of the Employee's net pay after all Deductions have been taken.

> **Example: For an Employee: if gross pay = $1,000, medical insurance Deduction =$20, State and Federal Withholdings = $80, then net pay = $900. This garnishment deduction is calculated as 20% or the net pay, $180.**

1. Select **Formulas** from the **Setup and Maintenance** menu.



2. The **Formulas** screen is displayed. Select the **New Record** icon ▶✳. A blank Formulas screen is displayed.



3. Enter the **Formula ID** and **Description**.
4. Click the green plus (+) sign at the bottom of the screen to add a new Formula for the current payroll Year.
5. Leave the **Table ID** blank.
6. Enter the percent of net pay, **.20**, in **Factor 1**.
7. Click the drop down list arrow in the **Formula** column to display the Formula box, into which to enter the Formula.
8. Enter line 1 as below, where ret is the line ID.

   ret = f.GetNumericVariable(“ADJEARN”) * f.GetNumericVariable(“FC1”)

9. Enter the next line as

   f.SetNumericVariable(“CALCVALUE”, ret)

   To return the value calculated in the first line.

To have this Deduction calculate after all other Deductions remember:

> Any Deductions set up as calculated on gross pay are calculated first and in the order they appear in the Scheduled Deduction section of the Employee Information Deductions tab. The Employee's Withholdings are calculated next. Then all Deductions set up as calculated on net pay are calculated in the order they appear in the Scheduled Deduction section of the Employee Information Deductions tab.

10. To add this Deduction to the Payroll system, select the **Deductions/Employer Costs** function from the **Codes Maintenance** menu.
11. This garnishment Deduction is not Employer Cost paid or deferred compensation and will be calculated on Net Pay so that it is calculated after all gross pay Deduction and Withholdings.
12. When you add this Deduction for an Employee, remember that since this Deduction is to be calculated after all other Deductions it should be the last Deduction listed in the Employee Information Deductions tab. Then if the Employee is taking other Deductions calculated on net pay, they will be Deducted before this garnishment Deduction is calculated.

> For details on setting up the Deduction for Employees, see the Creating a Maximum Net Pay Garnishment Formula section (page 9-25).

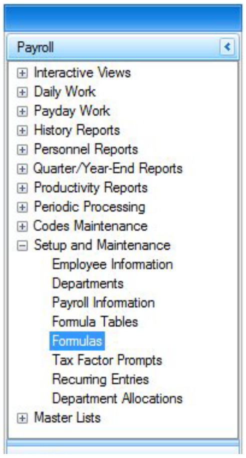
CREATING A 401K MATCHING EMPLOYER COST

The employer 401K match is calculated by deducting the amount of the Employee's 401K contribution from the Employee's gross pay and then calculating the employer's match as 25% of 3% of the employee's gross pay after subtracting the Employee's 401K contribution. This example shows the importance of the order of the Deductions on the Deduction tab in Employee Information when using the condition f.Deduction(id).

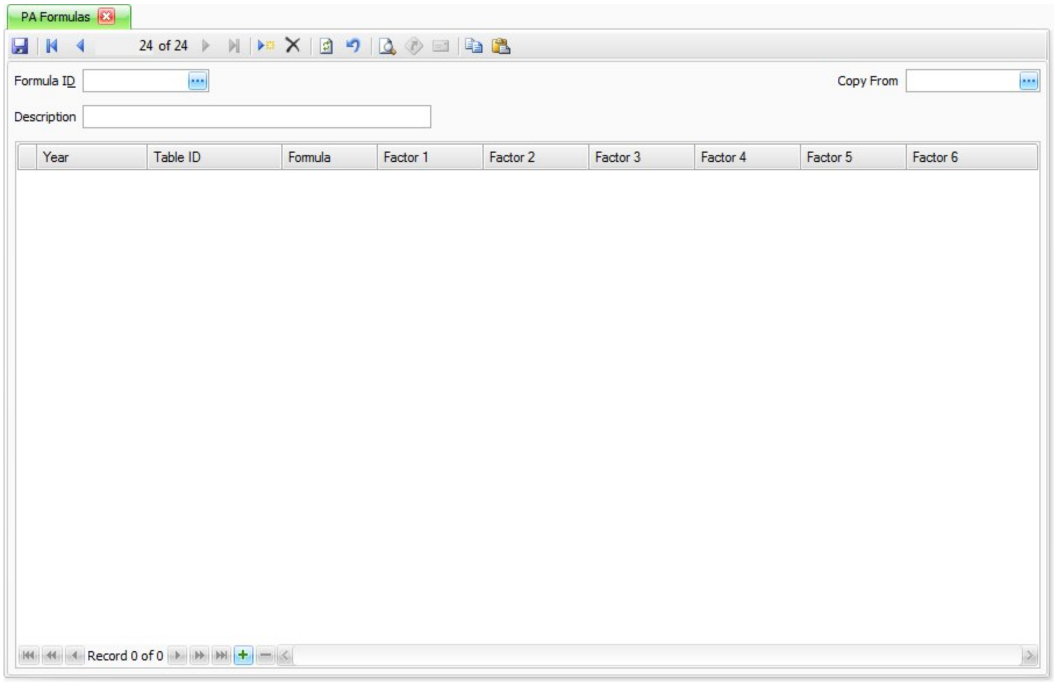
> **Example: If an Employee's gross pay is $1,000 and their 401K contribution is $30, which is 3%, the Employee's contribution is subtracted from the gross pay, $1,000 - $30 = $970. The employer will calculate his match on 3% of the $970, $970*3% = $29.10, with a 25% match, $29.10* 25% = $7.28.**

Set up the Formula from the same screen using the Formula function. This Formula is similar to the one we used when we covered the Deduction section except for the subtraction of the Employee's 401K contribution from gross pay before calculating the employer's match.

1. Select **Formulas** from the **Setup and Maintenance** menu.



2. The **Formulas** screen is displayed. Select the **New Record** icon ▶. A blank Formulas screen is displayed.



3. Enter the **Formula ID** as **401K25_3Max**, and the **Description** of **Gross-Emp 401K Contrib; 25 Max 3%**.

4. Click the green plus (+) sign at the bottom of the screen to add a new Formula for the current payroll Year.

5. Enter **3.0000** into **Factor 1** and skip Factors 2 though 6.

6. Click the drop down list arrow in the **Formula** column to display the Formula box, into which to enter the Formula.

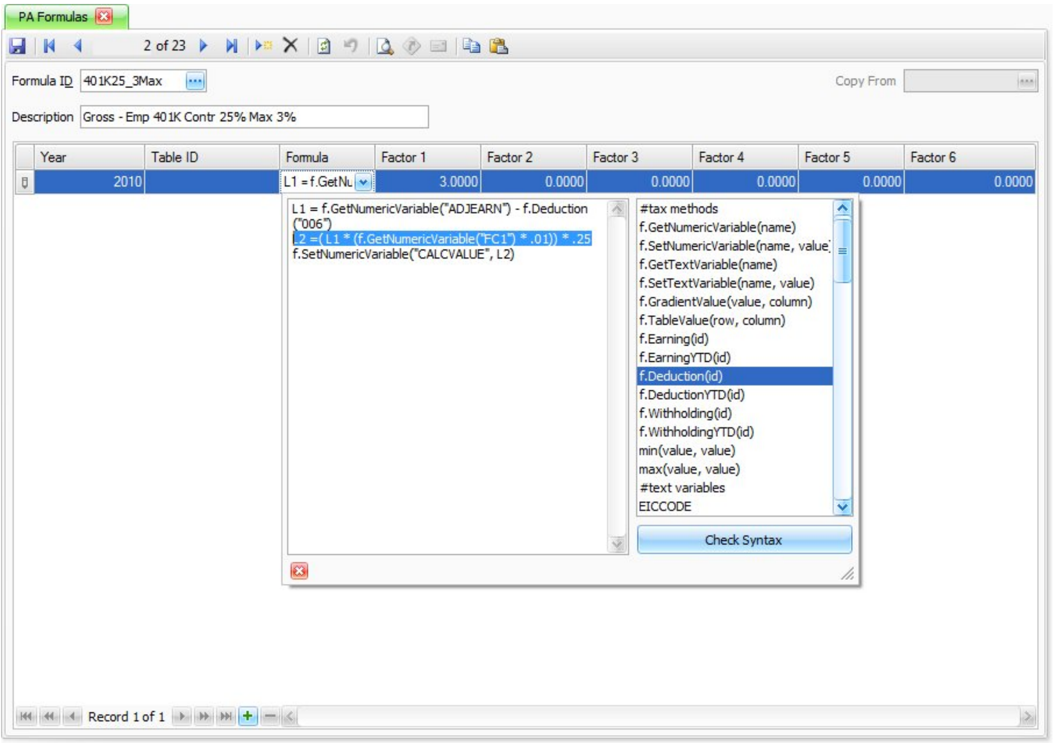
7. Enter the following formula lines: for line 1 enter:

L1 = f.GetNumericVariable(“ADJEARN”) - f.Deduction(“006”)

(006 is the Deduction Code for the Employee 401K)

8. For line 2 enter:

L2 =(L1 * (f.GetNumericVariable(“FC1”) * .01)) * .25.



2 of 23

Formula ID 401K25_3Max

Description Gross - Emp 401K Contr 25% Max 3%

Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2010		L1 = f.GetNu	3.0000	0.0000	0.0000	0.0000	0.0000	0.0000

```
L1 = f.GetNumericVariable("ADJEARN") - f.Deduction
("006")
L2 =( L1 * (f.GetNumericVariable("FC1") * .01)) * .25
f.SetNumericVariable("CALCVALUE", L2)
```

Record 1 of 1

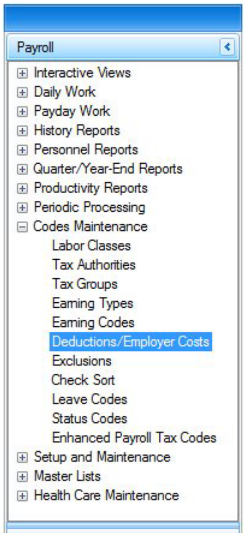
9. Enter the next line as:

f.SetNumericVariable(“CALCVALUE”, L2)

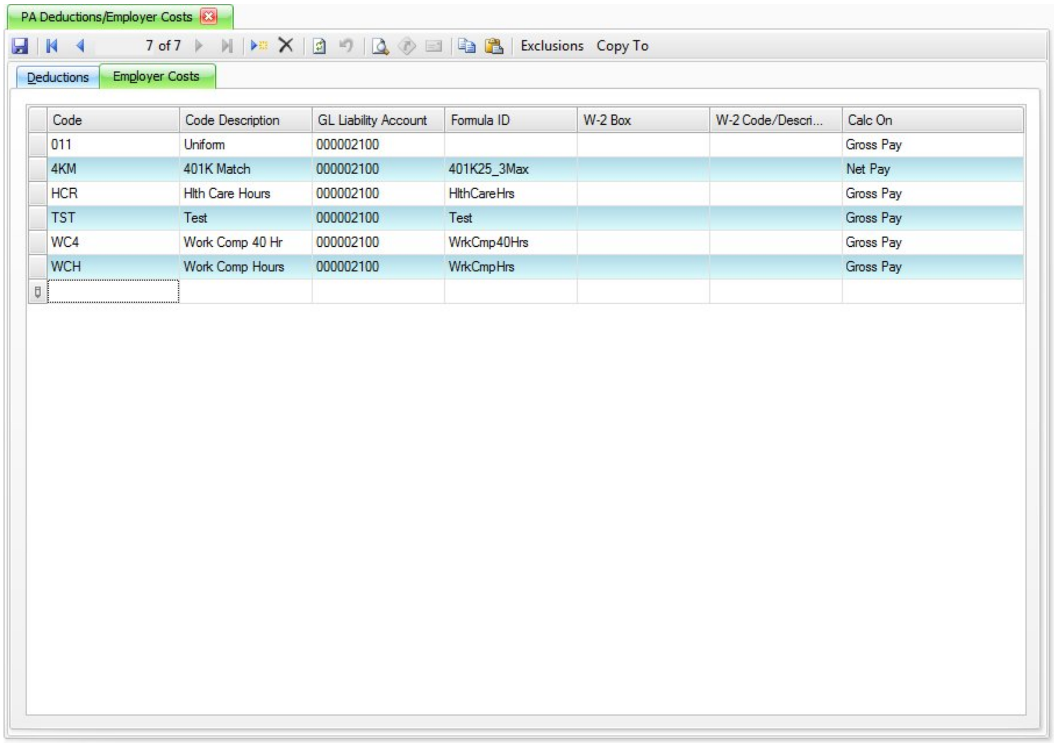
This returns the value calculated in line 2.

10. L1 is multiplied by FC1, the percent of the Employee’s pay that the employer will match, and then multiplied by .25 to calculate the actual amount of the employer’s contribution.

11. Set up the Deduction in the **Deductions/Employer Costs** function on the **Codes Maintenance** menu.



12. The **Deductions/Employer Costs** screen is displayed. Because this is an Employer Cost, select the **Employer Costs** tab.



Code	Code Description	GL Liability Account	Formula ID	W-2 Box	W-2 Code/Descri...	Calc On
011	Uniform	000002100				Gross Pay
4KM	401K Match	000002100	401K25_3Max			Net Pay
HCR	Hlth Care Hours	000002100	HlthCareHrs			Gross Pay
TST	Test	000002100	Test			Gross Pay
WC4	Work Comp 40 Hr	000002100	WrkCmp40Hrs			Gross Pay
WCH	Work Comp Hours	000002100	WrkCmpHrs			Gross Pay

13. Enter a descriptive **Code** and Code **Description**.

14. Select the GL **Liability Account**.

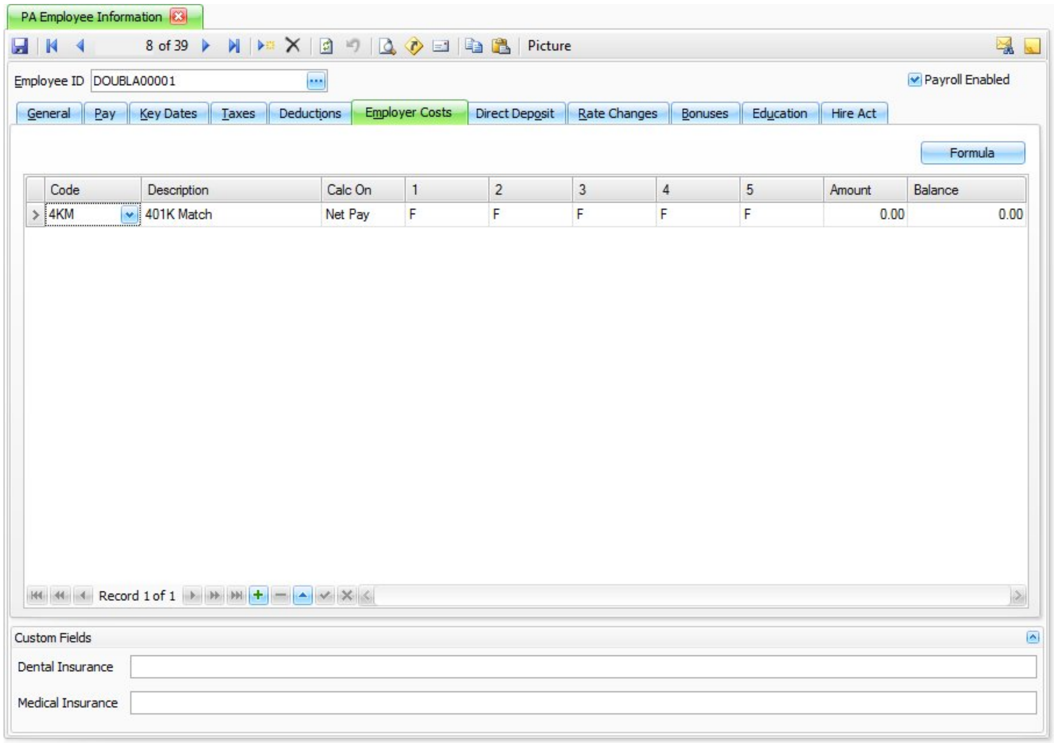
15. Select the **Formula ID**.

16. Enter 14 as the **W-2 Box** to print the amount in Box 14 of the Employee's W-2 if you want the amount printed. Leave the field blank if you do not want to print the total.

17. Select **Gross Pay**.

18. Select **Employee Information** from the **Setup and Maintenance** menu.

19. The **Employee Information** screen is displayed. Select the **Employer Cost** tab and enter the **Code** for your employer 401K match.



Employee ID DOUBLA00001

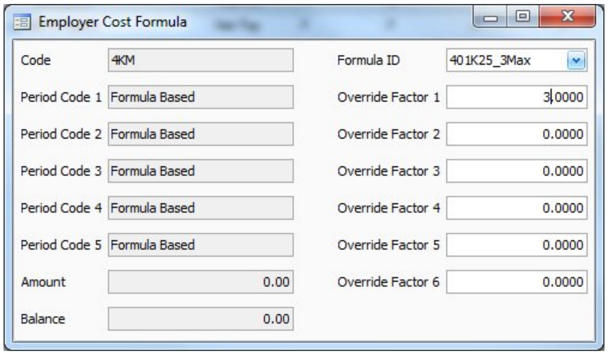
☑ Payroll Enabled

Code	Description	Calc On	1	2	3	4	5	Amount	Balance
4KM	401K Match	Net Pay	F	F	F	F	F	0.00	0.00

Dental Insurance

Medical Insurance

If the Employee is contributing less than 3% maximum employer match, select **Formula** to set **Override Factor 1** to the correct percentage for the Employee.



Employer Cost Formula

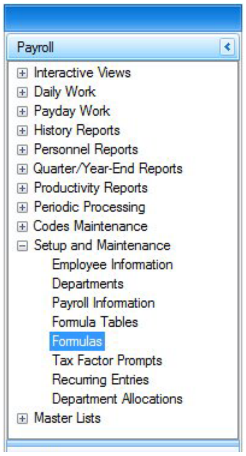
Code	4KM	Formula ID	401K25_3Max
Period Code 1	Formula Based	Override Factor 1	3.0000
Period Code 2	Formula Based	Override Factor 2	0.0000
Period Code 3	Formula Based	Override Factor 3	0.0000
Period Code 4	Formula Based	Override Factor 4	0.0000
Period Code 5	Formula Based	Override Factor 5	0.0000
Amount	0.00	Override Factor 6	0.0000
Balance	0.00		

WORKING WITH 3% NET MATCH FORMULA

This 401K match will look at the Employee's adjusted earnings, their net pay after all Deductions and Taxes have been taken out, and then calculate 3% of what is left and match 50% of the 3% calculated.

> **Example: The formula uses the ADJEARN variable to calculate the net pay so it can match 50% of the remaining 3% of the Employee's pay.**

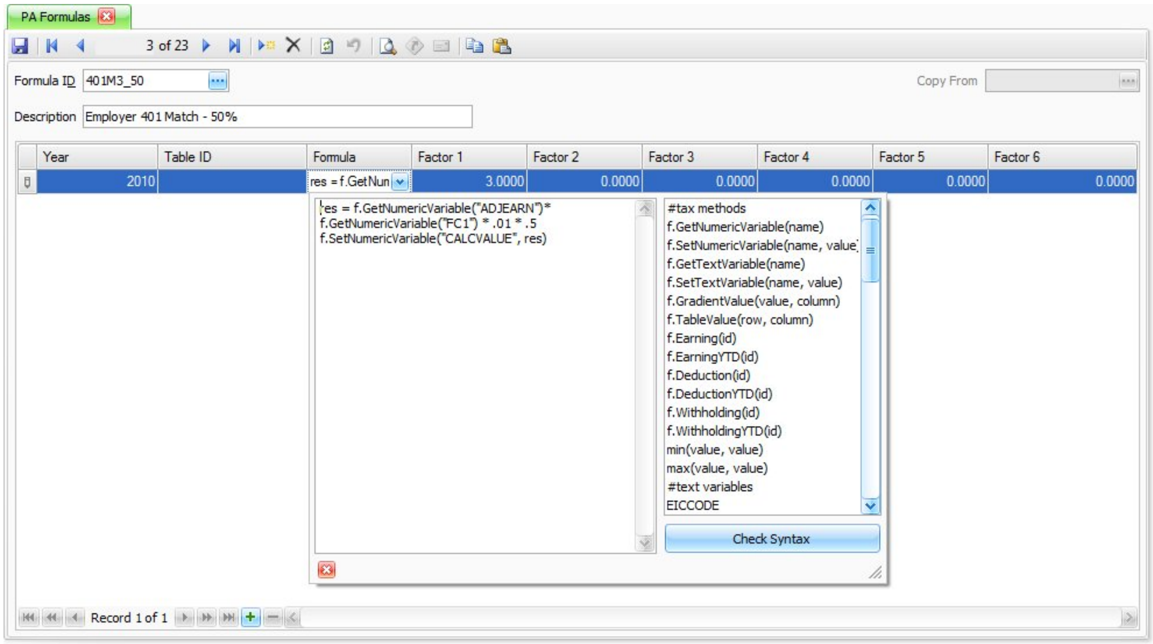
1. Select **Formulas** from the **Setup and Maintenance** menu.



2. The **Formulas** screen is displayed. Select the **New Record** icon . A blank Formulas screen is displayed.
3. For the **Formula ID**, enter **401M3_50**.
4. Tab to the **Description** field, and enter a Description for the formula. For the Description, enter **Employer 401 Match - 50% OF 3%**.
5. Click the green plus (+) sign at the bottom of the screen to add a new Formula for the current payroll Year.
6. Enter **3.0000** into **Factor 1** and skip Factors 2 though 6.
7. Click the drop down list arrow in the **Formula** column to display the Formula box, into which to enter the Formula.

8. For line 1, enter:

res = f.GetNumericVariable(“ADJEARN”)* f.GetNumericVariable(“FC1”) * .01 * .5.



Formula ID 401M3_50

Description Employer 401 Match - 50%

Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2010		res = f.GetNun	3.0000	0.0000	0.0000	0.0000	0.0000	0.0000

```
res = f.GetNumericVariable("ADJEARN")*
f.GetNumericVariable("FC1") * .01 * .5
f.SetNumericVariable("CALCVALUE", res)
```

9. For the second line to return the value calculated enter:

f.SetNumericVariable(“CALCVALUE”, res)

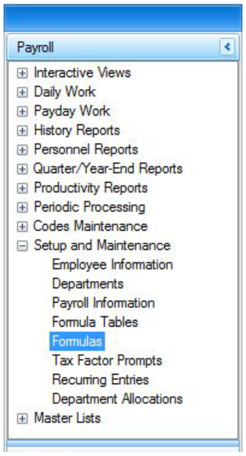
The value **res** is considered the line number in this case and that is the value that is being returned with the final statement. This is another example of how line number values can vary.

10. Set up the **Employer Cost** and add it to the **Employee Information**, **Employer Cost** tab just like the previous example.

WORKING WITH A 50% MATCH DEDUCTION

This 401K employer's match, will match 50% of the amount of the 401K Deduction taken out of the Employee's paycheck. This Formula uses the GETDEDUCTION variable to get the amount of the Deduction we specify in the Formula to match 50% of that amount as the Employer Cost.

1. Select **Formulas** from the **Setup and Maintenance** menu.



2. The **Formulas** screen is displayed. Select the **New Record** icon . A blank Formulas screen is displayed. For the formula ID, enter **401M50Per**.

3. Tab to the Description field, and enter a description for the formula. For the description, enter **EMPLOYER 401K MATCH - 50%**.

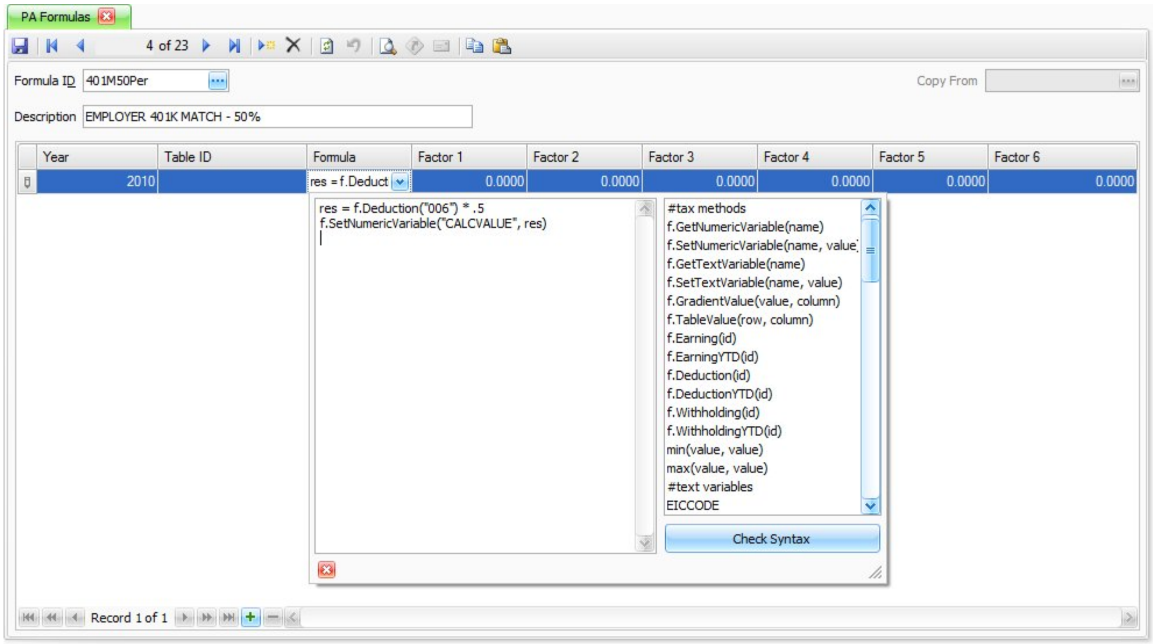
4. Click the green plus (+) sign at the bottom of the screen to add a new Formula for the current payroll Year.

5. Skip **Factors** 1 – 6.

6. Click the drop down list arrow in the **Formula** column to display the Formula box, into which to enter the Formula.

7. For line 1, enter:

   res = f.Deduction(“006”) * .5.



Formula ID 401M50Per

Description EMPLOYER 401K MATCH - 50%

Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2010		res = f.Deduct	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

```
res = f.Deduction("006") * .5
f.SetNumericVariable("CALCVALUE", res)
```

8. For the second line to return the value calculated enter:

   f.SetNumericVariable(“CALCVALUE”, res)

   The value **res** is considered the line number in this case and that is the value that is being returned with the final statement. This is another example of how line number values can vary.

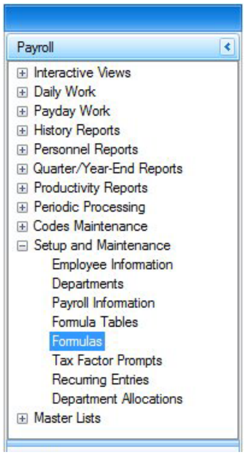
9. Set up the **Employer Cost** and add it to the **Employee Information**, **Employer Cost** tab just like the previous examples.

WORKING WITH A NEGATIVE OVERTIME DEDUCTION

A negative overtime Deduction will add an amount to the Employee's paycheck as a Deduction for the hours they work that are considered overtime. When you set up a Deduction as a negative number it will add to the Employee's pay, instead of taking away from the pay.

> **Example: We will be calculating the amount of pay that is strictly overtime and adding it to the Employee's paycheck as a Deduction.**

1. Select **Formulas** from the **Setup and Maintenance** menu.



2. The **Formulas** screen is displayed. Select the **New Record** icon . A blank Formulas screen is displayed.
3. For the **Formula ID**, enter **NegDedOVT**.
4. Tab to the **Description** field, and enter a Description for the Formula. For the Description, enter **Employee Negative OT Deduction**.
5. Click the green plus (+) sign at the bottom of the screen to add a new Formula for the current payroll Year.
6. For **Factor 1**, enter **-1.00**, and skip factors 2 – 6.
7. Click the drop down list arrow in the **Formula** column to display the Formula box, into which to enter the Formula.

8. For line 1, enter:

   L1 = f.GetNumericVariable(“GRANDTOTGROSS”) / f.GetNumericVariable(“HOURS”)

9. For line 2, enter:

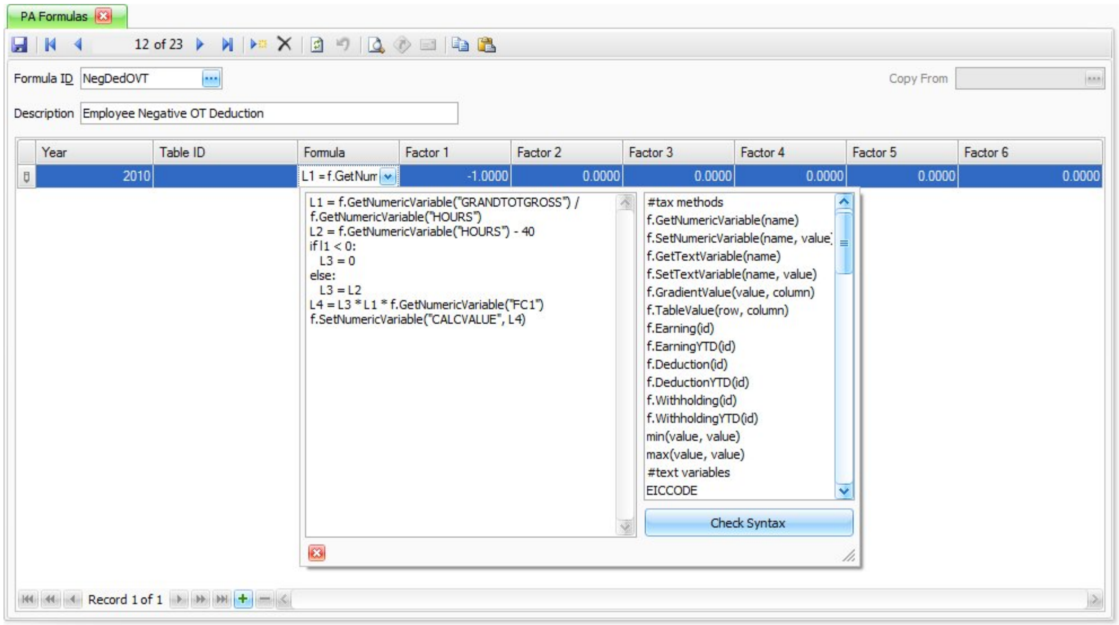
   L2 = f.GetNumericVariable(“HOURS”) - 40

10. For line 3, enter:

```
if L1 < 0:
  L3 = 0
else:
  L3 = L2
```

11. For line 4, enter:

    L4 = L3 * L1 * f.GetNumericVariable(“FC1”)



Formula ID NegDedOVT

Description Employee Negative OT Deduction

Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2010		L1 = f.GetNum	-1.0000	0.0000	0.0000	0.0000	0.0000	0.0000

```
L1 = f.GetNumericVariable("GRANDTOTGROSS") /
f.GetNumericVariable("HOURS")
L2 = f.GetNumericVariable("HOURS") - 40
if l1 < 0:
  L3 = 0
else:
  L3 = L2
L4 = L3 * L1 * f.GetNumericVariable("FC1")
f.SetNumericVariable("CALCVALUE", L4)
```

12. For the final line enter this to return the calculated value:

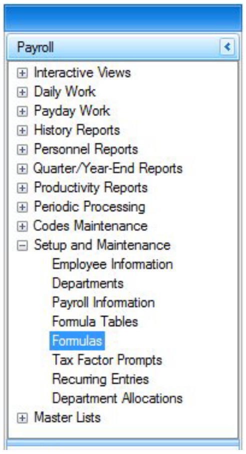
    f.SetNumericVariable(“CALCVALUE”, L4)

13. Set up the **Deduction** and add it to the **Employee Information**, **Deductions** tab just like the previous examples.

WORKING WITH A PRODUCTION BONUS

The production bonus will use a negative Deduction to add a dollar amount to the Employee's paycheck for hours they work over the regular hours with the overtime calculated and time and a half. When a negative Deduction amount is used, the result is adding an amount to the Employee's paycheck instead of deducting from it.

1. Select **Formulas** from the **Setup and Maintenance** menu.



2. The **Formulas** screen is displayed. Select the **New Record** icon . A blank Formulas screen is displayed.

3. For the **Formula ID**, enter **PROD**.

4. Tab to the **Description** field, and enter a Description for the Formula. For the Description, enter **Production Bonus**.

5. Click the green plus (+) sign at the bottom of the screen to add a new Formula for the current payroll Year.

6. For **Factor 1**, enter **-1.00** and **12.00** for **Factor 2** and skip Factors 3 – 6. Factor 2 will be the Employees Hourly Wage.

7. Click the drop down list arrow in the **Formula** column to display the Formula box, into which to enter the Formula.

8. For line 1, enter:

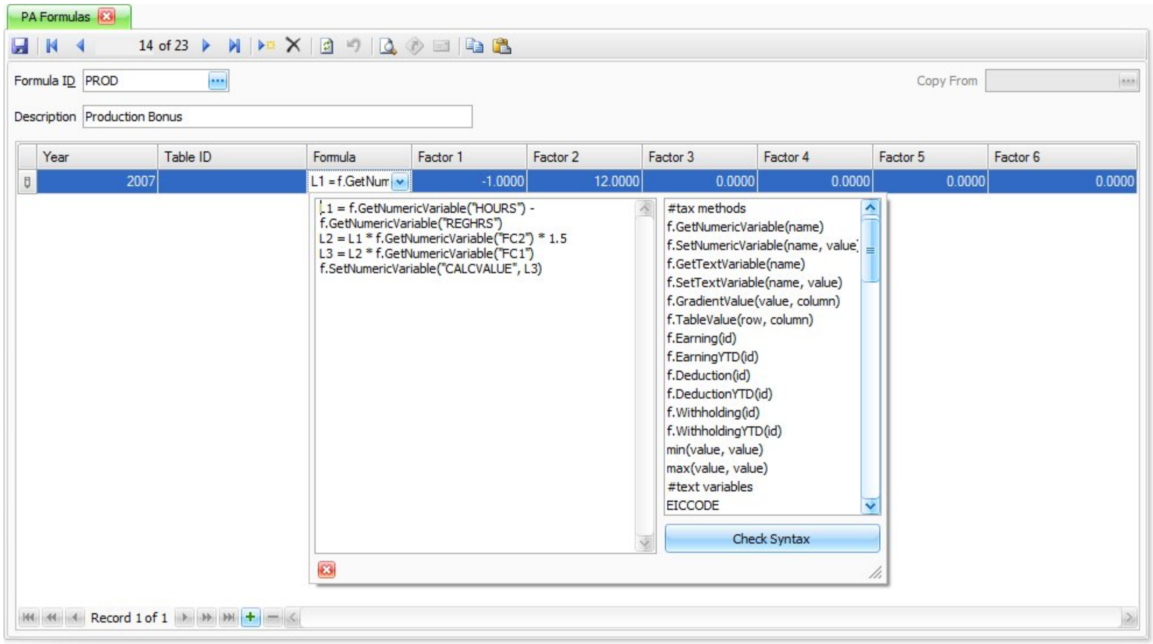
L1 = f.GetNumericVariable("HOURS") - f.GetNumericVariable("REGHRS")

9. For line 2, enter:

   L2 = L1 * f.GetNumericVariable(“FC2”) * 1.5

10. For line 3, enter:

   L3 = L2 * f.GetNumericVariable(“FC1”)



Formula ID: PROD

Description: Production Bonus

Copy From:

Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2007		L1 = f.GetNum	-1.0000	12.0000	0.0000	0.0000	0.0000	0.0000

```
L1 = f.GetNumericVariable("HOURS") -
f.GetNumericVariable("REGHRS")
L2 = L1 * f.GetNumericVariable("FC2") * 1.5
L3 = L2 * f.GetNumericVariable("FC1")
f.SetNumericVariable("CALCVALUE", L3)
```

11. For the final line enter this to return the calculated value:

   f.SetNumericVariable(“CALCVALUE”, L3)

12. Set up the **Deduction** and add it to the **Employee Information**, **Deductions** tab just like the previous examples.

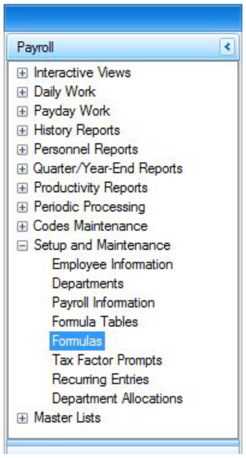
13. Use the **Override Factor 2** on the **Deductions** tab to put in the regular wage amount Override Factor for each Employee, when you select the **Formula** button.

WORKING WITH WORKMAN’S COMPENSATION

This is a generic description of how to set up a workman’s compensation Employer’s Cost or Employee Deduction. This example will take a percent of the Employee’s Adjusted Earnings, after Taxes and Deductions and calculate a Deduction from that calculated amount.

To make this workman’s compensation calculation follow these steps.

1. Select **Formulas** from the **Setup and Maintenance** menu.



2. The **Formula** screen is displayed. Select the **New Record** icon . A blank Formulas screen is displayed.

3. For the **Formula ID**, enter **WCOMP.**

4. Tab to the **Description** field, and enter a Description for the Formula. For the Description, enter **Workman’s Comp Calculation.**

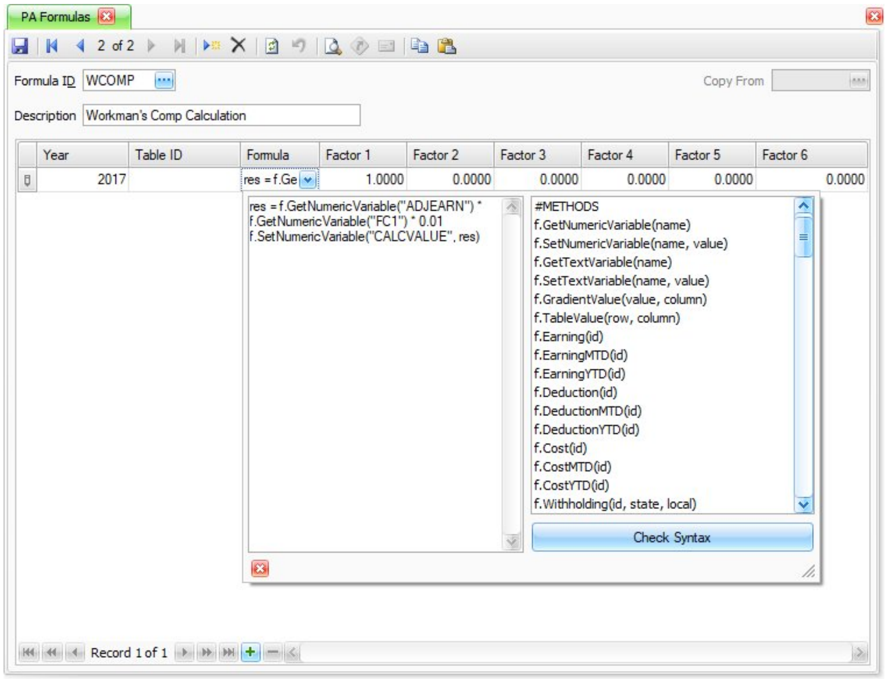
5. Click the green plus (+) sign at the bottom of the screen to add a new Formula for the current payroll Year.

6. For **Factor 1**, enter **1.0000**, and skip factors 2 - 6.

7. Click the drop down list arrow in the **Formula** column to display the Formula box, into which to enter the Formula.

8. For line 1, enter:

   res = f.GetNumericVariable("ADJEARN") * f.GetNumericVariable("FC1") * 0.01



Formula ID WCOMP

Description Workman's Comp Calculation

Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2017		res = f.Ge	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000

```
res = f.GetNumericVariable("ADJEARN") *
f.GetNumericVariable("FC1") * 0.01
f.SetNumericVariable("CALCVALUE", res)
```

#METHODS
f.GetNumericVariable(name)
f.SetNumericVariable(name, value)
f.GetTextVariable(name)
f.SetTextVariable(name, value)
f.GradientValue(value, column)
f.TableValue(row, column)
f.Earning(id)
f.EarningMTD(id)
f.EarningYTD(id)
f.Deduction(id)
f.DeductionMTD(id)
f.DeductionYTD(id)
f.Cost(id)
f.CostMTD(id)
f.CostYTD(id)
f.Withholding(id, state, local)

9. For the second line to return the value calculated enter:

   f.SetNumericVariable("CALCVALUE", res)

   The value **res** is considered the line number in this case and that is the value that is being returned with the final statement. This is another example of how line number values can vary.

10. Set up the **Deduction** or **Employer Cost** and add it to the **Employee Information**, **Deductions** or **Employer Cost** tab just like the previous examples.

11. Use the **Override Factor 1** on the **Deductions** or **Employer Costs** tab to put in the Workman's Comp percent amount Override Factor for each Employee, when you select the **Formula** button.

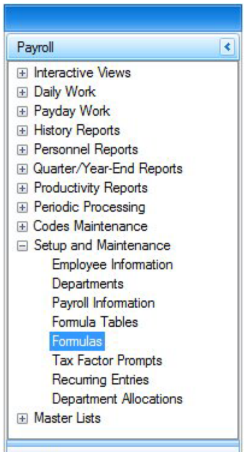
WORKING WITH SAIF

SAIF is used in some states to deduct from the Employee's paycheck to pay for an Employee paid disability tax. SAIF is set up as a Deduction and then the employer will pay the State from the amount deducted from the Employee's paycheck.

Example: This example will take 1% of each hour of regular pay the Employee works.

To set up this example of the SAIF Deduction follow these steps:

1. Select **Formulas** from the **Setup and Maintenance** menu.



2. The **Formulas** screen is displayed. Select the **New Record** Icon . A blank Formulas screen is displayed.
3. For the **Formula ID**, enter **SAIF.**
4. Tab to the **Description** field, and enter a Description for the Formula. For the Description, enter **SAIF.**
5. Click the green plus (+) sign at the bottom of the screen to add a new Formula for the current payroll Year.
6. For **Factor 1**, enter **1.0000**, and skip Factors 2- 6.
7. Click the drop down list arrow in the **Formula** column to display the Formula box, into which to enter the Formula.

8. For line 1, enter:

   L1 = f.GetNumericVariable(“GRANDTOTGROSS”) / f.GetNumericVariable(“REGHRS”)

   This will calculate their hourly rate.

9. For line 2, enter:

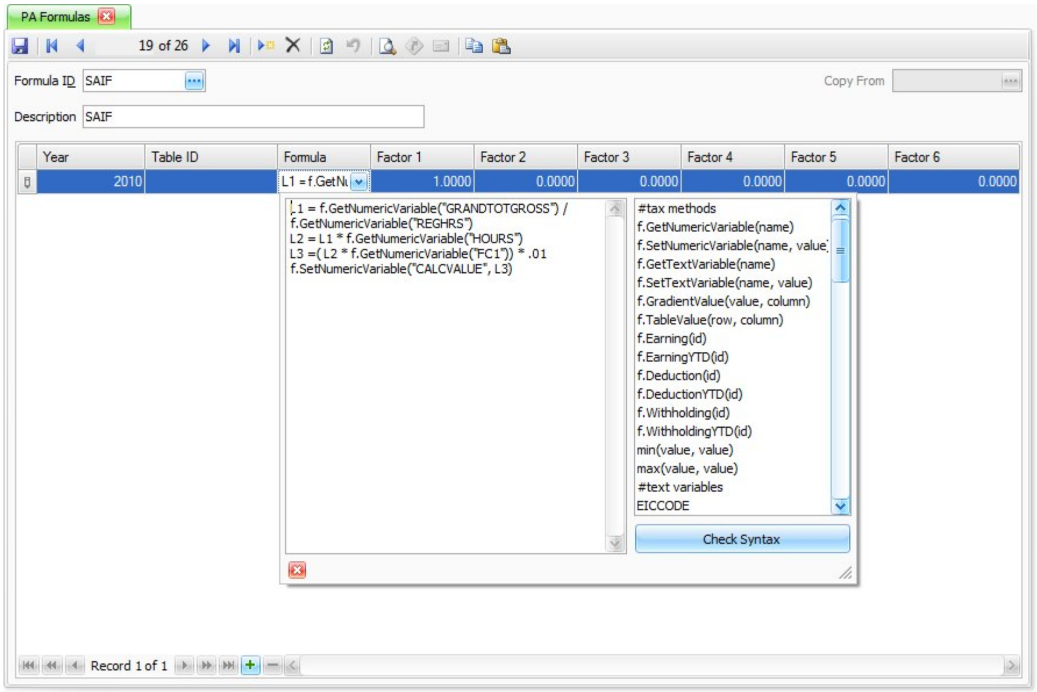
   L2 = L1 * f.GetNumericVariable(“HOURS”)

   This will calculate their regular pay.

10. For line 3, enter:

    L3 =(L2 * f.GetNumericVariable(“FC1”)) * 0.01

    This will calculate the amount of the tax.



Formula ID: SAIF

Description: SAIF

Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2010		L1 =f.GetNı	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000

```
L1 = f.GetNumericVariable("GRANDTOTGROSS") /
f.GetNumericVariable("REGHRS")
L2 = L1 * f.GetNumericVariable("HOURS")
L3 =( L2 * f.GetNumericVariable("FC1")) * .01
f.SetNumericVariable("CALCVALUE", L3)
```

11. For the final line enter this to return the calculated value:

    f.SetNumericVariable(“CALCVALUE”, L3)

12. Set up the **Deduction** and add it to the **Employee Information**, **Deductions** tab just like the previous examples.

13. Use the **Override Factor 1** on the **Deductions** tab to put in the SAIF percent amount Override Factor for each Employee, when you select the **Formula** button.

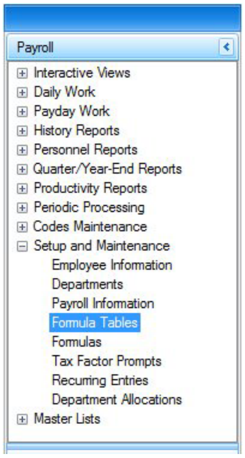
MAKING A COMPANY SPECIFIC SUI CALCULATION

If you have multiple companies set up in TRAVERSE and they are each using the Payroll module and have a different SUI rate for each company, you will need to set up company specific SUI rates. By default TRAVERSE sets the SUI rate at the maximum percentage for each State. You will need to manually change the percentage rate for each company.

Make sure you know the tax rate for each company before you begin.

To have different rates for different companies, you can do the following steps:

1. Select **Formula Tables** from the **Setup and Maintenance** menu.

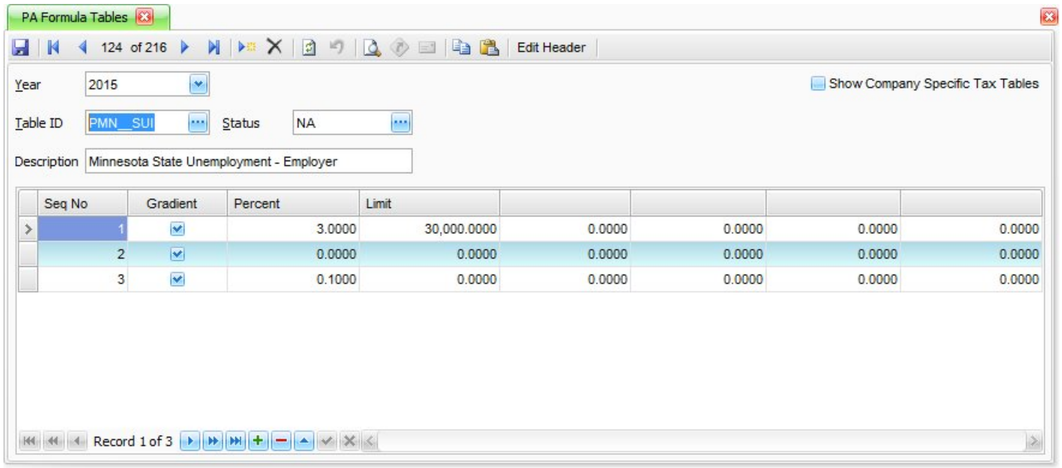


2. Select the **Show Company Specific Tax Tables** check box and add a new Formula Table patterned after a standard SUI table with two columns, one for **Percent** and one for **Limit**. Save your changes when you finish.

   Use the naming convention **PYYCCCSU** where **YY** is the state code and **CCC** is the three-character company code to which the table is connected.

   Use decimals, not fractions, when entering percentage rates. For example, if the rate is 1%, enter **1.000**. If the rate is ½%, enter **0.500**.

3. Selecting the check box saves the new table in the company specific database, which is where TRAVERSE's stored procedures look for the percentages and limits to use when calculating SUI.



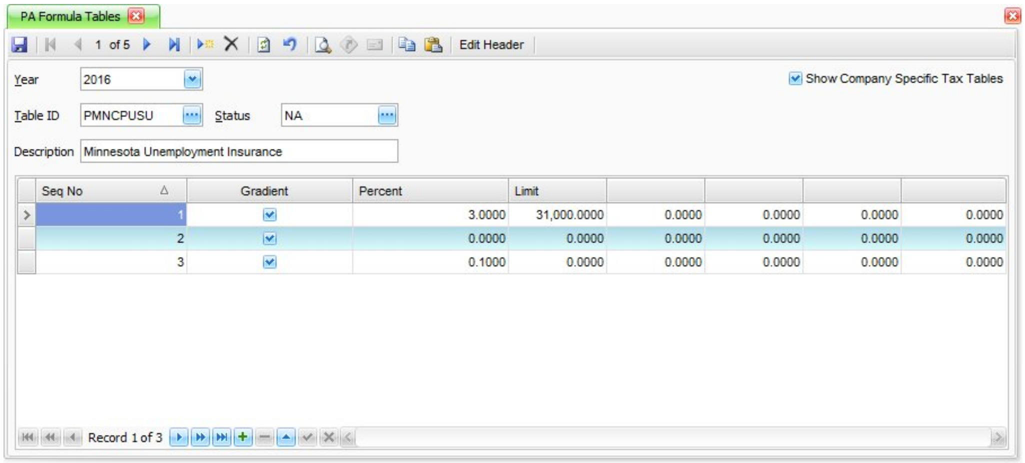
Year 2015

□ Show Company Specific Tax Tables

Table ID PMN_SUI Status NA

Description Minnesota State Unemployment - Employer

Seq No	Gradient	Percent	Limit				
1	☑	3.0000	30,000.0000	0.0000	0.0000	0.0000	0.0000
2	☑	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3	☑	0.1000	0.0000	0.0000	0.0000	0.0000	0.0000



Year 2016

☑ Show Company Specific Tax Tables

Table ID PMNCPUSU Status NA

Description Minnesota Unemployment Insurance

Seq No	Gradient	Percent	Limit				
1	☑	3.0000	31,000.0000	0.0000	0.0000	0.0000	0.0000
2	☑	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3	☑	0.1000	0.0000	0.0000	0.0000	0.0000	0.0000

4. Clear the **Show Company Specific Tax Tables** check box and create an identical copy of the table you created in step 3, including the table ID.

   The State Unemployment Report looks in the non-company specific ST database to generate the report, rather than the company specific database. In order for this report to print correctly, you need two copies of the table, one in the company specific database, one in the ST database.

5. Select **Formulas** from the **Setup and Maintenance** menu and create a new formula for SUI.

   Use the naming convention **PYY__SUI**, where **YY** is the company's state abbreviation. Note that there are two underscores and that this exact syntax must be used, otherwise it will not calculate correctly.

   Enter the ID of the Formula Table you created in step 3 in the **Table ID** field. When you calculate checks, TRAVERSE uses the percentage and limit from that table.

   Leave the **Formula** section blank. This makes the stored procedure program look at the company specific SUI table you created in step 3 for the percentages and limits to use in the calculation, but use the standard SUI calculation that is stored in the ST database.

6. Switch to the next company and repeat the process for each company in TRAVERSE.

Tips:

- Make sure the syntax for the Formula Table ID you make is exactly as listed in step 2 above or it will not calculate correctly.
- Each company can have a different tax rate; make sure you know what the rate is for each company.
- When putting in your percentage rates, put them in as whole numbers. For example if your rate is 1% put in 1.0000 and if your rate is ½% put it in as.5000. We multiply the number in the first field in the table by 0.01 in line 6 of the formula.

SETTING UP LOCAL TAXES

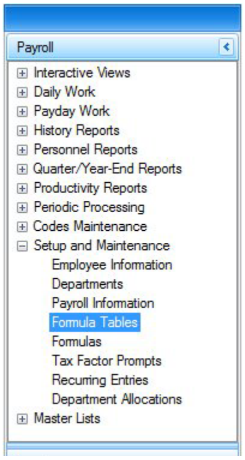
There are multiple ways to set up Local Taxes. Some Local Taxes are very simple and have a fixed percentage and do not require a table, and some Local Taxes are a bit more complicated and require a Tax Table and a Formula similar to the State Withholding Tax Formulas.

> **Example: There are examples of both the very basic, simple Local Tax calculation and more complicated table required Local Tax calculations.**

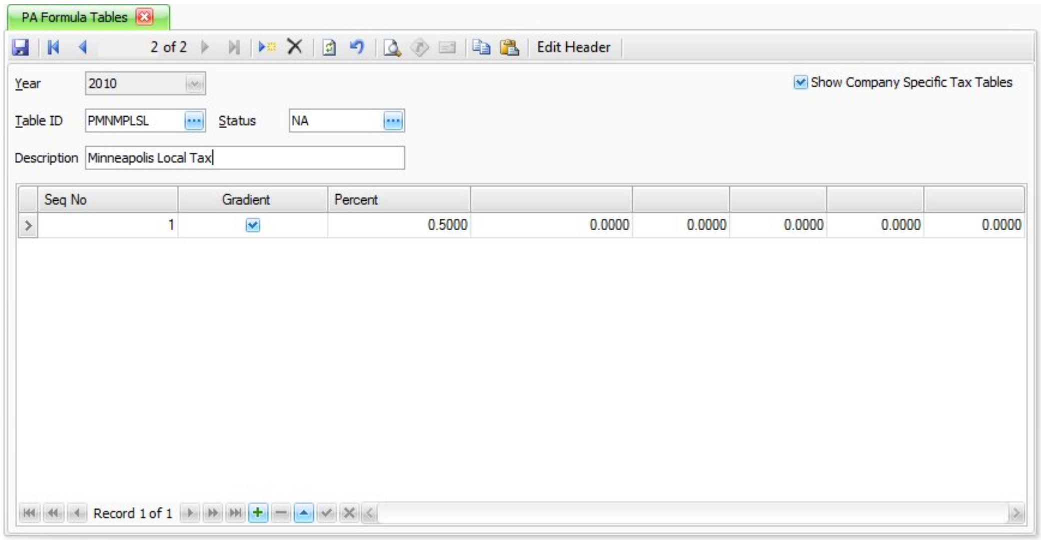
Here are the basic steps you will need to follow to set up your Local Tax.

To create a Local Tax that is similar to the standard State Withholding Tax calculations you need to set up a Local Withholding Tax Table prior to writing the Formula to calculate the tax.

1. Create the local Table. To do this open the **Formula Tables** in the **Setup and Maintenance** menu. Make sure to select the **Show Company Specific Tax Tables** check box.



2. Select the **New Record** Icon . Add a new **Table ID** for the local tax. The information to put on this Table needs to come from your State or Locality (city, county, school district, parish).



Year 2010

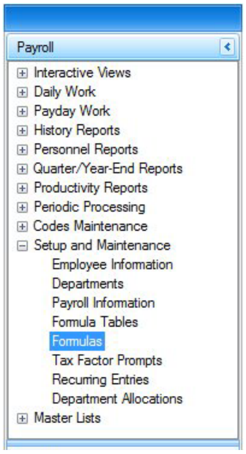
Table ID PMNMPLSL Status NA

Description Minneapolis Local Tax

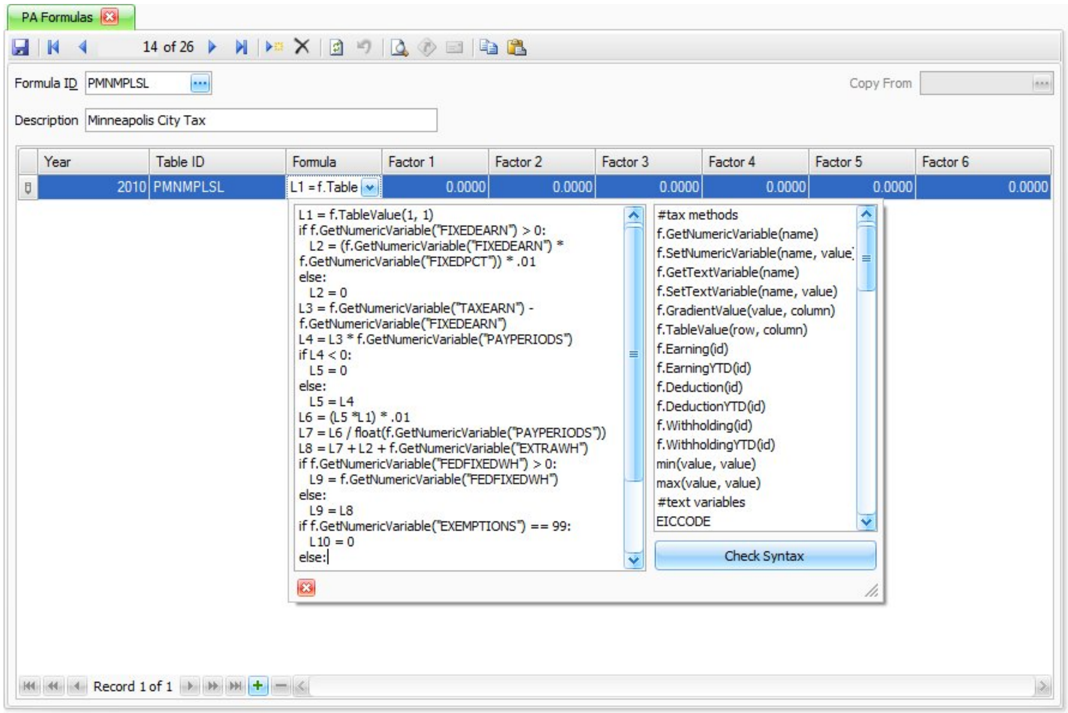
☑ Show Company Specific Tax Tables

Seq No	Gradient	Percent					
1	☑	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000

3. Write the Formula. Select **Formulas** from the **Setup and Maintenance** menu.



4. Select the **New Record** Icon ▶✳. Enter the Formula to calculate the Local Tax. Below is an example of a simple Formula that will read a one field table with a percentage in it.



Formula ID PMNMPLSL

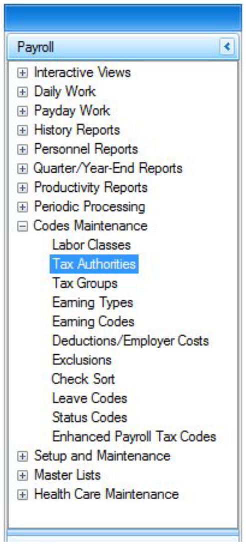
Description Minneapolis City Tax

Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2010	PMNMPLSL	L1 = f.Table	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

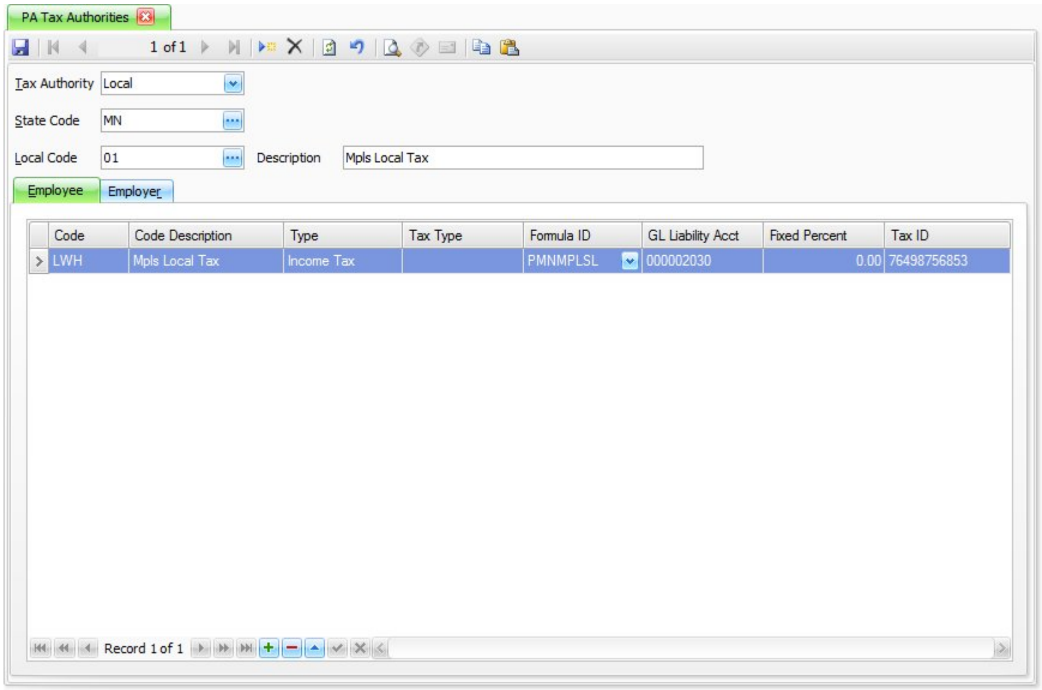
```
L1 = f.TableValue(1, 1)
if f.GetNumericVariable("FIXEDEARN") > 0:
  L2 = (f.GetNumericVariable("FIXEDEARN") *
f.GetNumericVariable("FIXEDPCT")) * .01
else:
  L2 = 0
L3 = f.GetNumericVariable("TAXEARN") -
f.GetNumericVariable("FIXEDEARN")
L4 = L3 * f.GetNumericVariable("PAYPERIODS")
if L4 < 0:
  L5 = 0
else:
  L5 = L4
L6 = (L5 *L1) * .01
L7 = L6 / float(f.GetNumericVariable("PAYPERIODS"))
L8 = L7 + L2 + f.GetNumericVariable("EXTRAWH")
if f.GetNumericVariable("FEDFIXEDWH") > 0:
  L9 = f.GetNumericVariable("FEDFIXEDWH")
else:
  L9 = L8
if f.GetNumericVariable("EXEMPTIONS") == 99:
  L10 = 0
else:
```

```
#tax methods
f.GetNumericVariable(name)
f.SetNumericVariable(name, value
f.GetTextVariable(name)
f.SetTextVariable(name, value)
f.GradientValue(value, column)
f.TableValue(row, column)
f.Earning(id)
f.EarningYTD(id)
f.Deduction(id)
f.DeductionYTD(id)
f.Withholding(id)
f.WithholdingYTD(id)
min(value, value)
max(value, value)
#text variables
EICCODE
```

5. Set up the **Local Tax Authority** from the **Codes Maintenance** menu.



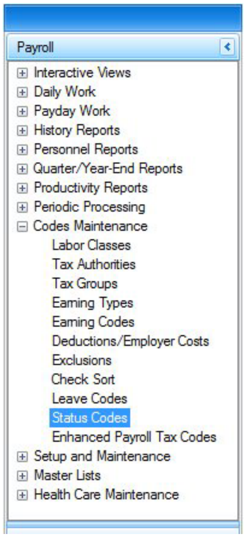
6. Select the **New Record** Icon ▶✳. Select your **State Code** and set up the **Local Code** with all the necessary information.



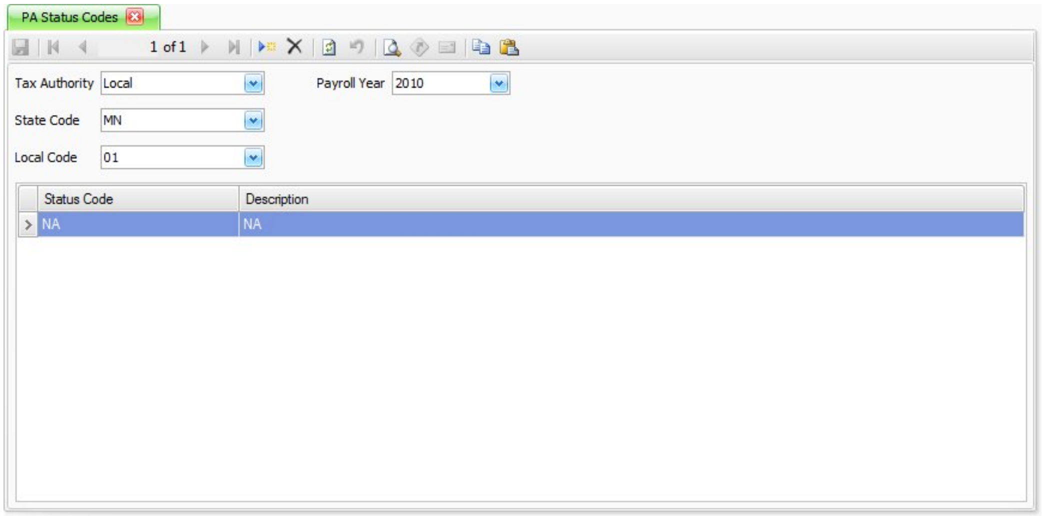
Tax Authority: Local
State Code: MN
Local Code: 01
Description: Mpls Local Tax

Code	Code Description	Type	Tax Type	Formula ID	GL Liability Acct	Fixed Percent	Tax ID
LWH	Mpls Local Tax	Income Tax		PMNMPLSL	000002030	0.00	76498756853

7. Add the Status Codes for the local tax, using **Status Codes** on the **Codes Maintenance** menu.



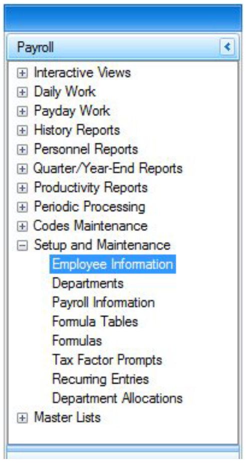
8. Select the **New Record** Icon . Select **Local Tax Authority**, then select your **State Code** and **Local Code**. Add the **Status Codes** from which your Local Code needs to pull information, for the correct Withholding Table. If you do not have a tax table, or multiple Statuses, then add **NA** as a status.



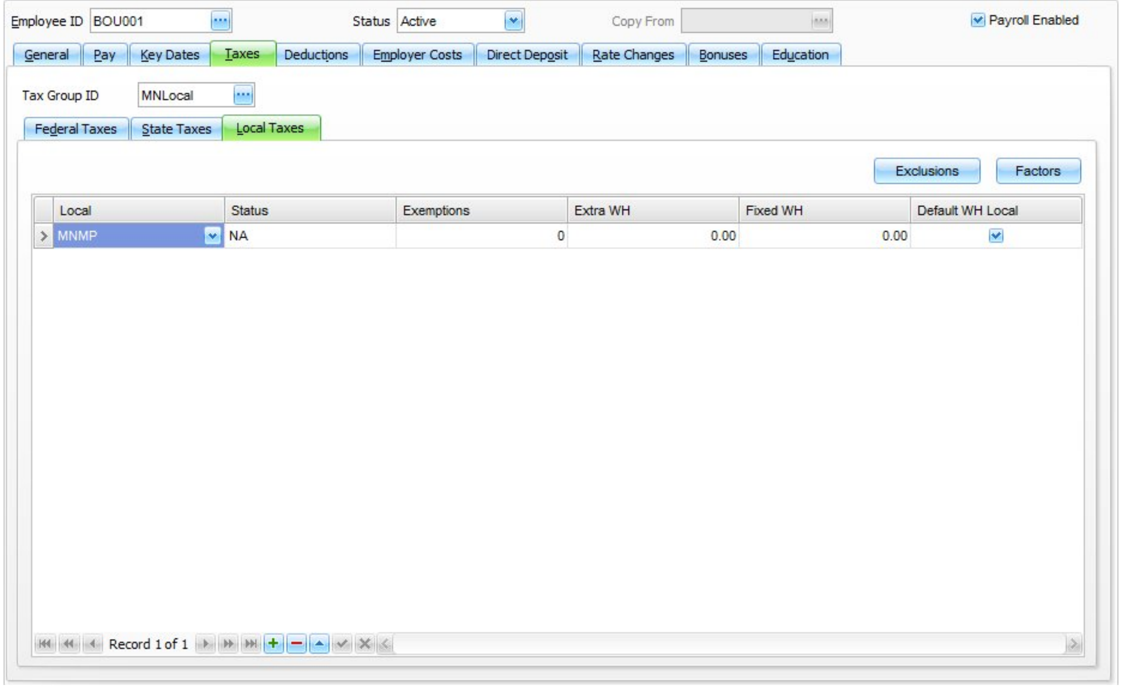
Tax Authority: Local
Payroll Year: 2010
State Code: MN
Local Code: 01

Status Code	Description
NA	NA

9. Add the tax record to the Employee, using **Employee Information** on the **Setup and Maintenance** menu. Go to the **Taxes** tab, then the **Local Taxes** tab.



10. Select the **Employee ID** for which to add the Local Tax, and go to the **Taxes** tab, then to the **Local Taxes** tab and select the **Local Code** you set up with a **Status**.



Employee ID BOU001 Status Active Copy From ☑ Payroll Enabled

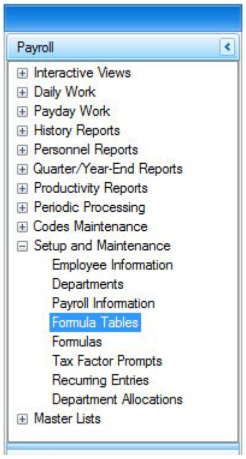
Tax Group ID MNLocal

Local	Status	Exemptions	Extra WH	Fixed WH	Default WH Local
MNMP	NA	0	0.00	0.00	☑

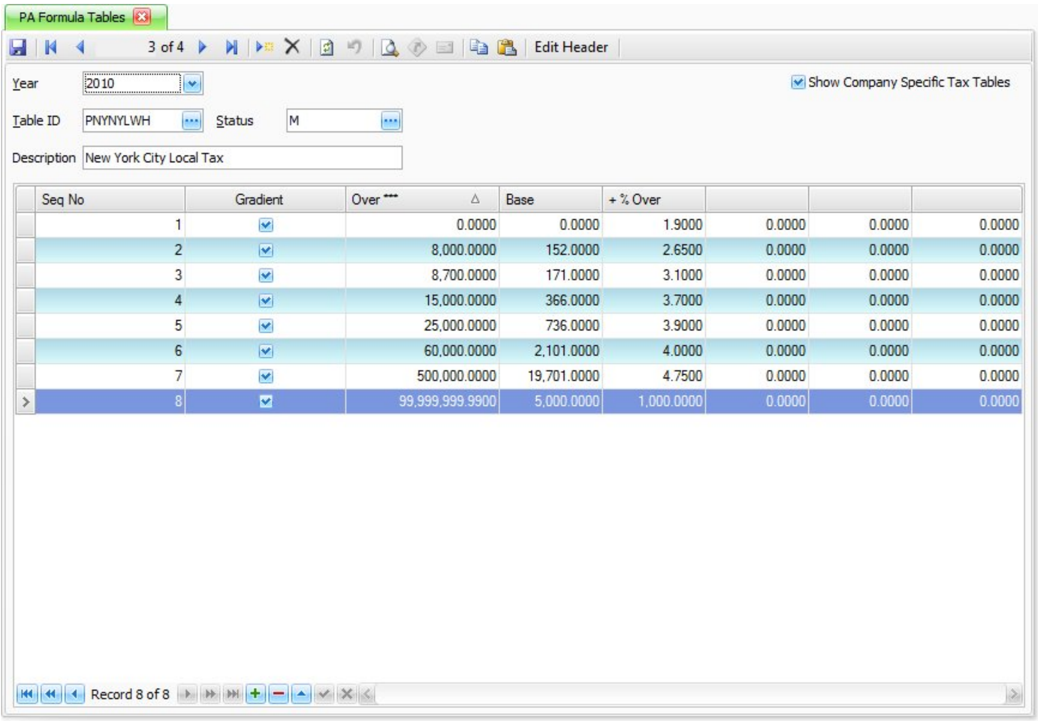
WRITING A LOCAL TAX FORMULA 1

This formula would work with a 3 column tax table similar to most State Withholding Tables, that have Wage Bracket, Base and Percent columns. This would also include a 99999999.99 row with an Exemption number in column 2.

1. Select **Formula Tables** from the **Setup and Maintenance** menu.



2. Set up your Local Table. Make sure to select the check box that says **Show Company Specific Tax Tables**. If you want to have the table put into the ST database, add the new table without the **Show company specific tables** check box selected.



Year 2010

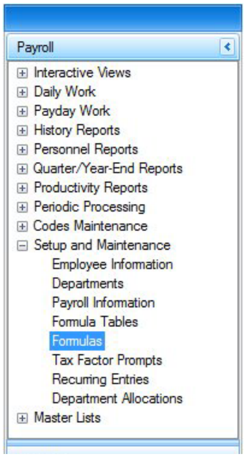
☑ Show Company Specific Tax Tables

Table ID PNYNYLWH Status M

Description New York City Local Tax

Seq No	Gradient	Over	Base	+ % Over			
1	☑	0.0000	0.0000	1.9000	0.0000	0.0000	0.0000
2	☑	8,000.0000	152.0000	2.6500	0.0000	0.0000	0.0000
3	☑	8,700.0000	171.0000	3.1000	0.0000	0.0000	0.0000
4	☑	15,000.0000	366.0000	3.7000	0.0000	0.0000	0.0000
5	☑	25,000.0000	736.0000	3.9000	0.0000	0.0000	0.0000
6	☑	60,000.0000	2,101.0000	4.0000	0.0000	0.0000	0.0000
7	☑	500,000.0000	19,701.0000	4.7500	0.0000	0.0000	0.0000
8	☑	99,999,999.9900	5,000.0000	1,000.0000	0.0000	0.0000	0.0000

3. Select **Formulas** from the **Setup and Maintenance** menu.



4. The **Formulas** screen is displayed. Select the **New Record** icon . A blank Formulas screen is displayed.

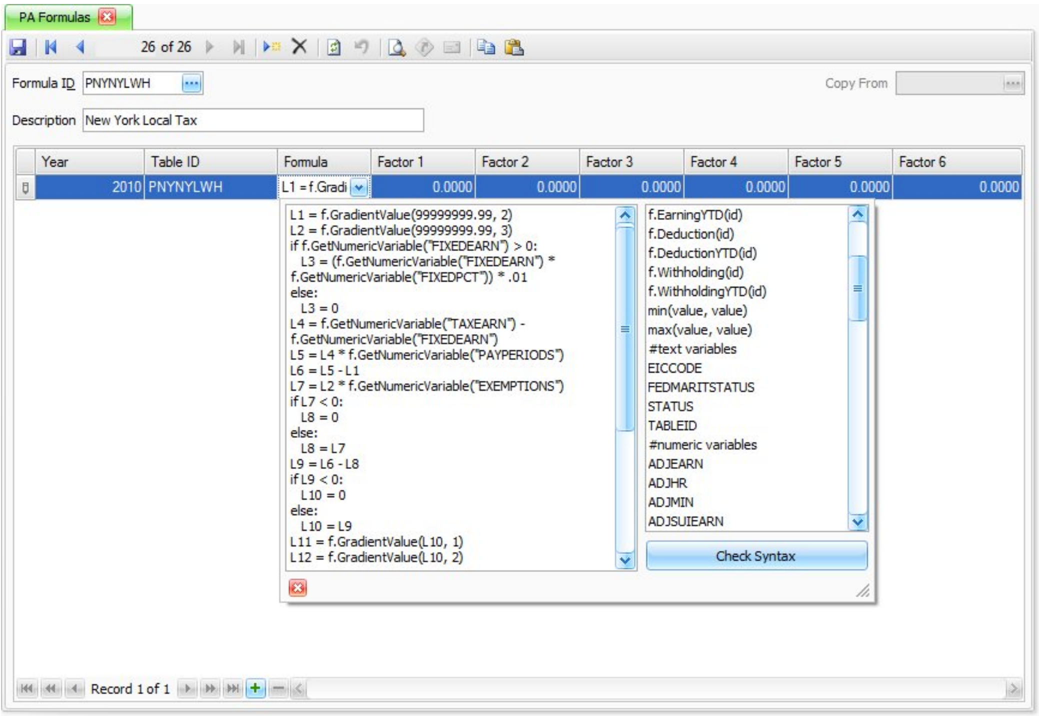
5. For the **Formula ID**, enter **PNYNYLWH**.

6. Tab to the **Description** field, and enter a Description for the Formula. For the Description, enter **New York Local Tax**.

7. Click the green plus (+) sign at the bottom of the screen to add a new Formula for the current payroll Year.

8. Select **PNYNYLWH** for the **Table ID** to use in this Formula.

9. Skip **Factors 1 – 6**.



Formula ID PNYNYLWH

Description New York Local Tax

Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2010	PNYNYLWH	L1 = f.Gradi	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

```
L1 = f.GradientValue(99999999.99, 2)
L2 = f.GradientValue(99999999.99, 3)
if f.GetNumericVariable("FIXEDEARN") > 0:
  L3 = (f.GetNumericVariable("FIXEDEARN") *
f.GetNumericVariable("FIXEDPCT")) * .01
else:
  L3 = 0
L4 = f.GetNumericVariable("TAXEARN") -
f.GetNumericVariable("FIXEDEARN")
L5 = L4 * f.GetNumericVariable("PAYPERIODS")
L6 = L5 - L1
L7 = L2 * f.GetNumericVariable("EXEMPTIONS")
if L7 < 0:
  L8 = 0
else:
  L8 = L7
L9 = L6 - L8
if L9 < 0:
  L10 = 0
else:
  L10 = L9
L11 = f.GradientValue(L10, 1)
L12 = f.GradientValue(L10, 2)
```

10. Click the drop down list arrow in the **Formula** column to display the Formula box, into which to enter the Formula.

11. For line 1, enter:

L1 = f.GradientValue(99999999.99, 2)

12. For line 2, enter:

L2 = f.GradientValue(99999999.99, 3)

13. For line 3, enter:

```
if f.GetNumericVariable(“FIXEDEARN”) > 0:
  L3 = (f.GetNumericVariable(“FIXEDEARN”) * f.GetNumericVariable(“FIXEDPCT”)) * 0.01
else:
  L3 = 0
```

14. For line 4, enter:

```
L4 = f.GetNumericVariable(“TAXEARN”) - f.GetNumericVariable(“FIXEDEARN”)
```

15. For line 5, enter:

```
L5 = L4 * f.GetNumericVariable(“PAYPERIODS”)
```

16. For line 6, enter:

```
L6 = L5 - L1
```

17. For line 7, enter:

```
L7 = L2 * f.GetNumericVariable(“EXEMPTIONS”)
```

18. For line 8, enter:

```
if L7 < 0:
  L8 = 0
else:
  L8 = L7
```

19. For line 9, enter:

```
L9 = L6 - L8
```

20. For line 10, enter:

```
if L9 < 0:
  L10 = 0
else:
  L10 = L9
```

21. For line 11, enter:

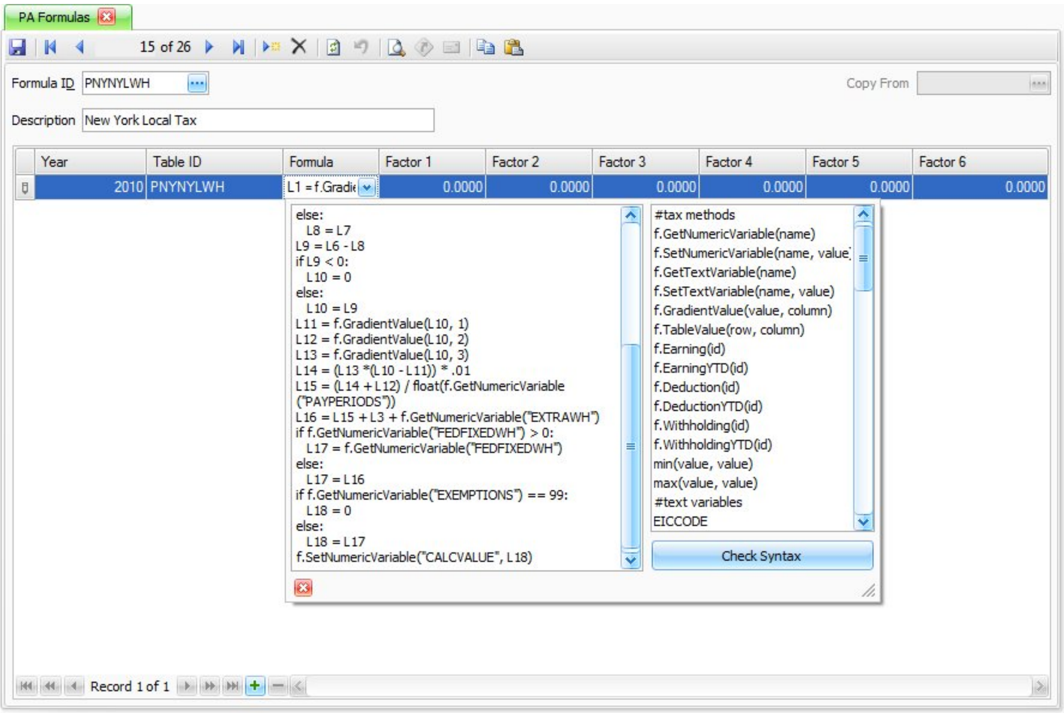
```
L11 = f.GradientValue(L10, 1)
```

22. For line 12, enter:

    L12 = f.GradientValue(L10, 2)

23. For line 13, enter:

    L13 = f.GradientValue(L10, 3)



Formula ID PNYNYLWH

Description New York Local Tax

Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2010	PNYNYLWH	L1 = f.Gradi	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

```
else:
  L8 = L7
L9 = L6 - L8
if L9 < 0:
  L10 = 0
else:
  L10 = L9
L11 = f.GradientValue(L10, 1)
L12 = f.GradientValue(L10, 2)
L13 = f.GradientValue(L10, 3)
L14 = (L13 *(L10 - L11)) * .01
L15 = (L14 + L12) / float(f.GetNumericVariable
("PAYPERIODS"))
L16 = L15 + L3 + f.GetNumericVariable("EXTRAWH")
if f.GetNumericVariable("FEDFIXEDWH") > 0:
  L17 = f.GetNumericVariable("FEDFIXEDWH")
else:
  L17 = L16
if f.GetNumericVariable("EXEMPTIONS") == 99:
  L18 = 0
else:
  L18 = L17
f.SetNumericVariable("CALCVALUE", L18)
```

24. For line 14, enter:

    L14 = (L13 *(L10 - L11)) * 0.01

25. For line 15, enter:

    L15 = (L14 + L12) / float(f.GetNumericVariable(“PAYPERIODS”))

26. For line 16, enter:

    L16 = L15 + L3 + f.GetNumericVariable(“EXTRAWH”)

27. For line 17, enter:

```
if f.GetNumericVariable(“FEDFIXEDWH”) > 0:
  L17 = f.GetNumericVariable(“FEDFIXEDWH”)
else:
  L17 = L16
```

28. For line 18, enter:

```
if f.GetNumericVariable(“EXEMPTIONS”) == 99:
  L18 = 0
else:
  L18 = L17
```

NOTE: It is important to enter two equal signs == before the 99 in the if part of the statement. This is the Iron Python comparison function.

29. For the final line enter this to return the calculated value:

```
f.SetNumericVariable(“CALCVALUE”, L18)
```

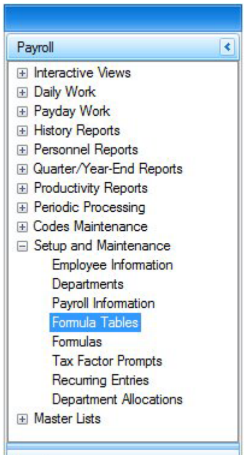
30. Set up Local Tax using **Local Tax Authority Setup** from the **Codes Maintenance** menu and follow instructions from the previous example.

31. Add Local Tax to **Employee Information**, **Taxes**, **Local Taxes** following instructions from the previous example.

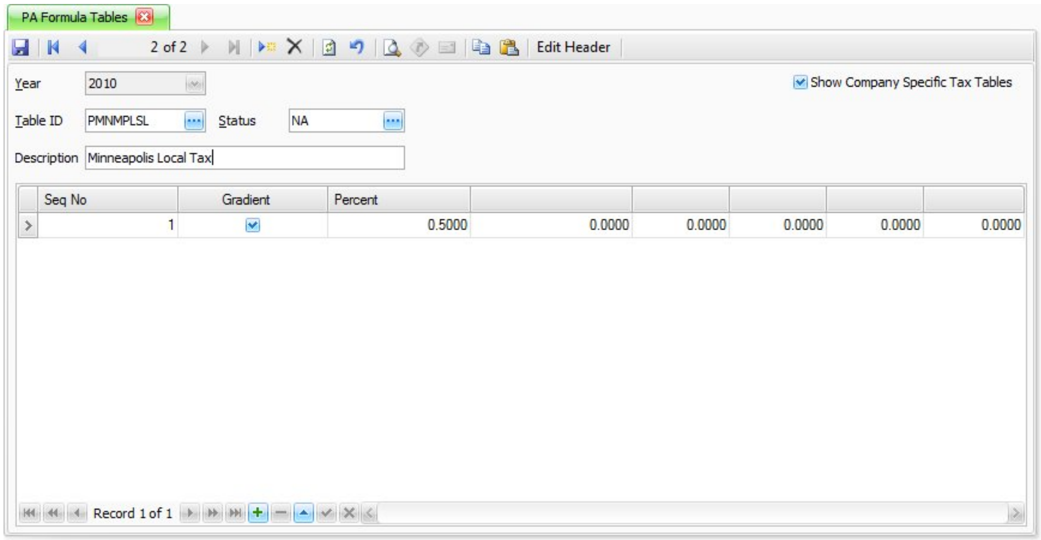
WORKING WITH A CITY TAX

With this City Tax we have a one column one row tax table. The Formula will take into account any Fixed Earnings, Fixed Percent, and Extra Withholdings the Employee has. The Formula will also allow you to put 99 for Exemptions and make the Employee exempt from this tax.

1. Select **Formula Tables** from the **Setup and Maintenance** menu.



2. Set up your Local Table. Make sure to select the check box that says **Show Company Specific Tax Tables**. If you want to put the Local Tax Table into the ST database instead of the company specific database, leave the **Show Company Specific Tax Tables** box cleared when adding the new table.



Year 2010

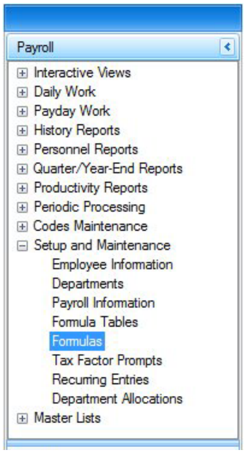
☑ Show Company Specific Tax Tables

Table ID PMNMPLSL Status NA

Description Minneapolis Local Tax

Seq No	Gradient	Percent					
1	☑	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000

3. Select **Formulas** from the **Setup and Maintenance** menu.



4. The **Formulas** screen is displayed. Select the **New Record** icon . A blank **Formulas** screen is displayed.

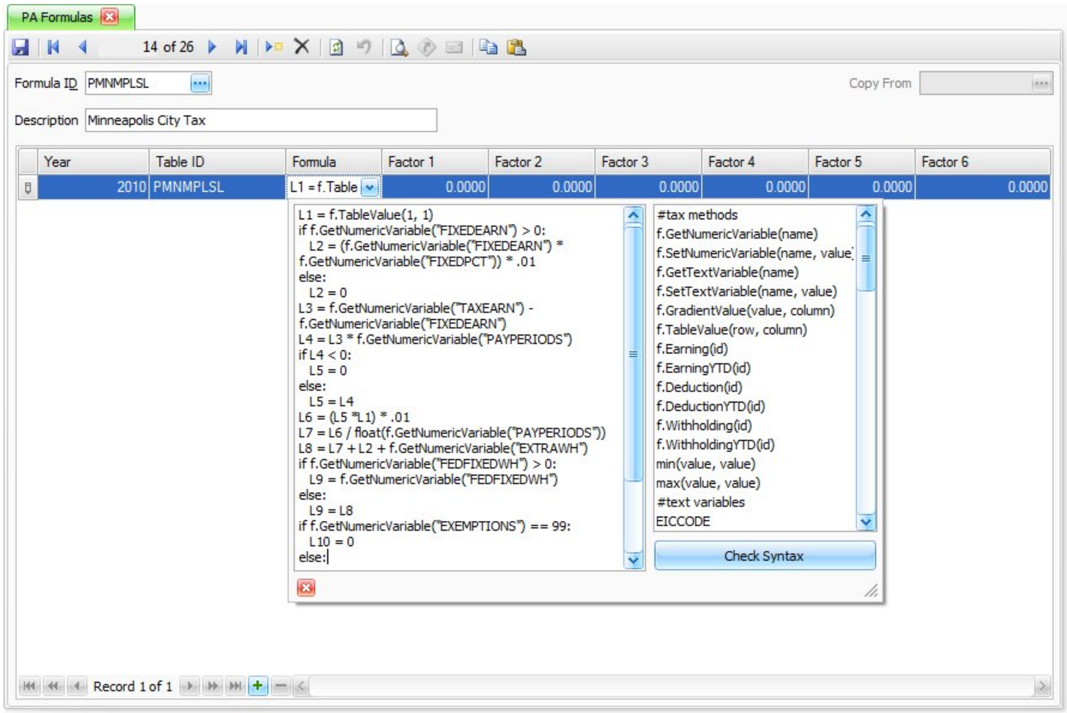
5. For the **Formula ID**, enter **PMNMPLSL**.

6. Tab to the **Description** field, and enter a Description for the Formula. For the Description, enter **Minneapolis City Local Tax**.

7. Click the green plus (+) sign at the bottom of the screen to add a new Formula for the current payroll Year.

8. Select **PMNMPLSL** for the **Table ID** to use in this formula.

9. Skip **Factors 1 – 6**.



Formula ID PMNMPLSL

Description Minneapolis City Tax

	Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
	2010	PMNMPLSL	L1 = f.Table	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

```
L1 = f.TableValue(1, 1)
if f.GetNumericVariable("FIXEDEARN") > 0:
  L2 = (f.GetNumericVariable("FIXEDEARN") *
f.GetNumericVariable("FIXEDPCT")) * .01
else:
  L2 = 0
L3 = f.GetNumericVariable("TAXEARN") -
f.GetNumericVariable("FIXEDEARN")
L4 = L3 * f.GetNumericVariable("PAYPERIODS")
if L4 < 0:
  L5 = 0
else:
  L5 = L4
L6 = (L5 *L1) * .01
L7 = L6 / float(f.GetNumericVariable("PAYPERIODS"))
L8 = L7 + L2 + f.GetNumericVariable("EXTRAWH")
if f.GetNumericVariable("FEDFIXEDWH") > 0:
  L9 = f.GetNumericVariable("FEDFIXEDWH")
else:
  L9 = L8
if f.GetNumericVariable("EXEMPTIONS") == 99:
  L10 = 0
else:
```

10. Click the drop down list arrow in the **Formula** column to display the Formula box, into which to enter the Formula.

11. For line 1, enter:

L1 = f.TableValue(1, 1)

12. For line 2, enter:

if f.GetNumericVariable("FIXEDEARN") > 0:

L2 = (f.GetNumericVariable("FIXEDEARN") * f.GetNumericVariable("FIXEDPCT")) * 0.01

else:

L2 = 0

13. For line 3, enter:

    ```
    L3 = f.GetNumericVariable(“TAXEARN”) - f.GetNumericVariable(“FIXEDEARN”)
    ```

14. For line 4, enter:

    ```
    L4 = L3 * f.GetNumericVariable(“PAYPERIODS”)
    ```

15. For line 5, enter:

    ```
    if L4 < 0:
      L5 = 0
    else:
      L5 = L4
    ```

16. For line 6, enter:

    ```
    L6 = (L5 *L1) * 0.01
    ```

17. For line 7, enter:

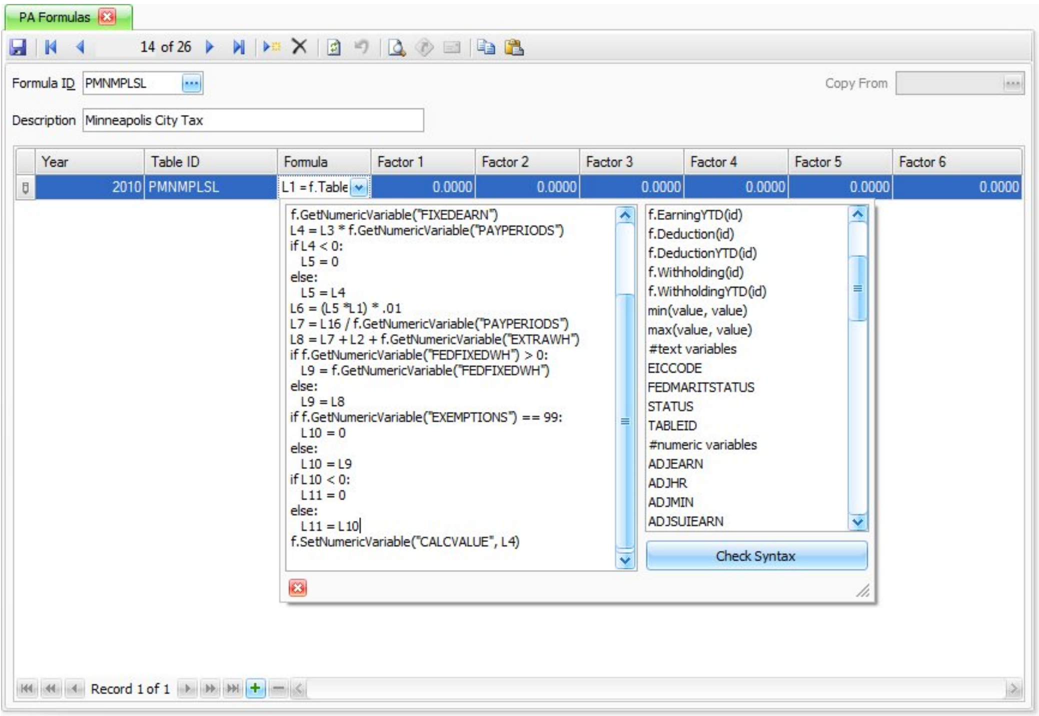
    ```
    L7 = L6 / float(f.GetNumericVariable(“PAYPERIODS”))
    ```

18. For line 8, enter:

    ```
    L8 = L7 + L2 + f.GetNumericVariable(“EXTRAWH”)
    ```

19. For line 9, enter:

```
if f.GetNumericVariable(“FEDFIXEDWH”) > 0:
  L9 = f.GetNumericVariable(“FEDFIXEDWH”)
else:
  L9 = L8
```



Formula ID: PMNMPLSL

Copy From:

Description: Minneapolis City Tax

Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2010	PMNMPLSL	L1 = f.Table	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

```
f.GetNumericVariable("FIXEDEARN")
L4 = L3 * f.GetNumericVariable("PAYPERIODS")
if L4 < 0:
  L5 = 0
else:
  L5 = L4
L6 = (L5 *L1) * .01
L7 = L16 / f.GetNumericVariable("PAYPERIODS")
L8 = L7 + L2 + f.GetNumericVariable("EXTRAWH")
if f.GetNumericVariable("FEDFIXEDWH") > 0:
  L9 = f.GetNumericVariable("FEDFIXEDWH")
else:
  L9 = L8
if f.GetNumericVariable("EXEMPTIONS") == 99:
  L10 = 0
else:
  L10 = L9
if L10 < 0:
  L11 = 0
else:
  L11 = L10
f.SetNumericVariable("CALCVALUE", L4)
```

```
f.EarningYTD(id)
f.Deduction(id)
f.DeductionYTD(id)
f.Withholding(id)
f.WithholdingYTD(id)
min(value, value)
max(value, value)
#text variables
EICCODE
FEDMARITSTATUS
STATUS
TABLEID
#numeric variables
ADJEARN
ADJHR
ADJMIN
ADJSUIEARN
```

20. For line 10, enter:

```
if f.GetNumericVariable(“EXEMPTIONS”) == 99:
  L10 = 0
else:
  L10 = L9
```

NOTE: It is important to enter two equal signs == before the 99 in the if part of the statement. This is the Iron Python comparison function.

21. For line 11, enter:

```
if L10 < 0:
  L11 = 0
else:
  L11 = L10
```

22. For the final line enter this to return the calculated value:

```
f.SetNumericVariable("CALCVALUE", L11)
```

23. Set up the city tax using **Local Tax Authority Setup** from the **Codes Maintenance** menu and follow instructions from previous example.

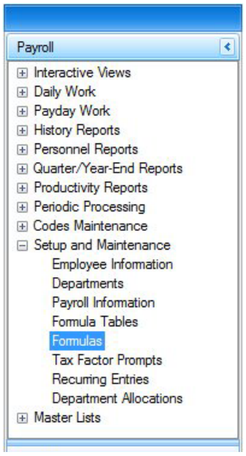
24. Add the city tax to **Employee Information**, **Taxes**, **Local Taxes** following instructions from previous examples.

WRITING A CITY TAX FORMULA 2

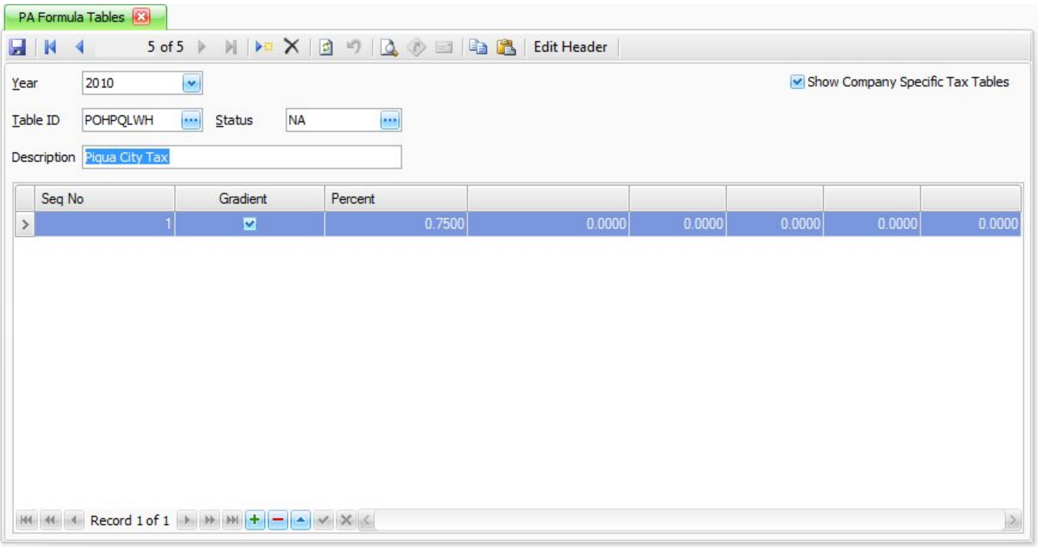
This city tax uses the same type of table as the previous city tax example, but the formula is a little more simple. The formula will account for fixed earnings only and then take out the tax percentage.

To make this City Tax, do the following steps:

1. Select **Formula Tables** for the **Setup and Maintenance** menu.



2. Set up your Local Table. Make sure to select the check box that says **Show Company Specific Tax Tables**. Call the **Table ID POHPQLWH,** and type in the **Description** of **Piqua City Tax.** If you want to set up the tax table in the ST database instead of the company database, set the table up with the **Show Company Specific Tax Tables** box cleared.



Year 2010

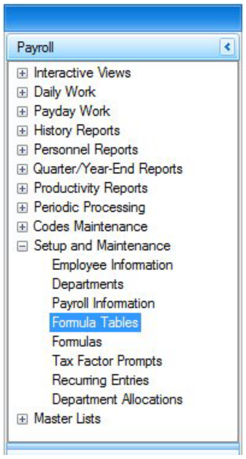
Table ID POHPQLWH   Status NA

Description Piqua City Tax

☑ Show Company Specific Tax Tables

Seq No	Gradient	Percent					
1	☑	0.7500	0.0000	0.0000	0.0000	0.0000	0.0000

3. Select **Formulas** from the **Setup and Maintenance** menu.



4. The **Formulas** screen is displayed. Select the **New Record** icon . A blank **Formulas** screen is displayed.

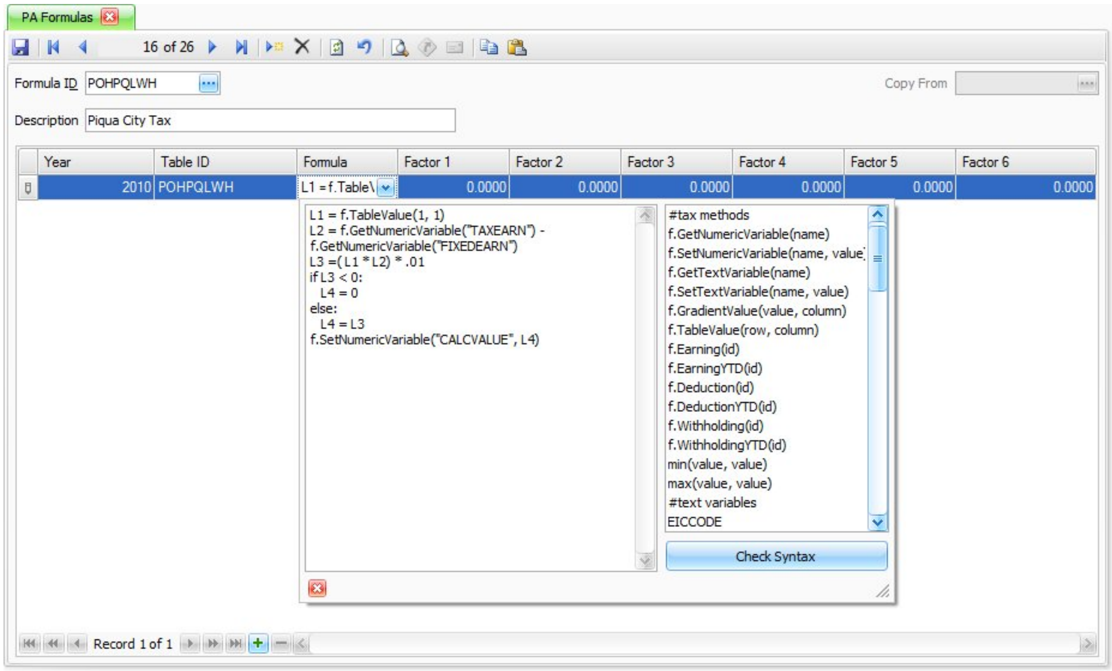
5. For the **Formula ID**, enter **POHPQLWH.**

6. Tab to the **Description** field, and enter a Description for the Formula. For the Description, enter **Piqua City Tax.**

7. Click the green plus (+) sign at the bottom of the screen to add a new Formula for the current payroll Year.

8. Select **POHPQLWH** for the **Table ID** to use in this Formula.

9. Skip **Factors 1 - 6**.



Formula ID POHPQLWH

Description Piqua City Tax

Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2010	POHPQLWH	L1 =f.Table\	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

```
L1 = f.TableValue(1, 1)
L2 = f.GetNumericVariable("TAXEARN") -
f.GetNumericVariable("FIXEDEARN")
L3 =( L1 * L2) * .01
if L3 < 0:
  L4 = 0
else:
  L4 = L3
f.SetNumericVariable("CALCVALUE", L4)
```

10. Click the drop down list arrow in the **Formula** column to display the Formula box, into which to enter the Formula.

11. For line 1, enter:

L1 = f.TableValue(1, 1)

12. For line 2, enter:

L2 = f.GetNumericVariable("TAXEARN") - f.GetNumericVariable("FIXEDEARN")

13. For line 3, enter**:**

L3 =(L1 * L2) * 0.01

14. For line 4, enter:

```
if L3 < 0:
  L4 = 0
else:
  L4 = L3
```

15. Enter the following as the last line to return the value calculated:

```
f.SetNumericVariable(“CALCVALUE”, L4)
```

16. Set up the city tax using **Local Tax Authority Setup** from the **Codes Maintenance** menu and follow instructions from the previous examples.

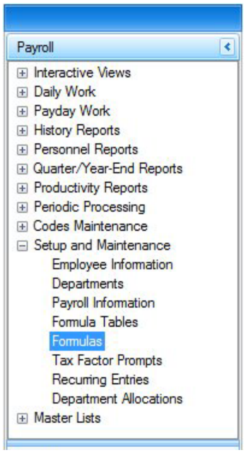
17. Add the city tax to **Employee Information**, **Taxes**, **Local Taxes** following instructions from the previous examples.

WRITING A SIMPLE LOCAL TAX FORMULA

This Local Tax example is the simplest tax Formula that can be written. This Formula uses no Tax Table and does not take into account any Fixed Percent, Fixed Withholdings, or Exemptions set up for the Employees. This Formula will calculate a Fixed Percent, from a Factor, and calculate that percent from their taxable income.

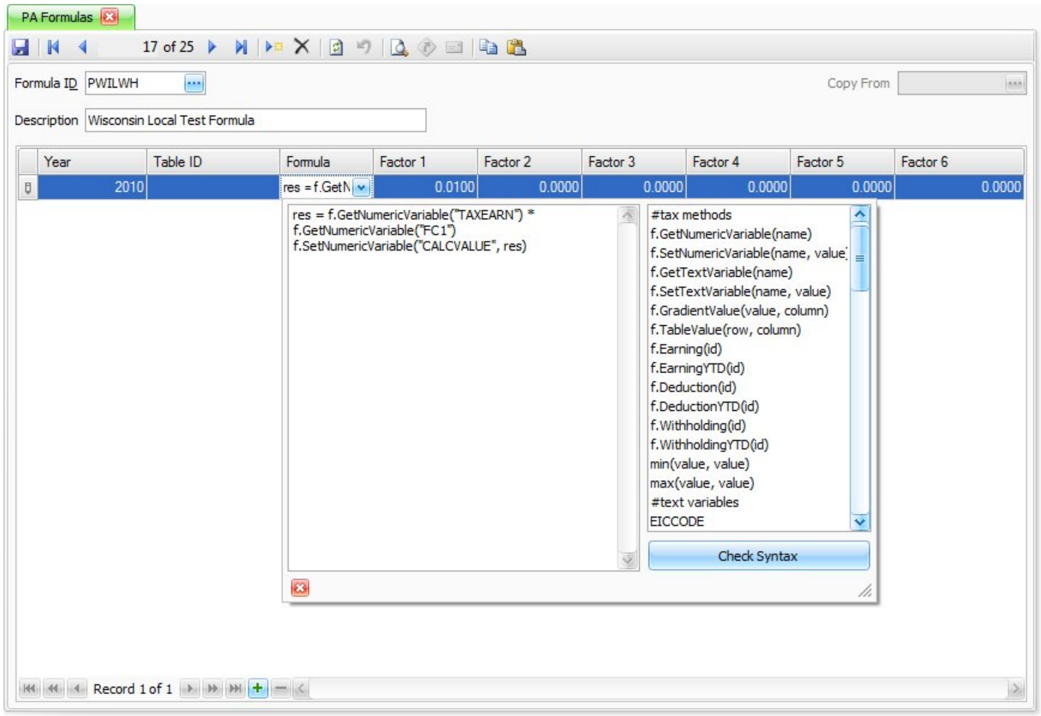
To make this Local Tax, following these steps:

1. Select **Formulas** for the **Setup and Maintenance** menu.



2. The **Formulas** screen is displayed. Select the **New Record** icon ▶✳ . A blank **Formulas** screen is displayed.
3. For the **Formula ID**, enter **PWILWH.**
4. Tab to the **Description** field, and enter a Description for the Formula. For the Description, enter **Wisconsin Local Test Formula.**
5. Click the green plus (+) sign at the bottom of the screen to add a new Formula for the current payroll Year.

6. For **Factor 1** enter 0.01. Skip **Factors 2 - 6**.



Formula ID: PWILWH

Description: Wisconsin Local Test Formula

Copy From:

Year	Table ID	Formula	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
2010		res = f.GetN	0.0100	0.0000	0.0000	0.0000	0.0000	0.0000

```
res = f.GetNumericVariable("TAXEARN") *
f.GetNumericVariable("FC1")
f.SetNumericVariable("CALCVALUE", res)
```

```
#tax methods
f.GetNumericVariable(name)
f.SetNumericVariable(name, value)
f.GetTextVariable(name)
f.SetTextVariable(name, value)
f.GradientValue(value, column)
f.TableValue(row, column)
f.Earning(id)
f.EarningYTD(id)
f.Deduction(id)
f.DeductionYTD(id)
f.Withholding(id)
f.WithholdingYTD(id)
min(value, value)
max(value, value)
#text variables
EICCODE
```

7. Click the drop down list arrow in the **Formula** column to display the Formula box, into which to enter the Formula.

8. For line 1, enter:

res = f.GetNumericVariable("TAXEARN") * f.GetNumericVariable("FC1")

9. For the second line to return the value calculated enter:

f.SetNumericVariable("CALCVALUE", res)

The value **res** is considered the line number in this case and that is the value that is being returned with the final statement. This is another example of how line number values can vary.

10. Set up the Local Tax using **Local Tax Authority Setup** from the **Codes Maintenance** menu and follow instructions from the previous examples.

11. Add the Local Tax to **Employee Information**, **Taxes**, **Local Taxes** following instructions from the previous examples.