

Strategic Series-7003

# **Derived Fields**

Version 9.0

---

Information in this document is subject to change without notice and does not represent a commitment on the part of Technical Difference, Inc. The software product described in this document is furnished under a license agreement or nondisclosure agreement. The software and this documentation may be used or copied only in accordance with the terms of that agreement.

All names of companies, products, street addresses, and persons are part of a completely fictitious scenario and are designed solely to document the use of People-Trak. Similarities to real companies, products, addresses, or persons are purely coincidental.

People-Trak is an extensively customizable HR software product. Screen and report samples rendered in this document reflect the default version of People-Trak. These samples may or may not match the screens and reports within your product if customization has been performed.

(C) Copyright Technical Difference, Inc., 2014

Technical Difference, Inc.  
5256 S. Mission Road #210  
Bonsall, CA 92003  
(800) 273-3760  
(866) 693-4869 (fax)

[www.people-trak.com](http://www.people-trak.com)

(Revision 02/01/2014)

---

# Contents

<b>Workbook</b>	<b>1</b>
Derived Fields	1
Derived Field Formulas	3
Derived Field Operators	4
Rounding Derived Fields	7
Literals in Derived Fields	8
Derived Field SQL Statements	9
Testing Derived Fields	9
<b>Lesson 1 Derived Fields 1</b>	<b>11</b>
Tutorial 1.1 – Create a Simple Derived Field	12
Tutorial 1.2 – Use a Derived Field	15
<b>Lesson 2 Derived Fields 2</b>	<b>17</b>
Tutorial 2.1 – Math Operators	18
Tutorial 2.2 – Rounding Methods	21
Tutorial 2.3 – String Operators	23
<b>Lesson 3 Derived Fields 3</b>	<b>25</b>
Tutorial 3.1 – Date Operators	26
Tutorial 3.2 – Complex Fields	29
Tutorial 3.3 – SQL Statement Derived Fields	31

---

## Notes

---

# Workbook

---

## Derived Fields

Derived fields can be custom built by you to fulfill requirements specific to your company. As their name implies, these fields are formulas that derive their definition from existing People-Trak fields. Derived fields can be used for various applications. Suppose your company needs a field to display both the First Name and Last Name fields. You can simply add both of those fields to the formula, and the resulting derived field would be a First and Last Name field.

Another simple example of a derived field is a calculated one. Suppose you want to know what an employee's pay would be if given a four-percent increase. You would enter the following formula:

`~SAL_ANNUAL~ * 1.04`

“SAL\_ANNUAL” is the Field ID for the Annual Salary field. It is surrounded by tildes (~) to denote it is a field. The “\*” is the Multiplication operator and the “1.04” is the number to multiply by.

Derived fields are not stored in the database; they are derived when they are used on a document or screen. They can be placed on custom screens where they have display-only status. You can also use a derived field in the formula of another derived field. Derived fields are treated just like other fields and can be protected through field security in Security Setup.

The Derived Fields screen provides a list of the current derived fields on the left and a data entry area to the right. A Menu Bar and a Tool Bar are provided to enable you to fully manage your list of derived fields. The following is a sample of the Derived Fields screen.

The screenshot shows a window titled "Derived Fields" with a menu bar (File, Edit, Help) and a toolbar. On the left is a list of derived fields, with "Address Block" selected. On the right, the configuration for "Address Block" is shown:

Field ID	DERIVED_204	<input type="button" value="Edit Help"/>
Field Name	Address Block	
Category	Personal	
Field Style	Formula	
Caption	Address Block	
Field Type	Text	
Edit Mask/Decimals	None	
Formula	<code>~NAME_FULL~ crlf ~ADDRESS_1~, ~ADDRESS_2~ crlf ~CityStateZip~</code>	

Below the formula field are buttons for "Field", "Date Variable", "Operator", "Rounding", and "Test Field". At the bottom is a "Sample" text area.

Upon creating a derived field, the Field ID is system generated and is for display only. Other initial details include defining a field name and caption. Additional settings for the derived field are listed and defined below.

## Field Style

The style of the field: Formula or SQL Statement.

The Formula style allows you to use the Field, Date Variable, Operator and Rounding buttons to build a formula that generates the derived field. The SQL Statement style allows you to enter an SQL Statement to generate the derived field.

## Field Type

The type of the field. Below are the field types that can be selected.

- Currency
- DateTime
- Numeric
- Text.

**Note:** The field type selected determines which Edit Mask or Decimal options are available.

The Currency field type allows you to select the number of decimal places in the data. The selection ranges from 0-4 decimal places.

The DateTime field type allows you to select either Date or Time. The Date option will format the field data in the Date format, while the Time option will set the data to the Time format

The Numeric field type also allows you to select the number of decimal places you'd like in the data. Just like the Currency field type, the decimal selection ranges from 0-4 decimal places. The Numeric type fields should be used for number values only. Fields that contain numbers as well as parentheses or dashes must be entered in text fields.

There are several important options for the Text field type. Each mask will edit the default database settings of the fields. Below are the options for the Text field type:

- Email
- Hyperlink
- None
- Proper Case
- Social Security
- Telephone
- Upper Case

The Email mask will check that the text entered in the field is in the valid email address format.

The Hyperlink option defines this field as hyperlinked, which then gives you a popup button on the field to execute the hyperlink.

There will be no format when the None option is selected. If you want the field to stay formatted just as it is in the database, then you should select this mask. Derived fields use this mask by default.

The Proper Case mask will ensure that the field is formatted in proper case. All entries will have their first letters in upper case to denote a proper noun.

The Social Security mask formats the field into the social security number format.

The Telephone mask formats the field into the telephone number format.

The Upper Case mask will simply make all letters in the field upper case.

## Edit Mask/ Decimals

Edit Mask/Decimals allows you to select a layout or format for the derived field. The mask for each field will dictate the format of the field data. Below are the options you will get for each Field Type.

## Derived Field Formulas

A well-based understanding of the derived field formulas is needed in order to use the Derived Field feature to its full potential. Simply put, the derived field formulas work with derived fields just like a chemistry equation works with a complex chemical solution. The formula must be in the right order and use the correct syntax in order to function correctly. You don't want the chemicals to explode in your face.

Most derived field formulas will be made up of combinations of two or more People-Trak fields (or other derived fields). Other formulas may have only one field and some other variable, such as a manually keyed in number or letter. Some formulas will include literals, which must be included in single quotes. No matter what the components, every formula will follow a certain syntax.

If the Field Style is Formula, you can enter the formula manually or use the Field, Date Variable, Operator and Rounding buttons to build the formula. These fields are described below.

### Field

Displays a list of all available fields in the current module. You can then select the field to include in the formula. When the field is returned to the formula, it is returned in the form of the field ID (the actual name of the field in the database). In addition, a tilde (~) is added just before and just after the field ID. These characters act as delimiters to let People-Trak know that they are fields and not operators, rounding methods or values. If you choose to manually type in a field, you need to enter the field ID with the tilde characters around it. This button is only available with a Field Style of Formula.

**Note:** If you want to reference a specific row in a grid, enter ':row' after the field name and before the tilde (~), where row = the row number or '+' for the last row. For example, to pull an employee's previous annual salary, use ~SAL\_ANNUAL:2~. To pull an employee's very first annual salary (the last row), use ~SAL\_ANNUAL:+~.

### Date Variable

Displays a list of the available date variables. You can then select a date variable to include in the formula. When the date variable is returned to the formula, it is returned with a tilde (~) just before and just after it. If you choose to manually type in a date variable, you need to enter the date variable with the tilde characters around it. This button is only available with a Field Style of Formula.

## Operator

Displays a list of available operators. You can then select an operator to include in the formula. This button is only available with a Field Style of Formula.

## Rounding

Displays a list of valid rounding methods. You can then select the rounding method to use in the formula. This button is only available with a Field Style of Formula.

# Derived Field Operators

Operators, as explained below, are the key to useful formulas and help form the syntax of every derived field. Derived field formulas rest on the correct use of operators. Operators are the different relationships or bonds between two fields. Operators take the value of the fields and cause them to act the way the operator demands. A general knowledge of how operators work is essential to creating formulas. Below is an overview of each operator, explaining how each is used.

When adding a new operator that includes a formula as part of it (for example DateDiff, DateAdd, Left, Right, etc.) or using rounding, the components of that formula are included as placeholders in order to help you with the syntax of the formula. You can then replace the placeholders with the actual fields or literals.

## Addition/Subtraction

The Addition operator (+) and Subtraction operator (-) are used to find the sum or difference of the first and second numbers. For example, your Annual Pay (50,000) – 10,000 will equal 40,000. The actual formula in this case is:

`~SAL_ANNUAL~ - 10000`

If you wanted to find the difference between someone's current salary and their previous salary, the formula would be the following, where ':2' is added to the field name to specify the row modifier for the previous salary:

`~SAL_ANNUAL~ - ~SAL_ANNUAL:2~`

Or, if you wanted to find the difference between someone's current salary and their very first salary (last row, when current is in first row), the formula would be the following, where ':+' is added to the field name to specify the row modifier for the last salary in the grid:

`~SAL_ANNUAL~ - ~SAL_ANNUAL:+~`

## Multiplication/Division

The Multiplication operator (\*) and Division operator (/) operate similarly to the addition and subtraction operators by multiplying or dividing the first and second numbers in your field. For example, to find your Annual Pay (53,024) after 2 years, you could set the formula to Annual Pay \* 2. It would equal 106,048. The actual formula in this case is:

`~SAL_ANNUAL~ * 2`

## Concatenation

The Concatenation operator (&) simply combines the first and second field side by side. It is not required, but can be added for readability.

The following formula creates a full name, with a space between each part of the name:

*~NAME\_FIRST~ & ~NAME\_MI~ & ~NAME\_LAST~*

The same result can be accomplished without the concatenation operator by simply using the following formula:

*~NAME\_FIRST~ ~NAME\_MI~ ~NAME\_LAST~*

## **crlf**

The crlf operator drops the second field a line lower in the text box. Crlf stands for Carriage Return Line Feed. The following formula puts the Full Name field on the first line and the Address 1 field on the second line:

*~NAME\_FULL~ crlf ~ADDRESS\_1~*

## **DateAdd**

The DateAdd operator is used to find a date that is a certain time away from another date. You can select whether you would like the difference to be in days, months, or years by selecting the correct suffix: D (days), M (months) or Y (years).

The equation for DateAdd is:

*DateAdd-x (#, date)*

where:

*x* = D (days), M (months) or Y (years)

*#* = the number of days, months or years

*date* = date or date variable to which to add the number of days, months or years

The following formula adds 3 days to today's date:

*DateAdd-D(3,~TODAY~)*

## **DateDiff**

The DateDiff operator is used to find the difference between two dates. You can select whether you would like the difference to be in days, months, or years by selecting the correct suffix: D (days), M (months) or Y (years).

The equation for DateDiff is:

*DateDiff-x (from date, to date)*

where:

*x* = D (days), M (months) or Y (years)

*from date* = date or date variable from which to find the difference from the *to date*

*to date* = date or date variable to which to find the difference from the *from date*

The following formula returns the number of years an employee has been working for the company:

*DateDiff-Y(~Hire\_Date~, ~TODAY~)*

## FlipName

The FlipName operator is used to flip a name field that has been entered as a full name and displays it in Last Name, First Name format.

The equation for FlipName is:

$\text{FlipName}(\sim\text{NameField}\sim)$

where:

$(\sim\text{NameField}\sim)$  = a People-Trak name field

The following formula “flips” the name of the primary emergency contact.

$\text{FlipName}(\sim\text{EM1}\sim)$

## LCase

The LCase operator is used to convert the string to lowercase.

The equation for LCase is:

$\text{LCase}(\sim\text{field}\sim)$

where:

$\text{field}$  = People-Trak field

## Left

The Left operator is used to return the leftmost characters of a string. You specify how many characters to return.

The equation for Left is:

$\text{Left}(\sim\text{field}\sim, \# \text{ of chars})$

where:

$\text{field}$  = People-Trak field

$\# \text{ of chars}$  = the number of characters to return, starting from the left side of the string

The following formula returns the middle initial for an employee:

$\text{Left}(\sim\text{NAME\_MI}\sim, 1)$

## Mid

The Mid operator is used to return part of a string. You specify what character to start with and how many characters to return.

The equation for Mid is:

$\text{Mid}(\sim\text{field}\sim, \text{start char}, \# \text{ of chars})$

where:

$\text{field}$  = People-Trak field

$\text{start char}$  = the starting character position

*# of chars* = the number of characters to return, starting from the starting character position

The following formula returns the middle initial for an employee:

*Mid(~NAME\_MI~, 1, 1)*

## Right

The Right operator is used to return the rightmost characters of a string. You specify how many characters to return.

The equation for Right is:

*Right(~field~, # of chars)*

where:

*field* = People-Trak field

*# of chars* = the number of characters to return, starting from the right side of the string

The following formula returns the last four digits of a SSN for an employee:

*Right(~SS\_NO~, 4)*

## UCase

The UCase operator is used to convert the string to uppercase.

The equation for UCase is:

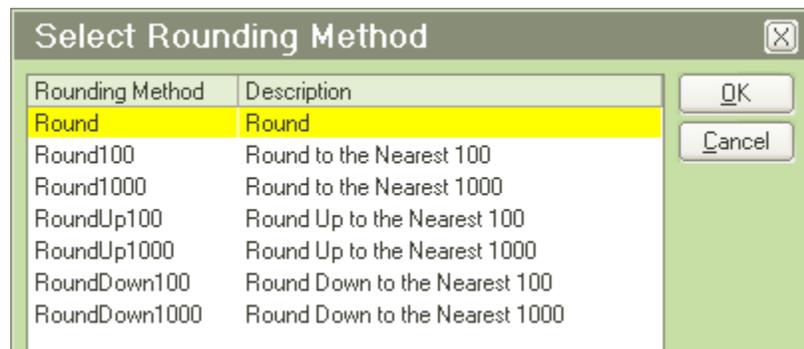
*UCase(~field~)*

where:

*field* = People-Trak field

## Rounding Derived Fields

The Rounding feature in derived fields allows you to round numeric fields or the results of numeric equations. You have the options of simply rounding to the nearest 100 or 1000 or rounding up or down to the nearest 100 or 1000. The RoundUp and RoundDown options are especially useful when simple rounding is not specific enough.



Say you want a salary raise to be rounded UP by the nearest \$100 and not simply rounded to the nearest \$100. If you entered the figure \$225, Round100 would round down to the nearest \$100 which happens to be \$200. However, if you set the formula to RoundUp100, the sum would be rounded to the next \$100, which is \$300. The RoundUp and RoundDown methods

are forced roundings, whereas the Round feature does the normal rounding to which we are accustomed.

The Round feature, as opposed to Round100, rounds to the nearest whole number. It is best used to round decimal numbers such as 1.32. Using Round, 1.32 will be rounded to 1.

The formula for rounding can differ depending on the situation but the following are two examples:

*Round100 (~field~)*

where:

*field* = a numeric field to be rounded to the nearest 100

*Round1000 (~field1~ \* ~field2~)*

where:

*field1* and *field2* = two numeric fields to be multiplied together, and the result to be rounded to the nearest 1000.

## Literals in Derived Fields

Most literals can be included in derived fields by simply adding them to the formula. For example, the 'Name: ' text can be added as shown:

*Name: ~NAME\_FIRST~ ~NAME\_MI~ ~NAME\_LAST~*

All literals can be included in single quotes, but if you need to use literals that are also used as mathematical operators, you need to use single quotes to tell the system that you want to use them as literals not operators.

For example, the following formula will result in the social security number being masked except for the last four digits:

*'\*\*\*\_\*\*\_~Right(~SS\_NO~,4)*

The result is:

*\*\*\*\_\*\*-4592*

Note that there was no space before the word 'Right'. This is because no space was wanted.

When you use the formula buttons, a space is automatically added for you when the appropriate syntax is added to the formula. You can either delete that space or add more.

**Note:** If you choose to add spaces to a formula, the spaces will be taken literally. For example, after adding two spaces after the second tilde to the above formula:

*'\*\*\*\_\*\*\_~Right(~SS\_NO~ , 4)*

Your result is:

*\*\*\*\_\*\*-92*

The two spaces that were added were counted as two of the four digits.

All literals can be included in single quotes, but the single quotes are only required when mathematical operators are to be used as literals.

---

## Derived Field SQL Statements

If the Field Style is SQL Statement, the statement is entered in the Formula field. The Field, Date Variable, Operator and Rounding buttons are not available. This field is supported by the multi-line text popup, mainly for use with a long SQL statement.

**Note:** In SQL Statements, 'XXX' can be used to specify the location of the 'Where ID = ##' logic. For example, 'Select Sal\_Rate From SalHistory XXX And Seq = 3' becomes 'Select Sal\_Rate From SalHistory Where ID = ## And Seq = 3'.

## Testing Derived Fields

You should always test the derived field before you put it into use. In order to see if your derived field formula works correctly you must use the Test Field feature. The Test Field feature allows you to apply your derived field formula to a record of your choosing and see the result.

When you click the Test Field button, a Lookup screen will appear. You must select the record you wish to use in the derived field test. The Sample field will then display the result of the derived field formula for that record. Using the Test Field feature will not affect the record's actual data. The test only temporarily displays what your derived field will look like.

**Note:** The record you select must contain relevant data in all the fields included in the derived field formula. Choosing a record without the necessary data will result in a flawed test and the Sample field will be blank.

## Notes

---

# Lesson 1

## Derived Fields 1

---

### Contents

Tutorial 1.1 – Create a Simple Derived Field	12
Tutorial 1.2 – Use a Derived Field	15

---

This lesson covers how to create and use a simple derived field. Derived fields are fields that are derived from one or more existing fields (including existing derived fields) within People-Trak.

### Tutorials

**Create a Simple Derived Field:** shows how to create a simple derived field.

**Use a Derived Field:** demonstrates how to use the derived field you created in the previous tutorial.

---

## Tutorial 1.1 – Create a Simple Derived Field

Derived fields are fields that are derived from one or more existing fields (including existing derived fields) within People-Trak.

1. On the **Tools** menu, select **Derived Fields**.

The Derived Fields screen is displayed. The Derived Fields screen provides a list of the current derived fields to the left and a data entry area to the right. A Menu Bar and a Tool Bar are provided to enable you to fully manage your list of derived fields. Let's get started and create a derived field.

2. On the **File** menu, select **New Derived Field** or on the Tool Bar, click the **New Derived Field** button to start a new derived field.

A default name is assigned. The name is NEW-#, where “#” is the next available, unused number.

3. Change the name of the field to **Long Name**.

This is the name that you will use when you select this field for use in a report or custom screen.

When the field was created, the Category field was set to User. You can change the category to one of the existing categories. The category is used for security purposes and for determining where the derived field will be shown in the Select Field popup. We will leave the category for this derived field set to User.

4. Leave the **Category** as **User**.

When the field was created, the Field Style was set to Formula. We will be building a formula in this example, so we will leave it as that.

5. Leave the **Field Style** as **Formula**.

When the field was created, the Caption was set to New Field. The Caption is displayed as a column heading when the field is used on a report and as a field caption when the field is used on a custom screen.

6. In the **Caption** field, enter **Long Name**.

When the field was created, the Field Type was set to Text. There are other field types, but you can leave this as Text for now.

7. Leave the **Field Type** as **Text**.

8. In the **Edit Mask/Decimals** field, click the popup button to display a list of the valid text formats.

These formats are pretty self-explanatory.

9. Select the default of **None** because our new field will not utilize any special formatting.

Now it's time to create a formula. Now that your cursor is positioned to the Formula field, the first four buttons below the field are activated. These buttons are used when

creating Formula style derived fields. The last button, Test Field, is always available so you can test without being in the Formula field. We'll learn about each as this lesson progresses.

The Field button is used to display a list of all the available fields in the current module. The screen that is displayed is the same screen used when selecting fields for reports or custom screens. Let's start our formula by using that button.

10. In the **Formula** field, click the **Field** button and select the **First Name** field.

When the field is returned to the formula, it is returned in the form of the field ID (the actual name of the field in the database) and not the more meaningful field name that you normally use. In addition, a tilde (~) is added just before and just after the field ID. These characters act as delimiters to let People-Trak know that these are fields and not operators or rounding methods. If you choose to manually type in a field, you need to enter the field ID and the tilde characters. You will also want to include a space after that closing tilde if you do want one after the field. You can add as many spaces as you want between components of derived fields and those spaces will be in the result. A space is automatically added for you when you use any of the formula buttons, but you can remove it if you choose.

At the moment our derived field is merely a copy of the already existing employee first name field. We need to do something to modify the field or to derive something else from it. That is, after all, the purpose of derived fields. We are going to create a derived field that contains the first name, middle name and last name of an employee.

Let's select the second and third fields.

11. Click the **Field** button and select the **Middle Name** field.
12. Click the **Field** button again and select the **Last Name** field.

The formula should look like this:

`~NAME_FIRST~ ~NAME_MI~ ~NAME_LAST~`

**Note:** Remember to make sure that you have a space between each field.

13. We have now created a full name field that does not abbreviate the middle name into an initial. That is why we named the field Long Name.

We will learn about the Date Variable, Operator and Rounding buttons in a bit. The Test Field button is used to try your derived field to see if what you get is what you want. You can do this without leaving the field.

14. Click the **Test Field** button.

The Lookup Employee screen is displayed.

15. Select **Donald Stern**.

The result of your formula for Donald Stern's record is shown in the Sample field, which is for display only. It should be his first name, followed by his middle name, followed by his last name.

If it all looks right, you have successfully created a derived field.

16. Close the **Derived Fields** screen, saving your changes.
17. Continue with the next tutorial.

---

## Tutorial 1.2 – Use a Derived Field

Derived fields can be used in reports, templates, custom screens, and other derived fields. We'll learn how to do each of these in this tutorial.

1. Create a new **Adhoc Report** and name it **Derived Field Report**.
2. Click the **Report Definition** tab.
3. In the **Field Name** column, click the popup button to display a list of the fields in the Personnel Management module. The fields are arranged by category. Select the **User** category, which is where we placed our derived field. (This derived field is also listed in the All Fields category.)
4. Select **Long Name** and then click **OK**.
5. Finish the **Report Definition** by selecting the individual fields for **First Name**, **Middle Name**, and **Last Name**. Run the report.

You can clearly see that the long name you have created is an accurate concatenation of the three names for each employee selected. You can also see that the caption entered for the field becomes the column heading for the report.

6. Close the **Document Viewer** and then the **Adhoc Reports** screen, saving your changes. Close the **Explorer**.

Now let's use our new field in a template.

7. Create a new **Letter Template** and name the template **Derived Field Test**. Place it in the **Personal** category. Leave the **Paper Size** as **Letter**.
8. Click the **Edit Template** button to start the template editor. Type in a sentence or two. Now use the **Insert Field** feature to insert the **Long Name** field. Then add three more fields to the template, the **First Name**, the **Middle Name**, and the **Last Name**. You can put them anywhere on the letter template you would like.
9. Close the **Letter Template Editor**, saving your changes.
10. Close the **Letter Templates** screen, saving your changes.
11. Create a new **Letter** document and then at the **Letter Name** field, select the **Derived Field Test** template. Run the document.

A letter will display for the first employee. The letter will show the derived field and each of the fields that make up the derived field. You can move through the records to see various employees.

12. Close the **Document Viewer**, and close the **Letters** screen, without saving. Close the **Explorer**.

Derived fields can also be used in custom screens.

13. On the **Tools** menu, select **Custom Screens** and create a new custom screen. Name your screen **Test Derived Field**. Leave the **Screen Type** field set to **Standard**. Add a **Frame** and then add four **Fields**: **Long Name**, **First Name**, **Middle Name**, and **Last Name**.

Preview your custom screen to make sure everything looks right. Increase the length of the Long Name field so there is enough room for a long name. Note that the Caption you entered for the derived field is the label that is adjacent to the field. Close preview mode, saving your change. Close the Custom Screens screen, saving your changes.

14. On the **Tools** menu, select **Workflows**. Select the **Standard - Employee** workflow. On the **Screens** tab, insert a new row at the top of the grid and add your **Test Derived Field** screen. Leave the rest of the **Standard-Employee** workflow as it was. Close the **Workflows** screen, saving your changes.

15. Open **Donald Stern's** record in the **Standard-Employee** workflow.

The new first screen should be displayed and it should show your derived field and the fields that comprise it. Note that the derived field is grayed out, which means that it is for display only. All derived fields are for display only.

16. Close **Donald Stern's** record.

There is one more way in which a derived field can be used. A derived field can be used in another derived field.

17. On the **Tools** menu, select **Derived Fields** and create a new derived field. Name the field **Really Long Name**. You can set the **Caption** to the same value.

18. In the **Formula** field, click the **Field** button to display a list of the available fields. Select the **Prefix** field. Use the **Field** button to select the **Long Name** derived field we created earlier, and then add the **Suffix** field.

The formula should look like this, where “xxx” is the number of the derived field ID that corresponds to the Long Name field:

~**PREFIX**~ ~**DERIVED\_**xxx~ ~**SUFFIX**~

Let's test it.

19. Click the **Test Field** button and select **Donald Stern**.

In the Sample field, you will see a nice long name displayed that happens to be a derived field that contains another derived field.

Before concluding this lesson, let's confirm that this field works in a report.

20. Close the **Derived Fields** screen, saving your changes.

21. Open the **Derived Field Report** created earlier. On the **Report Definition** tab, add **Really Long Name** to the bottom of the list of fields. Run the report.

You should see a whole lot of names including your latest derived field.

22. Close the **Document Viewer** and then close the **Adhoc Reports** screen, saving your changes. Close the **Explorer**.

23. Continue with the next tutorial.

---

# Lesson 2

## Derived Fields 2

---

### Contents

Tutorial 2.1 – Math Operators	18
Tutorial 2.2 – Rounding Methods	21
Tutorial 2.3 – String Operators	23

---

This lesson covers the math operators, rounding methods and string operators available in derived fields.

### Tutorials

**Math Operators:** shows how to use some of the math operators available in derived fields.

**Rounding Methods:** demonstrates how to use rounding in derived fields.

**String Operators:** shows how to use string operators in derived fields to manipulate strings of characters.

---

## Tutorial 2.1 – Math Operators

In this lesson, we will learn about the math operators available in derived fields.

1. On the **Tools** menu, select **Derived Fields**.
2. Create a new derived field. Ignore the **Field Name** through **Caption** fields for now, but change the **Field Type** to **Currency**.

This means that the result of the field we are creating will be treated as a currency field. If you use a currency field in a report and you have chosen to display a currency sign, a currency sign will be added to this field when the report is run.

3. Leave the **Decimals** field set to **2**.
4. In the **Formula** field, we need to select a currency field. Click the **Field** button to display the list of fields. Select the **Annual Pay** field.

We are going to use the Annual Pay field to create a “what if” report. The “what if” report will list every employee’s annual pay “if” they are given a four percent increase.

5. Click the **Operator** button to display a list of operators. Select the **Multiplication** operator.
6. Now, enter the value **1.04**.

The formula should look like this:

**~SAL\_ANNUAL~ \* 1.04**

This formula will multiply the Annual Pay by 1.04. The result will be the equivalent of a 4% increase. We are going to create our “what if” report in just a moment. First, however, we need to make a few other changes.

7. Change the **Field Name** and the **Caption** to **Four Percent Increase**. Close the **Derived Fields** screen, saving your changes.
8. Create a new **Adhoc Report**.
9. In the **Document Name** field, enter **What if Salary Change Report**. In the **Comment** field, enter **For 3 and 4 Percent Changes**.
10. On the **Report Definition** tab, add the following **Field Names**.

**Full Name**

**Annual Pay**

**Three Percent Increase** (a derived field that was already made)

**Four Percent Increase** (the derived field we just created)

11. Now run the report.

The report will show the name of each employee, their current annual pay, their annual pay if a three percent raise were given, and their annual pay if a four percent raise were given. When you see a report like this, you can understand why it is so nice to be able to

export to Excel. You could export this to Excel and forward the spreadsheet to the CFO. That would be impressive and you can let the CFO work out the details!

12. Close the **Adhoc Reports** screen, saving your changes. Close the **Explorer**.

We are going to make it better while demonstrating some other features.

13. Return to the **Derived Fields** screen and select your **Four Percent Increase** field. On the **File** menu, select **Copy Derived Field**.

Note that the new field was named after the first field, but that a “-1” was appended.

14. The “-1” field should already be highlighted and ready for editing. Change the **Field Name** and the **Caption** of the field to **PY Four Percent Increase** (the PY stands for “prior year”).

Now, here comes something new. We need to put the four percent increase on annual pay from the prior year. There is no modifier column as there is in adhoc reports. So how do we indicate that we want a row from a grid (such as the annual pay from the compensation grid) other than the first row? It’s easy.

15. Click just before the second tilde at the end of the **SAL\_ANNUAL** field ID. Insert a colon character “:” and a **2**.

The colon is used to indicate that the next value is a row modifier. The 2 is the row modifier. The formula should look like the following.

**~SAL\_ANNUAL:2~ \* 1.04**

This will derive a field that is 1.04 times the annual pay in the second row of the compensation grid. (We are assuming that the second row represents the annual pay in the prior year.) Of course, you can use any other number if you want to use the annual pay from any other row. Use a ‘+’ in place of the number if you want to use the last row in a grid.

16. Use the **Test Field** button to test this field for **Donald Stern**.

In the Sample field, you will see Donald’s salary from the prior year with a four percent increase. Now, on to the “what if” report.

17. Close the **Derived Field** screen, saving your changes.

18. Open the **What if Salary Change Report** we created earlier. On the **Report Definition** tab, after the **Full Name** field, insert two rows. In the first blank row, add the **Annual Pay** field with a **Row Modifier** of **2**. This will print the actual salary from the prior year. On the second blank row, add the **PY Four Percent Increase** field that we just created. This will print the annual salary from the prior year, if a four percent increase had been granted. Delete the **Three Percent Increase** row. Run the report and be amazed at the things you can do with derived fields.

Ah, but we’re not finished. We have more to learn.

19. Close the **Document Viewer**. Close the **Adhoc Report** screen, saving your changes. Close the **Explorer**.

20. Create another derived field. Set the **Field Name** and **Caption** to **Salary Differential**. Now use what you have learned to create a formula that shows the difference between a four percent raise planned for the prior year and the actual salary change given.

Here is some help... The formula should subtract the derived field named PY Four Percent Increase from the current Annual Pay.

The formula should look like this, where “xxx” is the number of the derived field ID that corresponds to the PY Four Percent Increase field:

`~SAL_ANNUAL:1~ - ~DERIVED_xxx~`

You could remove “:1” and it would have the same result.

21. Remember to set the **Field Type** to **Currency**. Leave the **Edit Mask/Decimals** field set to **2**. Close the **Derived Fields** screen, saving your changes.
22. Open the **What if Salary Change Report** and add this new field to the bottom of the field list. Delete the **Four Percent Increase** row. Run the report.

For each employee, the report shows the employee’s name, their prior salary, their prior salary if they had received a four percent raise, their actual next salary, and then the difference between the salary received and the four percent raise planned.

Along the way you have learned a lot about math operators and the use of derived fields within other derived fields. There are other math operators such as addition and division, but they work as expected and with what you now know, you can use them effectively.

23. Close the **Document Viewer**, close the **Adhoc Report** screen, saving your changes. Close the **Explorer**.
24. Continue with the next tutorial.

## Tutorial 2.2 – Rounding Methods

After that last tutorial, it's time for a short discussion of rounding methods. Derived fields can utilize a bunch of rounding methods. They all work the same to round a numeric value up or down to the nearest value. We'll try a few of these.

1. Create a new derived field. Ignore the **Field Name** through **Caption** fields for now, but change the **Field Type** to **Currency**.
2. In the **Formula** field, add the **Annual Pay** field. Before proceeding, test the field for **Donald Stern** and jot down Donald's annual pay for future comparison.
3. Click just prior to the first tilde and then click the **Rounding** button to display a list of the rounding methods. Select the method **RoundUp1000**.

The method will round Donald's annual pay up to the nearest 1000. This is a forced round up, and thus an annual pay such as 72,000 will be rounded up to 73,000.

When you selected the rounding method, the correct syntax with placeholders was inserted into the formula. We already had our field chosen, so we just need to remove the placeholder.

4. Delete the **~expression~** part of the formula. Also delete the closing parenthesis because we need that at the very end. Now, add a close parenthesis after the field.

This tells the rounding feature to round the contents of what is in the parenthesis. In our little example, what is between the parentheses is the annual pay, but it could be more than that. The formula should look like the following:

**RoundUp1000 (~SAL\_ANNUAL~)**

5. Test the field for **Donald Stern**.

Note that the value you wrote down has been rounded up to the nearest 1000.

6. Now change the rounding method to **Round100** by just removing the **Up** and a **0**.

This is not a forced rounding. Round 100 (as well as Round and Round1000) follows the rules we were taught in grammar school. The number will be rounded up if the next digit (to the right of the digit being rounded) is five or greater. The number will be rounded down if the next digit is less than five.

7. Test the field for **Donald Stern**.
8. Switch the rounding method to **RoundDown100**.

This will round the annual pay down to the nearest 100. Like RoundUp, RoundDown is a forced rounding.

9. Test the field for **Donald Stern**.

Rounding is rounding is rounding. If you understand the rules, you can apply rounding as you see fit. Remember that computations within parenthesis are performed prior to the rounding and that you do need parenthesis when you use any type of rounding.

10. Delete the current derived field.
11. Continue with the next tutorial.

## Tutorial 2.3 – String Operators

There are several string operators available that allow you to manipulate strings of characters.

1. Create a new derived field and just leave the fields defaulted as is.

There are many times when it is best to just show the last four digits of a social security number. Let's see how to grab just those four digits.

2. In the **Formula** field, click the **Operator** button.
3. Select **Right**.

The Right formula is returned with placeholders.

4. Delete the **~field~** placeholder and select the **Social Security** field.
5. Delete the **#** placeholder and replace it with **4**.
6. Click the **Test Field** button and select any employee.

The last four digits of that employee's social security number are displayed in the Sample field.

Let's assume you also want to have placeholders for the other digits. We are going to use the '\*' and '-' literals for the placeholders. Those symbols are also used as mathematical operators. In order to tell the system that we want to use these as literals and not operators, we will need to enclose them in single quotes. Literals that are not also operators are not required to be in quotes, but can be if you choose.

7. Click at the beginning of the formula and add the following:

```
'***_**_'
```

Your formula should now look like this, with no space before the word Right because we do not want a space in the result:

```
'***_**_'Right(~SS_NO~,4)
```

8. Test the field again.

Easy as that!

As we all know, it is best not to broadcast the year of a birth date. Here is how you can strip that information.

9. Clear the current formula.
10. Click the **Operator** button.
11. Select **Left**.

The Left formula is returned with placeholders.

12. Delete the **~field~** placeholder and select the **Birth Date** field.

13. Delete the # placeholder and replace it with **5**.
14. Click the **Test Field** button and select any employee.

The employee's birth date is displayed in the Sample field, without that nasty year.

There are several other string operators that we won't take the time to demonstrate. There is a Mid operator that returns a specified number of characters starting at a specified location. The LCase and UCase operators change the string to all lower case or all uppercase accordingly.

As you have seen, when you select an operator that requires a certain formula, that formula is defaulted with placeholders. This makes using any operator a breeze.

15. Close the **Derived Fields** screen, without saving your changes.
16. Continue with the next tutorial.

---

# Lesson 3

## Derived Fields 3

---

### Contents

Tutorial 3.1 – Date Operators	26
Tutorial 3.2 – Complex Fields	29
Tutorial 3.3 – SQL Statement Derived Fields	31

---

This lesson covers how to use date operators in derived fields as well as how to build complex derived fields. It concludes with showing how to create a derived field using an SQL statement.

### Tutorials

**Date Operators:** shows how to use the two date operators in derived fields.

**Complex Fields:** demonstrates how to build complex derived fields.

**SQL Statement Derived Fields:** shows how to create a derived field using an SQL statement.

---

## Tutorial 3.1 – Date Operators

There are two types of date operators. One type determines the number of days, months, or years between two dates, and the other is used to add a specified number of days, months, or years to another date. We'll learn about both here.

A few of the dates in People-Trak have built in calculations to determine the number of days, months, or years between that date and the current date. It is simply not feasible to create these calculations for every date and thus we have given you the ability to perform such calculations on those dates for which you need this information.

Calculations to determine the difference between two dates are made using a series of operations known as date-difference operators. There are three, one for days between two dates, one for months between two dates, and one for years between two dates.

1. On the **Tools** menu, select **Derived Fields** and create a new derived field. Set the **Field Name** and **Caption** to **Length of Employment**. Set the **Field Type** to **Numeric** since the result of the calculation will be a number, not a date or a currency value. Leave the **Edit Mask/Decimals** field set to **0**.
2. In the **Formula** field, click the **Operator** button to display a list of operators. Select the operator named **DateDiff-D**.

This operator is used to determine the number of days between two dates. The valid formula for this operator is added with placeholders.

3. Delete the **~date1~** placeholder and click the **Field** button to display a list of fields. Select the **Hire Date** field.

This is the first date in our comparison. We must now select a second date. We'll try something simple first.

4. Delete the **~date2~** placeholder and select the **Termination Date** field.

The formula should look something like this:

**DateDiff-D(~HIRE\_DATE~ , ~TERM\_DATE~)**

5. Test this derived field for **Donald Stern**.

What you find is that Donald Stern does not have a termination date, so, there are no days to calculate. We can fix this.

6. Close the **Derived Fields** screen, saving your changes.
7. Open **Donald Stern's** record and without a moment of remorse, proceed to the Status screen and enter today's date in the **Termination Date** field. Close **Donald's** record, saving your change. Go back to your derived field. Now try the test again.

You should see the number of days between Donald's hire date and today. This is a very typical formula for determining the number of days between two named dates.

Ah, but what if you want to determine the number of days between Donald's hire date and today? There is no named field for today. And today keeps changing. We thought of

that and you will recall using date substitution variables in the report writer. These are also available in derived fields.

8. In the **Formula** field, delete the phrase **~TERM\_DATE~**.
9. Click the **Date Variable** button.

The Date Variable screen is displayed and gives you access to all of the date variables available within People-Trak.

10. Select **TODAY**.

As you may recall, TODAY is the date variable that substitutes the current date. The formula should now look like this:

**DateDiff-D(~HIRE\_DATE~ , ~TODAY~)**

11. Go ahead and test this for **Donald Stern**.

The result will be the number of days between Donald's hire date and today. If you run the report tomorrow, the number will be one greater. If you ran it yesterday, it would be one day less. This is a very, very common formula to determine the number of days since any prior date in People-Trak. Consider such calculations as the number of days an employee has been in a current job, the number of days since a physical exam, or the number of days since an applicant submitted an application. Very powerful stuff.

There are many date variables and you can use them as either the "to" or "from" date. For example, if you use the formula shown below, you can compute the number of days until an employee's next appraisal.

DateDiff-D(~TODAY~ , ~NEXT\_REVIEW~)

And finally, yes, you actually can use date variables for both dates. And, if you use TODAY and YESTERDAY, you will get 1 day. If you use YESTERDAY and TOMORROW, you will get 2 days. Now that this question is resolved, let's get back to work.

Remember that we can determine the number of months or the number of years as well.

12. Change the **Operator** in your current formula to use the **DateDiff-M** operator. Now test the formula for **Donald Stern**.

Rather than the number of days, you should see the number of months (which should be somewhat smaller).

13. Finally, change the current formula to use the **DateDiff-Y** operator. Test that.

This number is even smaller.

Now there is some fine print. The number of days between two days is exact. The number of months is the number of whole months. You cannot assume that a value of 12 months is 365 days. It could be as many as 395 days or so because the last month is not quite complete. The same issue exists for years. If the calculation returns a value of 5 years, the number of days could be nearly 365 days off.

As indicated earlier, derived fields also have the capability of adding days, months, or years to another date. We'll show a quick example.

14. Change the formula in the current derived field to the following. You can type it or use your buttons.

**DateAdd-D(90 , ~HIRE\_DATE~)**

This formula indicates that we wish to know the date that is exactly 90 days from the employee's Hire Date.

15. Now, before testing, change the **Field Type** from **Numeric** to **DateTime** because the result of a date-add operator is a date. Leave the **Edit Mask/Decimals** field set to **Date**. Test the field for **Donald Stern**.

The date shown will be the Hire Date plus 90 days. As this date is in the past, it is not very meaningful. However, for new hires, this field can indicate when benefits are effective or when a probation period will end.

There may be a time when you wish to subtract days from another date. In this case, simply put a negative sign (a dash) in front of the number. Try it now.

16. Make the formula compute the date **90** days prior to the **Hire Date**. Test it for **Donald Stern**.

There are DateAdd operators for days, months, and years. The results can be less predictable for months and years. If you add 3 months to November 30, the new date will be the last day in February and not February 30, which does not exist. Likewise, if you add 2 years to February 29 (in a leap year, of course), the new day will be February 28, because there is no February 29 that year.

We will need this field in the next tutorial, so let's give it a new name that reflects the changed formula.

17. Set the **Field Name** and **Caption** to **90 Days Before Hire**.
18. Stay in the **Derived Fields** screen for the next tutorial.
19. Continue with the next tutorial.

## Tutorial 3.2 – Complex Fields

Now, let's create even more complex derived fields.

1. Create a new derived field. Leave the defaulted fields as is for now. In the **Formula** field, select each of the following **Fields**.

**Full Name**  
**Address 1**  
**Address 2**  
**City, State Zip**

We'll now create a single field known as an address block. This one field can be used in forms, letters, and labels so that you do not need to keep creating this address block manually from a collection of fields.

2. Use the list of operators to insert a **crLf** operator between **Full Name** and **Address 1**.

The **crLf** operator is used to create a multi-line field. The field after the **crLf** will be moved down one line and to the left.

We are about to add the **Address 2** field. If we used the **crLf** operator and the **Address 2** field was blank, we would get a blank line in our address block. In order to avoid that, we will use a ',' to separate those addresses.

3. Add a comma after **Address 1**. Finally, add another **crLf** operator after **Address 2**.

The formula should look like this:

```
~NAME_FULL~ crLf ~ADDRESS_1~, ~ADDRESS_2~ crLf ~CityStateZip~
```

4. Go ahead and test this field for **Donald Stern**.

The result will be a multi-line address block. This may or may not be the precise formula your company will use, but it is very convenient to have a single derived field to ensure consistency among your documents that use an address block.

Now, it just so happens that we have supplied an **Address Block** derived field in applicable modules in **People-Trak**. Let's take a look at that.

5. Select **Address Block** in the list on the left.

As you can see, it uses the same formula you just built.

You may be wondering how adhoc reports can digest a multi-line field like this. Let's find out.

6. Close the **Derived Fields** screen, saving your changes.
7. Create a new **Adhoc Report**. On the **Report Definition** tab, add the **Address Block** field. Then add another field such as the **Hire Date**. Run the report.

Note that the adhoc report creates vertical space to accommodate multi-line fields. Of course, labels and letters and forms work just the same.

8. Close the **Document Viewer**. Close the **Adhoc Reports** screen. Close the **Explorer**.

Our final example combines what we have learned in previous tutorials.

9. Return to the **Derived Fields** process and select the derived field named **90 Days Before Hire**.

Let's see how complex we can make this thing.

10. In the **Formula** field, change the formula to look like this:

**DateDiff-D( DateAdd-D(-90, ~HIRE\_DATE~), ~TODAY~)**

11. Because the left-most calculation is a number, change the **Field Type** to **Numeric**.

If you have not figured this formula out yet, here is what it is doing. This formula computes the number of days between the Hire Date minus 90 days and Today. It could be worse, but you get the point. This is high-school algebra all over again and you can obtain some very interesting fields.

12. Test this field for **Donald Stern** and see the results.
13. Stay in the **Derived Fields** screen.
14. Continue with the next tutorial.

---

## Tutorial 3.3 – SQL Statement Derived Fields

So far, we have been working with derived fields that were created using a formula. Now, let's create a derived field using an SQL statement.

1. Create a new derived field.
2. In the **Field Name** field, enter **Average Salary Percent Increase**.
3. Leave the **Category** field as **User**.
4. In the **Field Style** field, select **SQL Statement**.
5. In the **Caption** field, enter **Average Salary Percent Increase**.
6. In the **Field Type** field, enter **Numeric**.
7. In the **Edit Mask/Decimals** field, enter **2**.

Your cursor should now be in the Formula field. Of the five buttons below the Formula field, only the Test Field button is highlighted. This is because you must type your SQL statement without any help from those buttons. This requires that you are very familiar with SQL. You can use the multi-line popup button if you need to enter a long SQL statement. For our demonstration, we will use just a simple Select statement.

8. In the **Formula** field, enter the following SQL statement:

**Select Avg(Sal\_Percent) from PMSalaryDetail**

This will give you the average salary percent increase for a given employee.

9. Test this field for **Donald Stern**.

You are given Donald's average salary percent increase.

The SQL Statement is a powerful feature in creating derived fields, but you must know SQL in order to take advantage of it.

**Note:** In SQL Statements, 'XXX' can be used to specify the location of the 'Where ID = ##' logic. For example, 'Select Sal\_Rate From PMSalaryDetail XXX And Seq = 3' becomes 'Select Sal\_Rate From PMSalaryDetail Where ID = ## And Seq = 3'.

10. Close any open screens, without saving your changes.
11. Exit **People-Trak**.
12. This concludes this tutorial and lesson.