

# Solution 7 Training Guide

## Advanced Aggregation

## Solution 7 Software Documentation Disclaimer and Copyright

**Solution 7**, Magdalen Centre, 1 Robert Robinson Ave, Oxford, OX4 4GA. [www.solution7.co.uk](http://www.solution7.co.uk)

Copyright © 2020 by Solution 7, Oxford, England.

*Solution 7 Limited provides this material 'as-is' and make no representations or warranties with respect to the accuracy or completeness of the contents and specifically disclaim all warranties, including without limitation warranties of merchantability and fitness for a particular purpose. Solution 7 Limited shall not be liable for errors contained herein or for incidental or consequential damages, including lost profits, in connection with the furnishing, performance or use of this material whether based on warranty, contract or other legal theory. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation.*

*This work is provided with the understanding that Solution 7 Limited is not engaged in rendering professional legal or accounting services. If professional assistance is required, the services of a competent professional person should be sought. Solution 7 Limited shall not be liable for damages arising therefrom.*

*The fact that an organization or website is referred to in this work is a citation and/or a potential source of further information and does not mean that the publisher or the author endorses the information the organization or website may provide or recommendations it may take. Further, readers should be aware that internet websites listed in this work may have changed or disappeared between when this work was written and when it is read.*

*This document contains proprietary information that is protected by copyright (see notice below). Permission is hereby given for licensees, of the Solution 7 product to which this document relates to print additional copies of this document in association with use of the product. Otherwise no part of this document may be printed, photocopied, reproduced, incorporated in other materials or translated to another language without the prior written consent of Solution 7 Limited.*

*The information in this document is subject to change without notice.*

### **COPYRIGHT**

*Warning: copyright of this documentation and the software, to which it relates, is the property of Solution 7 Limited or its licensors. Any unauthorized use, copying or sale of the above may constitute an infringement of copyright and may result in criminal or other legal proceedings.*

*For general information on our other products and services, please contact [support@solution7.co.uk](mailto:support@solution7.co.uk).*

**Copyright © 2020 by Solution 7 Limited, All rights reserved.**

## Introduction

In *Solution 7 Training Guide – Basic Concepts*, we learned how easy it is to build high-level, summarised reports utilising Solution 7's Lists and Functions. We can further extend the capabilities of the functions by employing various techniques to control the summarisation and aggregation of the results.

In this lesson we will explore how, with a combination of Excel's native functionality and Solution 7 Lists and Functions, you can create accurate, highly aggregated reports from your NetSuite data.

## NetSuite

The information in this training guide is positioned from a NetSuite OneWorld user of Solution 7. If you are not running NetSuite OneWorld, you will be able to follow the exercises, but some dialogs may appear slightly differently to the screenshots (i.e. missing Subsidiary field). In that case, where this document refers to Subsidiary, please substitute Class, Department or Location as appropriate to your instance of NetSuite.

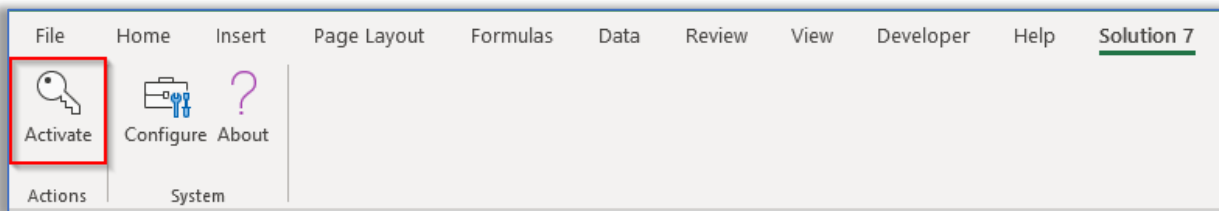
## Before We Begin

Before we begin, there are some items that we will refer to during the lesson.

### Activation

When you first start Excel, Solution 7 is not automatically activated. You may be opening and/or editing Excel workbooks that have no NetSuite content. Solution 7 only needs to be activated if you are building new NetSuite report templates or need to refresh data in existing templates. Once activated, one of your Solution 7 licenses will be considered "in use". To release the license for another user, simply close Excel.

To activate Solution 7 within Excel, click **Solution 7 > Activate**.

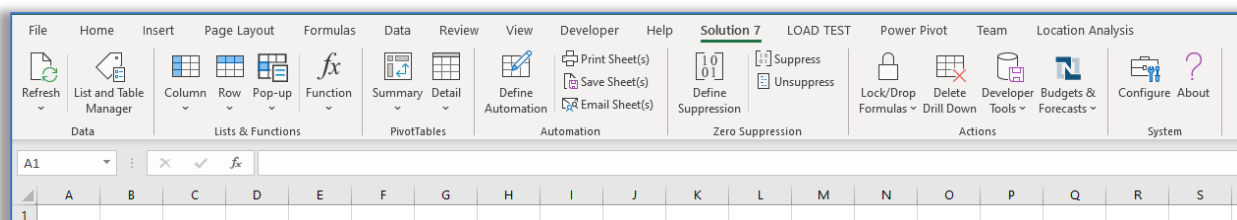


Solution 7 will now attempt to establish a connection to NetSuite and, if successful, will expand the full Solution 7 ribbon.

If you don't see the Solution 7 menu or Solution 7 fails to activate, please refer to the [Troubleshooting Guide](#) or contact [support@solution7.co.uk](mailto:support@solution7.co.uk).

## Solution 7 Ribbon

The Solution 7 Ribbon expands after activation and is the starting point when using any of Solution 7's features.

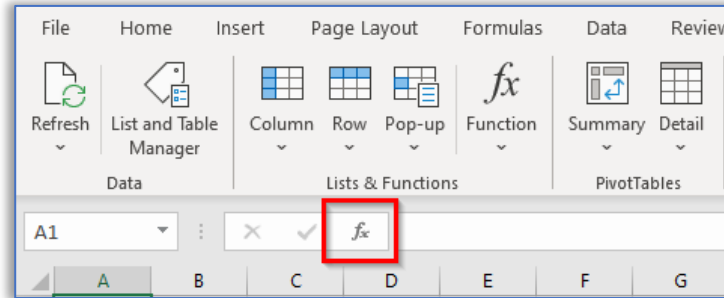


### Small and Large fx Buttons

When inserting or editing a function we will often refer to the 'Small fx' and 'Large fx' buttons.

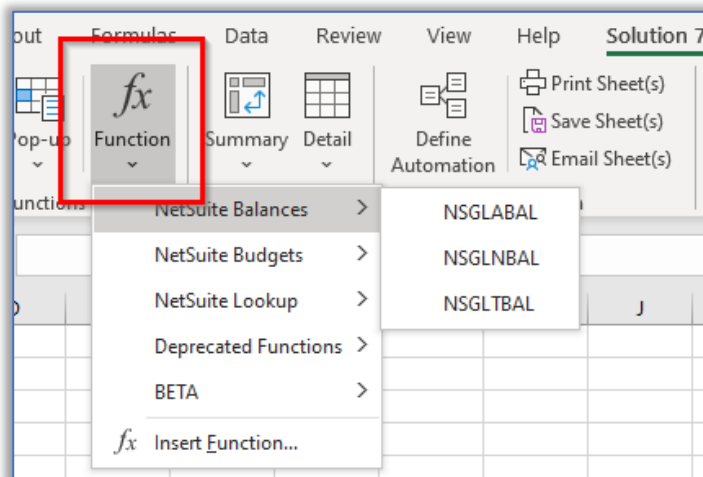
#### Small fx Button

The Small fx button is Excel's "Insert function" button and is to the left of Excel's formula bar.



#### Large fx Button

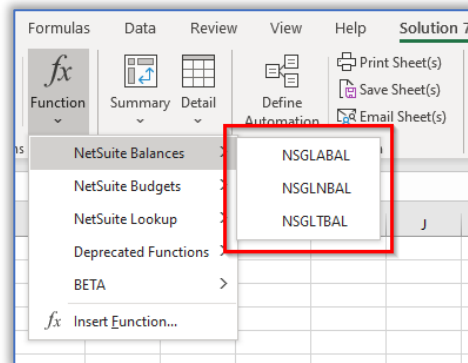
The large fx button is Solution 7's quick-access Function button in the Lists & Functions group of the Solution 7 ribbon.



## Accounting Functions

Solution 7 extends the Excel function set with additional functions that can be used to build your own report templates to return live, real-time data from NetSuite.

In the following exercises, we will learn how to drive high-level reports using Solution 7's Balance functions to provide summarized account balances.



When using the Balance functions, values can be pulled from NetSuite in three different ways:-

- NSGLABAL – returns a general ledger balance by account **number**.
- NSGLNBAL – returns a general ledger balance by account **name**.
- NSGLTBAL – returns a general ledger balance by account **type**.

### Exercise 1 – Returning a Summary Balance

In this exercise we will learn how to use the NSGLABAL function to return a summarized account balance from NetSuite.

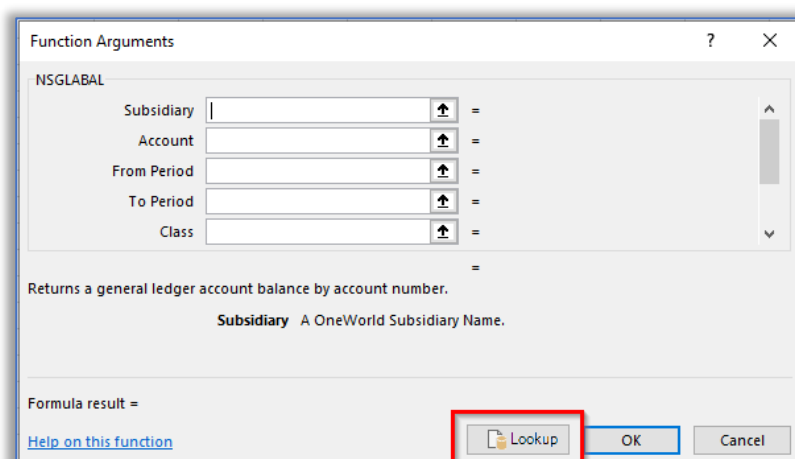
Begin on a blank Excel workbook and with Solution 7 activated, follow these steps:-

#### Step 1:

Select empty cell C3, click the Large *fx* button and select **NetSuite Balances** > **NSGLABAL**.

#### Step 2:

With your cursor in the Subsidiary argument, click **Lookup**.



### Step 3:

Select the "Consolidated" item at the top of the list and click OK.

### Step 4:

With your cursor in the Account argument, enter the account number "4000".

### Step 5:

With your cursor in the From Period argument, enter "Jan 2019"

### Step 6:

Click OK to return a summarized account balance.

Function Arguments

NSGLABAL

Subsidiary: "HH Inc. (Consolidated)" = "HH Inc. (Consolidated)"

Account: 4000 = 4000

From Period: "Jan 2019" = "Jan 2019"

To Period: =

Class: =

Returns a general ledger account balance by account number.

Subsidiary A OneWorld Subsidiary Name.

Formula result = -1804603.42

[Help on this function](#)

	A	B	C	D	E	F	G	H	I
1									
2									
3			-1804603.42						
4									
5									
6									

As you can see, Solution 7 has summarized the January 2019 transactions for account '4000' and presented the result in the cell. Excel will treat this cell in the same way as any other cell value.

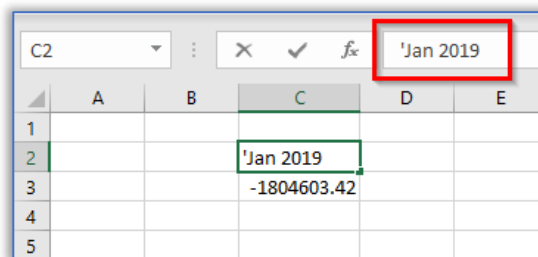
## Exercise 2 – Making Summarized Reports More Dynamic

In this exercise, we are going to drive the function directly from the cell to create some dynamic summarized values.

Continue with the worksheet from the previous exercise and with Solution 7 activated, follow these steps:-

### Step 1:

In empty cell C2, enter "Jan 2019". Precede the period with a single apostrophe to prevent Excel interpreting it as a date. Period is case sensitive and needs to be set as text. (i.e. 'Jan 2019).



### Step 2:

In empty cell D2, enter "Feb 2019". (i.e. 'Feb 2019).

### Step 3:

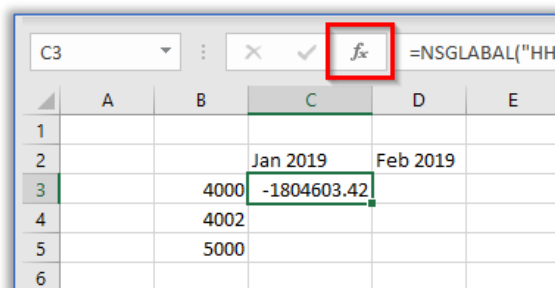
In empty cell B3, type the Account number "4000".

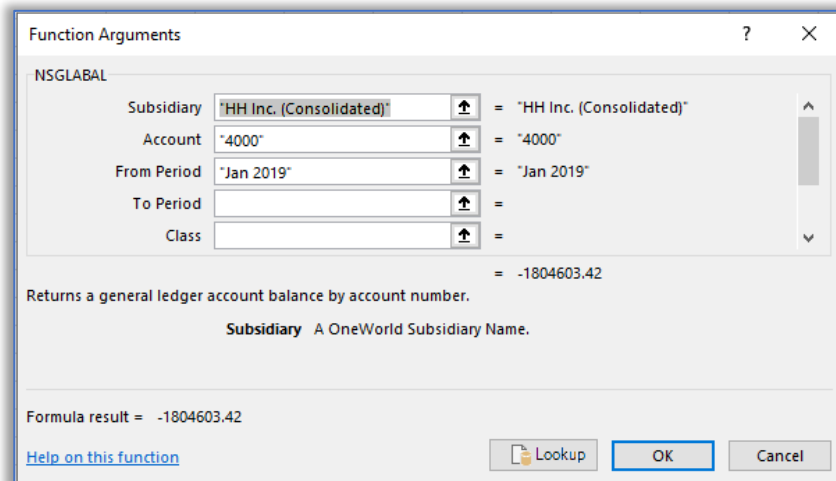
In cell B4 enter "4002".

In cell B5 enter "5000".

### Step 4:

Select cell C3 and click the small fx button.





**Step 5:**

Delete the account number "4000".

**Step 6:**

With your cursor in the Account argument, select cell B3. Press the F4 key 3 times to lock the cell (i.e. \$B3).

**Step 7:**

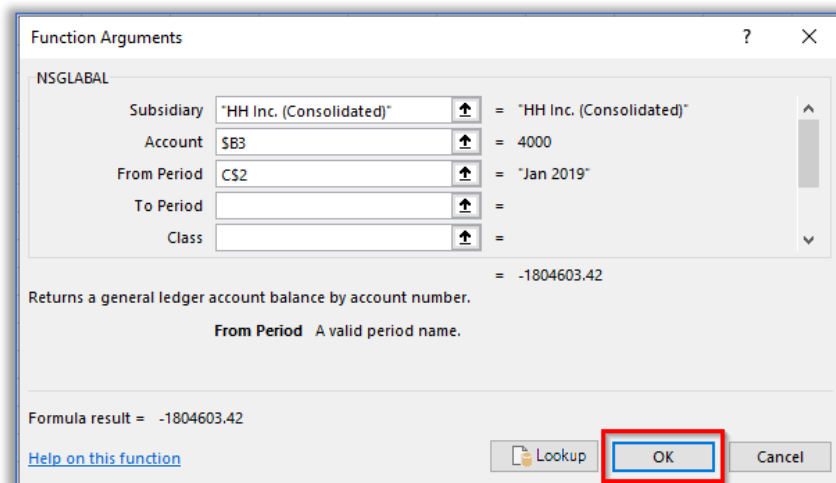
Delete the period "Jan 2019".

**Step 8:**

With your cursor in the From Period argument, select cell C2. Press the F4 key 2 times to lock the cell (i.e. C\$2).

**Step 9:**

Click OK.



**Step 10:**

Copy the formula to cells C5 – D5 to complete the report.

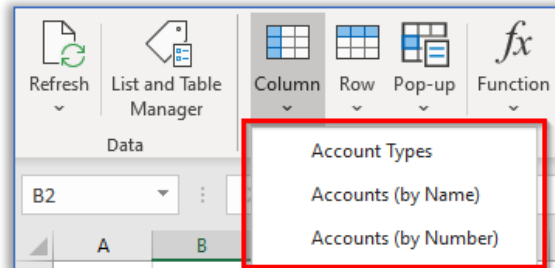
The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H
1								
2			Jan 2019	Feb 2019				
3		4000	-1804603.42	-2556582.86				
4		4002	-1536886.89	-2074102.52				
5		5000	5558100.42	4068067.27				
6								

As you can see, this is an easy way of building an aggregated report using Solution 7's functions.

## Accounting Lists

To simplify building the framework of the report, Solution 7 lists allow NetSuite information to be referenced and inserted into Excel. Lists are dynamically linked to NetSuite and can be refreshed to include new values from NetSuite.



In this lesson, we will learn how to drive our summarized reports using Solution 7's Accounting lists:

- Account Types
- Accounts (By Name)
- Accounts (By Number)

## Exercise 3 – Preparing the Lists

In this exercise, we will insert each Accounting list into a new worksheet within our workbook. These lists will be the starting point for more advanced aggregation later in this lesson.

Continue with the Excel workbook from the previous exercise and with Solution 7 activated, follow these steps:-

### Step 1:

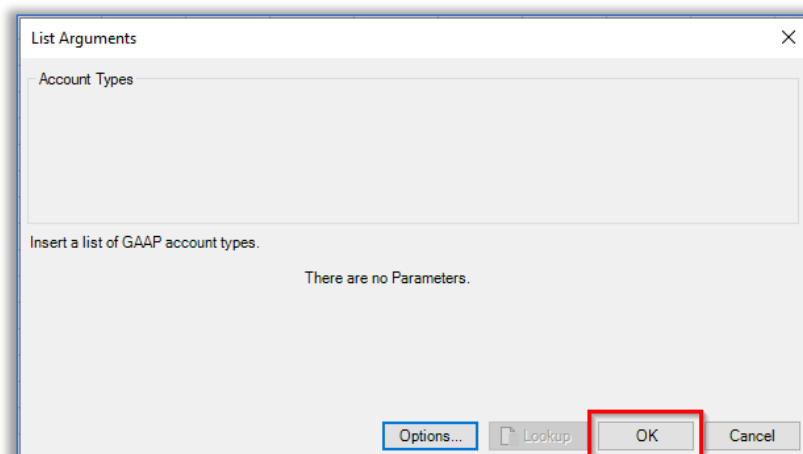
Add a new sheet to the workbook.

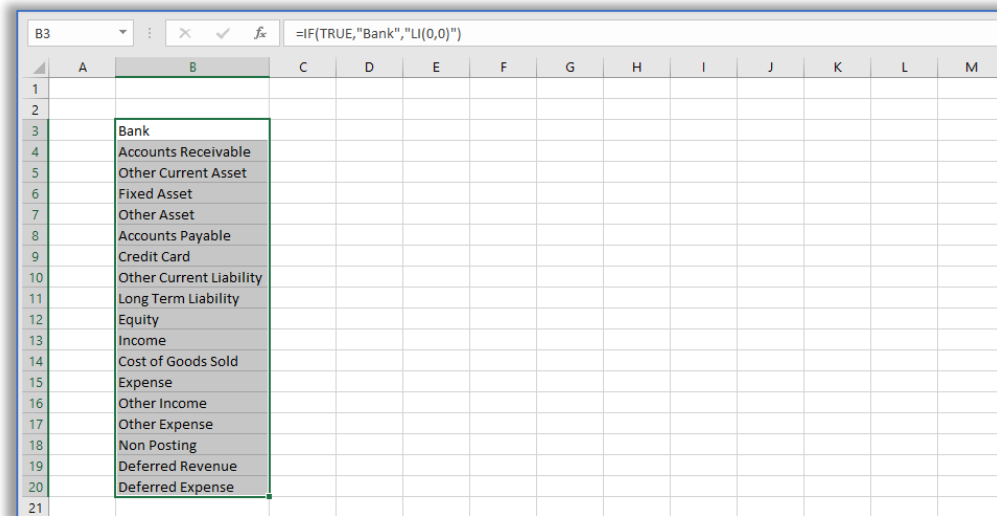
### Step 2:

With the cursor in empty cell B3, select **Column > Account Types**.

### Step 3:

On the List Arguments dialog, there are no available arguments. Simply click OK to insert a list of all Account Types.





#### Step 4:

Add a new sheet to the workbook.

#### Step 5:

With the cursor in empty cell B3, select **Column > Accounts (By Name)**.

#### Step 6:

On the List Arguments dialog, with the cursor in the Subsidiary argument, click **Lookup**.

#### Step 7:

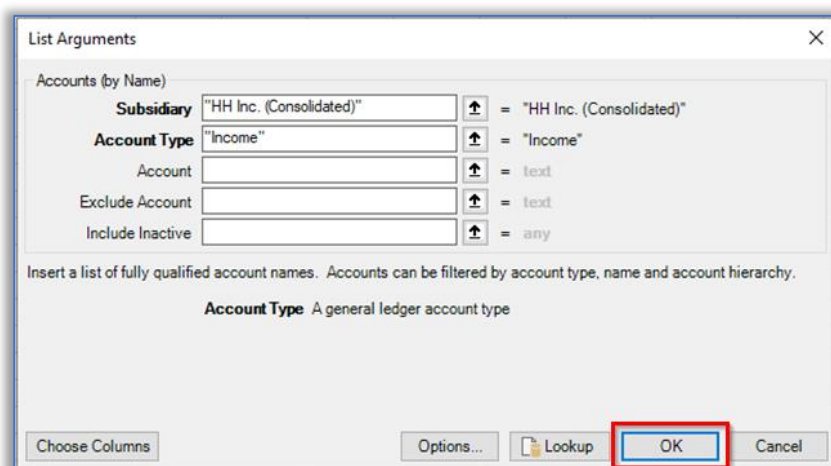
Select the "Consolidated" item at the top of the list and click OK. This will ensure that you get a consistent list of accounts across all subsidiaries.

#### Step 8:

With the cursor in the Account Type argument, enter "Income".

#### Step 9:

Click OK to insert the list.



	A	B	C	D	E
1					
2					
3		Sales			
4		Sales : Sales - Clearance			
5		Sales : Sales - Merchandise			
6		Sales : Sales - Service			
7		Sales : Sales - Warranty			
8		Uncategorized Income			
9		WIP eRev			
10					
11					

### Step 10:

Add a new sheet to the workbook.

### Step 11:

With the cursor in empty cell B3, select **Column > Accounts (By Number)**.

### Step 12:

On the List Arguments dialog, with the cursor in the Subsidiary argument, click **Lookup**.

### Step 13:

Select the "Consolidated" item at the top of the list and click OK.

### Step 14:

With the cursor in the Account argument, enter "4\*".

### Step 15:

Click OK to insert the list.

List Arguments

Accounts (by Number)

**Subsidiary** "HH Inc. (Consolidated)" = "HH Inc. (Consolidated)"

**Account** "4\*" = "4\*"

Exclude Accounts = text

Include Inactive = any

Insert a list of accounts. Accounts can be filtered by account number.

**Subsidiary** A OneWorld Subsidiary Name.

Choose Columns Options... Lookup **OK** Cancel

	A	B	C	D	E
1					
2					
3		4000	Sales		
4		4002	Sales - Merchandise		
5		4004	Sales - Service		
6		4006	Sales - Clearance		
7		4008	Sales - Warranty		
8		4100	WIP eRev		
9					
10					

As you can see, Solution 7's lists allow you to easily build the framework for your reports. In the next section, we will populate each list with some summarized account balances.

## Using NSGLABAL, NSGLNBAL and NSGLTBAL

In this section we will use Solution 7's NSGLABAL, NSGLNBAL and NSGLTBAL functions to return summarized account balance values by account number, account name and account type.

### Exercise 4 – Using the NSGLABAL Function

In this exercise, we will learn to use the NSGLABAL function to return summarized balances by account number using the Accounts (By Number) list.

Begin with 'Sheet4' in the workbook and with Solution 7 activated, follow these steps:-

#### Step 1:

With the cursor in empty cell D3, click the Large fx button and select **NetSuite Balances** > **NSGLABAL** to return GL account balances by the account **number**.

#### Step 2:

Select the Subsidiary.

#### Step 3:

With the cursor in the Account argument, select cell B3 ("4000"). Hit the F4 key 3 times to lock the cell (i.e. \$B3).

#### Step 4:

With the cursor in the From Period argument, enter "Jan 2019".

#### Step 5:

Click OK to return a summarized value.

The screenshot shows the 'Function Arguments' dialog box for the NSGLABAL function. The arguments are as follows:

Argument	Value	Result
Subsidiary	"HH Inc. (Consolidated)"	"HH Inc. (Consolidated)"
Account	\$B3	"4000"
From Period	"Jan 2019"	"Jan 2019"
To Period		
Class		
Returns a general ledger account balance by account number.		-1804603.42

Below the arguments, there is a note: **From Period** A valid period name.

The formula result is: Formula result = -1804603.42

At the bottom, there are three buttons: 'Lookup', 'OK' (highlighted with a red box), and 'Cancel'. There is also a link for 'Help on this function'.

### Step 6:

Copy the formula to complete the report.

	A	B	C	D	E	F	G	H
1								
2								
3		4000	Sales	-1804603.42				
4		4002	Sales - Merchandise	-1536886.89				
5		4004	Sales - Service	-288448.69				
6		4006	Sales - Clearance	-26339.73				
7		4008	Sales - Warranty	-3164827.72				
8		4100	WIP eRev	0				
9								
10								

As you can see, Solution 7 has returned a summarized account balance for each of the “4” accounts during January 2019.

## Exercise 5 – Using the NSGLNBAL Function

In this exercise, we will learn to use the NSGLNBAL function to return summarized balances by account name using the Accounts (By Name) list.

Begin with 'Sheet3' in the workbook and with Solution 7 activated, follow these steps:-

### Step 1:

With the cursor in empty cell C3, click the Large *fx* button and select **NetSuite Balances** > **NSGLNBAL** to return GL account balances by the account **name**.

### Step 2:

Select the Subsidiary.

### Step 3:

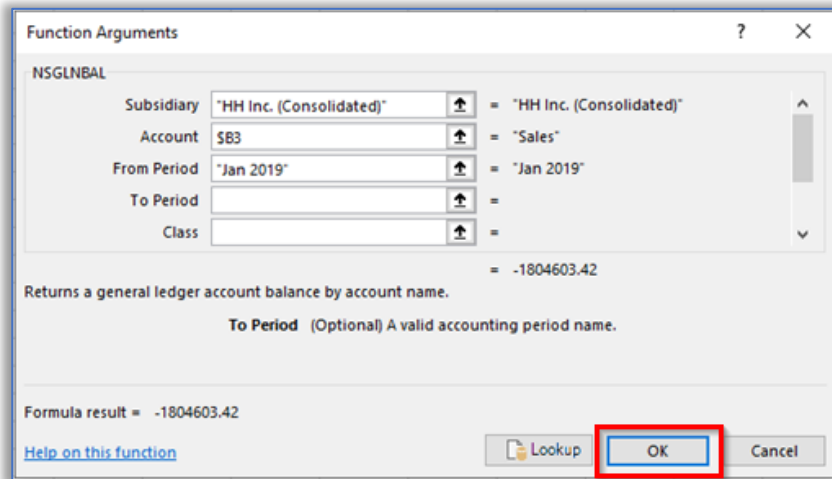
With the cursor in the Account argument, reference cell B3 from your Accounts (By Name) list. Hit the F4 key 3 times to lock the column (\$B3).

### Step 4:

With the cursor in the From Period argument, enter “Jan 2019”.

### Step 5:

Click OK to return a summarized value.



**Step 6:**

Copy the formula to complete the report.

**Step 7:**

Use Excel’s AutoSum feature to provide a total.

	A	B	C	D
1				
2				
3		Sales	-1804603.42	
4		Sales : Sales - Clearance	-26339.73	
5		Sales : Sales - Merchandise	-1536886.89	
6		Sales : Sales - Service	-288448.69	
7		Sales : Sales - Warranty	-3164827.72	
8		Uncategorized Income	0	
9		WIP eRev	0	
10			-6821106.45	
11				

As you can see, Solution 7 has returned a summarized balance for each of our “Sales” accounts during January 2019.

**Exercise 6 – Using the NSGLTBAL Function**

In this exercise, we will learn to use the NSGLTBAL function to return summarized balances by account type using the Account Type list.

Begin with ‘Sheet2’ in the workbook and with Solution 7 activated, follow these steps:-

**Step 1:**

With the cursor in empty cell C13 (or your equivalent), click the Large fx button and select **NetSuite Balances > NSGLTBAL** to return the GL account balance by the account type.

**Step 2:**

Select the Subsidiary.

**Step 3:**

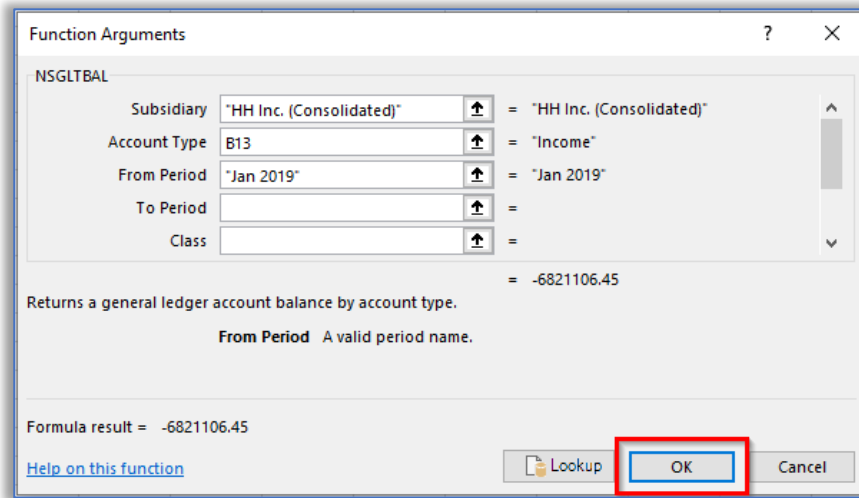
With the cursor in the Account Type argument, select the "Income" cell from the list.

**Step 4:**

With the cursor in the From Period argument, enter "Jan 2019".

**Step 5:**

Click OK to return a summarized balance value.



	A	B	C
1			
2			
3		Bank	
4		Accounts Receivable	
5		Other Current Asset	
6		Fixed Asset	
7		Other Asset	
8		Accounts Payable	
9		Credit Card	
10		Other Current Liability	
11		Long Term Liability	
12		Equity	
13		Income	-6821106.45
14		Cost of Goods Sold	
15		Expense	
16		Other Income	
17		Other Expense	
18		Non Posting	
19		Deferred Revenue	
20		Deferred Expense	
21			

As you can see, this summarized value using account types matches the total value in the previous exercise. Using cell references in Solution 7 can make building a report more efficient and ensures the values can be dynamically updated.

## Multiple Values

When using Solution 7, rather than referencing a single argument value, we can reference multiple values to further aggregate balances and increase flexibility in your reporting. In the following exercises, we will learn to reference multiple argument values by using:

- Cell ranges
- Wildcard characters
- Arrays of values

### Exercise 7 – Returning a Value Using Cell Ranges

In this exercise, we will learn how to return a summarized account balance value using a cell range. Begin with 'Sheet4' in the workbook and with Solution 7 activated, follow these steps:-

#### Step 1:

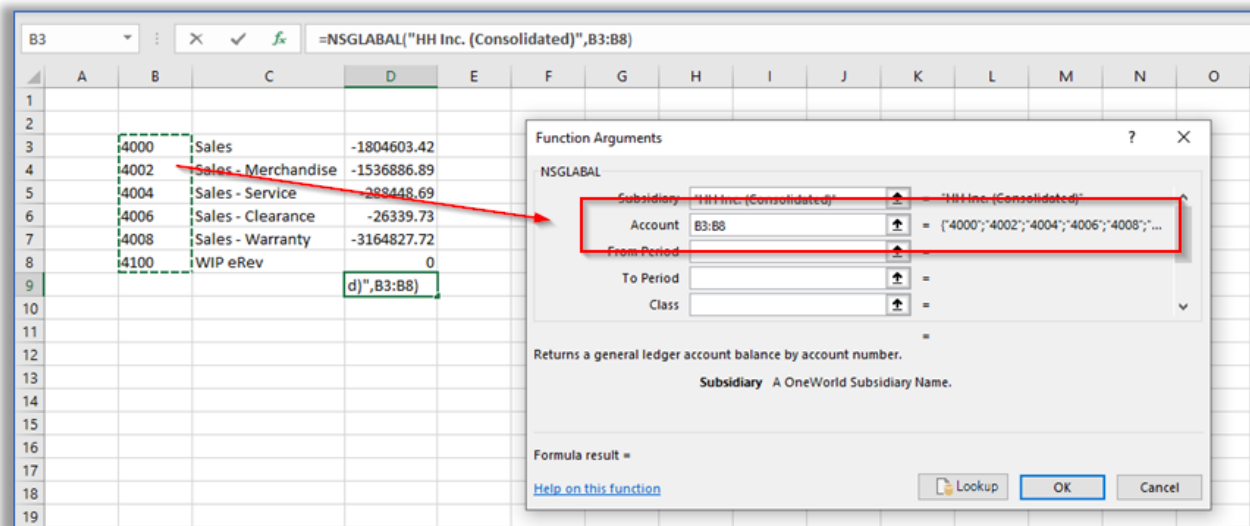
With your cursor in empty cell D9 (or your equivalent), click the Large fx button and select **NetSuite Balances > NSGLABAL**.

#### Step 2:

Select the Subsidiary.

#### Step 3:

With your cursor in the Account argument, reference the account numbers in the list. This presents the accounts as an array which we will learn about later in the lesson.

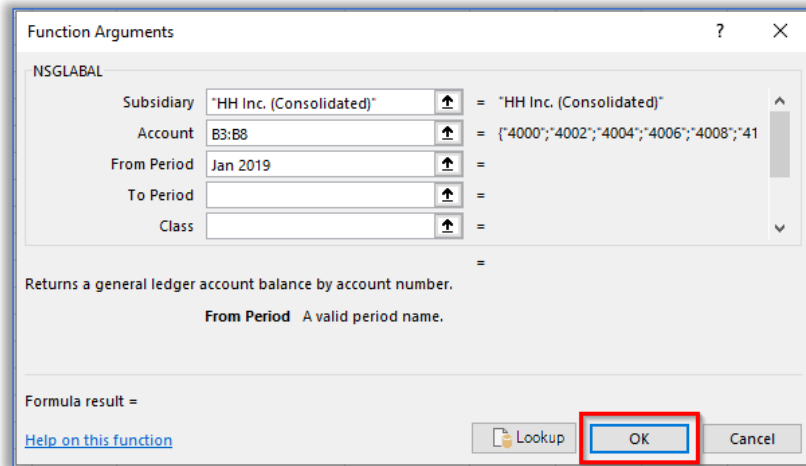


#### Step 4:

With the cursor in the From Period argument, enter "Jan 2019".

#### Step 5:

Click OK to return a summarized account balance value for the Sales accounts.



	A	B	C	D	E	F	G	H
1								
2								
3		4000	Sales	-1804603.42				
4		4002	Sales - Merchandise	-1536886.89				
5		4004	Sales - Service	-288448.69				
6		4006	Sales - Clearance	-26339.73				
7		4008	Sales - Warranty	-3164827.72				
8		4100	WIP eRev	0				
9				-6821106.45				
10								

As you can see, referencing a cell range is an easy way to quickly return a more summarized balance.

## Using Wildcard Characters

A wildcard is a special character which matches one or more characters within a formula, function or list argument. Wildcards allow pattern matching to be performed on the arguments for you to quickly return accurate summarized values. Solution 7 supports two types of wildcard:

- An asterisk (\*) to match multiple characters.
- A question mark (?) to match any single character.

In the following exercises, we will learn how to return a general ledger account balance value using wildcards.

### Exercise 8 – Using a (\*) Wildcard

In this exercise, we will learn how to use the (\*) wildcard to summarize your Accounts (By Name) values directly within the worksheet.

Begin with 'Sheet3' in the workbook and with Solution 7 activated, follow these steps:-

#### Step 1:

In empty cell B12 (or your equivalent), enter "Sales :\*" to match the Sales parent and child account codes.

#### Step 2:

With the cursor in empty cell C12 (or your equivalent), click the Large fx button and select **NetSuite Balances > NSGLNBAL**.

#### Step 3:

Select the Subsidiary.

#### Step 4:

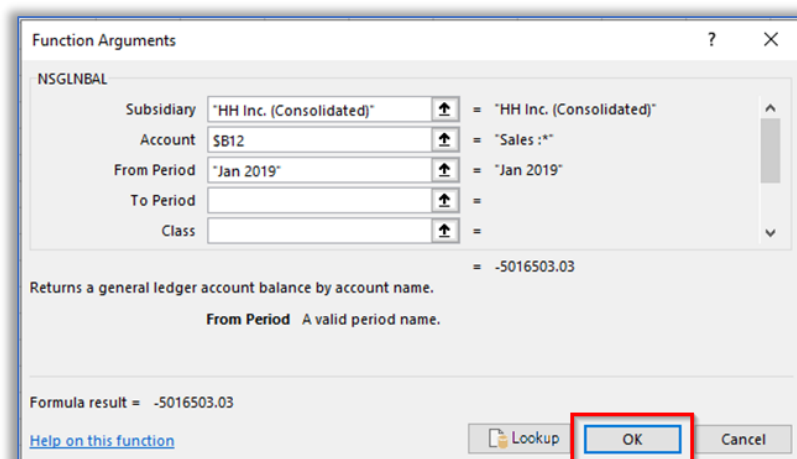
With the cursor in the Account argument, reference the "Sales :\*" cell and use F4 to lock the cell (\$B12).

#### Step 5:

With the cursor in the From Period argument, enter "Jan 2019".

#### Step 6:

Click OK to return a summarized value.



	A	B	C	D
1				
2				
3		Sales	-1804603.42	
4		Sales : Sales - Clearance	-26339.73	
5		Sales : Sales - Merchandise	-1536886.89	
6		Sales : Sales - Service	-288448.69	
7		Sales : Sales - Warranty	-3164827.72	
8		Uncategorized Income	0	
9		WIP eRev	0	
10			-6821106.45	
11				
12		Sales :*	-5016503.03	
13				

**When reporting on account hierarchies, use a (\*) wildcard in an array formula to avoid including accounts in error (see below).**

**Example:-**

**In your worksheet, replace "Sales :\*" with { "Sales" , "Sales :\*" } and press Enter.**

### Exercise 9 – Using a (?) Wildcard

In this exercise, we will learn how to use the (?) wildcard to return a summarized balance for our expense accounts directly within the cell.

Begin with 'Sheet4' in the workbook and with Solution 7 activated, follow these steps:-

**Step 1:**

With the cursor in empty cell F3, select **Column > Accounts (By Number)**.

**Step 2:**

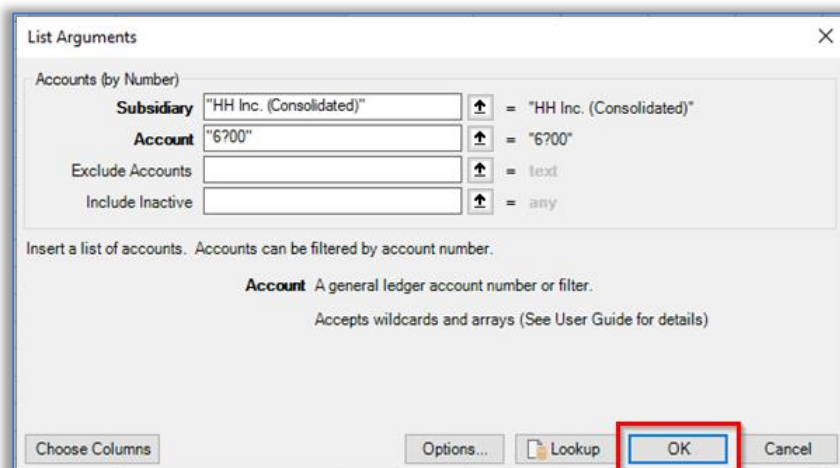
Select the Subsidiary.

**Step 3:**

In the Account argument, enter "6?00".

**Step 4:**

Click OK to insert the list.



	A	B	C	D	E	F	G	H
1								
2								
3		4000	Sales	-1804603.42		6100	Insurance Expense	
4		4002	Sales - Merchandise	-1536886.89		6200	Rent Expense	
5		4004	Sales - Service	-288448.69		6300	Utilities	
6		4006	Sales - Clearance	-26339.73		6400	Salaries & Wages Expense	
7		4008	Sales - Warranty	-3164827.72		6500	Payroll Expenses	
8		4100	WIP eRev	0		6600	Manufacturing Expenses	
9								

As you can see, Solution 7 has pattern matched the wildcard character in the account numbers list to return our expense accounts.

We will now use the NSGLABAL function to return a summarized balance for these accounts.

**Step 5:**

With the cursor in empty cell H3, click the Large fx button and select **NetSuite Balances > NSGLABAL**.

**Step 6:**

Select the Subsidiary.

**Step 7:**

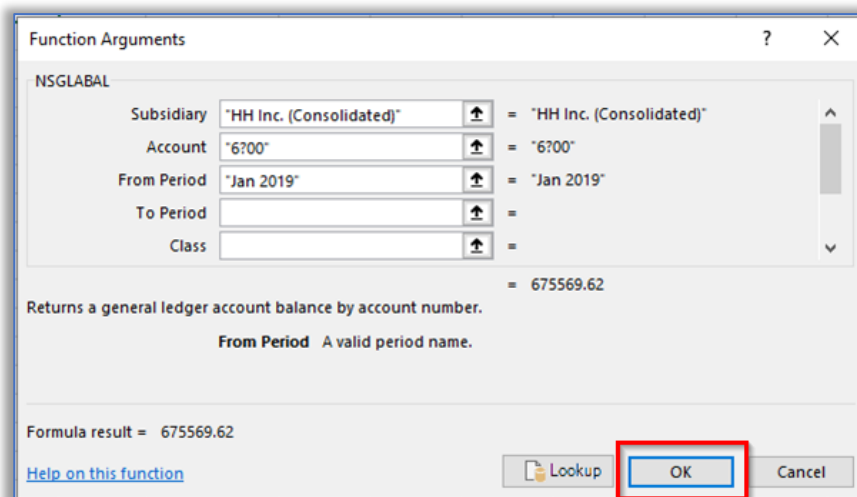
In the Account argument, enter "6?00".

**Step 8:**

With the cursor in the From Period argument, enter "Jan 2019".

**Step 9:**

Click OK to return a summarized value for our expense accounts.



	A	B	C	D	E	F	G	H	I
1									
2									
3		4000	Sales	-1804603.42		6100	Insurance Expense	675569.62	
4		4002	Sales - Merchandise	-1536886.89		6200	Rent Expense		
5		4004	Sales - Service	-288448.69		6300	Utilities		
6		4006	Sales - Clearance	-26339.73		6400	Salaries & Wages Expense		
7		4008	Sales - Warranty	-3164827.72		6500	Payroll Expenses		
8		4100	WIP eRev	0		6600	Manufacturing Expenses		
9									

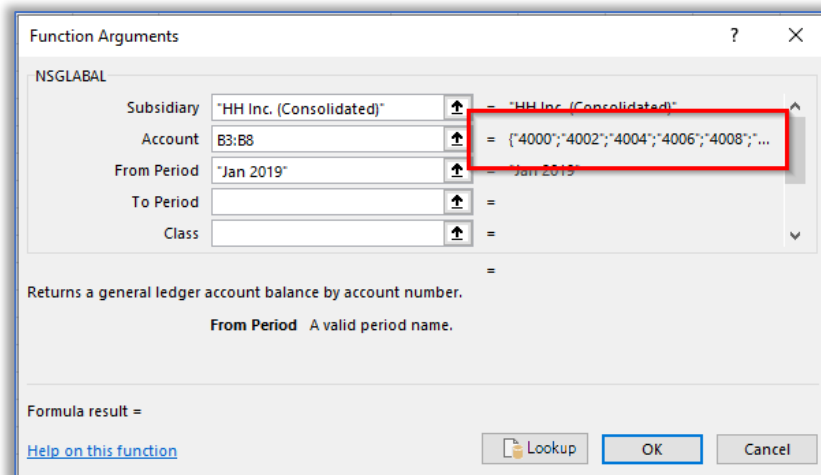
As you can see, Solution 7 has returned a summarized balance value for all the listed accounts during January 2019.

## Arrays

Arrays can be used to return a summarized value for an ad-hoc list of argument values. You can use an array to provide multiple argument values in a function or list.

As you can see in the screenshot below, when using a cell range the account numbers are presented as an array; bound by curly parentheses and separated by commas.

We can use the same syntax to hard code an ad hoc list of values.



## Exercise 10 – Returning Values Using an Array

In this exercise, we will use an array to summarize multiple account balances as a single function.

Begin 'Sheet3' in the workbook and with Solution 7 activated, follow these steps:-

### Step 1:

With your cursor in empty cell C13 (or your equivalent), click the Large *fx* button and select **NetSuite Balances > NSGLNBAL**.

### Step 2:

Select the Subsidiary.

### Step 3:

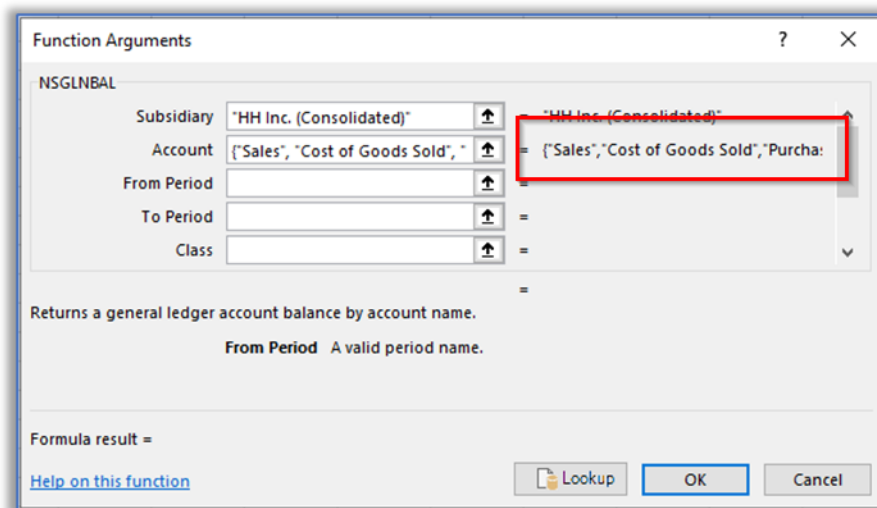
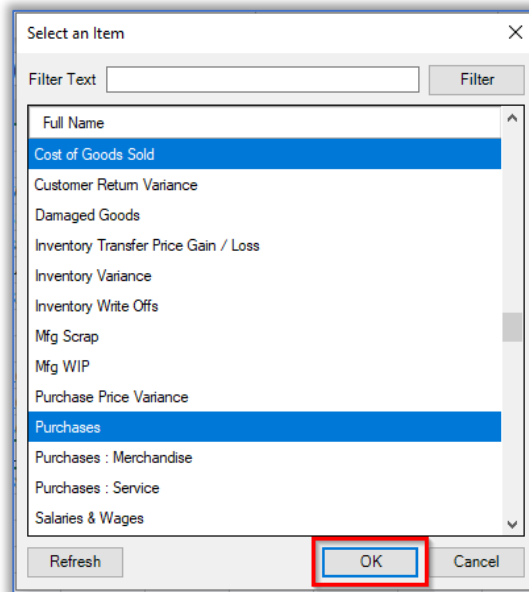
With the cursor in the Account argument, click **Lookup**.

### Step 4:

Hold down Ctrl and select the following Accounts: "Sales", "Cost of Goods Sold" and "Purchases".

### Step 5:

Click OK to insert the accounts in the function argument.



As you can see, the account values are presented as an array. Alternatively, you can simply type the array into the argument. (I.e. {"Sales","Cost of Goods Sold","Purchases"}).

**Step 6:**

With the cursor in the From Period argument, enter "Jan 2019".

**Step 7:**

Click OK to return a summarized value for our chosen accounts.

**Function Arguments** ? X

**NSGLNBAL**

Subsidiary: "HH Inc. (Consolidated)" = "HH Inc. (Consolidated)"

Account: {"Sales","Cost of Goods Sold","P..."} = {"Sales","Cost of Goods Sold","Purc..."}

From Period: "Jan 2019" = "Jan 2019"

To Period: =

Class: =

= 4445892.54

Returns a general ledger account balance by account name.

**From Period** A valid period name.

Formula result = 4445892.54

[Help on this function](#) Lookup **OK** Cancel

	A	B	C	D	E	F	G	H
1								
2								
3		Sales	-1804603.42					
4		Sales : Sales - Clearance	-26339.73					
5		Sales : Sales - Merchandise	-1536886.89					
6		Sales : Sales - Service	-288448.69					
7		Sales : Sales - Warranty	-3164827.72					
8		Uncategorized Income	0					
9		WIP eRev	0					
10			-6821106.45					
11								
12		Sales :*	-5016503.03					
13			4445892.54					
14								
15								

## Exercise 11 – Using an Array in a Cell

Continue with 'Sheet3' from the previous exercise and with Solution 7 activated, follow these steps:-

### Step 1:

With your cursor in empty cell B14 (or your equivalent), enter the array as it appears below. (I.e. {"Sales", "Cost of Goods Sold", "Purchases"})

13			4445892.54
14		{"Sales", "Cost of Goods Sold", "Purchases"}	
15			

### Step 2:

Select cell C14 (or your equivalent) and click the Large fx button.

Select **NetSuite Balances > NSGLNBAL**.

### Step 3:

Select the Subsidiary.

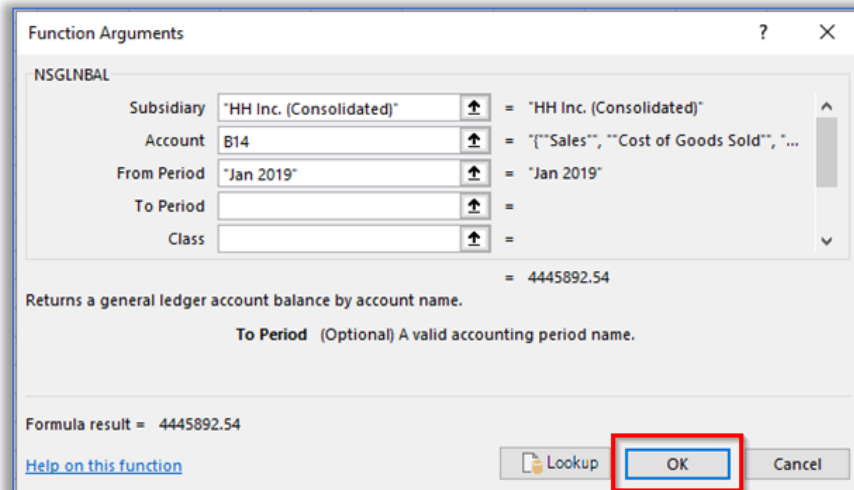
### Step 4:

With the cursor in the Account argument, reference cell B14.

With the cursor in the From Period argument, enter "Jan 2019".

### Step 5:

Click OK to return a summarized value.



	A	B	C	D	E
1					
2					
3		Sales	-1804603.42		
4		Sales : Sales - Clearance	-26339.73		
5		Sales : Sales - Merchandise	-1536886.89		
6		Sales : Sales - Service	-288448.69		
7		Sales : Sales - Warranty	-3164827.72		
8		Uncategorized Income	0		
9		WIP eRev	0		
10			-6821106.45		
11					
12		Sales :*	-5016503.03		
13			4445892.54		
14		{\"Sales\", \"Cost of Goods Sold\", \"Purchases\"}	4445892.54		
15					

As you can see, you can hard code the array using the Function Arguments dialog or you can enter the array directly within the cell. By using an array, we have returned a summarized value for our Sales, Cost of Goods Sold and Purchases accounts.

## Named Ranges

When using Solution 7, you can use named ranges to reference a cell range in the function.

In this section, we will learn how to summarize your NetSuite data using Excel's Named Ranges, both directly and by using Excel's INDIRECT function.

### Exercise 12 – Working with Named Ranges

In this exercise, we will learn how to return a summarized account balance value using Excel's Named Range functionality.

Continue with 'Sheet3' in the workbook and with Solution 7 activated, follow these steps:-

#### Step 1:

Select the Accounts (By Name) list.

#### Step 2:

Using Excel's naming feature, name the account list "Income".

	A	B	C	D
1				
2				
3		Sales	-1804603.42	
4		Sales : Sales - Clearance	-26339.73	
5		Sales : Sales - Merchandise	-1536886.89	
6		Sales : Sales - Service	-288448.69	
7		Sales : Sales - Warranty	-3164827.72	
8		Uncategorized Income	0	
9		WIP eRev	0	
10			-6821106.45	
11				

#### Step 3:

In empty cell C11 (or your equivalent), click the Large *fx* button and select **NetSuite Balances > NSGLNBAL**.

#### Step 4:

Select the Subsidiary.

#### Step 5:

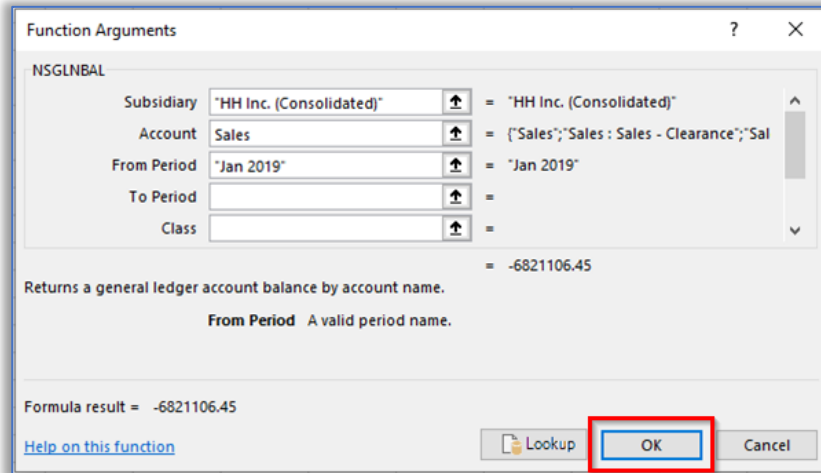
With your cursor in the Account argument, enter "Income" (without the quotes) to reference the named range.

#### Step 6:

With the cursor in the From Period argument, enter "Jan 2019".

#### Step 7:

Click OK to return a summarized value.



	A	B	C	D
1				
2				
3		Sales	-1804603.42	
4		Sales : Sales - Clearance	-26339.73	
5		Sales : Sales - Merchandise	-1536886.89	
6		Sales : Sales - Service	-288448.69	
7		Sales : Sales - Warranty	-3164827.72	
8		Uncategorized Income	0	
9		WIP eRev	0	
10			-6821106.45	
11			-6821106.45	
12		Sales :*	-5016503.03	
13			4445892.54	
14		{"Sales", "Cost of Goods Sold", "Purchases"}	4445892.54	
15				

As you can see, you can use a combination of Solution 7 and Excel functionality to return the same balance and therefore provide check values in your report.

### Exercise 13 – Working with Indirect Named Ranges

In this exercise, we will learn how to use Excel’s INDIRECT function to return a summarized balance for our expense accounts.

Begin with ‘Sheet4’ in the workbook and with Solution 7 activated, follow these steps:-

**Step 1:**

Name the Expense account numbers “Expenses”.

	A	B	C	D	E	F	G	H	I
1									
2									
3		4000	Sales	-1804603.42		6100	Insurance Expense	675569.62	
4		4002	Sales - Merchandise	-1536886.89		6200	Rent Expense		
5		4004	Sales - Service	-288448.69		6300	Utilities		
6		4006	Sales - Clearance	-26339.73		6400	Salaries & Wages Expense		
7		4008	Sales - Warranty	-3164827.72		6500	Payroll Expenses		
8		4100	WIP eRev	0		6600	Manufacturing Expenses		
9				-6821106.45					

**Step 2:**

In empty cell G10 (or your equivalent), enter “Expenses”.

	A	B	C	D	E	F	G	H	I
1									
2									
3		4000	Sales	-1804603.42		6100	Insurance Expense	675569.62	
4		4002	Sales - Merchandise	-1536886.89		6200	Rent Expense		
5		4004	Sales - Service	-288448.69		6300	Utilities		
6		4006	Sales - Clearance	-26339.73		6400	Salaries & Wages Expense		
7		4008	Sales - Warranty	-3164827.72		6500	Payroll Expenses		
8		4100	WIP eRev	0		6600	Manufacturing Expenses		
9				-6821106.45					
10							Expenses		
11									

**Step 3:**

In empty cell H10 (or your equivalent), click the Large fx button and select **NetSuite Balances > NSGLABAL**.

**Step 4:**

Select the Subsidiary.

**Step 5:**

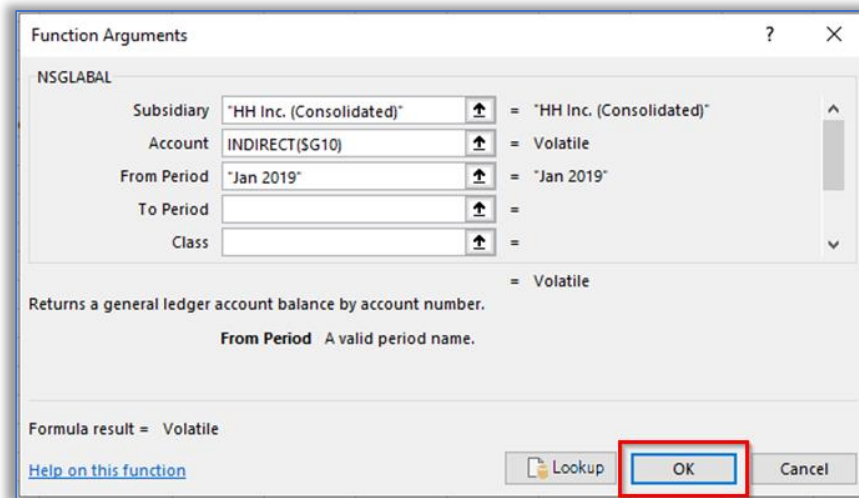
With your cursor in the Account argument, enter INDIRECT(\$G10).

**Step 6:**

With the cursor in the From Period argument, enter “Jan 2019”.

**Step 7:**

Click OK to return a summarized balance value.



	A	B	C	D	E	F	G	H	I
1									
2									
3		4000	Sales	-1804603.42		6100	Insurance Expense	675569.62	
4		4002	Sales - Merchandise	-1536886.89		6200	Rent Expense		
5		4004	Sales - Service	-288448.69		6300	Utilities		
6		4006	Sales - Clearance	-26339.73		6400	Salaries & Wages Expense		
7		4008	Sales - Warranty	-3164827.72		6500	Payroll Expenses		
8		4100	WIP eRev	0		6600	Manufacturing Expenses		
9				-6821106.45					
10							Expenses	675569.62	
11									

As you can see, you can use Excel's INDIRECT function in Solution 7 to populate other areas of your reports.

**When using indirect named ranges, the function can be 'volatile' and cause the report to refresh less efficiently.**

## Lesson Summary

In this lesson we have learned how to aggregate, or summarize, your NetSuite data using a combination of Excel and Solution 7 functionality. To understand the basic concept of aggregation, we used a simple NSGLABAL function to generate a summarized value, which we expanded into a dynamic, summarized report. We have also learned how best to use Solution 7's lists and corresponding NSGLABAL, NSGLNBAL and NSGLTBAL functions to format and populate a summarized report.

When aggregating data, we have learned how to use two different Wildcard characters to produce accurate, summarized values. We also investigated the use of arrays to drive functions and summarize values directly within the worksheet.

Finally, we have learned how to use Excel's Named Range and Indirect Named Range functionality to drive our aggregated report and improve readability.

Using all of these techniques provides total flexibility and control in how you want to present your data.

For more information on the Aggregation techniques supported by Solution 7, see our [Aggregation Reference Guide](#).

If you require any further assistance on aggregating your data within Solution 7, please contact [support@solution7.co.uk](mailto:support@solution7.co.uk)