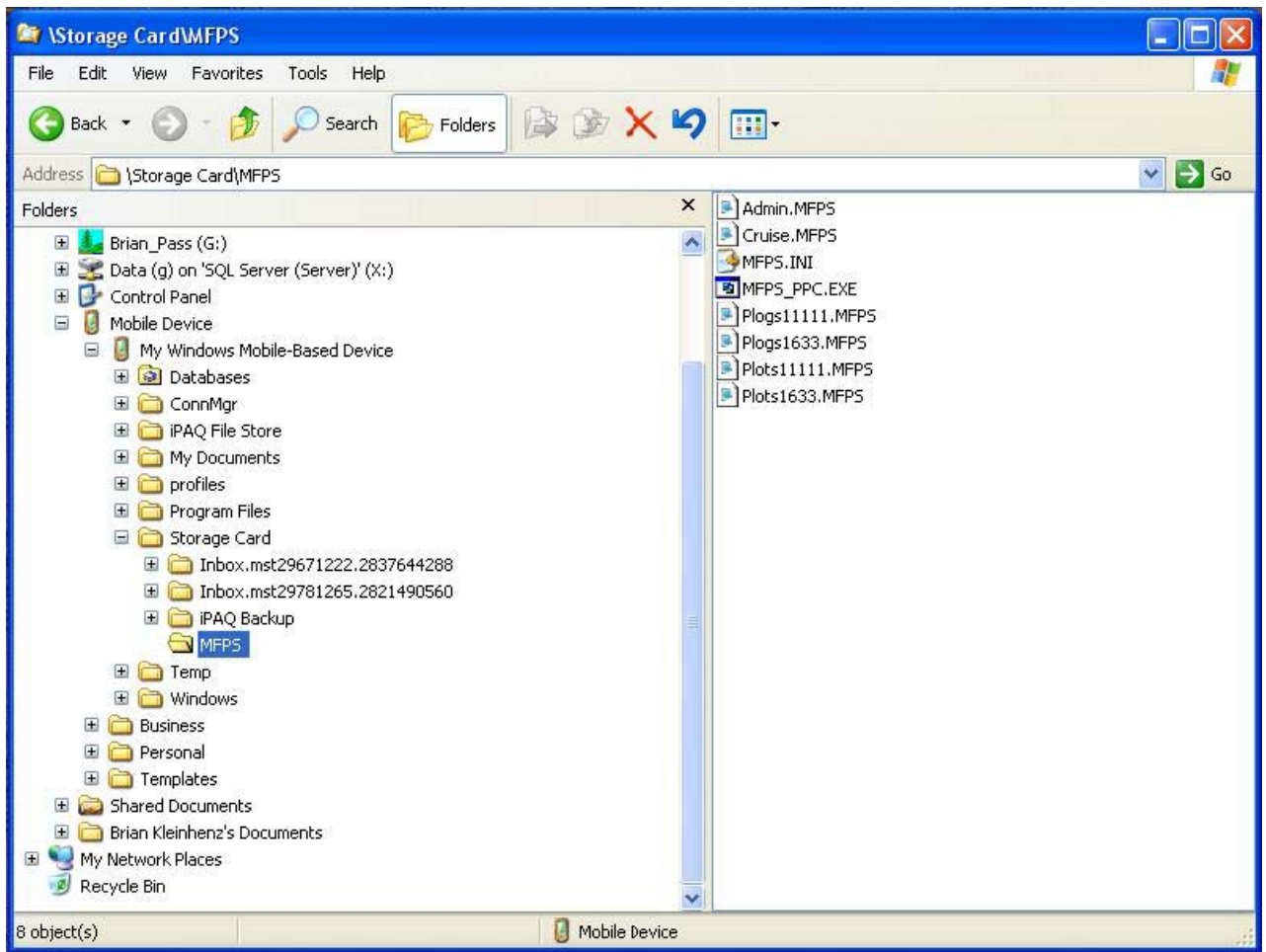


An Alternate Procedure for Importing of MFPS data

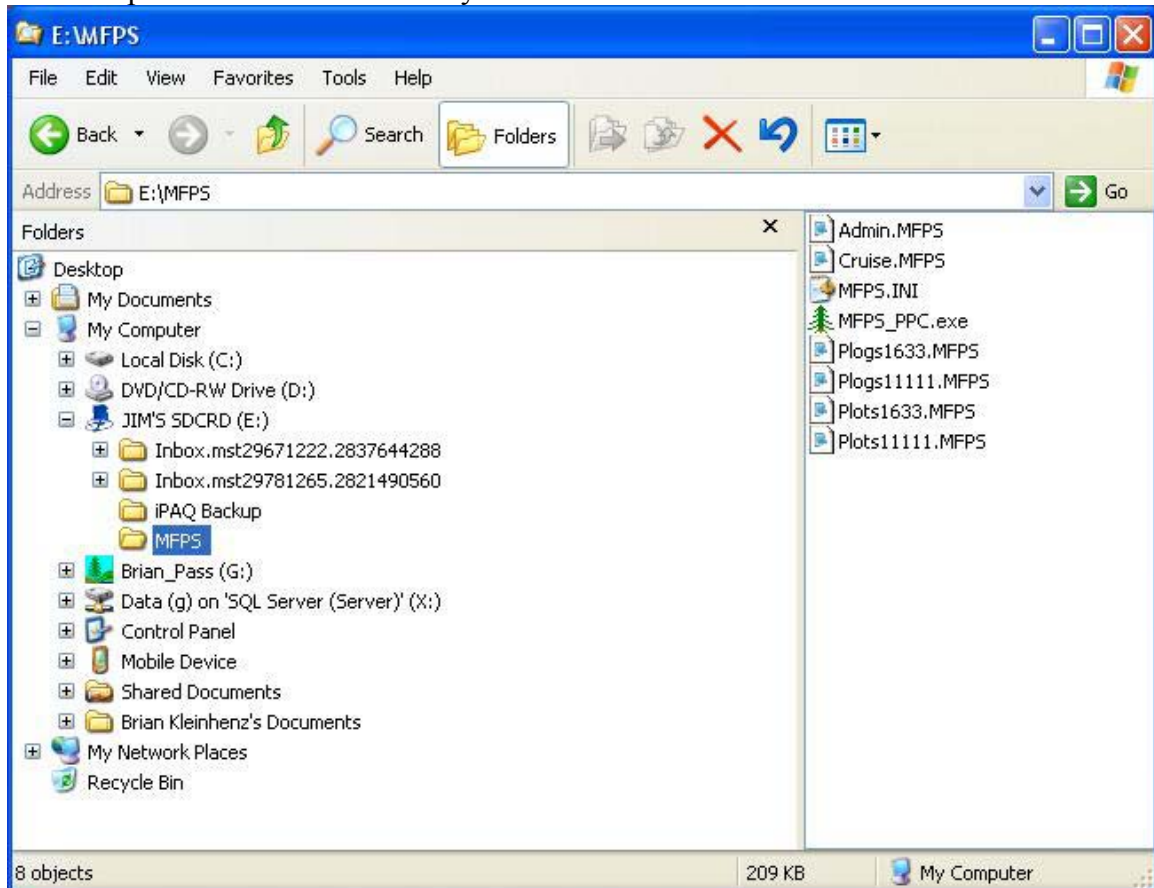
Users of FPS and its field collection counterpart, MFPS, may notice that they are no longer able to import MFPS data from directly inside of the FPS database application. Due to new security protocols that Microsoft is implementing for Windows Vista, FBRI is re-engineering this import application. Since this new import utility is not yet ready, we have laid out the following interim workflow for importing data. The basic steps of this procedure can be modified to allow import of data from other formats (spreadsheets, text files...).

1. Transfer MFPS data from your mobile device.

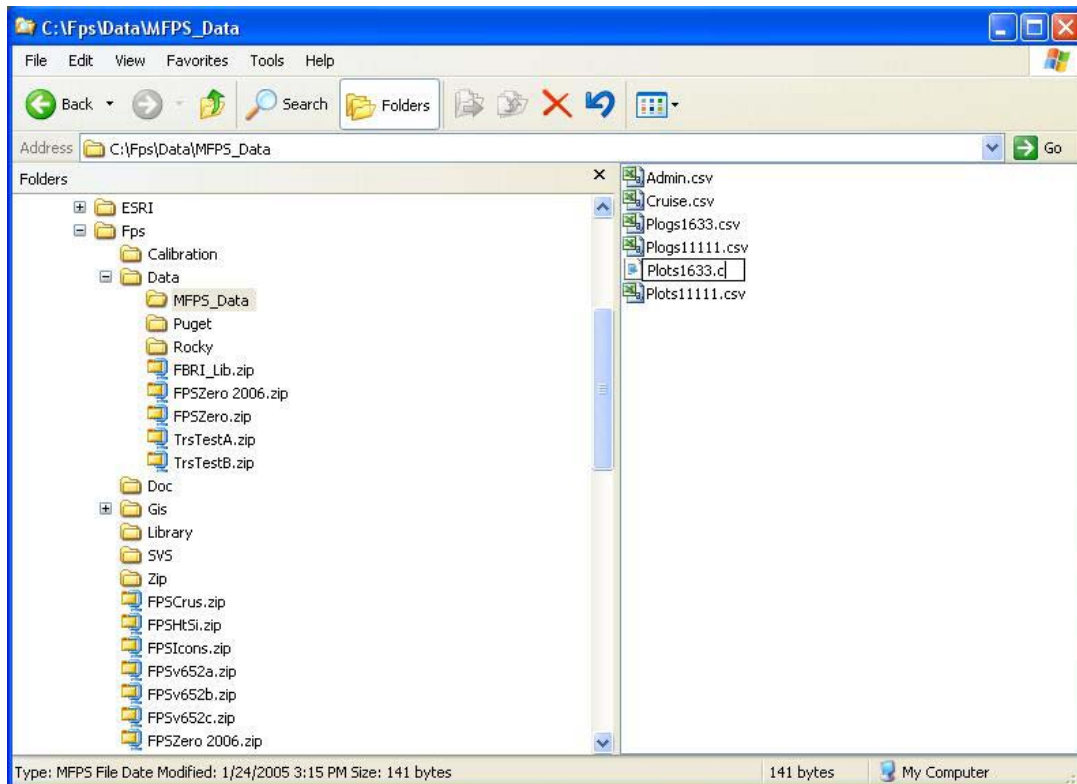
Data collected on a Windows CE device can be transferred directly to the user's PC if Microsoft ActiveSync is installed on the PC. In windows explorer navigate to the MFPS folder on the "Mobile Device". Copy all files with the .MFPS extension to some local directory (for example C:\FPS\Data).



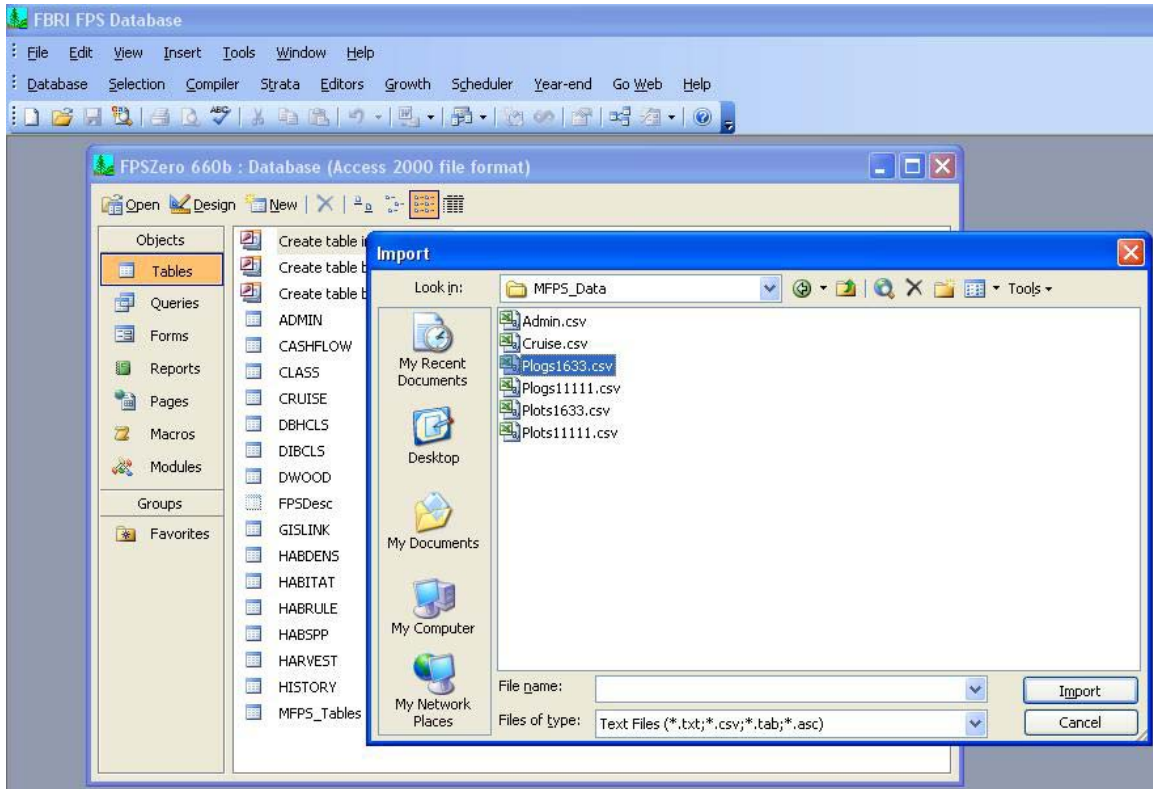
MFPS will default to storing data on the storage card. If the user's mobile device has a storage card, it is possible to remove the storage card from the mobile device and insert it into a card reader. The card reader will show up as a separate drive and data can be copied off to a local directory.



2. When all the files are copied to the hard drive, change the file extension of each file from “.MFPS” to “.csv”. This will allow MS access to find and read these files.



3. Open the FPS database that the MFPS data will be imported into. Under the Database menu select "Import Data". Make sure the "files of type:" drop down at the bottom is set to see text files. Now click "Import" and follow the steps of the data import wizard. Make sure that the first row of data is designated as the header row. Select one of the plots files. MFPS stores each cruised stand in a separate file which contains all the plot data for that stand. Once the plot data for the first stand is imported the user will see a new table in the database with the name of the stand (for example plots1633). Now the user can write an append query to append the plot data into the PLOTS table in FPS. The user will need to import each plot and plogs file one at a time. The cruise table will also need to be imported and appended to the CRUISE table in FPS.



4. FBRI has written a set of append queries to help you add the MFPS data into the tables in your FPS database. There is one query for appending PLOTS data, one for appending PLOGS data and one for appending cruise design data into the CRUISE table. MFPS tables will need to be imported one at a time and renamed to the following generic names for the FBRI update queries to identify them.

Imported Table name	New table name
Plots file data (for example plots1633)	MFPS_plots
Plogs file data (for example plogs1633)	MFPS_plogs
Cruise file data (for example Cruise)	MFPS_cruise

5. In this example the user has imported a file called plots1633.csv into the FPS database. The user has renamed the imported table to “MFPS_plots”. Now the user has an opportunity to open the table and check for errors and data entry mistakes. Once the user is satisfied with the data quality, they can run the update query called “MFPS_plots_import”. This query will append the data to the PLOTS table in FPS. Now the user will delete the table called “MFPS_plots” and start over by imported the next plots file (in this example plots11111.csv), renaming it and running the update query. There are separate update queries for PLOGS data and CRUISE table data. All three queries are available for download at the FBRI website and the SQL language for each is included with this document. The user can import the update queries or copy and paste the SQL statements into a query.

PLOTS update:

```
INSERT INTO PLOTS ( STD_ID, SPECIES, DBH, PLOT, TREE, [GROUP], X_ARC, Y_DIST,
TREES, HEIGHT, HT_CODE, TAP_DIA, TAP_HT, TAP_CODE, AGE, AGE_CODE, CROWN,
CRN_CODE, CRN_CLS, STATUS, DAMAGE, DEFECT1, DEFECT2, DEFECT3 )
SELECT MFPS_Plots.Std_ID, MFPS_Plots.SPP, MFPS_Plots.DBH, MFPS_Plots.Plot,
MFPS_Plots.Tree, MFPS_Plots.Grp, MFPS_Plots.X, MFPS_Plots.Y, MFPS_Plots.Trs,
MFPS_Plots.HT, MFPS_Plots.HTCd, MFPS_Plots.TapD, MFPS_Plots.TapH, MFPS_Plots.TapC,
MFPS_Plots.Age, MFPS_Plots.AgeC, MFPS_Plots.Cn, MFPS_Plots.CnCd, MFPS_Plots.CnCl,
MFPS_Plots.Stat, MFPS_Plots.Damg, MFPS_Plots.Def1, MFPS_Plots.Def2, MFPS_Plots.Def3
FROM MFPS_Plots;
```

PLOGS update:

```
INSERT INTO PLOGS ( Std_ID, Msmt, Plot, Tree, Piece, Length, Grade, Deduct, Code )
SELECT MFPS_Plogs.Std_ID, MFPS_Plogs.Msmt, MFPS_Plogs.Plot, MFPS_Plogs.Tree,
MFPS_Plogs.Piece, MFPS_Plogs.Length, MFPS_Plogs.Grade, MFPS_Plogs.Deduct,
MFPS_Plogs.Code
FROM MFPS_Plogs;
```

CRUISE update:

```
INSERT INTO CRUISE ( STD_ID, M_DATE, CRUISER, BAF, BAF_DBH, PLOT_AREA,
VEG_AREA, TRANSECT )
SELECT MFPS_Cruise.Std_ID, MFPS_Cruise.M_Date, MFPS_Cruise.Cruiser, MFPS_Cruise.BAF,
MFPS_Cruise.BAF_DBH, MFPS_Cruise.Plot_Area, MFPS_Cruise.Veg_Area, MFPS_Cruise.Transect
FROM MFPS_Cruise;
```