How to refresh STG DBs

Author:Andrew Jia

First version:2017-11-22

Update:2018-07-04

# 1.Stop service and cache on servers(AUI&Cache)

# 2.Get latest “ClearProductionData.sql” and submit a ticket to DBA

## Ticket

Ticket detail may like this—

Reason: refresh perf STG DB from live customer data.   
Source DB Server: DBANET10vs  
DB name: naparec  
  
Target DB server: DBSTGANET01.dev.activenetwork.com  
DB name: lstgnaparec  
  
Steps:  
1. Backup source DB "naparec" and copy the backup file to targer DB server's path (G:\ajia)  
2. Restore to target DB "lstgnaparec"  
3. Run ClearProductionData.sql for restored DB - ACM01VegasJetty  
4. I'll run SQLs to insert/modify data to make it for performance testing.

## Latest “ClearUpProdData.sql”

1. Access this URL- <https://gitlab.dev.activenetwork.com/ActiveNet> with your account
2. Find Database related floder( <https://gitlab.dev.activenetwork.com/ActiveNet/activenet-servlet/blob/develop/KitManager/Databases/ActiveNet/DataScriptsNoDeploy/ClearUpProdData.sql>)
3. Find “ClearUpProdData.sql” and save it on local
4. [Option]Replace “delete from dbo.ICVERIFYLOG;” with

USE [DBName]  
GO  
  
ALTER TABLE [dbo].[RECEIPTPAYMENTSPENDING]   
DROP CONSTRAINT [FK\_RECEIPTPAYMENTSPENDING\_\_ICVERIFYLOG\_ID]  
  
truncate table dbo.ICVERIFYLOG;  
  
ALTER TABLE [dbo].[RECEIPTPAYMENTSPENDING] WITH NOCHECK ADD CONSTRAINT [FK\_RECEIPTPAYMENTSPENDING\_\_ICVERIFYLOG\_ID] FOREIGN KEY([ICVERIFYLOG\_ID])  
REFERENCES [dbo].[ICVERIFYLOG] ([ICVERIFYLOG\_ID])  
GO  
  
ALTER TABLE [dbo].[RECEIPTPAYMENTSPENDING]   
CHECK CONSTRAINT [FK\_RECEIPTPAYMENTSPENDING\_\_ICVERIFYLOG\_ID]  
GO

1. Run “ClearUpProdData.sql” on SSMS

# 3. Run follow sqls to change identifier via SSMS

use DBName

select \* from SYSTEMINFO where keyword like '%identi%'

update SYSTEMINFO set keywordvalue = 'Stage' where keyword = 'environment\_identifier'

select \* from SYSTEMINFO where keyword like '%identi%'

# 4. Change Org mode to ignite remote

use DBName

if not exists ( select \* from systeminfo where keyword = 'org\_cache\_mode' )

BEGIN

INSERT INTO systeminfo (KEYWORD,KEYWORDVALUE) VALUES ('org\_cache\_mode','REMOTE')

END

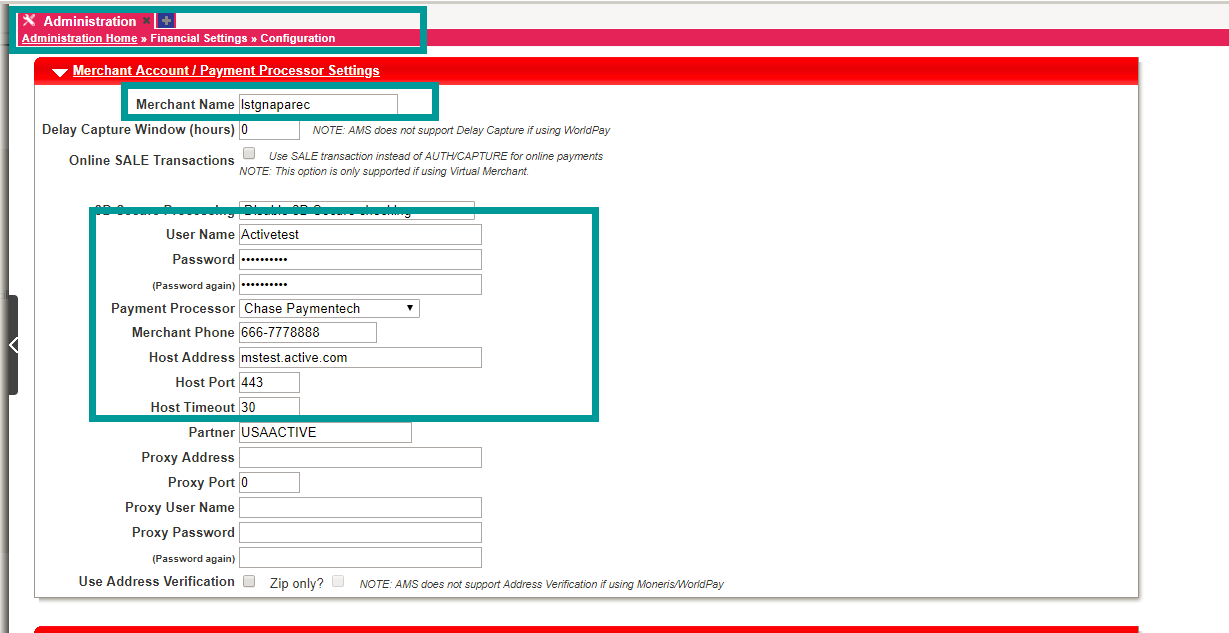
else

BEGIN

update systeminfo set KEYWORDVALUE = 'REMOTE' where KEYWORD = 'org\_cache\_mode'

END

# 5. Update paymenet settings on AUI like below:



Merchant Name 🡪 test org name

Detail account info 🡪 Username\PWD: Activetest\ Activetest

Payment Processer 🡪 Chase Paymentech

Merchant Phone 🡪 666-7778888

Host Address 🡪 mstest.active.com

# Allow login as CustomerID

Run sqls like below

use DBName

select \* from SYSTEMINFO where keyword = 'force\_email\_as\_login\_name'

update systeminfo set keywordvalue = 'false' where keyword = 'force\_email\_as\_login\_name'

select \* from SYSTEMINFO where keyword = 'force\_email\_as\_login\_name'

# Insert perf customer type

Run sqls like:

use DBName

if not exists (select 1 FROM CUSTOMERTYPES where DESCRIPTION like '%STG Perf%')

BEGIN

insert CUSTOMERTYPES (DESCRIPTION,ABSORB\_CONVENIENCE\_FEE,ALLOWED\_ONLINE,HIDE\_FROM\_CUSTOMERS,USE\_TYPE)

values('STG Perf',0,-1,0,0)

END

# Enable cache control

USE DBName

if not exists (select 1 FROM systeminfo where KEYWORD = 'enable\_cache\_control\_for\_non\_prod\_site')

BEGIN

insert into systeminfo (keyword,keywordvalue) values ('enable\_cache\_control\_for\_non\_prod\_site','true')

END

# Set CustomerQuestion as not required

use DBName

update [CUSTOMQUESTIONS] set required = 0 where required = -1

# UpdateCashSummarySheetPrompt

use DBName

update [WORKSTATIONS] set [CSS\_PROMPT\_TYPE] = 0

# Disable DB autobackup if needed

Replace table name in sqls with yours

use lstgbreckenridgerec

update systeminfo set keywordvalue = '/opt/active/ActiveNet/perf/SQLBACKUPS/lstgbreckenridgerec' where keyword = 'full\_backup\_unc'

update systeminfo set keywordvalue = 'W:\perf\SQLBACKUPS\lstgbreckenridgerec' where keyword = 'full\_backup\_unc\_remote'

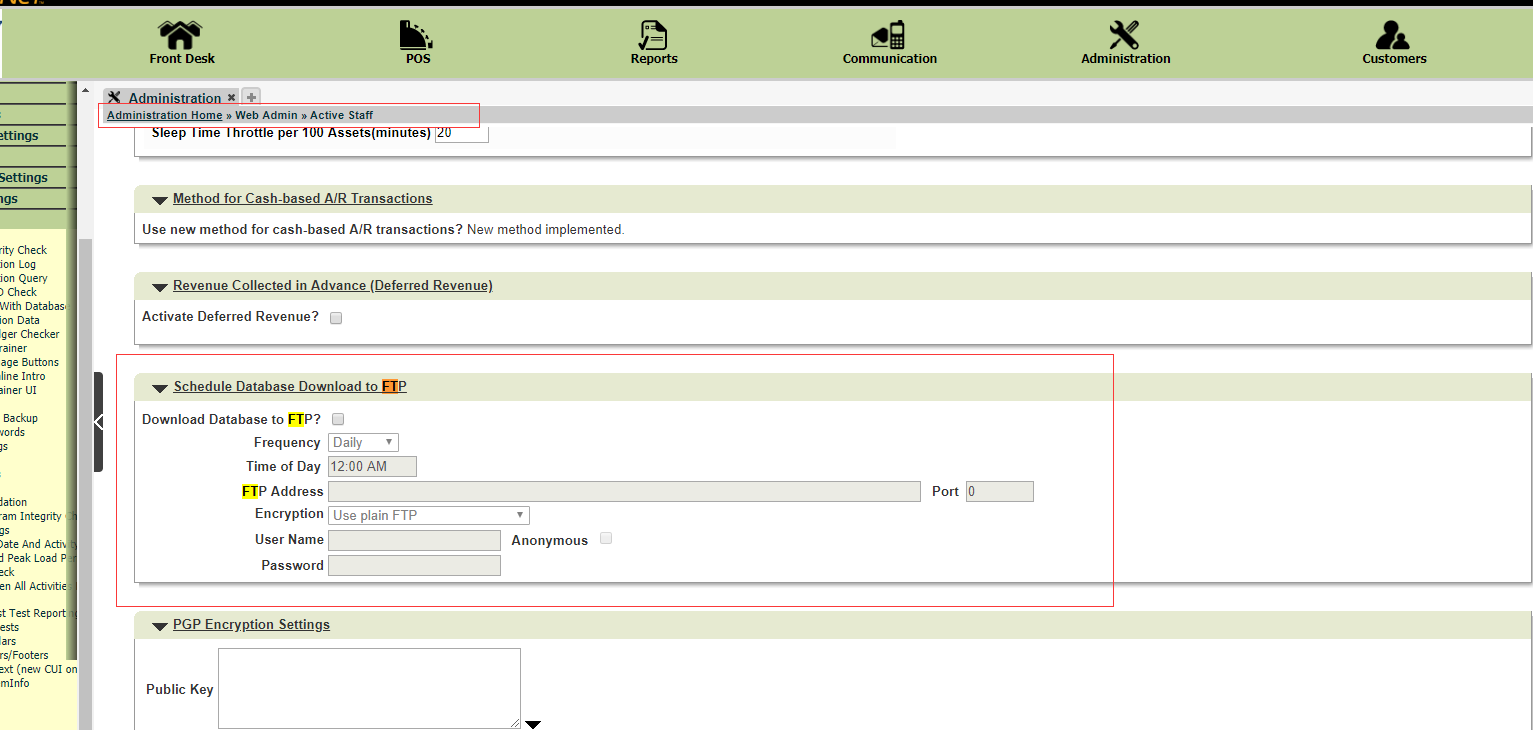
update systeminfo set keywordvalue = 'X:\lstgbreckenridgerec' where keyword = 'image\_storage\_path'

update systeminfo set keywordvalue = '/opt/active/data/an\_filedata/lstgbreckenridgerec' where keyword = 'image\_storage\_path\_local'

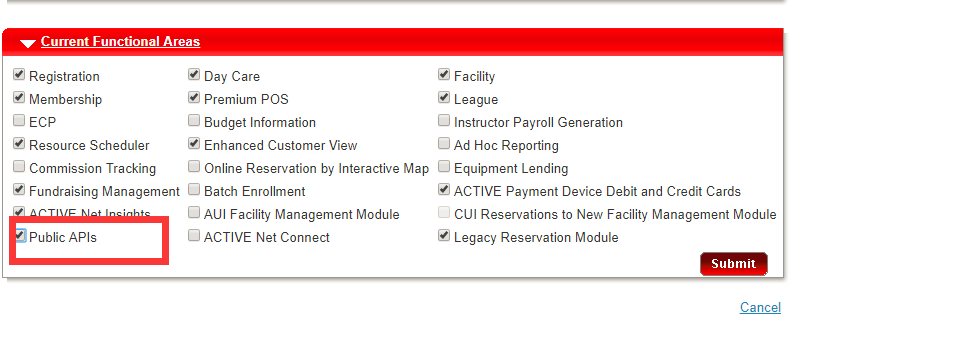
update systeminfo set keywordvalue = '/lstgbreckenridgerec/jreport/' where keyword = 'jreport\_jsp\_path'

select \* from systeminfo where keyword in ('full\_backup\_unc','full\_backup\_unc\_remote','image\_storage\_path','image\_storage\_path\_local','jreport\_jsp\_path')

Check AUI if the auto backup is open.



# Check if org license is correct.

OpenAPI :

# Add perf activities

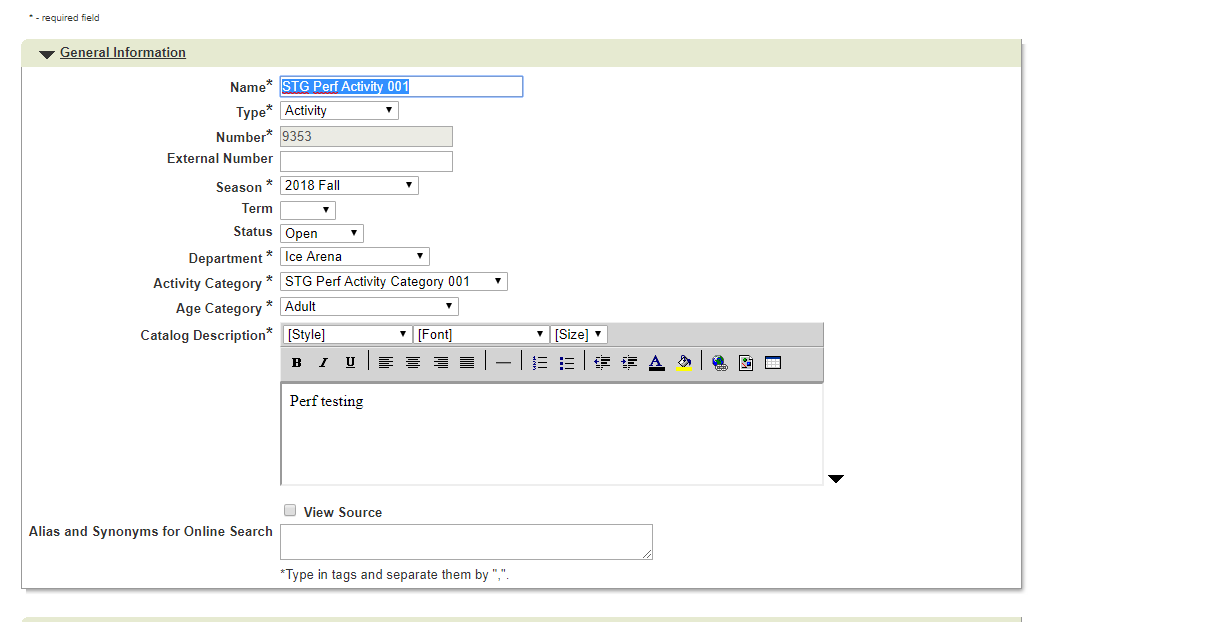
## Activities

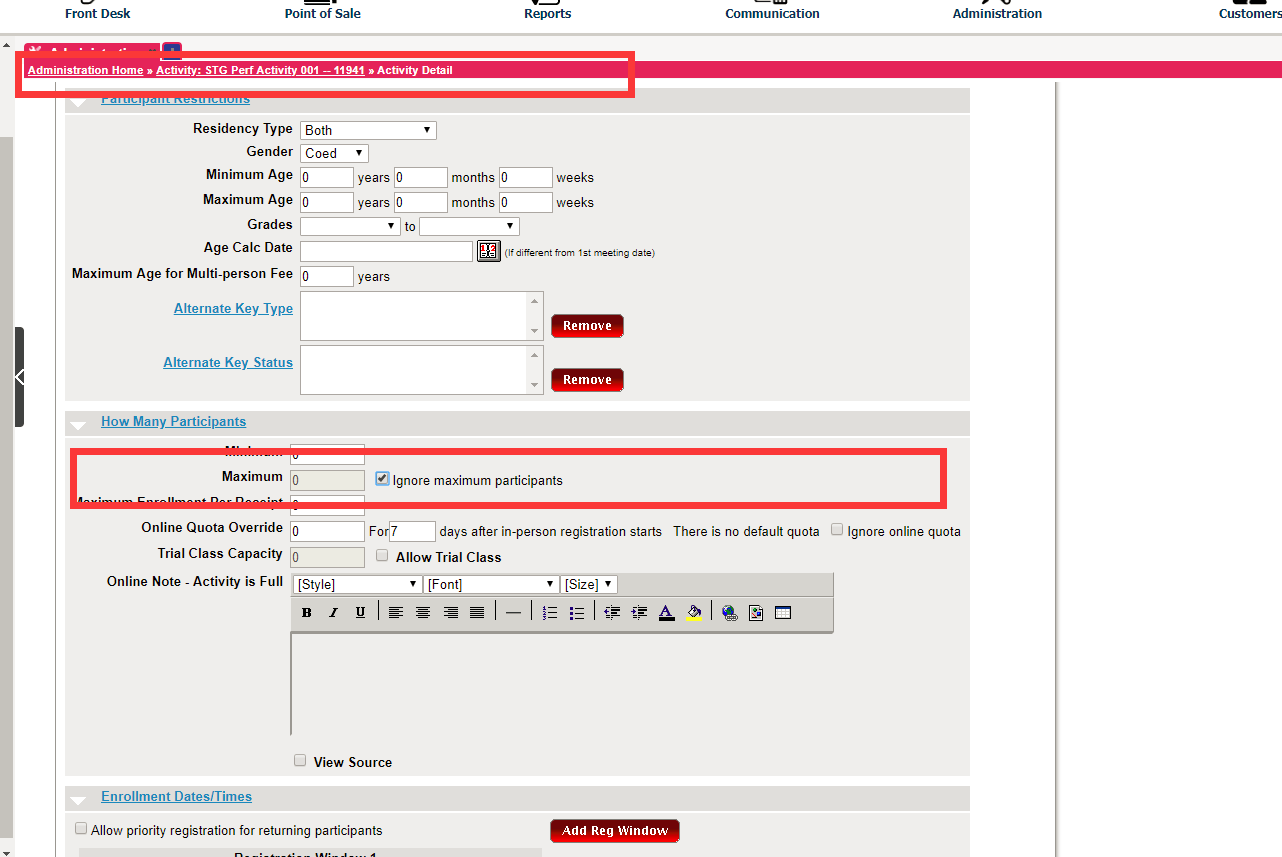
select Activity\_id from ACTIVITIES where ACTIVITYNAME like '%STG%'

1. New acticity category 🡪 [**Administration Home**](https://anstg.active.com/lstgymcala/servlet/anstg.active.com/lstgymcala/servlet/homepage.sdi?topmost_menu_id=tab_admin)**» Registration Settings » Activity Program Categories**
2. New Activities 🡪 Administration Home » Activity Search » Add New

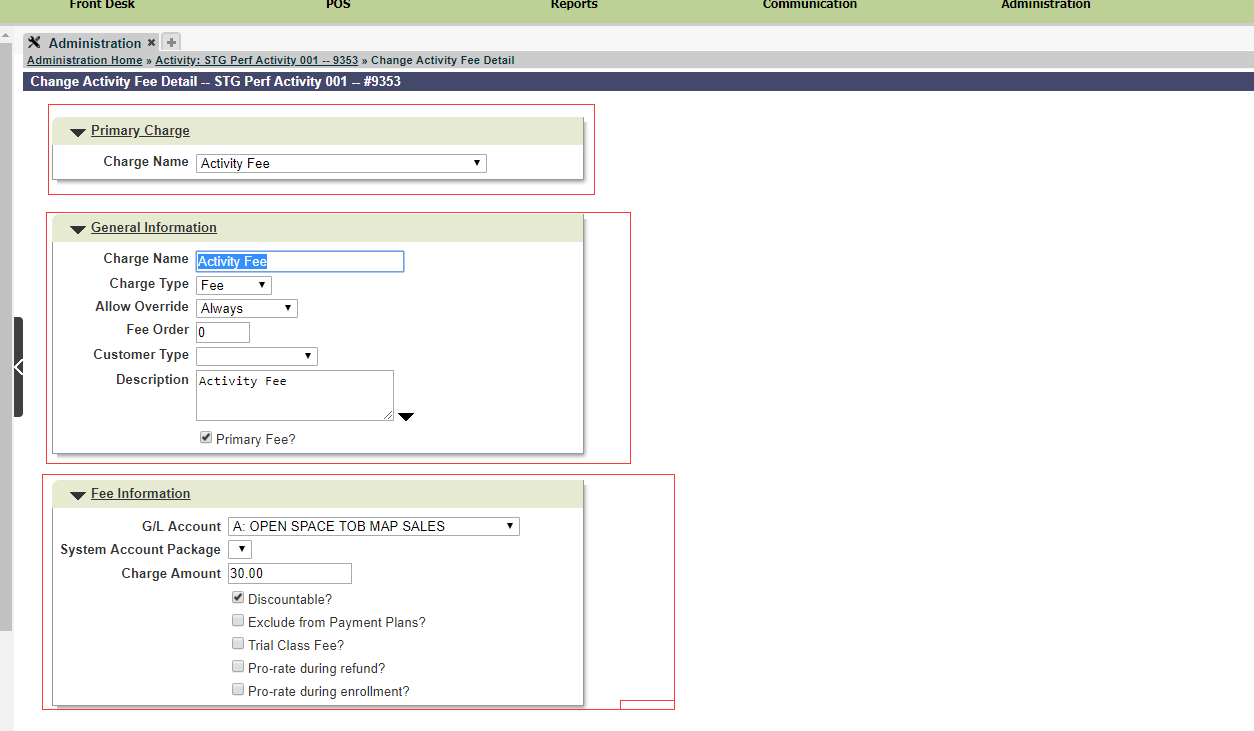
Key point:

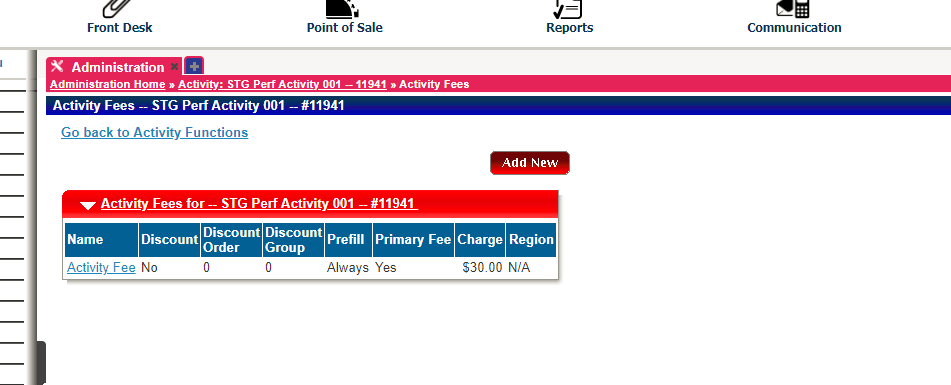
Select “ignore maximum paticipants”





Add activite fee:





Key poiot:

Only activities fee is required.

Update activity info:

use DBName

-- update SEASONS set

-- KEYBOARDDATE = '1899-12-30 00:00:00.000',

-- INTERNETDATE = '1899-12-30 00:00:00.000',

-- INTERNETENDDATE = '2017-12-31 00:00:00.000'

-- where seasonname like '2016%'

update ACTIVITIES set

activitygender=0

,agesmax=99

,agesmin=0

,nointernetreg=0

,enrollmax=9999

,maxenrolledonline=9999

,internetdate='12/30/1899'

,internetenddate='12/30/1899'

,endingdate='12/31/2018'

,keybrdentryenddate='12/30/1899'

,DONTRESERVEFACILITIES=-1

where ACTIVITY\_ID = (select Activity\_id from ACTIVITIES where ACTIVITYNAME like '%STG%')

update ACTIVITYREGISTRATIONWINDOWS set

INTERNET\_DATE='1/1/2018'

,INTERNET\_END\_DATE='12/30/1899'

,[KEYBOARD\_ENTRY\_DATE]='1/1/2018'

,[NON\_RES\_KEYBOARD\_ENTRY\_DATE]='1/1/2018'

,[MEMBER\_KEYBOARD\_ENTRY\_DATE]='1/1/2018'

,[NON\_RES\_INTERNET\_DATE]='1/1/2018'

,[MEMBER\_INTERNET\_DATE]='1/1/2018'

where activity\_id = (select Activity\_id from ACTIVITIES where ACTIVITYNAME like '%STG%')

update ACTIVITYSTATISTICS set number\_open = 888

where activity\_id = (select Activity\_id from ACTIVITIES where ACTIVITYNAME like '%STG%')

## Customers

1. Create customer

Replace DBName with yours

declare @x integer

declare @y integer

declare @z NVARCHAR(50)

use DBName

-- actually, lets just fill in the big hole from the end of current customers up to 1000000

set @x = 200000 --max customers

set @y = 100001

set @z = CONVERT(NVARCHAR,@y)

print convert(nvarchar, @x)

SET NOCOUNT ON

set identity\_insert CUSTOMERS on

while @y <= @x

begin

if not exists (select customer\_id from CUSTOMERS where CUSTOMER\_ID=@y)

begin

insert CUSTOMERS (CUSTOMER\_ID, FIRSTNAME, LASTNAME, PASSWORD1) values (@y, 'F' + @z, 'L'+ @z, @y)

update customers set

city = 'Dallas'

, customertype\_id = 4

, state = 'TX'

, ZIPCODE = '75201'

, ADDRESS1 = '717 North Harwood Street'

, ADDRESS2 = 'Suite ' + convert(nvarchar, CUSTOMER\_ID)

, HOMEPHONE = LEFT( '804' + convert(nvarchar, @y)+ '000000',10)

, WORKPHONE = LEFT( '804' + convert(nvarchar, @y)+ '000000',10)

, CEllphone = LEFT( '804' + convert(nvarchar, @y)+ '000000',10)

, FAXPHONE = LEFT( '804' + convert(nvarchar, @y)+ '000000',10)

, PAGERPHONE = LEFT( '804' + convert(nvarchar, @y)+ '000000',10)

, OTHERPHONE = LEFT( '804' + convert(nvarchar, @y)+ '000000',10)

, MAILINGCITY = 'Dallas'

, MAILINGSTATE = 'TX'

, MAILINGZIPCODE = '75201'

, ssn = ''

, question = 'futureville'

, ANSWER = 'futureville'

, EMERGENCYFNAME1 = 'E'+ FirstName

, EMERGENCYLNAME1 = 'E' + LastName

, EMERGENCYPHONE1 = LEFT( '804' + convert(nvarchar, @y)+ '000000',10)

, EMERGENCYPHONE2 = ''

, EMERGENCY\_OTHER\_PHONE1 = ''

, EMERGENCY\_OTHER\_PHONE2 = ''

, EMERGENCYRELATION1 = 'Family Member'

, EMAIL = 'Mail' + @z + '@test.com'

, PASSWORD1 = convert(nvarchar, customer\_id)

, PASSWORD2 = convert(nvarchar, customer\_id)

, BIRTHDATE = '1/1/1981'

, gender = CAST(RAND() AS decimal(1,0)) + 1

, COUNTY = 'Dallas'

, COUNTRY = 'US'

, ADDITIONAL\_EMAIL = 'ADDITIONALMail' + @z + '@test.com'

where CUSTOMER\_ID=@y

end

else

begin

update customers set

PASSWORD1 = convert(nvarchar, customer\_id)

, PASSWORD2 = convert(nvarchar, customer\_id)

, RETIRED = 0

, EMAIL = 'Mail' + @z + '@test.com'

where CUSTOMER\_ID=@y

end

if @y = 10000 print @z

else if @y = 15000 print @z

else if @y = 20000 print @z

else if @y = 25000 print @z

else if @y = 30000 print @z

else if @y = 35000 print @z

else if @y = 40000 print @z

else if @y = 45000 print @z

else if @y = 50000 print @z

else if @y = 55000 print @z

set @y=@y+1

set @z = CONVERT(NVARCHAR,@y)

end

set identity\_insert customers off

print @z

1. Create family member

Replace DBName with yours

-- Add family members to customer table

-- @Wayne Ran, 2016-05-18

declare @x integer

declare @y integer

declare @z NVARCHAR(50)

declare @totalCustomer integer

use lstgbreckenridgerec

-- actually, lets just fill in the big hole from the end of current customers up to 1000000

set @x = 200000 -- Max customers

set @y = 100001

set @z = CONVERT(NVARCHAR,@y)

-- If missing any customerID between @y and @x, then quit this script!

set @totalCustomer = (select count(\*) from customers where customer\_id between @y and @x)

if @totalCustomer < (@x-@y+1)

begin

print 'Missing Customer! Quit!'

select @totalCustomer

return

end

-- If exists family member which customerID between @y+10000000 and @x+10000000, the quit this script!

if exists(select customer\_id from CUSTOMERS where CUSTOMER\_ID between (@y+10000000) and (@x+10000000))

begin

select top 100 \* from CUSTOMERS where CUSTOMER\_ID between (@y+10000000) and (@x+10000000)

return

end

print convert(nvarchar, @x)

SET NOCOUNT ON

set identity\_insert CUSTOMERS on

while @y <= @x

begin

insert CUSTOMERS (

customer\_id

, FIRSTNAME

, LASTNAME

, customertype\_id

, city

, state

, ZIPCODE

, ADDRESS1

, ADDRESS2

, HOMEPHONE

, WORKPHONE

, CEllphone

, FAXPHONE

, PAGERPHONE

, OTHERPHONE

, MAILINGCITY

, MAILINGSTATE

, MAILINGZIPCODE

, BIRTHDATE

, GENDER

,IS\_ENC\_PASSWORD

) values (

@y + 10000000

, 'FamilyOfF' + @z

, 'L'+ @z

, 4

, 'Dallas'

, 'TX'

, '75201'

, '717 North Harwood Street'

, 'Suite ' + convert(nvarchar, @y)

, LEFT( '804' + convert(nvarchar, @y)+ '000000',10)

, LEFT( '804' + convert(nvarchar, @y)+ '000000',10)

, LEFT( '804' + convert(nvarchar, @y)+ '000000',10)

, LEFT( '804' + convert(nvarchar, @y)+ '000000',10)

, LEFT( '804' + convert(nvarchar, @y)+ '000000',10)

, LEFT( '804' + convert(nvarchar, @y)+ '000000',10)

, 'Dallas'

, 'TX'

, '75201'

, '1/1/2012'

, CAST(RAND() AS decimal(1,0)) + 1

, 0

)

set @y=@y+1

set @z = CONVERT(NVARCHAR,@y)

end

set identity\_insert customers off

print @z

1. CreateFamily

Replace DBName with yours

-- Make sure there are already customer which ID between 1000001 - 9999999

-- @Wayne Ran, 2016-05-18

declare @x integer

declare @y integer

declare @z NVARCHAR(50)

declare @totalCustomer integer

declare @sql NVARCHAR(max)

declare @sqlLoop integer

use DBName

set @x = 200000 -- Max customers

set @y = 100001

set @z = CONVERT(NVARCHAR,@y)

set @sqlLoop = 1

-- If missing any customerID between @y and @x, then quit this script!

set @totalCustomer = (select count(\*) from customers where customer\_id between @y and @x)

if @totalCustomer < (@x-@y+1)

begin

print 'Missing Customer! Quit!'

select @totalCustomer

return

end

print convert(nvarchar, @x)

SET NOCOUNT ON

SET identity\_insert FAMILIES on

set @sql = 'insert into FAMILIES (FAMILY\_ID, FAMILYNAME, EXTERNALID)'

while @y <= @x

begin

if not exists (select FAMILY\_ID FROM FAMILIES where FAMILY\_ID = @y )

BEGIN

set @sql = @sql + ' select '+ @z + ', ' + '''L' + @z + ''','+ '0' + ' union all'

END

if @sqlLoop = 100

begin

set @sql = substring(@sql,1,len(@sql)-9)

--print @sql

execute(@sql)

set @sqlLoop = 0

set @sql = 'insert into FAMILIES (FAMILY\_ID, FAMILYNAME, EXTERNALID) '

end

set @sqlLoop = @sqlLoop +1

set @y=@y+1

set @z = CONVERT(NVARCHAR,@y)

end

if len(@sql) > 60

begin

set @sql = substring(@sql,1,len(@sql)-9)

execute(@sql)

--print @sql

end

SET identity\_insert FAMILIES off

print @sql

print @z

1. AddFamilyMember

Replace DBname with yours

-- Add family members to familyMember table

-- @Wayne Ran, 2016-05-18

declare @x integer

declare @y integer

declare @z NVARCHAR(50)

declare @sql NVARCHAR(max)

declare @sqlLoop integer

use DBName

-- actually, lets just fill in the big hole from the end of current customers up to 1000000

set @x = 200000 -- Max customers

set @y = 100001

set @z = CONVERT(NVARCHAR,@y)

set @sqlLoop = 1

print convert(nvarchar, @x)

if exists (select familymember\_id FROM FAMILYMEMBERS where (customer\_id between @y and @x)

or (customer\_id between @y+10000000 and @x+10000000))

BEGIN

select top 100 \* from FAMILYMEMBERS where (customer\_id between @y and @x) or (customer\_id between @y+10000000 and @x+10000000)

return

END

SET NOCOUNT ON

set @sql = 'insert FAMILYMEMBERS (customer\_id, family\_id, role, familyrole\_id) '

while @y <= @x

begin

set @sql = @sql + ' select '+ @z + ',' + @z + ','+ '0' + ','+ '4' +' union all'

set @sql = @sql + ' select '+ CONVERT(NVARCHAR,(@y+10000000)) + ',' + @z + ','+ '0' + ','+ '4' +' union all'

if @sqlLoop = 50

begin

set @sql = substring(@sql,1,len(@sql)-9)

--print @sql

execute(@sql)

set @sqlLoop = 0

set @sql = 'insert FAMILYMEMBERS (customer\_id, family\_id, role, familyrole\_id) '

end

set @sqlLoop = @sqlLoop +1

set @y=@y+1

set @z = CONVERT(NVARCHAR,@y)

end

if len(@sql) > 70

begin

set @sql = substring(@sql,1,len(@sql)-9)

execute(@sql)

--print @sql

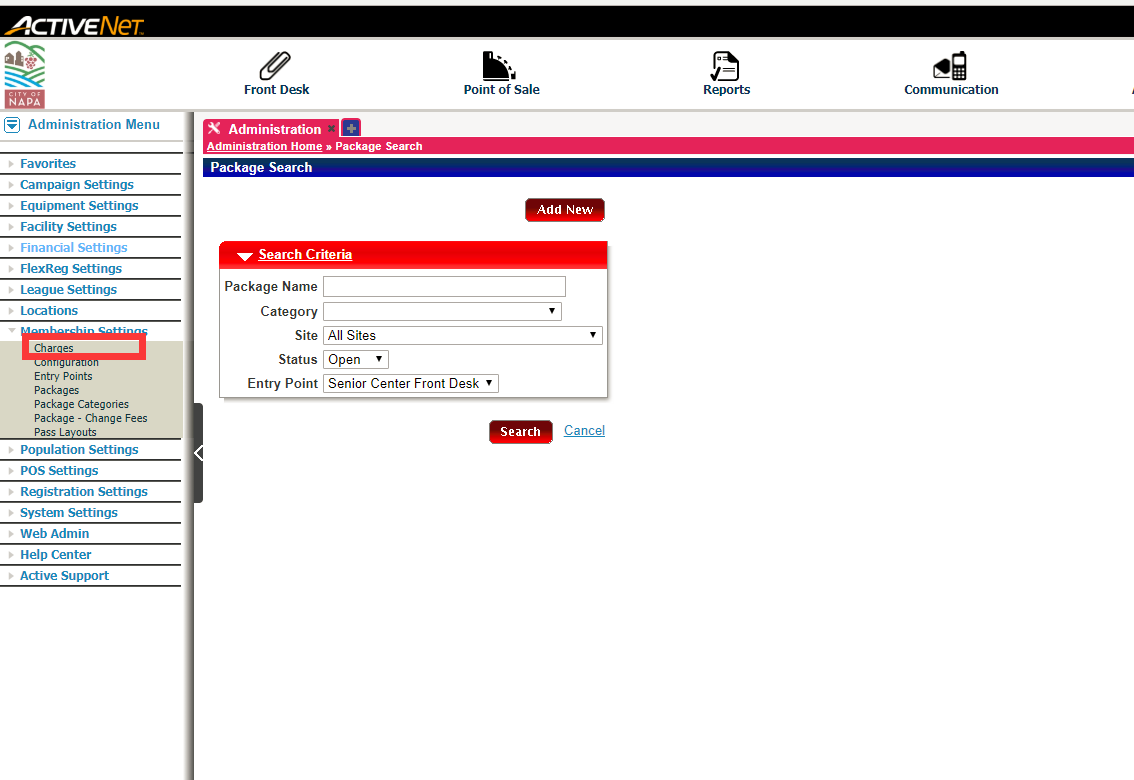
end

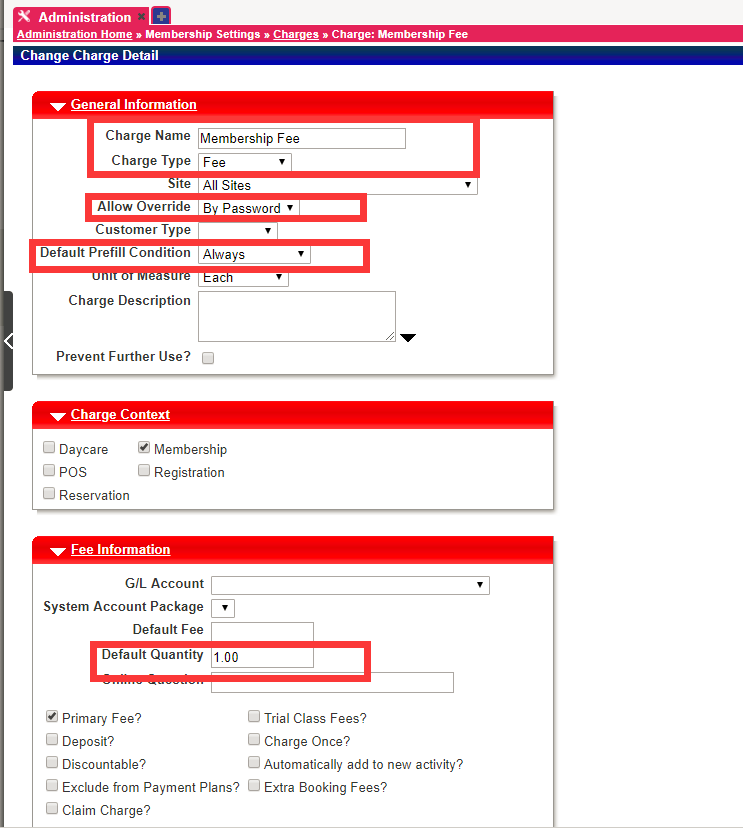
print @y

# Add Membership:

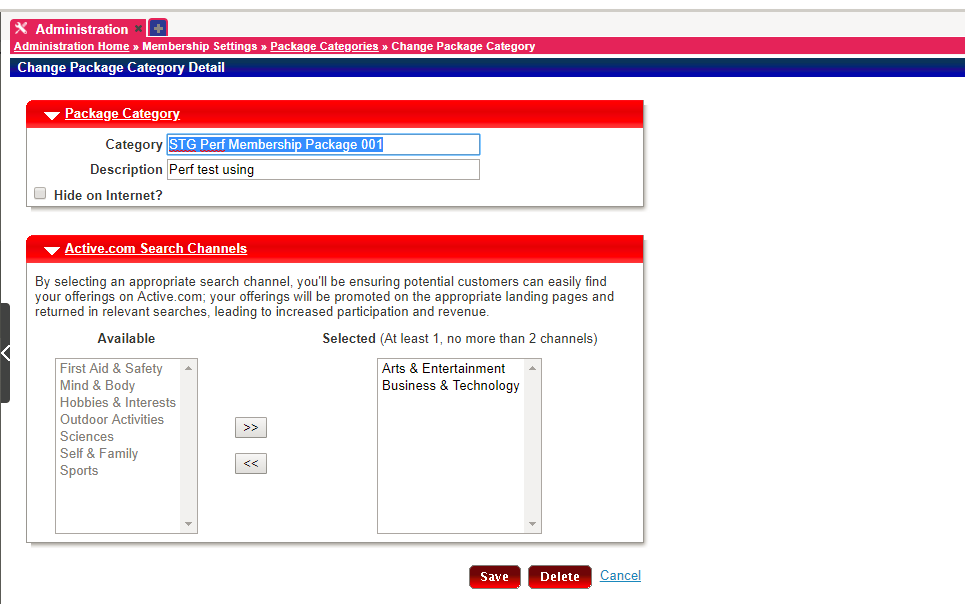
select PACKAGE\_ID from packages where PACKAGENAME like '%STG%'

Add membership charge:

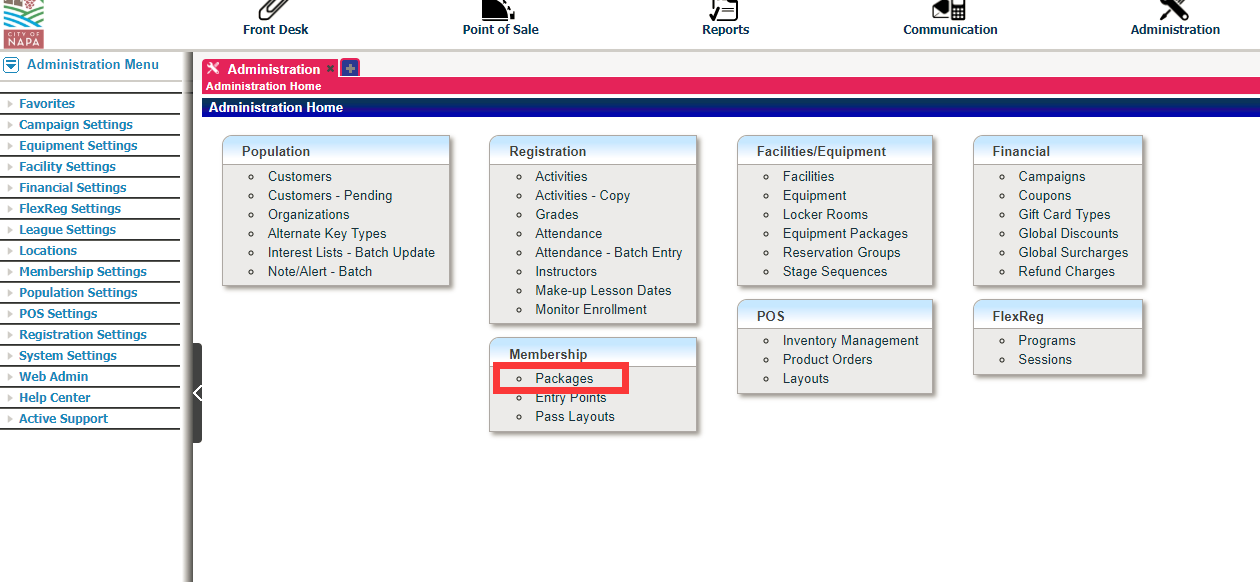


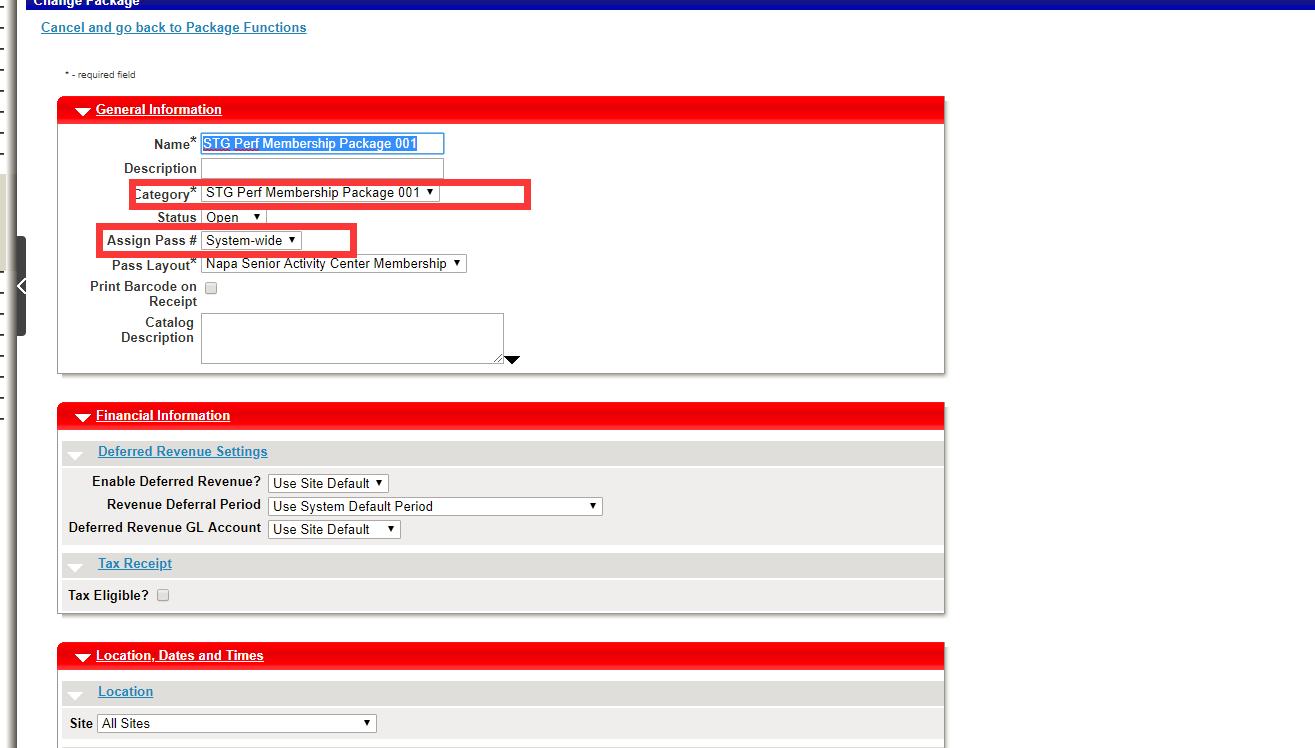


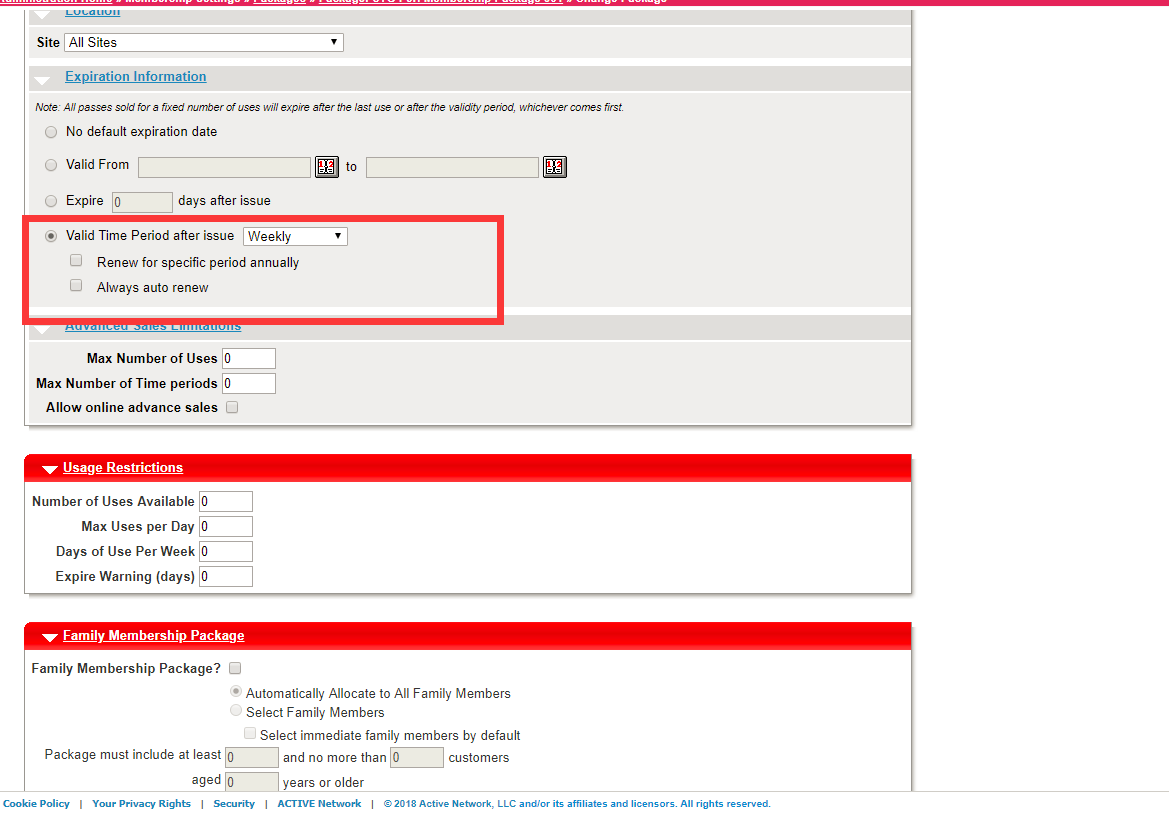
Add membership category



Add package

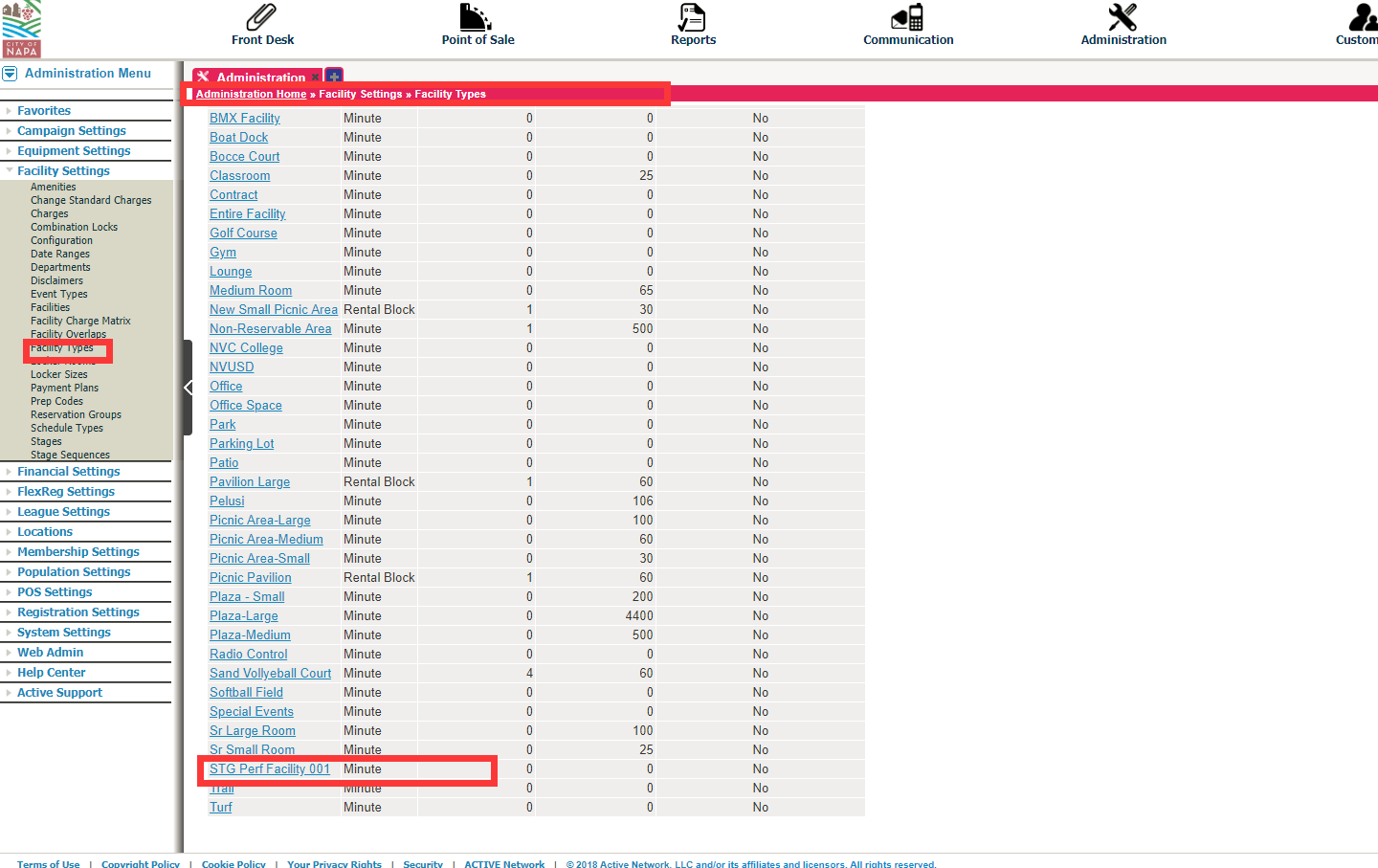




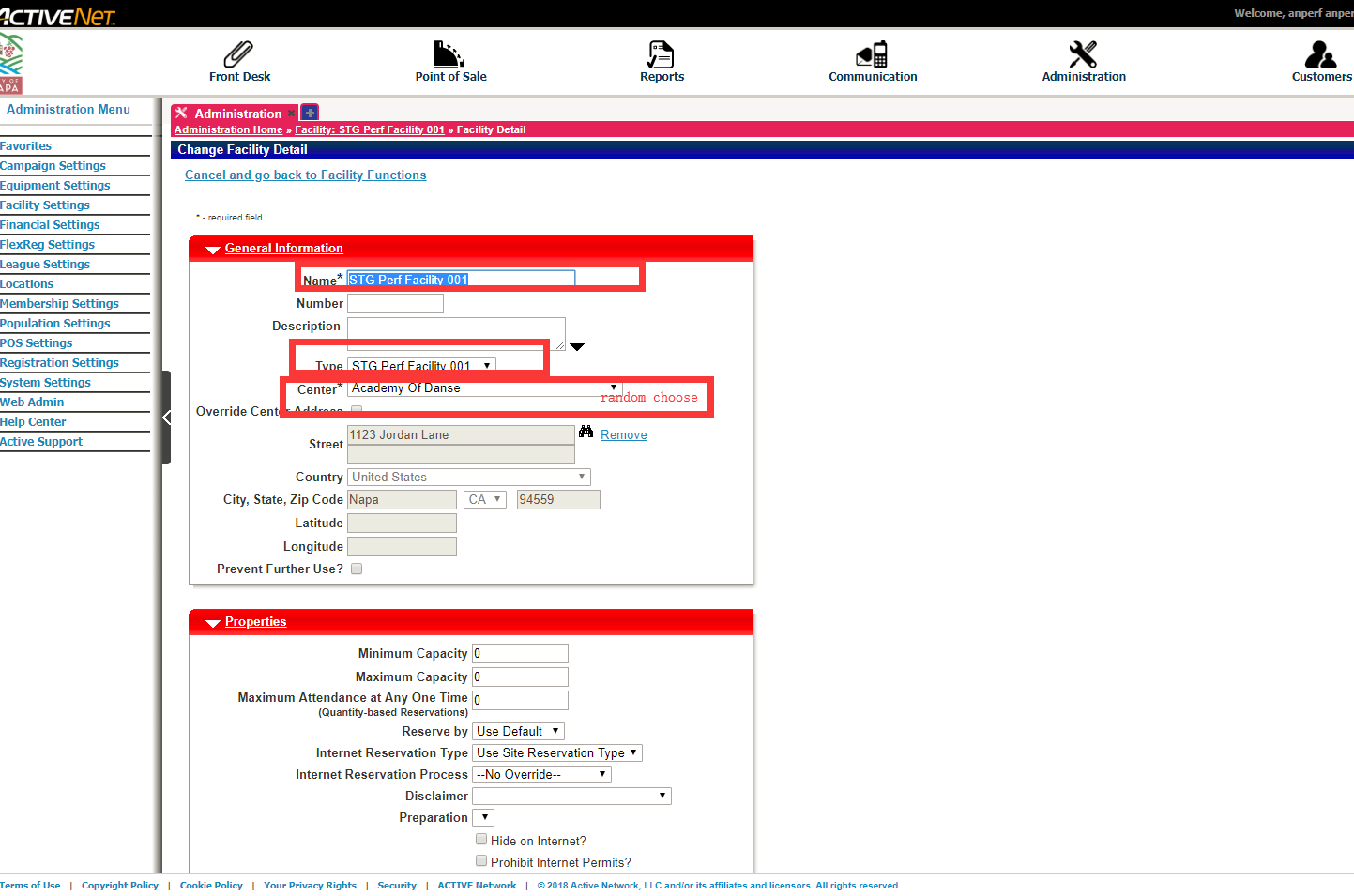


# Add Facility

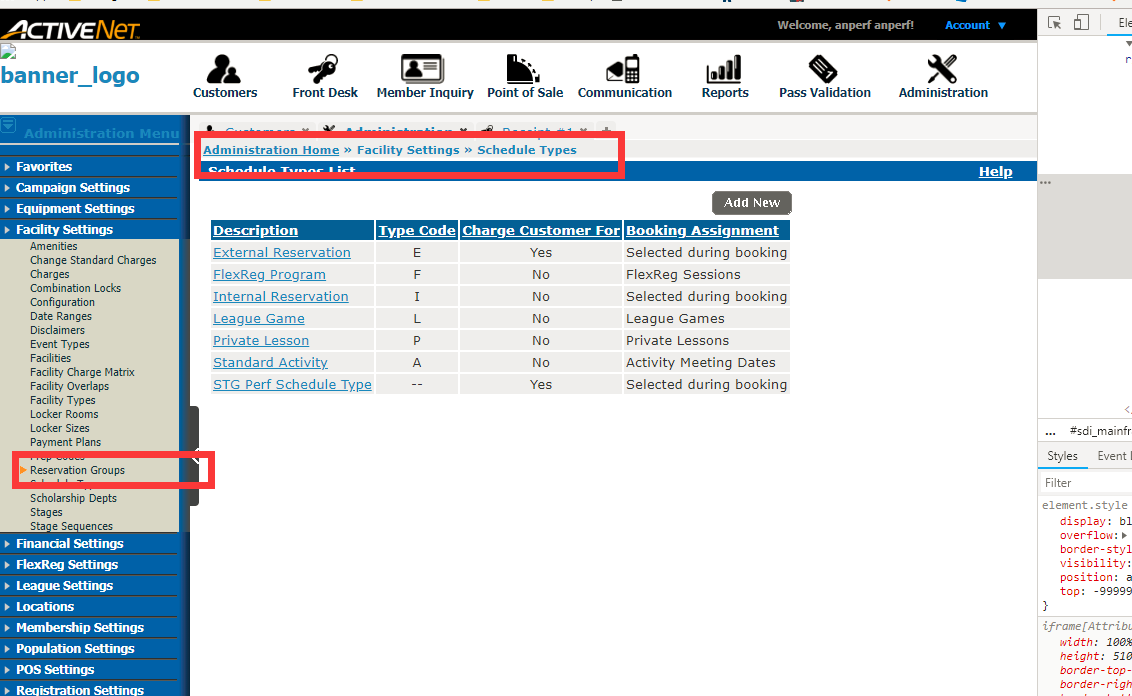
Add Facility type



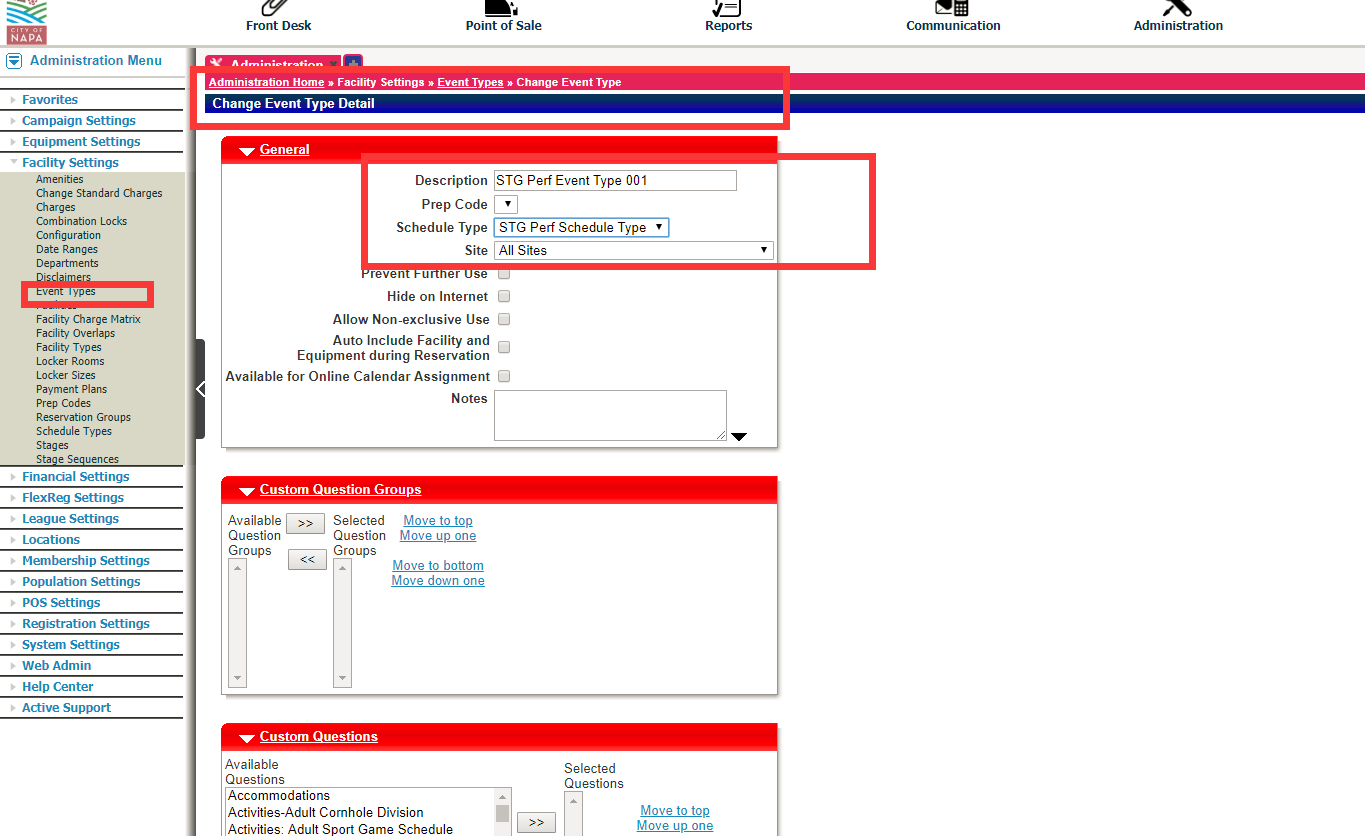
Add Facility

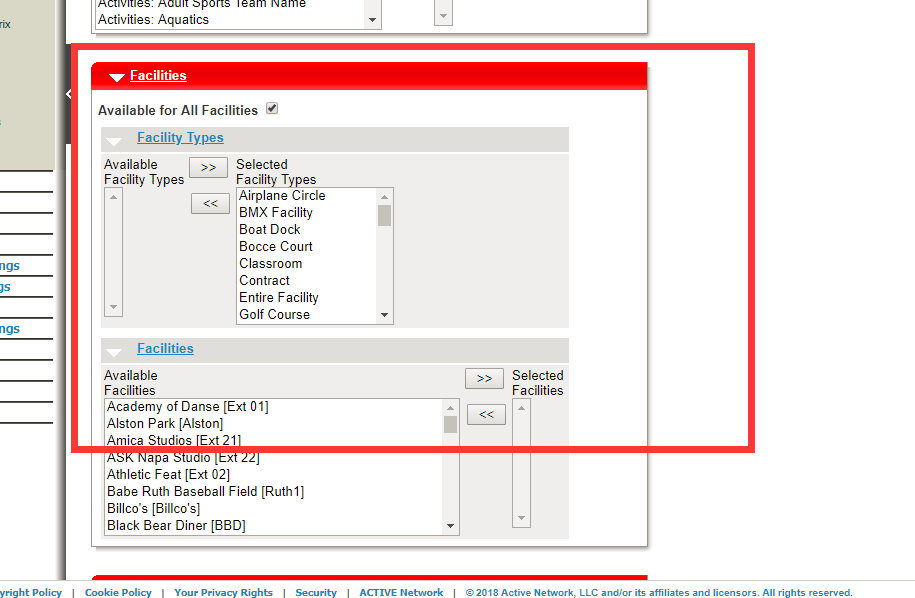


Add Schedule type



Add EventType

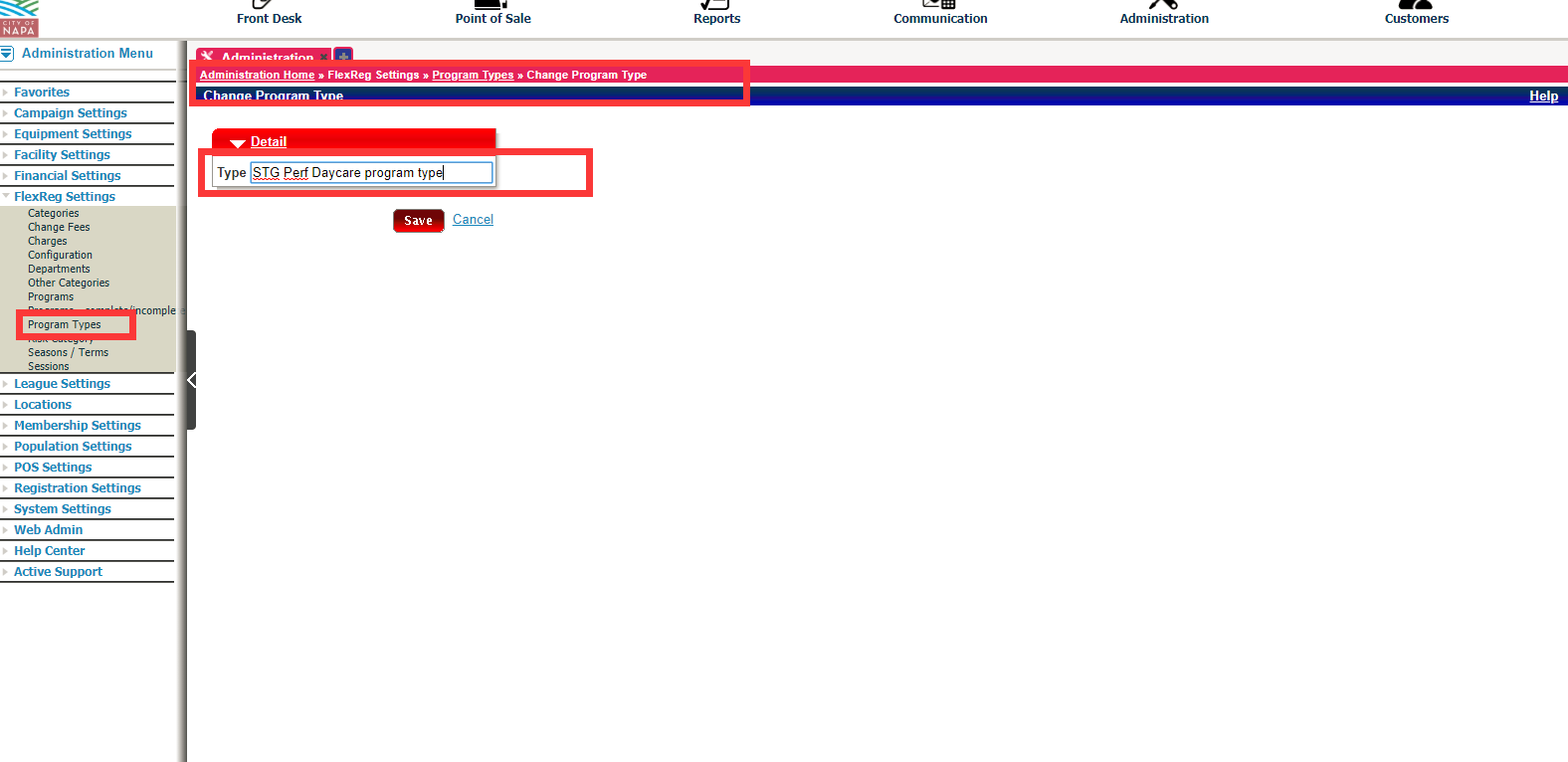




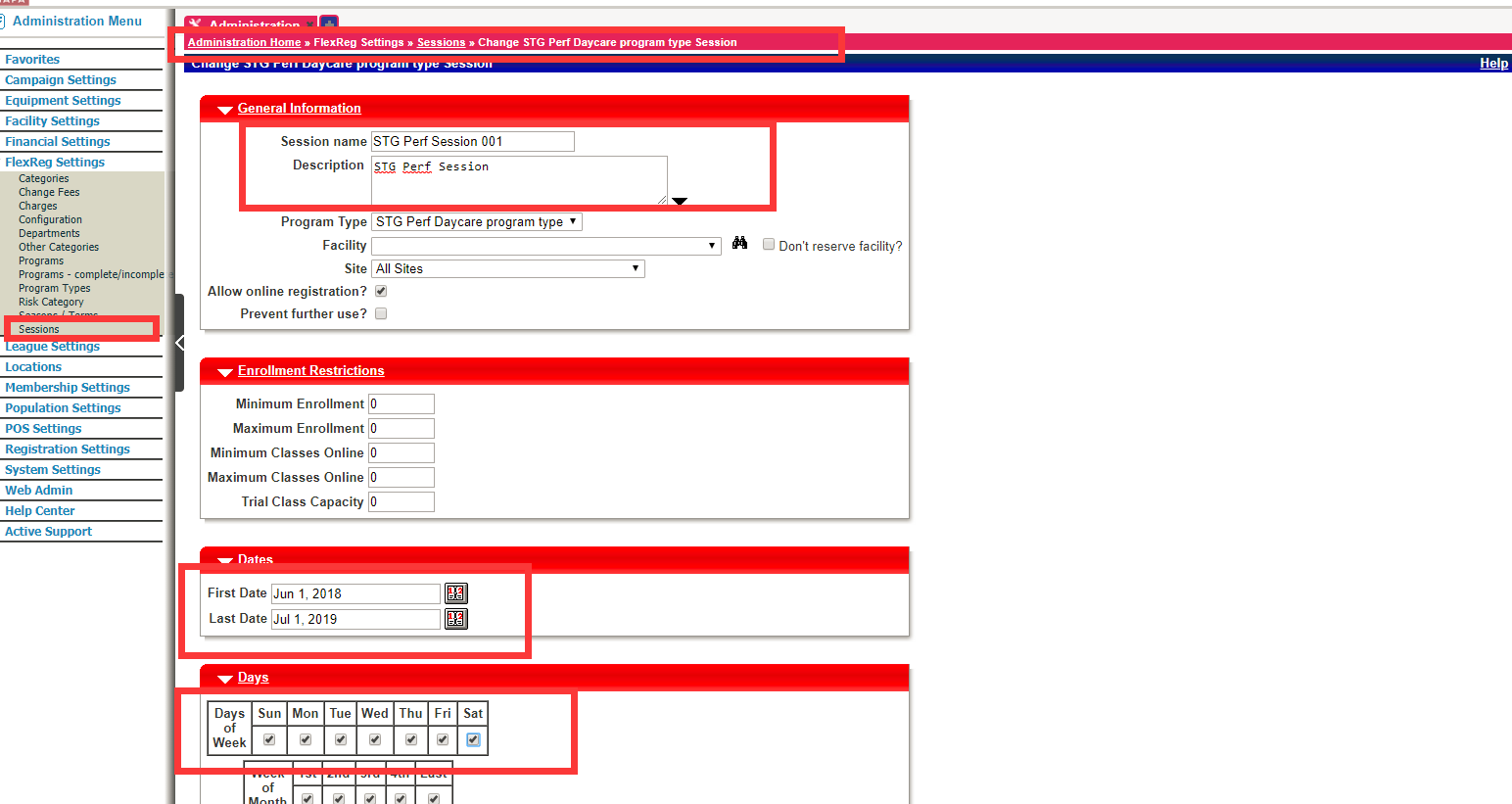
# Add Daycare

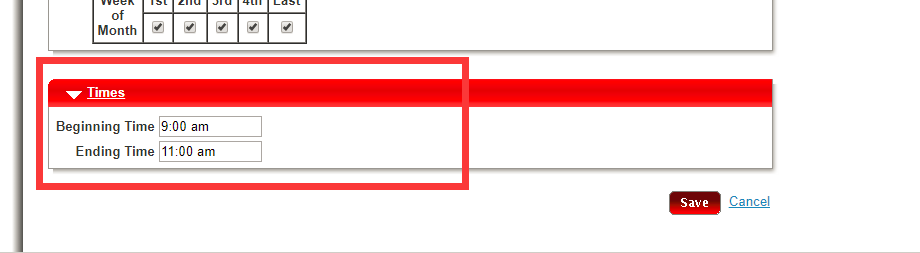
select DCPROGRAM\_ID,programtype\_id from DCPROGRAMS where DCPROGRAMNAME like '%STG%'

Add Daycareprogrem type

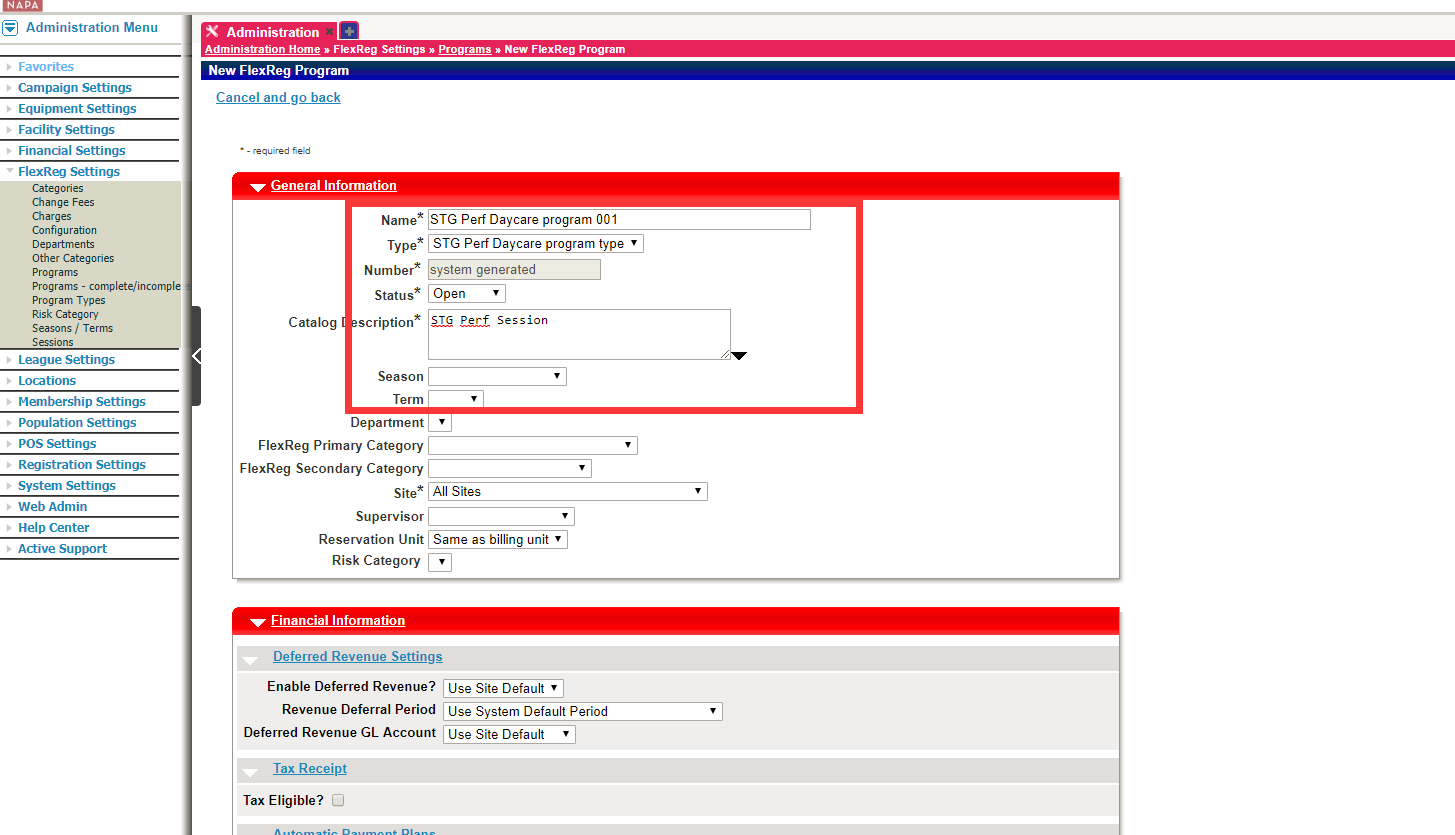


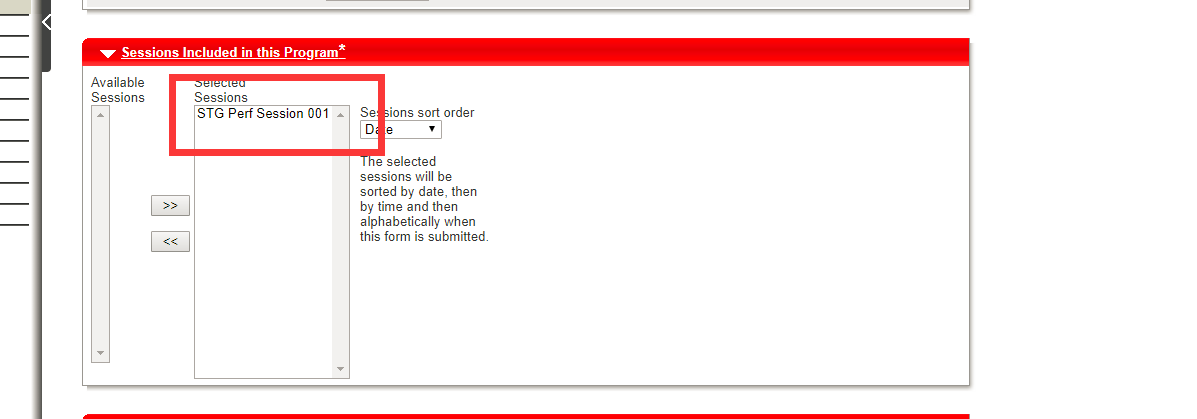
Add Session

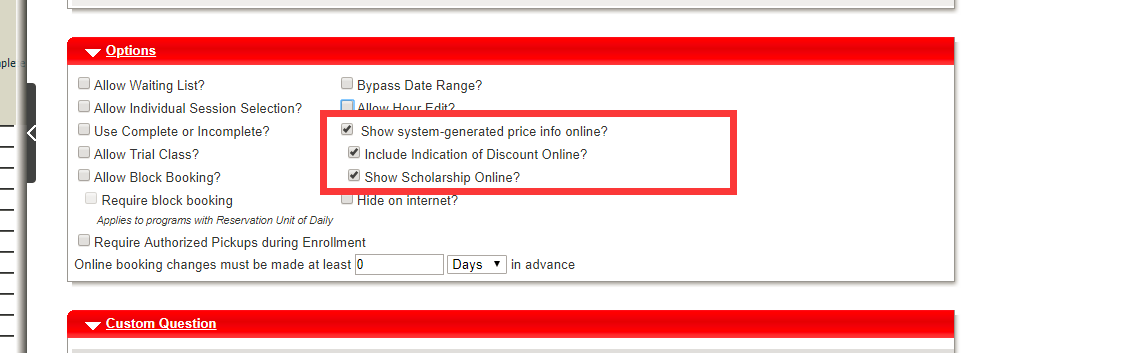




Add DaycareProgram







Add a new fee for this program

