Report No. CSS06-16
November 2, 2006

Center for Sustainable Systems
University of Michigan

A Technical Update on Environmental Reporting for the University of Michigan Ann Arbor Campus:

The U-M Environmental Data Repository

Arthur Chan, Gregory A. Keoleian and Helaine Hunscher

A Technical Update on Environmental Reporting for the University of Michigan Ann Arbor Campus:

The U-M Environmental Data Repository

Arthur Chan
Gregory A. Keoleian
Helaine Hunscher

Center for Sustainable Systems

University of Michigan
Ann Arbor

November 2, 2006

A report of the Center for Sustainable Systems

Report No. CSS06-16

Document Description

A TECHNICAL UPDATE ON ENVIRONMENTAL REPORTING FOR THE UNIVERSITY OF MICHIGAN ANN ARBOR CAMPUS: THE U-M ENVIRONMENTAL DATA REPOSITORY
Arthur Chan, Gregory A. Keoleian and Helaine Hunscher
Center for Sustainable Systems, Report No. CSS06-16
University of Michigan, Ann Arbor, Michigan
November 2, 2006
18 pp., 7 tables, 2 appendices

This document is available online at: http://css.snre.umich.edu

Center for Sustainable Systems
School of Natural Resources and Environment
University of Michigan
440 Church Street, Dana Building
Ann Arbor, MI 48109-1041
Phone: 734-764-1412
Fax: 734-647-5841
Email: css.info@umich.edu
Web: http://css.snre.umich.edu

Acknowledgements

This update to the Environmental Data Repository (EDR) was made possible with funding from the Rosina Bierbaum, Dean of the School of Natural Resources and Environment. The update includes both improvements to the Visual Basic code upon which the EDR software tool is built, and the output of environmental performance results for the University of Michigan based on available 2005 data.

The Advisory group for the EDR project included Professors Jonathan Bulkley and Gregory Keoleian from SNRE, and Terry Alexander (Director) and Andrew Berki (Coordinator for Environmental Services) from Occupational Safety and Environmental Health.

In addition, this update would not be possible without the contribution of the following University staff to the project:

Tracy Artley, Waste Management Services
Mike Bowen, Utilities/Plant - Mechanical Engineering
Andy Cameron, Office of Registrar
Brandi Campbell, Occupational Safety and Environmental Health
Shelly Carpenter, Utilities
Jim Day, Property Disposition
Rob Doletzky, Grounds & Waste Management Services
Doug Hanna, University Architect/Planner Office & Plant Extension
Kandie Hines, Office of Budget and Planning
Mike Hommel, Botanical Gardens & Arboretum
Tracey Jones, Intercollegiate Athletics/Radrick Farms
Renee Jordan, Parking & Transportation Services
John Lawter, Grounds
Kevin Marsh, Hospital Safety and Management Services
William McAllister, Grounds
Dave Miller, Parking & Transportation Services
Marvin Pettway, Grounds & Waste Mgmt Services
Terry Ramsey, Plant Extension
Kenn Rapp, Plant Landscape Arch
Becky Seiser, University Parking Services
Mike Swanson, Utilities
David Tyler, University of Michigan Hospital Waste Management
Carol Varney, Utilities

Table of Contents

Introduction

Environmental Reporting for University of Michigan Ann Arbor Campus: The U-M Environmental Data Repository (Report Number CSS05-11) resulted from a project conducted by Sarah Deslauriers, Colin McMillan and David Spitzley in 2005 in response to the *Environmental Task Force Advisory Report* (2004) urging the University to track and report environmental performance in a on-going and systematic basis. As a result, the Environmental Data Repository (EDR) was developed. It is a Microsoft Excel-based software program that facilitates data collection, computation and tracking of the eight key environmental performance indicators and more than 50 operational metrics recommended by the *Advisory Report*.

This UM environmental reporting is to be conducted annually to track the UM environmental performance over time. Such information can be distributed to the UM community to inform the University's daily decision-making processes and to demonstrate the University's commitment to sustainability issues. The project was resumed in May 2006 with the following goals:

1) Collect the same set of environmental-related data for calendar/fiscal year 2005 as was reported for 2004, so that data are consistent and can be compared with each other.
2) Review data collection sources.
3) Examine the applicability of EDR model to 2005 data, and update the model to accommodate changes in data sources and formats.
4) Explore the possibility of collecting additional environmental attributes from departments across the campus.
5) Develop factsheets[1] to facilitate environmental performance communication among the UM populations, in order to make an impact in decision-making processes.

With support from operations departments across the campus, we were able to collect most of the necessary environmental data for calendar/fiscal year 2005.

[1] Four factsheets (Environmental Data Repository; Energy; Emissions; and Waste) are developed.

Environmental Data Collection

The sets of environmental data are primarily obtained from over 21 UM staff working in 11 departments.[2] With the help of Andy Berki from Occupational Safety and Environmental Health (OSEH), contacts were initiated with the personnel who supplied data last year.

With the co-operation and support of the contacted departments across the campus, we were able to collect almost all of the environmental data in one-month time. Most of the data can be collected from the same personnel or department as last year, and the details and formats of the data were similar as well.

Energy – Buildings and Transportation

This category includes two key performance indicators (total energy consumption, renewable energy contribution) and 13 operational metrics. For building energy consumption, the primary data were mainly collected from the personnel in Utilities and Plant Engineering. For transportation energy consumption, data were acquired from Parking & Transportation Services and the Plant Extension.

Mr. Michael Swanson (mswaney@umich.edu, 763-3011) is the major contact person in the Utilities and Plant Engineering, with his colleagues Ms. Carol Varney and Ms. Shelly Carpenter respectively providing the Annual Report of Utilities and energy consumption data of Central Power Plant and Hoover Boiler Plant. Water consumption data were requested at the same time. The Annual Report of Utilities is also available in PDF format at http://www.plantops.umich.edu/ utilities/Utilities/reports.html, but a report in Excel format is needed to be imported to the EDR. There were no major issues with the data requested as they were provided in the same format as last year.

There were energy and water consumption data from new buildings identified in the Annual Report of Utilities. Four buildings were located within the UM-Ann Arbor campus area and therefore were included in our analysis. To incorporate the new data, the code and name of the four new buildings were added into the "BldgList" worksheet of the EDR model. The additions were highlighted in yellow in the "BldgList" worksheet.

[2] Department of Occupational Safety and Environmental Health
Grounds & Waste Management Services
Matthaei Botanical Gardens and Nichols Arboretum
Office of Registrar
Parking and Transportation Services
Plant Extension
Property Disposition
UM Athletic Department
UM Hospital Waste Management/ Recycling Foreman
UM Office of Budget and Planning
Utilities and Plant Engineering

Data on electricity generated with a renewable source was originally provided by Mr. Andy Berki from OSEH. It was suggested that no electricity is generated with renewable sources on campus. Yet, photovoltaic (PV) panels are installed on the roof top of the Dana Building. David Spitzley of Center for Sustainable Systems provided an estimate on electricity generated by the PV panels in 2005.[3]

Mr. David Miller (dvmiller@umich.edu, 647-0948) was the primary contact person in Parking and Transportation Services. He shared with us the AATA M-ride ridership data, and his colleague Ms. Renee Jordan filled out the "TransportationStats" template with data about bus and fleet fuel consumption, M-bus ridership and vanpool ridership. Not tracked in 2004, fleet fuel economy data is not tracked in 2005 as well. The "TransportationStats" template designed last year was used this year without any major modification. There were no issues with the data.

Mr. Terrence Ramsey (tramsey@bf.umich.edu, 260-2959) from the Plant Extension was contacted directly for bike rack maps of each campus (Central, Medical, South and North). Secondary data were obtained from the Annual Energy Review and EIA 906/920 Monthly Time Series File published by the Energy Information Administration as noted last year. Updated data were available from EIA 906 and EIA 920 database for the heat content of different energy carriers and the fuel mix of purchased electricity. However, the same sets of assumptions or data were used for the heating values of different fuel types because there were no updated versions of the source documents.

Water Use

This category is comprised of one key performance indicator (total water consumption) and four operational metrics. The majority of data (water use for UM buildings, CPP and Hoover Boiler Plant) were collected along with energy consumption data requested from the Utilities and Plant Engineering. Irrigation water data for the golf courses were previously collected from MDEQ. However, the person from MDEQ (Mr. Ron van Til) has retired. Personnel from Athletic Department asserted that the 2005 data is the same as 2004.

There were no updated data regarding the percentages of purchased water consumption for irrigation and discharges to sewers. Therefore, the same sets of data were used. For the four new buildings identified in the Annual Report of Utilities, it was assumed that all the water consumed would end up in the sewer system.

[3] 35,000 kWh in 2005. On-site renewable power represents the 12-month average generation for the installed photovoltaic system. Average is taken from 16 months of acquired operating data as reported by the data acquisition system in June of 2006.

Land Use – Built and Natural Spaces

One performance indicator (impervious surface area) and seven operational metrics were chosen to measure the University's relative impacts on storm water runoff and land resource management. Data are respectively collected from three different departments within the University for the areas that the University owns and maintains: 1) campus Grounds, 2) UM golf courses, and 3) nature preserves (Matthaei Botanical Gardens, Nichols Arboretum, Horner's Woods and Mud Lake Bog). **However, the current model fails to capture all the land owned by the University within Ann Arbor.**

Grounds and Waste Management Services provided the data of UM campus grounds. Mr. Kenneth Rapp (kennr@bf.umich.edu, 647-2028) was contacted for the campus land use inventory. The inventory did not include the athletic campus nor the 200-acre East Medical Campus on Plymouth and Earhart Roads. Requested through Mr. John Lawter (jlawter@umich.edu, 764-3422), we respectively received the tree population and salt & sand use data from Mr. Marvin Pettway and Mr. Robert Doletzky from the Grounds Operations. No issues were found with the data.

The Athletics Department was contacted for land use data for the athletic fields and University-owned golf courses. However, it was advised that 2004 data can be used for 2005. The same situation applied to the land use data for the nature preserves (e.g., the Nichols Arboretum).

There are serious issues with the data provided by the Athletics Department. In a document provided by Ms. Tracey Jones last year, the size of the two golf courses is 169 acres. However, there are two sources citing that the size of the Radrick Farm Golf Course alone is 260-270 acres.[4] Also, the data do not include the size of the athletic fields. The Athletics Department did not return to us an updated figure. Through email communications Mr. Kenneth Rapp, of Grounds, suggested that the athletic campus spans over 1,000 acres.

The deck parking data were obtained from the same person (Ms. Rebecca Seiser, rseiser@bf.umich.edu, 647-3615) as last year. No issues were noticed.

Overall, the current model fails to capture all the land owned by the University within Ann Arbor. The model only includes the acreage in the nature preserves, part of the golf courses and the land maintained by the Grounds Department. These land uses contribute to only around 1,900 acres out of 3,070 acres of land[5] owned by the University in Ann Arbor. Office of Budget and Planning, who published the 3,070-acre figure in its Facilities factsheet, and Mr. Rapp from the Grounds Department were contacted regarding the discrepancy. There were no conclusive answers on the issue, but both advised that the majority of the remaining land parcels should be located within the athletic campus, which is maintained by the Athletic Department.

[4] http://www-personal.umich.edu/~luriea/intro.html, http://www.umich.edu/~radrick/infoguide.html

[5] http://sitemaker.umich.edu/obpinfo/files/umaa_facilities.pdf

Emissions – Air and Water Pollutants

In this category, one key performance indicator (total greenhouse gas emissions) and five operational metrics are included. The data are calculated from the fuel consumption data of the University's fleet, CPP, Hoover Power Plant and other services. The fuel consumption data of the fleet were available once "TransportationStats" template was imported into the model mentioned in the energy section. Data are provided by the same personnel as last year's and no issues have been found with the data.

The air emissions data from on-site combustion were requested through Ms. Brandi Campbell (campbelb@umich.edu, 647-9017) from the Department of Occupational Safety and Environment (OSEH). The template "StationaryEmissions.xls" was sent to Ms. Campbell to fill out the natural gas and distilled oil usage data for CPP, Hoover Boiler Plant and other sources. The template calculates the air emissions from these sources with detailed assumptions on various air pollutants emission factors. No issues were identified with the data.

This year the same carbon or carbon dioxide emission factors for all fuels were used (see SecondaryInput sheet). An updated version of the document "Emissions of Greenhouse Gases in the United States" was not available at the Energy Information Administration website as of June 2006.

Salt and sand use data were collected as mentioned in the previous section (Mr. John Lawter jlawter@umich.edu, 764-3422). A spreadsheet with tables and graphs showing the salt and sand use data for the past 17 years was received. There were no issues with the data. Fertilizers and pesticides data were not available.

Material Use and Solid Waste

Two performance and 20 operational metrics are reported for the material use and solid waste category. Contacts were initiated with five departments within the University to collect the data.

Data on regulated recycling items, reuse property disposition and woody debris were collected from the same personnel as last year. Mr. Andy Berki from OSEH supplied the updated data on regulated recycling items. Mr. Jim Day (dayj@umich.edu, 764-2470) at the Office of Financial Analysis shared the property disposition data in dollar terms. Mr. William McAllister from Grounds and Waste Management Services provided the data on the amount of woody debris collected during ground maintenance activities. The assumption on woody debris density was not changed (170lb/cubic yard).

Refuse and recycling data of the UM Hospital were collected this year from Mr. Kevin Marsh (kevmarsh@umich.edu, 764 4427) at UMH Safety Management Services.

Ms. Tracy Artley from the Waste Management Services was contacted for composting, recycling and reuse data. With an updated database at the Waste Management Services, a new template

“WMS_Input.xls” and an instruction sheet “WMS_Input Instructions.doc” were written (Appendix B) and provided to enter the data. New queries were written and stored in the MS Access database at the department’s computers. In the future, Ms. Artley and her colleagues can collect the data for the project by simply clicking and running pre-written queries from their computers. The changes in the code and worksheet structures will be discussed in the “ERD Model Improvement” section. Ms. Artley helped us request the data on the composition of the mixed containers for the City of Ann Arbor through Mr. Bill Leonidas from FCR Recycling.

Cross-Cutting and Emerging Issues

Two new metrics (number of undergraduate, masters & doctoral degrees, and enrollment of 429 sustainability-related courses) were added. The former metric was obtained from the Graham Environmental Sustainability Institute (GESI) http://provost.umich.edu/gesi/academics/degree.html . For the latter metric, the lists of courses are defined by GESI, and the enrollment data were requested through the Office of Registrar. The 2005 data included the enrollment of the sustainability-related courses offered in Fall 2004 and/or Winter 2005. Data for the metrics used in the original report can be requested from the same personnel as last year.

Summary of Contacts for 2005 data

Department	Email	Phone	Data
Athletic Department			
Ms. Tracey Jones	tracmon@umich.edu	998-7040	Land use data for athletic golf courses
Center for Sustainable Systems			
Mr. David Spitzley (left U-M in July 2006)	spitz@umich.edu	764-1412	Electricity generated by PV panels in Dana Building
Dept of Occupational Safety and Environment (OSEH)			
Mr. Andy Berki	aberki@umich.edu	647-3120	Generated electricity from renewable source; Regulated recycling data
Ms. Brandi Campbell	campbelb@umich.edu	647-9017	Air pollutant emissions of on-site fuel combustion
Grounds and Waste Management Services			
- Grounds Operations			
Mr. John Lawter	jlawter@umich.edu	764-3422	
Mr. Robert Doletzky	doletzky@umich.edu	764-3537	Salt and sand use data
Mr. Marvin Pettway	marvinp@umich.edu	764-3422	Tree population on campus
- Landscape Architect			
Mr. Kenneth Rapp	kennr@umich.edu	647-2028	Land Use Inventory of campus
- Moving and Trucking Dept			
Mr. William McAllister	bjom@umich.edu	764-3424	Mulch data

Continued on next page

- *Waste Management Services*			
Ms. Tracy Artley	artleyt@umich.edu	763-5539	WMS_Input (Compost, reuse and recycling data excluding hospital); Helped collect container mix data from FCR Recycling
Matthaei Botanical Gardens & Nichols Arboretum			
Mr. Michael Hommel	mhommel@umich.edu	647-8062	Land Use data for UM-owned nature areas (person unable to reach)
Office of Budget and Planning			
Ms. Kandie Hines	khines@umich.edu	647-3778	Land Use data within and outside Ann Arbor
Office of Financial Analysis			
Mr. James Day	dayj@umich.edu	764-2470	Property Disposition data
Office of Registrar			
Mr. Andy Cameron	umich_regoff@mailnj.custhelp.com		Enrollment of sustainability-related courses
Parking and Transportation Services			
- *Transportation Services*			
Mr. David Miller	dvmiller@umich.edu	764-7529	AATA ridership
Ms. Renee Jordan			TransportationStats
- *Parking Services*			
Ms. Rebecca Seiser	rseiser@umich.edu	764-8291	Deck parking
Plant Extension			
- *University Architect's Office*			
Mr. Douglas Hanna	dougha@umich.edu	764-2456	LEED certification; Aesthetic awards
- *University Planner's Office*			
Mr. Terence Ramsey	tramsey@bf.umich.edu	260-2959	Bike rack maps of campuses
UM Hospital Safety Management Services			
Mr. Kevin Marsh	kevmarsh@umich.edu	764-4427	Recycling and refuse data for hospital
Utilities and Plant Engineering			
- *Plant Engineering Team*			
Mr. Michael Bowen	mbowen@umich.edu	647-5797	FCA data
- *Utilities Service Team*			
Mr. Michael Swanson	mswaney@umich.edu	763-3011	
Ms. Carol Varney			Annual Report of Utilities sorted by buildings
Ms. Shelly Carpenter			CPP; Hoover Plant fuel and water consumption
External Sources			
EIA 906 & 920 database, and Annual Energy Review			Heat content of fuels; Fuel mix of electricity
EPA			Carbon emission factors and higher heating values of fuels
Bill Leonidas@FCR Recycling (contacted through Tracy Artley)			Michigan mix for commingled containers

Summary of Environmental Performance Data

NOTE: +/- symbols in the 'Changes' column of the following tables simply indicate whether there was a measured increase or decrease in the result of that indicator, without implying an improvement or decline in sustainability, e.g., an increase in energy consumption indicated by a '+' symbol might actually suggest a decline in sustainability performance. To clarify, trends showing improved sustainability performance have been marked in green.

Energy – Buildings and Transportation

	2004	2005	Changes
Total Energy Consumption (Bbl/person)	**18.03**	**18.31**	+
Total Energy Consumption (Btu)	**7.6025E+12**	**7.84296E+12**	+
Total Energy Consumption (Btu/person)	**104,563,527**	**106,205,513**	+
Percentage Renewable Energy	**0.30%**	**0.26%**	-
AATA Bus Ridership	851,000	1,676,211	+++
Bicycle Ridership (Number of bike racks)	3,366	3,410	+
Building Energy (Btu)	7.50E+12	7.74E+12	+
Building Energy (Btu/ft^2)	271,233	278,216	+
Building Energy (Btu/ft^2/person)	3.73	3.77	+
Building Energy (Btu/person)	103,175,762	104,809,671	+
Bus Energy Consumption (Btu)	3.52E+10	3.81E+10	+
Bus Energy Consumption (Btu/passenger)	7,488	7,347	-
Campus Bus Ridership (Total passengers)	4,702,261	5,187,602	+
Generated Electricity Consumption (MWh)	238,248	249,959	+
Purchased Electricity from Renew. Sources (MWh)	1,211	321	--
Purchased Electricity (MWh)	242,176	247,280	+
Fleet Vehicles Energy Consumption (Btu)	6.57E+10	6.50E+10	-
Renewable Percentage for Transportation Energy (%)	0.18	0.18	~
Transportation Energy Consumption (Btu)	1.01E+11	1.03E+11	+
Transportation Energy Consumption (Btu/person)	1,387,765	1,395,842	+
Van Pooling (Passenger miles)	3,094,729	5,712,293	++
Van Pooling (Vehicle miles)	471,210	894,017	++
Purchased Electricity from Renew. Sources (%)	0.005	0.0013	--
Generated Electricity from Renew. Sources (%)	0	0.00014	~

Water Use

	2004	2005	Changes
Total Water Use (gal)	**1,237,461,853**	**1,313,372,834***	+
Total Water Use (gal/person)	**17,020**	**17,785**	+
Building Specific Water Use (gal)	1,016,409,438	1,107,176,730	+
Building Specific Water Use (gal/person)	13,980	14,993	+
Total Discharge to Sewers (gal)	902,032,593	989,441,187	+
Total Discharge to Sewers (gal/person)	12,406	13,399	+
Total Irrigation Water (gal)	149,195,063	152,553,694*	+
Total Purchased Water (gal)	1,202,643,703	1,278,554,684	+
Total Purchased Water (gal/person)	16,541	17,314	+
Total Irrigation Water (gal/acre green space)	249,116	220,399	incomplete

* it is assumed that the golf course irrigation data in 2005 is the same as that in 2004

Land Use – Built and Natural Spaces

	2004	2005	Changes
Total Impervious Surface Area (% of campus area)	**22.34%**	**21.66%**	**-**
Total Impervious Surface Area (acres)	**409**	**419**	+
Deck Parking (# of deck parking spots)	9,492	10,197	+
Deck Parking (% of parking spaces that are parking decks)	40.82%	43.72%	+
Maintained Green Space (% of campus area)	34.12%	37.13%	+
Maintained Green Space (acres)	625	718	+
Total Building Area (ft^2)	27,657,452	27,819,689	+
Total Building Area (ft^2/person)	380	377	-
Total Green Space (% of campus area)	77.66%	78.34%	+
Total Green Space (acres)	1,422	1,515	+
Total Land Area (acres)	1,831	1,934	+
Tree Population	7,198	7,198	~
Un-maintained (Natural) Green Space (% of campus area)	43.54%	41.21%	-
Un-maintained (Natural) Green Space (acres)	797	797	~

N.B. same estimates were used for golf courses and Matthaei Botanical Gardens & the Nichols Arboretum in 2005. The data do not include other land owned by the Athletic Department

Emissions – Air and Water Pollutants

	2004	2005	Changes
Total Greenhouse Gas Emissions (MTCE)	**129,557**	**133,406**	+
Total Greenhouse Gas Emissions (MTCE/person)	**1.78**	**1.81**	+
Total Greenhouse Gas Emissions (MTCO2E)	**474,981**	**489,093**	+
Total Greenhouse Gas Emissions (MTCO2E/person)	**6.53**	**6.62**	+
Mobile Sources: Renewable GHG Emissions (MTCO2E)	1,220	1,260	+
Mobile Sources: Fossil GHG Emissions (MTCO2E)	5,834	5,951	+
Salt Use (lbs)	3,026,000	4,012,000	+
Salt Use (lbs/person)	41.62	54.33	+
Sand Use (lbs)	20,000	0	---
Sand Use (lbs/person)	0.28	0.00	---
Stationary Sources: CO (lbs)	287,068	354,765	+
Stationary Sources: CO (lbs/person)	3.95	4.80	+
Stationary Sources: Greenhouse Gas Emissions (MTCE)	127,949	131,766	+
Stationary Sources: Greenhouse Gas Emissions (MTCO2E)	469,147	483,142	+
Stationary Sources: NOx (lbs)	651,752	699,011	+
Stationary Sources: NOx (lbs/person)	8.96	9.47	+
Stationary Sources: Pb (lbs)	3	3	~
Stationary Sources: Pb (lbs/person)	0	0	~
Stationary Sources: PM-10 (lbs)	27,237	27,813	+
Stationary Sources: PM-10 (lbs/person)	0.37	0.38	+
Stationary Sources: PM-2.5 (lbs)	26,808	27,743	+
Stationary Sources: PM-2.5 (lbs/person)	0.37	0.38	+
Stationary Sources: SO2 (lbs)	6,842	3,165	--
Stationary Sources: SO2 (lbs/person)	0.09	0.04	--
Stationary Sources: Volatile Organic Compounds (lbs)	29,068	35,147	+
Stationary Sources: Volatile Organic Compounds (lbs/person)	0.40	0.48	+
Mobile Sources: Renewable GHG Emissions (MTCE)	337	348	+
Mobile Sources: Fossil GHG Emissions (MTCE)	1,608	1,640	+

Material Use and Solid Waste

	2004	2005	Changes
Total Solid Waste (tons)	**13,032**	**16,036**	+
Total Solid Waste (tons/person)	**0.18**	**0.22**	+
% Recycled	**28.26%**	**28.54%**	+
Composting (tons)	238	219	-
Composting (tons/person)	0.02116	0.01867	-
Construction In-house Waste Recycled (%)	9.51%	11.31%	+
Construction In-house Waste Recycled (tons)	153	150	-
Regulated Recycling- Batteries (tons)	17.23	24.7	+
Regulated Recycling- Lamp Ballasts (tons)	17.87	31.25	+
Regulated Recycling- Fluorescent Bulbs (tons)	35.05	86.04	+
Regulated Recycling- Consumer Electronics (tons)	0	55.03	incomplete
Regulated Recycling- Solvents (gallons)	0	6120	incomplete
Regulated Recycling- Transportation Oil (gallons)	0	1150	incomplete
Regulated Recycling- Latex Paint (gallons)	0	374	incomplete
Regulated Recycling- Coolants (gallons)	0	1270	incomplete
Glass Containers Recycled (% of total waste)	0.97%	0.67%	-
Glass Containers Recycled (tons)	126	107	-
Hospital Waste (tons)	3,111	5,733	+
Hospital Waste (tons/person)	0.28	0.49	+
Metal Containers Recycled (% of total waste)	0.12%	0.13%	+
Metal Containers Recycled (tons)	15.72	20.21	+
Paper Recycled (% of total waste)	22.38%	22.94%	+
Paper Recycled (tons)	2,917	3,679	+
Plastics Recycled (% of total waste)	0.31%	0.21%	-
Plastics Recycled (tons)	40.37	34.39	-
Reuse (tons)	2.88	1.83	-
Reuse (tons/person)	3.95E-05	2.48E-05	-
Reuse-PD ($)	$1,300,000	$1,400,000	+
Reuse-PD ($/person)	$17.88	$18.96	+
Waste from U-M Campus (tons)	9,850	10,106	+
Waste from U-M Campus (tons/person)	0.16	0.16	~

Cross Cutting and Emerging Issues

	2004	2005	Changes
Building Utilization (total conditioned ft^2)	**12,521,796**	**17,224,085**	++
Building Utilization (total conditioned ft^2/person)	**172.22**	**233.24**	++
Aesthetics (# of awards received)	4	4	~
LEED Certification (# of LEED Bronze certified buildings)	0	0	
LEED Certification (# of LEED certified buildings)	0	1	+
LEED Certification (# of LEED Gold certified buildings)	0	1	+
LEED Certification (# of LEED Platinum certified buildings)	0	0	
LEED Certification (# of LEED Silver certified buildings)	0	0	
LEED Certification (% of all buildings)	0.00%	0.29%	+

Environmental-focused Degrees offered by UM as of 6/30/2006

Undergraduate Degrees	10
Master's Degrees	11
Doctoral's Level Degrees	15

N.B. Criteria is defined by Graham Environmental Sustainability Institute

Enrollment of sustainability-related courses in Fall 2004 and Winter 2005

Courses offered: 278

Enrollment: 9,563

EDR Model Improvement

Environmental Data Repository (EDR) was a Microsoft Excel-based tool developed to facilitate annual data collection, computation and tracking of the eight key environmental performance indicators and more than 50 operational metrics recommended by the *University of Michigan Environmental Task Force Advisory Report to President Mary Sue Coleman*. The model was designed so environmental data could be entered annually and stored separately in the model. This year, the model was revisited to re-examine its applicability with new data in 2005. Bugs in the code were fixed. Redundant codes were trimmed to reduce the size of the Excel spreadsheet and increase the efficiency of running the Visual Basic codes. The final version of the code can be viewed in Appendix A of this report.

General

There were 14 worksheets and 11 "user forms" in the EDR model. The "user forms" consist of Visual Basic codes (macros) to load the data from manual input or templates into the Excel worksheets. Since every user form basically performs the same function (to import data), the Visual Basic codes are theoretically similar and thus were standardized. The interface of the EDR was kept, but the following enhancements were made to reduce code size, eliminate data input errors and increase the readability of the codes:

1) The code structure was unified in each data importing process.
   - i) The code will check if the user chooses a data year from the "year drop-down menu."
   - ii) If there is more than one piece of data from a source, it will enforce the user to enter all data before proceeding.
   - iii) It prevents the code from running if the contact person info is not chosen by the user.
   - iv) The model will locate the cell(s) where the data are stored in the worksheet, e.g., PrimaryInput, SecondaryInput, RawOutput.
   - v) If data are already present partially or completely in the model, a message box will be prompted to ask the user for permission to overwrite the existing data.
   - vi) After the above data-checking processes, the input data will be transferred and stored in the designated worksheet. Contact information will be entered into the "ContactTracking" worksheet.
   - vii) The textbox contents of the user form will then be unloaded. The code structure guarantees that data will not be entered into the model when any error is found.

2) The codes were organized in a "per data source basis." Codes that deal with data from the same source are put altogether. This allows future users to modify the codes for one data source without accidentally changing the codes for another data source (e.g., the code structure of "EnterTransBut_Click()" procedure under "Transform".)

3) We acknowledged that data from different sources will not be available at the same time; codes were re-written so data from different sources but in the same user form do not need be

entered into the model simultaneously. It was not the case for every user form in the previous version.

4) For the user forms that input data from multiple sources (e.g., TransForms), data entry error from one data source will only stop the model from importing that data. The codes for other data sources will be run and those data will be imported into the model if there are no errors.

5) The process of importing the four templates was standardized. By clicking “Choose File and Import”, the templates will be imported simultaneously after the file name and path are chosen. The data-checking processes described above in 1) apply to importing templates in a similar manner. The contents of the worksheets temporarily storing data from templates will be cleared once the data are entered into the model successfully.

6) The codes of different user forms serve the same function, such as importing templates into temporary worksheets and loading data year and contact drop-down menus in every user form. Seven functions and procedures were written to reduce the code size and to reduce errors during future edits.

Written in the “LoadList” module:

i) Procedure “contact” - to load the contact list for all user forms
ii) Procedure “year” - to load the data year list for all user forms
iii) Function “import1” - to import those data with only one attribute into PrimaryInput worksheet (e.g., golf course irrigation, tree population)
iv) Function “filename” - to obtain the file name and path of the template to be imported
v) Function “importcsv” - to convert the template into .csv format and store it in temporarily worksheet
vi) Function “landuse” - to input the land use data for UM campus, Athletic Fields and Matthaei Botanical Gardens & Nichols Arboretum into PrimaryInput worksheet

Written in the “TransForm” user form:

i) Function “importraw” - to import AATA ridership and bike rack data to RawOutput worksheet

7) The cell that stores a parameter of a data year will be searched by matching the parameter name and the data year in the worksheets with those in the codes. For example:

Row_loc = WorksheetFunction.Match(parameter name, Range(“PrimaryInput!B:B”),[0])
Col_loc = WorksheetFunction.Match(year, Range(“PrimaryInput!3:3”),[0])
Sheets(“PrimaryInput”).Cells(Row_loc, Col_loc) = textbox1.value

In the previous version of the model, the cell location is explicitly specified in the codes (*e.g., Sheets(“PrimaryInput”).Cells(1,2) = textbox1.value*). The change compressed the original 100-line code into a few lines and allowed the flexibility of moving the storage locations of parameters without changing the codes. As a result, the parameter names become very important as they are the key to match input data with their storage locations in the model.

When the parameter names in the model are changed, the names of parameters in the code must be changed as well for the model to function properly.

8) Comments were added for almost all of the variables and procedures. Dates of latest edit were specified.

9) A message box will be prompted to advise the user whether data is entered successfully or not.

AddContactForm

Data validation codes were added to the "AddContactForm" user form. Phone and email formats were checked (Phone: xxx-xxx-xxxx, Email: xxxxx@xxxxx.xxx).

EmissionsForms

The code structure of "EmissionsForm" was modified to uncouple data entry from multiple sources, as described in 4) of the "General" section. As a result, stationary source emissions and salt & sand use data can be imported at different times.

Land Use Forms

AthleticLandForm, CampusLandForm and MBGNALandForm were used respectively, to import land use data for the athletic fields, UM campus and University-owned nature areas (e.g., Nichols Arboretum, Matthaei Botanical Gardens). They had the same code structure, and thus most of the codes in these user forms were put into "landuse" function in "LoadList" module and shared by the three user forms.

StartForm

The codes for each tab in StartForm have been edited for a more consistent code structure. Similar codes were put into functions in "LoadList" module. As a result, the number of lines of codes was reduced by 40%. Also, part of the codes to input cross-cutting and emerging issues data were moved to AwardsForm and EduForm

AwardsForm and EduForm

Two forms were added to accommodate the new metrics in cross-cutting and emerging issues. AwardsForm enters the number of LEED-certified buildings and aesthetic awards. EduForm handles the new academic data, namely, the number of different sustainability-related degrees and the enrollment data of sustainability-related courses at UM.

TransForm

A function "importraw" was written to locate the parameter locations and import data into RawOutput worksheet for AATA ridership and Bike rack data. In addition, there were bugs in the previous version of the model. Any new data imported from the TransportationStats template would overwrite all the old data. The model failed to copy the data located higher than row 56 in the template. The bugs were fixed when the codes were restructured. Lastly, the codes that defined the format of the "Transportation_Input" temporary spreadsheet were completely removed as they were deemed unnecessary.

UtilForm

The codes to import the *Annual Report of Utilities* were moved from procedure "UtilImportBut_Click" to "UtilFileChoose_Click" to unify the template-importing interface as stated in 5) in the "General" section. To maintain consistency in the "UtilFileChoose" module, the conversion factor of CCF to gallons was changed from 748.052 to 748.05195, which is the value used in row 6 of SecondaryInput sheet. In the previous version, the difference led to the issue that "total water use" differed from the sum of "total irrigation water" and "discharge to drainage" by more than 100 gallons.

WMS_Input

The codes were rewritten to accommodate the changes in source data structure and new template format as discussed in the previous section. The codes to import the "WMS_InputMod" template were moved from procedure "WMSImport_Click" to "WMSChoose_Click" to comply with 50 in the "General" section. New codes were written to locate different waste categories in the template and the "PrimaryInput" worksheet in the model. New codes were also written to compute and put the data in the template into different categories. New categories were added to the reuse items: 1) polystyrene and plastics, 2) computer and electronics, and 3) others. The three new parameters are stored in row 25-27 in "PrimaryInput" worksheet.

There is a known issue with the codes when handling data overwriting operations. If the new data do not fill up all the 30 parameters, the parameters that are left out will not be assigned a zero value. Instead, the values remain the same as those before overwriting.

Future Development

Stated in the report last year, there were limitations to the data or the desired data were not available. At this time, the issues remain in place as there are no major changes in the data-management practices of the contacted operations departments. At the same time, the land use data issue and the coordination problem with the Athletics Department have to be resolved in the near future for the model to track the actual land use patterns more accurately.

Another known issue with the ERD model is that it fails to capture the actual number of buildings of the UM Ann Arbor campus from the *Annual Report of Utilities*. According to the Office of Budget and Planning, UM owned 538 buildings on the Ann Arbor campus[6] compared to 346 counted by the model. It is partly due to the fact that the 223 north-campus apartment buildings were grouped into a few building complexes in the *Annual Report of Utilities*.

The EDR model can be easily expanded to incorporate new data in the future. Besides, the EDR model macros can further be enhanced to increase flexibility and running efficiency. Macro codes can be written to handle the calculations for the parameters in the RawOutput worksheet instead of assigning formulae to specific cells in the worksheet to compute them. Furthermore, the strategy in 7) in the “General” section can be adapted to the “ContactTracking” and “SecondaryInput” worksheets.

[6] http://sitemaker.umich.edu/obpinfo/other_institutional_information

Appendix A - Macro Codes

“This workbook” – load when the file is opened

```
Private Sub workbook_open()
'   Contains code for setting up the EDR
'   The Interface worksheet is activated, a greeting msgbox is displayed,
'   the main user interface is shown, and the scroll area for the Interface
'   worksheet is set
    Worksheets("Interface").Activate
    Application.ScreenUpdating = False
    MsgBox "Welcome to the University of Michigan's Environmental Data Repository" & vbCrLf & " " _
        & vbCrLf & _
        "Designed and programmed by: " & vbCrLf & "Center for Sustainable Systems" & vbCrLf & _
        "3012 Dana Bldg. 440 Church St." & vbCrLf & "The University of Michigan, Ann Arbor, MI 48109” _
               & “-1115" & vbCrLf & "Version 2.0, July 2006", vbOKOnly, "U-M Environmental Data Repository"
    Load StartForm
    StartForm.Show
    Worksheets("Interface").ScrollArea = "a1"
    With Worksheets("SecondaryInput")
        .Protect UserInterfaceOnly:=True
        .ScrollArea = "A1:P43"
    End With
    Application.ScreenUpdating = True
End Sub
```

StartForm

```
Private Sub AddContactBut_Click()
    AddContactForm.Show
End Sub

Private Sub CommandButton6_Click()
    AddContactForm.Show
End Sub

Private Sub EmissFormBut_Click()
    EmissionsForm.Show
End Sub

Private Sub TransFormBut_Click()
    TransForm.Show
End Sub

Private Sub CommandButton1_Click()
    UHS_Input.Show
End Sub

Private Sub CommandButton2_Click()
   WMS_Input.Show
End Sub

Private Sub LaunchBuildECB_Click()
```

```
   UtilForm.Show
End Sub
Private Sub SaveEDRbut_Click()
   Dim modelname As String
   modelname = ActiveWorkbook.name
   Workbooks(modelname).Save
End Sub

Private Sub CommandButton4_Click()
   Unload StartForm
   'StartForm.Hide
End Sub

Private Sub CommandButton5_Click()
   Unload StartForm
   Worksheets("Summary").Activate
End Sub

Private Sub EnterWaterBut_Click() 'Edited by Arthur Chan on 6/5/2006
   Dim WaterYear As Integer
   Dim WaterColumn As Integer
   Dim WaterCase As String     'use in overwrite warning
   Dim golf As String          'title of golf course irrigatiin in PrimaryInput sheet
   Dim IrrRow As Long          'find and store row number of golf course irrigation in PrimaryInput
   Dim a As Integer            'variable for checking data entry
   a = 0        '(0: no data entered, 1: error with data, 2: no issue with data)
   b = False    'no error is default value

   Application.ScreenUpdating = False

'   Forces user to select a year
   If StartForm.WaterYearCB.Value = "" Then
      MsgBox "Please select a year", vbOKOnly, "Year Not Specified"
      Exit Sub
   End If
   WaterYear = StartForm.WaterYearCB.Value

'   Import golf course irrigation data
   If StartForm.IrrWaterTB.Value <> "" Then
      Sheets("PrimaryInput").Activate
      golf = "Golf Course Irrigation"
      WaterColumn = WorksheetFunction.Match(WaterYear, Range("3:3"), [0])
      a = LoadList.import1(IrrWaterTB, GolfIrrContactCB, golf, 26, WaterColumn)
      If a = 2 Then StartForm.IrrWaterTB.Value = ""
   End If

'   Import CPP and Hoover water use data
   If StartForm.CPPwater.Value <> "" Or StartForm.HoovWater.Value <> "" Then
      Sheets("BldgData").Activate
      WaterColumn = WorksheetFunction.Match(WaterYear, Range("1:1"), 0)
      If StartForm.CPPwater.Value = "" Or StartForm.HoovWater.Value = "" Then
         MsgBox "Please enter water use data" & vbNewLine & "of both plants before proceeding."
         Exit Sub
      End If
      'Forces user to select a Contact
      If StartForm.WaterContactCB.Value = "" Then
```

```
            MsgBox "Please enter contact info of CPP and Hoover Plant water use"
            Exit Sub
        End If
        'Overwrite warning
        If Cells(3, WaterColumn).Value <> "" Or Cells(4, WaterColumn).Value <> "" Then
            WaterCase = MsgBox("You are about to overwrite CPP and Hoover water use data." _
                & vbNewLine & "Do you want to proceed?", vbYesNo, "Caution: Data Overwrite")
            If WaterCase = vbNo Then Exit Sub
        End If

        'Enter data into BldgData and contact into ContactTracking
        Cells(3, WaterColumn) = StartForm.CPPwater.Value
        Cells(4, WaterColumn) = StartForm.HoovWater.Value
        MsgBox "CPP and Hoover Water Use data are entered sucessfully"
        Sheets("ContactTracking").Cells(25, WorksheetFunction.Match(WaterYear, Range _
            ("ContactTracking!3:3"), [0])).Value = Right(StartForm.WaterContactCB.Value, _
            (Len(StartForm.WaterContactCB.Value) - WorksheetFunction.Find(" ", _
            WaterContactCB.Value)))
        a = 2
        StartForm.CPPwater.Value = ""
        StartForm.HoovWater.Value = ""
        StartForm.WaterContactCB.Value = ""
    End If

    If a = 0 Then MsgBox "No data is entered"
    Sheets("Interface").Activate
    Application.ScreenUpdating = True
End Sub

Private Sub SecDataEnterBut_Click()
    Dim PopYear As Integer      'data year selected by user
    Dim PopCol As Integer       'column for a particular data year
    Dim FControl As Control
    Dim PopAns As Variant       'overwrite warning answer

    Application.ScreenUpdating = False
    Sheets("SecondaryInput").Activate

'   Forces user to select a year
    If StartForm.PopYearCB.Value = "" Then
        MsgBox "Please select a year", vbOKOnly, "Year Not Specified"
        Exit Sub
    End If
    PopYear = StartForm.PopYearCB.Value

'   Forces user to enter all data
    If StartForm.StuPopTB.Value = "" Or StartForm.StaffPopTB.Value = "" Or _
        StartForm.HStaffPopTB.Value = "" Then
        MsgBox "Please enter all population data", vbOKOnly, "Enter all data"
        Exit Sub
    End If

'   Checks if data have already been written and prompts for data overwrite
    PopCol = WorksheetFunction.Match(PopYear, Range("2:2"), [0])
    If Cells(9, PopCol).Value <> "" Or Cells(10, PopCol).Value <> "" Then
        PopAns = MsgBox("You are about to overwrite population data." _
            & vbNewLine & "Do you wish to continue?", vbYesNo, "Caution: Data Overwrite")
        If PopAns = vbNo Then Exit Sub
    End If

'  Pastes data into SecondaryInput sheet
    Cells(9, PopCol) = Val(StartForm.StuPopTB.Value) + Val(StartForm.StaffPopTB.Value)
    Cells(10, PopCol) = StartForm.HStaffPopTB.Value
    MsgBox "Data are sucessfully entered"

'  Clears textboxes on form
    For Each FControl In StartForm.Controls
        On Error Resume Next
        FControl.Value = ""
    Next FControl
    Sheets("Interface").Activate
    Application.ScreenUpdating = True
End Sub

Private Sub userform_initialize()
    Dim list(1 To 5) As Object, j As Integer
    'load contact list to 5 boxes
    Set list(1) = StartForm.DeckContactCB
    Set list(2) = StartForm.TreeContactCB
    Set list(3) = StartForm.WaterContactCB
    Set list(4) = StartForm.BUtzContactBut
    Set list(5) = StartForm.GolfIrrContactCB
    For j = 1 To 5
        Call LoadList.contact(list(j))
    Next j
    'load year list to 4 boxes
    Set list(1) = StartForm.PopYearCB
    Set list(2) = StartForm.LandYearCB
    Set list(3) = StartForm.CrossCutYearCB
    Set list(4) = StartForm.WaterYearCB
    For j = 1 To 4
        Call LoadList.year(list(j))
    Next j
End Sub

'  Land Use data input
Private Sub CampusLandBut_Click()
    If StartForm.LandYearCB.Value = "" Then
        MsgBox "Please select a year", vbOKOnly, "Year Not Specified"
        Exit Sub
    End If
    CampusLandForm.Show
End Sub

Private Sub AthleticLandBut_Click()
    If StartForm.LandYearCB.Value = "" Then
        MsgBox "Please select a year", vbOKOnly, "Year Not Specified"
        Exit Sub
    End If
    AthleticLandForm.Show
End Sub
```

```
Private Sub MBGNALandBut_Click()
   If StartForm.LandYearCB.Value = "" Then
      MsgBox "Please select a year", vbOKOnly, "Year Not Specified"
      Exit Sub
   End If
   MBGNALandForm.Show
End Sub

Private Sub EnterLandBut_Click() 'Edited by Arthur on 6/5/2006
   Dim LandYear As Integer     'data year as chosen by user
   Dim LandColumn As Integer   'The column number of input year in Primary Input
   Dim LandAns As String       'overwrite warning answer
   Dim LandRow(1) As Integer   'The row having the desired parameter in PrimaryInput
   Dim a As Integer            'variable to check data entry
   Dim nam(2) As String        'name of parameters as in PrimaryInput sheet

'   Activate LandUse_Raw Worksheet
   Application.ScreenUpdating = False
   Sheets("PrimaryInput").Activate

'   Forces user to enter the year
   If StartForm.LandYearCB = "" Then
      MsgBox "Please select the year first"
      Exit Sub
   End If

'   Defines variable
   LandYear = StartForm.LandYearCB.Value
   LandColumn = WorksheetFunction.Match(LandYear, Range("PrimaryInput!3:3"), 0)
   nam(0) = "Deck Parking"
   nam(1) = "Total Parking"
   nam(2) = "Tree Population"
   a = 0  '(0: no data entered, 1: error with data, 2: no issue with data)

'   Import tree population data
   If StartForm.TextBox3.Value <> "" Then
      a = LoadList.import1(TextBox3, TreeContactCB, nam(2), 16, LandColumn)
      If a = 2 Then
         StartForm.TextBox3.Value = ""
         StartForm.TreeContactCB.Value = ""
      End If
   End If

'   Import Parking Spaces Data
   If StartForm.TextBox1.Value <> "" Or StartForm.TextBox2.Value <> "" Then
      'Make sure all data are entered
      If StartForm.TextBox1.Value = "" Or StartForm.TextBox2.Value = "" Then
         MsgBox "Please enter both parking data"
         Exit Sub
      End If
      'Make sure contact person is chosen
      If StartForm.DeckContactCB.Value = "" Then
         MsgBox "Please enter contact person for parking info"
         Exit Sub
      End If
      LandRow(0) = WorksheetFunction.Match(nam(0), Range("B:B"), [0])
      LandRow(1) = WorksheetFunction.Match(nam(1), Range("B:B"), [0])
      'Overwrite warning
      If Cells(LandRow(0), LandColumn).Value <> "" And Cells(LandRow(1), LandColumn) _
         <> "" Then
         LandAns = MsgBox("You are about to overwrite Land Use data." _
            & vbNewLine & "Do you wish to continue?", vbYesNo, _
            "Caution: Data Overwrite")
         If LandAns = vbNo Then Exit Sub
      End If
      'Enter data into PrimaryInput and contact into ContactTracking
      Cells(LandRow(0), LandColumn) = StartForm.TextBox1.Value
      Cells(LandRow(1), LandColumn) = StartForm.TextBox2.Value
      Sheets("ContactTracking").Cells(15, WorksheetFunction.Match(LandYear, _
         Range("ContactTracking!3:3"), [0])).Value = Right(StartForm _
         .DeckContactCB.Value, (Len(StartForm.DeckContactCB.Value) - _
         WorksheetFunction.Find(" ", DeckContactCB.Value)))
      MsgBox "Parking data are entered"
      a = 2
      StartForm.TextBox1.Value = ""
      StartForm.TextBox2.Value = ""
      StartForm.DeckContactCB.Value = ""
   End If

   If a = 0 Then MsgBox "No data is entered"
   Sheets("Interface").Activate
   Application.ScreenUpdating = True
End Sub

'   Start Cross-Cutting and Emerging Issues data input (Last edited on 6/21/2006 by Arthur)
Private Sub BldgUtzBut_Click()
   Dim CrossCuttingFileName As Variant     'store template path and name
   Dim modelname As String
   Dim lastrow As Variant            'last row containing data in temp worksheet
   Dim CrossYear As Integer           'data year selected by user
   Dim CrossCol As Integer            'column storing a data year in PrimaryInput
   Dim sum(1) As Double              'sum up the conditioned sq footage
   Dim nam As String               'name of parameter in PrimaryInput
   Dim r As Long                 'use in for-loop

   modelname = ActiveWorkbook.name
   Application.ScreenUpdating = False

'   Forces user to choose data year
   If StartForm.CrossCutYearCB.Value = "" Then
      MsgBox "Please select a year", vbOKOnly, "Year not specified"
      Exit Sub
   End If
   CrossYear = StartForm.CrossCutYearCB.Value

'   Forces user to select a contact
   If StartForm.BUtzContactBut.Value = "" Then
      MsgBox "Please select a contact for Building Utilization"
      Exit Sub
   End If

'   Obtain file name from user
```

```
   CrossCuttingFileName = LoadList.filename("Building Utilization")
   If CrossCuttingFileName = False Then Exit Sub

'   Open CrossCuttingFileName & export range to csv file and temp worksheet
   Workbooks.Open filename:=CrossCuttingFileName
   Sheets("Conditioned").Activate  'error occured if sheet name is changed in template
   lastrow = LoadList.importcsv(Workbooks(modelname).Path & "\CrossCutTemfile.csv", _
      Workbooks(modelname).Sheets("CrossCutting_Input"))

'   Calculate sum of conditioned space
   sum(0) = 0
   sum(1) = 0
   For r = 2 To lastrow
      If Cells(r, 1) <> "" And Cells(r, 4) <> 0 Then
         sum(0) = sum(0) + Val(Cells(r, 4) * Cells(r, 5) / 100)
         sum(1) = sum(1) + 1
      End If
   Next r

'   Input sum to PrimaryInput sheet
   CrossColumn = WorksheetFunction.Match(CrossYear, Range("PrimaryInput!3:3"), 0)
   nam = "Conditioned Square Footage"
   Call LoadList.import1(sum(0), StartForm.BUtzContactBut.Value, nam, 20, CrossColumn)

   Sheets("CrossCutting_Input").Cells.ClearContents
   Sheets("Interface").Activate
   Application.ScreenUpdating = True
End Sub

Private Sub AwardBut_click()   'Edited by Arthur Chan on 7/12/2006
   If StartForm.CrossCutYearCB = "" Then
      MsgBox "Please select a year", vbOKOnly, "Year not specified"
      Exit Sub
   End If
   AwardsForm.Show
End Sub

Private Sub EduBut_click()     'Added by Arthur on 7/12/2006
   If StartForm.CrossCutYearCB = "" Then
      MsgBox "Please select a year", vbOKOnly, "Year not specified"
      Exit Sub
   End If
   EduForm.Show
End Sub
```

AddContactForm

```
Option Explicit

Private Sub AddContactBut2_Click()
'   Enters new contact information to ContactNames list
'   Will not enter information unless all textboxes are completed
    Dim FControl As Control
    Dim Contacts As Variant
    Dim ConCount1 As Integer, ConCount2 As Integer
    Dim j As Integer
    Dim tele As String
    Dim email As String

    Application.ScreenUpdating = False
    ConCount1 = Sheets("ContactList").Range("ContactNames").Rows.count
    Contacts = Sheets("ContactList").Range("ContactNames").Value

    Do Until WMS_Input.MixContactCB.ListCount = 0
        AthleticLandForm.AthLanContactCB.RemoveItem 0
        CampusLandForm.CampLanContactCB.RemoveItem 0
        EmissionsForm.EmissContactCB.RemoveItem 0
        EmissionsForm.SalSanContactCB.RemoveItem 0
        MBGNALandForm.MBGContactCB.RemoveItem 0
        StartForm.DeckContactCB.RemoveItem 0
        StartForm.TreeContactCB.RemoveItem 0
        AwardsForm.AwardContactBut.RemoveItem 0
        EduForm.EduContactBut.RemoveItem 0
        StartForm.BUtzContactBut.RemoveItem 0
        StartForm.WaterContactCB.RemoveItem 0
        TransForm.BikeContactCB.RemoveItem 0
        TransForm.AATAContactCB.RemoveItem 0
        TransForm.TransContactCB.RemoveItem 0
        UHS_Input.UHContactCB.RemoveItem 0
        UtilForm.AURContactCB.RemoveItem 0
        UtilForm.REContactCB.RemoveItem 0
        UtilForm.CPPContactCB.RemoveItem 0
        WMS_Input.PDContactCB.RemoveItem 0
        WMS_Input.MulchContactCB.RemoveItem 0
        WMS_Input.MixContactCB.RemoveItem 0
    Loop

    If CfNameTB.Value <> "" And ClNameTB.Value <> "" And ClNameTB.Value <> "" _
        And CTeleTB.Value <> "" And CEmailTB.Value <> "" Then
            'Data validation for phone number and email address
            'Added by Arthur Chan on 6/2/2006
            tele = CTeleTB.Value
            email = CEmailTB.Value
            If (Mid(tele, 4, 1) <> "-" Or Mid(tele, 8, 1) <> "-" Or Len(tele) _
                <> 12) Then
                MsgBox "Invalid telephone number format." & vbNewLine & _
                "Should be in xxx-xxx-xxxx format."
                Exit Sub
            End If
            For j = 1 To Len(tele)
                If (Asc(Mid(tele, j, 1)) < 45 Or Asc(Mid(tele, j, 1)) > 57) Then
```

```
                    MsgBox "Invalid telephone number. Pls retry"
                    Exit Sub
                End If
            Next j
            If (InStr(1, email, "@") = 0 Or InStr(1, email, ".") = 0) Then
                MsgBox "Invalid email format. Should be in xxx@xxx.xxx format"
                Exit Sub
            End If
            'end of addition by Arthur
            With Sheets("ContactList")
                .Rows("14:14").Insert Shift:=xlDown
                .Range("A14") = ClNameTB.Value
                .Range("B14") = CfNameTB.Value
                .Range("c14") = COrgTB.Value
                .Range("d14") = CTeleTB.Value
                .Range("e14") = CEmailTB.Value
            End With
            Application.GoTo Reference:=Range("ContactNames")
            Selection.Sort Key1:=Range("A2"), Order1:=xlAscending, Header:=xlNo, _
                OrderCustom:=1, Orientation:=xlTopToBottom
            Sheets("Interface").Activate
    Else
        MsgBox "Please enter full contact information"
        Exit Sub
    End If

'   Enters updated contact names in each ComboBox
    Dim i As Integer
    Dim CName As String
    Contacts = Sheets("contactList").Range("ContactNames").Value
    For i = 1 To UBound(Contacts, 1)
        CName = Contacts(i, 2) & " " & Contacts(i, 1)
        AthleticLandForm.AthLanContactCB.AddItem CName
        CampusLandForm.CampLanContactCB.AddItem CName
        EmissionsForm.EmissContactCB.AddItem CName
        EmissionsForm.SalSanContactCB.AddItem CName
        MBGNALandForm.MBGContactCB.AddItem CName
        StartForm.DeckContactCB.AddItem CName
        StartForm.TreeContactCB.AddItem CName
        AwardsForm.AwardContactBut.AddItem CName
        EduForm.EduContactBut.AddItem CName
        StartForm.BUtzContactBut.AddItem CName
        StartForm.WaterContactCB.AddItem CName
        TransForm.BikeContactCB.AddItem CName
        TransForm.AATAContactCB.AddItem CName
        TransForm.TransContactCB.AddItem CName
        UHS_Input.UHContactCB.AddItem CName
        UtilForm.AURContactCB.AddItem CName
        UtilForm.REContactCB.AddItem CName
        UtilForm.CPPContactCB.AddItem CName
        WMS_Input.PDContactCB.AddItem CName
        WMS_Input.MulchContactCB.AddItem CName
        WMS_Input.MixContactCB.AddItem CName
    Next i

    ConCount2 = Sheets("ContactList").Range("ContactNames").Rows.count
    If ConCount1 + 1 = ConCount2 Then
        MsgBox "Contact information added successfully"
        Else
        MsgBox "Contact information was not added"
    End If

    AddContactForm.Hide
    Unload AddContactForm
    Application.ScreenUpdating = True
End Sub

Private Sub FindContactBut_Click()
'   Displays contact information
    Dim ContactYear As Integer
    Dim ContactCat As String
    Dim ContactName As String

    'Enforces the user to enter year (Added by Arthur Chan on 6/2/2006)
    If (AddContactForm.ContactYearCB = "" Or AddContactForm.ContactCatCB = "") Then
        MsgBox "please enter a year or data categorgy first"
        Exit Sub
    End If
    'end of edit by Arthur
    ContactYear = AddContactForm.ContactYearCB.Value
    ContactCat = AddContactForm.ContactCatCB.Value

    If Sheets("ContactTracking").Cells((WorksheetFunction.Match _
        (ContactCat, Sheets("ContactTracking").Range("ContactCats"), [0]) + 3), Application _
            .WorksheetFunction.Match(ContactYear, Sheets("ContactTracking") _
                .Range("3:3"), [0])).Value = "" Then
                MsgBox "There is no contact for " & ContactYear
    Else
        ConNameTB.Value = Sheets("ContactTracking").Cells((Application.WorksheetFunction _
            .Match(ContactCat, Sheets("ContactTracking").Range("ContactCats"), [0]) + 3), _
                Application.WorksheetFunction.Match(ContactYear, Sheets("ContactTracking") _
                    .Range("3:3"), [0])).Value
        ContactName = AddContactForm.ConNameTB.Value
        On Error Resume Next
        ConOrgTB.Value = Sheets("ContactList").Cells(Application.WorksheetFunction.Match _
            (ContactName, Sheets("ContactList").Range("A:A"), [0]), 3).Value
        ConPhTB.Value = Sheets("ContactList").Cells(Application.WorksheetFunction.Match _
            (ContactName, Sheets("ContactList").Range("A:A"), [0]), 4).Value
        On Error Resume Next
        ConEmailTB.Value = Sheets("ContactList").Cells(Application.WorksheetFunction.Match _
            (ContactName, Sheets("ContactList").Range("A:A"), [0]), 5).Value
        On Error Resume Next
    End If
End Sub

Private Sub userform_initialize()
    Dim list As Object, i As Integer
    Dim ContactCats As Variant
'   Adds years to dropdown menu
    Set list = AddContactForm.ContactYearCB
    Call LoadList.year(list)
'   Adds source categories to dropdown menu
```

```
    ContactCats = Sheets("ContactTracking").Range("ContactCats").Value
    For i = 1 To UBound(ContactCats, 1)
        AddContactForm.ContactCatCB.AddItem ContactCats(i, 1)
    Next i
End Sub
```

AthleticLandForm

```
Private Sub userform_initialize()
    Dim list As Object
    Set list = AthleticLandForm.AthLanContactCB
    Call LoadList.contact(list)     ' to load contact list
End Sub
Private Sub EnterAthleticBut_Click()    'Edited by Arthur on 6/22/2006
    Dim issue As Boolean
    issue = LoadList.landuse( _
        AthleticLandForm.TextBox2, AthleticLandForm.TextBox3, _
        AthleticLandForm.TextBox4, AthleticLandForm.TextBox5, _
        AthleticLandForm.AthLanContactCB, "U-M Athletic Fields", 13)
    If issue = True Then Unload AthleticLandForm
End Sub

Private Sub CancelButton_Click()
    MsgBox "Data is not entered"
    Unload AthleticLandForm
End Sub
```

CampusLandForm

```
Private Sub userform_initialize()
    Dim list As Object
    Set list = CampusLandForm.CampLanContactCB
    Call LoadList.contact(list)     'to load contact list
'   Assign zero value to unmaintained green space as assumed.
'   See documentation for details
    CampusLandForm.TextBox4.Value = 0
End Sub

Private Sub EnterCampusBut_Click()  'Edited by Arthur on 6/22/2006
    Dim issue As Boolean
    issue = LoadList.landuse( _
        CampusLandForm.TextBox2.Value, CampusLandForm.TextBox3.Value, _
        CampusLandForm.TextBox4.Value, CampusLandForm.TextBox5.Value, _
        CampusLandForm.CampLanContactCB, "U-M Campus (sq ft)", 12)
    If issue = True Then Unload CampusLandForm
End Sub

Private Sub CancelButton_Click()
    MsgBox "Data is not entered"
    Unload CampusLandForm
End Sub
```

MBGNALandForm

```
Private Sub userform_initialize()
   Dim list As Object
   Set list = MBGNALandForm.MBGContactCB
   Call LoadList.contact(list)        ' load contact list
End Sub

Private Sub EnterMBGNABut_Click() 'Edited by Arthur on 6/22/2006
   Dim issue As Boolean
   issue = LoadList.landuse( _
      MBGNALandForm.TextBox2.Value, MBGNALandForm.TextBox3.Value, _
      MBGNALandForm.TextBox4.Value, MBGNALandForm.TextBox5.Value, _
      MBGNALandForm.MBGContactCB.Value, "MBG & NA", 14)
   If issue = True Then Unload MBGNALandForm
End Sub

Private Sub CancelButton_Click()
   MsgBox "Data is not entered"
   Unload MBGNALandForm
End Sub
```

AwardsForm

```
Private Sub CommandButton7_Click()
   AddContactForm.Show
End Sub

Private Sub userform_initialize()
   Dim list As Object
   Set list = AwardsForm.AwardContactBut
   Call LoadList.contact(list)
End Sub

Private Sub EnterCrossBut_click()   'Moved from StartForm to here on 7/12/2006
   Dim CrossYear As Integer        'data year selected by user
   Dim CrossRow(1 To 6) As Variant 'locate row # of parameters in RawOutput
   Dim n(1 To 6) As Variant        'name of parameters in RawOutput
   Dim Cross(1 To 5) As Variant    'temporarily storing textbox input
   Dim CrossAns As String          'overwrite warning answer
   Dim i As Integer, j As Integer, row As Integer, lastrow As Variant

   Application.ScreenUpdating = False
   Sheets("RawOutput").Activate
   CrossYear = StartForm.CrossCutYearCB.Value

'   Forces user to enter all data
   Cross(1) = AwardsForm.TextBox4.Value     'platinum
   Cross(2) = AwardsForm.TextBox7.Value     'gold
   Cross(3) = AwardsForm.TextBox8.Value     'silver
   Cross(4) = AwardsForm.TextBox9.Value     'bronze
   Cross(5) = AwardsForm.TextBox5.Value     'aesthetic
   For i = 1 To 5
      If Cross(i) = "" Then
         MsgBox "Please enter all data before proceeding"
         Exit Sub
      End If
   Next i

'  Enforce user to enter contact information
   If AwardsForm.AwardContactBut = "" Then
      MsgBox "Please select contact info"
      Exit Sub
   End If

'  locate row number storing LEED and Aesthetic awards in RawOutput
   n(1) = "LEED Certification (# of LEED Platinum certified buildings)"
   n(2) = "LEED Certification (# of LEED Gold certified buildings)"
   n(3) = "LEED Certification (# of LEED Silver certified buildings)"
   n(4) = "LEED Certification (# of LEED Bronze certified buildings)"
   n(5) = "Aesthetics (# of awards received)"
   n(6) = "LEED Certification (# of LEED certified buildings)"
   row = 2
   j = 0
   lastrow = ActiveSheet.Cells.SpecialCells(xlLastCell).row
   Do
      For i = 1 To 6
         If (Cells(row, 2) = n(i) And Cells(row, 3) = CrossYear) Then
            CrossRow(i) = row
            j = j + 1
            Exit For
         End If
      Next i
      row = row + 1
   Loop Until (j = 6 Or row > lastrow)
   If row > lastrow Then
      MsgBox "Parameter names in EDR database and VB codes do not match." _
         & vbNewLine & "Please check the codes", vbOKOnly, "Error!"
      Exit Sub
   End If

'  Overwrite warning
   j = 0
   For i = 1 To 6
      If Cells(CrossRow(i), 4) <> "" Then j = j + 1
   Next i
   If j = 6 Then
      CrossAns = MsgBox("You are about to overwrite data." & vbNewLine & _
         "Do you want to continue?", vbYesNo, "Caution: Overwrite Warning")
      If CrossAns = vbNo Then Exit Sub
   ElseIf (j > 0 And j < 6) Then
      CrossAns = MsgBox("Current database has incomplete data." & vbNewLine & _
         "Do you want to overwrite to correct the problem?", vbYesNo, _
         "Caution: Incomplete data")
      If CrossAns = vbNo Then Exit Sub
   End If

'  Import # of LEED Certified buildings & Aesthetic awards for output
   Cells(CrossRow(6), 4) = 0
   For i = 1 To 5
      Cells(CrossRow(i), 4) = Cross(i)
      If i <> 5 Then Cells(CrossRow(6), 4) = Cells(CrossRow(6), 4) + Cross(i)
   Next i

'   Adds the last name of contact to ContactTracking worksheet
   Sheets("ContactTracking").Cells(19, WorksheetFunction.Match(CrossYear, Range _
      ("ContactTracking!3:3"), [0])).Value = Right(AwardsForm.AwardContactBut.Value, _
      (Len(AwardsForm.AwardContactBut.Value) - WorksheetFunction.Find(" ", _
      AwardContactBut.Value)))

   MsgBox "Data are entered sucessfully"
   Unload AwardsForm
   Sheets("Interface").Activate
   Application.ScreenUpdating = True
End Sub
```

EduForm

```
Private Sub CommandButton7_Click()
   AddContactForm.Show
End Sub

Private Sub userform_initialize()
   Dim list As Object
   Set list = EduForm.EduContactBut
   Call LoadList.contact(list)
End Sub

Private Sub EnterEduBut_click() 'Added by Arthur on 7/12/2006
   Dim lastrow As Integer            'last row of RawOuput sheet
   Dim nam(1 To 3) As String         'parameter names in RawOutput sheet
   Dim CrossYear As Integer          'data year selected by user
   Dim CrossRow(1 To 3) As Variant   'row # in RawOutput storing parameters
   Dim CrossAns As String            'Overwrite warning answer
   Dim i As Integer, j As Integer    'use in for loop
   Dim a As Integer

   Application.ScreenUpdating = False
   Sheets("RawOutput").Activate
   CrossYear = StartForm.CrossCutYearCB.Value
   lastrow = ActiveSheet.Cells.SpecialCells(xlLastCell).row
   a = 0

'   Degrees offered by UM
   If TextBox1.Value <> "" Or TextBox2.Value <> "" Or TextBox3.Value <> "" Then
      If TextBox1.Value = "" Or TextBox2.Value = "" Or TextBox3.Value = "" Then
         MsgBox "Please enter all degrees data"
         GoTo LineDegEnd
      End If
      ' find storage location in RawOutput
      nam(1) = "Undergraduate Degrees"
      nam(2) = "Masters Degrees"
      nam(3) = "Doctoral Degrees"
      For i = 1 To 3
         CrossRow(i) = LoadList.findraw(nam(i), CrossYear, lastrow)
         If CrossRow(i) = 0 Then GoTo LineDegEnd
```

```
        Next i
        ' Overwrite warning
        j = 0
        For i = 1 To 3
            If Cells(CrossRow(i), 4) <> "" Then j = j + 1
        Next i
        If j > 0 Then
            CrossAns = MsgBox("You are about to overwrite degrees data." & vbNewLine & _
                "Do you want to continue?", vbYesNo, "Caution: Overwrite Warning")
            If CrossAns = vbNo Then GoTo LineDegEnd
        End If
        ' import data
        Cells(CrossRow(1), 4) = TextBox1.Value
        Cells(CrossRow(2), 4) = TextBox2.Value
        Cells(CrossRow(3), 4) = TextBox3.Value
        MsgBox "Degrees info have been entered sucessfully"
        EduForm.TextBox1.Value = ""
        EduForm.TextBox2.Value = ""
        EduForm.TextBox3.Value = ""
        a = 2
    End If
LineDegEnd:

'   Envir-related class enrollment
    If TextBox4.Value <> "" Or TextBox5.Value <> "" Then
        If TextBox4.Value = "" Or TextBox5.Value = "" Then
            MsgBox "Please enter all enrollment data"
            Exit Sub
        End If
        ' Enforces user to enter contact info
        If EduForm.EduContactBut = "" Then
            MsgBox "Please select a contact", vbOKOnly, "Contact not specified"
            Exit Sub
        End If
        ' find storage location in RawOutput
        nam(1) = "Courses offered this year"
        nam(2) = "Enrollment data"
        nam(3) = "Enrollment per course"
        For i = 1 To 3
            CrossRow(i) = LoadList.findraw(nam(i), CrossYear, lastrow)
            If CrossRow(i) = 0 Then Exit Sub
        Next i
        ' Overwrite warning
        j = 0
        For i = 1 To 3
            If Cells(CrossRow(i), 4) <> "" Then j = j + 1
        Next i
        If j > 0 Then
            CrossAns = MsgBox("You are about to overwrite enrollment data." & vbNewLine & _
                "Do you want to continue?", vbYesNo, "Caution: Overwrite Warning")
            If CrossAns = vbNo Then Exit Sub
        End If
        ' import data and contact
        Cells(CrossRow(1), 4) = TextBox4.Value
        Cells(CrossRow(2), 4) = TextBox5.Value
        Cells(CrossRow(3), 4) = TextBox5.Value / TextBox4.Value
        Sheets("ContactTracking").Cells(28, WorksheetFunction.Match(CrossYear, _
            Range("ContactTracking!3:3"), [0])) = Right(EduContactBut, (Len(EduContactBut) - _
            WorksheetFunction.Find(" ", EduContactBut)))
        MsgBox "Education info have been entered sucessfully"
        EduForm.TextBox4.Value = ""
        EduForm.TextBox5.Value = ""
        EduForm.EduContactBut.Value = ""
        a = 2
    End If

    If a = 0 Then MsgBox "No data is entered"
    If a = 2 Then Unload EduForm
    Sheets("Interface").Activate
    Application.ScreenUpdating = True
End Sub
```

EmissionsForm

```
Private Sub AddContactBut_Click()
    AddContactForm.Show
End Sub

Private Sub userform_initialize()
    Dim listc As Object
    Dim listy As Object
    ' load contact list
    Set listc = EmissionsForm.EmissContactCB
    Call LoadList.contact(listc)
    Set listc = EmissionsForm.SalSanContactCB
    Call LoadList.contact(listc)
    ' load year list
    Set listy = EmissionsForm.ComboBox1
    Call LoadList.year(listy)
End Sub

'The Emissions Input Section is edited by Arthur Chan on 6/8/2006 to allow separate
'input of stationary source emissions and salt & sand use data
Private Sub EnterEmissBut_Click()
    Dim EmissYear As Integer
    Dim EmissColumn As Integer     'column # of data year in PrimaryInput
    Dim EmissAns As String         'overwrite warning answer
    Dim Em(1 To 8) As Variant      'temporarily storing the stationary emission data
    Dim loc_st As Integer          'Row # of cell storing salt use in PrimaryInput
    Dim loc_sd As Integer          'Row # of cell storing sand use in PrimaryInput
    Dim loc_e(1 To 8) As Single    'Row # of cell storing emissions in PrimaryInput
    Dim n(1 To 8) As String        'name of emissions data in PrimaryInput sheet
    Dim a As Integer, i As Integer 'for stationary source emission data checking
    Dim c As Integer, b As Boolean 'data error check
    c = 0      '(0: no data entered, 1: error with data, 2: no error with data)
    b = True

'   Forces user to select a year
    If EmissionsForm.ComboBox1.Value = "" Then
        MsgBox "Please select a year", vbOKOnly, "Year Not Specified"
        Exit Sub
    End If

'   Activate Emissions_Raw Worksheet
    Application.ScreenUpdating = False
    Sheets("PrimaryInput").Activate

'   Defines variables
    EmissYear = EmissionsForm.ComboBox1.Value
    EmissColumn = WorksheetFunction.Match(EmissYear, Range("3:3"), 0)
    Em(1) = EmissionsForm.TextBox1.Value
    Em(2) = EmissionsForm.TextBox2.Value
    Em(3) = EmissionsForm.TextBox3.Value
    Em(4) = EmissionsForm.TextBox4.Value
    Em(5) = EmissionsForm.TextBox5.Value
    Em(6) = EmissionsForm.TextBox6.Value
    Em(7) = EmissionsForm.TextBox7.Value
    Em(8) = EmissionsForm.TextBox8.Value

'   Input Salt and Sand Use
    If (EmissionsForm.TextBox10.Value <> "" Or EmissionsForm.TextBox11.Value <> "") Then
        c = 1
        'Make sure both salt and sand use are entered
        If (EmissionsForm.TextBox10.Value = "" Or EmissionsForm.TextBox11.Value = "") Then
            MsgBox "Please enter both salt and sand use data", vbOKOnly, "Incomplete Data"
            GoTo LineSaltend
        End If
        'Find row # in PrimaryInput storing salt and sand use data
        loc_st = WorksheetFunction.Match("Salt Use", Range("B:B"), 0)
        loc_sd = WorksheetFunction.Match("Sand Use", Range("B:B"), 0)
        'Make sure contact is chosen
        If SalSanContactCB.Value = "" Then
            MsgBox "Please select a contact person for salt and sand data first", vbOKOnly
            GoTo LineSaltend
        End If
        'Search if data is already present in worksheet
        If Cells(loc_st, EmissColumn).Value <> "" Or Cells(loc_sd, EmissColumn).Value <> "" Then
            EmissAns = MsgBox("You are about to overwrite salt and sand use data." _
                & vbNewLine & "Do you wish to continue?", vbYesNo, "Caution: Data Overwrite")
            If EmissAns = vbNo Then GoTo LineSaltend
        End If
        'Enter data to PrimaryInput and the contact name to ContactTracking worksheet
        Cells(loc_st, EmissColumn) = EmissionsForm.TextBox10.Value
        Cells(loc_sd, EmissColumn) = EmissionsForm.TextBox11.Value
        Sheets("ContactTracking").Cells(18, WorksheetFunction.Match(EmissYear, _
            Range("ContactTracking!3:3"), [0])).Value = Right(EmissionsForm.SalSanContactCB.Value, _
            (Len(EmissionsForm.SalSanContactCB.Value) - WorksheetFunction.Find(" ", _
            EmissionsForm.SalSanContactCB.Value)))
        MsgBox "Salt and Sand Use Data are entered successfully"
        EmissionsForm.TextBox10.Value = ""
        EmissionsForm.TextBox11.Value = ""
        c = 2
LineSaltend:
        If c = 1 Then b = False
    End If

'   Input Stationary Source of Emissions Data
```

```
a = 0
For i = 1 To 8      'Count the number of data entered in textboxes
   If Em(i) <> "" Then a = a + 1
Next i
If a > 0 Then
   c = 1
   'Make sure all emissions data are entered at the same time
   If a <> 8 Then
      MsgBox "Please enter all stationary emissions data", vbOKOnly, "Incomplete Data"
      Exit Sub
   End If
   'Make sure contact is chosen
   If EmissContactCB.Value = "" Then
      MsgBox "Please select a contact person for emissions data"
      Exit Sub
   End If
   'Find its row # location in PrimaryInput worksheet
   n(1) = "Stationary Source Pounds of CO2 Equivalence"
   n(2) = "Stationary Source Carbon Monoxide"
   n(3) = "Stationary Source Nitrogen Oxides"
   n(4) = "Stationary Source Volatile Organic Compounds"
   n(5) = "Stationary Source Particulate Matter - 10"
   n(6) = "Stationary Source Particulate Matter - 2.5"
   n(7) = "Stationary Source Sulfur Dioxide"
   n(8) = "Stationary Source Lead"
   For i = 1 To 8
      loc_e(i) = WorksheetFunction.Match(n(i), Range("B:B"), 0)
   Next i
   'Warning of overwriting data
   a = 0
   For i = 1 To 8
      If Cells(loc_e(i), EmissColumn).Value <> "" Then a = a + 1
   Next i
   If a = 8 Then
      EmissAns = MsgBox("You are about to overwrite emissions data." _
         & vbNewLine & "Do you wish to continue?", vbYesNo, "Caution: Data Overwrite")
      If EmissAns = vbNo Then Exit Sub
   ElseIf (a > 0 And a < 8) Then
      EmissAns = MsgBox("Existing database has incomplete Emission data." & vbNewLine _
         & "Do you wish to input these data to rectify the problem?", vbYesNo, _
         "Problem: Incomlpete data in database")
      If EmissAns = vbNo Then Exit Sub
   End If
   'Enter data to PrimaryInput and contact to ContactTracking
   For i = 1 To 8
      Cells(loc_e(i), EmissColumn) = Em(i)
   Next i
   Sheets("ContactTracking").Cells(17, WorksheetFunction.Match(EmissYear, _
      Range("ContactTracking!3:3"), [0])).Value = Right(EmissionsForm.EmissContactCB.Value, _
      (Len(EmissionsForm.EmissContactCB.Value) - WorksheetFunction.Find(" ", _
      EmissionsForm.EmissContactCB.Value)))
   MsgBox "Emissions Data is entered"
   c = 2
End If

If c = 0 Then MsgBox "No data is entered"
```

```
    If c = 2 And b = True Then Unload EmissionsForm
    Sheets("Interface").Activate
    Application.ScreenUpdating = True
End Sub

Private Sub CancelButton_Click()
    MsgBox "No data is entered"
    Unload EmissionsForm
End Sub
```

RecRecycleForm

```
Option Explicit

Private Sub ContactBut_Click()
    Load AddContactForm
    AddContactForm.Show
End Sub

Private Sub EntDataBut_Click() 'Edited by Arthur on 6/7/2006
    Dim RegYear As Integer     'data year chosen by user
    Dim RegTB(9) As Variant    'temporarily storing the input of 10 parameters
    Dim n(9) As String         'name of these 10 parameters in PrimaryInput sheet
    Dim loc(9) As Integer      'locate row # of parameters in PrimaryInput sheet
    Dim Yr_Column As Integer   'locate column # of data year in PrimaryInput sheet
    Dim a As Integer, b As Integer 'variable used in for-loop and counter
    Dim RegAns As Variant      'overwrite warning answer

'   Forces user to select a year
    If RegRecycleForm.RegYearCB.Value = "" Then
        MsgBox "Please select a year", vbOKOnly, "Year Not Specified"
        Exit Sub
    End If

'   Define variables
    Application.ScreenUpdating = False
    Sheets("PrimaryInput").Activate
    RegYear = RegRecycleForm.RegYearCB.Value
    RegTB(0) = BattTB.Value
    RegTB(1) = LBTB.Value
    RegTB(2) = CETB.Value
    RegTB(3) = FLBTB.Value
    RegTB(4) = AcetoneTB.Value
    RegTB(5) = XyleneTB.Value
    RegTB(6) = FormalinTB.Value
    RegTB(7) = TransOilTB.Value
    RegTB(8) = LatexTB.Value
    RegTB(9) = CoolTB.Value

'   Assure that all data are entered
    For a = 0 To 9
        If RegTB(a) = "" Then
            MsgBox "Please enter all data before proceeding"
            Exit Sub
        End If
    Next a

'   Forces the user to enter contact info
    If RegRecycleForm.RegContactCB.Value = "" Then
        MsgBox "Please select a contact", vbOKOnly, "Contact Info Not Specified"
        Exit Sub
    End If

'   Find row location of parameters in PrimaryInput sheet
    n(0) = "Batteries (lbs)"
    n(1) = "Lamp Ballasts (lbs)"
    n(2) = "Consumer Electronics (lbs)"
    n(3) = "Fluorescent light bulbs (lbs)"
    n(4) = "Acetone (gallons)"
    n(5) = "Xylene (gallons)"
    n(6) = "Formlain (gallons)"
    n(7) = "Transportation Oil (gallons)"
    n(8) = "Latex Paint (gallons)"
    n(9) = "Coolants (gallons)"
    For a = 0 To 9
        loc(a) = WorksheetFunction.Match(n(a), Range("B:B"), [0])
    Next a
'   Locate column number of data year in PrimaryInput sheet
    Yr_Column = WorksheetFunction.Match(RegYear, Range("3:3"), 0)

'   Overwrite warning
    b = 0
    For a = 0 To 9
        If Cells(loc(a), Yr_Column).Value <> "" Then
            b = b + 1
        End If
    Next a
    If b = 10 Then
        RegAns = MsgBox("You are about to overwrite Regulated Recycling Data." _
            & vbNewLine & "Do you wish to continue?", vbYesNo, _
            "Caution: Data Overwrite")
        If RegAns = vbNo Then Exit Sub
    ElseIf (b > 0 And b < 10) Then
        RegAns = MsgBox("There are incomplete data present in database." _
            & vbNewLine & "Do you want to input these data to rectify the problem?" _
            , vbYesNo, "Problem: Incomplete data in database")
        If RegAns = vbNo Then Exit Sub
    End If

'   Enter data
    For a = 0 To 9
        Cells(loc(a), Yr_Column) = RegTB(a)
    Next a

'   Writes contact to "ContactTracking" worksheet
    Sheets("ContactTracking").Cells(27, WorksheetFunction.Match(RegYear, _
        Range("ContactTracking!3:3"), [0])).Value = Right(RegRecycleForm _
        .RegContactCB.Value, (Len(RegRecycleForm.RegContactCB.Value) - _
        WorksheetFunction.Find(" ", RegRecycleForm.RegContactCB.Value)))

    MsgBox "Data are entered sucessfully"
    Unload RegRecycleForm
    Sheets("Interface").Activate
    Application.ScreenUpdating = True
End Sub

Private Sub userform_initialize()
    Dim list As Object
    ' load contact list
    Set list = RegRecycleForm.RegContactCB
    Call LoadList.contact(list)
    ' load data year
    Set list = RegRecycleForm.RegYearCB
    Call LoadList.year(list)
End Sub
```

TransForm

```
Private Sub CommandButton2_Click()
    AddContactForm.Show
End Sub
Private Sub userform_initialize()
    Dim listc As Object
    Dim listy As Object
    ' load contact list to 3 boxes
    Set listc = TransForm.BikeContactCB
    Call LoadList.contact(listc)
    Set listc = TransForm.AATAContactCB
    Call LoadList.contact(listc)
    Set listc = TransForm.TransContactCB
    Call LoadList.contact(listc)
    ' load year list
    Set listy = TransForm.ComboBox1
    Call LoadList.year(listy)
End Sub

Private Sub CommandButton1_Click()  'Last edited by Arthur on 6/20/2006
    Dim TransportFileName As Variant
    Dim NumRows As Long              'number of active rows of template
    Dim a As Integer, b As Integer   'use in for-loop and counter
    Dim modelname As String
    Dim TransYear As Integer         'data year chosen by user
    Dim TransColumn As Integer       'column location of data year in PrimaryInput
    Dim TransAns As String           'overwrite warning answer
    Dim TransRow(1 To 18) As Integer 'row locations of parameters in PrimaryInput

    modelname = ActiveWorkbook.name
    Application.ScreenUpdating = False

'   Forces user to select a year
    If TransForm.ComboBox1.Value = "" Then
        MsgBox "Please select a year first", vbOKOnly, "Year Not Specified"
        Exit Sub
    End If
    TransYear = TransForm.ComboBox1.Value
```

```
'  Forces user to choose contact
   If TransForm.TransContactCB.Value = "" Then
      MsgBox "Please choose a contact first", vbOKOnly
      Exit Sub
   End If

'  Obtain file name from user
   TransportFileName = LoadList.filename("TransportationStats")
   If TransportFileName = False Then Exit Sub

'  Open TransportFileName & export range to csv file
   Workbooks.Open filename:=TransportFileName
   Sheets("Trans_input").Activate
   NumRows = LoadList.importcsv(Workbooks(modelname).Path & "\Transtempfile.csv", _
      Workbooks(modelname).Sheets("Transportation_Input"))

'  This code warns against inputting Transportation data to wrong year
   Sheets("Transportation_Input").Activate
   Select Case Cells(1, 3).Value
   Case ""
      MsgBox "The data year is not specified by your template." & vbNewLine & _
      "Please go back and enter year in cell C1 in template", vbOKOnly, "Year Not Specified"
      Exit Sub
   Case Is <> TransYear
      TransAns = MsgBox("You are entering Yr " & TransYear & " data with a Yr " _
         & Cells(1, 3) & " datasheet." & vbNewLine & "Are you sure it is correct?", vbYesNo, _
         "Caution: Data of Different Year")
      If TransAns = vbNo Then Exit Sub
   End Select

'  locate the rows storing the parameters in PrimaryInput
   TransColumn = WorksheetFunction.Match(TransYear, Range("PrimaryInput!3:3"), 0)
   For a = 1 To 15
      TransRow(a) = WorksheetFunction.Match(Cells(a + 1, 1) & " " & Cells(a + 1, 2), Range( _
         "PrimaryInput!B:B"), [0])
      If a = 12 Then a = 13   'to skip row 14 of template
   Next a
   TransRow(16) = WorksheetFunction.Match("Vanpool Total Passengers", Range("PrimaryInput!B:B"),
[0])
   TransRow(17) = WorksheetFunction.Match("Vanpool Vehicle-Miles Traveled",
Range("PrimaryInput!B:B"), [0])
   TransRow(18) = WorksheetFunction.Match("Vanpool Passenger-Miles Traveled",
Range("PrimaryInput!B:B"), [0])

'  This code warns against overwriting Transportation data
   b = 0
   For a = 1 To 18
      If Sheets("PrimaryInput").Cells(TransRow(a), TransColumn).Value <> "" Then b = b + 1
      If a = 12 Then a = 13
   Next a
   If b = 17 Then
      TransAns = MsgBox("You are about to overwrite Transportation data." _
         & vbNewLine & "Do you wish to continue?", vbYesNo, "Caution: Data Overwrite")
      If TransAns = vbNo Then Exit Sub
   ElseIf (b > 0 And b < 17) Then
      TransAns = MsgBox("There are incomplete data present in database." _
            & vbNewLine & "Do you want to input these data to rectify the problem?" _
            , vbYesNo, "Problem: Incomplete data in database")
        If TransAns = vbNo Then Exit Sub
    End If

'   Calculate the total vanpool passengers, vehicle and passenger miles of travel
    Dim Pas As Long
    Dim VMT As Single
    Dim PMT As Single

    Pas = 0
    VMT = 0
    PMT = 0
    For a = 18 To NumRows
        Pas = Pas + Cells(a, 3).Value
        VMT = VMT + Cells(a, 4).Value
        PMT = PMT + Cells(a, 3).Value * Cells(a, 4).Value
    Next a

    Sheets("PrimaryInput").Activate
    'Copy Energy Consumption and bus ridership to PrimaryInput
    For a = 1 To 15
        Cells(TransRow(a), TransColumn) = Sheets("Transportation_Input").Cells(a + 1, 3).Value
        If a = 12 Then a = 13
    Next a
    'Copy Vanpooling data to Primary Input
    Cells(TransRow(16), TransColumn) = Pas
    Cells(TransRow(17), TransColumn) = VMT
    Cells(TransRow(18), TransColumn) = PMT

'   Add contact name to ContactTracking Sheet
    Sheets("ContactTracking").Cells(23, WorksheetFunction.Match(TransYear, _
        Range("ContactTracking!3:3"), [0])).Value = Right(TransForm.TransContactCB.Value, _
        (Len(TransForm.TransContactCB.Value) - WorksheetFunction _
        .Find(" ", TransForm.TransContactCB.Value)))

    MsgBox "Transportation Stats template is imported sucessfully"
    TransForm.Hide
    Sheets("Interface").Activate
    Application.ScreenUpdating = True
End Sub

Private Sub EnterTransBut_Click()  'Last edited on 6/23/2006 by Arthur Chan
    Dim TransYear As Integer  'Store data year entered
    Dim AATA As String        'Name of AATA bus ridership indicator stored in RawOutput
    Dim Bike As String        'Name of Bicycle ridership indicatorstored in RawOutput
    Dim MySum As Double       'sum up the bike ridership data
    Dim a As Integer, b As Boolean  'data entry check
    a = 0  ' (0: no data, 1: error with data, 2: no error with data)
    b = False

'   Activate RawOutput Worksheet
    Application.ScreenUpdating = False
    Sheets("RawOutput").Activate

'   Forces user to select a year
```

```
    If TransForm.ComboBox1.Value = "" Then
        MsgBox "Please select a year", vbOKOnly, "Year Not Specified"
        Exit Sub
    End If
    TransYear = TransForm.ComboBox1.Value

'   Import AATA Bus Ridership Data for output
    If TransForm.TextBox5.Value <> "" Then
        AATA = "AATA Bus Ridership"
        a = importraw(TransForm.TextBox5, AATA, TransYear, TransForm.AATAContactCB.Value, 22)
        If a = 2 Then        'if no error found then...
            TransForm.TextBox5.Value = ""
            TransForm.AATAContactCB.Value = ""
        Else
            b = True
        End If
    End If

'   Import Bike Ridership Data for output
    If (TransForm.TextBox1.Value <> "" Or TransForm.TextBox2.Value <> "" Or _
        TransForm.TextBox3.Value <> "" Or TransForm.TextBox4.Value <> "") Then
        'Make sure data is entered for each campus
        If (TransForm.TextBox1.Value = "" Or TransForm.TextBox2.Value = "" Or _
            TransForm.TextBox3.Value = "" Or TransForm.TextBox4.Value = "") Then
            MsgBox "Please enter bike ridership data for each campus"
        Else
            Bike = "Bicycle Ridership (Number of bike racks)"
            MySum = Val(TransForm.TextBox1.Value) + Val(TransForm.TextBox2.Value) _
                  + Val(TransForm.TextBox3.Value) + Val(TransForm.TextBox4.Value)
            a = importraw(MySum, Bike, TransYear, TransForm.BikeContactCB.Value, 21)
            If a = 2 Then
                TransForm.TextBox1.Value = ""
                TransForm.TextBox2.Value = ""
                TransForm.TextBox3.Value = ""
                TransForm.TextBox4.Value = ""
                TransForm.BikeContactCB.Value = ""
            Else
                b = True
            End If
        End If
    End If

    If a = 0 Then MsgBox "No data is entered"
    If a = 2 And b = False Then Unload TransForm
    Sheets("Interface").Activate
    Application.ScreenUpdating = True
End Sub

Private Sub CancelButton_Click()
    MsgBox "Data is not entered"
    Unload TransForm
End Sub

Function importraw(i As Double, nam, yr, contact, crow As Integer)
    Dim loc As Variant          'row # storing parameter in RawOutput
    Dim lastrow As Variant      'last row containing data in RawOutput
```

```
   Dim OverAns As String       'overwrite warning answer
   importraw = 1               'default: error occured

'   Enforces the user to enter contact info
   If contact = "" Then
      MsgBox "Please enter contact info for " & nam, vbOKOnly, "Contact not specified"
      Exit Function
   End If

'   Locate the row number storing the parameter in RawOutput
   lastrow = ActiveSheet.Range("B1").End(xlDown).row
   loc = LoadList.findraw(nam, yr, lastrow)
   If loc = 0 Then Exit Function

'   Overwrite warning
   If Cells(loc, 4).Value <> "" Then
      OverAns = MsgBox("You are about to overwrite " & nam & " data." _
      & vbNewLine & "Do you wish to continue?", vbYesNo, "Caution: Data Overwrite")
      If OverAns = vbNo Then Exit Function
   End If

'   Enter data into RawOutput and contact into ContactTracking
   Cells(loc, 4) = i
   Sheets("ContactTracking").Cells(crow, WorksheetFunction.Match(yr, _
      Range("ContactTracking!3:3"), [0])) = Right(contact, (Len(contact) _
      - WorksheetFunction.Find(" ", contact)))
   MsgBox nam & " Data is entered"
   importraw = 2    'indicated no error found and data are entered
End Function
```

UHS_Input

```
Option Explicit

Private Sub AddContactBut_Click()
   AddContactForm.Show
End Sub

Private Sub EnterButton_Click()  'Edited by Arthur on 6/8/2006
'   Pastes UHS data into worksheet
   Dim UHSyear As Integer  'data year chosen by user
   Dim UHSTB(6) As Variant 'temporarily storing input from textbox
   Dim n(6) As String      'names of parameters in PrimaryInput
   Dim loc(6) As Integer   'row location of parameters in PrimaryInput
   Dim NextColumn As Integer 'column number of data year in PrimaryInput
   Dim i As Integer, b As Integer 'used in for loop and counters
   Dim UHSans As Variant   'Overwrite warning answer

   Application.ScreenUpdating = False
   Worksheets("PrimaryInput").Activate

'   Forces user to select a year
   If UHS_Input.UHS_Year.Value = "" Then
      MsgBox "Please select a year", vbOKOnly, "Year Not Specified"
      Exit Sub
   End If
   UHSyear = UHS_Input.UHS_Year.Value

'  Stores textbox values in the UHSTB array
   UHSTB(0) = UHS_Input.UHSrefuseTB.Value
   UHSTB(1) = UHSpaperTB.Value
   UHSTB(2) = UHScardTB.Value
   UHSTB(3) = UHSwoodTB.Value
   UHSTB(4) = UHSmetalTB.Value
   UHSTB(5) = UHSplasticTB.Value
   UHSTB(6) = UHSgreaseTB.Value

'  Make sure all data are entered and contact person is chosen
   For i = 0 To 6
      If UHSTB(i) = "" Then
         MsgBox "Please enter all data", vbOKOnly, "Incomplete data"
         Exit Sub
      End If
   Next i
   If UHS_Input.UHContactCB.Value = "" Then
      MsgBox "Please select a contact", vbOKOnly, "Contact Not Specified"
      Exit Sub
   End If

'  Locate the row number of storing parameters in PrimaryInput sheet
   n(0) = "UHS Total refuse"
   n(1) = "UHS Paper (hospital)"
   n(2) = "UHS Cardboard (hospital)"
   n(3) = "UHS Recycled wood"
   n(4) = "UHS Scrap metal"
   n(5) = "UHS Plastic recycled"
   n(6) = "UHS Yellow kitchen grease"
   For i = 0 To 6
      loc(i) = WorksheetFunction.Match(n(i), Range("B:B"), [0])
   Next i
'  Locate input column number to year of data
   NextColumn = WorksheetFunction.Match(UHSyear, Range("3:3"), 0)

'  Overwrite warning
   b = 0
   For i = 0 To 6
      If Cells(loc(i), NextColumn).Value <> "" Then
         b = b + 1
      End If
   Next i
   If b = 7 Then
      UHSans = MsgBox("You are about to overwrite UHS data." & vbNewLine & _
         "Do you want to continue?", vbYesNo, "Caution: Data Overwrite")
      If UHSans = vbNo Then Exit Sub
   ElseIf (b > 0 And b < 7) Then
      UHSans = MsgBox("Existing database has incomplete UHS data." & vbNewLine _
         & "Do you wish to input these data to rectify the problem?", vbYesNo, _
         "Problem: Incomlpete data in database")
      If UHSans = vbNo Then Exit Sub
   End If
```

```
'   Write data into PrimaryInput and contact info into ContactTracking
    For i = 0 To 6
        Cells(loc(i), NextColumn) = UHSTB(i)
    Next i
    Sheets("ContactTracking").Cells(8, WorksheetFunction.Match(UHSyear, _
        Range("ContactTracking!3:3"), [0])).Value = Right(UHS_Input.UHContactCB _
        .Value, (Len(UHS_Input.UHContactCB.Value) - WorksheetFunction.Find(" ", _
        UHS_Input.UHContactCB.Value)))

    MsgBox "Data are entered sucessfully"
    Unload UHS_Input
    Worksheets("Interface").Activate
    Application.ScreenUpdating = True
End Sub

Private Sub UserForm_Initialize()
    Dim listc As Object
    Dim listy As Object
    ' load contact list
    Set listc = UHS_Input.UHContactCB
    Call LoadList.contact(listc)
    ' load year list
    Set listy = UHS_Input.UHS_Year
    Call LoadList.year(listy)
End Sub
```

UtilForm

```
Option Explicit

Private Sub AddContactBut_Click()
    AddContactForm.Show
End Sub

Private Sub UtilImportBut_Click()   'Last edited by Arthur Chan on 6/19/2006
    Dim UtYear As Integer     'data year of entry
    Dim Yr_col(3) As Variant  'Store the column # of particular year in different sheets
    Dim UtAns As String       'overwrite warning ans
    Dim I As Integer          'used in for loop
    Dim a As Integer, b As Boolean 'data check
    a = 0
    b = False

    Application.ScreenUpdating = False

    'Make sure year is chosen
    If UtilForm.UtYearCB.Value = "" Then
        MsgBox "Please choose data year first"
        Exit Sub
    End If
    'Define variable
    UtYear = UtilForm.UtYearCB.Value
    Yr_col(1) = WorksheetFunction.Match(UtYear, Range("ContactTracking!3:3"), 0)

'   Enter data for Generated Electricity from Renewable Sources (converts from kWh to MWh)
```

```
    If UtilForm.RenewETB.Value <> "" Then
        Dim name As String  'title of this parameter in PrimaryInput sheet
        Dim loc As Integer  'row location of this parameter in PrimaryInput sheet
        name = "Electricity Generated from renewable sources (MWh)"
        loc = WorksheetFunction.Match(name, Range("PrimaryInput!B:B"), [0])
        Yr_col(0) = WorksheetFunction.Match(UtYear, Range("PrimaryInput!3:3"), [0])
        a = LoadList.import1(RenewETB.Value / 1000, REContactCB.Value, name, 10, Yr_col(0))
        If a = 2 Then UtilForm.RenewETB.Value = ""
        If a = 1 Then b = True
    End If

'   Enter CPP and Hoover data
    If (UtilForm.CPPFuelOil.Value <> "" Or UtilForm.CPPGas.Value <> "" Or _
        UtilForm.HoovGas.Value <> "") Then
        a = 1
        Yr_col(0) = WorksheetFunction.Match(UtYear, Range("BldgData!1:1"), 0)
        'Make sure all data is entered
        If (UtilForm.CPPFuelOil.Value = "" Or UtilForm.CPPGas.Value = "" Or _
        UtilForm.HoovGas.Value = "") Then
            MsgBox "All CPP and Hoover data must be entered. Please try again."
            GoTo LineCPPend
        End If
        'Make sure contact person is chosen
        If UtilForm.CPPContactCB.Value = "" Then
            MsgBox "Please choose a contact for CPP & Hoover data"
            GoTo LineCPPend
        End If
        'Overerite warning
        If Sheets("BldgData").Cells(3, Yr_col(0) + 5) <> "" Or _
            Sheets("BldgData").Cells(3, Yr_col(0) + 7) <> "" Or _
            Sheets("BldgData").Cells(4, Yr_col(0) + 5) <> "" Then
            UtAns = MsgBox("You are about to overwrite CPP data." _
                & vbNewLine & "Do you wish to continue?", vbYesNo, "Caution: Data Overwrite")
            If UtAns = vbNo Then GoTo LineCPPend
        End If
        Sheets("BldgData").Cells(3, Yr_col(0) + 5) = UtilForm.CPPGas.Value
        Sheets("BldgData").Cells(3, Yr_col(0) + 7) = UtilForm.CPPFuelOil.Value
        Sheets("BldgData").Cells(4, Yr_col(0) + 5) = UtilForm.HoovGas.Value
        'Enter contact person to ContactTracking sheet
        Sheets("ContactTracking").Cells(9, Yr_col(1)).Value = Right(UtilForm.CPPContactCB _
            .Value, (Len(UtilForm.CPPContactCB.Value) - WorksheetFunction.Find(" ", _
            UtilForm.CPPContactCB.Value)))
        MsgBox "CPP and Hoover data has been inserted"
        a = 2
        UtilForm.CPPContactCB.Value = ""
        UtilForm.CPPGas.Value = ""
        UtilForm.CPPFuelOil.Value = ""
        UtilForm.HoovGas.Value = ""
LineCPPend:
        If a = 1 Then b = True
    End If

'   Enter data from External Data tab
    If (CoalPerTB.Value <> "" Or CoalHRTB.Value <> "" Or NGPerTB.Value <> "" _
        Or NGHRTB.Value <> "" Or FOPerTB.Value <> "" Or FOHRTB.Value <> "" Or _
        HyPerTB.Value <> "" Or ORPerTB.Value <> "" Or _
        NucPerTB.Value <> "" Or NucHRTB.Value <> "") Then
        Dim HeatRate As Integer
        Sheets("SecondaryInput").Activate
        Yr_col(0) = WorksheetFunction.Match(UtYear, Range("SecondaryInput!2:2"), [0])
        ' Telling the user to enter all data
        If (CoalPerTB.Value = "" Or CoalHRTB.Value = "" Or NGPerTB.Value = "" _
        Or NGHRTB.Value = "" Or FOPerTB.Value = "" Or FOHRTB.Value = "" Or _
        HyPerTB.Value = "" Or HyHRTB.Value = "" Or ORPerTB.Value = "" Or _
        ORHRTB.Value = "" Or NucPerTB.Value = "" Or NucHRTB.Value = "") Then
            MsgBox "Please enter all external data before clicking import"
            Exit Sub
        End If
        ' Overwrite warning
        For I = 3 To 41
            If Cells(I, Yr_col(0)) <> "" Then
                UtAns = MsgBox("You are about to overwrite External data." _
                & vbNewLine & "Do you wish to continue?", vbYesNo, "Caution: Data Overwrite")
                Select Case UtAns
                    Case vbNo
                        Exit Sub
                    Case vbYes
                        Exit For
                End Select
            End If
            If I = 3 Then I = 33
        Next I
        ' Calculates and enters total heat rate for purchased electricity
        HeatRate = ((CoalPerTB.Value / 100) * CoalHRTB.Value) + ((NGPerTB.Value / 100) _
            * NGHRTB.Value) + ((NucPerTB.Value / 100) * NucHRTB.Value) + ((FOPerTB.Value / 100) * _
            FOHRTB.Value) + ((HyPerTB.Value / 100) * HyHRTB.Value) + ((ORPerTB.Value / 100) * _
            ORHRTB.Value)
        Cells(3, Yr_col(0)) = HeatRate
        ' Enters fuel mix and heat rate for coal, natural gas, and fuel oil. Also fuel mix %'s for
        ' hydro and other renewables.
        '  These values are used to calculate greenhouse gas emissions from purchased electricity
        Cells(34, Yr_col(0)).Value = CoalPerTB.Value / 100
        Cells(35, Yr_col(0)).Value = NGPerTB.Value / 100
        Cells(36, Yr_col(0)).Value = FOPerTB.Value / 100
        Cells(37, Yr_col(0)).Value = HyPerTB.Value / 100
        Cells(38, Yr_col(0)).Value = ORPerTB.Value / 100
        Cells(39, Yr_col(0)).Value = CoalHRTB.Value
        Cells(40, Yr_col(0)).Value = NGHRTB.Value
        Cells(41, Yr_col(0)).Value = FOHRTB.Value
        MsgBox "External Data entered successfully!", vbOKOnly
        a = 2
    End If
    If a = 0 Then MsgBox "No data is entered!"
    If (a = 2 And b = False) Then Unload UtilForm
    Worksheets("Interface").Activate
    Application.ScreenUpdating = True
End Sub

Private Sub userform_initialize()
    Dim listc As Object
    Dim listy As Object
    ' load contact list to 3 boxes
    Set listc = UtilForm.AURContactCB
    Call LoadList.contact(listc)
    Set listc = UtilForm.REContactCB
    Call LoadList.contact(listc)
    Set listc = UtilForm.CPPContactCB
    Call LoadList.contact(listc)
    ' load year list
    Set listy = UtilForm.UtYearCB
    Call LoadList.year(listy)

'   Displays the heat rates for hydropower and other renewables.
'   These are constant values and are not written to the EDR.
    UtilForm.HyHRTB.Value = 3412
    UtilForm.ORHRTB.Value = 3412
End Sub

Private Sub UtilFileChoose_Click()
'   Selects name of Annual Utilities Report to import building energy data
'   Moved to here by Arthur Chan on 6/2/2006
    Dim UtilFileName As Variant, csvFileName As String
    Dim UtYear As Integer   'data year of entry
    Dim UtYr_col As Long    'first column of desired yr in BldgData
    Dim roww As Variant     'row location of parameters
    Dim UtilAns As String   'overwrite warning decision (Yes, No)
    Dim BldgNum As Integer
    Dim Bldgs As Variant
    Dim EnergyCat As Variant 'Energy Category (Ekectricity, Water, etc)
    Dim BldgArea As Variant
    Dim c As Variant
    Dim h As Integer, i As Integer, j As Integer, l As Integer
    Dim firstAddress As Variant
    Dim AreaSum As Double
    AreaSum = 0
    Dim IrrWater As Double
    Dim SewerWater As Double
    Dim modelname As String
    modelname = ActiveWorkbook.name
    Application.ScreenUpdating = False

'   Forces user to select a year
    If UtilForm.UtYearCB.Value = "" Then
        MsgBox "Please select a year", vbOKOnly, "Year Not Specified"
        Exit Sub
    End If
    UtYear = UtilForm.UtYearCB.Value

'   Enforce user to enter contact person
    If UtilForm.AURContactCB.Value = "" Then
        MsgBox "Please choose a contact first", vbOKOnly, "Year not specified"
        Exit Sub
    End If

'   Obtain file name from user
    UtilFileName = LoadList.filename("Annual Report of Utilities")
    If UtilFileName = False Then Exit Sub
```

```
'   Overwrite warning
    UtYr_col = WorksheetFunction.Match(UtYear, Range("BldgData!1:1"), 0)
    If Application.WorksheetFunction.Max(Sheets("BldgData").Cells(5, UtYr_col), _
    Cells(370, UtYr_col)) > 0 Then
        UtilAns = MsgBox("You are about to overwrite Utilities Report data for " _
        & UtYear & vbNewLine & "Do you wish to continue?", vbYesNo, _
            "Caution: Data Overwrite")
        If UtilAns = vbNo Then
            UtilForm.UtilFileNameTB.Value = ""
            Exit Sub
        End If
    End If

'   Imports .xls sheet if input module was selected and converts to .csv file
    Workbooks.Open filename:=UtilFileName
    Sheets(1).Activate
    Cells.MergeCells = False
    Call LoadList.importcsv(Workbooks(modelname).Path & "\Annual Report by Building.csv", _
        Workbooks(modelname).Sheets("UtilCSV"))

'   Defines list of buildings as an array
    Bldgs = Worksheets("BldgList").Range("BldgNums").Value
'   Pastes Number of buildings
    Sheets("SecondaryInput").Cells(43, WorksheetFunction.Match(UtYear, Sheets("SecondaryInput") _
    .Range("2:2"), 0)).Value = WorksheetFunction.CountA(Range("BldgNames"))

'   Activates sheet with pasted .CSV data
'   Loops through building numbers in Bldg array, finding matching building
'   numbers in Utilities Report.
'   Copies and pastes data based on heading ("Electricity", "Electricity-CPP", etc)
    Worksheets("UtilCSV").Activate
    With Worksheets("UtilCSV").Range("A:A")
        For i = 1 To UBound(Bldgs, 1)
            Set c = .Find(Bldgs(i, 1), LookAt:=xlWhole, LookIn:=xlValues)
            If Not c Is Nothing Then
                firstAddress = c.Address
                Do
'               Finds and sums the building areas
                    For h = 1 To 7
                        BldgArea = Worksheets("UtilCSV").Range(firstAddress) _
                        .Offset(h, 1).Value
                        Select Case BldgArea
                            Case "Total"
                                AreaSum = AreaSum + Range(firstAddress).Offset(h, 5).Value
                            End Select
                    Next h
                    Set c = .FindNext(c)
                Loop While Not c Is Nothing And c.Address <> firstAddress
                Do
'               Finds and pastes the utilties data
                    For j = 1 To 5
                        EnergyCat = Worksheets("UtilCSV").Range(firstAddress). _
                        Offset(j, 1).Value
                        Select Case EnergyCat
                            Case "Ann Arbor Water and Sewer"
                                Range(firstAddress).Offset(j, 5).Copy _
```

```
                     Worksheets("BldgData").Cells(i + 4, UtYr_col)
                  Case "Water & Sewer"
                     Range(firstAddress).Offset(j, 5).Copy _
                     Worksheets("BldgData").Cells(i + 4, UtYr_col + 1)
                  Case "Electricity-CPP"
                     Range(firstAddress).Offset(j, 5).Copy _
                     Worksheets("BldgData").Cells(i + 4, UtYr_col + 2)
                  Case "Electricity-NC"
                     Range(firstAddress).Offset(j, 5).Copy _
                     Worksheets("BldgData").Cells(i + 4, UtYr_col + 3)
                  Case "Electricity"
                     Range(firstAddress).Offset(j, 5).Copy _
                     Worksheets("BldgData").Cells(i + 4, UtYr_col + 4)
                  Case "Recharge Bulk Gas"
                     Range(firstAddress).Offset(j, 5).Copy _
                     Worksheets("BldgData").Cells(i + 4, UtYr_col + 5)
                  Case "Natural Gas"
                     Range(firstAddress).Offset(j, 5).Copy _
                     Worksheets("BldgData").Cells(i + 4, UtYr_col + 6)
                  Case "Fuel Oil"
                     Range(firstAddress).Offset(j, 5).Copy _
                     Worksheets("BldgData").Cells(i + 4, UtYr_col + 7)
               End Select
            Next j
            Set c = .FindNext(c)
         Loop While Not c Is Nothing And c.Address <> firstAddress
      End If
   Next i
End With
' Pastes Total Building area to sheet
Sheets("SecondaryInput").Cells(8, Application.WorksheetFunction _
   .Match(UtYear, Sheets("SecondaryInput").Range("2:2"), [0])).Value = AreaSum

' Calculates the volume of water discharged to sewers and used for irrigation
IrrWater = 0
SewerWater = 0
Sheets("BldgData").Activate
For l = 5 To UBound(Bldgs, 1)
   IrrWater = IrrWater + Cells(l, WorksheetFunction.Match(UtYear, Range("1:1"), 0)) * _
      Sheets("BldgList").Cells(l - 3, 6) + Cells(l, WorksheetFunction.Match(UtYear, _
      Range("1:1"), 0) + 1) * Sheets("BldgList").Cells(l - 3, 6)
   SewerWater = SewerWater + Cells(l, WorksheetFunction.Match(UtYear, Range("1:1"), 0)) _
      * Sheets("BldgList").Cells(l - 3, 5) + Cells(l, WorksheetFunction.Match(UtYear, _
      Range("1:1"), 0) + 1) * Sheets("BldgList").Cells(l - 3, 5)
Next l
Sheets("PrimaryInput").Activate
roww = WorksheetFunction.Match("Building Irrigation Water", Range("B:B"), [0])
Cells(roww, (WorksheetFunction.Match(UtYear, Range("3:3"), 0))) = IrrWater * 748.05195 'fixed#
roww = WorksheetFunction.Match("Total Discharged to Sewers", Range("B:B"), [0])
Cells(roww, (WorksheetFunction.Match(UtYear, Range("3:3"), 0))) = SewerWater * 748.05195 'fixed#

If Sheets("BldgData").Cells(18, UtYr_col).Value = "" Then
   MsgBox "The Annual Utilities Report is not imported. Wrong file (format)", vbOKOnly
Else
   ' Input contact person to ContactTracking sheet
   Dim colc As Variant     'column of data year in ContactTracking sheet
```

```
      colc = WorksheetFunction.Match(UtYear, Range("ContactTracking!3:3"), [0])
      Sheets("ContactTracking").Cells(11, colc).Value = Right(UtilForm.AURContactCB _
         .Value, (Len(UtilForm.AURContactCB.Value) - WorksheetFunction.Find(" ", _
         UtilForm.AURContactCB.Value)))
      Sheets("ContactTracking").Cells(24, colc).Value = Sheets("ContactTracking"). _
         Cells(11, colc).Value
      MsgBox "Annual Report of Utilities is imported successfully"
   End If
   Worksheets("UtilCSV").Cells.Delete Shift:=xlUp
   Unload UtilForm
   Sheets("Interface").Activate
   Application.ScreenUpdating = True
End Sub
```

WMS_Input

```
Option Explicit

Private Sub RegRecBut_Click()
   RegRecycleForm.Show
End Sub

Private Sub AddContactBut_Click()
   AddContactForm.Show
End Sub

Private Sub GlassDefCB_Click()
'  Toggles default and user-specified value for % glass
'  (estimates given by Bill Leonidas from FCR Recycling on 6/15/2006)
   Select Case GlassDefCB.Value
      Case False: WMS_Input.GlassPer.Value = ""
      Case True: WMS_Input.GlassPer.Value = 60.1
      End Select
End Sub

Private Sub MetalDefCB_Click()
'  Toggles default and user-specified value for % metal
'  (estimates given by Bill Leonidas from FCR Recycling on 6/15/2006)
   Select Case MetalDefCB.Value
      Case False: WMS_Input.MetalPer.Value = ""
      Case True: WMS_Input.MetalPer.Value = 11.3
      End Select
End Sub

Private Sub PlasticDefCB_Click()
'  Toggles default and user-specified value for % plastic
'  (estimates given by Bill Leonidas from FCR Recycling on 6/15/2006)
   Select Case PlasticDefCB.Value
      Case False: WMS_Input.PlasticPer.Value = ""
      Case True: WMS_Input.PlasticPer.Value = 19.1
      End Select
End Sub

Private Sub WMSChoose_Click()   'Edited by Arthur on 6/16/2006
'  Import WMS template into model
```

```
   Dim WMSFileName As Variant        'store the file path and name
   Dim modelname As String
   Dim WMSYear As Integer          'data year selected by user
   Dim WMSans As String           'overwrite warning
   Dim i(1 To 2) As Integer      'data column in template
   Dim n(4) As Variant          'name of parameters in PrimaryInput or template
   Dim cat_s(1 To 4) As Integer, cat_e(1 To 4) As Integer 'start and end of category in template
   Dim NextColumn As Integer       'column # of particular data year in PrimaryInput
   Dim cat_range As Range          'used in output recycling, const and compost
   Dim sum(3) As Double           'use for summation purpose
   Dim D As Range               'use in .find command
   Dim x As Long, y As Integer    'used in for-loop

   Application.ScreenUpdating = False
   modelname = ActiveWorkbook.name

'  Forces user to select a year first
   If WMS_Input.WMSYear.Value = "" Then
      MsgBox "Please select a year before clicking the button", vbOKOnly, _
         "Error: Year not selected"
      Exit Sub
   End If
   WMSYear = WMS_Input.WMSYear.Value

'  Check if data has already been imported (Overwrite warning)
   NextColumn = WorksheetFunction.Match(WMSYear, Range("PrimaryInput!3:3"), [0])
   n(1) = "Compost: Betsey Barbour House"
   n(2) = "WMS Paper recycling"
   n(3) = "Re-use: others"
   For y = 1 To 3
      If Sheets("PrimaryInput").Cells(WorksheetFunction.Match(n(y), Range _
         ("PrimaryInput!B:B"), [0]), NextColumn) <> "" Then
         WMSans = MsgBox("You are about to overwrite WMS data" & vbNewLine _
            & "Do you want to continue?", vbYesNo, "Caution: Data Overwrite")
         If WMSans = vbNo Then Exit Sub
         If WMSans = vbYes Then Exit For
      End If
   Next y

'  Obtain file name from user
   WMSFileName = LoadList.filename("WMS template")
   If WMSFileName = False Then Exit Sub

'  Alerts user if the incorrect input module was selected
   Workbooks.Open filename:=WMSFileName
   Sheets(1).Activate
   If Range("A1").Value <> "Name:" Then
      MsgBox "The incorrect file was selected. Please make the correct selection."
      WMS_Input.WMSFileName.Value = ""
      Exit Sub
   ElseIf Range("B1").Value = "" Then
      MsgBox "Contact information was not entered in WMS_InputMod." & vbCrLf & _
         "Please open the file and enter the missing information."
      WMS_Input.WMSFileName.Value = ""
      Exit Sub
   End If
```

```
'   Imports .xls sheet and converts to .csv file
    Call LoadList.importcsv(Workbooks(modelname).Path & "\WMS_InputMod.csv", _
        Workbooks(modelname).Sheets("HidInput"))

'   Locate the different categories stored in WMS template (HidInput)
    n(1) = "compost"
    n(2) = "construction"
    n(3) = "recycling"
    n(4) = "reuse"
    For y = 1 To 4
        cat_s(y) = WorksheetFunction.Match(n(y), Range("A:A"), [0])
        cat_e(y) = cat_s(y) + WorksheetFunction.CountIf(Range("A:A"), n(y)) - 1
        If Cells(cat_e(y), 1).Value <> n(y) Then
            MsgBox "Non-continuous <" & n(y) & "> category label in column A" _
            & vbNewLine & "of WMS template. Please check template"
            MsgBox "No data is entered"
            Exit Sub
        End If
    Next y

'   Output compost and construction data (summed up before output)
    n(1) = "Compost: "
    n(2) = "WMS Construction In-House Waste "
    i(1) = 5    'column in template storing compost weight data
    i(2) = 3    'column in template storing construction waste weight data
    Set cat_range = Sheets("PrimaryInput").Range("B:B")
    With cat_range
        For x = 1 To 2
            For y = (cat_s(x) + 1) To cat_e(x)
                Set D = .Find(n(x) & Cells(y, 2), LookIn:=xlValues)
                If Not D Is Nothing Then
                    Sheets("PrimaryInput").Cells(D.row, NextColumn) = Cells(y, i(x))
                ElseIf Not IsEmpty(Cells(y, i(x))) Then
                    MsgBox "Model does not recognize <" & Cells(y, 2) & _
                    "> category in " & n(x) & vbNewLine & "Please check the template."
                End If
            Next y
        Next x
    End With

'   Output recycling data
    For y = 0 To 3
        sum(y) = 0
    Next y
    For y = (cat_s(3) + 1) To cat_e(x)
        Select Case Cells(y, 2)
        Case "Mixed Containers"
            sum(0) = sum(0) + Cells(y, 4)
        Case "Paper"
            sum(1) = sum(1) + Cells(y, 4)
        Case "Scrap Metal", "Scrap Wood", "Trash", "Other"
            If Cells(y, 3) = "" Then Cells(y, 3) = "unlabeled"
            With cat_range
                Set D = .Find("WMS " & Cells(y, 2) & " (" & Cells(y, 3) & ")", _
                LookIn:=xlValues)
            If Not D Is Nothing Then
               Sheets("PrimaryInput").Cells(D.row, NextColumn) = Cells(y, 4)
            End If
         End With
      Case ""
      Case Else
         MsgBox "Model does not recognize <" & Cells(y, 2) & "> category in " & _
            "Recycling." & vbNewLine & "Please check the template"
      End Select
   Next y
   Sheets("PrimaryInput").Cells(WorksheetFunction.Match("WMS Mixed containers", _
      cat_range, [0]), NextColumn) = sum(0)
   Sheets("PrimaryInput").Cells(WorksheetFunction.Match("WMS Paper recycling", _
      cat_range, [0]), NextColumn) = sum(1)

'  Output reuse data
   Dim check As Boolean
   check = True
   n(0) = "Re-use: polystyrene and plastics"
   n(1) = "Re-use: comp & electronics"
   n(2) = "Re-use: others"
   For y = 0 To 2
      sum(y) = 0
   Next y
   For y = (cat_s(4) + 1) To cat_e(4)
      Select Case Cells(y, 6)        ' for different categories
      Case "Polystyrene Block"                'assume 4 lb/bag
         If Cells(y, 5) = "lbs" Then sum(0) = sum(0) + Cells(y, 4)
         If Cells(y, 5) = "bags" Then sum(0) = sum(0) + (Cells(y, 4) * 4)
      Case "Polystyrene Peanut", "Bubble Wrap"  'assume 2 lb/bag
         If Cells(y, 5) = "lbs" Then sum(0) = sum(0) + Cells(y, 4)
         If Cells(y, 5) = "bags" Then sum(0) = sum(0) + (Cells(y, 4) * 2)
      Case "Transparencies"
         If Cells(y, 5) = "lbs" Then sum(0) = sum(0) + Cells(y, 4)
         If Cells(y, 5) <> "lbs" Then check = False
      Case "Electronic Media", "Electronics", "Computer Accessories"
         If Cells(y, 5) = "lbs" Then sum(1) = sum(1) + Cells(y, 4)
         If Cells(y, 5) <> "lbs" Then check = False
      Case "Laser Toner Cartridges", "Toner Cartridges" 'assume 3lb@
         If Cells(y, 5) = "quantity" Then sum(1) = sum(1) + (Cells(y, 4) * 3)
         If Cells(y, 5) <> "quantity" Then check = False
      Case Else
         If Cells(y, 5) = "lbs" Then sum(2) = sum(2) + Cells(y, 4)
         If Cells(y, 5) <> "lbs" Then check = False
      End Select
   Next y
   If check = False Then MsgBox "Model cannot read some of the units of reuse data." & _
      vbNewLine & "Please check the template", vbOKOnly, "Error with reuse data units"
   For y = 0 To 2
      Sheets("PrimaryInput").Cells(WorksheetFunction.Match(n(y), Range( _
         "PrimaryInput!B:B"), [0]), NextColumn) = sum(y)
   Next y

'  Enter contact information into ContactTracking
   Sheets("ContactTracking").Cells(5, WorksheetFunction.Match(WMSYear, Range( _
      "ContactTracking!3:3"), [0])) = Right(Sheets("HidInput").Range("B1").Value, _
```

```
        (Len(Worksheets("HidInput").Range("B1").Value) - WorksheetFunction.Find _
        (" ", Worksheets("HidInput").Range("B1").Value)))

    MsgBox "Data are entered into PrimaryInput sheet."
    Unload WMS_Input
    Worksheets("HidInput").Cells.Delete Shift:=xlUp
    Sheets("Interface").Activate
    Application.ScreenUpdating = True
End Sub

Private Sub WMSImport_Click()
'   Imports manually UM and external data fields. Last edited by Arthur on 6/16/2006
    Dim WMSYear As Integer    'data year selected by user
    Dim NextColumn As Integer 'column storing data of particular year in worksheets
    Dim n(1) As Variant       'name of 2 parameters as in PrimaryInput
    Dim WMSans As String      'overwrite warning answer for External data
    Dim a As Integer, b As Boolean  'data entry check

    a = 0  '(0: no data entered, 1: error with data, 2: no issue with data)
    b = False 'no error is default value
    Application.ScreenUpdating = False
    Sheets("PrimaryInput").Activate

'   Forces user to select a year
    If WMS_Input.WMSYear.Value = "" Then
        MsgBox "Please select a year", vbOKOnly, "Year Not Specified"
        Exit Sub
    End If
    WMSYear = WMS_Input.WMSYear.Value
    NextColumn = WorksheetFunction.Match(WMSYear, Range("3:3"), 0)

'   Import property disposition data
    n(0) = "Property Disposition Sales"
    If WMS_Input.PDTB.Value <> "" Then
        a = LoadList.import1(PDTB.Value, PDContactCB, n(0), 6, NextColumn)
        If a = 2 Then WMS_Input.PDTB.Value = ""
        If a = 1 Then b = True   'error found
    End If

'   Import mulch data
    n(1) = "Mulch"
    If WMS_Input.MulchVol.Value <> "" Then
        a = LoadList.import1(MulchVol, MulchContactCB, n(1), 7, NextColumn)
        If a = 2 Then WMS_Input.MulchVol.Value = ""
        If a = 1 Then b = True   'error found
    End If

'   Import External data (will run even 'input error' was found in previous sections)
    If WMS_Input.GlassPer.Value <> "" Or WMS_Input.PlasticPer.Value <> "" Or _
        WMS_Input.MetalPer.Value <> "" Then
        Sheets("SecondaryInput").Activate
        'Make sure all data is entered
        If WMS_Input.GlassPer.Value = "" Or WMS_Input.PlasticPer.Value = "" Or _
            WMS_Input.MetalPer.Value = "" Then
            MsgBox "Please enter all external data"
            Exit Sub
        End If
        'Enforce user to select contact
        If WMS_Input.MixContactCB.Value = "" Then
            MsgBox "Please select contact info for external data"
            Exit Sub
        End If
        'overwrite warning
        NextColumn = WorksheetFunction.Match(WMSYear, Range("2:2"), [0])
        If Cells(12, NextColumn) <> "" And Cells(13, NextColumn) <> "" And _
            Cells(14, NextColumn) <> "" Then
            WMSans = MsgBox("You are about to overwrite external data." & vbNewLine & _
                "Do you wish to continue?", vbYesNo, "Caution: Data Overwrite")
            If WMSans = vbNo Then Exit Sub
        End If
        'enter data into SecondaryInput and contact into ContactTracking
        Cells(12, NextColumn) = WMS_Input.GlassPer.Value / 100
        Cells(13, NextColumn) = WMS_Input.PlasticPer.Value / 100
        Cells(14, NextColumn) = WMS_Input.MetalPer.Value / 100
        Sheets("ContactTracking").Cells(4, WorksheetFunction.Match(WMSYear, _
            Range("ContactTracking!3:3"), [0])).Value = Right(WMS_Input.MixContactCB _
            .Value, (Len(WMS_Input.MixContactCB.Value) - WorksheetFunction.Find(" ", _
            WMS_Input.MixContactCB.Value)))
        MsgBox "External data is entered sucessfully"
        GlassPer.Value = ""
        MetalPer.Value = ""
        PlasticPer.Value = ""
        a = 2
    End If

    If a = 0 Then MsgBox "No data is entered"  'indicate no data is entered at all
    If (a = 2 And b = False) Then Unload WMS_Input 'if data is entered w/ no error, unload form
    Sheets("Interface").Activate
    Application.ScreenUpdating = True
End Sub

Private Sub userform_initialize()
    Dim list(1 To 4) As Object, i As Integer
    Set list(1) = WMS_Input.PDContactCB
    Set list(2) = WMS_Input.MulchContactCB
    Set list(3) = WMS_Input.MixContactCB
    Set list(4) = WMS_Input.WMSYear

'   enter contact names in each combobox
    For i = 1 To 3
        Call LoadList.contact(list(i))
    Next i
'   enter year list to dropdown menu
    Call LoadList.year(list(4))
End Sub
```

Module “Loadlist”

```
' A sub-procedure to load the contact list for all user forms
Sub contact(des As Object)  'Last edited by Arthur on 6/9/2006
   Dim Contacts As Variant
   Dim i As Integer
   Dim CName As String

   Contacts = Sheets("ContactList").Range("ContactNames").Value
   For i = 1 To UBound(Contacts, 1)
      CName = Contacts(i, 2) & " " & Contacts(i, 1)
      des.AddItem CName
   Next i
End Sub

' A sub-procedure to load year list for all user forms
Sub year(des As Object)  'Last edited by Arthur on 6/9/2006
   Dim yr_start As Integer, yr_end As Integer, i As Integer
   yr_start = 2000
   yr_end = 2010       'edited this parameter after year 2010
   For i = yr_start To yr_end
      des.AddItem i
   Next i
End Sub

' A function to import parameters into PrimaryInput sheets from userforms
Function import1(i As Double, contact As Variant, nam, crow As Integer, yrcol)
' i: inputted parameter value, contact: contact info, nam: name of parameter in PrimaryInput,
' crow: row number in ContactTracking, yrcol: column storing data of particular yr in PrimaryInput
   Dim prow As Variant     'row # storing parameter in designated worksheet
   Dim ans As String       'overwrite warning answer
   Sheets("PrimaryInput").Activate
   import1 = 1
   'Make sure contact name is chosen
   If contact = "" Then
      MsgBox "Please choose contact for " & nam, vbOKOnly, "Contact info needed"
      Exit Function
   End If
   'locate the row storing parameters in PrimaryInput
   prow = WorksheetFunction.Match(nam, Range("B:B"), [0])
   'overwrite warning
   If Cells(prow, yrcol) <> "" Then
      ans = MsgBox("You are about to overwrite " & nam & " data." & vbNewLine _
         & "Do you wish to continue?", vbYesNo, "Caution: Data overwrite")
      If ans = vbNo Then Exit Function
   End If
   'input data into PrimaryInput and contact into ContactTracking
   Cells(prow, yrcol) = i
   Sheets("ContactTracking").Cells(crow, WorksheetFunction.Match(Cells(3, yrcol) _
      , Range("ContactTracking!3:3"), [0])) = Right(contact, (Len(contact) - _
      WorksheetFunction.Find(" ", contact)))
   MsgBox nam & " data is entered sucessfully"
   import1 = 2
End Function
```

```
' A function to locate and select a template to be imported
Function filename(nam As String)   ' Added by Arthur on 6/20/2006
   Dim Filt As String
   Dim FilterIndex As Integer
   Dim Title As String

'   Sets up list of file filters
   Filt = "CSV Files (*.csv),*.csv," & _
      "Excel Files (*.xls),*.xls," & "All Files (*.*),*.*"
'   Displays *.csv by default
   FilterIndex = 2
'   Sets the dialog box caption
   Title = "Select " & nam & " file to import"
'   Gets the file name
   filename = Application.GetOpenFilename _
      (FileFilter:=Filt, FilterIndex:=FilterIndex, Title:=Title)
'   Exits if dialog box canceled
   If filename = False Then
      MsgBox "No file was selected"
      Exit Function
   End If
'   Displays full path and name of the Util file in the textbox
   MsgBox "You selected " & filename, vbOKOnly, "File selected"
End Function

'Imports .xls sheet and converts to .csv file and temp worksheet
Function importcsv(csvname As String, msheet As Worksheet) 'Added 6/21/2006
' csvname: name of .csv file, msheet: temp worksheet in model storing template data
   Dim lastrow As Double, lastcol As Integer
   Dim modelname As String
   Dim data
   modelname = ActiveWorkbook.name

'   Determine last row and column containing data
   lastrow = ActiveSheet.Cells.SpecialCells(xlLastCell).row
   importcsv = lastrow
   lastcol = ActiveSheet.Cells.SpecialCells(xlLastCell).Column

'   converts to .csv file
   Open csvname For Output As #1
      For r = 1 To lastrow
         For c = 1 To lastcol
            data = Cells(r, c).Value
            If IsNumeric(data) Then data = Val(data)
            If IsEmpty(Cells(r, c)) Then data = ""
            If c <> lastcol Then
               Write #1, data;
            Else
               Write #1, data
            End If
         Next c
      Next r
   Close #1
   ActiveWorkbook.Close savechanges:=False

'   Activate temp worksheet
```

```
    msheet.Activate
    Cells.ClearContents
'   Place exported range in temp worksheet for calculations
     With ActiveSheet.QueryTables.Add(Connection:= _
        "TEXT;" & csvname, Destination:=Range("A1"))
        .FieldNames = True
        .RowNumbers = False
        .FillAdjacentFormulas = False
        .PreserveFormatting = True
        .RefreshOnFileOpen = False
        .RefreshStyle = xlOverwriteCells
        .SavePassword = False
        .SaveData = True
        .AdjustColumnWidth = False
        .RefreshPeriod = 0
        .TextFilePromptOnRefresh = False
        .TextFilePlatform = xlWindows
        .TextFileStartRow = 1
        .TextFileParseType = xlDelimited
        .TextFileTextQualifier = xlTextQualifierDoubleQuote
        .TextFileConsecutiveDelimiter = False
        .TextFileTabDelimiter = False
        .TextFileSemicolonDelimiter = False
        .TextFileCommaDelimiter = True
        .TextFileSpaceDelimiter = False
        .TextFileColumnDataTypes = Array(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
    On Error Resume Next
        .Refresh BackgroundQuery:=False
    End With
End Function

' A function to input land use data for UM campus, athletic field and UM-owned natural areas
Function landuse(i1 As Variant, i2 As Variant, i3 As Variant, i4 As Variant, contact As String, _
site As String, prow As Integer) 'Added 6/22/2006
' i1-i4: input value, contact: input contact, site: parameter, prow: row in ContactTracking
    Dim year As Integer         'data year selected by user
    Dim cols As Integer         'column storing data of particular year in PrimaryInput
    Dim loc(1 To 5) As Integer  'row storing parameters in PrimaryInput
    Dim n(1 To 5) As String     'titles of parameters in PrimaryInput
    Dim i(1 To 4) As Variant    'temp storage of input values
    Dim LandAns As String       'overwrite warning answer
    Dim a As Integer, b As Double 'used in for-loop and summation

    Application.ScreenUpdating = False
    Sheets("PrimaryInput").Activate
    landuse = False
    year = StartForm.LandYearCB.Value
    cols = WorksheetFunction.Match(year, Range("3:3"), 0)
    i(1) = i1
    i(2) = i2
    i(3) = i3
    i(4) = i4
    n(1) = "Impervious Surface Area " & site
    n(2) = "Total Green Space " & site
    n(3) = "Maintained Green Space " & site
    n(4) = "Unmaintained (Natural) Green Spc " & site
    n(5) = "Total Land Area " & site

'   Forces user to enter all data
    For a = 1 To 4
        If i(a) = "" Then
            MsgBox "Please enter all data", vbOKOnly, "Incomplete data"
            Exit Function
        End If
    Next a
'   Data Validation
    b = 0
    b = Val(i(2)) + Val(i(3))
    If Val(i(1)) <> b Then
        MsgBox "Total green space not equals sum of maintained and non-maintained" & _
        " green spaces." & vbNewLine & "Please enter the data again"
        Exit Function
    End If
    If Val(i(4)) < Val(i(1)) Then
        MsgBox "Total land area is smaller than total green area." & vbNewLine & _
            "Please enter the data again", vbOKOnly, "Data error"
        Exit Function
    End If

'   Forces the user to select contact info
    If contact = "" Then
        MsgBox "Please select contact info", vbOKOnly, "Contact not specified"
        Exit Function
    End If

'   Find the row location storing these parameters in PrimaryInput
    For a = 1 To 5
        loc(a) = WorksheetFunction.Match(n(a), Range("B:B"), [0])
    Next a

'   This code warns against overwriting Land Use data
    b = 0
    For a = 1 To 5
        If Cells(loc(a), cols).Value <> "" Then b = b + 1
    Next a
    If b = 5 Then
        LandAns = MsgBox("You are about to overwrite " & site & " Land Use data." _
            & vbNewLine & "Do you wish to continue?", vbYesNo, _
            "Caution: Data Overwrite")
        If LandAns = vbNo Then Exit Function
    ElseIf (b > 0 And b < 5) Then
        LandAns = MsgBox("Existing database has incomplete land use data for Yr " _
            & year & vbNewLine & "Do you wish to overwrite and rectify the problem?", _
            vbYesNo, "Problem: Incomlpete data in database")
        If LandAns = vbNo Then Exit Function
    End If

'   Import data intp PrimaryInput and Enter contact into ContactTracking sheet
    Cells(loc(1), cols) = i(4) - i(1)
    For a = 2 To 5
        Cells(loc(a), cols) = i(a - 1)
    Next a
    Sheets("ContactTracking").Cells(prow, WorksheetFunction.Match(year, _
        Range("ContactTracking!3:3"), [0])).Value = Right(contact, (Len(contact) - _
        WorksheetFunction.Find(" ", contact)))

    MsgBox site & " Land Use data are entered sucessfully"
    Sheets("Interface").Activate
    Application.ScreenUpdating = True
    landuse = True
End Function

Function findraw(n, yr, lastrow) 'Added by Arthur on 7/12/2006
'To find the row number storing a parameter in RawOutput sheet
    Dim count As Integer
    count = 2
    findraw = 0
    Do
        If (Cells(count, 2) = n And Cells(count, 3) = yr) Then findraw = count
        count = count + 1
    Loop Until (findraw > 0 Or count > lastrow)
'   Tells user that the row location cannot be found (name or year not matched)
    If count > lastrow Then
        MsgBox "Error to VB Code or RawOutput datasheet." & vbNewLine & n & _
        " Indicator name or year of codes and datasheet do not match"
    End If
End Function
```

Appendix B – WMS Instruction Sheet

Instruction Sheets for Retrieving Data from Waste Management Services Database

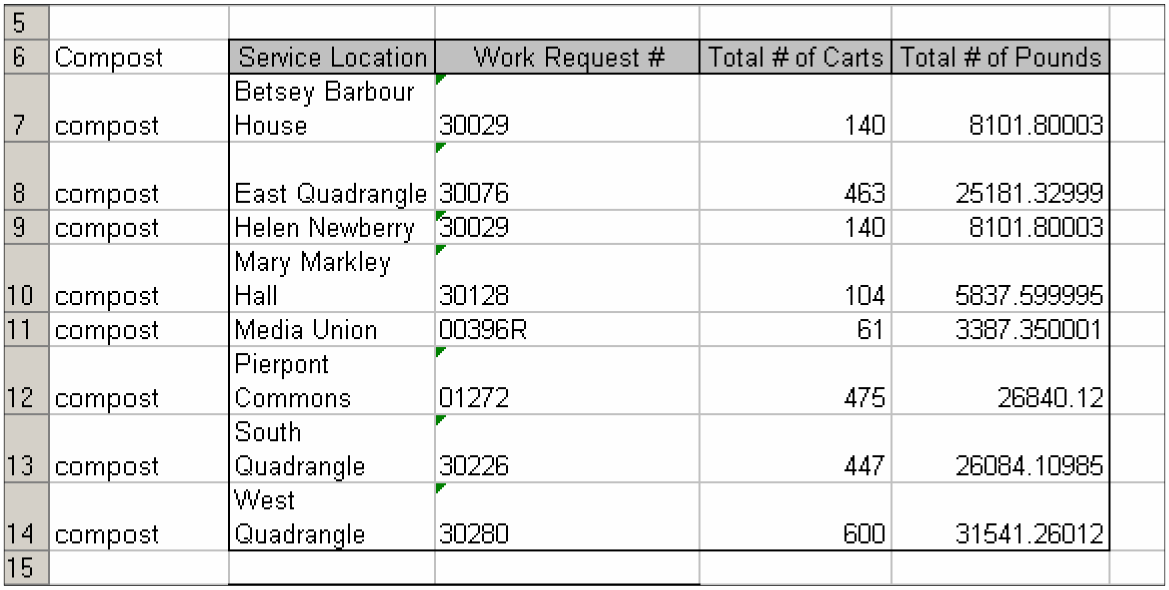
- Updated on 6/20/2006 by Arthur Chan

A template (usually an Excel spreadsheet named something like WMS_InputMod.xls) is prepared for our colleagues at the Waste Management Services to fill out the annual data on campus-wide (except hospital) refuse and recycling data. It simply requires running four existing database queries and pasting the results into the template. There are specific locations within the template where query data should be pasted. These locations are indicated by the labels shown in column A and include “compost”, “construction”, “recycling”, and “reuse”. Please note that while you can only add or delete rows for each data section, any additions or deletions of columns will render the file incompatible with the Environmental Data Repository model.

Here are the five simple steps:

1) Enter your name and contact information in the first four columns of the worksheet.

2) Open the WMS database. Run “Compost Data Query (by date)”. Enter the correct date range, copy the entire output table and paste the results next to the cells labeled “compost” in the template. Delete the extra row with item names if present. The template should now look similar to Figure 1.



5						
6	Compost	Service Location	Work Request #	Total # of Carts	Total # of Pounds	
7	compost	Betsey Barbour House	30029	140	8101.80003	
8	compost	East Quadrangle	30076	463	25181.32999	
9	compost	Helen Newberry	30029	140	8101.80003	
10	compost	Mary Markley Hall	30128	104	5837.599995	
11	compost	Media Union	00396R	61	3387.350001	
12	compost	Pierpont Commons	01272	475	26840.12	
13	compost	South Quadrangle	30226	447	26084.10985	
14	compost	West Quadrangle	30280	600	31541.26012	
15						

Figure 1: Compost data in template

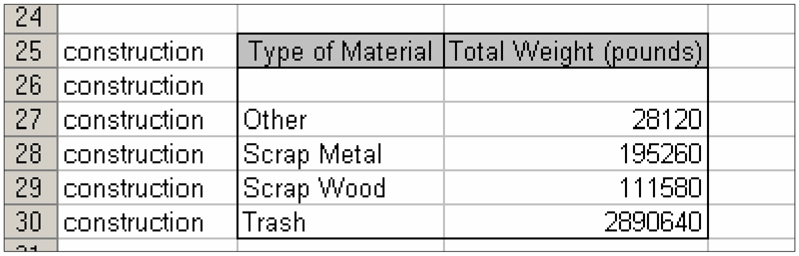
N.B.:

a) If there are more items than allowed in the template, please insert a few rows to accommodate the extra items. Inserting rows will not affect the ability of our model to read the template.

b) Make sure that when a row contains compost data, “compost” has been inserted in column A of that row. Failing to do so will cause errors to the model. In this case, cells A6 to A14 should be filled with “compost”.

c) Columns can not be added or deleted. If the pasted data do not match with the template columns, please contact us immediately. “Service Location” and “Total # of pounds” should be respectively located in column 2 and column 5.

3) Run the “Roll-Off Weight Ticket Query”. Enter the correct date range, copy the entire query output table and paste the results directly to the right of the cells labeled “construction” in the template. Delete the extra row with item names if present. The template should now look similar to Figure 2.



24			
25	construction	Type of Material	Total Weight (pounds)
26	construction		
27	construction	Other	28120
28	construction	Scrap Metal	195260
29	construction	Scrap Wood	111580
30	construction	Trash	2890640

Figure 2: Construction waste data section

N.B.:

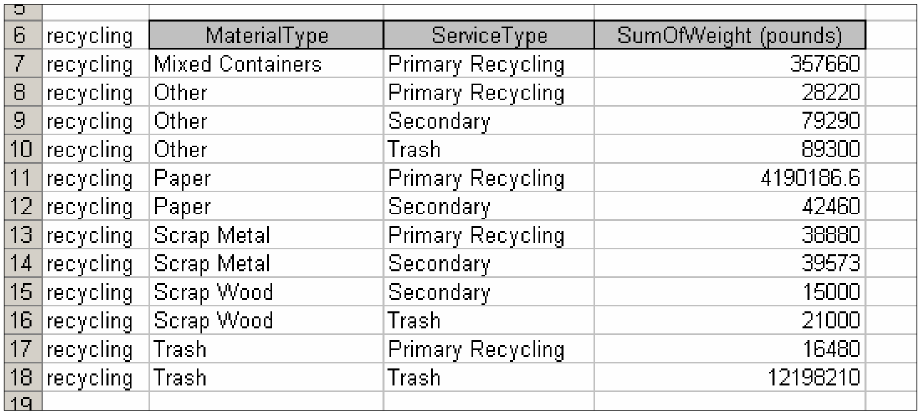
a) Same as step 2. Row 26 can be deleted or left as is, as long as “construction” is inserted in cell A26 in this case.

b) Same as step 2, except in this case cells A25 to A30 should be filled with “construction”.

c) Similar to step 2, except that “Type of Material” and “Total Weight (pounds)” should be located in column 2 and 3, correspondingly.

The next two queries were constructed by an SNRE/CSS intern in June 2006 to facilitate the data collection process. If they no longer exist in the WMS database, please refer to Appendix 1 and Appendix 2 of this WMS Instruction Sheet in order to rebuild the queries.

4) Run the “Recycling Query for OSEH/CSS”. Enter the correct date range. Copy the entire query output table and paste the results right next to cells labeled “recycling” in the template. Delete the extra row with item names. The template should look similar to Figure 3.

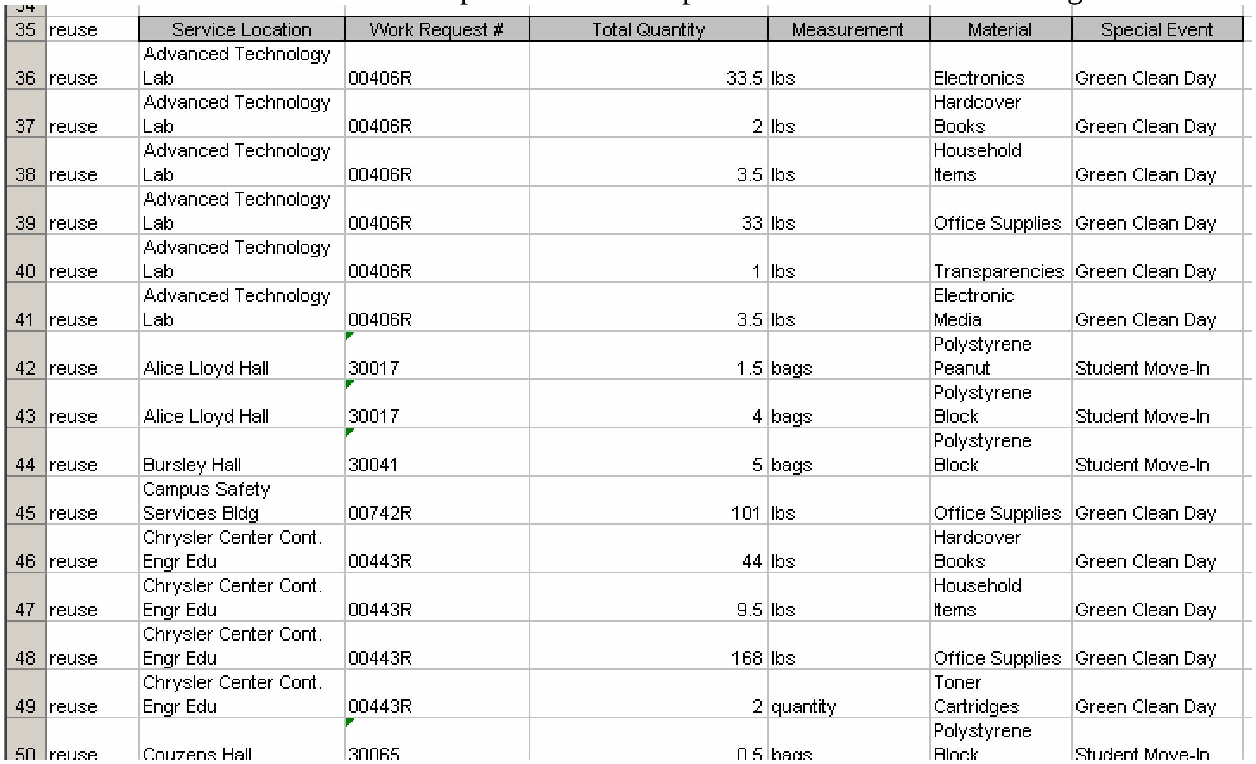


6	recycling	MaterialType	ServiceType	SumOfWeight (pounds)
7	recycling	Mixed Containers	Primary Recycling	357660
8	recycling	Other	Primary Recycling	28220
9	recycling	Other	Secondary	79290
10	recycling	Other	Trash	89300
11	recycling	Paper	Primary Recycling	4190186.6
12	recycling	Paper	Secondary	42460
13	recycling	Scrap Metal	Primary Recycling	38880
14	recycling	Scrap Metal	Secondary	39573
15	recycling	Scrap Wood	Secondary	15000
16	recycling	Scrap Wood	Trash	21000
17	recycling	Trash	Primary Recycling	16480
18	recycling	Trash	Trash	12198210

Figure 3: Recycling data section after pasting

N.B.:
a) Same as step 2.
b) Same as step 2, except in this case cells A6 to A18 should be filled with “recycling”.
c) Same as step 2, except that “MaterialType”, “ServiceType” and “Sum of Weight” should be located in column 2, 3, and 4, respectively.

5) Run “Secondary Materials Query for OSEH/CSS”. Enter the correct date range. Copy the whole table and paste the results next to the cells labeled “reuse” in the template. Delete the extra row with item names if it is present. The template should look similar to Figure 4.



35	reuse	Service Location	Work Request #	Total Quantity	Measurement	Material	Special Event
36	reuse	Advanced Technology Lab	00406R	33.5	lbs	Electronics	Green Clean Day
37	reuse	Advanced Technology Lab	00406R	2	lbs	Hardcover Books	Green Clean Day
38	reuse	Advanced Technology Lab	00406R	3.5	lbs	Household Items	Green Clean Day
39	reuse	Advanced Technology Lab	00406R	33	lbs	Office Supplies	Green Clean Day
40	reuse	Advanced Technology Lab	00406R	1	lbs	Transparencies	Green Clean Day
41	reuse	Advanced Technology Lab	00406R	3.5	lbs	Electronic Media	Green Clean Day
42	reuse	Alice Lloyd Hall	30017	1.5	bags	Polystyrene Peanut	Student Move-In
43	reuse	Alice Lloyd Hall	30017	4	bags	Polystyrene Block	Student Move-In
44	reuse	Bursley Hall	30041	5	bags	Polystyrene Block	Student Move-In
45	reuse	Campus Safety Services Bldg	00742R	101	lbs	Office Supplies	Green Clean Day
46	reuse	Chrysler Center Cont. Engr Edu	00443R	44	lbs	Hardcover Books	Green Clean Day
47	reuse	Chrysler Center Cont. Engr Edu	00443R	9.5	lbs	Household Items	Green Clean Day
48	reuse	Chrysler Center Cont. Engr Edu	00443R	168	lbs	Office Supplies	Green Clean Day
49	reuse	Chrysler Center Cont. Engr Edu	00443R	2	quantity	Toner Cartridges	Green Clean Day
50	reuse	Couzens Hall	30065	0.5	bags	Polystyrene Block	Student Move-In

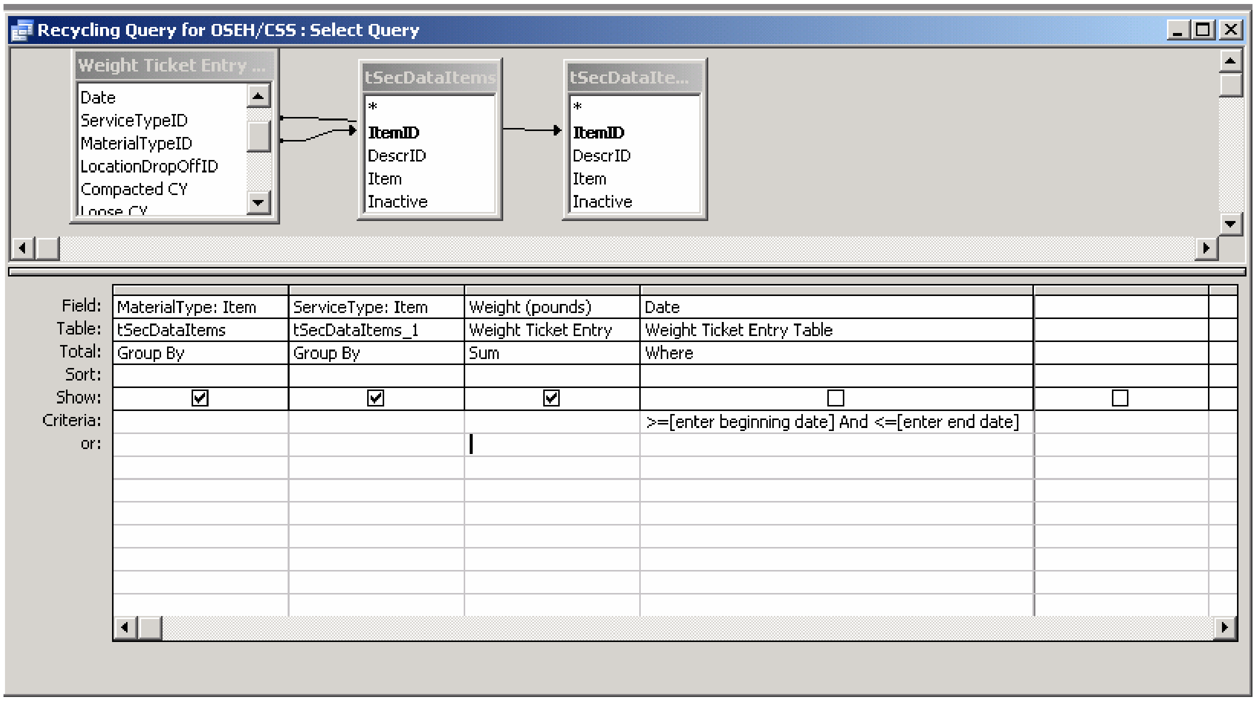
Figure 4: Reuse data

N.B.:
a) Same as step 2.
b) Same as step 2, except that cells A35 to A50 should be filled “reuse”.
c) Same as step 2, except that “Total Quantity”, “Measurement”, and “Material” should be located in column 4, 5 and 6, respectively.

After pasting all these data into the template and making sure there are no incompatibility issues, you are all set and can send us the template. Thanks for the time and support!!

Appendix 1 – Re-building the Query for Recycling Data

In MS Access, open the WMS database and click on “Create query in Design View”. Under the “Table” tab of “Show Table” box, add “Weight Ticket Entry Table” and 2 x “tSecDataItems”. Close the “Show Table” box and construct a query that **exactly** matches Figure 5 shown below.



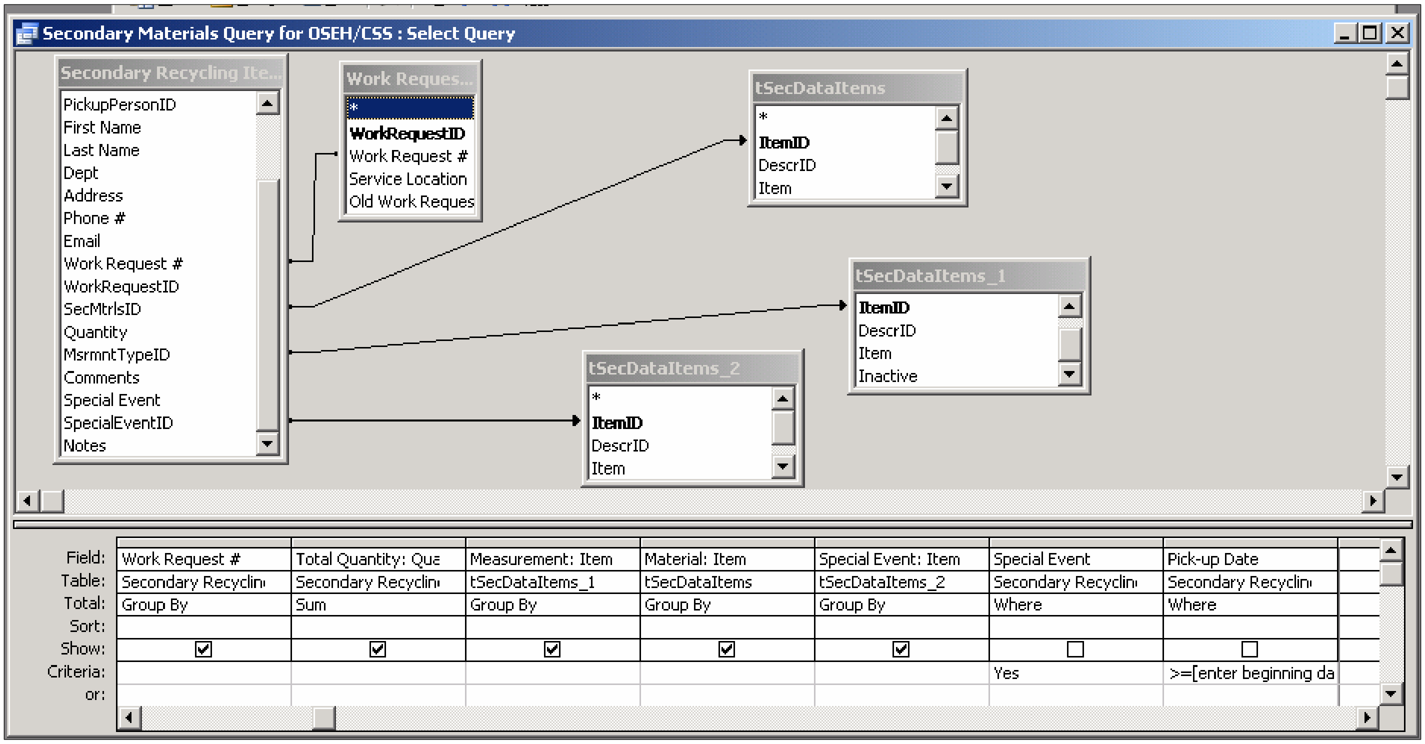
Field:	MaterialType: Item	ServiceType: Item	Weight (pounds)	Date	
Table:	tSecDataItems	tSecDataItems_1	Weight Ticket Entry	Weight Ticket Entry Table	
Total:	Group By	Group By	Sum	Where	
Sort:					
Show:	☑	☑	☑	□	□
Criteria:				>=[enter beginning date] And <=[enter end date]	
or:					

Figure 5: “Recycling Query for OSEH/CSS” created with “Weight Ticket Entry” and “tSecDataItems” tables

Notice that the sequence of fields must strictly follow Figure 5 for the template to work. Once the query design has been completed, select and run the new query following step 4.

Appendix 2 – Re-building the Query for Reuse Data

In MS Access, open the WMS database and click on “Create query in Design View”. Under the “Table” tab of “Show Table” box, add “Secondary Recycling Items Entry Table”, “Work Request Numbers Table” and 3 x “tSecDataItems” tables. Close the “Show Table” box and construct a query that **exactly** matches Figure 6 shown below.



Field:	Work Request #	Total Quantity: Qua	Measurement: Item	Material: Item	Special Event: Item	Special Event	Pick-up Date	
Table:	Secondary Recyclin	Secondary Recyclin	tSecDataItems_1	tSecDataItems	tSecDataItems_2	Secondary Recyclin	Secondary Recyclin	
Total:	Group By	Sum	Group By	Group By	Group By	Where	Where	
Sort:								
Show:	☑	☑	☑	☑	☑	□	□	
Criteria:						Yes	>=[enter beginning da	
or:								

Figure 6: “Secondary Materials Query for OSEH/CSS” created with “Secondary Recycling Items Entry Table”, “Work Request Numbers Table” and “tSecDataItems” tables

Notice that the sequence of fields must strictly follow Figure 6 in order for the template to work. Once the query design has been completed, select and run the new query following step 5.