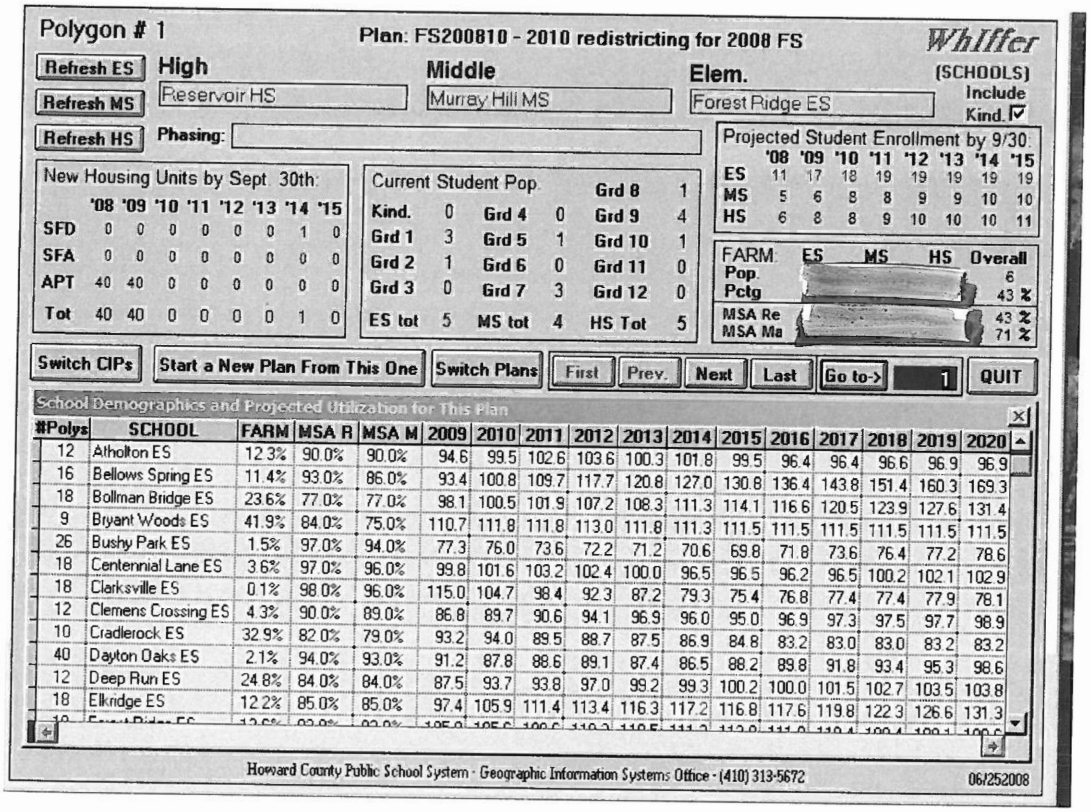
School Planning Office
Whiffer Application – Functional Requirements Description

The Whiffer is a software application that is used to simulate the effects of school attendance area boundary changes. Proposed changes are built and saved as redistricting plans. These plans are accessed by a software system that provides users reporting, display, and look-up capabilities.

Controls on the Main Application Screen

When the user starts the Whiffer by double-clicking its executable file (Whiffer.exe), the system responds by displaying the screen reproduced below.



Polygon # 1

Plan: FS200810 - 2010 redistricting for 2008 FS

Whiffer (SCHOOLS)

High: Reservoir HS

Middle: Murray Hill MS

Elem.: Forest Ridge ES

Include Kind. ☑

Phasing:

New Housing Units by Sept. 30th:

	'08	'09	'10	'11	'12	'13	'14	'15
SFD	0	0	0	0	0	0	1	0
SFA	0	0	0	0	0	0	0	0
APT	40	40	0	0	0	0	0	0
Tot	40	40	0	0	0	0	1	0

Current Student Pop.

				Grd 8	1
Kind.	0	Grd 4	0	Grd 9	4
Grd 1	3	Grd 5	1	Grd 10	1
Grd 2	1	Grd 6	0	Grd 11	0
Grd 3	0	Grd 7	3	Grd 12	0
ES tot	5	MS tot	4	HS Tot	5

Projected Student Enrollment by 9/30:

	'08	'09	'10	'11	'12	'13	'14	'15
ES	11	17	18	19	19	19	19	19
MS	5	6	8	8	9	9	10	10
HS	6	8	8	9	10	10	10	11

FARM	ES	MS	HS	Overall
Pop.	[redacted]			6
Pctg				43 %
MSA Re	[redacted]			43 %
MSA Ma				71 %

School Demographics and Projected Utilization for This Plan

#Polys	SCHOOL	FARM	MSA R	MSA M	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
12	Atholton ES	12.3%	90.0%	90.0%	94.6	99.5	102.6	103.6	100.3	101.8	99.5	96.4	96.4	96.6	96.9	96.9
16	Bellows Spring ES	11.4%	93.0%	86.0%	93.4	100.8	109.7	117.7	120.8	127.0	130.8	136.4	143.8	151.4	160.3	169.3
18	Bollman Bridge ES	23.6%	77.0%	77.0%	98.1	100.5	101.9	107.2	108.3	111.3	114.1	116.6	120.5	123.9	127.6	131.4
9	Bryant Woods ES	41.9%	84.0%	75.0%	110.7	111.8	111.8	113.0	111.8	111.3	111.5	111.5	111.5	111.5	111.5	111.5
26	Bushy Park ES	1.5%	97.0%	94.0%	77.3	76.0	73.6	72.2	71.2	70.6	69.8	71.8	73.6	76.4	77.2	78.6
18	Centennial Lane ES	3.6%	97.0%	96.0%	99.8	101.6	103.2	102.4	100.0	96.5	96.5	96.2	96.5	100.2	102.1	102.9
18	Clarksville ES	0.1%	98.0%	96.0%	115.0	104.7	98.4	92.3	87.2	79.3	75.4	76.8	77.4	77.4	77.9	78.1
12	Clemens Crossing ES	4.3%	90.0%	89.0%	86.8	89.7	90.6	94.1	96.9	96.0	95.0	96.9	97.3	97.5	97.7	98.9
10	Cradlerock ES	32.9%	82.0%	79.0%	93.2	94.0	89.5	88.7	87.5	86.9	84.8	83.2	83.0	83.0	83.2	83.2
40	Dayton Oaks ES	2.1%	94.0%	93.0%	91.2	87.8	88.6	89.1	87.4	86.5	88.2	89.8	91.8	93.4	95.3	98.6
12	Deep Run ES	24.8%	84.0%	84.0%	87.5	93.7	93.8	97.0	99.2	99.3	100.2	100.0	101.5	102.7	103.5	103.8
18	Elkridge ES	12.2%	85.0%	85.0%	97.4	105.9	111.4	113.4	116.3	117.2	116.8	117.6	119.8	122.3	126.6	131.3

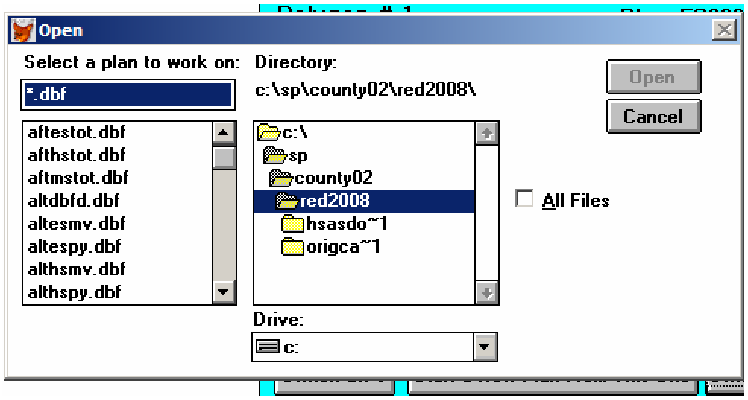
Howard County Public School System · Geographic Information Systems Office · (410) 313-5672

06/252008

The screen above is the only data entry screen used by the Whiffer. In its upper half, the Whiffer screen displays the data relevant to a single planning polygon, identified by the polygon number displayed in the upper left corner of the screen. The file of data underlying this display screen is essentially an electronic rolodex, organized in planning polygon order.

The boxed set of navigation buttons (First/Prev./Next/Last) in the middle of the screen move the record pointer to the designated planning polygon's record and refreshes the display with its contents. To display a specific polygon's data, the user enters the desired polygon number in the blue box to the right of the Go To button and clicks the mouse over the Go To button or presses the Enter key on the keyboard twice. Since the blue polygon number box is normally the active control on this screen, the user can rapidly move between polygons by simply entering a number on the keyboard, pressing the enter key twice, viewing the data, then repeating that sequence to view the next polygon of interest.

The first two buttons to the left of the navigation set are used to control the redistricting plan in use, i.e. which plan of many possible ones is currently being worked on. The Switch Plans button displays a prompt screen, on which the user can select a different plan to view or modify by double-clicking on its file name (see below). Once selected, the new plan's name and the first 30-40 characters of its description are shown at the top center of the display screen.



Open

Select a plan to work on: *.dbf

aftestot.dbf
afthstot.dbf
aftmstot.dbf
altdbfd.dbf
altesmv.dbf
altespy.dbf
althsmv.dbf
althspy.dbf

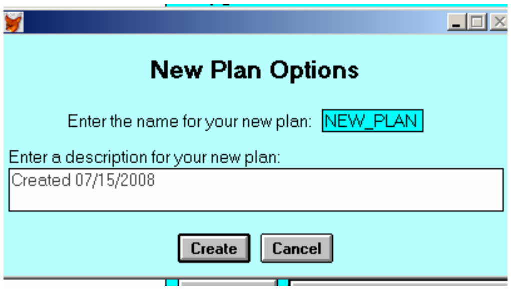
Directory: c:\sp\county02\red2008\

c:\
sp
county02
red2008
hsasdo~1
origca~1

Drive: c:

☐ All Files

The Start a New Plan From This One button will create an exact copy of the currently displayed plan and prompt the user to supply an eight-character name for the new plan.



New Plan Options

Enter the name for your new plan: NEW_PLAN

Enter a description for your new plan:
Created 07/15/2008

The user also has the option of entering a longer text description of the new plan on the prompt screen. When the user clicks on the Create button, the new plan is created and becomes the active plan in the Whiffer, and the data from its first polygon is displayed on the screen. The user can then proceed to modify the plan as necessary to create its new look. The newly-created plan and a portion of its description are displayed at the top of the screen.

The Switch CIPs button allows the user to designate a different CIP's set of building capacities for use in all of the Whiffer's capacity utilization calculations. The user selects the desired CIP file in a manner very similar to that described above for the Switch Plans button. The selected CIP is displayed in red immediately below the Whiffer logo in the upper right corner of the display. The Quit button terminates execution of the Whiffer, closes all of its files, and removes all Whiffer displays from the computer screen.

At the top right corner of the screen, below the Whiffer logo and the CIP name, is a checkbox with which the user can specify whether or not to count kindergarten students in the various printed reports and displays that the application creates. This option was useful during the time that kindergarten was a half-day program in the HCPSS, but is less important and seldom used since the advent of full-day kindergarten. This box is normally left checked, so that kindergarten students are included in all functions of the Whiffer.

The Whiffer controls that get the most use are the three pop-up boxes to the left of the Include Kind. Checkbox, under the headings "High", "Middle", and "Elem." These data entry fields are used to assign the displayed polygon to a school's attendance area. When the user clicks on one of these boxes, the appropriate set of HCPSS schools opens as a scrolling list. The currently assigned school is highlighted in the list when it opens. The user can select a different school from the scrolling list in any of the following ways:

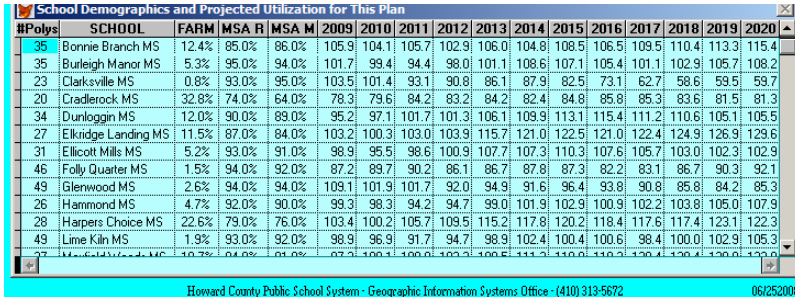
- scrolling the pick list display until the desired school is in view, highlighting its line in the display, and pressing the Enter key on the computer keyboard

- scrolling the pick list display until the desired school is in view and double-clicking its entry with the mouse

- typing the first few characters of the desired list entry, and pressing the Enter or Tab key when the desired entry is displayed and highlighted

The user may cancel their selection and leave the school unchanged by pressing the Esc key while the list is displayed. Changes are made directly into the displayed plan's data file as the scrolling list is closed, so there is no need for a separate step to save the user's changes.

In the bottom half of the Whiffer screen, there is a scrolling browse window with calculated data for each school. Schools are grouped by level (high, middle, elementary), and only one level is displayed at a time in this list. Along with the school name, the number of polygons assigned to the school, the FARM percentage, the pass rates for MSA Reading and Math tests, and the capacity utilization rates for the next 12 school years is shown. This data is automatically recalculated and refreshed every time a polygon is assigned to a different school in one of the three pop-up lists. Additionally, the user can refresh the display manually by clicking on one of the "Refresh" buttons in the upper left corner of the screen, under the polygon number. Clicking one of these refresh buttons for a level of schools that is different from the level currently displayed changes the browse window to display schools of the selected level.



School Demographics and Projected Utilization for This Plan

#Polys	SCHOOL	FARM	MSA R	MSA M	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
35	Bonnie Branch MS	12.4%	85.0%	86.0%	105.9	104.1	105.7	102.9	106.0	104.8	108.5	106.5	109.5	110.4	113.3	115.4
35	Burleigh Manor MS	5.3%	95.0%	94.0%	101.7	99.4	94.4	98.0	101.1	108.6	107.1	105.4	101.1	102.9	105.7	108.2
23	Clarksville MS	0.8%	93.0%	95.0%	103.5	101.4	93.1	90.8	86.1	87.9	82.5	73.1	62.7	58.6	59.5	59.7
20	Cradlerock MS	32.8%	74.0%	64.0%	78.3	79.6	84.2	83.2	84.2	82.4	84.8	85.8	85.3	83.6	81.5	81.3
34	Dunloggin MS	12.0%	90.0%	89.0%	95.2	97.1	101.7	101.3	106.1	109.9	113.1	115.4	111.2	110.6	105.1	105.5
27	Elkridge Landing MS	11.5%	87.0%	84.0%	103.2	100.3	103.0	103.9	115.7	121.0	122.5	121.0	122.4	124.9	126.9	129.6
31	Ellicott Mills MS	5.2%	93.0%	91.0%	98.9	95.5	98.6	100.9	107.7	107.3	110.3	107.6	105.7	103.0	102.3	102.9
46	Folly Quarter MS	1.5%	94.0%	92.0%	87.2	89.7	90.2	86.1	86.7	87.8	87.3	82.2	83.1	86.7	90.3	92.1
49	Glenwood MS	2.6%	94.0%	94.0%	109.1	101.9	101.7	92.0	94.9	91.6	96.4	93.8	90.8	85.8	84.2	85.3
26	Hammond MS	4.7%	92.0%	90.0%	99.3	98.3	94.2	94.7	99.0	101.9	102.9	100.9	102.2	103.8	105.0	107.9
28	Harpers Choice MS	22.6%	79.0%	76.0%	103.4	100.2	105.7	109.5	115.2	117.8	120.2	118.4	117.6	117.4	123.1	122.3
49	Lime Kiln MS	1.9%	93.0%	92.0%	98.9	96.9	91.7	94.7	98.9	102.4	100.4	100.6	98.4	100.0	102.9	105.3

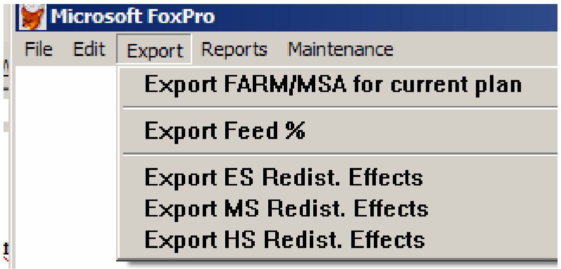
Howard County Public School System - Geographic Information Systems Office - (410) 313-5672 06/25200

Menu Options at the Main Application Screen

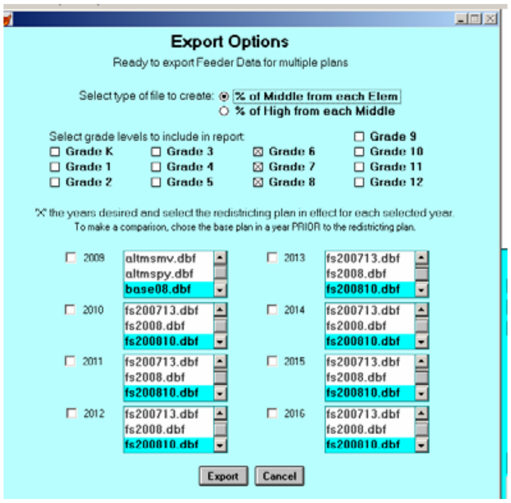
There are several menu options available at the Whiffer screen. The **File** menu contains a single entry to "Quit". The Quit option terminates execution of the Whiffer, closes all of its files, and removes all Whiffer displays from the computer screen.

The **Edit** menu contains the standard editing options, but they are not applicable to any of the processes taking place on this screen.

The **Export** menu has options to export three types of data files for use in Microsoft Excel spreadsheets that are linked into the Feasibility Study document (see menu displayed below). The "Export FARM/MSA for Current Plan" option writes a spreadsheet-compatible .xls file that contains a summary of FARM and MSA pass rates to the directory where the Whiffer data is stored. The export routine displays a message for the user with the name of the exported file when it is finished.


Microsoft FoxPro
File
Edit
Export
Reports
Maintenance
Export FARM/MSA for current plan
Export Feed %
Export ES Redist. Effects
Export MS Redist. Effects
Export HS Redist. Effects

The "Export Feed %" option provides a flexible means of exporting a file of feed rates between individual schools of different levels, for use in the Feasibility Study document. The user is prompted to select the type of file to export (see prompt screen below), the grade levels of students to be counted, and the redistricting plans to be used for each future school year. When the export is finished, a window appears with a message giving the name of the exported file.



Export Options

Ready to export Feeder Data for multiple plans

Select type of file to create: ● % of Middle from each Elem
○ % of High from each Middle

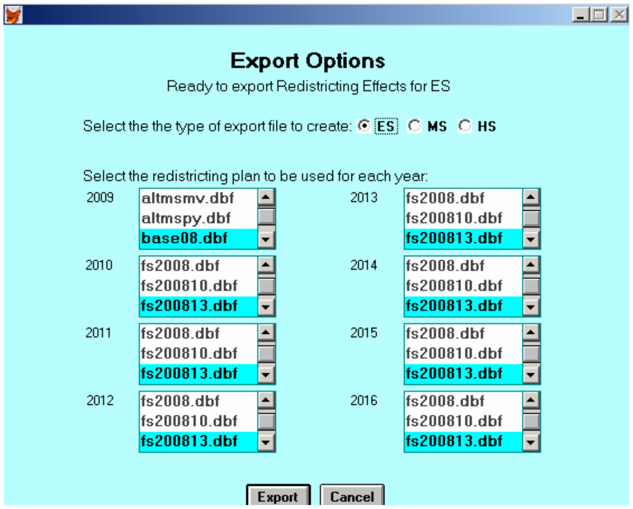
Select grade levels to include in report

			□ Grade 9
□ Grade K	□ Grade 3	☒ Grade 6	□ Grade 10
□ Grade 1	□ Grade 4	☒ Grade 7	□ Grade 11
□ Grade 2	□ Grade 5	☒ Grade 8	□ Grade 12

"X" the years desired and select the redistricting plan in effect for each selected year.
To make a comparison, chose the base plan in a year PRIOR to the redistricting plan.

□ 2009	altmsmv.dbf altmspy.dbf **base08.dbf**	□ 2013	fs200713.dbf fs2008.dbf **fs200810.dbf**
□ 2010	fs200713.dbf fs2008.dbf **fs200810.dbf**	□ 2014	fs200713.dbf fs2008.dbf **fs200810.dbf**
□ 2011	fs200713.dbf fs2008.dbf **fs200810.dbf**	□ 2015	fs200713.dbf fs2008.dbf **fs200810.dbf**
□ 2012	fs200713.dbf fs2008.dbf **fs200810.dbf**	□ 2016	fs200713.dbf fs2008.dbf **fs200810.dbf**

The “Export ES Effects/Export MS Effects/Export HS Effects” set of options creates files that contain the raw data to set up the redistricting effects report for the selected level of schools in the Feasibility Study document. The user is prompted, as shown above, to select the level of schools desired and the redistricting plan to use for each future school year.



Export Options

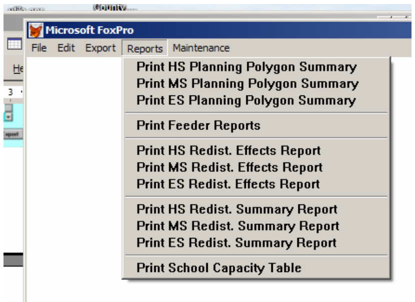
Ready to export Redistricting Effects for ES

Select the the type of export file to create: ● ES ○ MS ○ HS

Select the redistricting plan to be used for each year:

2009	altmsmv.dbf altmspy.dbf base08.dbf	2013	fs2008.dbf fs200810.dbf fs200813.dbf
2010	fs2008.dbf fs200810.dbf fs200813.dbf	2014	fs2008.dbf fs200810.dbf fs200813.dbf
2011	fs2008.dbf fs200810.dbf fs200813.dbf	2015	fs2008.dbf fs200810.dbf fs200813.dbf
2012	fs2008.dbf fs200810.dbf fs200813.dbf	2016	fs2008.dbf fs200810.dbf fs200813.dbf

The **Reports** menu contains a number of options to print redistricting plan data in various formats (see below). There is an example of each report included in Appendix A of this document.

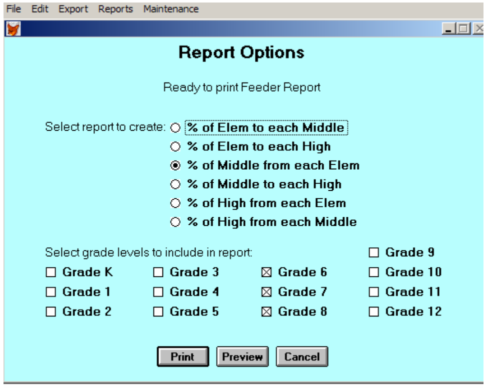


Microsoft FoxPro
File
Edit
Export
Reports
Maintenance
Print HS Planning Polygon Summary
Print MS Planning Polygon Summary
Print ES Planning Polygon Summary
Print Feeder Reports
Print HS Redist. Effects Report
Print MS Redist. Effects Report
Print ES Redist. Effects Report
Print HS Redist. Summary Report
Print MS Redist. Summary Report
Print ES Redist. Summary Report
Print School Capacity Table

Each of these reports is summarized in the bulleted items below:

- Polygon Summaries: Polygon Summaries are listings of the planning polygons, grouped by school to which they are assigned. User is prompted to Print or Preview the report, to select whether to print a detailed list or a school summary, and to select whether to include FARM and MSA statistics in the data listed for each polygon and school. The most comprehensive option of this report (detailed listing with FARM and MSA data included) prints a line of data which includes polygon number, FARM percentage, number of FARM students, pass rate for MSA Reading test takers, pass rate for MSA Math test takers, current student enrollment for the selected level of schools, and proposed future housing units by type of unit and projected student enrollment for the selected level of schools for each of the next six school years, for each polygon. Also included are summary statistics of the same data for each school. The “summary data only” option suppresses the polygon by polygon data lines and only prints the school summary statistics; the “no demographic data” option suppresses the FARM and MSA data from the polygon lines and from the school summaries.

- Feeder Reports: Feeder reports show the number and percentage breakdown of students currently residing in the geographic attendance areas shared between “feeding” schools at one level and “fed” schools at the next highest level. There are six versions of this report, as shown on the prompt screen above. The lower portion of the prompt screen allows the user to override the default grade level selections for printing a customized version of the feeder report.



Report Options

Ready to print Feeder Report

Select report to create:
○ % of Elem to each Middle
○ % of Elem to each High
● % of Middle from each Elem
○ % of Middle to each High
○ % of High from each Elem
○ % of High from each Middle

Select grade levels to include in report:

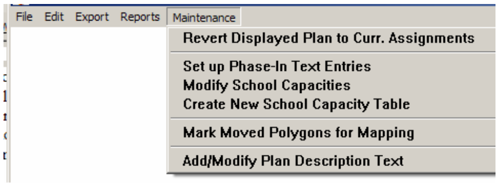
			□ Grade 9
□ Grade K	□ Grade 3	☒ Grade 6	□ Grade 10
□ Grade 1	□ Grade 4	☒ Grade 7	□ Grade 11
□ Grade 2	□ Grade 5	☒ Grade 8	□ Grade 12

- Redistricting Effects Reports: Redistricting Effects Reports are listings, by school, of the proposed changes in attendance areas for a given plan in a given school year. Changes are listed in terms of the planning polygons that are being moved out of, or into, a school’s attendance area. Each moved polygon is listed twice in the report, once for the school that is losing it, and once for the school that is gaining it. The number of students involved is listed with each polygon moved, and the school header and footer information shows the before and after school enrollments and capacity utilization rates. When running the Effects report, the user is prompted to select schools years for which reports are needed and to specify which redistricting plan will be in effect for each year selected.

- Redistricting Effects Summaries: The Redistricting Effects Summaries show the cumulative effects of proposed redistricting plans over the next 13 school years, for each school. For each future year listed, the Summary displays the proposed capacity and projected enrollments and capacity utilization rates, before redistricting and after redistricting. No polygon-specific data is shown on these reports.

- School Capacity Table: This report simply lists the data that is entered in the currently-selected CIP, in a formatted printout.

The **Maintenance** menu contains options to perform functions that are used infrequently, and thus do not require a control on the main screen. It is reproduced below:


File
Edit
Export
Reports
Maintenance
Revert Displayed Plan to Curr. Assignments
Set up Phase-In Text Entries
Modify School Capacities
Create New School Capacity Table
Mark Moved Polygons for Mapping
Add/Modify Plan Description Text

The “Revert Displayed Plan to Curr. Assignments” option is an “undo” function for reassigning all of the polygons back to the schools to which they are currently assigned in the present school year. This function acts only on the redistricting plan that is currently displayed on the Whiffer screen. Care should be taken in choosing to exercise this option, because all changes made in the current plan will be lost and will not be recoverable after it executes.

“Set up Phase-In Text Entries” creates a file to hold the text used to describe how changes in the high school attendance areas will be phased-in over the years following the adoption of a high school redistricting plan. Once this option has been executed, the user may select the planning polygons involved in high school redistricting, and enter the appropriate phase-in text in the data entry area next to the heading “Phasing:” on the Whiffer screen. The text editing functions available under the Edit menu may be used to facilitate the repetitive entry of the same text on multiple polygons.

The “Modify School Capacities” option opens the currently-selected CIP file in a spreadsheet-like data entry window for editing by the user. This option is used to correct errors in the CIP data and to create a new CIP by changing appropriate schools’ entries to their new values. In the latter case, it is used immediately after executing the “Create New School Capacity Table” option to copy the currently-selected CIP file to a new CIP file and make the new CIP file the selected CIP in the Whiffer.

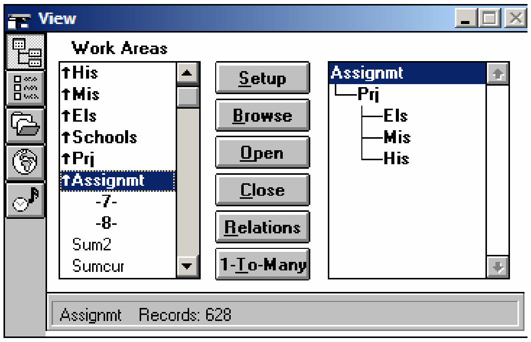
“Mark Moved Polygons for Mapping” sets one or more flags on each polygon in the current redistricting plan that has been assigned to a different school than it is assigned to in the school boundaries that are currently in effect. There is a separate flag for elementary, middle, and high school assignments. These “moved” flags are used in software programs that analyze the strengths and weaknesses of redistricting plans, and in the GIS software when creating maps showing movement of polygons due to redistricting.

The “Add/Modify Plan Description Text” option allows the user to change the currently-displayed redistricting plan’s description. The description is initially available for editing when a new plan is created, and this option allows the user to add a description if one was not entered then, or to modify it, at any time after the plan is established.

Data Environment Description

The default data environment for the Whiffer application consists of three data files and two memory cursors (SQL shorthand for “current sort” file, which is basically a data file stored in memory that is the result of a query against data files that exist on mass storage media, such as a computer hard drive.) Some of the procedures that make up the Whiffer’s software code open additional files during the time that they are executing. Most of these temporary use files are closed when the procedure returns control to the main Whiffer program.

The table below shows a schematic diagram of the default Whiffer data environment:



The top four files listed in the left-hand column of the diagram are alternate views of the same physical current CIP data file, with each view given a generic alias when it is opened by the program. The alias "His" refers to the CIP data file filtered so as to make only the high school records visible to the Whiffer; the "Mis" file filters out all but the middle school records; and the "Els" file filters out all but the elementary school records. The "Schools" file is a view of the CIP file with all records visible for use. The "His", "Mis", and "Els" views have the appropriate school records sorted by school name, and this sort sequence is implemented via a conditional index, which also accomplishes the necessary filtering to complete the desired views. The "Schools" data file is also sorted by school name, but via a conventional, non-filtering index. The default CIP data file is named "Schools.dbf", but additional CIP files can be created as desired by the Whiffer user, and given any desired name.

The "Prj" data file is a file of projection data, by planning polygon. It is sorted by planning polygon ID number. The default projection data file is always named "PpprojNN.dbf", where NN is the last two digits of the calendar year in which the projection was created (e.g. the 2008 projection's data would be stored by polygon in a file named "Ppproj08.dbf".) The projection file is the source for the enrollment, housing unit, FARM rate, and MSA pass rate data displayed on the main Whiffer screen.

The "Assignmt" data file is a file of plan data, by planning polygon. A plan file contains the planning polygon ID number and the school assignments for each planning polygon, among other data, as specified by the redistricting plan it represents. The plan file is sorted by planning polygon ID number, and it is the source of the school assignment data shown on the main Whiffer screen. The default Whiffer plan data file name is "BaseNN.dbf", where NN is the last two digits of the calendar year in which the plan was created (e.g. 2008's default plan data would be stored by polygon in a file named "Base08.dbf".) This "Base" plan represents the HCPSS school boundaries as they currently exist, without any proposed redistricting. Additional plan files can be created as desired by the Whiffer user, and given any desired name.

The "Sum2" data file is a cursor created by the "getstat" procedure of the Whiffer code (see code below). This cursor contains a record for each school, with data showing the school's FARM percentage, MSA Math pass rate, and MSA Reading pass rate. At any point in time, it only contains records for the user's currently-selected school level, and it is sorted in the natural order of its creation, which is alphabetically by school name.

The “Sumcur” data file is a cursor created by the “getstat” procedure of the Whiffer code (see code below). This cursor contains a record for each school, with data showing the school’s FARM percentage, MSA Math pass rate, and MSA Reading pass rate, number of polygons assigned, and capacity utilization rates for each of the next twelve years. At any point in time, it only contains records for the user’s currently-selected school level, and it is sorted in the natural order of its creation, which is alphabetically by school name. The data in this cursor are displayed in the scrolling browse list of school data, located in the bottom half of the main Whiffer screen.

The right-hand column of the diagram above shows the default linkages, or relations, established by the Whiffer program when it opens its default data files. The plan data file (“Assignmt”) is related to the projection data file (“Prj”) by planning polygon ID number. There is a one-to-one relationship between records in these two files. The projection data file (“Prj”) is related to each of the three views of the CIP file (“His”, Mis”, and ”Els”) by school name (respectively, “hs_home”, “ms_home”, and “es_home”). Each projection file record will be linked to three CIP records, one at each school level, by this arrangement.

Additional Data Files

Five additional data files are typically present with the default Whiffer data files. These files are integral to the Whiffer set-up process (described in the documentation of the “MSA_Fred” FoxPro program). They are stored with the Whiffer data so that an exact copy of the source data (as it it was when the Whiffer was established for the current school year) is available for later trouble-shooting or additional data analysis.

The student enrollment data file is always named “bst_stNN.dbf”, where NN is the last two digits of the school year for which the file was created (e.g. default student enrollment data for any date during the period 10/1/2007 to 9/30/2008 would be stored in a file named “bst_st08.dbf”.) This file contains a record of data for each student enrolled in HCP)SS at the time of its creation. The data in this file should be considered CONFIDENTIAL.

The FoxPro color scheme used by the Whiffer is stored in a file named “colorrsc.dbf”. FoxPro color schemes are very complex to set up and maintain, and it is recommended that this file NOT be modified in any way.

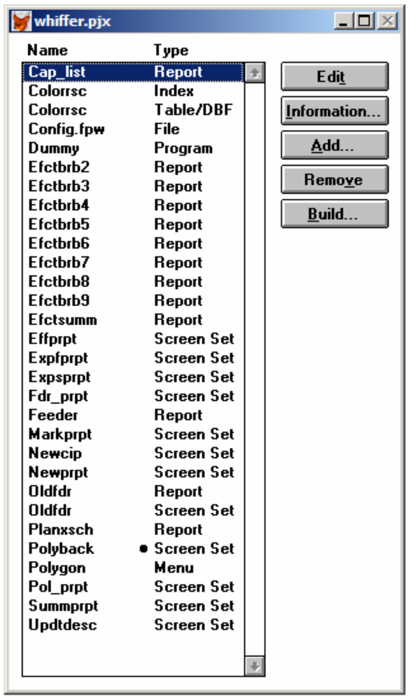
The HCPSS set of planning polygons is contained in the data file named “planpoly.dbf”. This file also contains the history of school assignments for each polygon since approximately 2002. It is not accessed by the Whiffer, but is included because it is the source of the Whiffer’s “baseNN.dbf” data file, during the set up process.

The “pp_hsg.dbf” data file is the file of future housing units, as created by the County Department of Planning and Zoning. It is not directly accessed by the Whiffer, but is included because it is the source of the Whiffer’s future housing unit data during the set up process.

The “schlupdt.dbf” file is a file of “by school” projection data that is exported from the HCPSS enrollment projection tool. It is not directly accessed by the Whiffer, but is included because it is the source of the Whiffer’s future enrollment data during the set up process.

Technical Description

The Whiffer is a FoxPro application, compiled (and distributed with the FoxPro support library) as a stand-alone executable file. In the FoxPro development environment, it is managed as a project, with the components shown below.



Name	Type
Cap_list	Report
Colorrsc	Index
Colorrsc	Table/DBF
Config.fpw	File
Dummy	Program
Efctbrb2	Report
Efctbrb3	Report
Efctbrb4	Report
Efctbrb5	Report
Efctbrb6	Report
Efctbrb7	Report
Efctbrb8	Report
Efctbrb9	Report
Efctsumm	Report
Effprpt	Screen Set
Expfprpt	Screen Set
Expsprpt	Screen Set
Fdr_prpt	Screen Set
Feeder	Report
Markprpt	Screen Set
Newcip	Screen Set
Newprpt	Screen Set
Oldfdr	Report
Oldfdr	Screen Set
Planxsch	Report
Polyback	● Screen Set
Polygon	Menu
Pol_prpt	Screen Set
Summprpt	Screen Set
Updtdesc	Screen Set

Items listed as reports in the project are self-explanatory; the screen sets are primarily single display screens used as user prompt screens for the export and report-generating processes of the Whiffer. The one exception to this is the screen set called “polyback”, which consists of a background screen and the upper half of the main Whiffer display screen, associated together in a single screen set so they will activate and execute as a single entity.

The “polygon” menu is the main menu object that generates all of the Whiffer’s menu options.

Miscellaneous files include the two “colorrsc” files that set up and control the Whiffer’s teal color scheme, the config.fpw file that contains default FoxPro runtime environmental settings, and the “dummy” program file that exists solely to head off compiler reference errors and does not need to be modified once it is initially created.

The remainder of this section is devoted to a complete listing of the Whiffer program code.

Polyback screen setup code: The Setup code on the Polyback screen executes once whenever the Whiffer.exe is first started by the user. This block of code is used primarily to set up the Whiffer’s data and operational environment.

```
* initial environmental settings:
SET TALK OFF
```

```
set escape on
set lock off
set exclusive off
set century on
set bell off
set confirm off
set deleted on
set near on
set exact off
set status off
set status bar off
set safety off
m.yr=2008                 && current calendar year, is also first projected Sept 30th enrollment -
                          move up one yr each year
m.schyr='0809'    && should always agree with line above, school year of first projected enrollment
m.prjfile='ppproj'+left(m.schyr,2)      && derived name of file containing polygon-level housing
                                        projections
* initialize Whiffer settings - get and overwrite from memory if settings files exist
m.k_in=.f.        && kindergarten set in or out?
if file('k.mem')
   restore from k.mem additive
endif
m.cur_lev='ES'    && currently level of schools displayed in browse list
if file('lev.mem')
   restore from lev.mem additive
endif
m.cur_db='a'      && name of currently-displayed plan ("a" is just a place-holder)
if file('db.mem')
   restore from db.mem additive
endif
m.cur_cip='a'     && name of currently-selected CIP ("a" is just a place-holder)
if file('cip.mem')
   restore from cip.mem additive
endif
* additional settings
m.ori=''          && default orientation of printed reports (port/land - not used at this time)
m.rept_text='Ready to print report'     && default title on user prompt screens for reports
m.go='Cancel'     && default action on report prompt screens
m.wait='Save'     && default action on file selection prompt screens
m.toprec=0                && default value for record number of first record in sort sequence
m.botrec=0                && default value for record number of last record in sort sequence
M.CHOICE=0                && default value of screen object number chosen by user
m.go_to=1         && default value for user's chosen polygon number
m.cip_in_use='(none)'     & initial screen value of current CIP name
m.pln_in_use='(none)'     & initial screen value of current plan name
close databases           && close any stray open data files and start in a known state
```

```
*wait window "Error handling is OFF"                && turn on when debugging, comment out
                                                    the following line
on error do err_msg with error(),message()    && turn on for production use, comment out
                                              preceding line

do dummy.prg    && just included to make dummy.prg part of the compiled exe file
do colorset             && sets a color scheme for the whole application
do setup        && calls setup proc to open standard data file configuration
do polygon.mpr  && activates main menu
* establish a window in polyback screen for the school browse list display on the bottom half of
polyback screen
define window browind from 20.5,1.4 to 39,118.15 nofloat close none font 'MS Sans Serif',8 in
window polyback
* call the "get_stat" procedure to refresh and display the school browse list in polyback screen
=get_stat(m.cur_lev)
```

<u>**Polyback screen Cleanup code:**</u> The Cleanup code for the Whiffer is a set of procedures that are called by other procedures or by controls, such as menu options and buttons, on the surface of the main Whiffer screen.

```
procedure prt_summ              && prints redistricting effects summary report, redistricting effects
                                report, or effects export
    parameters lev,typ          && call from menu code passes school level ('ES','MS','HS) and
                                type of report/export ('sum','eff','exp')
    lev=alltrim(lev)            && remove any extra spaces
    m.plan2=m.cur_db            && intitalize all of the future year plan fields
    m.plan3=m.cur_db
    m.plan4=m.cur_db
    m.plan5=m.cur_db
    m.plan6=m.cur_db
    m.plan7=m.cur_db
    m.plan8=m.cur_db
    m.plan9=m.cur_db
    if file('plnyrs.mem')       && bring back previously-saved settings from prompt screen
            restore from plnyrs.mem additive
    endif
    m.go='Cancel'               && initialize user's action button
    * set up macro substitution variables for 13 iterations, but year number is 2 thru 14 to match field
    names in ppprojNN.dbf
    for n=2 to 14
            m.yrfld='M.year'+alltrim(str(n))        && create a set of fields "m.yearN"
            &yrfld=alltrim(str(year(date())-1+n))   && initialize each m.yearN field with a
                                                    different year's value,
    *                                               starting with next calendar year
            m.yeer='m.yr'+alltrim(str(n))                   && create a set of fields "m.yrN"
```

```
        m.yr_txt='m.yr'+alltrim(str(n))+'_txt'    && create a set of fields "m.yr_txtN"
        &yeer=.f.                       && initialize each m.yrN field with a value of "false"
        &yr_txt=alltrim(str(year(date())-1+n))    && initialize each m.yr_txtN field with the
                                                  text version of
*                                       a different year's value, starting with next calendar year
endfor
do case                                 && intitalize screen title and open user prompt screen
case typ='sum'                          && settings for summary report
        m.rept_text='Ready to print Redistricting Summary Report for '+alltrim(lev)
        do summprpt.spr                 && call summary report prompt screen
case typ='eff'                          && settings for effects report
        m.rept_text='Ready to print Redistricting Effects Report(s) for '+alltrim(lev)
        do effprpt.spr                  && call effects report prompt screen
case typ='exp'                          && settings for effects export
        * initialize an array for each future year's data
        dimension mv2[1],mv3[1],mv4[1],mv5[1],mv6[1],mv7[1],mv8[1],mv9[1],mv10[1],
                mv11[1],mv12[1],mv13[1],mv14[1]
        m.xfile=''              && initialize a variable to hold the name of the export file
        m.rept_text='Ready to export Redistricting Effects for '+alltrim(lev)
        do expsprpt.spr                 && call effects export prompt screen
endcase
if m.go#'Cancel'        && proceed if the user has not elected to cancel the process from the
                        prompt screen
        m.plan10=m.plan9        && prompt screen only has room to specify 8 future years'
                                redistricting plans
        m.plan11=m.plan9        && so make years 9 thru 13 the same plan as year 8
        m.plan12=m.plan9
        m.plan13=m.plan9
        m.plan14=m.plan9
        m.rec=recno()           && save the record number of current record, so can get back
                                to it at the end
        m.pphm='pp.'+lev+'_home'        && eventual field name for school names in
                                        projections file,
*                                       for substitution into SQL select command below
        m.asghm='asg.'+lev+'_home'      && eventual field name for school names in
                                        plan files,
*                                       for substitution into SQL select command below
        m.schdb=iif(lev='HS','his',iif(lev='MS','mis','els'))       && name of school file
        m.schl=m.schdb+'.schl_name'     && fully-qualified name of school name field in
                                        school file
        do case                         && call appropriate processes to print or export data
        case typ='exp'                          && "effects" export
                if m.lev='ES'           && elementary school data requested
                        m.set_k=m.k_in  && save current setting of "K included" flag
                        m.k_in=.t.      && reset "K included" flag temporarily to "true"
                        =get_curs()     && call procedure to create cursor from which export
```

```
                                        is done
                        =exp_effs()     && call procedure to export data
                        use             && close cursor in current work area
                        m.k_in=.f.      && reset "K included" flag temporarily to "false"
                        =get_curs()     && call procedure to create cursor from which export
                                        is done
                        =exp_effs()             && call procedure to export data
                        m.k_in=m.set_k          && reset "K included" flag to original value
                else                            && other than ES data requested
                        =get_curs()     && call procedure to create cursor from which export
                                        is done
                        =exp_effs()             && call procedure to export data
                endif
        case typ='sum'                  && redistricting effects summary report
                =get_curs()     &&  call procedure to create cursor from which report is done
                =prt_rept('efctsumm')           && call general-purpose report printing
                                                procedure to print report form "efctsumm.frx"
        case typ='eff'          && redistricting effects report
                =get_curs()     && call procedure to create cursor from which report is done
*               && restrict record access to those of desired school level
                set filter to level=iif(lev='HS','3',iif(lev='MS','2',iif(lev='ES','1','0')))
                for m=2 to 9    && repeat for each possible future year
                        m.rept="efctbrb"+alltrim(str(m))        && create name of report for
                                                                macro substitution later
                        m.yeer='m.yr'+alltrim(str(m))           && create name of user's desired
                                                                years field for macro substitution
                        if &yeer        && if user has called for this year's report on the
                                        prompt screen
                                =prt_rept(m.rept)       && call the general-purpose report
                                                        printing procedure to print report form
                                                        "efctbrb2.frx", etc.
                        endif
                endfor
        endcase
        use             && close all open cursors
        select moves2
        use
        select moves3
        use
        select moves4
        use
        select moves5
        use
        select moves6
        use
        select moves7
        use
        select moves8
        use
        select moves9
        use
        select moves10
        use
        select moves11
        use
        select moves12
        use
        select moves13
        use
        select moves14
        use
        select assignmt
        goto m.rec     && go back to record that was current before this procedure was run
        =get_stat(lev)          && call getstat procedure to refresh the school browse list
    endif
    save to plnyrs.mem all like plan*    && save prompt screen settings for future use
    _CUROBJ=OBJNUM(M.go_to)              && place screen's cursor on the "Go to" button
    if typ='exp'     && if the user has chosen to export, tell them the names of the exported files
        wait window 'Data written to "'+alltrim(m.xfile)+'".  '+chr(13)+"Press any key
            to continue . . ."
    endif
    return          && return control to the main Whiffer screen program

procedure get_curs              && creates a set of SQL cursors, one for each future year
    * initialize a set of expressions containing the field names to be summed together to get the
        future enrollment projection
    * Add in K totals if user requested ES and K is being counted
    m.prj2='prj.'+lev+'proj2'+iif(m.k_in and lev='ES','+prj.ksproj2','')     && 2nd yr projection
    m.prj3='prj.'+lev+'proj3'+iif(m.k_in and lev='ES','+prj.ksproj3','')     && 3rd yr projection
    m.prj4='prj.'+lev+'proj4'+iif(m.k_in and lev='ES','+prj.ksproj4','')     && 4th yr projection
    m.prj5='prj.'+lev+'proj5'+iif(m.k_in and lev='ES','+prj.ksproj5','')     && 5th yr projection
    m.prj6='prj.'+lev+'proj6'+iif(m.k_in and lev='ES','+prj.ksproj6','')     && 6th yr projection
    m.prj7='prj.'+lev+'proj7'+iif(m.k_in and lev='ES','+prj.ksproj7','')     && 7th yr projection
    m.prj8='prj.'+lev+'proj8'+iif(m.k_in and lev='ES','+prj.ksproj8','')     && 8th yr projection
    m.prj9='prj.'+lev+'proj9'+iif(m.k_in and lev='ES','+prj.ksproj9','')     && 9th yr projection
    m.prj10='prj.'+lev+'proj10'+iif(m.k_in and lev='ES','+prj.ksproj10','')        && 10th yr
                                                                                    projection
    m.prj11='prj.'+lev+'proj11'+iif(m.k_in and lev='ES','+prj.ksproj11','')        && 11th yr
                                                                                    projection
    m.prj12='prj.'+lev+'proj12'+iif(m.k_in and lev='ES','+prj.ksproj12','')        && 12th yr
                                                                                    projection
    m.prj13='prj.'+lev+'proj13'+iif(m.k_in and lev='ES','+prj.ksproj13','')        && 13th yr
```

```
                                                                        projection
m.prj14='prj.'+lev+'proj14'+iif(m.k_in and lev='ES','+prj.ksproj14','')  && 14th yr
                                                                        projection
for m=2 to 14           && for each future year
        m.lpln='m.plan'+alltrim(str(m-1))       && name of memvar containing name of
                                                        last year's plan
        m.plan='m.plan'+alltrim(str(m))         && name of memvar containing name of
                                                        current year's plan
        m.pln='pln'+alltrim(str(m))             && selected plan's name
        m.mov='moves'+alltrim(str(m))                   && name of current year's cursor
        m.ndx='temp'+alltrim(str(m))+'.ndx'     && name of temporary index file for cursor
        m.db=&plan                      && put current year's plan name in m.db field
*       && put projection file name if processing first future year, otherwise, last year's plan
                name in m.prvdb
        m.prvdb=iif(m=2,m.prjfile,&lpln)
        select 0        && open new work area
        use &prvdb again alias pp   && open last year's plan, give it generic alias of "pp"
        select 0        && open new work area
        use &db again alias asg         && open current plan, give it generic alias of "asg"
        set order to plan_id        && set sort sequence to plan ID number
        m.lvlfld=alltrim(lev)+'_home'   && put name of appropriate home school field in
                                        m.lvlfld
*       && the following SQL select command creates a cursor i.e. a read-only dbf file in
                memory
*       && the SQL cursor has one record per planning polygon moved in the designated year,
                with the following structure:
*       && field 1: the name of the current home school
*       && field 2: the text 'To   ' plus the name of the new home school
*       && field 3: the planning polygon ID number
*       && field 4 through field 16: projected enrollment of this polygon for each future year
*       && this data is pulled out of the current plan file, the planning polygons file, the school
                file, and
*       && the projections file as needed. the files are related to each other as shown in the
                "where" clause below
*       && the resulting cursor is given the name "mv_a"
        select padr(&pphm,25,' ') as &lvlfld,;
                        'To   '+padr(alltrim(&asghm),25,' ') as desc, ;
                        pp.plan_id,;
                        iif(m<3,-1*round(&prj2,0),0) as pop2, ;
                        iif(m<4,-1*round(&prj3,0),0) as pop3, ;
                        iif(m<5,-1*round(&prj4,0),0) as pop4, ;
                        iif(m<6,-1*round(&prj5,0),0) as pop5, ;
                        iif(m<7,-1*round(&prj6,0),0) as pop6, ;
                        iif(m<8,-1*round(&prj7,0),0) as pop7, ;
                        iif(m<9,-1*round(&prj8,0),0) as pop8, ;
                        iif(m<10,-1*round(&prj9,0),0) as pop9, ;
                        iif(m<11,-1*round(&prj10,0),0) as pop10, ;
                        iif(m<12,-1*round(&prj11,0),0) as pop11, ;
                        iif(m<13,-1*round(&prj12,0),0) as pop12, ;
                        iif(m<14,-1*round(&prj13,0),0) as pop13, ;
                        -1*round(&prj14,0) as pop14 ;
                from asg,pp,(schdb),prj ;
                where pp.plan_id=asg.plan_id and asg.plan_id=prj.plan_id and ;
                        alltrim(&pphm)=alltrim(&schl) and ;
                        alltrim(&asghm)#alltrim(&pphm) ;
                into cursor mv_a
*       && the following SQL select command creates a second cursor with the same number of
                records as mv_a
*       && this cursor has one record per planning polygon moved in the designated year, with
                the following structure:
*       && field 1: the name of the new home school
*       && field 2: the text 'From ' plus the name of the current home school
*       && field 3: the planning polygon ID number
*       && field 4 through field 16: projected enrollment of this polygon for each future year
*       && this data is pulled out of the current plan file, the planning polygons file, the school
                file, and
*       && the projections file as needed. the files are related to each other as shown in the
                "where" clause below
*       && the resulting cursor is given the name "mv_b"
        select padr(&asghm,25,' ') as &lvlfld,;
                        'From '+padr(alltrim(&pphm),25,' ') as desc, ;
                        asg.plan_id,;
                        iif(m<3,round(&prj2,0),0) as pop2, ;
                        iif(m<4,round(&prj3,0),0) as pop3, ;
                        iif(m<5,round(&prj4,0),0) as pop4, ;
                        iif(m<6,round(&prj5,0),0) as pop5, ;
                        iif(m<7,round(&prj6,0),0) as pop6, ;
                        iif(m<8,round(&prj7,0),0) as pop7, ;
                        iif(m<9,round(&prj8,0),0) as pop8, ;
                        iif(m<10,round(&prj9,0),0) as pop9, ;
                        iif(m<11,round(&prj10,0),0) as pop10, ;
                        iif(m<12,round(&prj11,0),0) as pop11, ;
                        iif(m<13,round(&prj12,0),0) as pop12, ;
                        iif(m<14,round(&prj13,0),0) as pop13, ;
                        round(&prj14,0) as pop14 ;
                from asg,pp,(schdb),prj ;
                where pp.plan_id=asg.plan_id and asg.plan_id=prj.plan_id and ;
                        alltrim(&pphm)=alltrim(&schl) and ;
                        alltrim(&asghm)#alltrim(&pphm) ;
                into cursor mv_b
*       && the following SQL select command creates a third cursor containing a record for
                every record in
```

```
*       && both mv_a and mv_b, with the following structure:
*       && field 1: the name of the current home school
*       && field 2: the text 'From ' or the text 'To   ' plus the name of the new home school
*       && field 3: the planning polygon ID number
*       && field 4 through field 16: projected enrollment of this polygon for each future year
*       && the resulting cursor is given the name "move" plus the number of the future year
        select &lvlfld,mv_a.desc,plan_id,val(str(pop2,5)) as pop2,val(str(pop3,5)) as
                        pop3,val(str(pop4,5)) as pop4,;
                val(str(pop5,5)) as pop5,val(str(pop6,5)) as pop6,val(str(pop7,5)) as
                        pop7,val(str(pop8,5)) as pop8,;
                val(str(pop9,5)) as pop9,val(str(pop10,5)) as pop10,val(str(pop11,5)) as
                        pop11,val(str(pop12,5)) as pop12,;
                val(str(pop13,5)) as pop13,val(str(pop14,5)) as pop14 from mv_a ;
                union select &lvlfld,mv_b.desc,plan_id,val(str(pop2,5)) as
                        pop2,val(str(pop3,5)) as pop3,;
                val(str(pop4,5)) as pop4,val(str(pop5,5)) as pop5,val(str(pop6,5)) as
                        pop6,val(str(pop7,5)) as pop7,;
                val(str(pop8,5)) as pop8,val(str(pop9,5)) as pop9,val(str(pop10,5)) as
                        pop10,val(str(pop11,5)) as pop11,;
                val(str(pop12,5)) as pop12,val(str(pop13,5)) as pop13,val(str(pop14,5)) as
                        pop14 from mv_b ;
                into cursor &mov
        select asg              && close the current plan file
        use
        select pp               && close the planning polygon file
        use
        select &mov             && select the "moves" cursor
        index on &lvlfld to &ndx    && create an index on home school and sort the cursor
                                        accordingly
endfor  && repeat above for each future year until all 13 cursors have been created
SELECT 0    && open a new work area
use (cur_cip) again alias cip order name    && open the current CIP file and sort it by school
*       && relate the CIP file into each of the 13 cursors
set relation to alltrim(schl_name) into moves2,alltrim(schl_name) into moves3,
        alltrim(schl_name) into moves4,;
        alltrim(schl_name) into moves5,alltrim(schl_name) into moves6,
        alltrim(schl_name) into moves7,;
        alltrim(schl_name) into moves8,alltrim(schl_name) into moves9,
        alltrim(schl_name) into moves10,;
        alltrim(schl_name) into moves11,alltrim(schl_name) into moves12,
        alltrim(schl_name) into moves13,;
        alltrim(schl_name) into moves14
    && establish a one-to-many relationship from the CIP records to its related records in
        each cursor
set skip to moves2,moves3,moves4,moves5,moves6,moves7,moves8,moves9,moves10,
        moves11,moves12,moves13,moves14
    *       && select only CIP schools of the appropriate school level, then ready to export or report
    set filter to level=iif(lev='HS','3',iif(lev='MS','2',iif(lev='ES','1','0')))
    return && to procedure prt_summ

procedure exp_effs          && writes export file of redistricting effects data for one level of schools
    do case         && set up proper text string to describe grade levels exported
    case m.lev='ES' and m.k_in              && user requests ES and is counting grade K
            m.grds='K5'             && export contains grades K through 5
    case m.lev='ES' and not m.k_in                  && user requests ES and is not counting grade K
            m.grds='15'             && export contains grades 1 through 5
    case m.lev='MS'                 && user requests MS
            m.grds='68'             && export contains grades 6 through 8
    case m.lev='HS'                 && user requests HS
            m.grds='912'  && export contains grades 9 through 12
    endcase
    m.ufile='Efct_'+m.grds+".xls"       && compose text string for name of export file (with grade
                                                levels embedded)
    copy file hrdata to (ufile)         && copy "seed" file to export file
    m.filhand=fopen(m.ufile,2)          && open export file for use
    =fchsize(m.filhand,0)               && clear out contents of export file
    set skip to                         && turn off one-to-many relationship of CIP file into
                                                cursors (set up in get_curs proc)
    set relation to
    *       && write header record to export file
    =fwrite(m.filhand,'School'+chr(9)+chr(9)+m.year2+chr(9)+m.year3+chr(9)+m.year4+
            chr(9)+m.year5+chr(9)+m.year6+chr(9)+m.year7)
    =fputs(m.filhand,chr(9)+m.year8+chr(9)+m.year9+chr(9)+m.year10+chr(9)+m.year11+
            chr(9)+m.year12+chr(9)+m.year13+chr(9)+m.year14)
    scan    && work sequentially through each record of the CIP file, i.e school by school
            =fwrite(m.filhand,schl_name)                    && write school name to export file
            for n=2 to 14           && for each future year
    *       && initialize a set of fields containing the name of the cursor holding a future year's data
                    m.db='moves'+alltrim(str(n))
                    m.ara='mv'+alltrim(str(n))  && and the name of a corresponding memory array
                    select (db)             && activate the current year's cursor
    *       && sum each future enrollment across all polygons assigned to the school in the current
                    CIP record
    *       && and put the result in the appropriate memory array
                    sum pop2,pop3,pop4,pop5,pop6,pop7,pop8,pop9,pop10,pop11,pop12,pop13,
                            pop14 to array (ara) ;
                            for alltrim(iif(lev='ES',es_home,iif(lev='MS',ms_home,hs_home)))=
                            alltrim(cip.schl_name)
                    select cip      && switch back to CIP so scan:endscan will work properly
            endfor
    *       && create the future enrollments from the current enrollment of the school plus future
                    adjustments due
```

```
*                   && to potential redistricting in each future year
          m.enr2=round(geopop2+oodadj2+iif(m.k_in and m.lev='ES',kgeopop2+koodadj2,0)
                    +mv2[1],0)
          m.enr3=round(geopop3+oodadj3+iif(m.k_in and m.lev='ES',kgeopop3+koodadj3,0)
                    +mv2[2]+mv3[2],0)
          m.enr4=round(geopop4+oodadj4+iif(m.k_in and m.lev='ES',kgeopop4+koodadj4,0)
                    +mv2[3]+Mv3[3]+mv4[3],0)
          m.enr5=round(geopop5+oodadj5+iif(m.k_in and m.lev='ES',kgeopop5+koodadj5,0)
                    +mv2[4]+Mv3[4]+mv4[4]+mv5[4],0)
          m.enr6=round(geopop6+oodadj6+iif(m.k_in and m.lev='ES',kgeopop6+koodadj6,0)
                    +mv2[5]+Mv3[5]+mv4[5]+mv5[5]+mv6[5],0)
          m.enr7=round(geopop7+oodadj7+iif(m.k_in and m.lev='ES',kgeopop7+koodadj7,0)
                    +mv2[6]+Mv3[6]+mv4[6]+mv5[6]+mv6[6]+mv7[6],0)
          m.enr8=round(geopop8+oodadj8+iif(m.k_in and m.lev='ES',kgeopop8+koodadj8,0)
                    +mv2[7]+Mv3[7]+mv4[7]+mv5[7]+mv6[7]+mv7[7]+mv8[7],0)
          m.enr9=round(geopop9+oodadj9+iif(m.k_in and m.lev='ES',kgeopop9+koodadj9,0)
                    +mv2[8]+Mv3[8]+mv4[8]+mv5[8]+mv6[8]+mv7[8]+mv8[8]+mv9[8],0)
          m.enr10=round(geopop10+oodadj10+iif(m.k_in and m.lev='ES',kgeopop10+
                    koodadj10,0)+mv2[9]+Mv3[9]+mv4[9]+mv5[9]+mv6[9]+mv7[9]+
                    mv8[9]+mv9[9]+mv10[9],0)
          m.enr11=round(geopop11+oodadj11+iif(m.k_in and ;
                    m.lev='ES',kgeopop11+koodadj11,0)+mv2[10]+Mv3[10]+mv4[10]+mv5[10]+
                    mv6[10]+mv7[10]+mv8[10]+mv9[10]+mv10[10]+mv11[10],0)
          m.enr12=round(geopop12+oodadj12+iif(m.k_in and ;
                    m.lev='ES',kgeopop12+koodadj12,0)+mv2[11]+Mv3[11]+mv4[11]+mv5[11]+
                    mv6[11]+mv7[11]+mv8[11]+mv9[11]+mv10[11]+mv11[11]+mv12[11],0)
          m.enr13=round(geopop13+oodadj13+iif(m.k_in and ;
                    m.lev='ES',kgeopop13+koodadj13,0)+mv2[12]+Mv3[12]+mv4[12]+mv5[12]+
                    mv6[12]+mv7[12]+mv8[12]+mv9[12]+mv10[12]+mv11[12]+mv12[12]+
                    mv13[12],0)
          m.enr14=round(geopop14+oodadj14+iif(m.k_in and ;
                    m.lev='ES',kgeopop14+koodadj14,0)+mv2[13]+Mv3[13]+mv4[13]+mv5[13]+
                    mv6[13]+mv7[13]+mv8[13]+mv9[13]+mv10[13]+mv11[13]+mv12[13]+
                    mv13[13]+mv14[13],0)
*         && write all future enrollments for this school to output file in appropriate columns
*                   && i.e. tab-delimited output, with a line-feed/carriage return at end
          =fwrite(m.filhand,chr(9)+chr(9)+alltrim(str(m.enr2))+chr(9)+alltrim(str(m.enr3))+
                    chr(9)+alltrim(str(m.enr4))+chr(9)+alltrim(str(m.enr5)))
          =fwrite(m.filhand,chr(9)+alltrim(str(m.enr6))+chr(9)+alltrim(str(m.enr7))+chr(9)+
                    alltrim(str(m.enr8))+chr(9)+alltrim(str(m.enr9)))
          =fwrite(m.filhand,chr(9)+alltrim(str(m.enr10))+chr(9)+alltrim(str(m.enr11))+
                    chr(9)+alltrim(str(m.enr12))+chr(9)+alltrim(str(m.enr13)))
          =fputs(m.filhand,chr(9)+alltrim(str(m.enr14)))
endscan                       && repeat above steps for each school
*         && write file footer info to output file
*         && plan in use each future year
```

```
    =fwrite(m.filhand,'Plan in use:'+chr(9)+chr(9)+alltrim(strtran(m.plan2,'.dbf'))+chr(9)+
            alltrim(strtran(m.plan3,'.dbf'))+chr(9)+alltrim(strtran(m.plan4,'.dbf')))
    =fwrite(m.filhand,chr(9)+alltrim(strtran(m.plan5,'.dbf'))+chr(9)+
            alltrim(strtran(m.plan6,'.dbf'))+chr(9)+alltrim(strtran(m.plan7,'.dbf')))
    =fwrite(m.filhand,chr(9)+alltrim(strtran(m.plan8,'.dbf'))+chr(9)+
            alltrim(strtran(m.plan9,'.dbf'))+chr(9)+alltrim(strtran(m.plan10,'.dbf')))
    =fwrite(m.filhand,chr(9)+alltrim(strtran(m.plan11,'.dbf'))+chr(9)+
            alltrim(strtran(m.plan12,'.dbf'))+chr(9)+alltrim(strtran(m.plan13,'.dbf')))
    =fputs(m.filhand,chr(9)+alltrim(strtran(m.plan14,'.dbf')))
    *       && CIP in use and current date
    =fwrite(m.filhand,chr(9)+chr(9)+'CIP in use:'+chr(9)+m.cip_in_use+chr(9)+chr(9)+chr(9)+
            chr(9)+dtoc(date())+chr(9)+chr(9)+chr(9)+chr(9))
    *       && school level requested by user and grade levels counted for enrollment totals
    =fputs(m.filhand,alltrim(m.lev)+' population - grades '+left(alltrim(m.grds),1)+
            ' to '+alltrim(substr(m.grds,2)))
    =fclose(m.filhand)    && close output file
    m.xfile=alltrim(m.xfile)+'; '+alltrim(m.ufile)        && append current export file name to
                                                          existing string of file names
    return && to procedure prt_summ

procedure expt_fdr                  && exports a file of feed rate data
    parameters typ && currently "FS" for Feasibility Study (i.e. the destination of the exported file)
    m.rept_text='Ready to export Feeder Data for multiple plans'
    for n=2 to 9            && initialize fields for user prompt screen (8 years' worth)
            m.pln='m.plan'+alltrim(str(n))  && plan in effect each year in field named "m.planN"
            m.yeer='m.yr'+alltrim(str(n))        && year selected? T/F in field named "m.yrN"
            m.yr_txt='m.yr'+alltrim(str(n))+'_txt'        && year as char. string in field named
                                                          "m.yrN_txt"
            &pln=m.cur_db           && start with currently-selected plan in all plan name fields
            &yeer=.f.               && start with all years' check boxes unselected
            &yr_txt=alltrim(str(year(date())-1+n))        && calculate the appropriate calendar year
                                                          value for each future year
    endfor
    for m=0 to 12           && initialize all grade level chaeck boxes to unselected
            m.gr_fld='m.g'++alltrim(str(m))  && grade level selected? T/F in field named "m.gM"
            &gr_fld=.f.
    endfor
    m.choi=1        && default export will be "% of Middle from Each Elementary" (option 1 on
                            control named "m.choi" on prompt screen)
*   if file('fplnyrs.mem')  && inactive code
*           restore from fplnyrs.mem additive
*   endif
    m.plan2="base"+left(m.schyr,2)+".dbf"                 && initialize first year's plan to be the
                                                          current "base" plan
    for m=3 to 9            && initialize all other years to current plan
            m.pln='m.plan'+alltrim(str(m))                && create a memvar to hold name of
```

```
                            current year's plan field
    &pln=ALLTRIM(lower(m.pln_in_use))+".dbf"  && put name of current plan in
                                              current year's plan field
endfor
m.go='Export'           && initialize user's action button
do expfprpt.spr         && activate the user prompt screen
if m.go#'Cancel'        && proceed if the user has not elected to cancel the process from
                        the prompt screen
    select assignmt         && select the work area with the current plan open in it
    m.rec=recno()           && save the record number of current record, so can get
                            back to it at the end
    copy structure extended to feeds.dbf    && save the structure of the current plan file
                                            in a file called "feeds.dbf"
    select 0        && switch to an unused work area
    use feeds exclusive     && open the file containing the plan file structure
    zap             && clear out the structure records i.e. will create a new structure in this
                    file,
    *               && to describe a new data file with the fields defined below:
    insert into feeds (field_name,field_type,field_len,field_dec)
                    values ('recv_schl','C',25,0)       && add a receiving school field
    insert into feeds (field_name,field_type,field_len,field_dec)
                    values ('send_schl','C',25,0)       && add a sending school field
    for n=2 to 9    && add up to 8 sets of "number of students" and "feed percentage" fields
    *               && one set for each year that the user selected on the prompt screen
        m.yr_txt='m.yr'+alltrim(str(n))+'_txt'      && year as char. string in field
                                                    named "m.yrN_txt"
        m.yeer='m.yr'+alltrim(str(n))               && year selected? T/F in field
                                                    named "m.yrN"
        if &yeer        && insert a set of field descriptions if user selected the year
            insert into feeds (field_name,field_type,field_len,field_dec)
                                    values ('fd_'+&yr_txt+'_nm','N',4,0)
            insert into feeds (field_name,field_type,field_len,field_dec)
                                    values ('fd_'+&yr_txt+'_pc','N',5,1)
        endif
    endfor
    create tempfeed.dbf from feeds      && use the structure file to create an empty data
                                        file with the described structure
    index on alltrim(recv_schl)+alltrim(send_schl) tag schl   && create an index on
                                                receiving school plus sending school
    m.num_plans=0       && initialize a plan counter
    for n=2 to 9    && for each future year
        m.yeer='m.yr'+alltrim(str(n))               && name of the prompt screen field
                                                    for this iteration
        if &yeer        && if user selected this year on prompt screen
            m.num_plans=m.num_plans+1               && count how many plans' data are
                                                    being output, for later
```

```
                              m.pln='m.plan'+alltrim(str(n))                   && name of plan file (as selected by
                                                                     user on prompt screen) for this iteration
                              m.pln=strtran(&pln,'.dbf')                       && remove file extension to get plan
                                                                               name
                              m.yr_txt='m.yr'+alltrim(str(n))+'_txt'                     && calendar year as char.
                                                           string in field named "m.yrN_txt" for this iteration
                              m.num_fld='fd_'+&yr_txt+'_nm'                    && name of the "number of
                                                                               students" field for this iteration
                              m.pct_fld='fd_'+&yr_txt+'_pc'                    && name of the "feed percentage"
                                                                               field for this iteration
                              m.indx='tmp'+alltrim(str(n))+'.ndx'                        && name of index (to be
                                                                                         created later) for this iteration
                              select 0                     && select an empty work area
                              use &pln again               && open a second copy of the current plan
                              do case                      && determine proper settings depending on user’s
                                                           desired type of export
                              case m.choi=2                && user selected "High from Each Middle"
                                        m.field='asg.hs_home'                  && field name of receiving school
                                        INDEX ON hs_home+ms_home To &indx                            && create
                                                                                                   desired sort sequence
                                        m.title='% of High from each Middle'                       && report title
                                        m.ufile="Hi_fm_Mi.xls"                 && export file name
                                        m.rcvr='High School'                   && heading for receiving school
                                                                               column in output file
                                        m.sndr='MS Feeds'                      && heading for sending school
                                                                               column in output file
                              case m.choi=1                && user selected "Middle from Each Elementary"
                                        m.field='asg.ms_home'                  && field name of receiving school
                                        INDEX ON ms_home+es_home To &indx                            && create
                                                                                                   desired sort sequence
                                        m.title='% of Middle from each Elementary'                   && report title
                                        m.ufile="Mi_fm_El.xls"                 && export file name
                                        m.rcvr='Middle School'                 && heading for receiving school
                                                                               column in output file
                                        m.sndr='ES Feeds'                      && heading for sending school
                                                                               column in output file
                              otherwise                              && no user choice - warn user with alert
                                                                     message and return
                                        wait window "No option selected - re-run report from menu."+
                                                  chr(13)+;
                                                  "Press any key to continue . . ."
                                        set order to plan_id         && restore default sort sequence in plan file
                                        goto m.rec         && position record pointer on original record
                                        return && to main Whiffer screen without exporting anything
                              endcase
*         && the following SQL select command creates a cursor with one record per receiving
```

```
        school,
*       && with the following structure:
*       && field 1: the name of the receiving school
*       && field 2: the sum of the user's selected HS grade levels' enrollment, across all
        polygons assigned to the receiving school
*       && field 3: the sum of the user's selected MS grade levels' enrollment, across all
        polygons assigned to the receiving school
*       && field 4: the sum of the user's selected ES grade levels' enrollment, across all polygons
        assigned to the receiving school
*       && this data is pulled out of the current plan file and
*       && the projections file as needed. the files are related to each other by planning polygon
        ID number
*       && the resulting cursor is given the name "schtot"
                select alltrim(&field) as schl, ;
                        iif(m.g12,sum(pp.gr12_pop),0)+;
                        iif(m.g11,sum(pp.gr11_pop),0)+
                        iif(m.g10,sum(pp.gr10_pop),0)+
                        iif(m.g9,sum(pp.gr9_pop),0) as hs_tot, ;
                        iif(m.g8,sum(pp.gr8_pop),0)+
                        iif(m.g7,sum(pp.gr7_pop),0)+
                        iif(m.g6,sum(pp.gr6_pop),0) as ms_tot, ;
                        iif(m.g5,sum(pp.gr5_pop),0)+
                        iif(m.g4,sum(pp.gr4_pop),0)+
                        iif(m.g3,sum(pp.gr3_pop),0)+;
                        iif(m.g2,sum(pp.gr2_pop),0)+
                        iif(m.g1,sum(pp.gr1_pop),0)+
                        iif(m.g0,sum(pp.gr0_pop),0) as es_tot ;
                        from &pln asg,prj pp into cursor schtot where
                        pp.plan_id=asg.plan_id group by &field
                select schtot   && switch to the cursor data file
                index on schl to temp.ndx    && sort the cursor by school name
                select &pln             && switch back to the current plan file
                m.fld=substr(m.field,5)        && store the field name without the file
                                               name qualifier in "m.fld"
                set relation to alltrim(&fld) into schtot,plan_id into prj  && relate the
                                               plan file to the cursor by school name,
                *       && and to the projection file by planning polygon ID number
                m.fdr_tot=0             && intitalize some accumulators
                m.rcvr_tot=0
                m.adv_tot=0
                m.rec_schl=alltrim(iif(m.choi=2,hs_home,ms_home))      && store
                                                               name of receiving school
                m.snd_schl=alltrim(iif(m.choi=2,ms_home,es_home))      && store
                                                               name of sending school
*       && initialize a flag to indicate that the first line of a new receiving school is currently
        being written to the output file
```

```
            m.firstln=.t.
            scan    && sequentially look at every plan file record
*               && write first line for new receiving school
                if m.firstln            && first line for receiving school - get its
                                        total enrollment
                    m.firstln=.f.   && turn off "first line" flag
                    m.adv_tot=schtot.es_tot+schtot.ms_tot+schtot.hs_tot
                && need receiving school's total in advance (from schtot cursor)
                endif
*   && accumulate sending school's totals by adding current record's values to what was
    previously accumulated
*   && only count those grade levels selected by user on the prompt screen
                m.fdr_tot=m.fdr_tot+iif(m.g0,prj.gr0_pop,0)+
                    iif(m.g1,prj.gr1_pop,0)+iif(m.g2,prj.gr2_pop,0)+;
                    iif(m.g3,prj.gr3_pop,0)+iif(m.g4,prj.gr4_pop,0)+
                    iif(m.g5,prj.gr5_pop,0)+;
                    iif(m.g6,prj.gr6_pop,0)+iif(m.g7,prj.gr7_pop,0)+
                    iif(m.g8,prj.gr8_pop,0)+;
                    iif(m.g9,prj.gr9_pop,0)+iif(m.g10,prj.gr10_pop,0)+
                    iif(m.g11,prj.gr11_pop,0)+iif(m.g12,prj.gr12_pop,0)
                skip            && look ahead one record to see if it's a new
                                sending school
*           && if sending school changes or at end of file, write sending school totals
                if not alltrim(iif(m.choi=2,ms_home,es_home))==m.snd_schl or
                    EOF()
*  && if this combination of schools already has a record in the output file, replace the values
                    if seek(alltrim(m.rec_schl)+alltrim(m.snd_schl),
                        'tempfeed')
                        select tempfeed         && switch to the output file
*   && replace number of students and feed % on existing record
                        replace &num_fld with ROUND(m.fdr_tot,0),
                        &pct_fld with round(m.fdr_tot*100/m.adv_tot,1)
                        select &pln             && switch back to plan file
                                        so scan:endscan will work properly
                    else    && if no record currently exists for this school
                            combination, create a new record with SQL insert
                        insert into tempfeed
                            (recv_schl,send_schl,&num_fld,&pct_fld)
                            values (m.rec_schl,m.snd_schl,
                            ROUND(m.fdr_tot,0),
                            round(m.fdr_tot*100/m.adv_tot,1))
                    endif
*       && accumulate sending school's students into receiving school's totals
                    m.rcvr_tot=m.rcvr_tot+m.fdr_tot
*       && if receiving school is also changing, write receiving school's totals
                    if not alltrim(iif(m.choi=2,hs_home,ms_home))==
                        m.rec_schl or EOF()
*       && reset receiving school counter, first line indicator, and receiving school name
                        m.rcvr_tot=0
                        m.firstln=.t.
                        m.rec_schl=alltrim(iif(m.choi=2,
                            hs_home,ms_home))
                    endif
                    m.fdr_tot=0         && reset sending school counters
                                        and sending school name
                    m.snd_schl=alltrim(iif(m.choi=2,ms_home,es_home))
                endif
                skip -1         && go back to previous record in plan file so
                                scan/endscan will work properly
            endscan     && look at next plan file record and repeat above steps
            select &pln         && switch to current plan and close the data file
            use
        endif
    endfor      && repeat for each future year
    copy file hrdata to (ufile)         && copy "seed" file to Excel output file
    m.filhand=fopen(m.ufile,2)          && open Excel output file for use
    =fchsize(m.filhand,0)                   && clear out contents of Excel output file
*       && first line of Excel output file (HEADER)
    if typ="FS"         && currently, the only active option
*       && write the report title, grade levels being counted, and the current date
        =fPUTS(m.filhand,m.title+chr(9)+
            "Grades: "+iif(m.g0,'K','')+iif(m.g1,'1','')+
            iif(m.g2,'2','')+iif(m.g3,'3','')+;iif(m.g4,'4','')+iif(m.g5,'5','')
            +iif(m.g6,'6','')+iif(m.g7,'7','')+iif(m.g8,'8','')+iif(m.g9,'9','')+;
            iif(m.g10,'10','')+iif(m.g11,'11','')+iif(m.g12,'12','')+
            chr(9)+dtoc(date()))
    else    && not used currently
        =fPUTS(m.filhand,iif(m.choi=2,"hs_home","ms_home")+chr(9)+
            iif(m.choi=2,"ms_home","es_home")+chr(9)+"feed")
    endif
*       && write the names of selected plan for each year in a header record
    =fwrite(m.filhand,chr(9))           && output a single "Tab" character to align
                                        columns properly
    p=1         && initialize a counter to keep track of how many plans have been output
                vs number of plans counted above
    for n=2 to 9    && cycle through each of the possible future years
        m.yeer='m.yr'+alltrim(str(n))       && this year selected? user's selections in
                                            field named "m.yrN"
        if &yeer            && if user selected this year for output, write something in
                            Excel output file
            m.pln='m.plan'+alltrim(str(n))          && plan in effect each year
                                                    in field named "m.planN"
            if p=m.num_plans    && if now writing data for last plan, write plan
                                name and carriage return/line feed character
                =fputs(m.filhand,chr(9)+chr(9)+&pln)    && write two "tab”
                                        characters first to get proper spacing
            else        && if more plans follow, write just the plan name
                p=p+1 && increment the "number of plans written" counter
                =fwrite(m.filhand,chr(9)+chr(9)+&pln)   && write two "tab"
                                        characters first to get proper spacing
            endif
        endif
    endfor
    if typ='FS'         && currently, the only active option
*       && write calendar year headers
        =fwrite(m.filhand,chr(9))       && output a single "Tab" character to align
                                    columns properly
        p=1             && initialize a counter to keep track of how many plans have been
                        output vs number of plans counted above
        for n=2 to 9    && cycle through each of the possible future years
            m.yeer='m.yr'+alltrim(str(n))       && this year selected? user's
                                        selections in field named "m.yrN"
            if &yeer        && if user selected this year for output, write something in
                            Excel output file
                m.yr_txt='m.yr'+alltrim(str(n))+'_txt'      && create year as
                                    char. string in field named "m.yrN_txt"
                if p=m.num_plans    && if now writing data for last plan, write
                                plan name and carriage return/line feed character
                    =fputs(m.filhand,chr(9)+chr(9)+&yr_txt)  && write two
                                    "tab" characters first to get proper spacing
                else    && if more plans follow, write just the calendar year
                    p=p+1           && increment the "number of plans written"
                                    counter
                    =fwrite(m.filhand,chr(9)+chr(9)+&yr_txt)  && write
                                two "tab" characters first to get proper spacing
                endif
            endif
        endfor          && repeat above steps for each possible future year
    endif
    if typ='FS'         && currently, the only active option
*       && write detail line headers
        =fwrite(m.filhand,m.rcvr+chr(9)+m.sndr)         && write receiving school,
                                        "tab" character, then sending school
        for n=1 to m.num_plans      && write column headers once for each plan
            if n=m.num_plans    && if now writing data for last plan, write headers
                                and carriage return/line feed character
              =fputs(m.filhand,chr(9)+'# of Students'+chr(9)+'%')
              &&  write one "tab" character first to get proper spacing
```

```
                        else            && if more plans follow, write just the headers
                            =fwrite(m.filhand,chr(9)+'# of Students'+chr(9)+'%')
                            && write one "tab" character first to get proper spacing
                        endif
                endfor
        endif
        select tempfeed                 && switch to the output data file of feed rates and numbers
                                        created above
        goto top        && move to the top record
        m.first=.t.             && initialize a "first line of output for the receiving school" flag
        m.rcvr=alltrim(recv_schl)                       && initialize a field to hold the name of the
                                                        receiving school
        for p=1 to m.num_plans                  && create and initialize a field to accumulate totals
                                                for the receiving school for each plan
                m.tot_fld='m.tot'+alltrim(str(p))               && set up name of field for macro
                                                                substitution on next line
                &tot_fld=0              && initialize the field to 0
        endfor
*               && write detail data lines
        scan    && read the records in tempfeed.dbf sequentially
                if m.first              && if first line of output for the receiving school, write the
                                        receiving school's name
                        if typ="FS"
                                =fputs(m.filhand,recv_schl)             && write line with only
                                                                        receiving school's name
                        endif
                        m.first=.f.             && turn off the "first line" flag
                        m.num_lines=0           && initialize field to count number of sending
                                                schools written
                endif
*                       && write sending school name on detail line
                if typ="FS"
                        =fwrite(m.filhand,chr(9)+send_schl)             && write one "tab" character
                                                                        first to get proper spacing
                endif
                p=0
                for n=2 to 9    && cycle through each of the possible future years
                        m.yeer='m.yr'+alltrim(str(n))           && this year selected? user's
                                                                selections in field named "m.yrN"
                        if &yeer                && if user selected this year for output, write
                                                something in Excel output file
                                p=p+1           && increment the "number of plans written"
                                                counter
                                m.yr_txt='m.yr'+alltrim(str(n))+'_txt'          && create year as
                                                        char. string in field named "m.yrN_txt"
                                m.num_fld='fd_'+&yr_txt+'_nm'   && set up name of field to
```

```
                    hold number of students for macro substitution below
                m.pct_fld='fd_'+&yr_txt+'_pc'       && set up name of field to
                        hold feed percentage for macro substitution below
                m.tot_fld='m.tot'+alltrim(str(p))   && set up name of field to
                        hold school total for macro substitution below
*           && finish writing detail lines
*           && if now writing data for last plan, write # and % for sending school and
            carriage return/line feed character
                if p=m.num_plans
                    if typ="FS"
                        =fPUTS(m.filhand,chr(9)+
                            alltrim(str(ROUND(&num_fld,0)))+
                            chr(9)+alltrim(str(round(&pct_fld,1),5,1))
                            +'%')
                    endif
                else                        && if more plans follow, just write # and %
                                            for sending school
                    if typ="FS"
                        =fwrite(m.filhand,chr(9)+
                            alltrim(str(ROUND(&num_fld,0)))+
                            chr(9)+alltrim(str(round(&pct_fld,1),5,1))
                            +'%')
                    endif
                endif
                &tot_fld=&tot_fld+&num_fld          && accumulate total students
                                                    "sent" to receiving school
            endif
        endfor          && repeat above steps for each possible future year
        m.num_lines=m.num_lines+1           && increment "# of sending schools
                                            written" counter
        skip 1              && skip ahead one record to see if receiving school is
                            changing on next record
        if not alltrim(recv_schl)==m.rcvr or EOF()      && receiving school changes
                                                        or no more records in file
*           && write receiving school totals line
            if typ="FS"
                =fwrite(m.filhand,'Total'+chr(9))   && start with the word
                                                    "Total" and a "tab" character
                for p=1 to m.num_plans          && write totals for the
                                                receiving school for each plan
                    m.tot_fld='m.tot'+alltrim(str(p))   && set up name of
                            totals field for macro substitution below
                    if p=m.num_plans        && if now writing data for last plan,
                        write total and carriage return/line feed character
                        =fPUTS(m.filhand,chr(9)+
                            alltrim(str(ROUND(&tot_fld,0)))+chr(9)+
                              '100.0%')
                      else    && if more plans follow, just write total
                          =fwrite(m.filhand,chr(9)+
                              alltrim(str(ROUND(&tot_fld,0)))+chr(9)+
                              '100.0%')
                      endif
                  endfor      && repeat above steps for each plan
              endif
              for p=1 to m.num_plans    && reset all total fields for a new receiving
                                        school
                  m.tot_fld='m.tot'+alltrim(str(p))    && set up name of field for
                                                       macro substitution on next line
                  &tot_fld=0            && initialize the field to 0
              endfor          && repeat above steps for each plan
              m.first=.t.          && reset "first line of output" flag for the new
                                   receiving school
              m.rcvr=alltrim(recv_schl)    && store name of new receiving
                                           school
          endif
          skip -1      && skip back to original record after "look ahead" so scan:endscan
                       will work properly
      endscan          && move on to next tempfeed.dbf record, repeat above steps until
                       entire file has been processed
      =fclose(m.filhand)      && close Excel output file
      select tempfeed         && switch to tempfeed data file
      use                         && and close it
      select assignmt         && switch to current plan's data file
      set order to plan_id    && set sort order to planning polygon ID number
      goto m.rec              && move record pointer to the record on which it sat
                              before this procedure was executed
      save to fplnyrs.mem all like plan*    && save user's prompt screen settings to
                                            bring back next time
  *   && display message on screen to tell user the name of the output file
      wait window 'Data written to "'+alltrim(m.ufile)+'".  '+chr(13)+
          "Press any key to continue . . ."
  endif
  _CUROBJ=OBJNUM(M.go_to)      && put the screen's cursor on the "Go to" button
  return       && return control to the main Whiffer screen program

procedure prt_fdr          && prints feeder report
  m.go='Cancel'            && initialize user action button for prompt screen
  m.rept_text='Ready to print Feeder Report'    && initialize header for prompt screen
  for m=0 to 12        && initialize grade level check boxes for prompt screen
      m.gr_fld='m.g'+alltrim(str(m))    && set up field name for macro substitution
                                        on next line
      &gr_fld=.f.          && initialize each box as "unchecked"
```

```
endfor
m.choi=3                    && initial type of report to "% of Middle from each Elem"
do oldfdr.spr               && call the user prompt screen
if m.go#'Cancel'            && proceed if the user has not elected to cancel the process from
                            the prompt screen
     select assignmt        && switch to the current plan's data file
     m.rec=recno()          && record the number of the active record so program can return
                            to it at the end
     do case                && initialize field names, report titles, and sort sequences,
                            depending upon user's selected report
     case m.choi=6          &&'% High from each Middle'
          m.field='asg.hs_home'
          INDEX ON hs_home+ms_home TAG HFM_fdr ADDITIVE
          m.title='% of High from each Middle'
     case m.choi=5          &&'% High from each Elem'
          m.field='asg.hs_home'
          INDEX ON hs_home+es_home TAG HFE_fdr ADDITIVE
          m.title='% of High from each Elementary'
     case m.choi=4          &&'% Middle to each High'
          m.field='asg.ms_home'
          INDEX ON ms_home+hs_home TAG MTH_fdr ADDITIVE
          m.title='% of Middle to each High'
     case m.choi=3          &&'% Middle from each Elem'
          m.field='asg.ms_home'
          INDEX ON ms_home+es_home TAG MFE_fdr ADDITIVE
          m.title='% of Middle from each Elementary'
     case m.choi=2          &&'% Elem to each High'
          m.field='asg.es_home'
          INDEX ON es_home+hs_home TAG ETH_fdr ADDITIVE
          m.title='% of Elementary to each High'
     case m.choi=1          &&'% Elem to each Middle'
          m.field='asg.es_home'
          INDEX ON es_home+ms_home TAG ETM_fdr ADDITIVE
          m.title='% of Elementary to each Middle'
     otherwise                          && no user choice - do nothing and return
          wait window "No report selected - re-run report from menu."+chr(13)+;
               "Press any key to continue . . ."          && tell user why program is
                                                          doing nothing
          set order to plan_id          && set sort order to default
          goto m.rec                    && return to previously-selected record
          return            && return control to main Whiffer screen program
     endcase
     *    && the following SQL select command creates a cursor with one record per
          receiving school,
     *    && with the following structure:
     *    && field 1: the name of the receiving school
```

```
        *       && field 2: the sum of the user's selected HS grade levels' enrollment, across all
                polygons assigned to the receiving school
        *       && field 3: the sum of the user's selected MS grade levels' enrollment, across all
                polygons assigned to the receiving school
        *       && field 4: the sum of the user's selected ES grade levels' enrollment, across all
                polygons assigned to the receiving school
        *       && this data is pulled out of the current plan file and
        *       && the projections file as needed. the files are related to each other by planning
                polygon ID number
        *       && the resulting cursor is given the name "schtot"
        select alltrim(&field) as schl, ;
                iif(m.g12,sum(pp.gr12_pop),0)+;
                iif(m.g11,sum(pp.gr11_pop),0)+iif(m.g10,sum(pp.gr10_pop),0)+
                iif(m.g9,sum(pp.gr9_pop),0) as hs_tot, ;
                iif(m.g8,sum(pp.gr8_pop),0)+iif(m.g7,sum(pp.gr7_pop),0)+
                iif(m.g6,sum(pp.gr6_pop),0) as ms_tot, ;
                iif(m.g5,sum(pp.gr5_pop),0)+iif(m.g4,sum(pp.gr4_pop),0)+
                iif(m.g3,sum(pp.gr3_pop),0)+;
                iif(m.g2,sum(pp.gr2_pop),0)+iif(m.g1,sum(pp.gr1_pop),0)+
                iif(m.g0,sum(pp.gr0_pop),0) as es_tot ;
                from assignmt asg,prj pp into cursor schtot
                where pp.plan_id=asg.plan_id group by &field
        select schtot                       && make the cursor the active data file
        index on schl to temp.ndx           && sort the file by school name
        select assignmt                     && make the current plan the active data file
        m.fld=substr(m.field,5)             && store the "school name" field's name without
                                            the file name qualifier in "m.fld"
        set relation to alltrim(&fld) into schtot additive   && relate the current plan file to the
                                                            cursor by school name
        =prt_rept("oldfdr")         && call the general-purpose report printing
                                            procedure to print the report form "oldfdr.frx"
        set relation off into schtot  && remove relationship between plan data file and cursor
        set order to plan_id          && reset sort sequence to default in plan file
        goto m.rec                    && return to previously-selected record
    endif
    _CUROBJ=OBJNUM(M.go_to)           && set screen's cursor to the "Go to" button
    return                            && return control to the main Whiffer screen

procedure exp_farm      && exports FARM/MSA data for the Feasibility Study document in an
                        Excel-compatible format
    *       && formulate an export filename that contains the plan name and the school level
                being exported
    m.ufile=right(alltrim(m.pln_in_use),6)+alltrim(m.cur_lev)+".xls"      && use right 6 chars
                                                    of plan name and two-letter school level
    copy file hrdata to (ufile)       && copy "seed" file to export file
    m.filhand=fopen(m.ufile,2)        && open export file for output
```

```
=fchsize(m.filhand,0)                             && clear out the contents of the export file
*         && write headers to output file; write export title, plan name, and current date in first line
=fPUTS(m.filhand,'FARM/MSA Data'+chr(9)+m.pln_in_use+chr(9)+chr(9)+dtoc(date()))
=fPUTS(m.filhand,'')                              && skip a line
=fPUTS(m.filhand,'School Name'+chr(9)+'FARM'+chr(9)+'MSA-Read'+chr(9)+
          'MSA-Math')                             && write text headers for detail lines
select sum2                   && sum2 is the always-present cursor that contains the data displayed in
                              the scrolling list at bottom of main Whiffer screen
scan      && read the records in the cursor one school at a time
*                   && write the detail lines to the output file as data are read from the cursor
*                   && detail lines contain: school name, FARM %, MSA Reading pass rate, MSA
                              Math pass rate
          =fPUTS(m.filhand,schl+chr(9)+padl(alltrim(str(round(farm_pct/100,2),4,2)),4,' ')+
                    chr(9)+padl(alltrim(str(round(read_avg,2),4,2)),4,' ')+;
                    chr(9)+padl(alltrim(str(round(math_avg,2),4,2)),4,' '))
endscan             && continue to output detail lines until last record of cursor, i.e. last school, is
                    processed
goto top            && return record pointer to top record in cursor
select prj          && switch to the projections data file
m.popfld="m."+alltrim(m.cur_lev)+'_pop'                     && set up field name for macro
                                                                      substitution below
do case                       && set up text strings for macro substitution below
case alltrim(m.cur_lev)='ES'                                && user requests elementary school report
          m.grdtxt="gr0_pop+gr1_pop+gr2_pop+gr3_pop+gr4_pop+gr5_pop"  && text string
                                                  containing expression to use for calculating ES enrollment
          m.frmtxt="es_farm+ks_farm"                        && text string containing expression to use for
                                                            calculating ES FARM enrollment
case alltrim(m.cur_lev)='MS'                                && user requests middle school report
          m.grdtxt="gr6_pop+gr7_pop+gr8_pop"                          && text string containing expression to use
                                                                      for calculating MS enrollment
          m.frmtxt="ms_farm"                                && text string containing expression to use for
                                                            calculating ES FARM enrollment
case alltrim(m.cur_lev)='HS'                                && user requests high school report
          m.grdtxt="gr9_pop+gr10_pop+gr11_pop+gr12_pop"                         && text string containing
                                                            expression to use for calculating HS enrollment
          m.frmtxt="hs_farm"                                && text string containing expression to use for
                                                            calculating ES FARM enrollment
endcase
sum &grdtxt to m.lev_pop                          && calculate total enrollment for user-selected school level
sum &frmtxt to m.farm_pop                         && calculate total FARM enrollment for user-selected
                                                  school level
m.farm_pct=m.farm_pop/m.lev_pop                             && calculate overall FARM percentage for user-
                                                            selected school level
m.mtdbfld=alltrim(m.cur_lev)+'_take_ma'                               && set up database field name for number
                                                                      of MSA Math test takers
m.mtmfld="m."+alltrim(m.cur_lev)+'_tk_ma'                             && set up memvar field name for number of
```

```
                                                        MSA Math test takers
    sum &mtdbfld to &mtmfld             && calculate overall number of MSA Math test takers
    m.mpdbfld=alltrim(m.cur_lev)+'_pass_ma'             && set up database field name for number
                                                        of MSA Math test passers
    m.mpmfld="m."+alltrim(m.cur_lev)+'_ps_ma'           && set up memvar field name for number of
                                                        MSA Math test passers
    sum &mpdbfld to &mpmfld             && calculate overall number of MSA Math test passers
    m.rtdbfld=alltrim(m.cur_lev)+'_take_re'             && set up database field name for number
                                                        of MSA Reading test takers
    m.rtmfld="m."+alltrim(m.cur_lev)+'_tk_re'           && set up memvar field name for number of
                                                        MSA Reading test takers
    sum &rtdbfld to &rtmfld             && calculate overall number of MSA Reading test takers
    m.rpdbfld=alltrim(m.cur_lev)+'_pass_re'             && set up database field name for number
                                                        of MSA Reading test passers
    m.rpmfld="m."+alltrim(m.cur_lev)+'_ps_re'           && set up memvar field name for number of
                                                        MSA Reading test passers
    sum &rpdbfld to &rpmfld             && calculate overall number of MSA Reading test passers
    m.read_avg=&rpmfld/&rtmfld                          && calculate overall MSA Math pass rate
    m.math_avg=&mpmfld/&mtmfld                 && calculate overall MSA Reading pass rate
    =fPUTS(m.filhand,"")                       && write a blank line to output file
    *        && write summary line to output file; summary contains the overall totals calculated
             above
    =fPUTS(m.filhand,"System-wide total"+chr(9)+
             padl(alltrim(str(round(m.farm_pct,2),4,2)),4,' ')+chr(9)+;
             padl(alltrim(str(round(m.read_avg,2),4,2)),4,' ')+chr(9)+;
             padl(alltrim(str(round(m.math_avg,2),4,2)),4,' '))
    select assignmt                 && switch to currently-active plan file
    _CUROBJ=OBJNUM(M.go_to)                    && move screen's cursor to the "Go to" button
    =fclose(m.filhand)              && close the output file
    *        && notify user that processing is done and display the name of the output file on the
             screen
    wait window 'Data written to "'+alltrim(m.ufile)+'".  '+chr(13)+
             "Press any key to continue . . ."
    return          && return control to the main Whiffer screen program

procedure incl_k            && refreshes main Whiffer screen contents when user selects/deselects
                            "Incl K" check box
    *        && build appropriate text string for text color, per user's selection (red if K included,
             black otherwise)
    m.clr=iif(m.k_in,"COLOR RGB(255,0,0,,,,)","")            && (black text is default if no color
             specified)
    @ 8.0,87.5 clear to 9.077,120.667          && clear existing text for projected enrollment
    *        && redisplay appropriate projected enrollments, in appropriate color
    @ 8.0,87.5 SAY round(iif(m.k_in,prj.ksproj1+prj.esproj1,prj.esproj1),0) ;
             SIZE 1.000,4.000,0.000 FONT "Arial", 8 STYLE "T" &clr
    @ 8.0,91.667 SAY round(iif(m.k_in,prj.ksproj2+prj.esproj2,prj.esproj2),0) ;
```

```
            SIZE 1.000,4.000,0.000 FONT "Arial", 8 STYLE "T" &clr
    @ 8.0,95.833 SAY round(iif(m.k_in,prj.ksproj3+prj.esproj3,prj.esproj3),0) ;
            SIZE 1.000,4.000,0.000 FONT "Arial", 8 STYLE "T" &clr
    @ 8.0,100.000 SAY round(iif(m.k_in,prj.ksproj4+prj.esproj4,prj.esproj4),0) ;
            SIZE 1.000,4.000,0.000 FONT "Arial", 8 STYLE "T" &clr
    @ 8.0,104.167 SAY round(iif(m.k_in,prj.ksproj5+prj.esproj5,prj.esproj5),0) ;
            SIZE 1.000,4.000,0.000 FONT "Arial", 8 STYLE "T" &clr
    @ 8.0,108.333 SAY round(iif(m.k_in,prj.ksproj6+prj.esproj6,prj.esproj6),0) ;
            SIZE 1.000,4.000,0.000 FONT "Arial", 8 STYLE "T" &clr
    @ 8.0,112.500 SAY round(iif(m.k_in,prj.ksproj7+prj.esproj7,prj.esproj7),0) ;
            SIZE 1.000,4.000,0.000 FONT "Arial", 8 STYLE "T" &clr
    @ 8.0,116.667 SAY round(iif(m.k_in,prj.ksproj8+prj.esproj8,prj.esproj8),0) ;
            SIZE 1.000,4.000,0.000 FONT "Arial", 8 STYLE "T" &clr
    @ 13.385,90.667 clear to 15.399,95.333            && clear existing tewxt for FARM pop/pct
    *       && redisplay appropriate FARM enrollments and %, in appropriate color
    @ 13.385,91.5 SAY prj.es_farm+iif(m.k_in,prj.ks_farm,0) ;
            SIZE 1.000,3.833 FONT "Arial", 8 STYLE "T" &clr
    @ 14.385,90.833 SAY round(iif(m.k_in,iif(prj.esproj0+
            prj.ksproj0=0,0,(prj.es_farm+prj.ks_farm)*100/;
            (prj.esproj0+prj.ksproj0)),prj.esfrm_pct),0) SIZE 1.000,4.833 FONT "Arial", 8
            STYLE "T" &clr
    @ 16.077,47.5 clear to 17.5,51.333          && clear existing text for current enrollments
    *       && redisplay appropriate current enrollments, in appropriate color
    @ 16.077,47.500 SAY iif(m.k_in,prj.gr0_pop,0)+prj.gr1_pop+prj.gr2_pop+prj.gr3_pop+;
            prj.gr4_pop+prj.gr5_pop SIZE 1.000,3.667 FONT "Arial", 10 STYLE "BT" &clr
    *       && call the "get_stat" procedure to refresh the contents of the scrolling schools list at the
            bottom of the Whiffer screen
    =get_stat(m.cur_lev)
    save to k.mem all like k_in         && save user's setting of "include K" box
    return          && return control to the main Whiffer screen

procedure updtdesc      && updates the plan description
    *       && when the user selects the "Add/Modify Plan Description Text" option from the
            Maintenance menu
    m.go='Cancel'               && initialize user action button for prompt screen
    m.rec=recno()               && save the record number of the current record for later return
    m.desc=assignmt.desc        && copy the current plan description text to a memvar
    do updtdesc.spr             && display the screen where the description can be modified
    if m.go#'Cancel'            && if the user does not choose to cancel, proceed
    *       && replace the current description with the one entered by the user on the prompt screen
            replace all assignmt.desc with m.desc
            goto m.rec          && go back to the original record
            show gets           && refresh the screen with the contents of the original record
    endif
    return                      && return control to the main Whiffer screen program
```

```
procedure markem                    && marks moved polygons
    *       && when user selects the "Mark Moved Polygons for Mapping" option from the
            Maintenance menu
    *       && initialize prompt screen fields that hold plan names for future years
    m.plan2=m.cur_db
    m.plan3=m.cur_db
    m.plan4=m.cur_db
    m.plan5=m.cur_db
    m.plan6=m.cur_db
    m.plan7=m.cur_db
    if file('plnyrs.mem')           && if a saved settings file exists for this procedure, restore settings
                                    from it
            restore from plnyrs.mem additive
    endif
    m.go='Preview'                  && initialize user action button for prompt screen
    for n=2 to 7            && set up and initialize prompt screen fields for future years
            m.yrfld='M.year'+alltrim(str(n))    && set up calendar year text field for macro
                                                substitution below
            &yrfld=alltrim(str(year(date())-1+n))       && initialize calendar year field with proper
                                                        year value
            m.yeer='m.yr'+alltrim(str(n))       && set up year's selection check box field for
                                                macro susbstitution below
            m.yr_txt='m.yr'+alltrim(str(n))+'_txt'      && set up calendar year header field for
                                                        macro susbstitution below
            &yeer=.f.                           && initialize year's check box field as "unchecked"
            &yr_txt=alltrim(str(year(date())-1+n))      && store proper text in year header field
    endfor
    m.rept_text='Ready to mark moved polygons'          && initialize title of prompt screen
    do markprpt.spr                 && call the user prompt screen
    if m.go#'Cancel'                && if the user does not choose to cancel, proceed
            for m=2 to 7            && cycle through each possible future year
                    close databases         && close all data files currently open to start from a known
                                            environment
                    m.lpln='m.plan'+alltrim(str(m-1))   && field containing name of plan in effect
                                                        for previous year
                    m.plan='m.plan'+alltrim(str(m))     && field containing name of plan in effect
                                                        for this year
                    m.db=&plan              && copy this year's plan into a generically-named field
                    m.prvdb=iif(m=2,m.prjfile,&lpln)    && copy previous year's plan into a
                                                        generically-named field
                                                        && but use the projections file if currently
                                                        processing year 2
                    select 0                            && open a new work area
                    use &prvdb order plan_id exclusive alias pp         && open the previous year's
                                                                        plan data file
                    select 0                && open a new work area
```

```
*       && if the current and previous plans are the same, move on to the next year
*       && if the two years' plans are different, open the current plan data file in the new work
                area
        if m.db=m.prvdb                 && both plans are the same, therefore nothing to
                                        mark because nothing moved
                loop            && exit the "for" loop and return to the "for" command for
                                next iteration
        else    && there are two different plans selected by the user in consecutive years
                use &db exclusive alias asg             && open the current year's plan data
                                                        file and continue
        endif
        set relation to plan_id into pp         && relate the current plan to the previous
                                                one by planning polygon ID number
        replace all es_movd with .f.            && set ES "moved" flags to "not moved" on
                                                all polygon records
*       && if the same polygon is assigned to different ES in the two years' plans, reset "moved"
        flag to "moved"
        replace es_movd with .t. for alltrim(asg.es_home)#alltrim(pp.es_home)
        count for es_movd to m.es_moved                 && store the number of polygons
                                                        moved at the ES level
        replace all ms_movd with .f.            && set MS "moved" flags to "not moved"
                                                on all polygon records
*       && if the same polygon is assigned to different MS in the two years' plans, reset
        "moved" flag to "moved"
        replace ms_movd with .t. for alltrim(asg.ms_home)#alltrim(pp.ms_home)
        count for ms_movd to m.ms_moved                 && store the number of polygons
                                                        moved at the MS level
        replace all hs_movd with .f.            && set HS "moved" flags to "not moved"
                                                on all polygon records
*       && if the same polygon is assigned to different HS in the two years' plans, reset "moved"
        flag to "moved"
        replace hs_movd with .t. for alltrim(asg.hs_home)#alltrim(pp.hs_home)
        count for hs_movd to m.hs_moved                 && store the number of polygons
                                                        moved at the MS level
*       && display the number of moves at each school level to user
        wait window alltrim(str(m.es_moved))+
                ' polygons moved at the ES level,'+chr(13)+;
                alltrim(str(m.ms_moved))+
                ' polygons moved at the MS level, and'+chr(13)+;
                alltrim(str(m.hs_moved))+
                ' polygons moved at the HS level in '+alltrim(&plan)+chr(13)+;
                "Press any key to continue . . ."
        use             && close the current plan data file
        select pp       && switch to previous plan's work area
        use             && close the previous plan data file
    endfor          && repeat the above steps for each possible future year
```

```
        wait window "Done - press any key to continue . . ."      && let user know process is
                                                                     finished
        close databases              && close all data files
        =setup()        && call the "setup" procedure to restore the default data environment
    *   && re-establish the browse window in the lower half of the main Whiffer screen
        define window browind from 20.5,1.4 to 39,118.15 nofloat close none
            font 'MS Sans Serif',8 in window polyback
    *   && call the "get_stat" procedure to refresh the contents of the scrolling schools list at the
        bottom of the Whiffer screen
        =get_stat(m.cur_lev)
    endif
    save to plnyrs.mem all like plan*   && save the user's settings on the prompt screen
                                        for next time
    _CUROBJ=OBJNUM(M.go_to)                 && move the screen's cursor to the "Go to" button
    return          && return control to the main Whiffer screen program

procedure new_cip       && clones the currently-selected CIP as a new CIP file and makes the
                        new one the current CIP selection
    *                   && when user selects the "Create New School Capacity Table" option
                        from the Maintenance menu
    m.newname='new_cip'         && initialize "new name" field for prompt screen
    m.go='Cancel'               && initialize user action button for prompt screen
    do newcip.spr               && call the user prompt screen
    if m.go#'Cancel'            && if the user does not choose to cancel, proceed
        select schools          && select the current CIP data file
        m.dot=rat(".",dbf())    && store the position of the last "period" character in the current
                                CIP's full file name
        m.last_slash=rat("\",dbf())         && store the position of the last "backslash"
                                            character in the current CIP's file name
    *   && extract the data path from the full file name as the text from position 1 to the position
        of the last backslash
        m.path=substr(dbf(),1,m.last_slash)
    *   && extract the old CIP file name from the full file name as the text from position 1 to the
        position of the last period
        m.oldname=substr(dbf(),1,m.dot)
        copy structure with production to (newname)     && create a new file with the user's
                                                    specified name and identical structure
        m.newname=m.path+alltrim(m.newname)+'.'     && add the data path and a "period"
                                                    to the new CIP file name
    *   && copy the old CIP's shape file components to new shape files with the new CIP name
        copy file m.oldname+'shp' to m.newname+'shp'
        copy file m.oldname+'shx' to m.newname+'shx'
        m.newname=alltrim(m.newname)+'dbf'          && assemble the name of the .dbf
                                                file for the new CIP for macro substitution
        use (newname) exclusive             && open the new CIP data file
        m.oldname=m.oldname+'dbf'           && assemble the name of the .dbf file for the old
```

```
                                                    CIP for macro substitution
            append from (oldname)                   && copy all records in the old CIP to the new CIP
            m.cur_cip=dbf()                         && reset the contents of the memvar that keeps
                                                    track of the currently-selected CIP
            use                             && close the new CIP
            =get_schl()             && call the "get_schl" procedure to set up the new school and CIP
                                    files in the default data environment
            select assignmt         && switch to the current plan data file
            =showgets()             && call the "showgets" procedure to refresh the contents of the
                                    main Whiffer screen
    endif
    _CUROBJ=OBJNUM(M.go_to)                 && move the screen's cursor to the "Go to" button
    return

procedure new_plan          && clones the currently-selected plan as a new plan file and makes the
                            new one the current plan selection
    *                       && when user selects the "Start a New Plan From This One" button on the
                            main Whiffer screen
    m.newname='new_plan'            && initialize the "new plan name" field for the prompt screen
    m.desc='Created '+dtoc(date())          && initialize the "plan description" field for the prompt
                                            screen
    m.go='Cancel'                   && initialize the user action button for thr prompt screen
    do newprpt.spr                  && call the prompt screen
    if m.go#'Cancel'        && if the user does not choose to cancel, proceed
            m.pln_in_use=m.newname                  && reset the memvar containing the name of the
                                                    currently-selected plan
            m.dot=rat(".",dbf())            && store the position of the last "period" character in the
                                            current plan's file name
            m.last_slash=rat("\",dbf())             && store the position of the last "backslash"
                                                    character in the current plan's file name
            m.path=substr(dbf(),1,m.last_slash)             && store the data path i.e. the text between
                                                            the first char. and the last backslash
            m.oldname=substr(dbf(),1,m.dot)                 && store the file name i.e. the text between
                                                            the first char. and the last period
            copy structure with production to (newname)             && copy the field and index
                                                    structure of the current plan to a new plan file
            m.newname=m.path+alltrim(m.newname)+'.'                 && add the data path and a "period"
                                                                    to the new file name
            copy file m.oldname+'shp' to m.newname+'shp'            && copy the shape file components
                                                                    to the new plan
            copy file m.oldname+'shx' to m.newname+'shx'            && copy the shape file components
                                                                    to the new plan
            m.newname=alltrim(m.newname)+'dbf'                      && append the text "dbf" to the
                                                                    name of the new plan
            use (newname) ALIAS assignmt exclusive                  && close the old plan and open the
                                                                    new one in the same work area
            index on plan_id tag plan_id            && recreate the planning polygon ID number index
                                                    for the new plan file
            m.oldname=m.oldname+'dbf'               && add the "dbf" file extension to the old plan's file
                                                    name
            append from (oldname)                   && copy all records in the old plan to the new
                                                    plan's ".dbf" file
            replace all desc with m.desc            && replace the contents of the new plan's
                                                    description field with the new description
            set relation to plan_id into prj        && relate the new plan to the projections file by
                                                    planning polygon ID number
            goto bottom             && go to the record with the highest ID number
            m.botrec=recno()        && store the record number of this record in a memvar that keeps
                                    track of last record on file
            goto top                && go to the record with the lowest ID number
            M.TOPREC=recno()                && store the record number of this record in a memvar that
                                            keeps track of first record on file
            m.cur_db=dbf()          && reset the "current plan" environment memvar
            save to db.mem all like cur_db          && save the current plan's name for next start up of
                                                    Whiffer
            =showgets()             && call the "showgets" procedure to refresh the contents of the
                                    main Whiffer screen
    *       && call the "get_stat" procedure to refresh the contents of the scrolling schools list at the
            bottom of the Whiffer screen
            =get_stat(m.cur_lev)
    endif
    _CUROBJ=OBJNUM(M.go_to)                         && move the screen's cursor to the "Go to" button
    return

procedure get_stat          && refreshes the contents of the scrolling schools list at the bottom of the
                            main Whiffer screen
    parameters lev          && the calling procedure specifies which school level the user desires to
                            refresh in the browse window
    m.cur_lev=lev           && change the value of the Whiffer environment memvar to reflect the
                            user's choice
    for n=2 to 13           && set up fields for later macro substitution and initialize them
    *       && create a field, one per future year, to hold the name of the school capacity field
            m.cap='m.c'+alltrim(str(n))
    *       && store the name of the appropriate capacity field and add file name qualifier
            &cap='s.'+iif(lev='ES' and m.k_in,'kcap','cap')+alltrim(str(n))
    *       && create a field to hold enrollment projections
            m.pop='m.p'+alltrim(str(n))
    *       && store the name of the appropriate projection field and add file name qualifier
            &pop='pp.'+alltrim(m.lev)+'proj'+alltrim(str(n))
    *       && create a field to hold kindergarten enrollment projections
            m.kpop='m.k'+alltrim(str(n))
    *       && store the name of the appropriate kindergarten projection field and add file name
```

```
        qualifier
        if lev='ES' and m.k_in          && need K proj field only if user has requested ES
                &kpop='pp.'+'ksproj'+alltrim(str(n))
        else
                &kpop='0'    && leave empty if not doing ES
        endif
endfor          && repeat above steps for each future year displayed in browse window
do case                 && set up remaining fields needed for later macro substitution
case lev='HS'           && user requested HS
        m.mps='pp.hs_pass_ma'   && field name for HS MSA Math passers
        m.mtk='pp.hs_take_ma'   && field name for HS MSA Math takers
        m.rps='pp.hs_pass_re'   && field name for HS MSA Reading passers
        m.rtk='pp.hs_take_re'   && field name for HS MSA Reading takers
        m.farm='pp.hs_farm'             && field name for HS FARM enrollment
        m.tpop='pp.gr9_pop+pp.gr10_pop+pp.gr11_pop+pp.gr12_pop'  && expression for
                                                calculating total HS enrollment
case lev='MS'           && user requested MS
        m.mps='pp.ms_pass_ma'   && field name for MS MSA Math passers
        m.mtk='pp.ms_take_ma'   && field name for MS MSA Math takers
        m.rps='pp.ms_pass_re'   && field name for MS MSA Reading passers
        m.rtk='pp.ms_take_re'   && field name for MS MSA Reading takers
        m.farm='pp.ms_farm'     && field name for MS FARM enrollment
        m.tpop='pp.gr6_pop+pp.gr7_pop+pp.gr8_pop'   && expression for calculating total
                                                MS enrollment
case lev='ES'   && user requested ES
        m.mps='pp.es_pass_ma'   && field name for ES MSA Math passers
        m.mtk='pp.es_take_ma'   && field name for ES MSA Math takers
        m.rps='pp.es_pass_re'   && field name for ES MSA Reading passers
        m.rtk='pp.es_take_re'   && field name for ES MSA Reading takers
        m.farm='pp.es_farm'     && field name for ES FARM enrollment
*       && expression for calculating total ES enrollment
m.tpop='pp.gr5_pop+pp.gr4_pop+pp.gr3_pop+pp.gr2_pop+pp.gr1_pop+
        iif(m.k_in,pp.gr0_pop,0)'
endcase
m.kfarm='pp.ks_farm'            && field name for grade K FARM enrollment
m.feeld='asg.'+lev+'_home'      && name of field containing school name
*       && the following SQL select command creates a cursor with one record per school of the
        user's selected level,
*       && with the following structure:
*       && field 1: the name of the school
*       && field 2: the school's FARM percentage
*       && field 3: the school's MSA Math test pass percentage
*       && field 4: the school's MSA Reading test pass percentage
*       && This data is pulled out of the current plan file, the schools file, and
*       && the projections file as needed. The files are related to each other by planning
        polygon ID number
```

```
*       && and school name.  The resulting cursor is given the name "sum2", and
*       && is kept open in a work area the entire time that the Whiffer is operating
select trim(&feeld) as schl,100*((sum(&farm)+iif(lev='ES' and m.k_in,sum(&kfarm),
        0))/sum(&tpop)) as farm_pct,;
        sum(&mps)/sum(&mtk) as math_avg,sum(&rps)/sum(&rtk) as read_avg ;
        from schools s,assignmt asg,prj pp into cursor sum2 ;
        where pp.plan_id=asg.plan_id and trim(&feeld)=trim(s.schl_name) ;
        group by &feeld
*       && the following SQL select command creates a cursor with one record per school of the
        user's selected level,
*       && with the following structure:
*       && field 1: the name of the school
*       && field 2: the current enrollment of the school
*       && field 3: the current FARM enrollment of the school
*       && fields 4 through 15: future projected enrollments of the school
*       && field 16: the school's FARM percentage
*       && field 17: the school's number of MSA Math test passers
*       && field 18: the school's number of MSA Math test takers
*       && field 19: the school's MSA Math test pass percentage
*       && field 20: the school's number of MSA Reading test passers
*       && field 21: the school's number of MSA Reading test takers
*       && field 22: the school's MSA Reading test pass percentage
*       && This data is pulled out of the current plan file, the schools file, and
*       && the projections file as needed. The files are related to each other by planning polygon
        ID number
*       && and school name.  The resulting cursor is given the name "sumcur", and
*       && is kept open in a work area the entire time that the Whiffer is operating
select count(asg.plan_id) as num_poly,;
        trim(&feeld) as schl,;
        sum(pp.studnt_pop) as num_stud,;
        sum(&farm)+iif(lev='ES' and m.k_in,sum(&kfarm),0) as num_farm,;
        round(100*(sum(&p2+&k2))/&c2,1) as u1,;
        round(100*(sum(&p3+&k3))/&c3,1) as u2,;
        round(100*(sum(&p4+&k4))/&c4,1) as u3,;
        round(100*(sum(&p5+&k5))/&c5,1) as u4,;
        round(100*(sum(&p6+&k6))/&c6,1) as u5,;
        round(100*(sum(&p7+&k7))/&c7,1) as u6,;
        round(100*(sum(&p8+&k8))/&c8,1) as u7,;
        round(100*(sum(&p9+&k9))/&c9,1) as u8,;
        round(100*(sum(&p10+&k10))/&c10,1) as u9,;
        round(100*(sum(&p11+&k11))/&c11,1) as u10,;
        round(100*(sum(&p12+&k12))/&c12,1) as u11,;
        round(100*(sum(&p13+&k13))/&c13,1) as u12,;
        100*((sum(&farm)+iif(lev='ES' and m.k_in,
        sum(&kfarm),0))/sum(&tpop)) as farm_pct,;
        sum(&mps) as pas_math,;
        sum(&mtk) as tak_math,;
        sum(&mps)/sum(&mtk) as math_avg,;
        sum(&rps) as pas_read,;
        sum(&rtk) as tak_read,;
        sum(&rps)/sum(&rtk) as read_avg ;
        from schools s,assignmt asg,prj pp into cursor sumcur ;
        where pp.plan_id=asg.plan_id and trim(&feeld)=trim(s.schl_name) ;
        group by &feeld
    select sumcur        && make the "sumcur" cursor the active data file
    *       && activate a browse list of the cursor's records in a window previously established at
            the bottom of the main Whiffer screen
    browse fields pol=padl(alltrim(str(num_poly)),5,' ') :6 :H='#Polys',;
                schl :20 :H='SCHOOL',;
                free=padl(alltrim(str(farm_pct,4,1)),5,' ')+'%' :7 :H='FARM',;
                msa_r=padl(alltrim(str(read_avg*100,4,1)),5,' ')+'%' :8 :H='MSA R',;
                msa_m=padl(alltrim(str(math_avg*100,4,1)),5,' ')+'%' :8
                    :H='MSA M',;
                u1 :5 :H=str(m.yr+1,4),;
                u2 :5 :H=str(m.yr+2,4),;
                u3 :5 :H=str(m.yr+3,4),;
                u4 :5 :H=str(m.yr+4,4),;
                u5 :5 :H=str(m.yr+5,4),;
                u6 :5 :H=str(m.yr+6,4),;
                u7 :5 :H=str(m.yr+7,4),;
                u8 :5 :H=str(m.yr+8,4),;
                u9 :5 :H=str(m.yr+9,4),;
                u10 :5 :H=str(m.yr+10,4),;
                u11 :5 :H=str(m.yr+11,4),;
                u12 :5 :H=str(m.yr+12,4) ;
        title 'School Demographics and Projected Utilization for This Plan'
            window browind in window polyback nowait save ;
        noedit nodelete noappend nomenu
    select assignmt                 && switch to currenlty-selected plan data file
    _CUROBJ=OBJNUM(M.go_to)         && move Whiffer screen's cursor to the "Go to" button
    save to lev.mem all like cur_lev    && save the user's settings (i.e. which school level to
                                    display in browse) for future use
    return          && return to calling procedure

procedure reset         && sets all school assignments in the currently-selected plan back to the
                        projection file's school assignments
                        && when user selects the "Revert Displayed Plan to Curr. Assignments"
                        option from the Maintenance menu
    select assignmt             && switch to the current plan's data file
    m.rec=recno()               && save the record number of the polygon that is currently
                                displayed on the Whiffer screen
    *       && replace the current plan's schools with the schools from the projection file
    *       && the projections file is already linked to the plan file by ID number; the projection file
            has the current school
    *       && assignments embedded in each polygon's record
    replace all hs_home with prj.hs_home,ms_home with prj.ms_home ,
            es_home with prj.es_home
    goto m.rec              && return to the record that is displayed
    =showgets()             && refresh the contents of the main Whiffer screen
    *       && call the "get_stat" procedure to refresh the contents of the scrolling schools list at the
            bottom of the Whiffer screen
    =get_stat(m.cur_lev)
    return

procedure pop_phas          && opens a pop-up list of comments when the user clicks on the data
                            entry field next to the "Phasing" header
    *                       && and updates the contents of the "phasing" field in the current plan file
                            when the user selects an entry
    *                       && from the pop-up list
    declare phas_ara(1)             && initialize a memory array to hold comments
    store '(table is empty)' to phas_ara[1]             && put default text entry in array in case
                                                        none load below
    select text from phs_text into array phas_ara       && load all comments from current phasing
                                                        text entries into array
    =asort(phas_ara,1)              && sort array contents alphabetically
    *       && define a window in which to open the browse list
    define window list at 0,0 size 10,51 system ;
            title 'dbl-click=select;esc=cancel' font 'MS Sans Serif',10 in window (woutput())
    move window list center                 && center the window on the screen
    activate window list            && make the window active for user selection
    if not empty(phaseout) .and. ascan(phas_ara,phaseout)>0         && if contents of phasing
                                                        field exist in array, highlight them in pop-up
            m.chce=asubscript(phas_ara,ascan(phas_ara,phaseout,1),1)
    else    && if field is currently empty, or no match found, highlight the first entry in pop-up list
            M.chce=1
    endif
    *       && open the pop-up list on the screen and allow user to select entries
    @0,0 get m.chce from phas_ara picture '@&T' default 1 range 1
            size 10,30 font 'MS Sans Serif',10
    read    && store the user-selected entry when the pop-up is closed
    release window list             && close the browse window
    if readkey()=12 or readkey()=268 or readkey()=270       && if the user cancels via Esc key or
                                                            just closes window
            * do nothing *
    else            && closing pop-up by any other means assumed to mean "save changes"
            replace phaseout with alltrim(phas_ara[m.chce,1])       && replace current contents
                                                        of "phasing" field with user-selected entry
    endif
    _curobj=objnum(m.go_to)          && move screen's cursor to "Go to" button
    show get phaseout     && refresh contents of phasing field displayed on the Whiffer screen
    return                           && return control to the main Whiffer screen program

procedure pop_when       && opens a pop-up list of schools when the user clicks on one of the data
                         entry field under the school headers
    *                    && and updates the contents of the appropriate field in the current plan
                         file when the user selects an entry
    *                    && from the pop-up list
    parameters fld       && the name of the data entry field the user is currently attempting to
                         update
    *                    && this value is passed to this procedure when the procedure is called
                         from the "When" code snippet
    *                    && of one of the three school fields on the main Whiffer screen
    m.cur_lev=' '+upper(left(fld,2))      && temporarily store the school level as derived from the
                         field name, with a space char. in front
    do case              && check for conditions that require a user warning:
    *       && if the polygon being moved contains the school, do not allow school field to be
            changed
    case (alltrim(m.cur_lev)='ES' and es_loc) or
            (alltrim(m.cur_lev)='MS' and ms_loc) or
            (alltrim(m.cur_lev)='HS' and hs_loc)
    *       && notify user that he has messed up
            wait window "This polygon contains the school you are changing!"+chr(13)+;
                         "IT CANNOT BE REDISTRICTED, AS THE WHIFFER
                         CURRENTLY OPERATES"+chr(13)+;
                         "Press any key to continue . . ."
            m.cur_lev=alltrim(m.cur_lev)         && restore original value of m.cur_lev
            return       && don't open pop-up and return control to the main Whiffer screen program
    *       && if the polygon is part of the sending school's walk area, warn the user but allow them
            to proceed
    case (alltrim(m.cur_lev)='ES' and alltrim(in_es_walk)$'PW') or
            (alltrim(m.cur_lev)='MS' and alltrim(in_ms_walk)$'PW') or ;
            (alltrim(m.cur_lev)='HS' and alltrim(in_hs_walk)$'PW')
    *               && warn user that he may be messing up
            wait window "This polygon is a part of the walk area for the school you are
                    changing!"+chr(13)+;
                    "Press any key to continue . . ."
    *               && program continues after "endcase"
    endcase
    *       && if the polygon was previously redistricted, warn the user but allow them to proceed
    *       && period of reckoning is 5 yrs for ES, 2 yrs for MS, and 3 yrs for HS
    if (alltrim(m.cur_lev)='ES' and lst_es_red>0 and lst_es_red+5>year(date())) or ;
            (alltrim(m.cur_lev)='MS' and lst_ms_red>0 and lst_ms_red+2>year(date())) or ;
            (alltrim(m.cur_lev)='HS' and lst_hs_red>0 and lst_hs_red+3>year(date()))
    *               && warn user that he may be messing up
```

```
        wait window "This polygon was redistricted previously in "+
                iif(alltrim(m.cur_lev)='ES',alltrim(str(lst_es_red)),;
                iif(alltrim(m.cur_lev)='MS',alltrim(str(lst_ms_red)),
                alltrim(str(lst_hs_red))))+"!"+;
                chr(13)+"Press any key to continue . . ."
endif
declare schl_ara(1)             && initialize an array to hold school names
store '(table is empty)' to schl_ara[1]         && store a default value in case nothing loads from
                                                file on next line
*       && load array with names of appropriate level schools
select schl_name from schools into array schl_ara having m.cur_lev$schl_name
=asort(schl_ara,1)              && sort array entries alphabetically
m.width=min(fsize('&fld')+3,101)                && set width of pop-up window to smallest of field
                                                length or 101 characters
*       && define the window in which to open the pop-up
define window list at 0,0 size 10,30 system ;
        title 'dbl-click=select;esc=cancel' font 'MS Sans Serif',10 in window (woutput())
move window list center                 && center the pop-up window on the screen
activate window list    && make the pop-up window the active control surface
*       && if there is already an entry in the school field and it matches an entry in the pop-up
        list
if not empty(&fld) .and. ascan(schl_ara,&fld)>0
        m.chce=asubscript(schl_ara,ascan(schl_ara,&fld,1),1)    && highlight the matching
                                                                entry in the pop-up list
else    && no entry or entry doesn't match
        M.chce=1        && highlight the first entry in the pop-up list
endif
*       && open the pop-up list for user selection
@0,0 get m.chce from schl_ara picture '@&T' default 1 range 1 size 10,30
        font 'MS Sans Serif',10
read    && store the user-selected entry when the pop-up is closed
release window list             && close the browse window
if readkey()=12 or readkey()=268 or readkey()=270       && if the user cancels via Esc key or
                                                        just closes window
        * do nothing *
else            && closing pop-up by any other means assumed to mean "save changes"
        replace &fld with alltrim(schl_ara[m.chce,1])   && replace current contents of
                                                assigned school field with user-selected entry
*       && the following do case:endcase segment clears or resets the contents of the polygon's
        data field that indicates
*       && which school's walk area, if any, the polygon lies within.  For example, if the
        contents of the "walk_es" field match the
*       && newly-assigned ES school, then the contents of the "old_wlk_es" field ("W" or "P")
        are copied to the "in_es_walk" field.
*       && If the newly-assigned school doesn't match the stored walk area school, the
        "in_es_wlk" field is cleared.
```

```
*         && Similar processes are followed for MS and HS fields.
          do case
          case alltrim(m.cur_lev)='ES'
                    replace in_es_walk with
                              iif(alltrim(walk_es)==alltrim(es_home),old_wlk_es,'')
          case alltrim(m.cur_lev)='MS'
                    replace in_ms_walk with
                              iif(alltrim(walk_ms)==alltrim(ms_home),old_wlk_ms,'')
          case alltrim(m.cur_lev)='HS'
                    replace in_hs_walk with
                              iif(alltrim(walk_hs)==alltrim(hs_home),old_wlk_hs,'')
          endcase
endif
m.cur_lev=alltrim(m.cur_lev)        && restore the value of m.cur_lev to its former format
*         && call the "get_stat" procedure to refresh the contents of the scrolling schools list at the
          bottom of the Whiffer screen
=get_stat(m.cur_lev)
_curobj=objnum(m.go_to)             && move Whiffer screen's cursor to the "Go to" button
m.fld='assignmt.'+m.fld             && set up field name for macro substitution on next line
show get &fld             && refresh contents of school assignment field on screen
DO CASE                   && refresh contents of appropriate flags related to school assignments
case alltrim(m.cur_lev)='HS'        && a high school assignment was changed
          @ 2.6,22.9 clear to 3.7,40          && clear old values from the display area
*         && display current value for last year redistricted at this level
          @ 2.615,23.000 SAY lst_hs_red SIZE 1.000,5.600 FONT "Arial", 8 STYLE "BT"
                    PICTURE "@Z" COLOR RGB(255,0,0,,,,)
*         && display current value for "polygon contains the school" field
          @ 2.615,29.167 SAY iif(hs_loc,'Hm','') SIZE 1.000,4.600 FONT "Arial", 8
                    STYLE "BT" PICTURE "@I" COLOR RGB(255,0,0,,,,)
*         && display current value for "polygon contains the walk area" field
          @ 2.615,34.500 SAY iif(in_hs_walk='W','Wlk',iif(in_hs_walk='P','PtW','')) ;
                    SIZE 1.000,5.000 FONT "Arial", 8 STYLE "BT" PICTURE "@I"
                    COLOR RGB(255,0,0,,,,)
case alltrim(m.cur_lev)='MS'        && a middle school assignment was changed
          @ 2.6,56.9 clear to 3.7,75          && clear old values from the display area
*         && display current value for last year redistricted at this level
          @ 2.615,57.000 SAY lst_ms_red SIZE 1.000,5.600 FONT "Arial", 8 STYLE "BT"
                    PICTURE "@Z" COLOR RGB(255,0,0,,,,)
*         && display current value for "polygon contains the school" field
          @ 2.615,63.167 SAY iif(ms_loc,'Hm','') SIZE 1.000,4.600 FONT "Arial", 8
                    STYLE "BT" PICTURE "@I" COLOR RGB(255,0,0,,,,)
*         && display current value for "polygon contains the walk area" field
          @ 2.615,68.5 SAY iif(in_ms_walk='W','Wlk',iif(in_ms_walk='P','PtW','')) ;
                    SIZE 1.000,5.000 FONT "Arial", 8 STYLE "BT" PICTURE "@I"
                    COLOR RGB(255,0,0,,,,)
case alltrim(m.cur_lev)='ES'        && an elementary school assignment was changed
```

```
            @ 2.6,86.8 clear to 3.7,105         && clear old values from the display area
    *       && display current value for last year redistricted at this level
            @ 2.615,86.833 SAY lst_es_red SIZE 1.000,5.600 FONT "Arial", 8 STYLE "BT"
                    PICTURE "@Z" COLOR RGB(255,0,0,,,,)
    *       && display current value for "polygon contains the school" field
            @ 2.615,93 SAY iif(es_loc,'Hm','') SIZE 1.000,4.600 FONT "Arial", 8 STYLE "BT"
                    PICTURE "@I" COLOR RGB(255,0,0,,,,)
    *       && display current value for "polygon contains the walk area" field
            @ 2.615,98.33 SAY iif(in_es_walk='W','Wlk',iif(in_es_walk='P','PtW','')) ;
                    SIZE 1.000,5.000 FONT "Arial", 8 STYLE "BT" PICTURE "@I"
                    COLOR RGB(255,0,0,,,,)
    endcase
    return          && return control to the main Whiffer screen program

procedure prt_ppsm                  && prints planning polygon summary reports
    parameters lev          && school level requested by user, from the option selected on the
                            "Reports" menu
    m.go='Cancel'           && initialize value of user action button for the prompt screen
    m.sum='Yes'             && initialize value of the "summary only" control for the prompt screen
    m.dem='Yes'  && initialize value of the "include demographics" control for the prompt screen
    m.rept_text='Ready to print Planning Polygon Summary by '+alltrim(lev)      && compose
                                                                    title for prompt screen
    do pol_prpt.spr         && call the prompt screen
    if m.go#'Cancel'        && if the user has not elected to cancel the process on the prompt screen,
                            continue
            select assignmt         && make sure the currently-selected plan is the active data file
            m.rec=recno()           && save the reord number of the polygon whose data is displayed
    *       && recreate the relevant indexes
            INDEX ON es_home+STR(plan_id) TAG Es ADDITIVE
            INDEX ON ms_home+STR(plan_id) TAG Ms ADDITIVE
            INDEX ON hs_home+STR(plan_id) TAG Hs ADDITIVE
            set order to &lev       && sort the plan file according to the user's requested report
            =prt_rept("planxsch"+iif(m.sum='Yes',' summary',''))   && call the general-purpose
                                    report printing procedure to print report form "planxsch.frx"
            set order to plan_id    && restore sort sequence to planning polygon ID number
            goto m.rec              && reset record pointer to record whose data is displayed
    endif
    _CUROBJ=OBJNUM(M.go_to)             && move Whiffer screen's cursor to "Go to" button
    return                          && return control to the main Whiffer screen program

procedure mntphs            && opens the contents of the phasing text file for editing in a browse
                            window
    *                       && when user selects the "Set up Phase-In Text Entries" option from the
                            Maintenance menu
    if used('phs_text')             && if phasing text file is already in use
            select phs_text         && make it the active data file
    else
            select 0                    && otherwise, open a new work area
    endif
    use phs_text exclusive          && open or re-open the pahsing text file for exclusive use by user
    browse                  && display its contents in a simple browse window for editing
    use                     && close the phasing text data file when the user is done editing
    return          && return control to the main Whiffer screen program

procedure mntschl           && opens the contents of the currently-selected CIP file for editing in a
                            browse window
    parameters use
    *       && "use" parameter indicates how this procedure is to be used -
    *       &&      "upd" means user selected the "Modify School Capacities" option from the
                    Maintenance menu
    *       &&      "prt" means user selected the "Print School Capacity Table" option from the
                    Reports menu
    for n=1 to 13   && set up browse window's headers for each calendar year
            m.hdr='m.h'+alltrim(str(n))             && one set for "without K" capacity
            &hdr=alltrim(str(m.yr-1+n))
            m.khdr='m.k'+alltrim(str(n))            && one set for "with K" capacity
            &khdr='K'+alltrim(str(m.yr-1+n))
    endfor
    if use='upd'    && i.e. procedure was called to allow user to modify capacities
            *       && close all instances of CIP data file currently in use
            select his
            use
            select mis
            use
            select els
            use
            select schools
            use (cur_cip) order level exclusive             && open the currently-selected CIP data file
                                                            for exclusive use
            *       && set up a window in which to open the browse list
            define window browsit from 0,0 to 40,85 float grow close none font 'MS Sans Serif',8
            *       && display CIP file's contents in a formatted browse window for editing
            browse fields schl_name :16,cap1 :4 :H=m.h1,cap2 :4 :H=m.h2,
                    cap3 :4 :H=m.h3,cap4 :4 :H=m.h4,;
                    cap5 :4 :H=m.h5,cap6 :4 :H=m.h6,cap7 :4 :H=m.h7,cap8 :4 :H=m.h8,;
                    cap9 :4 :H=m.h9,cap10 :4 :H=m.h10,cap11 :4 :H=m.h11,
                    cap12 :4 :H=m.h12,cap13 :4 :H=m.h13,;
                    kcap1 :6 :H=m.k1,kcap2 :6 :H=m.k2,kcap3 :6 :H=m.k3,kcap4 :6 :H=m.k4,;
                    kcap5 :6 :H=m.k5,kcap6 :6 :H=m.k6,kcap7 :6 :H=m.k7,kcap8 :6 :H=m.k8,;
                    kcap9 :6 :H=m.k9,kcap10 :6 :H=m.k10,kcap11 :6 :H=m.k11,
                    kcap12 :6 :H=m.k12,kcap13 :6 :H=m.k13 ;
                    noappend nodelete title upper(alltrim(m.cip_in_use))+
                " SCHOOL CAPACITIES on SEPT 30th - Press Ctrl-End to save;
                 Esc to cancel" ;
                window browsit
            use (cur_cip) ALIAS schools ORDER TAG "name"        && re-open CIP as "schools"
            select 0                    && open a new work area
            *           && open the CIP data file as "his", filtered to show HS data only
            USE (cur_cip) again ALIAS his ORDER TAG "hs_only"
            SELECT 0                    && open a new work area
            *           && open the CIP data file as "mis", filtered to show MS data only
            USE (cur_cip) again ALIAS mis ORDER TAG "ms_only"
            SELECT 0                    && open a new work area
            *           && open the CIP data file as "els", filtered to show ES data only
            USE (cur_cip) again ALIAS els ORDER TAG "es_only"
            SELECT prj                  && switch to the projections data file
            *           && relate the projections file into each of the three single-level versions of the
                        CIP file, by school name
            set relation to alltrim(hs_home) into his,alltrim(ms_home) into mis,
                alltrim(es_home) into els
    else    && user wants to print contents of CIP
            select schools              && switch to the currently-selected CIP data file
            set order to level          && sort by level, then school name within level
            =prt_rept('cap_list')       && call the general-purpose report printing procedure to print
                                        report form "cap_list.frx"
            set order to name           && change sort sequence to default: school name
    endif
    select assignmt                     && switch to currently-selected plan file
    return                              && return to main Whiffer screen program

procedure showgets                      && refreshes display of data and controls on main Whiffer screen
    *       && call the "setnavbutn" procedure to refresh the settings of the navigation buttons in the
            middle of the Whiffer screen
    =setnavbutn()
    m.go_to=assignmt.plan_id            && copy the planning polygon ID number of the current record to
                                        the "go to" field
    show gets                                   && refresh the contents of the main Whiffer screen
    _CUROBJ=OBJNUM(M.go_to)                     && move the Whiffer screen's cursor to the "Go to" button
    return          && return control to the main Whiffer screen program

procedure go                    && moves to the user-specified record and displays its contents on the
                                main Whiffer screen
    seek m.go_to                && seek a record in the plan data file whose sort key matches the contents
                                of the "m.go_to" memvar,
    *                           && and move the record pointer to it
    if eof()        && if record pointer ends up at the end-of-file indicator,
            go bottom       && reposition it on the last record
    endif
    =showgets()             && call the "showgets" procedure to refresh the contents of the main
                            Whiffer screen
    return                  && return control to the main Whiffer screen program

procedure sign_out          && quits execution of the main Whiffer screen program when the user
                            clicks the "Quit" button,
    *                       && or selects the "Quit" option on the File menu
    quit                    && stop the program and close all of its windows
    return                  && return control to the computer's operating system

procedure nav_val           && moves the user through records according to which of four navigation
                            buttons is clicked on the main Whiffer screen
    do case                 && determine which button has been clicked and act accordingly
    case M.CHOICE = 1               && "first" button
        goto m.toprec               && move the record pointer to the record whose number was
                                    previously stored in "m.toprec"
    case M.CHOICE = 2               && "prev." button
        skip -1                     && move the record pointer to the record with the next-lowest
                                    planning polygon ID number
    case M.CHOICE = 3               && "next" button
        skip 1                      && move the record pointer to the record with the next-highest
                                    planning polygon ID number
    case M.CHOICE = 4               && "last" button
        goto m.botrec               && move the record pointer to the record whose number was
                                    previously stored in "m.botrec"
    endcase
    =showgets()                     && call the "showgets" procedure to refresh the contents of the
                                    main Whiffer screen
    return

procedure setnavbutn                && enables and disables navigation buttons on the main Whiffer
                                    screen
    do case                         && determine the current position of the record pointer in the
                                    currently-displayed plan file
    case recno()<>m.toprec and recno()<>m.botrec    && currently-displayed record is not the
                                                    first or last polygon record
        show get m.choice enable            && enable all of the navigation buttons
    case recno() = m.botrec         && currently-displayed record is the last polygon record
        show get M.CHOICE, 1 enable                 && enable the "first" button
        show get M.CHOICE, 2 enable                 && enable the "prev." button
        show get M.CHOICE, 3 disable        && disable the "next" button
        show get M.CHOICE, 4 disable        && disable the "last" button
    case recno() = m.toprec         && currently-displayed record is the first polygon record
        show get M.CHOICE, 1 disable        && disable the "first" button
        show get M.CHOICE, 2 disable        && disable the "prev." butto
        show get M.CHOICE, 3 enable                 && enable the "next" button
```

```
            show get M.CHOICE, 4 enable               && enable the "last" button
    endcase
    return          && return control to the calling procedure

procedure prt_rept                  && general-purpose procedure to print reports
    parameters rept                 && calling procedure passes the name of the report to be printed
    if m.go='Preview'               && user has selected the "preview" option on the calling
                                    procedure's prompt screen
            m.rept='report form '+m.rept+' preview'            && assemble a text string for macro
                                                               substitution below
            &rept                   && execute the text string as a command
            *       && give the user an option to print the report after closing the preview window
            *       && store user's response in memvar "m.prt_it"
            wait window "Do you want to print the previewed report? (Y/N)" to m.prt_it
            if upper(m.prt_it)='Y'          && the user wants to print after previewing
                    m.rept=strtran(m.rept,' preview')+' noconsole to printer prompt'
                            && assemble a different text string
                    &rept           && execute the text string as a command
            endif           && if the user doesn't want to print, simply fall through to the "return"
                            command without doing anything
    else            && user has selected the "print" option on the calling procedure's prompt screen
            m.rept='report form '+m.rept+' noconsole to printer prompt'   && assemble a text
                                                               string for macro substitution below
            &rept                   && execute the text string as a command
    endif
    return                          && return control to the calling procedure

procedure setup             && establishes the default data file environment for the Whiffer screen
    =get_schl()             && call the "get_schl" procedure to open the current CIP's data file
    SELECT 0                && select an new work area
    *       && the current projection file's file name should be stored in m.prjfile, so try to open it
    *       && if it can't be found, prompt the user to identify it by browsing files in the current
            directory
    *       && once it is opened, give it the generic alias "prj"
    uSE (LOCFILE(m.prjfile+".dbf","DBF","Where is the projections file?"))
            ALIAS prj exclusive
    index on plan_id tag plan_id additive        && sort the file by planning polygon ID number
    * relate the projections file to each of the one-school-level versions of the CIP file
    set relation to alltrim(hs_home) into his,alltrim(ms_home) into mis,
            alltrim(es_home) into els
    SELECT 0                && open a new work area
    =get_plan()             && call the "get_plan" procedure to open the current plan's data file
    return                  && return control to the calling procedure

procedure get_schl          && opens the currently-selected CIP data file under several aliases
    *       && close all instances of the current CIP data file that might be in existence
```

```
if used('schools')              && if there is currently a data file called "schools" in use, close it
        select schools
        use
endif
if used('his')          && if there is currently a data file called "his" in use, close it
        select his
        use
endif
if used('mis')          && if there is currently a data file called "mis" in use, close it
        select mis
        use
endif
if used('els')          && if there is currently a data file called "els" in use, close it
        select els
        use
endif
*       && the currently-selected CIP's file name should be stored in m.cur_cip, so try to open it
*       && if it can't be found, prompt the user to identify it by browsing files in the current
        directory
*       && once it is opened, give it the generic alias "schools"
USE (LOCFILE(m.cur_cip,"DBF","Select which CIP to use:")) exclusive ALIAS schools
if empty(dbf()) and file('cip.mem')  && if no user selection, but a previous settings file exists
        restore from cip.mem additive       && try to get the CIP name from the settings file
        wait window "No CIP selected - using previous CIP" nowait     && tell the user what
                                                                      has happened
        USE (cur_cip) exclusive ALIAS schools              && use the CIP file named in the
                                                           settings file
endif
*       && recreate the important CIP sort sequences
INDEX ON schl_name FOR " HS"$schl_name TAG Hs_only ADDITIVE
INDEX ON schl_name FOR " MS"$schl_name TAG Ms_only ADDITIVE
INDEX ON schl_name FOR " ES"$schl_name TAG Es_only ADDITIVE
INDEX ON number TAG Number ADDITIVE
INDEX ON level+schl_name TAG Level ADDITIVE
INDEX ON schl_name TAG Name ADDITIVE
m.cur_cip=dbf()                 && store the name of the currently-selected CIP file in a memvar
save to cip.mem all like cur_cip            && save the current CIP's name for next start up of
                                            Whiffer
use             && close the CIP and re-open it in shared use
select 0                && open a new work area
USE (cur_cip) ALIAS his ORDER TAG "hs_only"        && open the CIP data file as "his",
                                                   filtered to show HS data only
SELECT 0                && open a new work area
USE (cur_cip) again ALIAS mis ORDER TAG "ms_only"       && open the CIP data file as
                                                   "mis", filtered to show MS data only
SELECT 0                && open a new work area
```

```
    USE (cur_cip) again ALIAS els ORDER TAG "es_only"          && open the CIP data file as
                                                    "els", filtered to show ES data only
    SELECT 0                && open a new work area
    USE (cur_cip) again ALIAS schools ORDER TAG "name"         && open the CIP data file as
                                                               "schools", with no filtering
    m.last_slash=rat("\",dbf())         && store the position of the last "backslash" character in
                                        the current CIP's file name
    *&& store the text string from the left of the last backslash to the end of the CIP's file name
    m.cip_in_use=substr(dbf(),m.last_slash+1)
    *       && remove the "period" and file extension from the CIP name
    m.cip_in_use=left(m.cip_in_use,at(".",m.cip_in_use)-1)
    return          && return control to the calling procedure

procedure get_plan                  && open the currently-selected plan data file
    *       && the currently-selected plan's file name should be stored in m.cur_db, so try to open it
    *       && if it can't be found, prompt the user to identify it by browsing files in the current
            directory
    *       && once it is opened, give it the generic alias "assignmt"
    USE (LOCFILE(m.cur_db,"DBF","Select a plan to work on:")) exclusive ALIAS assignmt
    index on plan_id tag plan_id additive       && recreate the planning polygon ID number index
    set relation to plan_id into prj            && relate the plan data file to the projection data
                                                file by planning polygon ID number
    m.cur_db=dbf()                      && store the file name of the current plan in a memvar
    save to db.mem all like cur_db              && save the current plan's name for next start up of
                                                Whiffer
    goto bottom             && move to the record with the highest planning polygon ID number
    m.botrec=recno()        && store its record number in a memvar for nav button use
    goto top                && move to the record with the lowest planning polygon ID number
    M.TOPREC=recno()            && store its record number in a memvar for nav button use
    m.last_slash=rat("\",dbf())  && store the position of the last backslash in the plan file name
    m.pln_in_use=substr(dbf(),m.last_slash+1)           && extract just the plan name plus file
                                                        extension from the full name
    m.pln_in_use=left(m.pln_in_use,at(".",m.pln_in_use)-1)         && remove the file extension
                                                    and save plan name in memvar
    return          && return control to the calling procedure

procedure swit_cip          && changes the active CIP file after the user clicks on the "Switch CIPs"
                            button
    m.cur_cip='a.dbf'       && load a dummy value to force the user file selection prompt to appear
    =get_schl()             && call the "get_schl" procedure to allow the user to select a different
                            CIP data file
    select assignmt         && switch to the current plan's data file
    =showgets()             && call the "showgets" procedure to refresh the contents of the main
                            Whiffer screen
    return                      && return control to the calling procedure
```

```
procedure swit_pln      && changes the active plan file after the user clicks on the "Switch Plans"
                        button
    select assignmt             && switch to the current plan's data file
    use                                 && close it
    m.cur_db='a.dbf'    && load a dummy value to force the user file selection prompt to appear
    =get_plan()         && call the "get_plan" procedure to allow the user to select a different
                        plan data file
    =showgets()         && call the "showgets" procedure to refresh the contents of the main
                        Whiffer screen
    *       && call the "get_stat" procedure to refresh the contents of the scrolling schools list at the
            bottom of the Whiffer screen
    =get_stat(m.cur_lev)
    return                      && return control to the calling procedure

procedure colorset              && establishes a uniform color scheme for the application
    private m.oldselect         && establish a local memvar
    m.oldselect=select()        && load it with the name of the currnet active data file
    if not used("colorrsc")             && if "colorrsc" file is not already open
            use colorrsc again alias colorrsc in 0      && open it
    endif
    select colorrsc                     && switch to the colorrsc data file
    locate for upper(name)='MICROSSIS'          && find the record containing the "microssis" color
                                                scheme
    private m.xx,m.yy,m.zz      && establish three additional local memvars
    for m.xx=1 to 12            && cycle through 12 iterations of this loop
            *  get the contents of this color scheme's memo fld
            m.zz=eval("colorrsc.scheme"+alltrim(str(m.xx)))
            *  find the beginning of the 6th set of RGB values
            m.yy=ATC("R",m.zz,6)
            *  set up the color scheme in two sections, to avoid "String too long" error
            set color of scheme m.xx to (substr(m.zz,1,m.yy-1))
            set color of scheme m.xx to (",,,,,"+substr(m.zz,m.yy))
    endfor          && repeat above for all 12 color settings
    select colorrsc             && make sure colorrsc file is active
    use                 && close it
    select (oldselect)          && make original data file the active one
    return          && return control to the main Whiffer screen's set up program

procedure esc_proc              && lets user use Esc key to close pop-ups
    *       && Escape key is normally disabled to keep user from terminating program accidentally
    if wvisible('List')         && if there is a visible window named "list" (all pop-up lists are
                                opened in a window called "List")
            keyboard '{ctrl+w}'         && simulate the action of the Ctrl-W key combination (this
                                        key combo closes the active window)
    endif
    return          && return control to the calling procedure
```

```
procedure err_msg                   && handles display of error messages to user,
    *         && also lets user determine what to do with data when record is locked by another
    *         && "on error" command in setup code calls this proc with FoxPro error number &
              message text when an error condition occurs
    parameters errnum,msg
    define window err_win from 0,0 to 10,79 font 'geneva',10          && define a new window
    ?? chr(7)       && make an alert sound
    do case         && look at the record number to determine what to display to the user
    case errnum=1705                && "Can't get exclusive use" error
        line1='Another user is already using the '+alltrim(m.fil_in_use)+' file.'  && compose
                                                                              message line 1
        line2='You cannot run this process with other users logged on.
                Please try again later . . .'                 && compose message line 2
        m.exit_now=.t.              && causes calling procedure to abort
    case errnum=108                 &&      "Can't open file" error
        line1='File has been locked by another user.'       && compose message line 1
        line2='Please try again later. . .'     && compose message line 2
        m.exit_now=.t.              && causes calling procedure to abort
        m.wait='Discard'            && sets up "Discard changes" as the default user action on
                                    prompt screen
    case errnum=109                 && "Can't lock record for editing" error
        line1='Record has been locked by another user.'          && compose message line 1
        if m.deleting           && if user is in "delete record" mode
            line2='You must have sole access to this record to delete it.'     && compose
                                                                              message line 2
            m.wait='Hold'           && sets up "Hold changes" as the default user action on
                                    prompt screen
        else                        && user is trying to modify record
            line2='You may HOLD your changes to this screen and
                try to save them again later,'          && compose message line 2
            line3='or you may DISCARD your changes to this record and
                work on other records.'                 && compose message line 3
        endif
    otherwise                       && all other types of errors
        line1=msg          && compose message line 1 (just repeat the FoxPro error message)
        line2='Please write down this message and call Larry Weaver at x5672' && compose
                                                                              message line 2
        m.wait='Save'    && sets up "Save changes" as the default user action on prompt screen
    endcase
    move window err_win center          && center the error message window
    activate window err_win top         && display the error message window on top of all others
    @1,(wcols()-len(line1))/2 say line1 font 'geneva',10         && display message line 1
    @3,(wcols()-len(line2))/2 say line2 font 'geneva',10         && display message line 2
    do case
    case errnum=109 and not m.deleting         && add a user action button to the window if user
```

```
                                                                   has options
            @5,(wcols()-len(line3))/2 say line3 font 'geneva',10       && display message line 3
            @7,30 get m.wait picture "@*T3H \!Hold;Discard" size 1.5,9,2
                    font 'geneva',10 style '@B'          && display the button
            read            && get the user's choice from the button that is clicked
    otherwise
            wait window          && pause execution until user clicks mouse or presses a key
    endcase
    release window err_win              && close the error message window
                     return            && return control to the calling procedure
```

<u>Polyback screen activate code:</u> This code executes whenever the main Whiffer screen is made the topmost window on the computer dislay, either by program or user action:

```
if used('assignmt')         && if there is a data file in use named “assignmt”
    select assignmt         && make it the active work area
endif
```

<u>Expfprpt screen cleanup code:</u> This code executes whenever the user clicks on the “Select type of file to export:” prompt at the top of the Export Feed % file prompt screen:

```
procedure setboxes          && sets check boxes appropriate to export user requested
    if m.choi=1             && % of Middle from each Elem – turn off ES and HS grade levels
            m.g0=.f.
            m.g1=.f.
            m.g2=.f.
            m.g3=.f.
            m.g4=.f.
            m.g5=.f.
            m.g6=.t.        && turn on MS grade levels
            m.g7=.t.
            m.g8=.t.
            m.g9=.f.
            m.g10=.f.
            m.g11=.f.
            m.g12=.f.
    else                    && % of High from each Middle – turn off ES and MS grade levels
            m.g0=.f.
            m.g1=.f.
            m.g2=.f.
            m.g3=.f.
            m.g4=.f.
            m.g5=.f.
            m.g6=.f.
            m.g7=.f.
            m.g8=.f.
            m.g9=.t.        && turn on HS grade levels
            m.g10=.t.
            m.g11=.t.
            m.g12=.t.
    endif
    for n=0 to 12           && cycle through grade level checkboxes
            m.gfld='m.g'+alltrim(str(n))            && set up for macro substitution on next line
            show get &gfld          && refresh display of field on prompt screen
    endfor
    return                  && return control to prompt screen program
```

Oldfdr screen cleanup code: This code executes whenever the user clicks on the “Select report to create:” prompt at the top of the Report Options prompt screen:

```
procedure setboxes          && sets check boxes appropriate to report user requested
    do case
    case m.choi=1 or m.choi=2       && % of Elem to each Middle or % of Elem to each High
            m.g0=.t.                && turn on ES grade levels
            m.g1=.t.
            m.g2=.t.
            m.g3=.t.
            m.g4=.t.
            m.g5=.t.
            m.g6=.f.                && turn off MS grade levels
            m.g7=.f.
            m.g8=.f.
            m.g9=.f.                && turn off HS grade levels
            m.g10=.f.
            m.g11=.f.
            m.g12=.f.
    case m.choi=3 or m.choi=4       && % of Middle to each High or % of Middle from each Elem
            m.g0=.f.                && turn off ES grade levels
            m.g1=.f.
            m.g2=.f.
            m.g3=.f.
            m.g4=.f.
            m.g5=.f.
            m.g6=.t.                && turn on MS grade levels
            m.g7=.t.
            m.g8=.t.
            m.g9=.f.                && turn off HS grade levels
            m.g10=.f.
            m.g11=.f.
            m.g12=.f.
```

```
otherwise       && % of Middle to each High or % of Middle from each Elem
        m.g0=.f.                && turn off ES grade levels
        m.g1=.f.
        m.g2=.f.
        m.g3=.f.
        m.g4=.f.
        m.g5=.f.
        m.g6=.f.                && turn off MS grade levels
        m.g7=.f.
        m.g8=.f.
        m.g9=.t.                && turn on HS grade levels
        m.g10=.t.
        m.g11=.t.
        m.g12=.t.
endcase
for n=0 to 12                   && cycle through grade level checkboxes
        m.gfld='m.g'+alltrim(str(n))        && set up for macro substitution on next line
        show get &gfld          && refresh display of field on prompt screen
endfor
            return                      && return control to prompt screen program
```

Appendix A: Examples of Whiffer Reports

Planning Polygon Summary

Atholton HS

Plan		MSA	Curr.	Proposed			2009	Proposed			2010	Proposed			2011	Proposed			2012	Proposed			2013	Proposed			2014
Poly	FARM	Read/Math	HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS
ID #	% (#)	(% passing)	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop
Tot.	4%(55)	91%/86%	1354	7	86	60	1465	5	92	60	1466	24	95	50	1449	64	36	0	1449	21	6	0	1413	55	8	0	1402

Centennial HS

Plan		MSA	Curr.	Proposed			2009	Proposed			2010	Proposed			2011	Proposed			2012	Proposed			2013	Proposed			2014
Poly	FARM	Read/Math	HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS
ID #	% (#)	(% passing)	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop
Tot.	5%(81)	89%/87%	1500	12	0	0	1509	25	0	0	1503	24	0	0	1505	23	0	0	1511	1	0	0	1458	2	0	0	1458

Glenelg HS

Plan		MSA	Curr.	Proposed			2009	Proposed			2010	Proposed			2011	Proposed			2012	Proposed			2013	Proposed			2014
Poly	FARM	Read/Math	HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS
ID #	% (#)	(% passing)	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop
Tot.	3%(36)	90%/90%	1220	95	0	0	1335	59	0	0	1347	77	0	0	1319	60	0	0	1340	90	0	0	1287	67	0	0	1261

Hammond HS

Plan		MSA	Curr.	Proposed			2009	Proposed			2010	Proposed			2011	Proposed			2012	Proposed			2013	Proposed			2014
Poly	FARM	Read/Math	HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS
ID #	% (#)	(% passing)	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop
Tot.	16%(211)	80%/69%	1314	23	80	50	1299	14	103	130	1274	7	40	130	1264	42	47	139	1245	25	0	130	1220	21	9	108	1198

Howard HS

Plan		MSA	Curr.	Proposed			2009	Proposed			2010	Proposed			2011	Proposed			2012	Proposed			2013	Proposed			2014
Poly	FARM	Read/Math	HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS
ID #	% (#)	(% passing)	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop
Tot.	6%(82)	82%/82%	1383	60	35	64	1474	41	35	80	1486	37	35	66	1488	72	35	0	1528	119	32	0	1508	144	1	0	1514

Long Reach HS

Plan		MSA	Curr.	Proposed			2009	Proposed			2010	Proposed			2011	Proposed			2012	Proposed			2013	Proposed			2014
Poly	FARM	Read/Math	HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS
ID #	% (#)	(% passing)	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop
Tot.	17%(212)	83%/76%	1261	5	100	134	1232	37	106	104	1273	3	58	100	1309	12	58	36	1333	10	40	0	1357	9	60	0	1362

Marriotts Ridge HS

Plan		MSA	Curr.	Proposed			2009	Proposed			2010	Proposed			2011	Proposed			2012	Proposed			2013	Proposed			2014
Poly	FARM	Read/Math	HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS
ID #	% (#)	(% passing)	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop
Tot.	3%(40)	92%/80%	1290	86	70	40	1290	63	70	8	1288	81	46	0	1292	77	70	40	1265	47	66	179	1263	24	100	135	1252

Mt Hebron HS

Plan		MSA	Curr.	Proposed			2009	Proposed			2010	Proposed			2011	Proposed			2012	Proposed			2013	Proposed			2014
Poly	FARM	Read/Math	HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS	Housing Units			HS
ID #	% (#)	(% passing)	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop
Tot.	7%(98)	83%/81%	1407	62	17	0	1461	89	0	0	1442	112	0	0	1391	62	7	0	1384	1	0	0	1326	5	0	40	1342

Planning Polygon Summary

Kindergarten Included

Atholton ES

Plan Poly ID #	FARM % (#)	MSA Read/Math (% passing)	Curr. ES Pop	Proposed Housing Units			2009 ES Pop	Proposed Housing Units			2010 ES Pop	Proposed Housing Units			2011 ES Pop	Proposed Housing Units			2012 ES Pop	Proposed Housing Units			2013 ES Pop	Proposed Housing Units			2014 ES Pop
				SFD	SFA	APT		SFD	SFA	APT		SFD	SFA	APT		SFD	SFA	APT		SFD	SFA	APT		SFD	SFA	APT	
13	<5%	91%/91%	26	0	0	0	26	0	0	0	27	0	0	0	28	1	0	0	27	2	0	0	27	0	0	0	27
14	6%(2)	86%/79%	33	0	0	0	33	5	0	0	36	0	0	0	37	0	1	0	37	0	0	0	35	1	0	0	36
16	<5%	>95%/>95%	15	0	0	0	16	0	0	0	16	0	0	0	17	0	0	0	17	0	0	0	17	0	0	0	16
17	42%(19)	82%/82%	45	0	0	0	45	0	0	0	46	0	0	0	48	0	0	0	47	0	0	0	45	0	0	0	45
56	8%(5)	82%/88%	63	0	0	0	62	0	0	0	67	0	0	0	67	13	0	0	70	2	0	0	70	4	0	0	72
57	6%(2)	94%/100%	32	0	0	0	32	0	0	0	34	0	0	0	36	8	0	0	38	1	0	0	36	4	0	0	38
1013	*	*	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
1014	9%(2)	>95%/75%	23	0	0	0	23	2	0	0	25	0	0	0	26	4	0	0	27	0	0	0	26	0	0	0	26
1016	20%(2)	>95%/>95%	10	0	0	0	10	0	0	0	11	0	0	0	11	0	0	0	11	0	0	0	11	0	0	0	11
1017	25%(9)	86%/>95%	36	0	0	0	35	0	0	0	37	0	0	0	38	0	0	0	38	0	0	0	36	0	0	0	37
1056	7%(2)	89%/89%	27	0	0	0	27	0	0	0	27	0	0	0	27	2	0	0	29	0	0	0	27	2	0	0	28
1057	<5%	94%/94%	56	0	0	0	56	0	0	0	58	0	0	0	61	0	0	0	59	0	0	0	57	0	0	0	57
Tot.	12%(45)	90%/90%	367	0	0	0	366	7	0	0	385	0	0	0	397	28	1	0	401	5	0	0	388	11	0	0	394

Bellows Spring ES

Plan Poly ID #	FARM % (#)	MSA Read/Math (% passing)	Curr. ES Pop	Proposed Housing Units			2009 ES Pop	Proposed Housing Units			2010 ES Pop	Proposed Housing Units			2011 ES Pop	Proposed Housing Units			2012 ES Pop	Proposed Housing Units			2013 ES Pop	Proposed Housing Units			2014 ES Pop
				SFD	SFA	APT		SFD	SFA	APT		SFD	SFA	APT		SFD	SFA	APT		SFD	SFA	APT		SFD	SFA	APT	
33	50%(7)	>95%/>95%	14	0	0	0	15	0	50	90	37	0	50	60	57	0	50	60	77	0	40	60	90	4	42	60	107
35	*	*	8	0	0	0	8	0	30	100	29	0	30	100	50	0	30	100	72	0	30	100	88	0	30	100	109
76	13%(7)	>95%/80%	56	0	0	0	57	0	0	0	56	0	0	0	56	0	0	0	56	0	0	0	54	0	0	0	53
77	7%(2)	88%/88%	30	0	0	0	31	0	0	0	30	0	0	0	31	0	0	0	31	0	0	0	30	0	0	0	29
82	12%(5)	>95%/88%	43	0	0	0	45	0	0	0	44	0	0	0	44	0	0	0	43	0	0	0	43	0	0	0	42
269	<5%	>95%/>95%	13	0	0	0	13	0	0	0	13	0	0	0	13	0	0	0	13	0	0	0	12	0	0	0	12
298	<5%	89%/89%	97	0	0	0	99	0	0	0	97	0	0	0	99	0	0	0	96	0	0	0	96	0	0	0	95
1033	21%(12)	88%/81%	56	0	0	0	57	0	0	0	57	0	0	0	56	0	0	0	57	0	0	33	59	0	0	0	58
1035	<5%	>95%/67%	18	0	0	0	19	0	0	0	19	0	0	0	19	0	0	0	19	0	0	0	18	0	0	0	18
1076	30%(3)	>95%/>95%	10	0	50	0	24	0	41	44	39	0	45	40	54	0	58	0	67	0	40	0	72	0	60	0	83
1077	21%(7)	88%/63%	34	0	0	0	35	0	0	0	34	0	0	0	34	0	0	0	34	0	0	0	33	0	0	0	32
1082	13%(7)	85%/85%	52	0	0	0	54	0	0	0	53	0	0	0	53	0	0	0	53	0	0	0	51	0	0	0	50
1269	*	*	3	0	0	0	3	12	0	0	4	0	0	0	4	0	0	0	4	0	0	0	4	0	0	0	4
1298	5%(3)	>95%/93%	57	0	0	0	59	0	0	0	58	0	0	0	58	0	0	0	58	0	0	0	56	0	0	0	55
2035	7%(1)	>95%/>95%	14	0	0	0	15	0	0	0	15	0	0	0	15	0	0	0	15	0	0	0	14	1	0	0	14
2077	6%(5)	92%/88%	82	0	0	0	84	0	0	0	82	0	0	0	83	0	0	0	84	0	0	0	80	0	0	0	80
Tot.	11%(67)	93%/86%	587	0	50	0	618	12	121	234	667	0	125	200	726	0	138	160	779	0	110	193	800	5	132	160	841

Bollman Bridge ES

Plan Poly ID #	FARM % (#)	MSA Read/Math (% passing)	Curr. ES Pop	Proposed Housing Units			2009 ES Pop	Proposed Housing Units			2010 ES Pop	Proposed Housing Units			2011 ES Pop	Proposed Housing Units			2012 ES Pop	Proposed Housing Units			2013 ES Pop	Proposed Housing Units			2014 ES Pop
				SFD	SFA	APT		SFD	SFA	APT		SFD	SFA	APT		SFD	SFA	APT		SFD	SFA	APT		SFD	SFA	APT	
20	<5%	95%/90%	66	0	0	0	64	0	0	0	65	0	0	0	63	0	0	0	67	0	0	0	64	0	0	0	64
21	6%(2)	92%/92%	35	0	0	0	35	0	0	0	35	0	0	0	34	0	0	0	35	0	0	0	35	0	0	0	35
22	8%(1)	86%/86%	13	0	0	0	13	0	0	0	13	0	0	0	13	0	0	0	13	0	0	0	13	0	0	0	13
23	26%(16)	72%/72%	61	0	0	0	60	0	0	0	60	0	0	0	60	0	0	0	60	0	0	0	61	0	0	0	61
24	30%(21)	59%/67%	71	0	0	0	69	0	0	0	71	0	0	0	69	0	0	0	69	0	0	0	70	1	0	0	72
25	23%(3)	60%/80%	13	0	0	0	14	0	0	0	14	0	0	0	14	0	0	0	14	0	0	0	14	0	0	0	14
27	25%(4)	50%/50%	16	9	0	0	20	1	0	0	20	5	0	0	21	12	0	0	26	0	0	0	26	2	0	0	27

Planning Polygon Summary

Kindergarten Included

Atholton ES

Plan	Curr.	Proposed			2009	Proposed			2010	Proposed			2011	Proposed			2012	Proposed			2013	Proposed			2014
Poly	ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES
ID #	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop
Tot.	367	0	0	0	366	7	0	0	385	0	0	0	397	28	1	0	401	5	0	0	388	11	0	0	394

Bellows Spring ES

Plan	Curr.	Proposed			2009	Proposed			2010	Proposed			2011	Proposed			2012	Proposed			2013	Proposed			2014
Poly	ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES
ID #	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop
Tot.	587	0	50	0	618	12	121	234	667	0	125	200	726	0	138	160	779	0	110	193	800	5	132	160	841

Bollman Bridge ES

Plan	Curr.	Proposed			2009	Proposed			2010	Proposed			2011	Proposed			2012	Proposed			2013	Proposed			2014
Poly	ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES
ID #	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop
Tot.	556	9	0	0	555	1	0	80	569	6	0	80	577	18	0	80	607	0	0	80	613	7	0	96	630

Bryant Woods ES

Plan	Curr.	Proposed			2009	Proposed			2010	Proposed			2011	Proposed			2012	Proposed			2013	Proposed			2014
Poly	ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES
ID #	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop
Tot.	389	0	0	0	393	0	0	0	397	0	0	0	397	0	0	0	401	0	0	0	397	0	0	0	395

Bushy Park ES

Plan	Curr.	Proposed			2009	Proposed			2010	Proposed			2011	Proposed			2012	Proposed			2013	Proposed			2014
Poly	ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES
ID #	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop
Tot.	659	32	0	0	609	28	0	0	599	22	0	0	580	31	0	0	569	53	0	0	561	32	0	0	556

Centennial Lane ES

Plan	Curr.	Proposed			2009	Proposed			2010	Proposed			2011	Proposed			2012	Proposed			2013	Proposed			2014
Poly	ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES
ID #	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop
Tot.	617	1	0	0	627	4	0	0	638	1	0	0	648	8	0	0	643	0	0	0	628	1	0	0	606

Clarksville ES

Plan	Curr.	Proposed			2009	Proposed			2010	Proposed			2011	Proposed			2012	Proposed			2013	Proposed			2014
Poly	ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES
ID #	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop
Tot.	781	8	0	0	729	21	0	0	664	0	0	0	624	22	1	0	585	13	0	0	553	11	3	0	503

Clemens Crossing ES

Plan	Curr.	Proposed			2009	Proposed			2010	Proposed			2011	Proposed			2012	Proposed			2013	Proposed			2014
Poly	ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES	Housing Units			ES
ID #	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop
Tot.	397	0	0	0	453	3	0	0	468	0	0	0	473	31	2	0	491	1	0	0	506	8	2	0	501

Planning Polygon Summary

Bonnie Branch MS

Plan	Curr.	Proposed			2009	Proposed			2010	Proposed			2011	Proposed			2012	Proposed			2013	Proposed			2014
Poly	MS	Housing Units			MS	Housing Units			MS	Housing Units			MS	Housing Units			MS	Housing Units			MS	Housing Units			MS
ID #	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop
68	**49**	0	0	0	**43**	0	0	0	**42**	0	0	0	**42**	0	0	0	**41**	0	0	0	**42**	0	0	0	**42**
69	**26**	0	0	0	**27**	0	0	0	**27**	0	0	0	**27**	0	0	0	**27**	0	0	0	**28**	0	0	0	**28**
76	**27**	0	0	0	**28**	0	0	0	**27**	0	0	0	**28**	0	0	0	**27**	0	0	0	**28**	0	0	0	**27**
83	**9**	0	0	0	**8**	0	0	0	**8**	0	0	0	**8**	0	0	0	**8**	0	0	0	**8**	0	0	0	**8**
84	**22**	2	0	0	**15**	1	0	0	**14**	0	0	0	**14**	0	0	0	**13**	0	0	0	**14**	6	0	0	**13**
85	**47**	0	0	0	**47**	0	0	0	**45**	0	0	0	**44**	0	0	0	**42**	0	0	0	**43**	0	0	0	**41**
86	**33**	0	0	0	**22**	0	0	0	**21**	0	0	0	**21**	0	0	0	**20**	0	0	0	**20**	0	0	0	**20**
88	**17**	19	0	0	**28**	7	0	0	**29**	1	0	0	**31**	1	0	0	**30**	0	0	0	**31**	1	0	0	**31**
89	**21**	0	0	0	**25**	0	0	0	**25**	0	0	0	**25**	1	0	0	**24**	0	0	0	**24**	0	0	0	**23**
90	**4**	0	0	0	**7**	0	0	0	**7**	0	0	0	**7**	0	0	0	**7**	0	0	0	**7**	0	0	0	**6**
91	**8**	13	0	0	**12**	2	0	0	**13**	0	0	0	**14**	0	0	0	**14**	14	0	0	**15**	0	0	0	**16**
95	**29**	0	0	0	**32**	0	0	0	**31**	0	0	0	**31**	0	0	0	**30**	0	0	0	**30**	0	0	0	**29**
261	**8**	0	0	0	**14**	0	0	0	**14**	0	0	0	**15**	0	0	0	**14**	0	0	0	**15**	0	0	0	**15**
264	**5**	0	0	0	**3**	0	0	0	**3**	0	0	0	**3**	0	0	0	**3**	0	0	0	**3**	0	0	0	**3**
301	**5**	0	0	0	**7**	0	0	0	**7**	0	0	0	**7**	0	0	0	**7**	0	0	0	**7**	0	0	0	**7**
1068	**41**	0	0	0	**49**	0	0	0	**48**	0	0	0	**49**	0	0	0	**49**	0	0	0	**51**	0	0	0	**50**
1069	**34**	0	0	0	**46**	0	0	0	**47**	0	0	0	**48**	0	0	0	**47**	0	0	0	**49**	0	0	0	**49**
1076	**1**	0	50	0	**5**	0	41	44	**8**	0	45	40	**12**	0	58	0	**15**	0	40	0	**18**	0	60	0	**21**
1083	**13**	0	0	0	**24**	0	0	0	**24**	0	0	0	**25**	0	0	0	**25**	0	0	0	**26**	3	0	0	**26**
1084	**2**	0	0	0	**7**	0	0	0	**7**	0	0	0	**7**	0	0	0	**7**	0	0	0	**7**	0	0	0	**7**
1085	**42**	0	0	0	**32**	0	0	0	**30**	0	0	0	**29**	0	0	0	**28**	0	0	0	**28**	0	0	0	**27**
1086	**25**	0	0	0	**16**	0	0	0	**15**	0	0	0	**15**	0	0	0	**14**	0	0	0	**14**	0	0	0	**14**
1088	**28**	0	0	0	**23**	0	0	0	**22**	0	0	0	**21**	20	0	0	**20**	0	0	0	**21**	20	0	0	**20**
1089	**11**	0	0	0	**11**	0	0	0	**10**	0	0	0	**10**	1	0	0	**10**	0	0	0	**10**	50	0	0	**10**
1090	**14**	0	0	0	**8**	0	0	0	**7**	0	0	0	**7**	0	0	0	**6**	0	0	0	**6**	0	0	0	**6**
1091	**2**	0	0	0	**2**	0	0	0	**2**	0	0	0	**2**	0	0	0	**2**	3	0	0	**2**	16	0	0	**2**
1095	**31**	0	0	0	**24**	0	0	0	**22**	0	0	0	**22**	0	0	0	**20**	0	0	0	**20**	0	0	0	**20**
1261	**3**	0	0	0	**3**	0	0	0	**3**	0	0	0	**3**	0	0	0	**2**	0	0	0	**3**	0	0	0	**3**
1264	**11**	0	0	0	**9**	0	0	0	**9**	0	0	0	**9**	0	0	0	**8**	0	0	0	**9**	0	0	0	**8**
1301	**19**	0	0	0	**18**	0	0	0	**17**	0	0	0	**17**	0	0	0	**16**	0	0	0	**16**	0	0	0	**16**
2068	**19**	0	0	0	**17**	0	0	0	**17**	0	0	0	**17**	0	0	0	**17**	0	0	0	**17**	0	0	0	**17**
2089	**3**	0	0	0	**6**	0	0	0	**6**	0	0	0	**6**	0	0	0	**6**	0	0	0	**6**	0	0	0	**6**
2091	**35**	0	0	0	**54**	0	0	0	**54**	0	0	0	**56**	0	0	0	**56**	0	0	0	**58**	0	0	0	**58**
2095	**31**	0	0	0	**27**	0	0	0	**26**	0	0	0	**26**	0	0	0	**24**	0	0	0	**24**	0	0	0	**23**
3091	**3**	0	0	0	**2**	0	0	0	**2**	0	0	0	**2**	0	0	0	**2**	0	0	0	**2**	1	0	0	**2**
Tot.	**678**	34	50	0	**701**	10	41	44	**689**	1	45	40	**700**	23	58	0	**681**	17	40	0	**702**	97	60	0	**694**

Burleigh Manor MS

Plan	Curr.	Proposed			2009	Proposed			2010	Proposed			2011	Proposed			2012	Proposed			2013	Proposed			2014
Poly	MS	Housing Units			MS	Housing Units			MS	Housing Units			MS	Housing Units			MS	Housing Units			MS	Housing Units			MS
ID #	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop	SFD	SFA	APT	Pop
19	**32**	0	0	0	**36**	0	0	0	**36**	0	0	0	**34**	0	0	0	**36**	0	0	0	**37**	0	0	0	**40**
97	**0**	0	0	0	**0**	0	0	0	**0**	0	0	0	**0**	0	0	0	**0**	0	0	0	**0**	0	0	0	**0**
147	**6**	0	0	0	**11**	1	0	0	**12**	0	0	0	**12**	0	0	0	**12**	0	0	0	**13**	1	0	0	**14**
153	**11**	0	0	0	**21**	0	0	0	**21**	0	0	0	**21**	0	0	0	**22**	0	0	0	**23**	0	0	0	**24**
154	**12**	0	0	0	**16**	0	0	0	**16**	0	0	0	**15**	0	0	0	**15**	0	0	0	**16**	0	0	0	**17**
156	**48**	0	0	0	**35**	0	0	0	**33**	0	0	0	**31**	1	0	0	**31**	0	0	0	**32**	0	0	0	**34**
157	**23**	0	0	0	**21**	0	0	0	**20**	1	0	0	**19**	0	0	0	**20**	0	0	0	**20**	0	0	0	**21**

% of Elementary to each High
Grades: K12345

Atholton ES	Hammond HS	277	75.4%
	Oakland Mills HS	90	24.5%
		367	**100%**
Bellows Spring ES	Long Reach HS	509	86.7%
	Oakland Mills HS	78	13.2%
		587	**100%**
Bollman Bridge ES	Hammond HS	556	100.0%
		556	**100%**
Bryant Woods ES	Atholton HS	177	45.5%
	Wilde Lake HS	212	54.5%
		389	**100%**
Bushy Park ES	Glenelg HS	659	100.0%
		659	**100%**
Centennial Lane ES	Centennial HS	617	100.0%
		617	**100%**
Clarksville ES	Atholton HS	230	29.4%
	River Hill HS	551	70.5%
		781	**100%**
Clemens Crossing ES	Atholton HS	295	74.3%
	Wilde Lake HS	102	25.6%
		397	**100%**
Cradlerock ES	Oakland Mills HS	453	100.0%
		453	**100%**
Dayton Oaks ES	Atholton HS	70	8.7%
	Glenelg HS	111	13.8%
	River Hill HS	618	77.3%
		799	**100%**
Deep Run ES	Hammond HS	112	21.6%
	Long Reach HS	405	78.3%
		517	**100%**
Elkridge ES	Howard HS	405	58.8%
	Long Reach HS	283	41.1%
		688	**100%**
Forest Ridge ES	Hammond HS	347	58.9%
	Reservoir HS	242	41.0%
		589	**100%**
Fulton ES	Atholton HS	3	0.6%
	Reservoir HS	436	94.7%
	River Hill HS	21	4.5%
		460	**100%**
Gorman Crossing ES	Atholton HS	230	43.3%
	Reservoir HS	301	56.6%
		531	**100%**
Guilford ES	Hammond HS	344	87.0%
	Oakland Mills HS	51	12.9%
		395	**100%**
Hammond ES	Atholton HS	233	51.7%
	Hammond HS	129	28.6%
	Reservoir HS	88	19.5%

% of Elementary to each Middle
Grades: K12345

Atholton ES	Hammond MS	196	53.4%
	Oakland Mills MS	90	24.5%
	Patuxent Valley MS	81	22.0%
		367	**100%**
Bellows Spring ES	Bonnie Branch MS	66	11.2%
	Mayfield Woods MS	521	88.7%
		587	**100%**
Bollman Bridge ES	Patuxent Valley MS	556	100.0%
		556	**100%**
Bryant Woods ES	Wilde Lake MS	389	100.0%
		389	**100%**
Bushy Park ES	Folly Quarter MS	132	20.0%
	Glenwood MS	527	79.9%
		659	**100%**
Centennial Lane ES	Burleigh Manor MS	617	100.0%
		617	**100%**
Clarksville ES	Clarksville MS	781	100.0%
	Folly Quarter MS	0	0.0%
		781	**100%**
Clemens Crossing ES	Harpers Choice MS	102	25.6%
	Wilde Lake MS	295	74.3%
		397	**100%**
Cradlerock ES	Cradlerock MS	453	100.0%
		453	**100%**
Dayton Oaks ES	Folly Quarter MS	350	43.8%
	Lime Kiln MS	449	56.2%
		799	**100%**
Deep Run ES	Mayfield Woods MS	405	78.3%
	Patuxent Valley MS	112	21.6%
		517	**100%**
Elkridge ES	Elkridge Landing MS	688	100.0%
		688	**100%**
Forest Ridge ES	Murray Hill MS	278	47.2%
	Patuxent Valley MS	311	52.8%
		589	**100%**
Fulton ES	Hammond MS	190	41.3%
	Lime Kiln MS	270	58.7%
		460	**100%**
Gorman Crossing ES	Hammond MS	85	16.0%
	Murray Hill MS	446	83.9%
		531	**100%**
Guilford ES	Cradlerock MS	157	39.7%
	Patuxent Valley MS	238	60.2%
		395	**100%**
Hammond ES	Hammond MS	450	100.0%
	Murray Hill MS	0	0.0%
		450	**100%**

% of Middle from each Elementary
Grades: 678

Bonnie Branch MS	Bellows Spring ES	28	4.1%
	Ilchester ES	348	51.3%
	Jeffers Hill ES	11	1.6%
	Phelps Luck ES	185	27.2%
	Rockburn ES	106	15.6%
		678	**100%**
Burleigh Manor MS	Centennial Lane ES	418	61.3%
	Manor Woods ES	154	22.6%
	Northfield ES	109	16.0%
		681	**100%**
Clarksville MS	Clarksville ES	500	68.9%
	Pointers Run ES	225	31.0%
		725	**100%**
Cradlerock MS	Cradlerock ES	233	50.6%
	Guilford ES	70	15.2%
	Jeffers Hill ES	99	21.5%
	Talbott Springs ES	58	12.6%
		460	**100%**
Dunloggin MS	Northfield ES	229	44.3%
	Thunder Hill ES	29	5.6%
	Veterans ES	258	50.0%
		516	**100%**
Elkridge Landing MS	Elkridge ES	408	65.0%
	Rockburn ES	219	34.9%
		627	**100%**
Ellicott Mills MS	Phelps Luck ES	7	1.0%
	Thunder Hill ES	90	13.6%
	Veterans ES	192	29.0%
	Waterloo ES	179	27.1%
	Worthington ES	192	29.0%
		660	**100%**
Folly Quarter MS	Bushy Park ES	81	13.8%
	Clarksville ES	0	0.0%
	Dayton Oaks ES	244	41.7%
	Triadelphia Ridge ES	260	44.4%
		585	**100%**
Glenwood MS	Bushy Park ES	320	48.4%
	Lisbon ES	340	51.5%
		660	**100%**
Hammond MS	Atholton ES	118	19.0%
	Fulton ES	120	19.3%
	Gorman Crossing ES	74	11.9%
	Hammond ES	309	49.7%
		621	**100%**
Harpers Choice MS	Clemens Crossing ES	66	12.0%
	Longfellow ES	204	37.2%
	Swansfield ES	278	50.7%
		548	**100%**
Lime Kiln MS	Dayton Oaks ES	302	47.1%
	Fulton ES	144	22.5%
	Pointers Run ES	194	30.3%
		640	**100%**

% of Middle to each High
Grades: 678

Bonnie Branch MS	Howard HS	552	81.4%
	Long Reach HS	126	18.5%
		678	**100%**
Burleigh Manor MS	Centennial HS	530	77.8%
	Marriotts Ridge HS	151	22.1%
		681	**100%**
Clarksville MS	Atholton HS	285	39.3%
	River Hill HS	440	60.6%
		725	**100%**
Cradlerock MS	Hammond HS	55	11.9%
	Oakland Mills HS	405	88.0%
		460	**100%**
Dunloggin MS	Centennial HS	186	36.0%
	Mt Hebron HS	190	36.8%
	Wilde Lake HS	140	27.1%
		516	**100%**
Elkridge Landing MS	Howard HS	370	59.0%
	Long Reach HS	257	40.9%
		627	**100%**
Ellicott Mills MS	Centennial HS	257	38.9%
	Howard HS	210	31.8%
	Mt Hebron HS	193	29.2%
		660	**100%**
Folly Quarter MS	Glenelg HS	245	41.8%
	Marriotts Ridge HS	63	10.7%
	River Hill HS	277	47.3%
		585	**100%**
Glenwood MS	Glenelg HS	660	100.0%
		660	**100%**
Hammond MS	Atholton HS	157	25.2%
	Hammond HS	187	30.1%
	Reservoir HS	277	44.6%
		621	**100%**
Harpers Choice MS	Wilde Lake HS	548	100.0%
		548	**100%**
Lime Kiln MS	Atholton HS	224	35.0%
	Reservoir HS	139	21.7%
	River Hill HS	277	43.2%
		640	**100%**
Mayfield Woods MS	Howard HS	0	0.0%
	Long Reach HS	586	90.0%
	Oakland Mills HS	65	9.9%
		651	**100%**
Mount View MS	Marriotts Ridge HS	716	100.0%
		716	**100%**
Murray Hill MS	Atholton HS	81	12.1%
	Hammond HS	12	1.7%
	Reservoir HS	576	86.1%
		669	**100%**

% of High from each Elementary
Grades: 9101112

Atholton HS	Bryant Woods ES	161	11.8%
	Clarksville ES	181	13.3%
	Clemens Crossing ES	270	19.9%
	Dayton Oaks ES	60	4.4%
	Fulton ES	3	0.2%
	Gorman Crossing ES	81	5.9%
	Hammond ES	194	14.3%
	Pointers Run ES	404	29.8%
		1354	**100%**
Centennial HS	Centennial Lane ES	594	39.6%
	Manor Woods ES	6	0.4%
	Northfield ES	363	24.2%
	Veterans ES	439	29.2%
	Waterloo ES	98	6.5%
		1500	**100%**
Glenelg HS	Bushy Park ES	531	43.5%
	Dayton Oaks ES	90	7.3%
	Lisbon ES	473	38.7%
	Triadelphia Ridge ES	126	10.3%
		1220	**100%**
Hammond HS	Atholton ES	209	15.9%
	Bollman Bridge ES	471	35.8%
	Deep Run ES	90	6.8%
	Forest Ridge ES	176	13.3%
	Guilford ES	262	19.9%
	Hammond ES	106	8.0%
		1314	**100%**
Howard HS	Elkridge ES	292	21.1%
	Ilchester ES	366	26.4%
	Jeffers Hill ES	23	1.6%
	Phelps Luck ES	188	13.5%
	Rockburn ES	249	18.0%
	Thunder Hill ES	113	8.1%
	Waterloo ES	152	10.9%
		1383	**100%**
Long Reach HS	Bellows Spring ES	238	18.8%
	Deep Run ES	316	25.0%
	Elkridge ES	184	14.5%
	Ilchester ES	35	2.7%
	Jeffers Hill ES	82	6.5%
	Phelps Luck ES	173	13.7%
	Rockburn ES	107	8.4%
	Waterloo ES	126	9.9%
		1261	**100%**
Marriotts Ridge HS	Manor Woods ES	469	36.3%
	Triadelphia Ridge ES	86	6.6%
	Waverly ES	358	27.7%
	West Friendship ES	377	29.2%
		1290	**100%**
Mt Hebron HS	Hollifield Station ES	367	26.0%
	St Johns Lane ES	569	40.4%
	Veterans ES	236	16.7%
	Worthington ES	235	16.7%
		1407	**100%**
Oakland Mills HS	Atholton ES	73	5.5%
	Bellows Spring ES	54	4.1%
	Cradlerock ES	326	24.9%

% of High from each Middle
Grades: 9101112

Atholton HS	Clarksville MS	354	26.1%
	Hammond MS	194	14.3%
	Lime Kiln MS	288	21.2%
	Murray Hill MS	81	5.9%
	Wilde Lake MS	437	32.2%
		1354	**100%**
Centennial HS	Burleigh Manor MS	786	52.4%
	Dunloggin MS	304	20.2%
	Ellicott Mills MS	410	27.3%
		1500	**100%**
Glenelg HS	Folly Quarter MS	351	28.7%
	Glenwood MS	869	71.2%
		1220	**100%**
Hammond HS	Cradlerock MS	84	6.3%
	Hammond MS	252	19.1%
	Murray Hill MS	22	1.6%
	Patuxent Valley MS	956	72.7%
		1314	**100%**
Howard HS	Bonnie Branch MS	683	49.3%
	Elkridge Landing MS	421	30.4%
	Ellicott Mills MS	276	19.9%
	Mayfield Woods MS	3	0.2%
		1383	**100%**
Long Reach HS	Bonnie Branch MS	165	13.0%
	Elkridge Landing MS	291	23.0%
	Mayfield Woods MS	805	63.8%
		1261	**100%**
Marriotts Ridge HS	Burleigh Manor MS	172	13.3%
	Folly Quarter MS	86	6.6%
	Mount View MS	1032	80.0%
		1290	**100%**
Mt Hebron HS	Dunloggin MS	235	16.7%
	Ellicott Mills MS	236	16.7%
	Patapsco MS	936	66.5%
		1407	**100%**
Oakland Mills HS	Cradlerock MS	598	45.7%
	Mayfield Woods MS	80	6.1%
	Oakland Mills MS	628	48.0%
		1306	**100%**
Reservoir HS	Hammond MS	434	30.2%
	Lime Kiln MS	154	10.7%
	Murray Hill MS	848	59.0%
		1436	**100%**
River Hill HS	Clarksville MS	599	44.1%
	Folly Quarter MS	366	26.9%
	Lime Kiln MS	391	28.8%
		1356	**100%**
Wilde Lake HS	Dunloggin MS	180	12.8%
	Harpers Choice MS	863	61.3%
	Wilde Lake MS	363	25.8%
		1406	**100%**

```
Atholton HS       Proj. Capacity:1332   Before Redistricting:Proj. Enrollment:1466  Proj. Util.:110.1%

                                                Plan ID  Gain     Loss
                      From Wilde Lake HS             66    17
                      From Wilde Lake HS            134    22
                      From Wilde Lake HS           1134    29
                              Total From Wilde Lake HS     68

                                Totals for Atholton HS     68        0     Net change:    68

                                       After Redistricting: Proj. Enrollment:1534  Proj. Util.:115.2%


Centennial HS     Proj. Capacity:1332   Before Redistricting:Proj. Enrollment:1503  Proj. Util.:112.8%

                              Totals for Centennial HS      0        0     Net change:     0

                                       After Redistricting: Proj. Enrollment:1503  Proj. Util.:112.8%


Glenelg HS        Proj. Capacity:1332   Before Redistricting:Proj. Enrollment:1248  Proj. Util.: 93.7%

                                                Plan ID  Gain     Loss
                      From River Hill HS            205    12
                      From River Hill HS            206    13
                      From River Hill HS            207    12
                      From River Hill HS            208     4
                      From River Hill HS           1205    50
                      From River Hill HS           1206    28
                      From River Hill HS           1207    30
                      From River Hill HS           1208    12
                      From River Hill HS           2205    23
                              Total From River Hill HS    184

                                 Totals for Glenelg HS    184        0     Net change:   184

                                       After Redistricting: Proj. Enrollment:1432  Proj. Util.:107.5%


Hammond HS        Proj. Capacity:1332   Before Redistricting:Proj. Enrollment:1274  Proj. Util.: 95.6%

                                 Totals for Hammond HS      0        0     Net change:     0

                                       After Redistricting: Proj. Enrollment:1274  Proj. Util.: 95.6%


Howard HS         Proj. Capacity:1332   Before Redistricting:Proj. Enrollment:1538  Proj. Util.:115.5%

                                                Plan ID  Gain     Loss
                      To   Long Reach HS             42            ( 52)
                              Total To   Long Reach HS            (  52)

                                  Totals for Howard HS      0     (  52)   Net change: (  52)

                                       After Redistricting: Proj. Enrollment:1486  Proj. Util.:111.6%


Long Reach HS     Proj. Capacity:1332   Before Redistricting:Proj. Enrollment:1309  Proj. Util.: 98.3%

                                                Plan ID  Gain     Loss
                      From Howard HS                 42    52
                                  Total From Howard HS     52

                      To   Oakland Mills HS          33            ( 11)
                      To   Oakland Mills HS          35            (  5)
                      To   Oakland Mills HS         266            ( 13)
                      To   Oakland Mills HS        1033            ( 35)
                      To   Oakland Mills HS        1266            ( 24)
                           Total To   Oakland Mills HS            (  88)
```

MS Redistricting Effects Summary for 2009 - 2020

Capital Improvement Plan in Use: SCHOOLS

Plan: base08 fs200813 fs200813 fs200813 fs200813 fs200813 fs200813 fs200813 fs200813 fs200813 fs200813 fs200813 fs200813

Bonnie Branch MS

	2009		2010		2011		2012		2013		2014		2015		2016		2017		2018		2019		2020		2021	
Cap:	662		662		662		662		662		662		662		662		662		662		662		662		662	
	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.
Enr:	701	701	689	689	700	700	681	681	702	702	694	694	718	718	705	705	725	725	731	731	750	750	764	764	774	774
Util:	105.9%	105.9%	104.1%	104.1%	105.7%	105.7%	102.9%	102.9%	106.0%	106.0%	104.8%	104.8%	108.5%	108.5%	106.5%	106.5%	109.5%	109.5%	110.4%	110.4%	113.3%	113.3%	115.4%	115.4%	116.9%	116.9%

Burleigh Manor MS

	2009		2010		2011		2012		2013		2014		2015		2016		2017		2018		2019		2020		2021	
Cap:	662		662		662		662		662		662		662		662		662		662		662		662		662	
	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.
Enr:	673	673	658	658	625	625	649	649	669	669	719	719	709	709	698	698	669	669	681	681	700	700	716	716	733	733
Util:	101.7%	101.7%	99.4%	99.4%	94.4%	94.4%	98.0%	98.0%	101.1%	101.1%	108.6%	108.6%	107.1%	107.1%	105.4%	105.4%	101.1%	101.1%	102.9%	102.9%	105.7%	105.7%	108.2%	108.2%	110.7%	110.7%

Clarksville MS

	2009		2010		2011		2012		2013		2014		2015		2016		2017		2018		2019		2020		2021	
Cap:	662		662		662		662		662		662		662		662		662		662		662		662		662	
	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.
Enr:	685	685	671	746	616	698	601	687	570	663	582	681	546	651	484	590	415	521	388	495	394	508	395	509	395	510
Util:	103.5%	103.5%	101.4%	112.7%	93.1%	105.4%	90.8%	103.8%	86.1%	100.2%	87.9%	102.9%	82.5%	98.3%	73.1%	89.1%	62.7%	78.7%	58.6%	74.8%	59.5%	76.7%	59.7%	76.9%	59.7%	77.0%

Cradlerock MS

	2009		2010		2011		2012		2013		2014		2015		2016		2017		2018		2019		2020		2021	
Cap:	584		584		584		584		584		584		584		584		584		584		584		584		584	
	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.
Enr:	457	457	465	465	492	492	486	486	492	492	481	481	495	495	501	501	498	498	488	488	476	476	475	475	475	475
Util:	78.3%	78.3%	79.6%	79.6%	84.2%	84.2%	83.2%	83.2%	84.2%	84.2%	82.4%	82.4%	84.8%	84.8%	85.8%	85.8%	85.3%	85.3%	83.6%	83.6%	81.5%	81.5%	81.3%	81.3%	81.3%	81.3%

Dunloggin MS

	2009		2010		2011		2012		2013		2014		2015		2016		2017		2018		2019		2020		2021	
Cap:	526		526		526		526		526		526		526		526		526		526		526		526		526	
	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.
Enr:	501	501	511	511	535	535	533	533	558	558	578	578	595	595	607	607	585	585	582	582	553	553	555	555	559	559
Util:	95.2%	95.2%	97.1%	97.1%	101.7%	101.7%	101.3%	101.3%	106.1%	106.1%	109.9%	109.9%	113.1%	113.1%	115.4%	115.4%	111.2%	111.2%	110.6%	110.6%	105.1%	105.1%	105.5%	105.5%	106.3%	106.3%

Elkridge Landing MS

	2009		2010		2011		2012		2013		2014		2015		2016		2017		2018		2019		2020		2021	
Cap:	662		662		662		662		662		662		662		662		662		662		662		662		662	
	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.	BEF.	AFT.
Enr:	683	683	664	664	682	682	688	688	766	766	801	801	811	811	801	801	810	810	827	827	840	840	858	858	851	851
Util:	103.2%	103.2%	100.3%	100.3%	103.0%	103.0%	103.9%	103.9%	115.7%	115.7%	121.0%	121.0%	122.5%	122.5%	121.0%	121.0%	122.4%	122.4%	124.9%	124.9%	126.9%	126.9%	129.6%	129.6%	128.5%	128.5%

Schools and Capacities Currently on File in the Whiffer

	Excluding Kindergarten													Including Kindergarten												
Name	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
Atholton ES	321	321	321	321	321	321	321	321	321	321	321	321	321	387	387	387	387	387	387	387	387	387	387	387	387	387
Bellows Spring ES	552	552	552	552	552	552	552	552	552	552	552	552	552	662	662	662	662	662	662	662	662	662	662	662	662	662
Bollman Bridge ES	478	478	478	478	478	478	478	478	478	478	478	478	478	566	566	566	566	566	566	566	566	566	566	566	566	566
Bryant Woods ES	289	289	289	289	289	289	289	289	289	289	289	289	289	355	355	355	355	355	355	355	355	355	355	355	355	355
Bushy Park ES	678	678	678	678	678	678	678	678	678	678	678	678	678	788	788	788	788	788	788	788	788	788	788	788	788	788
Centennial Lane	540	540	540	540	540	540	540	540	540	540	540	540	540	628	628	628	628	628	628	628	628	628	628	628	628	628
Clarksville ES	546	546	546	546	546	546	546	546	546	546	546	546	546	634	634	634	634	634	634	634	634	634	634	634	634	634
Clemens Crossing	434	434	434	434	434	434	434	434	434	434	434	434	434	522	522	522	522	522	522	522	522	522	522	522	522	522
Cradlerock ES	421	421	421	421	421	421	421	421	421	421	421	421	421	487	487	487	487	487	487	487	487	487	487	487	487	487
Dayton Oaks ES	678	678	678	678	678	678	678	678	678	678	678	678	678	788	788	788	788	788	788	788	788	788	788	788	788	788
Deep Run ES	491	491	491	491	491	491	491	491	491	491	491	491	491	601	601	601	601	601	601	601	601	601	601	601	601	601
Elkridge ES	647	647	647	647	647	647	647	647	647	647	647	647	647	779	779	779	779	779	779	779	779	779	779	779	779	779
Forest Ridge ES	516	516	516	516	516	516	516	516	516	516	516	516	516	626	626	626	626	626	626	626	626	626	626	626	626	626
Fulton ES	640	640	640	640	640	640	640	640	640	640	640	640	640	772	772	772	772	772	772	772	772	772	772	772	772	772
Gorman Crossing	452	452	452	452	452	452	452	452	452	452	452	452	452	540	540	540	540	540	540	540	540	540	540	540	540	540
Guilford ES	396	396	396	396	396	396	396	396	396	396	396	396	396	462	462	462	462	462	462	462	462	462	462	462	462	462
Hammond ES	434	434	434	434	434	434	434	434	434	434	434	434	434	500	500	500	500	500	500	500	500	500	500	500	500	500
Hollifield	578	578	578	578	578	578	578	578	578	578	578	578	578	688	688	688	688	688	688	688	688	688	688	688	688	688
Ilchester ES	529	529	529	529	529	529	529	529	529	529	529	529	529	617	617	617	617	617	617	617	617	617	617	617	617	617
Jeffers Hill ES	377	377	377	377	377	377	377	377	377	377	377	377	377	421	421	421	421	421	421	421	421	421	421	421	421	421
Laurel Woods ES	452	452	452	452	452	452	452	452	452	452	452	452	452	540	540	540	540	540	540	540	540	540	540	540	540	540
Lisbon ES	465	465	465	465	465	465	465	465	465	465	465	465	465	553	553	553	553	553	553	553	553	553	553	553	553	553
Longfellow ES	352	352	352	352	352	352	352	352	352	352	352	352	352	418	418	418	418	418	418	418	418	418	418	418	418	418
Manor Woods ES	559	559	559	559	559	559	559	559	559	559	559	559	559	647	647	647	647	647	647	647	647	647	647	647	647	647
Northfield ES	434	584	584	584	584	584	584	584	584	584	584	584	584	522	672	672	672	672	672	672	672	672	672	672	672	672
Phelps Luck ES	452	452	452	452	452	452	452	452	452	452	452	452	452	540	540	540	540	540	540	540	540	540	540	540	540	540
Pointers Run ES	666	666	666	666	666	666	666	666	666	666	666	666	666	776	776	776	776	776	776	776	776	776	776	776	776	776
Rockburn ES	535	535	535	535	535	535	535	535	535	535	535	535	535	667	667	667	667	667	667	667	667	667	667	667	667	667
Running Brook ES	339	339	339	339	339	339	339	339	339	339	339	339	339	405	405	405	405	405	405	405	405	405	405	405	405	405
St Johns Lane ES	509	509	509	509	509	509	509	509	509	509	509	509	509	597	597	597	597	597	597	597	597	597	597	597	597	597
Stevens Forest ES	289	289	289	289	289	289	289	289	289	289	289	289	289	333	333	333	333	333	333	333	333	333	333	333	333	333
Swansfield ES	440	440	440	440	440	440	440	440	440	440	440	440	440	528	528	528	528	528	528	528	528	528	528	528	528	528
Talbott Springs	377	377	377	377	377	377	377	377	377	377	377	377	377	443	443	443	443	443	443	443	443	443	443	443	443	443
Thunder Hill ES	302	302	302	302	302	302	302	302	302	302	302	302	302	368	368	368	368	368	368	368	368	368	368	368	368	368
Triadelphia Ridge	434	434	434	434	434	434	434	434	434	434	434	434	434	544	544	544	544	544	544	544	544	544	544	544	544	544
Veterans ES	678	678	678	678	678	678	678	678	678	678	678	678	678	788	788	788	788	788	788	788	788	788	788	788	788	788
Waterloo ES	484	484	484	484	484	484	484	484	484	484	484	484	484	594	594	594	594	594	594	594	594	594	594	594	594	594
Waverly ES	665	665	665	665	665	665	665	665	665	665	665	665	665	775	775	775	775	775	775	775	775	775	775	775	775	775
West Friendship	302	302	302	302	302	302	302	302	302	302	302	302	302	368	368	368	368	368	368	368	368	368	368	368	368	368
Worthington ES	403	403	403	403	403	403	403	403	403	403	403	403	403	491	491	491	491	491	491	491	491	491	491	491	491	491

Schools and Capacities Currently on File in the Whiffer

Name	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
Bonnie Branch MS	662	662	662	662	662	662	662	662	662	662	662	662	662
Burleigh Manor MS	662	662	662	662	662	662	662	662	662	662	662	662	662
Clarksville MS	662	662	662	662	662	662	662	662	662	662	662	662	662
Cradlerock MS	584	584	584	584	584	584	584	584	584	584	584	584	584
Dunloggin MS	526	526	526	526	526	526	526	526	526	526	526	526	526
Elkridge Landing	662	662	662	662	662	662	662	662	662	662	662	662	662
Ellicott Mills MS	662	662	662	662	662	662	662	662	662	662	662	662	662
Folly Quarter MS	662	662	662	662	662	662	662	662	662	662	662	662	662
Glenwood MS	584	584	584	584	584	584	584	584	584	584	584	584	584
Hammond MS	584	584	584	584	584	584	584	584	584	584	584	584	584
Harpers Choice MS	506	506	506	506	506	506	506	506	506	506	506	506	506
Lime Kiln MS	701	701	701	701	701	701	701	701	701	701	701	701	701
Mayfield Woods MS	682	682	682	682	682	682	682	682	682	682	682	682	682
Mount View MS	662	662	662	662	662	662	662	662	662	662	662	662	662
Murray Hill MS	662	662	662	662	662	662	662	662	662	662	662	662	662
New Western MS	0	0	0	0	662	662	662	662	662	662	662	662	662
Oakland Mills MS	506	506	506	506	506	506	506	506	506	506	506	506	506
Patapsco MS	662	662	662	662	662	662	662	662	662	662	662	662	662
Patuxent Valley	662	662	662	662	662	662	662	662	662	662	662	662	662
Wilde Lake MS	506	506	506	506	506	506	506	506	506	506	506	506	506

Schools and Capacities Currently on File in the Whiffer

Name	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
Atholton HS	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332
Centennial HS	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332
Glenelg HS	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332
Hammond HS	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332
Howard HS	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332
Long Reach HS	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332
Marriotts Ridge	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332
Mt Hebron HS	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332
Oakland Mills HS	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332
Reservoir HS	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332
River Hill HS	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332
Wilde Lake HS	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332	1332