

Unit Event Defaults
Pushing out changes

Peter Keeler
April 18, 2017

XLERANT

The image features a network diagram background with orange and blue sections. The orange section on the left contains a complex network of nodes and connections, with several nodes highlighted by circles. The blue section on the right contains the main title and author information. The XLERANT logo is located in the bottom left corner of the orange section.

Goals

- Know what Unit Event Defaults are
- Understand what changing them does to your budgets
- Learn the financial impact of each kind of change



Topics



- What are Unit Event Defaults?
- Setting baseline values
- Options when making changes
- Financial impact
- User impacts

The Basics

What are Unit Event Defaults?

- Define *how* for raises and bonuses: \$ or %
- Defines *when* for raises and bonuses
- Separate for each *Employee Set*
- 'Locked in' when a budget starts

Today's focus



Section	Event type (BudgetPak internal description)	Employee class	Currency	Event method	Percent	Amount	Pool	Default event scope	Date usage	Default date MM-DD
Salary Increases	Raise		USD	percent	0.00	0	0	same for all	Default date	2011-06
Bonus	Bonus	A	USD	percent	50.00	0	0	same for all	Default date	2011-09
Bonus	Bonus	B	USD	percent	0.00	0	0	same for all	Default date	2011-06
Bonus	Bonus	C	USD	percent	0.00	0	0	same for all	Default date	2011-06
Bonus	Bonus	Overtime	USD	percent	0.00	0	0	same for all	Default date	2011-06
Other Compensation	Other Compensation		USD	percent	0.00	0	0	same for all	Default date	2011-04

4

Unit event defaults are the definitions for the moment-in-time events that can take place in your budget. Currently, these are defined as raises and bonuses for your employees. These are baseline values only. In keeping with the BudgetPak philosophy, entering more precise details like 'Bob is retiring in May' or 'Jane is getting a larger raise because she was a star performer' are left to the budgetholders to enter, since they're closer to the ground and will be more aware of upcoming events like those.

In addition to defining the amount of each raise or bonus, your defaults also set the initial way those events will be distributed – by percent, by amount, or from a fixed pool. They also defined when each event will take place by default – either a fixed date, or on the anniversary of a given employee's hire date.

You can create a different set of unit event defaults for each employee set you create – in fact, you can create multiple sets if you want, though they won't have any effect unless there's a set of employees in place to receive those defaults.

The most important detail to understand about Unit Event Defaults is that those values are 'locked in' when a budgetholder actually starts their budget. The only way to update them for your in-progress budgets at that point is to use the 'push out unit event defaults' function.

Example

Consider this scenario...

- You've set up your system assuming a 5% raise for all employees in the coming year
- The budget process has already started
- A budgetholder has already made some decisions
 - Most employees have the default raise
 - One employee has a 15% raise
 - One employee has a 10% raise
- Right now, the salary total is \$857,477 for this unit
- You realized that the default should be a 7% increase and have changed configuration.
- What happens?
 - Nothing by default; the total is not changed
 - The defaults were 'locked in' when the users started their budgets
- You want your updated configuration to take effect.
- How does this work?

5

Here you can see the Salary Increases screen with some budgetholder decisions made. As you can see, they've chosen to accept the default method, which is 'by percent of salary,' and to use 'give 'everyone except' the same raises,' and to similarly 'Use HR-defined date for 'everyone except.''" As a result, most of the employees have the default of a 5% raise, taking place on April 5th. However, one employee as a 15% raise, one employee has a raise on a different date, and one employee has a different raise on a different date. You can see the defined 'Standard raise' in the highlighted box on the screen.

At the start, the total salary cost in this example is \$857,477, which we can see in the Salaries account. Your users have already started their budgeting; some of them have already finished and signed off.

However, you as the administrator have now realized that the default should really be a 7% increase, not 5%. You went into configuration and changed the default to 7%. But what happens to the completed and in-progress budgets?

The answer is, nothing. By default, your users' decisions are retained. They said to use a certain raise, and BudgetPak is saving that amount. Those 5% raises are locked in.

Is this what you want to happen? That's up to you. You can use the 'push out defaults'

option on the Configure Unit Event Defaults page to make BudgetPak propagate your changes out to active budgets.

Example

Before making changes:

Salary Increases [Back] [Discard] [Close] [Save] [Next]

Please review and answer the questions step by step. You can always come back later and change your answers.

Step 1 How would you like to budget raises?
 By percent of salary
 Allocate from a pool
 Fixed amount per employee

Step 2 How would you like to handle percentage raises?
 Give everyone the same raises
 Give 'everyone except' the same raises
 Give raises by individual
Standard raises: 5.00 %

Step 3 How would you like to set the raises date?
Note: HR-defined date is Sunday, April 05, 2009
 Use HR-defined date for everyone
 Use HR-defined date for 'everyone except'
 Provide dates individually

Step 4 Designate raises for each current employee (snapshot as of 12/31/2008) [Recalc] [Show/hide columns...] [Actions...]

Employee ID	Name	Net headcount	Hire date	Current salary	Std raises	Salary Increase %	Salary Increases \$	New salary	Std date	Salary Increases date	Est. impact on budget
313887	Gurk, Steven	1.000	5/7/1991	\$179,538	<input type="checkbox"/>	%	\$17,954	\$197,491	<input type="checkbox"/>	9/5/2009	\$6,965
110015	London, William	1.000	4/6/1990	\$177,260	<input type="checkbox"/>	%	\$26,664	\$204,424	<input checked="" type="checkbox"/>	4/5/2009	\$23,757
235826	Seaburn, William	1.000	4/7/1989	\$160,000	<input checked="" type="checkbox"/>	%	\$8,000	\$168,000	<input checked="" type="checkbox"/>	4/5/2009	\$7,128
161058	Tangers, Steven	1.000	5/6/2003	\$181,333	<input checked="" type="checkbox"/>	%	\$9,067	\$190,400	<input checked="" type="checkbox"/>	4/5/2009	\$8,078
262016	Blake, William	1.000	4/6/1987	\$19,695	<input checked="" type="checkbox"/>	%	\$985	\$20,680	<input checked="" type="checkbox"/>	4/5/2009	\$877
179001	Laurel, William	1.000	6/15/2001	\$19,500	<input checked="" type="checkbox"/>	%	\$975	\$20,475	<input checked="" type="checkbox"/>	4/5/2009	\$869
111127	Whitestag, Stanley	1.000	3/6/1991	\$19,500	<input type="checkbox"/>	%	\$985	\$20,680	<input type="checkbox"/>	1/1/2009	\$1,182

Salaries 1000
The annual budget for this line item has been automatically calculated based on headcount or assets.
You may review it here, but not change it.
\$857,477

Here's what our example looks like, when a budgetholder goes into the Salary Increases screen and looks at their decisions thus far, before we've pushed out any changes at all.

Push Options

What happens when you make configuration changes and want to push them out?

1: Update default only

No change to started budgets. Employees keep the raises they were assigned.

2: Update employees with defaults

Apply the new default to all budgets, but only for employees marked as getting the default

3: Update all

Apply the new default to all budgets for all employees, replacing any budgetholder decisions

Unit event defaults

For each unit, designate the defaults for each employee-based event (raise, bonus, etc).

Unit event defaults set: Select "All event types" to view or edit the defaults for the entire company (which will be propagated downward throughout the unit hierarchy).
Select company:
Event type: Select a single event to view or edit the defaults for individual units within the hierarchy. Any changes you make will override the company-level defaults.

7

When you push out defaults, these go out to an entire version at a time – even if you only changed the default for a single unit, the entire version will get the defaults updated. If you only want to change one unit at a time, you can use the regular budgetholder interface instead of the configuration menu.

The first option is to just update the defaults. If you choose this, then in the budget you select, the next time a budgetholder goes in, they'll see the new defaults in place, but everyone will still have the actual dollar amounts they assigned previously. In this case, you're saying that the budgetholders made their decisions about the actual dollars to commit, and you're giving them new information about how much they should aim for, but not changing the dollar amount decisions.

The second option is to update the defaults and also update the dollar values for any employees who were marked as receiving the default amount. This does change the dollar amounts in affected budgets immediately. If a user chose to use the default budget method, such as 'by amount' or 'by percent,' and you change the default amount or the default percent, then any employees who had been getting the default will now automatically be updated to still be getting the default – but the default is now the new value. In this case, you're saying that the budgetholders made the decision to follow the defaults, regardless of the dollar values involved.

The third option is to override all user decisions and update all employees to use the defaults no matter what. Doing this basically says that the changes in configuration have made any existing budgetholder decisions invalid, and they should be completely replaced with your central decisions. This can be very useful in situations where budgetholders don't have authority over their own headcount, for example.

1: Defaults only

Option 1: Update the stored defaults

The screenshot shows the 'Salary Increases' application interface. It consists of four steps for configuring raises:

- Step 1:** How would you like to budget raises? Options: By percent of salary (selected), Allocate from a pool, Fixed amount per employee.
- Step 2:** How would you like to handle percentage raises? Options: Give everyone the same raises (selected), Give 'everyone except' the same raises, Give raises by individual. A 'Standard raises' dropdown is set to 7.00%.
- Step 3:** How would you like to set the raises date? Note: HR-defined date is Sunday, April 05, 2009. Options: Use HR-defined date for everyone (selected), Use HR-defined date for 'everyone except', Provide dates individually.
- Step 4:** Designate raises for each current employee (snapshot as of 12/31/2008) and new hires: A table with columns: Employee ID, Name, Hrs headcount, Hire date, Current salary, STD raises, Annual % increases, Salary crosses \$, New salary, STD date, Salary increases date, Est. impact on budget.

Employee ID	Name	Hrs headcount	Hire date	Current salary	STD raises	Annual % increases	Salary crosses \$	New salary	STD date	Salary increases date	Est. impact on budget
313887	Gulf, Steven	1.000	5/7/1991	\$179,55	<input type="checkbox"/>	10 %	\$17,954	\$197,491	<input type="checkbox"/>	9/5/2009	\$6,965
130015	London, William	1.000	4/6/1990	\$177,74	<input type="checkbox"/>	15 %	\$26,664	\$204,424	<input checked="" type="checkbox"/>	4/5/2009	\$23,757
233836	Seaburn, William	1.000	4/7/1989	\$160,00	<input checked="" type="checkbox"/>	7 %	\$11,200	\$171,200	<input checked="" type="checkbox"/>	4/5/2009	\$9,979
161068	Tangers, Steven	1.000	5/6/2003	\$181,35	<input checked="" type="checkbox"/>	7 %	\$12,693	\$194,026	<input checked="" type="checkbox"/>	4/5/2009	\$11,309
262016	Blake, William	1.000	4/6/1987	\$19,64	<input checked="" type="checkbox"/>	7 %	\$1,379	\$21,074	<input checked="" type="checkbox"/>	4/5/2009	\$1,228
179001	Laurel, William	1.000	6/15/2001	\$19,58	<input checked="" type="checkbox"/>	7 %	\$1,365	\$20,865	<input checked="" type="checkbox"/>	4/5/2009	\$1,216
111127	Whitestag, Stanley	1.000	3/6/1991	\$19,69	<input checked="" type="checkbox"/>	7 %	\$1,379	\$21,074	<input type="checkbox"/>	1/1/2009	\$1,654
				7.000							

Below the table, a 'Salaries' summary box shows: Salaries 1000. The annual budget for this line item has been automatically calculated based on headcount or assets. You may review it here, but not change it. \$857,477

- The computed salary does *not* change
- Reports show *no change*
- Budgetholder will see the new default the *next* time they enter the Salary Increases page
- Budgetholder can click 'Save' to take the new defaults, and then salary will recompute

This is what users will see if you choose the 'update defaults' option. No budget numbers change until a user takes action. The Salaries account still shows the original amount, \$857,477, and the Budget Decisions report will still show the same amount. However, the next time the user goes into the Raise or Bonus screen, they will see the new default values – and the Raise or Bonus screen will show the user what the impact of the new defaults is. In this case, we've changed the original 5% raise to a 7% raise. The listed 'standard raise' is updated on the screen, and everyone getting the standard raise shows the user what their raise will be if they NOW go in and accept the new default.

Until a user enters this screen and hits save, their dollars stay as they were before. If you run a Budget Decisions report or a Headcount Expenses report, you'll be able to see the actual current decisions for each employee. This can be very useful when you want to update defaults but budgets have already been signed off; you can then use reporting to see what actual outputs exist, while the system screen indicates what the users are being guided to use.

2: Update employees with defaults

Option 2: Update the employees who had the defaults

Salary Increases

Please review and answer the questions step by step. You can always come back later and change your answers.

Step 1
How would you like to budget raises?
 By percent of salary
 Allocate from a pool
 Fixed amount per employee

Step 2
How would you like to handle percentage raises?
 Give everyone the same raises
 Give 'everyone except' the same raises
 Give raises by individual
Standard raises: 7.00 %

Step 3
How would you like to set the raises date?
 Note: HR-defined date is Sunday, April 05, 2009
 Use HR-defined date for everyone
 Use HR-defined date for 'everyone except'
 Provide dates individually

Step 4
Designate raises for each current employee (snapshot as of 12/31/2008) and new hires:

Employee ID	Name	Net headcount	Hire date	Current salary	Std raises	Salary increases %	Salary increases \$	New salary	Std date	Salary increases date	Est. impact on budget
313887	Gurt, Steven	1.000	5/7/1991	\$179.5	<input type="checkbox"/>	10 %	\$17,954	\$197,491	<input type="checkbox"/>	9/5/2009	\$6,965
110015	London, William	1.000	4/6/1990	\$177.7	<input type="checkbox"/>	15 %	\$26,664	\$204,424	<input type="checkbox"/>	4/5/2009	\$23,757
23836	Seaburn, William	1.000	4/7/1989	\$160.0	<input checked="" type="checkbox"/>	7 %	\$11,200	\$171,200	<input checked="" type="checkbox"/>	4/5/2009	\$9,979
161068	Tangers, Steven	1.000	5/6/2003	\$181.3	<input checked="" type="checkbox"/>	7 %	\$12,693	\$194,026	<input checked="" type="checkbox"/>	4/5/2009	\$11,309
202016	Blake, William	1.000	4/9/1987	\$159.6	<input checked="" type="checkbox"/>	7 %	\$11,379	\$21,074	<input checked="" type="checkbox"/>	4/9/2009	\$1,220
179001	Laurel, William	1.000	6/15/2001	\$19.5	<input checked="" type="checkbox"/>	7 %	\$1,365	\$20,865	<input checked="" type="checkbox"/>	4/5/2009	\$1,216
111127	Whitsteg, Stanley	1.000	3/6/1993	\$18.6	<input checked="" type="checkbox"/>	7 %	\$1,379	\$21,074	<input type="checkbox"/>	11/2009	\$1,654
		7.000									

Salaries 1000
 The annual budget for this line item has been automatically calculated based on headcount or assets.
 You may review it here, but not change it.
 \$864,170

- Any employee who *did* have the standard increase is updated
- The computed salary *updates*
- Reports show the *new* raise
- Budgetholder will see the new default if they enter the Salary Increases page, but do *not* need to save to apply the new default
- If the budgetholder has not made raise decisions yet, *no raises exist* and will be created normally when the budgetholder gets to it

If you choose the second option, to update all employees with defaults, your salaries account will be immediately updated, and if you look at the Budget Decisions report or the Headcount Expenses report, you *will* see the new default applied to all employees. The employees who were not marked as getting the default before are still getting their custom, individual values. As you can see, after choosing 'Update employees with defaults,' the Salaries account immediately updated to show the new total - \$864,170, almost \$7000 higher than the original value, without the user doing anything.

There's one important nuance to be aware of: this *only* applies to employees who have already been marked as getting the default. If the budgetholder has not visited the raise section, then they have not made any decision – so no one has any raises at all, so no one has the default, either. However, when they do enter the raise screen, at that point they will see the default raise, and create raises normally then. This is mostly important if you chose to start your budgets in bulk, using the 'start from scratch' option, then wanted to modify raises.

3: Update all

Option 3: Update everything

Salary Increases

Please review and answer the questions step by step. You can always come back later and change your answers.

Step 1 How would you like to budget raises?
 By percent of salary
 Allocate from a pool
 Fixed amount per employee

Step 2 How would you like to handle percentage raises?
 Give everyone the same raises
 Give 'everyone except' the same raises
 Give raises by individual
Standard raises: 7.00 %

Step 3 How would you like to set the raises date?
Note: HR-defined date is Sunday, April 05, 2009
 Use HR-defined date for everyone
 Use HR-defined date for 'everyone except'
 Provide dates individually

Step 4 Designate raises for each current employee (snapshot as of 12/31/2008) and new hires:

Employee ID	Name	Net headcount	Hire date	Current salary	Salary raises %	New salary	Salary increases date	Est. impact on budget	
313887	Gurk, Steven	1.000	5/7/1991	\$179,538	7 %	\$12,568	\$192,105	4/5/2009	\$11,197
110015	London, William	1.000	4/6/1990	\$177,760	7 %	\$12,443	\$190,203	4/5/2009	\$11,086
235836	Seaburn, William	1.000	4/7/1989	\$160,000	7 %	\$11,200	\$171,200	4/5/2009	\$9,979
161668	Tangers, Steven	1.000	5/6/2003	\$181,333	7 %	\$12,693	\$194,026	4/5/2009	\$11,309
262016	Blake, William	1.000	4/6/1987	\$19,695	7 %	\$1,379	\$21,074	4/5/2009	\$1,228
179001	Laurel, William	1.000	6/15/2001	\$19,500	7 %	\$1,365	\$20,865	4/5/2009	\$1,216
111127	Whitestag, Stanley	1.000	3/8/1991						
				7.000					

Salaries 1000
The annual budget for this line item has been automatically calculated based on headcount or assets. You may review it here, but not change it.
\$849,680

- Every employee is updated
- The computed salary *updates*
- Reports show the *new* raise
- Budgetholder exceptions are *replaced* by the new defaults
- *Method* and *date* decisions are replaced by the new defaults
- If the budgetholder has not made raise decisions yet, raises are *created with the defaults*

We can see a much more dramatic change if you select the 'Update all' option. First, we can see that the budgetholder decision to use the "Give 'everyone except' the same raises" has been overridden with the default, as has their decision about dates. Now all of the employees have the exact same raise date and the exact same raise percent, even the ones who were marked as getting exceptions earlier. Taking this option means that the guidance that budgetholders were using before should be completely negated, or that budgetholders are not expected to be making changes to headcount. The total immediately changed to eliminate those two exceptions from before, both of which were giving extra-large raises. With those extra-large raises replaced by the default, the total has gone down by nearly \$8000 relative to the original total, without the user doing anything.

Note that this really is 'update all' – even if a budgetholder hasn't yet entered raises and bonuses, this option will actually create raises and bonuses for all employees in every budget that's been started.

Setting a new baseline

How changing the unit event defaults affects the baseline

- **The baseline version is a read-only version automatically computed by BudgetPak during initialization**

- Shows what you get if you apply all of the configured defaults
 - Unit event defaults, starting employees, unit event defaults, and so on
- Demonstrates the impact that budgetholder decisions have made relative to the defaults
- Read-only; always shows exactly what the administrator has configured



11

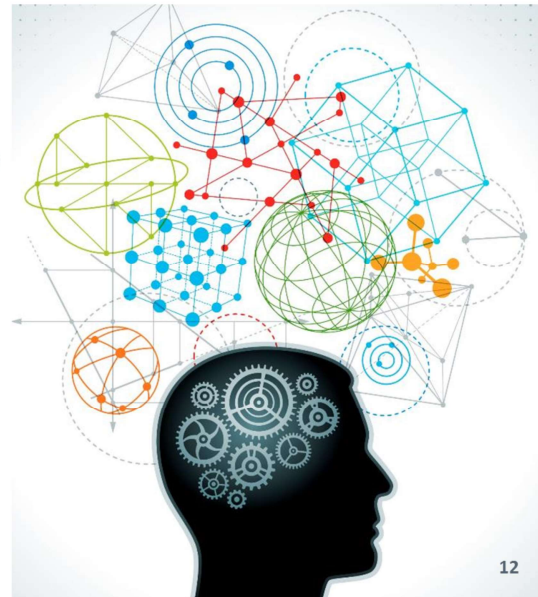
It's important that a budgetholder knows what the state of their budget is while they're in the process of making it. If they made the decision that everyone should get, say, a three percent raise, then they should be able to come back later and see their numbers just as they were before. If they don't have a fixed value to guide them, it makes it hard for them to give you the best input they can.

Your baseline budget doesn't take user input, though, so the baseline will automatically update when you change your unit event defaults. That way, you and your users can see what the high-level estimate is for their budget. If you're using zero-based budgeting, that still is valuable, because your headcount decisions are still going to mean every unit starts with some dollars in place. The baseline computes its values as though every budgetholder had gone in and chosen to use the default budget method, the default amount, and the default date for every employee, with no new hires, terminations, or leaves.

Your actual budgetholders have to make those decisions for their own units when they're budgeting. That's when you can see the effects of a change. If Jane decided that everyone should get the default raise of five percent on the default date of June first, then Jane expects to see those values when she comes back in to look at the budget later. And, in fact, Jane will, even if the administrator has changed the values in configuration, because those numbers were saved when the budget started.

User Impact

1. **Plan carefully**
 - **No undo option on pushing out**
2. **Pushing out with option 3 will erase budgetholder decisions**
3. **Communicate with your users**
 - ... **Especially after signoffs**



A key part of BudgetPak's philosophy is that the users should trust their numbers and feel accountable for what they produced. That's a big part of why you want to avoid making configuration changes after your users have starting the budget process – if they don't have confidence that the decisions they made and the numbers they saw were correct, they won't be happy with their budgeting process. Sometimes making configuration changes after the process starts is inevitable – that's why this feature exists. But it's much better to not rely on this function, and to have everything set up before your users get started. Remember, too, that once you've pushed out new defaults, there is no 'undo' function available other than rolling back your entire system.

Always take a step back and consider before pushing out new defaults, particularly using the 'update all' option. Sometimes it really does reflect a change in expectations that needs to be included in the budget, but make sure to consider whether it's a global change, and whether there could be any important user input that could be lost.

The other important thing to keep in mind, in this as in so many other parts of your budgeting process, is to think about what impact this will have on your users. Particularly before using the 'Update all' option, make sure that you really don't need any input that was already in place about one-offs and custom changes. If you have high-level information, consider whether there might be any exceptions that would be important to

know about. Keeping your users aware when something is going to affect their budgets will keep them engaged with the process and satisfied that they know and understand what's going on. If some of your budgetholders have already signed off on their budget, you may want to explicitly revoke that signoff so that your budgetholders will specifically see their updated budgets before taking responsibility for them.



Q&A

Thank you!

