```
                if(nodbh.flag==T) points(A,O,cex=dbh.exp,col=col,pch=17,lwd=1)
                if(dbh.exp<1) pos<-1 else pos<-dbh.exp
                text(A+pos,O+pos,data$PTN[i],cex=.5)
        }
}
#############################################################################################################
# Define top level mapping function
tree.map.auto2<-function(data,data.p,file,size.large=7,size.small=100,xlim=NULL,ylim=NULL){
        pdf(file=file,width=6.7)

        tree2014.detail<-list()

        for(i in 1:length(unique(data$PlotNumber))){

                data.t<-data[data$PlotNumber==sort(unique(data$PlotNumber))[i],]
                plot<-subset(plot2014, plot2014$PLOT_NUM==data.t$PlotNumber[1] & plot2014$COND_CLASS == 1)

                data.t$NORTHING<-plot$NORTHING[1]
                data.t$EASTING<-plot$EASTING[1]

                main<-paste("Plot",data.t$PlotNumber[1])
                sub<-paste("TPA = ", plot$TPA[1], ", BA = ", format(plot$Plot_BA[1], digits=5), ", CFVOL = ",
format(plot$CFVolAcre[1], digits=6), ", Biomass (tons) = ", format(plot$Biomass[1], digits=4))

        tree2014.detail[[i]]<-tree.map2(data=data.t,size.large=7,size.small=100,xlim=xlim,ylim=ylim,main=main,
        sub=sub)
}

 tree2014.update<-rbind.fill(tree2014.detail)
 write.csv(tree2014.update,file = "../FIA_RapidForestInventory/tree2014_UTM.csv")
        dev.off()
}

# Call top level mapping function with necessary inputs
tree.map.auto2(cfc.2014.clean, plot2014 ,file="../FIA_RapidForestInventory/trees2014.pdf")
```