In [1]:
```
import cv2
import numpy as np
import os
from random import shuffle
from tqdm import tqdm

TRAIN_DIR = 'C:/Users/upsto/Downloads/cats v dogs/train7/train/'
TEST_DIR = 'C:/Users/upsto/Downloads/cats v dogs/test7/test/'
IMG_SIZE = 50
LR = 1e-3

MODEL_NAME = 'dogsvscats-{}-{}.model'.format(LR, '2conv-basic')
```

In [2]:
```
def label_img(img):
    word_label = img.split('.')[-3]

    if word_label == 'cat': return [1,0]

    elif word_label == 'dog': return [0,1]
```

In [3]:
```
def create_train_data():
    training_data = []
    for img in tqdm(os.listdir(TRAIN_DIR)):
        label = label_img(img)
        path = os.path.join(TRAIN_DIR,img)
        img = cv2.imread(path,cv2.IMREAD_GRAYSCALE)
        img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))
        training_data.append([np.array(img),np.array(label)])
    shuffle(training_data)
    np.save('train_data.npy', training_data)
    return training_data
```

In [4]:
```
def process_test_data():
    testing_data = []
    for img in tqdm(os.listdir(TEST_DIR)):
        path = os.path.join(TEST_DIR,img)
        img_num = img.split('.')[0]
        img = cv2.imread(path,cv2.IMREAD_GRAYSCALE)
        img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))
        testing_data.append([np.array(img), img_num])

    shuffle(testing_data)
    np.save('test_data.npy', testing_data)
    return testing_data
```

In [5]:
```
train_data = create_train_data()
```

```
100%|██████████| 25000/25000 [00:18<00:00, 1350.07it/s]
```

In [6]:
```
import tflearn
from tflearn.layers.conv import conv_2d, max_pool_2d
from tflearn.layers.core import input_data, dropout, fully_connected
from tflearn.layers.estimator import regression

convnet = input_data(shape=[None, IMG_SIZE, IMG_SIZE, 1], name='input')

convnet = conv_2d(convnet, 32, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)

convnet = conv_2d(convnet, 64, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)

convnet = fully_connected(convnet, 1024, activation='relu')
convnet = dropout(convnet, 0.8)

convnet = fully_connected(convnet, 2, activation='softmax')
convnet = regression(convnet, optimizer='adam', learning_rate=LR, loss='categorical_crossentropy', name='targets')

model = tflearn.DNN(convnet, tensorboard_dir='log')
```

```
curses is not supported on this machine (please install/reinstall curses for
an optimal experience)
WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\he
lpers\summarizer.py:9: The name tf.summary.merge is deprecated. Please use t
f.compat.v1.summary.merge instead.

WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\he
lpers\trainer.py:25: The name tf.summary.FileWriter is deprecated. Please use
tf.compat.v1.summary.FileWriter instead.

WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\co
llections.py:13: The name tf.GraphKeys is deprecated. Please use tf.compat.v
1.GraphKeys instead.

WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\co
nfig.py:123: The name tf.get_collection is deprecated. Please use tf.compat.v
1.get_collection instead.

WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\co
nfig.py:129: The name tf.add_to_collection is deprecated. Please use tf.compa
t.v1.add_to_collection instead.

WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\co
nfig.py:131: The name tf.assign is deprecated. Please use tf.compat.v1.assign
instead.

WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\la
yers\core.py:81: The name tf.placeholder is deprecated. Please use tf.compat.
v1.placeholder instead.

WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\la
yers\conv.py:73: The name tf.variable_scope is deprecated. Please use tf.comp
at.v1.variable_scope instead.

WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\in
itializations.py:119: calling UniformUnitScaling.__init__ (from tensorflow.py
thon.ops.init_ops) with dtype is deprecated and will be removed in a future v
ersion.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it to th
e constructor
WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tensorflow
_core\python\util\deprecation.py:507: UniformUnitScaling.__init__ (from tenso
rflow.python.ops.init_ops) is deprecated and will be removed in a future vers
ion.
Instructions for updating:
Use tf.initializers.variance_scaling instead with distribution=uniform to get
equivalent behavior.
WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\la
yers\conv.py:552: The name tf.nn.max_pool is deprecated. Please use tf.nn.max
_pool2d instead.

WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\in
itializations.py:174: calling TruncatedNormal.__init__ (from tensorflow.pytho
n.ops.init_ops) with dtype is deprecated and will be removed in a future vers
ion.
Instructions for updating:
```

```
Call initializer instance with the dtype argument instead of passing it to th
e constructor
WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\la
yers\core.py:239: calling dropout (from tensorflow.python.ops.nn_ops) with ke
ep_prob is deprecated and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - k
eep_prob`.
WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\op
timizers.py:238: The name tf.train.AdamOptimizer is deprecated. Please use t
f.compat.v1.train.AdamOptimizer instead.

WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\ob
jectives.py:66: calling reduce_sum_v1 (from tensorflow.python.ops.math_ops) w
ith keep_dims is deprecated and will be removed in a future version.
Instructions for updating:
keep_dims is deprecated, use keepdims instead
WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\ob
jectives.py:70: The name tf.log is deprecated. Please use tf.math.log instea
d.

WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\la
yers\estimator.py:189: The name tf.trainable_variables is deprecated. Please
use tf.compat.v1.trainable_variables instead.

WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\he
lpers\trainer.py:571: The name tf.get_default_graph is deprecated. Please use
tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\he
lpers\trainer.py:115: The name tf.Session is deprecated. Please use tf.compa
t.v1.Session instead.

WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\su
mmaries.py:46: The name tf.summary.scalar is deprecated. Please use tf.compa
t.v1.summary.scalar instead.

WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tensorflow
_core\python\ops\math_grad.py:1424: where (from tensorflow.python.ops.array_o
ps) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\he
lpers\trainer.py:134: The name tf.train.Saver is deprecated. Please use tf.co
mpat.v1.train.Saver instead.

WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\he
lpers\trainer.py:164: The name tf.global_variables_initializer is deprecated.
Please use tf.compat.v1.global_variables_initializer instead.

WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\he
lpers\trainer.py:165: The name tf.local_variables_initializer is deprecated.
Please use tf.compat.v1.local_variables_initializer instead.

WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\he
lpers\trainer.py:166: The name tf.variables_initializer is deprecated. Please
use tf.compat.v1.variables_initializer instead.
```

```
WARNING:tensorflow:From C:\Users\upsto\Anaconda3\lib\site-packages\tflearn\he
lpers\trainer.py:167: The name tf.get_collection_ref is deprecated. Please us
e tf.compat.v1.get_collection_ref instead.
```

In [7]:
```
train = train_data[:-500]
test = train_data[-500:]
```

In [8]:
```
X = np.array([i[0] for i in train]).reshape(-1,IMG_SIZE,IMG_SIZE,1)
Y = [i[1] for i in train]

test_x = np.array([i[0] for i in test]).reshape(-1,IMG_SIZE,IMG_SIZE,1)
test_y = [i[1] for i in test]
```

Model 1

In [11]:
```
import datetime
import tensorflow as tf
tf.reset_default_graph()
```

In [13]:
```
start=datetime.datetime.now()
tf.reset_default_graph()
convnet = input_data(shape=[None, IMG_SIZE, IMG_SIZE, 1], name='input')
convnet = conv_2d(convnet, 32, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = conv_2d(convnet, 32, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = conv_2d(convnet, 64, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = conv_2d(convnet, 64, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = conv_2d(convnet, 128, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = conv_2d(convnet, 128, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = fully_connected(convnet, 1024, activation='relu')
convnet = dropout(convnet, 0.8)
convnet = fully_connected(convnet, 2, activation='softmax')
convnet = regression(convnet, optimizer='SGD', learning_rate=LR, loss='categor
ical_crossentropy', name='targets')
model = tflearn.DNN(convnet, tensorboard_dir='log', tensorboard_verbose=0)
model.fit({'input': X}, {'targets': Y}, n_epoch=10,
          validation_set=({'input': test_x}, {'targets': test_y}),
          snapshot_step=500, show_metric=True, run_id=MODEL_NAME)

end=datetime.datetime.now()

print(end-start)
```

```
Training Step: 3829  | total loss: 0.69075 | time: 16.199s
| SGD | epoch: 010 | loss: 0.69075 - acc: 0.5695 -- iter: 24448/24500
Training Step: 3830  | total loss: 0.69099 | time: 17.243s
| SGD | epoch: 010 | loss: 0.69099 - acc: 0.5626 | val_loss: 0.69048 - val_ac
c: 0.5740 -- iter: 24500/24500
--
0:02:57.019551
```

In [14]:
```
import matplotlib.pyplot as plt


test_data = process_test_data()


fig=plt.figure()

for num,data in enumerate(test_data[:12]):
    # cat: [1,0]
    # dog: [0,1]

    img_num = data[1]
    img_data = data[0]

    y = fig.add_subplot(3,4,num+1)
    orig = img_data
    data = img_data.reshape(IMG_SIZE,IMG_SIZE,1)
    #model_out = model.predict([data])[0]
    model_out = model.predict([data])[0]

    if np.argmax(model_out) == 1: str_label='Dog'
    else: str_label='Cat'

    y.imshow(orig,cmap='gray')
    plt.title(str_label)
    y.axes.get_xaxis().set_visible(False)
    y.axes.get_yaxis().set_visible(False)
plt.show()
```

```
100%|██████████| 12500/12500 [00:09<00:00, 1330.47it/s]
```



```
<Figure size 640x480 with 12 Axes>
```

In [15]:
```
with open('submission_file.csv','w') as f:
    f.write('id,label\n')

with open('submission_file.csv','a') as f:
    for data in tqdm(test_data):
        img_num = data[1]
        img_data = data[0]
        orig = img_data
        data = img_data.reshape(IMG_SIZE,IMG_SIZE,1)
        model_out = model.predict([data])[0]
        f.write('{},{}\n'.format(img_num,model_out[1]))
```

```
100%|██████████| 12500/12500 [00:13<00:00, 893.30it/s]
```

Model 2

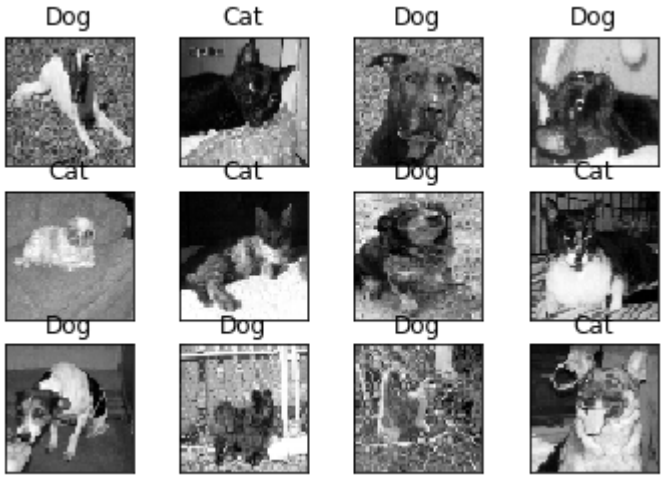
In [85]:
```
tf.reset_default_graph()
```

In [16]:
```
start=datetime.datetime.now()
tf.reset_default_graph()
convnet = input_data(shape=[None, IMG_SIZE, IMG_SIZE, 1], name='input')
convnet = conv_2d(convnet, 32, 10, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = conv_2d(convnet, 64, 10, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = conv_2d(convnet, 128, 10, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = conv_2d(convnet, 64, 10, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = conv_2d(convnet, 32, 10, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = fully_connected(convnet, 1024, activation='relu')
convnet = dropout(convnet, 0.8)
convnet = fully_connected(convnet, 2, activation='softmax')
convnet = regression(convnet, optimizer='SGD', learning_rate=LR, loss='categor
ical_crossentropy', name='targets')
model = tflearn.DNN(convnet, tensorboard_dir='log', tensorboard_verbose=0)
model.fit({'input': X}, {'targets': Y}, n_epoch=10,
          validation_set=({'input': test_x}, {'targets': test_y}),
          snapshot_step=500, show_metric=True, run_id=MODEL_NAME)
end=datetime.datetime.now()

print(end-start)
```

```
Training Step: 3829  | total loss: 0.69180 | time: 64.139s
| SGD | epoch: 010 | loss: 0.69180 - acc: 0.6023 -- iter: 24448/24500
Training Step: 3830  | total loss: 0.69177 | time: 65.307s
| SGD | epoch: 010 | loss: 0.69177 - acc: 0.6046 | val_loss: 0.69178 - val_ac
c: 0.5960 -- iter: 24500/24500
--
0:10:47.153636
```

In [17]:
```
import matplotlib.pyplot as plt


test_data = process_test_data()

fig=plt.figure()

for num,data in enumerate(test_data[:12]):


    img_num = data[1]
    img_data = data[0]

    y = fig.add_subplot(3,4,num+1)
    orig = img_data
    data = img_data.reshape(IMG_SIZE,IMG_SIZE,1)

    model_out = model.predict([data])[0]

    if np.argmax(model_out) == 1: str_label='Dog'
    else: str_label='Cat'

    y.imshow(orig,cmap='gray')
    plt.title(str_label)
    y.axes.get_xaxis().set_visible(False)
    y.axes.get_yaxis().set_visible(False)
plt.show()
```

```
100%|██████████| 12500/12500 [00:09<00:00, 1346.72it/s]
```



In [18]:
```
with open('submission_file2.csv','w') as f:
    f.write('id,label\n')

with open('submission_file2.csv','a') as f:
    for data in tqdm(test_data):
        img_num = data[1]
        img_data = data[0]
        orig = img_data
        data = img_data.reshape(IMG_SIZE,IMG_SIZE,1)
        model_out = model.predict([data])[0]
        f.write('{},{}\n'.format(img_num,model_out[1]))
```

```
100%|██████████| 12500/12500 [00:22<00:00, 552.60it/s]
```

Model 3

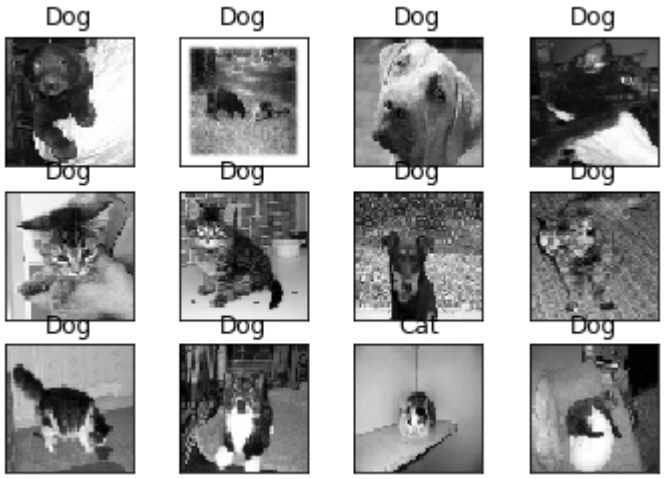
In [19]:
```
tf.reset_default_graph()
```

In [20]:
```
start=datetime.datetime.now()
tf.reset_default_graph()
convnet = input_data(shape=[None, IMG_SIZE, IMG_SIZE, 1], name='input')
convnet = conv_2d(convnet, 32, 10, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = conv_2d(convnet, 32, 10, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = conv_2d(convnet, 64, 10, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = conv_2d(convnet, 64, 10, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = conv_2d(convnet, 128, 10, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = conv_2d(convnet, 128, 10, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = fully_connected(convnet, 1024, activation='relu')
convnet = dropout(convnet, 0.8)
convnet = fully_connected(convnet, 2, activation='softmax')
convnet = regression(convnet, optimizer='SGD', learning_rate=LR, loss='categor
ical_crossentropy', name='targets')
model = tflearn.DNN(convnet, tensorboard_dir='log', tensorboard_verbose=0)
model.fit({'input': X}, {'targets': Y}, n_epoch=10,
          validation_set=({'input': test_x}, {'targets': test_y}),
          snapshot_step=500, show_metric=True, run_id=MODEL_NAME)
end=datetime.datetime.now()

print(end-start)
```

```
Training Step: 3829  | total loss: 0.69313 | time: 101.062s
| SGD | epoch: 010 | loss: 0.69313 - acc: 0.4832 -- iter: 24448/24500
Training Step: 3830  | total loss: 0.69313 | time: 102.317s
| SGD | epoch: 010 | loss: 0.69313 - acc: 0.4896 | val_loss: 0.69311 - val_ac
c: 0.5640 -- iter: 24500/24500
--
0:17:21.502748
```

In [21]:
```
import matplotlib.pyplot as plt


test_data = process_test_data()


fig=plt.figure()

for num,data in enumerate(test_data[:12]):


    img_num = data[1]
    img_data = data[0]

    y = fig.add_subplot(3,4,num+1)
    orig = img_data
    data = img_data.reshape(IMG_SIZE,IMG_SIZE,1)

    model_out = model.predict([data])[0]

    if np.argmax(model_out) == 1: str_label='Dog'
    else: str_label='Cat'

    y.imshow(orig,cmap='gray')
    plt.title(str_label)
    y.axes.get_xaxis().set_visible(False)
    y.axes.get_yaxis().set_visible(False)
plt.show()
```

```
100%|██████████| 12500/12500 [00:09<00:00, 1333.95it/s]
```



In [22]:
```
with open('submission_file3.csv','w') as f:
    f.write('id,label\n')

with open('submission_file3.csv','a') as f:
    for data in tqdm(test_data):
        img_num = data[1]
        img_data = data[0]
        orig = img_data
        data = img_data.reshape(IMG_SIZE,IMG_SIZE,1)
        model_out = model.predict([data])[0]
        f.write('{},{}\n'.format(img_num,model_out[1]))
```

```
100%|██████████| 12500/12500 [00:33<00:00, 374.05it/s]
```

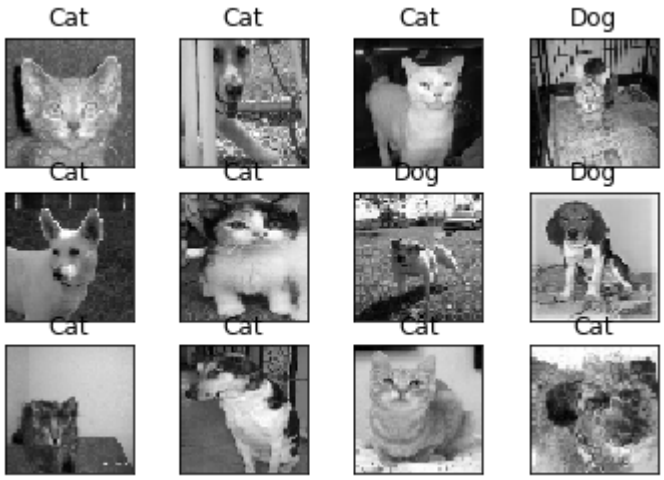
Model 4

In [23]:
```
start=datetime.datetime.now()
tf.reset_default_graph()
convnet = input_data(shape=[None, IMG_SIZE, IMG_SIZE, 1], name='input')
convnet = conv_2d(convnet, 32, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = conv_2d(convnet, 64, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = conv_2d(convnet, 128, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = fully_connected(convnet, 1024, activation='relu')
convnet = dropout(convnet, 0.8)
convnet = fully_connected(convnet, 2, activation='softmax')
convnet = regression(convnet, optimizer='SGD', learning_rate=LR, loss='categor
ical_crossentropy', name='targets')
model = tflearn.DNN(convnet, tensorboard_dir='log', tensorboard_verbose=0)
model.fit({'input': X}, {'targets': Y}, n_epoch=10,
          validation_set=({'input': test_x}, {'targets': test_y}),
          snapshot_step=500, show_metric=True, run_id=MODEL_NAME)
end=datetime.datetime.now()

print(end-start)
```

```
Training Step: 3829  | total loss: 0.57630 | time: 15.833s
| SGD | epoch: 010 | loss: 0.57630 - acc: 0.6933 -- iter: 24448/24500
Training Step: 3830  | total loss: 0.57583 | time: 16.874s
| SGD | epoch: 010 | loss: 0.57583 - acc: 0.6989 | val_loss: 0.59731 - val_ac
c: 0.6800 -- iter: 24500/24500
--
0:02:50.729635
```

In [24]:
```
import matplotlib.pyplot as plt


test_data = process_test_data()


fig=plt.figure()

for num,data in enumerate(test_data[:12]):


    img_num = data[1]
    img_data = data[0]

    y = fig.add_subplot(3,4,num+1)
    orig = img_data
    data = img_data.reshape(IMG_SIZE,IMG_SIZE,1)

    model_out = model.predict([data])[0]

    if np.argmax(model_out) == 1: str_label='Dog'
    else: str_label='Cat'

    y.imshow(orig,cmap='gray')
    plt.title(str_label)
    y.axes.get_xaxis().set_visible(False)
    y.axes.get_yaxis().set_visible(False)
plt.show()
```

```
100%|██████████| 12500/12500 [00:09<00:00, 1336.60it/s]
```



In [25]:
```
with open('submission_file4.csv','w') as f:
    f.write('id,label\n')

with open('submission_file4.csv','a') as f:
    for data in tqdm(test_data):
        img_num = data[1]
        img_data = data[0]
        orig = img_data
        data = img_data.reshape(IMG_SIZE,IMG_SIZE,1)
        model_out = model.predict([data])[0]
        f.write('{},{}\n'.format(img_num,model_out[1]))
```

```
100%|██████████| 12500/12500 [00:09<00:00, 1284.27it/s]
```

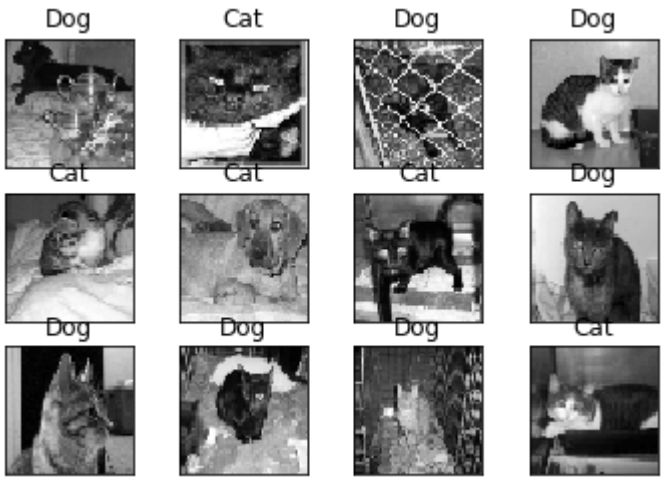
Model 5

In [26]:
```
start=datetime.datetime.now()
tf.reset_default_graph()
convnet = input_data(shape=[None, IMG_SIZE, IMG_SIZE, 1], name='input')
convnet = conv_2d(convnet, 32, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = conv_2d(convnet, 64, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = conv_2d(convnet, 64, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)
convnet = fully_connected(convnet, 1024, activation='relu')
convnet = dropout(convnet, 0.8)
convnet = fully_connected(convnet, 2, activation='softmax')
convnet = regression(convnet, optimizer='SGD', learning_rate=LR, loss='categor
ical_crossentropy', name='targets')
model = tflearn.DNN(convnet, tensorboard_dir='log', tensorboard_verbose=0)
model.fit({'input': X}, {'targets': Y}, n_epoch=10,
          validation_set=({'input': test_x}, {'targets': test_y}),
          snapshot_step=500, show_metric=True, run_id=MODEL_NAME)
end=datetime.datetime.now()

print(end-start)
```

```
Training Step: 3829  | total loss: 0.57180 | time: 15.625s
| SGD | epoch: 010 | loss: 0.57180 - acc: 0.7122 -- iter: 24448/24500
Training Step: 3830  | total loss: 0.57106 | time: 16.667s
| SGD | epoch: 010 | loss: 0.57106 - acc: 0.7128 | val_loss: 0.56857 - val_ac
c: 0.6920 -- iter: 24500/24500
--
0:02:50.744902
```

In [27]:
```
import matplotlib.pyplot as plt


test_data = process_test_data()


fig=plt.figure()

for num,data in enumerate(test_data[:12]):


    img_num = data[1]
    img_data = data[0]

    y = fig.add_subplot(3,4,num+1)
    orig = img_data
    data = img_data.reshape(IMG_SIZE,IMG_SIZE,1)

    model_out = model.predict([data])[0]

    if np.argmax(model_out) == 1: str_label='Dog'
    else: str_label='Cat'

    y.imshow(orig,cmap='gray')
    plt.title(str_label)
    y.axes.get_xaxis().set_visible(False)
    y.axes.get_yaxis().set_visible(False)
plt.show()
```

```
100%|██████████| 12500/12500 [00:09<00:00, 1335.70it/s]
```



In [28]:
```
with open('submission_file5.csv','w') as f:
    f.write('id,label\n')

with open('submission_file5.csv','a') as f:
    for data in tqdm(test_data):
        img_num = data[1]
        img_data = data[0]
        orig = img_data
        data = img_data.reshape(IMG_SIZE,IMG_SIZE,1)
        model_out = model.predict([data])[0]
        f.write('{},{}\n'.format(img_num,model_out[1]))
```

```
100%|██████████| 12500/12500 [00:09<00:00, 1309.28it/s]
```

In [ ]:
```

```