



# DYNAMIC QUANTITY SCHEDULES – EFFECTIVELY LINKING DATA

## 2010 ACEIT USER WORKSHOP January 2010

Joseph M. Golebieski  
US Army CECOM LCMC  
Fort Monmouth, NJ

[joseph.m.golebieski@us.army.mil](mailto:joseph.m.golebieski@us.army.mil)



# Introduction

- Effectively linking data enables the analyst to make a change in one place rather than in multiple places.
- For example, Quantity Schedules:
  - The Procurement (Buy) quantity schedule is entered in the Yearly Phasing Workscreen.
  - The Fielding (Delivery) quantity schedule is usually the Procurement (Buy) quantity schedule shifted out one year.
  - The Sustainment quantity schedule is the sum of the fielded quantities throughout the system life.
  - The Reprocurement quantity schedule is repeating the Fielding (Delivery) quantity schedule every “x” years.
  - The Software Maintenance/Upgrade quantity schedule is the sustainment quantity, in a particular year, associated with a cyclical schedule.



## Introduction (Cont)

- Why have five independent quantity schedules, each with their own set of entered data?
- This presentation will describe a methodology using various ACE functions for dynamically linking a given Procurement (Buy) Quantity Schedule in order to generate:
  - Fielding (Delivery) Quantity Schedule – **FYCVal()**
  - Sustainment Quantity Schedule – **OpFieldedUnits()**
  - Reprocurement Quantity Schedule – **FYRepeat()**
  - Software Maintenance/Upgrade Quantity Schedule – **OpCycle()**
- Enabling the analyst to complete quantity schedule what-if drills in a more timely and efficient manner by making a change in only one place – the Procurement (Buy) Quantity Schedule.

# Cost Elements Effected by Quantity Schedules



- Procurement (Buy) Quantity Schedule
  - 2.021 Manufacturing
- Fielding (Delivery) Quantity Schedule
  - 2.101 Initial Depot-Level Repairables (Spares)
  - 2.102 Initial Consumables (Repair Parts)
  - 2.104 Transportation (FDT)
- Sustainment Quantity Schedule
  - 4.01 Crew
  - 4.02 Maintenance
  - 5.03 Replenishment Depot-Level Repairables (Spares)
  - 5.04 Replenishment Consumables (Repair Parts)
- Reprocurement Quantity Schedule
  - 2.14 Other Procurement
  - 5.12 Other O&M
- Software Maintenance/Upgrade Quantity Schedule
  - 2.14 Other Procurement
  - 5.08 Software



# Quantity Schedule Example

Quantity Schedule	TOTAL	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024
Procurement	24	3	4	5	6	6										
Fielding, 0 Year Lag	24	3	4	5	6	6										
Fielding, 1 Year Lag	24		3	4	5	6	6									
Fielding, 2 Year Lag	24			3	4	5	6	6								
Sustainment	240		3	7	12	18	24	24	24	24	24	24	21	17	12	6
Reprocurement	45					3	4	5	9	10	5	9				
						3	4	5	6	6						
									3	4	5	6				
													3			
Reprocurement (A)						3	4	5	9	10	5	6	6			
						3	4	5	6	6						
									3	4	5	6	6			
Software Maint/Upg	66					18			24			24				

**NOTES:**

1. Fielding (Delivery) will be 1 year after procurement.
2. System Life is 10 years.
3. Reprourement will occur every 3 years after fielding until ramp-down starts.
4. Software Maintenance/Upgrade for all quantities will occur every 3 years after fielding until ramp-down starts.



# Fielding (Delivery) Quantity Schedule

- How do you generate the Fielding (Delivery) Quantity schedule from the Procurement (Buy) Quantity schedule?
- ACE Function: Fiscal Year Calculated Value  
FYCVal ( [@var], [FY1, FY2, ...] )
- Where:
  - @var identifies the ACE row where var is the Unique ID of the row to reference.
  - FY1, FY2, ... is used to reference the years (ACE columns) in which to use for evaluating the function.

# Fielding (Delivery) Quantity Schedule Equation



- Fielding (Delivery) Quantity – equation of the form  
$$\text{FYCVal} ( @\text{Proc\_Qty}, \text{FYYR} - \text{Lag} )$$
- Where:
  - FYCVal() function returns the calculated yearly value for a specified row.
  - @Proc\_Qty identifies the ACE row where Proc\_Qty is the Unique ID of the row to reference.
  - FYYR returns as a value the fiscal year for each of the columns in your ACE session.
  - Lag is the Unique ID that references the value in years to subtract from FYYR. In the example, this is 1 year.
  - Use the “F” Phasing Method.



## How FYCVal() Calculates

- Fielding (Delivery) =  $\text{FYCVal}(@\text{Proc\_Qty}, \text{FYR} - \text{Lag})$ ,
  - where Lag = 1 and using the "F" Phasing Method

**INPUT - Proc\_Qty: FY2009 = 0; FY2010 = 3; FY2011 = 4; etc...**

- When in FY2010, FYR-1 is 2009; the value in Proc\_Qty for FY2009 = 0; therefore the FY2010 Fielding (Delivery) result = 0.
- When in FY2011, FYR-1 is 2010; the value in Proc\_Qty for FY2010 = 3; therefore the FY2011 Fielding (Delivery) result = 3.
- When in FY2012, FYR-1 is 2011; the value in Proc\_Qty for FY2011 = 4; therefore the FY2012 Fielding (Delivery) result = 4.

**RESULT - Fielding (Delivery): FY2010 = 0; FY2011 = 3; FY2012 = 4; etc...**



# Sustainment Quantity Schedule

- How do you generate the Sustainment Quantity schedule from the Fielding (Delivery) Quantity schedule, maintaining a ramp-up, steady state, ramp-down structure?
- ACE Function: Operational Fielded Units  
OpFieldedUnits ( @BQ, OpLife [, LagInYears] )
- Where:
  - @BQ – the Buy Quantity Row (@var) is where you specify how many units are being procured or delivered in each year. It must be a time phased row (i.e. phased with IS, F or %).
  - OpLife – the Operational Life span is the number of years that a unit can be in operation. It is a constant value (i.e. phased with C method).
  - @LagInYears [Optional] – the number of years between purchase and delivery.



# Sustainment Quantity Schedule Equation

- Sustainment Quantity – equation of the form  
$$\text{OpFieldedUnits} ( @\text{Field\_Qty}, \text{Sys\_Life} )$$
- Where:
  - OpFieldedUnits() function determines a time phased quantity in use through the current year based on a buy/delivery schedule, life expectancy and lag.
  - @Field\_Qty is the Unique ID where you specify how many units are being fielded (delivered) in each year.
  - Sys\_Life is the Unique ID that represents the Operational Life in years that a unit can be in operation (must be a Constant Value, C Phasing Method). In the example, this is 10 years.
  - Since the Fielded (Delivery) Quantity is being used, LagInYears = 0, which is the default value and not part of the equation.
  - Use the "F" Phasing Method.



# Reprocurement Quantity Schedule

- How do you generate the Reprocurement Quantity schedule from the Fielding (Delivery) Quantity schedule?

- ACE Function: Fiscal Year Repeat

FYRepeat ( @var, NumTimes, RepeatSize [, FY] )

- Where

- @var is the row containing the procurement/fielding schedule.
- NumTimes is the number of times you want the schedule repeated.
- RepeatSize is the number of years before repeating the schedule again.
- FY [Optional] tells which fiscal year of the repeated schedule to retrieve relative to the first schedule year; essentially defining the number of lag years.

# Reprocurement Quantity Schedule Equation



- Reprocurement Quantity – equation of the form  
FYRepeat ( @Field\_Qty, FYLAST-FYFIRST, Rep\_Cycle, FYR-Rep\_Cycle )
- Where:
  - @Field\_Qty is the row containing the fielding (delivery) quantity schedule.
  - FYLAST-FYFIRST represents NumTimes, which is the number of times you want the schedule repeated. In the example, this calculates to 32.
  - The Unique ID Rep\_Cycle represents RepeatSize, which is the number of years before repeating the schedule again. In the example, this is 3 years.
  - FYR-Rep\_Cycle represents FY [Optional], which tells which fiscal year of the repeated schedule to retrieve relative to the first schedule year; defines the number of lag years. In the example, this represents a 3 year lag.
  - Use the “F” Phasing Method.
- This equation will Repeat 32 times, every 3 years, with a 3 year lag.

# Reprocurement Quantity Schedule Equation



- Since Reprocurement will end 1 Year Prior to Ramp-Down (FY2020), the Reprocurement equation must be bounded by a Start Date and a Finish Date.
  - Start Date = Unique ID Field\_Start = 2011
  - Finish Date = FYCFirstYr(@Sust\_Qty) + Sys\_Life – 1  
= 2011 + 10 – 1 = 2020
- Although the equation will Repeat 32 times, every 3 years, with a 3 year lag, it will stop at the Finish Date, 2020.

# Another Way to Calculate the NumTimes Argument



- You know that the fielding schedule cannot be repeated beyond the last year of the sustainment schedule, which is the end of the systems operational life.
- Therefore to calculate the upper limit of NumTimes just round-up the number of sustainment years divided by the number of fielding years (Rows 66-69 in the associated ACE session):
  - Number of Fielding Years =  $\text{Field\_End} - \text{Field\_Start} + 1 = 5$
  - Number of Sustainment Years =  $\text{Sust\_End} - \text{Sust\_Start} + 1 = 14$
  - NumTimes =  $\text{RndUp}(14 / 5) = 3$
  - With Start Date and Finish Date bounds

**FYREPEAT(@Field\_Qty, RndUp(sust\_yrs/field\_yrs), Rep\_Cycle, FYR - Rep\_Cycle)**

- This equation will Repeat 3 times, every 3 years, with a 3 year lag but end at Finish Date.
- To Repeat the Whole Fielding (Delivery) Quantity Schedule “x” times, NumTimes =  $\text{RndDn}(14 / 5) = 2$ , with NO Start Date and Finish Date bounds (Rows 70-71 in the associated ACE session).

**FYREPEAT(@Field\_Qty, RndDn(sust\_yrs/field\_yrs), Rep\_Cycle, FYR - Rep\_Cycle)**

- This equation will ONLY Repeat 2 times, every 3 years, with a 3 year lag.



## How FYRepeat() Calculates

2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
<b>Fielding (Delivery) Quantities</b>										
3	4	5	6	6						Ramp
			3	4	5	6	6			Down
						3	4	5	6	6
Row 39: FYREPEAT(@Field_Qty, FYLAST - FYFIRST, Rep_Cycle, FYR - Rep_Cycle)									3	4
Row 69: FYREPEAT(@Field_Qty, RndUp(sust_yrs/field_yrs), Rep_Cycle, FYR - Rep_Cycle)										
Rows	39, 69		3	4	5	9	10	5	9	10
			3	4	5	6	6			
Row 71: FYREPEAT(@Field_Qty, RndDn(sust_yrs/field_yrs), Rep_Cycle, FYR - Rep_Cycle)						3	4	5	6	6
Row	71		3	4	5	9	10	5	6	6

# Software Maintenance/Upgrade Quantity Schedule



- How do you generate the Software Maintenance/Upgrade Quantity schedule from the Sustainment Quantity schedule?
- ACE Function: Operational Cycle  
OpCycle ( Value, StartYear, CycleYears [,Multiplier, MaxCycles] )
- Where
  - Value is the value to be entered each cycle. This can be a number or a variable.
  - StartYear is the first year the cycle begins.
  - CycleYears is the number of years between each cycle.
  - Multiplier [Optional] is used to increase or decrease the value each cycle.
  - MaxCycles [Optional] is the maximum number of cycles in the schedule.

# Software Maintenance/Upgrade Quantity Schedule Equation



- Software Maintenance/Upgrade Quantity – equation of the form  $\text{OpCycle} ( @\text{Sust\_Qty}, \text{FYCFIRSTYR}(@\text{Field\_Qty}) + \text{Rep\_Cycle}, \text{Rep\_Cycle} )$
- Where:
  - @Sust\_Qty is the row containing the sustainment quantity schedule.
  - $\text{FYCFIRSTYR}(@\text{Field\_Qty}) + \text{Rep\_Cycle}$  represents StartYear, which is the first year the cycle begins. In the example, this calculates as  $2011 + 3 = 2014$ .
  - The Unique ID Rep\_Cycle represents CycleYears, which is the number of years between each cycle. In the example, this is 3 years.
  - Use the “F” Phasing Method.

# Software Maintenance/Upgrade Quantity Schedule Equation



- Since Software Maintenance/Upgrade will end 1 Year Prior to Ramp-Down (FY2020), the Software Maintenance/Upgrade equation must be bounded by a Start Date and a Finish Date.
  - Start Date = Unique ID Field\_Start = 2011
  - Finish Date = FYCFirstYr(@Sust\_Qty) + Sys\_Life – 1  
= 2011 + 10 – 1 = 2020

# ACE Example Session Screen Shot



ACE 7.1a - [2010 Quantity Schedule Example.aceit - Methodology (BY2008SK)]

File Edit View Documentation Calc Cases Reports Tools Window Help

Methodology

\*INPUT VARIABLE \*INPUT VARIABLES

	WBS/CES Description	Unique ID	Point Estimate	Phasing Method	Equation / Throughput	Start Date	Finish Date
14	*INPUT VARIABLES	*IN_VAR					
15							
16	***PROCUREMENT SCHEDULE	*Proc_Qty					
17	Procurement Quantity	Proc_Qty	24. *	IS	[Input Throughput]		
18	*** FY Schedule Information						
19	Production Start	Prod_Start	2010 *	C	FYCFIRSTYR(@Proc_Qty)		
20	Production End	Prod_End	2014 *	C	FYCLASTYR(@Proc_Qty)		
21	***FIELDING SCHEDULE	*Field_Qty					
22	*Calculate fielding schedule from procurement schedule						
23	System Procurement to Fielding Lag in Years	Lag	1. *	C	1		
24	Fielding Quantity	Field_Qty	24. *	F	FYCVL(@Proc_Qty, FYR Lag)		
25	*** FY Schedule Information						
26	Fielding Start	Field_Start	2011 *	C	FYCFIRSTYR(@Field_Qty)		
27	Fielding End	Field_End	2015 *	C	FYCLASTYR(@Field_Qty)		
28	***SUSTAINMENT SCHEDULE	*Sust_Qty					
29	*Calculate sustainment schedule (ramp up-steady state-ramp down)						
30	System Life in years (sustainment years)	sys_life	10. *	C	10		
31	Sustainment Quantity	SUST_Qty	240. *	F	OPFIELDEDUNITS(@Field_Qty, sys_life)		
32	*** FY Schedule Information						
33	Sustainment Start	Sust_Start	2011 *	C	FYCFIRSTYR(@SUST_Qty)		
34	Sustainment End	Sust_End	2024 *	C	FYCLASTYR(@SUST_Qty)		

# ACE Example Session Screen Shot (Cont)



ACE 7.1a - [2010 Quantity Schedule Example.aceit - Methodology (BY2008\$K)]

File Edit View Documentation Calc Cases Reports Tools Window Help

Methodology

\*\*\*REPROCURE \*\*\*REPROCUREMENT SCHEDULE

	WBS/CES Description	Unique ID	Point Estimate	Phasing Method	Equation / Throughput	Start Date	Finish Date
36							
37	***REPROCUREMENT SCHEDULE	*Reproc_Qty					
38	Recprocurement Cycle in years (every x years)	Rep_Cycle	3. *	C		3	
39	Recprocurement Quantity based on Fielding Quantity		45. *	F	FYREPEAT@Field_Qty, FYLAST - FYFIRST, Rep_Cycle, FYR - Rep_Cycle)	[>=] Field_Start	[<=] FYCFIRSTYR(@SUST_Qty) + sys_life - 1
40							
41							
42							
43							
44							
45							
46							
47							
48							
49	***SOFTWARE MAINTENANCE/UPGRADE SCHEDULE	*Sitr_Maint_Qty					
50	Maintenance/Upgrade Quantity based on Sustainment Quantity		66. *	F	OPCYCLE(SUST_Qty, FYCFIRSTYR(@Field_Qty) + Rep_Cycle, Rep_Cycle)	[>=] Field_Start	[<=] FYCFIRSTYR(@SUST_Qty) + sys_life - 1
51							
52							
53							
54							
55							
56							
57							
58							
59							
60							

# ACE Example Session Phased Report



Cost Element	Total	FY 2008	FY 2009	FY 2010	FY 2011	FY 2012	FY 2013	FY 2014	FY 2015	FY 2016	FY 2017	FY 2018	FY 2019	FY 2020	FY 2021	FY 2022	FY 2023	FY 2024	FY 2025	FY 2026
16 ***PROCUREMENT SCHEDULE																				
17 Procurement Quantity	24.			3.	4.	5.	6.	6.												
18 *** FY Schedule Information																				
19 Production Start	2010																			
20 Production End	2014																			
21 ***FIELDING SCHEDULE																				
22 *Calculate fielding schedule from procurement schedule																				
23 System Procurement to Fielding Lag in Years	1.																			
24 Fielding Quantity	24.				3.	4.	5.	6.	6.											
25 *** FY Schedule Information																				
26 Fielding Start	2011																			
27 Fielding End	2015																			
28 ***SUSTAINMENT SCHEDULE																				
29 *Calculate sustainment schedule (ramp up-steady state-ramp down)																				
30 System Life in years (sustainment years)	10.																			
31 Sustainment Quantity	240.				3.	7.	12.	18.	24.	24.	24.	24.	24.	24.	21.	17.	12.	6.		
32 *** FY Schedule Information																				
33 Sustainment Start	2011																			
34 Sustainment End	2024																			
35																				
36																				
37 ***REPROCUREMENT SCHEDULE																				
38 Reprourement Cycle in years (every x years)	3.																			
39 Reprourement Quantity based on Fielding Quantity	45.							3.	4.	5.	9.	10.	5.	9.						
40																				
41																				
42																				
43																				
44																				
45																				
46																				
47																				
48																				
49 ***SOFTWARE MAINTENANCE/UPGRADE SCHEDULE																				
50 Maintenance/Upgrade Quantity based on Sustainment Quantity	66.								18.		24.		24.							
51																				
52																				
53																				
54																				
55																				
56																				
57																				
58																				
59																				



# Reprocurement and Software Maintenance/Upgrade Costing Guidance



- Higher Headquarters general consensus for funding the Reprocurement cycle and the Software Maintenance/Upgrade cycle is to buy it with OPA funds until you are no longer in production, then switch over to OMA funds.
- Require two Reprocurement Quantity Equations and two Software Maintenance/Upgrade Equations
  - Equation for OPA funds
  - Equation for OMA funds

# Revised Reprourement Quantity Schedule and Software Maintenance/Upgrade Equations



- The same Reprourement Equation will be used for the OPA fund line and the OMA fund line.

$FYRepeat ( @Field\_Qty, FYLAST-FYFIRST, Rep\_Cycle, FYR-Rep\_Cycle )$

- The same Software Maintenance/Upgrade Equation will be used for the OPA fund line and the OMA fund line.

$OpCycle ( @Sust\_Qty, FYCFIRSTYR(@Field\_Qty) + Rep\_Cycle, Rep\_Cycle )$

- But the Start Date and Finish Date bounds will be different for the OPA fund line and the OMA fund line.

# Revised Reprourement Quantity Schedule Equation



- OPA Fund Line.
  - Start Date = Unique ID Prod\_Start = 2010
  - Finish Date = Unique ID Prod\_End = 2014
- OMA Fund Line
  - Start Date = Prod\_End + 1 = 2015
  - Finish Date = FYCFirstYr(@Sust\_Qty) + Sys\_Life – 1  
= 2011 + 10 – 1 = 2020

# Revised ACE Example Session Screen Shot



ACE 7.1a - [2010 Quantity Schedule Example.aceit - Methodology (BY2008SK1)]

ACE 7.1a - [2010 Quantity Schedule Example.aceit - Methodology (BY2008SK)]

Methodology

\*\*\*REPROCURE \*\*\*REPROCUREMENT SCHEDULE

	WBS/CES Description	Unique ID	Point Estimate	Phasing Method	Equation / Throughput	Start Date	Finish Date
37	***REPROCUREMENT SCHEDULE	*Reproc_Qty					
38	Reprocurement Cycle in years (every x years)	Rep_Cycle	3. *	C		3	
39							
40							
41							
42	*Reprocurement quantities split appropriations OPA and OMA						
43	*Reprocurement Quantities Within Procurement Years - OPA						
44	*Reprocurement Quantities After Procurement Years - OMA						
45	*See Start Date and Finish Date Columns						
46	*Ends 1 Year Prior to the Sustainment Ramp-Down of the Particular It						
47	Reprocurement Quantity (OPA) based on Fielding Quantity	ReprocOPA_Qty	3. *	F	FYREPEAT(@Field_Qty, FYLAST - FYFIRST, Rep_Cycle, FYR - Rep_Cycle)	[>=] Prod_Start	[<=] Prod_End
48	Reprocurement Quantity (OMA) based on Fielding Quantity	ReprocOMA_Qty	42. *	F	FYREPEAT(@Field_Qty, FYLAST - FYFIRST, Rep_Cycle, FYR - Rep_Cycle)	[>+] Prod_End + 1	[<=] FYCFIRSTYR(@SUST_Qty) - 1 + sys_life
49	***SOFTWARE MAINTENANCE/UPGRADE SCHEDULE	*Sftr_Maint_Qty					
50							
51							
52	*Software Maintenance/Upgrade quantities split appropriations OPA and OMA						
53	*Software Maintenance/Upgrade Quantities Within Procurement Year						
54	*Software Maintenance/Upgrade After Procurement Years - OMA						
55	*See Start Date and Finish Date Columns						
56	*Ends 1 Year Prior to the Sustainment Ramp-Down of the Particular It						
57	Maintenance/Upgrade Quantity (OPA) based on Sustainment Quantity		18. *	F	OPCYCLE(SUST_Qty, FYCFIRSTYR(@Field_Qty) + Rep_Cycle, Rep_Cycle)	[>=] Prod_Start	[<=] Prod_End
58	Maintenance/Upgrade Quantity (OMA) based on Sustainment Quantity		48. *	F	OPCYCLE(SUST_Qty, FYCFIRSTYR(@Field_Qty) + Rep_Cycle, Rep_Cycle)	[>+] Prod_End + 1	[<=] FYCFIRSTYR(@SUST_Qty) - 1 + sys_life
59							
60							

# Revised

# ACE Example Session Phased Report



ACE 7.1a - [2010 Quantity Schedule Example.aceit - BY Phased Costs (FY2008 SK, Time Phased, Case: Point Estimate)]		Total	FY 2008	FY 2009	FY 2010	FY 2011	FY 2012	FY 2013	FY 2014	FY 2015	FY 2016	FY 2017	FY 2018	FY 2019	FY 2020	FY 2021	FY 2022	FY 2023	FY 2024	FY 2025	FY 2026
16	***PROCUREMENT SCHEDULE																				
17	Procurement Quantity	24.			3.	4.	5.	6.	6.												
18	*** FY Schedule Information																				
19	Production Start	2010																			
20	Production End	2014																			
21	***FIELDING SCHEDULE																				
22	*Calculate fielding schedule from procurement schedule																				
23	System Procurement to Fielding Lag in Years	1.																			
24	Fielding Quantity	24.			3.	4.	5.	6.	6.												
25	*** FY Schedule Information																				
26	Fielding Start	2011																			
27	Fielding End	2015																			
28	***SUSTAINMENT SCHEDULE																				
29	*Calculate sustainment schedule (ramp up-steady state-ramp down)																				
30	System Life in years (sustainment years)	10.																			
31	Sustainment Quantity	240.			3.	7.	12.	18.	24.	24.	24.	24.	24.	24.	24.	21.	17.	12.	6.		
32	*** FY Schedule Information																				
33	Sustainment Start	2011																			
34	Sustainment End	2024																			
35																					
36																					
37	***REPROCUREMENT SCHEDULE																				
38	Reprocurement Cycle in years (every x years)	3.																			
39																					
40																					
41																					
42	*Reprocurement quantities split appropriations OPA and OMA																				
43	*Reprocurement Quantities Within Procurement Years - OPA																				
44	*Reprocurement Quantities After Procurement Years - OMA																				
45	*See Start Date and Finish Date Columns																				
46	*Ends 1 Year Prior to the Sustainment Ramp-Down of the Procurement																				
47	Reprocurement Quantity (OPA) based on Fielding Quantity	3.							3.												
48	Reprocurement Quantity (OMA) based on Fielding Quantity	42.								4.	5.	9.	10.	5.	9.						
49	***SOFTWARE MAINTENANCE/UPGRADE SCHEDULE																				
50																					
51																					
52	*Software Maintenance/Upgrade quantities split appropriations																				
53	*Software Maintenance/Upgrade Quantities Within Procurement Years																				
54	*Software Maintenance/Upgrade After Procurement Years - OMA																				
55	*See Start Date and Finish Date Columns																				
56	*Ends 1 Year Prior to the Sustainment Ramp-Down of the Procurement																				
57	Maintenance/Upgrade Quantity (OPA) based on Sustainment	18.							18.												
58	Maintenance/Upgrade Quantity (OMA) based on Sustainment	48.									24.			24.							
59																					