GCE Data Toolbox for MATLAB

Table of Contents

Software Usage Agreement and Disclaimer

This material is based upon work supported by the National Science Foundation under grants OCE-9982133, OCE-0620959, OCE-1237140 and DEB- 0823293. Any opinions, findings, conclusions, or recommendations expressed in the material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Introduction

Overview

The GCE Data Toolbox is a comprehensive software framework for metadata-based processing, quality control and analysis of environmental data. The toolbox is a free add-on library to the MATLAB® technical computing language (http://www.mathworks.com/products/matlab/), based on a generalized MATLAB data model for storing tabular data along with all metadata required to process and document the data set (Appendix I). Metadata fields are queried by toolbox functions for all operations. This semantic data processing approach supports highly automated and intelligent data analysis that ensures data set validity throughout all processing steps.

All GCE-LTER data products are distributed in data structure format, and data can be imported from a wide variety of local data sources (e.g. environmental data loggers, delimited text files, database queries and standard MATLAB files), online databases (e.g. LTER ClimDB, USGS NWIS, NOAA NCDC, NOAA HADS and LTER NIS) and other frameworks (e.g. Data Turbine). Additional import filters and metadata templates can be added to the toolbox to extend support to additional data types and workflows. Interactive GUI forms are provided, along with a function library for building custom workflows for unattended processing.

This toolbox and data structure specification were developed using the MATLAB® programming language (The MathWorks, www.mathworks.com), and require MATLAB 6.5 (R13) or higher to run. MATLAB is compatible with all major computer operating systems, including Microsoft Windows®, Unix/Linux, Sun Solaris®, and Apple OS/X®.

Example Use Cases

The GCE Data Toolbox can be used for a wide variety of environmental data management tasks. Some common uses of this software are:

- Importing raw data from environmental sensors for post-processing and analysis
- Performing quality control analysis on sensor data using rule-based and interactive flagging tools
- Gap-filling and correcting data using gated interpolation, drift correction and custom algorithms/models
- Visualizing data using frequency histograms, line/scatter plots and map plots
- Summarizing and re-sampling data sets using aggregation, binning, and date/time scaling tools
- Synthesizing data by combining multiple data sets using join and merge tools
- Mining near-real-time or historic data from the USGS NWIS, NOAA NCDC, NOAA HADS or LTER ClimDB servers over the Internet
- Harvesting and integrating channel data from Data Turbine servers

Data Structures

GCE Data Structures contain data values, value qualifiers, attribute (column) metadata and general dataset metadata in a multidimensional MATLAB “struct” variable composed of named fields and typed

arrays (fig.1). Data values are stored as a series of single column arrays, each containing one type of information (i.e. a single variable) composed of an equal number of rows, representing records or observations for the corresponding data column. Each value array is paired with a matching array of qualifier flags, allowing quality control information to be stored for each data value. The major attributes of each column (i.e. data descriptor metadata, such as column name, units, description, data type, semantic variable type, precision, and quality control rules) are stored as matching arrays in dedicated structure fields. Although information is stored internally in separate fields, functions in the GCE Data Toolbox rigorously maintain the consistency of column attributes and correspondence of rows in data structures to preserve the validity of the data from operation to operation.

General metadata information is stored as a parseable array of categories, fields, and values (i.e. two-tiered hierarchy). Metadata are automatically updated to reflect changes to the structure, and can be manually edited in a GUI application. This parseable storage format permits documentation to be meshed when two structures are merged together, preserving all the information from both structures without unnecessary duplication.

All operations that are performed on a data structure are also written to a history field by toolbox functions, allowing the complete processing lineage to be displayed in the dataset metadata and viewed at any time during processing. A flexible text-based style language was developed to convert metadata to printable documentation in various styles. Tools to convert metadata to XML format are also provided with the toolbox.

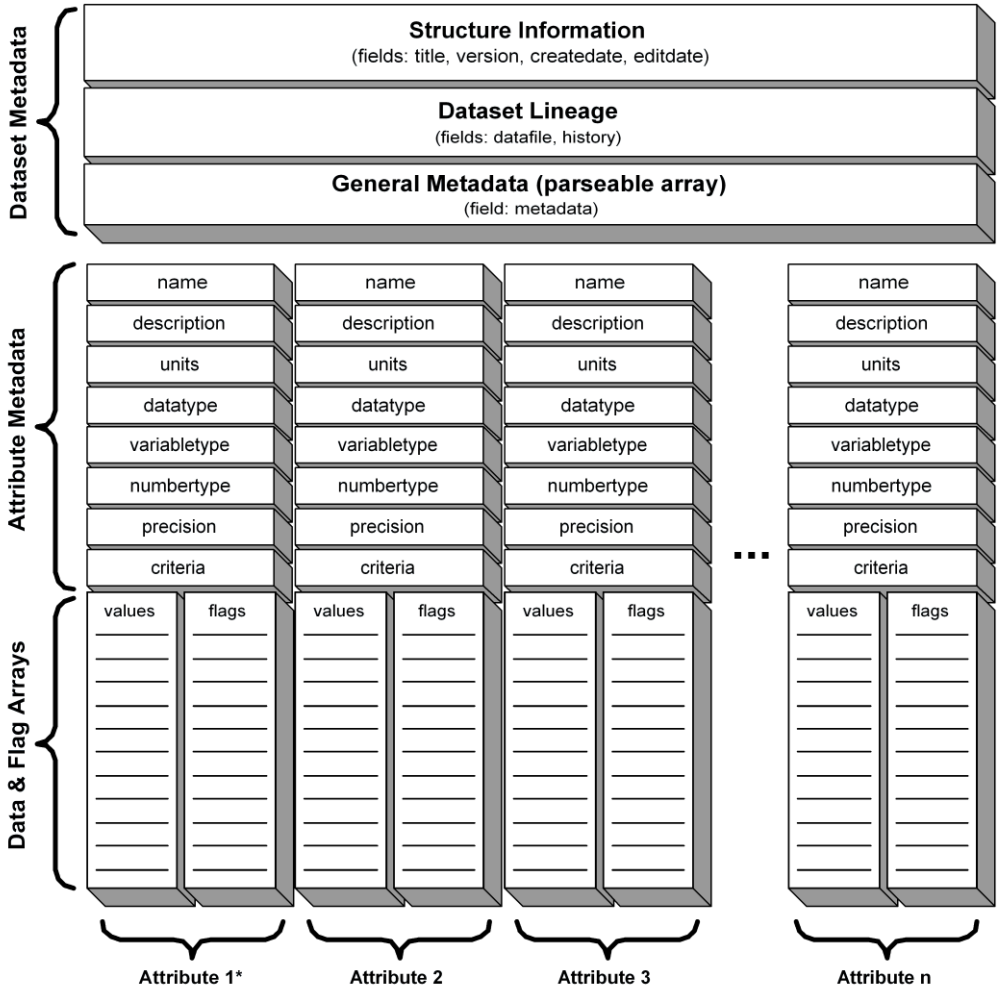

Figure 1. Conceptual model of the GCE Data Structure specification, illustrating the organization and cardinality of structure fields maintained by GCE Data Toolbox functions.

Metadata-Driven Analysis

Structure metadata fields are queried by toolbox functions for all data management, analysis, and display operations, allowing functions to process and format values appropriately based on the type of information they represent. This semantic processing approach maintains the validity of data and calculated parameters, and supports intelligent automation, such as:

- Automatic statistical report generation with appropriate statistics computed based on the data type, numerical characteristics, and variable category of each column

- Automatic unit conversions and calculation of related information, e.g. geographic coordinate system inter-conversions, date/time format inter-conversions

- Validation of column selections used for relational joins and unions (i.e. merging multiple data sets) based on variable category and unit compatibility

- Intelligent plotting of data, e.g. automatic recognition of date/time axes and encoding of text columns to allow plotting as serial integers with text displayed as labels, and automatic plotting of geo-coded data on maps

- Automatic validation of entries in the data editor application based on column data type, numerical type and precision

- Intelligent inter-conversion of data column types, e.g. conversion between numeric and text date representations

Import/Export Capabilities

Data and documentation can be imported from many sources to create GCE Data Structures, including existing data structures, delimited ASCII files, MATLAB files containing both vectors and matrices, and relational databases (requires the MATLAB Database Toolbox). Metadata can be imported along with the data (e.g. headers on ASCII files), imported from existing data structures as metadata templates, or entered manually. A series of specialized import filters have also been developed to directly parse data and documentation from specific types of data sources (e.g. SeaBird Electronics oceanographic instruments, Campbell Scientific array-based data loggers, Hydrolabs groundwater loggers) and national data centers (LTER ClimDB, USGS National Water Information System, NOAA National Climatic Data Center, NOAA NOS tide data).

Data, documentation, and statistical reports can also be exported in a wide variety of delimited ASCII text (including CSV - comma-separated value format) and MATLAB formats to support external programs or for archival purposes. Structures and variables can also be transferred to and from the base MATLAB workspace from the GUI editor application at any time to support mixed GUI and command-line processing.

Toolbox Installation and Organization

The GCE Data Toolbox is distributed as a compressed ZIP archive containing a library of MATLAB source code files (.m), MATLAB binary data files (.mat), MATLAB figure files (.fig) and other support files (e.g. .xsl, .txt, .html) in various formats organized into a series of subdirectories. In order to install the toolbox, the ZIP archive must be extracted onto a computer file system that is accessible to an instance of MATLAB 6.5 (release 13) or higher on any supported operating system. After the files are extracted, the toolbox can be used by navigating to the installation directory within MATLAB and typing "startup" to add the toolbox directories to the MATLAB path and launch the graphical startup dialog. These steps can also be automated by creating a MATLAB or operating system shortcut to simplify startup. To use the toolbox in command-line mode without the graphical dialogs, simply modify the included 'startup.m' script and remove the call to 'ui_aboutgce', or add the relevant toolbox directories to the permanent MATLAB command path using the path editor application.

Beginning with version 3.0 (September 2010), files constituting the GCE Data Toolbox are organized into a series of subdirectories based on functionality, as described in the table below. Note that directories listed as public access are included in the ZIP distribution archive, and those listed as SVN access are private and require an account on the GCE-LTER Subversion repository server to access via SVN protocols or the GCE Data Toolbox Trac software development web site (https://gce-svn.marsci.uga.edu/trac/GCE_Toolbox). Please contact Wade Sheldon (sheldon@uga.edu) for more information about accessing the GCE Data Toolbox SVN repository.

Directory	Function Category or Usage Description	Access
[root]	Startup script, GPL license file, documentation files	public
[root]/core	Core command-line functions for creating, updating, and managing data structures, analyzing data, and exporting data and metadata for archiving or analysis using other applications	public
[root]/gui	Graphical user interface functions that provide access to core library functions using operating system GUI dialogs and controls	public
[root]/parsers	Data parsing functions and import filters for loading data and metadata from various sources to create GCE Data Structures	public
[root]/qaqc	Quality control functions that can be referenced in quality control criteria rules to assign qualifiers to data values programmatically	public
[root]/plotting	Plotting and graphics functions for visualizing and analyzing data	public
[root]/mapping	Geographic functions and map figures that can be used to visualize data on maps or perform geospatial analyses	public
[root]/support	General support functions called by toolbox functions for various operations. Note that most of these functions do not require the GCE Data Toolbox API to run.	public
[root]/database	Support functions for interfacing the GCE Data Toolbox with the MATLAB Database Toolbox (not required for toolbox use)	public
[root]/xml	Support functions for working with XML and XSLT documents	public
[root]/workflows	Data harvesting workflows and workflow support functions	public
[root]/demo	Demonstration data and files for toolbox training	public
[root]/extensions	User extensions to the Dataset Editor GUI dialog	public
[root]/search_indices	Directory used to store search indices generated by the Search Engine application or downloaded from the GCE web server	public
[root]/search_webcache	Directory used to cache data downloaded from the Internet by the Search Engine or various data mining applications	public
[root]/search_temp	Directory used to store temporary files by the Search Engine application (e.g. data copied or exported from the Data Editor)	public
[root]/userdata	Data directory for user files (e.g. metadata templates, import filters, map files, custom files)	public
[root]/settings	Data directory for GUI preference files, reference data, maps, unit conversions. Many of the files in this directory are auto-generated on demand. Copying files in this directory to new toolbox installations will transfer settings and stored data	public
[root]/specialized	Specialized user-defined functions for extending the toolbox (e.g. workflow scripts built using the toolbox API, specialized import or export filters, reference data)	SVN
[root]/harvest	Specialized data harvesting workflow functions for automating data retrieval from Internet-accessible instruments and other data sources (e.g. USGS NWIS, NOAA HADS)	SVN
[root]/search_utils	Specialized functions for generating or managing search indices	SVN
[root]/gui_devel	Incomplete or provisional GUI functions still in development	SVN
[root]/build	Functions for building GCE Data Toolbox releases, including automatic generation of function documentation and startup files	SVN

Data Import/Export Reference

I. Introduction

The GCE Data Toolbox supports a wide variety of file formats for both importing and exporting data, and the toolbox can also be extended by end users to support other formats. Generalized import filters are provided for ASCII text and MATLAB file formats, as well as specialized import filters for USGS NWIS, LTER ClimDB/HydroDB, NOAA NCDC, NOAA HADS, and several common environmental data loggers. A dialog is also provided for building custom ASCII text import filters (i.e. MATLAB functions and corresponding metadata templates) and adding them to the toolbox menus. Importing data directly from relational databases via SQL query is also supported as an add-on for the MATLAB Database Toolbox. Data can be exported in a variety of ASCII text and MATLAB formats to support other programs, as well as LTER ClimDB/HydroDB harvester format for contributing data to that resource.

The GCE Data Structure specification used by the toolbox for data storage imposes strict requirements on the format and composition of tabular data sets. These requirements ensure the validity and proper interpretation of data values, but can also complicate importing data from unstructured or semi-structured text files. These requirements and tools provided for parsing text files are described in detail below.

II. Importing Data

1. MATLAB data files (.mat) -

MATLAB binary data files (.mat) can contain any native MATLAB variable type, including scalar numeric and text values and arrays of any size, multi-dimensional matrices, structures, and other objects. The GCE Data Toolbox currently supports importing scalar values and arrays from fields of a single structure variable ('struct') or from multiple individual variables stored in a .mat file. In either case array lengths of imported fields or variables must match in order to form a rectangular data set. Structure field names or variable names are used as column names, with column number suffixes added to arrays parsed from numeric matrices (e.g. Salinity_col1, Salinity_col2, etc.).

To import a MATLAB .mat file from the Data Structure Editor window, use the 'File > Import Data > MATLAB Data file' menu option, selecting 'Individual Arrays' or 'Structure Arrays' as appropriate. A dialog will then be displayed listing all compatible variables in the file and their characteristics; simply select the variables of interest and press the 'OK' button to complete the import process. To import MATLAB data from the command line or a script, use the corresponding 'imp_matlab' function directly.

2. ASCII text data files (including spreadsheet CSV files) -

Importing data from ASCII text and spreadsheet files into GCE Data Structures or any structured storage system (e.g. SQL database, R, SAS) can be simple or challenging, depending on the arrangement and consistency of the data in the file.

Ideally, text and CSV files should be structured as follows:

- A single header row containing the name of each column, delimited by tabs or commas (without internal spaces or symbols other than underscore)
- A rectangular table of data, with columns delimited by tabs or commas and each containing a single type of data (floating-point numbers, exponential numbers, integers, text strings)
- Any missing values represented as NaN (IEEE standard used by MATLAB) or empty fields
- Absolutely no non-numeric values in numeric data columns (comments, codes, flags) other than NaN

Files meeting the criteria above can generally be imported using the 'File > Load Other Data > Delimited Text File (ASCII) > Automatic Parsing' menu command, or 'imp_ascii' command-line function. However, various dialogs and functions are provided in MATLAB and by the GCE Data Toolbox to extract data from files that do not meet these ideals.

For files that contain multi-line headers without column labels adjacent to the data table, or that contain non-standard missing values codes (M, na, 9999, etc.), a filtered ASCII import dialog and command line function ('imp_filter') can be used to transform the source file prior to importing. For example, the 'File > Load Other Data > Delimited Text File (ASCII) > Custom Parsing' menu option in the Data Structure Editor opens a dialog for interactively defining custom text file import options. Column names to assign, the format string to use, number of header rows, and missing value codes to filter can be typed manually or parsed from rows in the data file using an interactive preview window, and an existing or new metadata template to apply can be specified. After suitable parameters are defined, data can be imported and a user-editable custom import filter can be added to the toolbox for future use with similarly-structured data files.

If the text file contains a variable number of fields in each row, a variable-length header, empty rows, non-numeric codes interspersed with numeric data, or other non-standard layouts then the data cannot be imported without pre-processing outside of MATLAB or development of a source-specific MATLAB import filter. Several specialized import filters are included with the GCE Data Toolbox (below), and users can contact the GCE Information Manager (gcelter@uga.edu) for advice on how to accomodate other data formats.

3. Specialized import filters -

A number of specialized import filters have been developed by the GCE LTER Project for specific data sources. These filters are listed in the Data Structure Editor 'File > Import Data' menu below the basic text and MATLAB import filters. For example:

- LTER ClimDB Data (WWW) - This filter opens a dialog to query the LTER ClimDB/HydroDB database and retrieve data over the World Wide Web (i.e. by proxying HTTP communication with the server). Information about registered sites, stations, parameters, and date ranges is retrieved from ClimDB and cached, and can be updated on demand from the query dialog. Date/time fields are automatically converted to MATLAB serial date and date component columns to support time series plotting and temporal aggregation. Site-assigned qualifier flags (other than 'G') are automatically converted to flag arrays for the respective column.

- EML Data Table (WWW) - This filter opens a dialog to retrieve an Ecological Metadata document (XML file) from a specified source, and then download and import any MATLAB-compatible text entities described. Information in the EML metadata is used to generate an m-file for retrieving and parsing the data to create a GCE Data Structure with metadata content from the original EML.

- USGS NWIS Data (WWW) - This filter opens a dialog to query the USGS National Water Information System (http://waterdata.usgs.gov/nwis) and retrieve data over the World Wide Web. Tab-delimited USGS RDB files are retrieved and parsed automatically, and measurement units are converted from English to metric equivalents based on user-editable unit mappings (see 'Edit > Unit Conversion Functions > View/Edit Enlish <-> Metric Conversions'). MATLAB serial date and date component columns are automatically generated, and USGS-assigned qualifier flags are also retained and converted to flag arrays for the respective column.

- NOAA NCDC GHCN-D Data (WWW) - This filter opens a dialog to query the NOAA NCDC server and retrieve climate data from Global Historic Climate Network stations all over the world. Downloaded files are parsed to generate a GCE Data Structure, with basic metadata added from the NCDC station database or user-specified templates. As with USGS data, values are automatically converted from English to metric units based on user-editable unit mappings and equations.

- Data Turbine Channel Data (WWW) - This filter opens a dialog for retrieving data from a Data Turbine streaming data server running on the local system (localhost) or over the Internet. Note that the DTMatlabTK must be installed and available in the MATLAB path to enable this filter (see https://gce-svn.marsci.uga.edu/trac/GCE_Toolbox/wiki/DataTurbine).

- User-editable import filters for Campbell Scientific, Sea-Bird, and other data loggers and specialized data formats, configured using Misc > Add/Edit Import Filters

Note that custom import filters can also be created using the 'File > Load Other Data > Delimited Text File (ASCII) > Custom Parsing' dialog, or manually written as MATLAB .m files, and added to the toolbox at any time. Custom import filters must accept a filename and pathname as the first two input arguments, resp., and return a valid GCE Data Structure as the first output argument. A character array can also be returned as a second output argument to convey error messages to the user, and additional input arguments can be specified as necessary. Custom filters can be added to the Data Structure Editor menus using the 'Misc > Add/Edit Import Filters' dialog, but note that only two additional input arguments other than filename and pathname are currently supported.

III. Exporting Data

Data values, QA/QC flags and data set metadata are organized within highly ordered data structures by the GCE Data Toolbox (i.e. based on the GCE Data Structure specification, http://gce-lter.marsci.uga.edu/public/im/technical_specs.htm). GCE Data Structures are stored to disk as 'struct' variables in MATLAB binary files, which can be loaded from within MATLAB on any supported

computer platform (Windows, MacIntosh, Unix/Linux). Although information in these variables can be extracted using standard MATLAB structure operations, users are encouraged to use the GCE Data Toolbox dialogs and function library (i.e. API) to export data and metadata in standard formats for use in other programs for best results.

The following export operations are currently supported:

- Standard Text Files - Data can be exported as standard text files, including tab-, comma- or space-delimited formats or in spreadsheet comma-separated value (CSV) format using the 'File > Export Data/Metadata > Text File > Standard Text File (*.txt,*.csv)' menu option or 'exp_ascii' command-line function. Various header formats, missing value codes, and metadata output options are supported, as well as options for encoding QA/QC flag information or excluding flagged values prior to export. Summary statistics reports can also optionally be appended to the export file.

- HTML/XML Files - Data can be exported in text-based markup languages, including both column- and row-oriented HTML table format, generic XML format and Google Earth KML format using 'File > Export > Text File > XML/HTML File'. Note that valid geographic data columns are required for KML export. These HTML/XML formats are useful for web-based data distribution scenarios and for creating dashboard applications for automated data harvesting applications.

- LTER ClimDB/HydroDB File - Data can be exported in the specialized text file format used by the LTER ClimDB/HydroDB harvester. The 'File > Export Data/Metadata > Text File > LTER ClimDB/HydroDB File' menu command opens a dialog for specifying the site and station codes and filename, along with other options. Note that this format requires time series data (at a daily time step) and pre-registration of sites and stations. Also, data set columns must be mapped in advance to ClimDB/HydroDB parameters. A dialog is available for defining these mappings, which can be opened using the 'View/Edit Attribute Mappings' button on the export dialog. If higher frequence data (e.g. hourly) are exported in ClimDB/HydroDB format, the data set will automatically be re-sampled to daily frequency using the 'aggr_datetime' function. In this case, be sure to use appropriate column names for the derived data set (e.g. Daily_Min_AirTemp rather than AirTemp) when defining attribute mappings. Contact the GCE Information Manage (gcelter@uga.edu) for more information about these requirements.

- MATLAB File - Data can be exported as conventional MATLAB binary files, with data columns in structure fields, as individual array variables, or numeric matrix columns using options under 'File > Export Data/Metadata > MATLAB file' or using the ‘exp_matlab’ command-line function. Q/C flags are instantiated and included as structure fields, variables or columns, as appropriate, and a padded character array containing formatted metadata is also included. These formats support using data in other MATLAB programs without using the GCE Data Toolbox function library.

- Copy Structure to Workspace - The current structure can also be copied to the base MATLAB workspace from the Editor window as the variable 'data'. This operation supports using

command-line toolbox functions outside of the GUI dialogs. The command 'ui_editor(data)' can then be used to open the modified data structure in a new Editor window.

- Copy Columns to Workspace - All or selected columns in the Data Structure Editor can also be exported to the base MATLAB workspace as named arrays to support conventional MATLAB operations.

- Copy Structure to Search Engine - The current structure can also be copied to the search results pane in the GCE Search Engine application. This operation can be performed to integrate modified data structures with other structures from searches (e.g. joins and merges/unions).

- Move Structure to Search Engine - The current structure can also be moved to the search results pane in the GCE Search Engine application (i.e. Editor window closed after copying). This operation can be performed to integrate modified data structures with other structures from searches (e.g. joins and merges/unions).

Quality Assurance/Quality Control Flagging Reference

I. Introduction

The GCE Data Toolbox for MATLAB provides a comprehensive framework for Quality Assurance, Quality Control flagging and analysis. In GCE Data Structures, the native storage format used by the toolbox, arrays of data quality "flags" (qualifiers) are created automatically whenever attributes (columns) are added to the structure. These flags are transparently maintained in synchrony with the data they describe throughout all processing steps and analyses. This separation of data values and QA/QC flags obviates the need to delete questionable values from data sets, permitting subsequent re-analysis and flexible handling and display of QA/QC information during analysis and data export.

Flags can be assigned automatically based on QA/QC criteria expressions (i.e. rules) defined for each data column, assigned manually in a spreadsheet-like data editor, or assigned graphically by selecting data points with the mouse. Criteria expressions can include simple conditionals, mathematical formulae and references to built-in or custom MATLAB functions in any combination. Criteria can also include cross-references to other data columns, and flags from multiple columns can be combined and propagated to dependent columns allowing users to perform QA/QC based on complex, multi-column dependency relationships (e.g. flagging of all measured values when a hydrographic instrument is out of the water, based on depth reading).

Flagging of invalid or questionable values in data sets is an important aspect of data processing and management, so QA/QC criteria should be defined whenever practical.

II. Automatic QA/QC Flagging

Flags can be assigned automatically to values in data columns by defining specific QA/QC criteria (i.e. rules) in the corresponding attribute metadata field (i.e. "criteria"). QA/QC criteria are MATLAB expressions that define alphanumerical flag characters to associate with column values that match the conditions specified. Basic QA/QC criteria (e.g. range or limit checks) can be defined using simple conditional statements, such as "x<0" or "x>=10", where x is a placeholder for the column values. Criteria can also reference any MATLAB statement or built-in function that returns a logical index of zeros and ones (i.e. zero for no flag, one for flag) or numerical index specifying flags to assign by array position (examples below). Custom QA/QC functions can also be referenced to assign flags based on advanced computations (e.g. statistical analysis, signal processing, time-series analysis), as long as a single logical or numerical index is returned from the function as the first output parameter. A variety of specialized QA/QC functions are provided with the GCE Data Toolbox distribution and additional functions can be added at any time and referenced in criteria.

Criteria expressions can also include cross references to other data columns, both in conditional statements and function calls, allowing complex dependency-based criteria to be defined. Column references are indicated by prefacing the respective column name with "col_" (e.g. "col_Salinity" to reference "Salinity"). The "col_" prefix can be used in place of "x" for the primary data column reference, if desired, to improve readability of criteria expressions in metadata. Note that missing values in any dependent column will cause the criteria expression to return 0 (no flag) for that value, and

incorrect column name spellings or deletion of a referenced column will cause the entire expression to be skipped; however, changes to column names and units performed in the Data Structure Editor (ui_editor) will automatically be propagated to all flag criteria expressions in the data set to maintain validity of QA/QC criteria and dependencies.

Note that QA/QC criteria defined in metadata templates are evaluated automatically whenever the template is applied to a dataset (e.g. on data import). Defining criteria in templates is therefore a powerful mechanism for providing automatic QA/QC for newly acquired or harvested raw data. Criteria are also re-evaluated automatically whenever criteria or data values are updated using GCE Data Toolbox programs, unless flags are locked by insertion of the "manual" token (see below).

III. QA/QC Criteria Syntax

Flag criteria expressions follow the pattern [condition]=[flag code], where [condition] is any MATLAB expression (or function call) that returns a logical or numerical index, and [flag code] is a corresponding alphanumeric flag code to assign when the condition is met. A GUI criteria editor is provided in the GCE Data Toolbox to simplify defining, editing and re-ordering Q/C criteria expressions. This editor can be invoked by pressing the "Edit" button next to the criteria field on the Data Editor window.

Specific syntax and examples are listed below:

1) Numeric conditionals (e.g. limit/range checks):

Syntax: x[operator][value]='[flag]' , where:
x (or col_[column name]) is an alias for values in the current data column
[operator] is ==, <, >, <=, >=, ~= (or <>)
[value] is a numeric value (scalar or array the same size as "x")
[flag] is any one text character, symbol, or digit enclosed in single quotes

Examples:

x<0='I' -- generates 'I' flags for negative values

x>=30='Q' -- generates 'Q' flags for values 30 or higher

x~=1='Q' -- generates 'Q' flags for values other than 1

2) Column cross-references (e.g. dependency checks):

Examples:

col_Depth<0='I' (in column Salinity) -- generates 'I' flags for salinity values when values in Depth are negative (out of the water)

col_Dry_Weight>col_Wet_Weight='I' (in column Dry_Weight) -- generates 'I'

flags for dry weights that exceed the total wet weight for a sample

3) Basic mathematical expressions (e.g. multi-column dependency checks):

Example:

col_Wet_Weight>(col_Dry_Weight+col_Water_Weight)='Q' (in column Wet_Weight) -- generates 'Q' flags for wet weights that exceed dry weight plus water weight (note that parenthesis can be used to control order of operations in expressions)

4) Built-in MATLAB numeric functions (e.g. statistical checks):

Examples:

isnan(x)='M' -- generates 'M' flags for any missing numerical values (NaN)

x<(mean(x)-3.*std(s))='Q' -- generates 'Q' flags for any values < 3 standard deviations below the column mean (assumes no missing values)

x<(mean(x(~isnan(x)))-3.*std(x(~isnan(x))))='Q' -- same as above, allowing for missing values

std([col_Temp1,col_Temp2,col_Temp3,col_Temp4],0,2)>0.2='Q' -- checks for excessive standard deviation of replicate sensor readings (rule would be repeated in columns Temp1, Temp2, Temp3 and Temp4). Note that the optional normalize and dimension arguments are used for the std() function to calculate non-normalized std across rows of the matrix of column values.

abs(x-mean([col_Temp1,col_Temp2,col_Temp3,col_Temp4],2))>0.5='Q' -- checks for excessive deviation from the mean of 4 redundant sensors (note that the optional dimension argument is used for the mean() function to calculate means for rows of the matrix of column values from Temp1, Temp2, etc.

5) Built-in MATLAB string functions (e.g. code checks):

Examples:

strcmp(x,'none')='M' -- generates 'M' flags for strings matching 'none'

~strcmp(x,'missing')='G' -- generates 'G' flags for strings not matching 'missing'

strncmp(x,'Spartina',8)='G' -- generates 'G' flags for strings with the first 8 characters matching 'Spartina'

6) Custom MATLAB functions (single column criteria):

Any MATLAB function that accepts column values as input and returns a logical or numeric index as its first output variable can be used in criteria expressions. Note

that a function call editor with syntax help is available from the 'Q/C Flag Criteria Editor' tool.

Examples:

flag_percentchange(x,20,20,3)='Q' -- generates 'Q' flags for any values that vary by more than 20% below or above the mean of the preceding 3 values (note: input parameters are 'value','lowlimit','highlimit' and 'framesize', resp.)

flag_notinlist(col_Plant_Species,{'Spartina','Juncus','Borrichia'})='Q' -- generates 'Q' flags for any values in 'Plant_Species' that are not in the specified list of allowed values (note that external code list files can also be referenced using this custom Q/C function)

flag_notinarray(col_SiteCode,[1,2,3,4])='Q' -- generates 'Q' flags for any values in 'SiteCode' that are not in the specified list of allowed numeric values

7) Custom MATLAB functions (multiple-column criteria):

Same as single-column custom function syntax, except addition column values are entered as function arguments, using the column reference format: col_[column name].

Examples:

flag_o2saturation(col_Oxygen,col_Temperature,col_Salinity,110,50)='Q' -- generates 'Q' flags for any oxygen values that are above 110% saturation or below 50% saturation based on the oxygen saturation calculated as a function of oxygen concentration, temperature and salinity.

flag_locationcoords(col_Site,col_Longitude,col_Latitude,0.2,'gce_locations.mat')='Q' -- generates 'Q' flags for any location names in 'Site' with longitude and latitude values that deviate more than 0.2km from the coordinates registered in 'gce_locations.mat' by dead reckoning (i.e. flags geo-referencing errors)

8) Compound criteria:

Multiple criteria can be specified for each column by using a semicolon to separate each expression. Overlapping criteria are supported, resulting in multiple flag assignments when more than one criteria is matched. Note that certain operations (e.g. encoding flags as unique integers - automatic for MATLAB file export) will only retain the first-assigned flag, therefore order of precedence should be considered when assigning multiple criteria (e.g. list rules that assign 'invalid' flags before rules that assign 'questionable' flags).

Example:

x<0='I';col_Depth<0.1='I';x>36='Q';flag_percentchange(x,20,20,3)='Q' (in "Salinity") -- generates 'I' flags for negative values, 'I' flags for values recorded when Depth was < 0.1, 'Q' flags for values > 36 and 'Q' flags for values that are 20% above or below the mean of the three preceding values.

IV. Manual QA/QC Flagging

Flags can also be assigned manually using various GCE Data Toolbox programs and utilities. For example, data values displayed on line/scatter plots can be flagged (or unflagged) visually with the mouse using the "Visual Q/C Tool" available in plot figure menus. The user just selects a column name and flag to assign, then clicks on individual values or drags a rectangle over a range of values with the mouse. Whenever flags are manually assigned or cleared, the term "manual" is appended to the criteria field for the respective data column(s) to lock the flags and prevent automatic recalculation. Automatic flagging can be reinstated by removing the "manual" token from the criteria string or by using the "Unlock Q/C Flags" option under the "Edit > Q/C Flag Functions" menu in the Data Editor window.

Similarly, flags assigned prior to importing data into the GCE Data Toolbox (e.g. flags assigned by a data provider, such as USGS, NOAA, or LTER ClimDB/HydroDB) can also be converted to flag arrays and meshed with (or replace) existing flags assigned by QA/QC criteria or manual editing. Predefined flag fields should be text columns that are named according to the convention "Flag_[column name]", e.g. "Flag_Salinity" for column "Salinity".

V. Flag Codes and Metadata

QA/QC flag codes should be documented in the metadata (i.e. 'Data' category, 'Codes' field) using the following format: "Q = questionable value, I = invalid value, M = missing", etc. This ensures that the flag codes are properly displayed in standard and XML metadata, and also allows column values codes to be automatically generated when flags are optionally converted to encoded integer columns during ASCII or MATLAB export operations or manually in the structure editor. A GUI flag definition editor is provided with the GCE Data Toolbox, which can be opened using the 'View/Edit Q/C Flag Definitions' option on the 'Edit > Q/C Flag Functions' menu.

Suggested flag codes are listed below:

I = invalid value (out of range) -- use for out-of-range/impossible values (e.g. negative mass)
Q = questionable value -- use for values outside of expected range (e.g. below detection limit, well outside of historical value range, pattern indicating data contamination)
E = estimated value -- use for values that were estimated by interpolation or other means
S = spike/noise -- use for sharp discontinuities/spikes indicating data contamination

VI. Automated QA/QC in Scripted Batch-mode Scenarios

The GCE Data Toolbox is well suited to use in scripted batch-mode data processing scenarios. Dataset metadata are used to automatically parameterize toolbox functions, so simple high-level commands can be used to carry out complex multi-step processing and analysis. All operations performed using GUI forms can be accomplished using a corresponding command line statement in a script, including propagation of flags to dependent columns, selective removal of flagged values, and automatic flagging of derived data sets (e.g. aggregated, temporally-resampled and binned data) based on number or percentage of flagged and/or missing values in primary data.

The key to performing automated QA/QC in unattended batch mode is to create a metadata template for the data source, containing appropriate QA/QC criteria (rules) for each attribute. When the template is applied to the raw data after loading or importing, QA/QC flags are automatically assigned to each attribute based on these criteria. The full suite of QA/QC-related functions can then be used to manage the display of flags in exported data products and plots, or to remove values assigned particular flags or perform other operations. Note that a GUI editor is provided with the GCE Data Toolbox for defining, managing and editing metadata templates.

Once a suitable metadata template is defined, simple functions or scripts can be used to fully process raw data files, for example:

```
[s,msg] = imp_ascii('weather.txt','d:\data\met','Weather Data','weather_template');
[s,msg] = clearflags(s,'I');
msg = exp_ascii(s,'tab','weather_qc.txt','d:\data\met','Weather Data','ST','M','FLED');
```

This script would perform the following operations:

1. import and parse a raw ASCII data file (d:\data\met\weather.txt), automatically applying the 'weather_template' metadata template and assigning QA/QC flags after import
2. remove values assigned 'I' flags, converting to NaN, retaining other flagged values
3. export the processed data in tab-delimited ASCII format, with column titles, separate metadata file (in ESA FLED style), and text flag columns following the corresponding data columns

Additional commands could also be included to fill in missing records to create monotonic time series, add derived parameters based on equations referencing data columns (each with their own QA/QC criteria), and resample or filter the data to produce derived data products that can be further manipulated and exported along with the primary data. Specialized import filters can also be defined to perform an entire prescribed workflow using a single command. Such filters are included with the GCE Data Toolbox distribution for USGS NWIS data, NOAA NCDC climate data, LTER ClimDB/HydroDB data, NOAA HADS data and other sources.

Recent versions of MATLAB also include support for timed program execution, network data access (via HTTP, FTP and UNC paths), and a SOAP web services client, allowing the GCE Data Toolbox to be used for automated remote data acquisition and QA/QC processing. At GCE, fully automated data harvesters have been developed for NOAA HADS data, USGS NWIS data, and LTER ClimDB/HydroDB data (i.e. the USGS data harvesting service for HydroDB).

VII. QA/QC Flag Handing in Post Processing

QA/QC flags are a constitutive component of GCE Data Structures, so most GCE Data Toolbox GUI dialogs and functions provide explicit options for handling flagged values in data sets during post processing and analysis. For example, flags can be displayed, ignored or removed when data are plotted, and summary statistics displays and reports can be generated with and without flagged values (or both), and numbers of flagged values are summarized for each attribute. Data export functions also provide various options for formatting flags in delimited ASCII and MATLAB files to support other programs and standards. Data integration tools (e.g. merge/union and join) also provide options for "locking" QA/QC flags to prevent inappropriate application of criteria after multiple data sets are combined.

Data aggregation, date/time re-sampling, and binning tools offer particularly fine-grained control over QA/QC flags. Values assigned specific flags can be removed prior to analysis, and QA/QC criteria can be defined automatically for derived data columns based on the number or percentage of flagged and/or missing values in each respective group, date/time interval or bin. Attributes listing the number (and percentage) of flagged and missing values are also included in derived data sets. Information on the quality and completeness of primary data can therefore be documented and preserved in derived data to guide usage and interpretation.

Data Harvesting Reference

I. Introduction

The GCE Data Toolbox can be used to develop comprehensive data harvesting workflows that include metadata generation, calculation of derived variables, automatic QA/QC checks, and statistical resampling. Harvested data, metadata and other products (e.g. plots) can then be exported in various formats for distribution. All GCE Data Toolbox functions automatically log changes and operations to the data set metadata, documenting all workflow steps automatically and greatly simplifying metadata generation for derived products as compared to other analytical software and workflow systems.

Data harvesting workflows can be run interactively from the command line, included in the Data Set Editor menus, and also set to run unattended on a timed basis as MATLAB timer objects. Demonstration workflows and utilities for data distribution and timed execution are included with the toolbox distributions, and are described briefly below.

II. Data Harvesting Workflows

A typical data harvesting worklow includes steps for importing or loading data, applying a metadata template containing documentation metadata and QA/QC rules, post-processing, and generation of data products. A wide variety of high level functions are available in the GCE Data Toolbox for performing these actions - see 'List of Functions' in the toolbox help, or open contents.html in a web browser to view descriptions of functions and their syntax, grouped by category.

Various generalized workflows are included in the /workflows directory of toolbox distributions, and sample data harvester workflows are included in /demo for study and customization, along with test data sets:

/demo/data_harvester.m -- comprehensive workflow for harvesting Campbell Scientific logger data and generating formatted data sets and plots along with web index pages

/demo/data_harvester_sql.m -- comprehensive workflow for harvesting data from an SQL Database and generating formatted data sets and plots along with web index pages

As in these examples, a workflow is typically implemented as a MATLAB function that accepts various input arguments (e.g. file names, template names, output paths, other options) and returns a status message. In addition to simplifying running the workflow, coding all processing steps in a function file allows workflows to be versioned and saved along with the data products, and even shared with other toolbox users.

III. Timed Workflow Execution

MATLAB includes built-in support for running commands on a timed basis in the background, without interfering with commands or GUI applications running interactively. This functionality is implemented using Java timer objects (see 'help timer' for more information). Any number of timer

objects can be running at the same time, and when conflicts occur events are queued and run when cpu time is available. Note that timers only operate while the MATLAB instance is running, and shutting down the MATLAB session clears the timers as well. However, multiple instances of MATLAB can be run simultaneously to prevent conflicts between automated data harvesters and interactive use of the GCE Data Toolbox or other MATLAB programs.

Timers can be created and configured manually from the command line, but the GCE Data Toolbox includes several helper functions that greatly simplify setting up and managing timed execution of workflows:

/core/start_harvesters.m -- creates time objects based on entries in the GCE Data Structure /demo/harvest_timers.mat (see below)

/core/stop_harvesters.m -- stops all or specified harvest timers and clears them from memory

/core/list_harvesters.m -- lists the name and status of all timer objects in memory (for use with stop_harvesters.m)

The harvest_timers.mat file is a GCE Data Structure with the following fields:

Name: name of the harvester (displayed by list_harvesters.m and used by stop_harvesters.m)

ExecutionMode: timer execution mode:
'singleShot' = run once (not generally used for harvesting)
'fixedDelay' = run repeatedly, with period measured from when execution starts
'fixedRate' = run repeatedly, with period measured from the designated start time (default)
'fixedSpacing' = run repeatedly, with period measured from when exeuction ends

Period: timer period in minutes

TimerFcn: m-file function to execute (e.g. data_harvester); note that function or statement must return a character array (message) as the first output or an error will result

StartTime: starting time (hh:mm:ss) of initial harvest on the designated day

StartDay: numeric day of the week to start for long-period harvests (0 = auto, 1 = Sunday 7 = Saturday)

Examples:

30-minute harvests, 15 and 45 minutes past the hour:
Period = 30
StartTime = 00:15:00
StartDay = 0

24-hour harvests at 2:30 AM

Period = 1440
StartTime = 02:30:00
StartDay = 0

Weekly harvests on Friday at 7:00 AM
Period = 10080
StartTime = 07:00:00
StartDay = 6

IV. Adding Workflows to the Dataset Editor Menus

The Dataset Editor GUI application (gui/ui_editor.m) dynamically generates menu items for metadata templates in /userdata/imp_templates.mat and import filters in /userdata/imp_filters.mat, as well as content in other reference databases (e.g. geographic references in /settings/thalweg_ref.mat). These menu items are also updated dynamically as metadata templates, import filters and other content are edited using the corresponding management applications.

In addition, the Dataset Editor can be customized by adding entries to /extensions/extensions.m. This m-file can be opened and edited using MATLAB or a text editor, and example code instructions are included as code comments. The first step is to add a 'uimenu' command for inclusion in the appropriate place in the hierarchy, and then add a code block to handle the callback events when the menu item is selected. The existing code provided in toolbox distributions can be used as a guide.

GUI Applications - Overview

A series of graphical user interface (GUI) applications have been developed to provide convenient access to most of the capabilities of the GCE Data Toolbox. These applications use standard menus, graphical controls, and platform-specific dialog boxes for input, and do not require any experience with the Matlab environment or programming language. These applications are briefly described below.

Principal GCE Data Toolbox Applications:

GCE Data Tools Startup Screen (ui_aboutgce) -- Default startup screen, providing access to the Data Search Engine, Data Structure Editor, function list, and documentation.

GCE Data Search Engine (ui_search_data) -- This application allows users to easily create and manage metadata-based search indices, and then perform detailed topical, temporal and geospatial queries to find and retrieve data sets of interest for analysis or transformation using GCE Data Toolbox programs. Search indices can include both local and web-based data holdings, and support is included for automatic registration, downloading, and local caching of public data sets from the GCE data catalog (http://gce-lter.marsci.uga.edu/public/app/data_catalog.asp). This capability provides the user with seemless access and management of GCE data alongside their own custom data sets. This application also supports many bulk operations, allowing users to copy or export large numbers of data sets at once in various ASCII text or MATLAB formats, with user-specified metadata and file formats and Q/C flag options. The entire work session can also be saved and re-loaded, allowing the user to store indices, queries and result sets for future analysis.

Data Structure Editor (ui_editor) -- This application is one of the primary starting points for all the GUI applications. It is used to create and edit GCE Data Structures, export data and metadata in various formats, and access various toolbox functions and other GUI applications via menu and button selections. Column descriptors can be edited using the controls below the variable list, and columns can be reordered, previewed, or deleted using the button panel to the right. Advanced editing commands and tools are accessed via the menubar at the top of the window.

Data Editor (ui_datagrid) -- Displays data structure values in a scrollable, resizable grid layout (i.e. spreadsheet) for display and editing. QA/QC-flagged values are displayed in red, and multiple view modes are supported. Values can be edited (with format enforcement based on metadata settings), and groups of rows can be selected for copying or deletion, with all changes individually logged to the structure processing history.

Metadata Editor (ui_editmetadata) -- Displays general metadata stored in a data structure and allows the contents of individual fields to be displayed and edited.

Auxiliary GUI Applications (i.e. invoked from primary applications or via the command line):

ASCII Export dialog (ui_exportasc) -- Dialog for saving data and metadata in data structures in a variety of delimited ASCII text formats. Various metadata formats and styles are available, and

column statistics reports can also be appended below the data table for archival validation purposes or reporting. Batch mode for processing all or selected files in a directory is also supported.

EML Package Export dialog (ui_export_eml) -- Dialog for exporting data and metadata as an EML described data package, including a CSV or delimited text file and accompanying XML metadata file containing both documentation metadata, attribute descriptors and file download specifications.

Batch Import dialog (ui_batch_import) -- Dialog for batch-processing data files in a directory for importing into the GCE Data Toolbox. This dialog supports import filter functions registed in 'imp_filters.mat', and edit fields are provided for specifying optional arguments to revise or augment defaults.

Binned Statistics (ui_bindata) -- Dialog for calculating statistics for selected columns after binning data by values in one column and optionally grouping records by values in one or more data columns

ClimDB/HydroDB Export dialog (ui_expclimdb) -- Dialog for exporting data sets as comma-delimited text files in LTER ClimDB/HydroDB format. Data sets are automatically resampled to daily intervals, values flagged 'I' (invalid) are removed, coded flag columns are created detailing flagged and missing values, and attribute names and units are converted to corresponding ClimDB/HydroDB names and units by the function 'exp_climdb', based on user-defined attribute mapping information stored in the file 'exp_climdb.mat'.

Column Calculator (ui_calculator) -- Dialog for adding new calculated columns to a data structure based on user-defined mathematical expressions. Both guided and manual expression-building is supported, and scalar results can optionally be expanded to fill the entire column.

Custom ASCII Import dialog (ui_importfilter) -- Dialog for customized importing of delimited ASCII text files. Support is provided for non-standard header formats and multiple missing value codes, and format strings and column titles can be automatically parsed or entered manually. An interactive file viewer display is also provided.

Date/Time Interval Statistics (ui_aggrdatetime) -- Dialog for creating customized statistical summaries for specifed date/time intervals (yearly, monthly, daily, hourly), optionally grouping by values in one or more non-date/time columns.

Documentation Viewer (ui_viewdocs) -- Dialog for view the GCE Data Toolbox documentation. Individual sections can be navigated using the drop-down menus and buttons at the top of the screen.

Grouped Statistics (ui_aggrstats) -- Dialog for creating customized statistical summaries of values in a data structure by specifying a series of columns to sort and group by and a series of columns to automatically calculate relevant statistics for. The results are returned in a separate editor window for further customization and analysis or export.

Interpolate Missing Values (ui_interp_missing) -- Dialog for filling in gaps in a data set using one-dimensional interpolation.

Join Data (ui_joindata) -- Dialog for joining two data structures based on common values in one or more matching 'key' columns, creating a new structure containing the key columns and user-selected data columns from both structures (optionally renamed by adding text prefixes to distinguish identical column names). All standard relational join types are supported (i.e. inner, left, right, and full outer).

LTER ClimDB/HydroDB Data Harvester (ui_fetch_climdb) -- GUI dialog for harvesting data from the LTER ClimDB/HydroDB database (climy.lternet.edu) over the Internet. A list of stations is provided for look-up by site, and data can be retrieved for specific date range and variables registered for each station.

Map Data (ui_mapdata) -- Dialog for plotting values or text in a data structure containing georeference columns on a map plot as symbols, text labels, or color-mapped patches with a color scale bar.

Metadata Style Editor (ui_metastyle) -- GUI dialog for creating, editing and managing metadata style definitions stored in 'metastyles.mat', which are used to generate formatted metadata for preview and file export. Style definitions specify general word wrap and indent options, and include any number of format description rows consisting of static text, expressions combining metadata fields with static text and/or MATLAB function output, and customized indent level and word wrapping options.

NCDC Data Harvester (ui_fetch_usgs) -- GUI dialog for harvesting data from the NOAA National Climatic Data Center WWW site. Daily data can be retrieved for any supported station for the specified date range. A list of stations is provided for look-up by state.

Plot Data (ui_plotdata) -- Dialog for creating multiple-Y vs. X symbol/line plots of values in a data structure, optionally restricting values by values in one specified column (inline query).

Plot Groups (ui_plotdata) -- Dialog for creating a Y vs. X symbol/line plot of values in a data structure after grouping rows by values in a specified column. One line segment is produced per group after optionally restricting values by values in another specified column (inline query).

Plot Vertical Profile (ui_plotvertprofile) – Dialog for creating a 3-D contour plot of a data set variable vs distance and depth.

Q/C Flag Copying Dialog (ui_copyflags) -- GUI dialog for copying composite flags from one or more columns and adding them to or replacing the existing flag arrays of one or more other columns (used to propagate flags to dependent/calculated columns)

Q/C Flag Criteria Editor (ui_qcflags) -- GUI dialog for editing QA/QC flag criteria for the active column in the Data Structure Editor.

Q/C Flag Definition Editor (ui_flagdefs) -- GUI dialog for editing QA/QC flag code definitions and descriptions of data anomalies stored in the metadata of the current structure in the Data Structure Editor.

Query Builder (ui_querybuilder) -- Dialog for interactively building a custom query string to select rows in a data structure by value and create a subset structure.

Search/Replace Text (ui_string_replace) -- Dialog for searching and replacing text in a specified string data column.

Selective Flag Removal Dialog (ui_clearflags) -- GUI dialog for selectively deleting data values or data rows based on QA/QC flag assignments. Subsets of data columns and flag definitions can be selected from lists of all available columns and definitions. The functions 'nullflags' or 'cullflags' are used to clear the affected values or rows, resp.

Sensor Drift Correction (ui_correct_drift) -- Dialog for correcting data values for sensor drift over a specified date range. Several methods are supported, including constant offset, linearly-weighted offset and custom-weighted offset (e.g. for non-linear corrections).

Sort Columns (ui_sortcolumns) -- Dialog for bidirectionally sorting rows in a data structure based on values in one or more specified columns.

Statistics Report Builder (ui_statreport) -- Dialog for generating customized column statistics reports in various delimited ASCII formats.

Template Editor (ui_template) -- GUI dialog for creating, editing and managing metadata templates used by the data structure editor and import filters to assign column descriptors and boilerplate metadata to new data structures based on column name and unit matching. Templates stored in the data file 'imp_templates.mat' are converted to data structures and opened in the data structure editor for inspection and editing.

Title Editor (ui_title) -- Displays the title of a data structure for editing.

Top/Bottom Values (ui_topbottom) -- Dialog for for extracting top and bottom data records from a data structure containing vertical profile data based on values in a depth or pressure column.

Unit Conversion (ui_unitconv) -- Dialog for performing unit conversions on an individual data column using predefined or user-customized multipliers or equations. The selected conversion formula is added to the metadata as a calculated column definition.

USGS Data Harvester (ui_fetch_usgs) -- GUI dialog for harvesting data from the USGS WWW server. Real-time, daily and finalized data can be retrieved for any supported station for the specified date range. A list of Real-time stations is provided for look-up by state.

Visual Q/C Tool (ui_visualqc) -- GUI dialog for assigning and clearing QA/QC flags visually by clicking on data points with the mouse after structure columns are graphed as scatter/line plots using 'ui_plotdata' or 'ui_plotgroups' (note: plots of encoded text columns are not supported).

GUI Applications - Dataset Editor

I. Introduction

This application is one of the primary starting points for all the other GCE Data Toolbox applications. It is used to create and edit GCE Data Structures, export data and metadata in various formats, and access various toolbox functions and other GUI applications via menu and button selections.

To begin working with the editor, use commands under the 'File' menu to load an existing data structure stored in a MATLAB .mat file (i.e. 'Load Data Structure') or create a new structure by loading data from another type of data file listed under the 'Load Other Data' menu (e.g. a delimited ASCII text file, a conventional MATLAB file containing data in arrays and matrices, ClimDB or USGS data retrieved over the Internet, or various custom formats). Alternatively, you can use the 'Add' button to create fields in a blank structure and assign all metadata descriptors manually to create an empty structure for manual data entry or to use as a metadata template. Note that most menu options and buttons are disabled until a structure is loaded or created.

II. Main Screen

When a structure is present, the controls on the main screen allow the metadata properties of each column to be viewed and edited. The current name and units of each column are displayed in the 'Column List' box. Columns are listed in data set order, with the leftmost column displayed at the top. The data descriptors for the selected column are displayed in the edit boxes and popup menus below the list.

Descriptors can be edited by clicking in each field and entering new text or selecting another option on the dropdown menu. Invalid selections will be overridden (e.g. spaces in column names will be converted to underscores) or reset to prior values and an error message displayed (e.g. selecting 'alphanumeric string (s)' for a numerical column). Note that changes to text in edit boxes will not be applied until another control is selected or the 'Enter' key is pressed.

Multiple columns can be selected for deletion, exporting to the base MATLAB workspace and other menu commands that support multiple selections (e.g. 'Convert Column Data Types'). Multiple listbox selection techniques vary by computer platform, but generally dragging with the mouse or pressing the 'Shift' key before left-clicking will select contiguous rows, and pressing 'Ctrl' or 'Option' and left-clicking will select non-contiguous rows. Note that all descriptor fields and command buttons other than 'Delete' and 'Restore' are automatically disabled when multiple rows are selected.

Command buttons are provided next to certain descriptor fields for editing the corresponding values in separate GUI dialogs. For example, the 'Edit' button next to the 'Flag Criteria' field opens a dialog for creating and editing QA/QC flag criteria strings, which are automatically formatted and returned to the 'Flag Criteria' field upon completion. When 'Variable Type' is specified as 'coded values (coded)', a 'Codes' button will be displayed for editing value code definitions for the selected column that are stored in the data structure metadata.

The buttons to the right of the column list can be used to perform various operations on the selected column, as follows:

Move First/Move Up/Move Down/Move Last: Repositions the selected column within the list, and consequently within the data structure.

Preview: Displays the top 1000 or fewer records of the current column in a scrolling list box to enable format selections to be previewed.

Histogram: Displays a frequency histogram plot for the selected column to provide a quick overview of the range and distribution of values in the column.

Manual QA/QC: Opens a GUI dialog for manually assigning or editing QA/QC flags assigned to the selected data column.

Add: Adds a blank column to an empty data structure for creating a metadata template (disabled for structures containing data - use 'Edit|Add Data Columns' instead).

Delete: Deletes the selected column from the list and from the data structure.

Restore: Restores all deleted columns to the list and structure.

Convert: Opens a dialog for performing predefined or user-customized unit conversions on the selected column (with formulas added to the metadata).

Edit (Flag Criteria): Allows the Q/C flag criteria for the selected column to be defined or edited in a GUI dialog.

III. File Menu

This menu contains commands for loading/importing and saving/exporting data structures, as well as closing the editor and quitting the MATLAB session. Specific functions are as follows:

Return Data: (Note: this command only appears when the editor window is invoked by another toolbox dialog) Returns the edited data set to the original toolbox dialog that invoked the editor session for modifying the data structure prior to some other operation.

Load Data Structure: Loads a new GCE Data Structure into the editor from the specified location. Note that an overwrite warning message will be displayed if the current structure has been altered since it was loaded or last saved.

- Load Structure from File: Opens a dialog box to select a MATLAB (.mat) file containing a GCE Data Structure to load. If more than one valid structure is present in the file, a selection list will be displayed to choose among the structures by variable name.

- Load Structure from Workspace: Displays a list of structure variables in the base MATLAB workspace to load (or displays an error message if no structure variables are present).

Load Other Data: Loads data (and metadata as possible) from various file formats, including:

- Delimited Text File (ASCII): Loads an ASCII text data file containing columns delimited by tabs, commas, or spaces (optionally containing metadata in a formatted header) to create a new data structure. The 'Automatic Parsing' option attempts to automatically identify file header, delimiter, and data type characteristics by parsing the first few data rows that are identified, and the 'Custom Parsing' option opens a GUI dialog for interactively setting various parsing options to support loading non-standard file formats (e.g. files with multiple or custom missing value codes, compound attributes, or hard-to-identify column data type characteristics).

- MATLAB Data File: Loads selected variables from a standard MATLAB data file (containing either arrays or matrices) to create a new data structure with columns named after the source variables.

- MATLAB Workspace Variables: Loads selected variables from the base MATLAB workspace. If the 'Individual Arrays' option is selected, one or more compatible numeric, character and cell arrays can be selected from a list and imported to create a new data structure with clumns named after the source variables. If 'Structure Arrays' is selected, then a single structure can be imported to create a data set with columns created for each field.

- LTER ClimDB Data (WWW): Opens a GUI dialog for retrieving data for any station registered in the LTER Network All-site climate and hydrology database hosted at Andrews LTER to create a new data structure (requires networking features in MATLAB 6.5+)

- USGS NWIS Data (WWW): Opens a GUI dialog for retrieving near-real-time or historic daily data for any station in the USGS NWIS database to create a new data structure (requires networking features in MATLAB 6.5+)

- NOAA NCDC GHCN-D Data (WWW): Opens a GUI dialog for retrieving long-term daily climate data for any station in the National Climatic Data Center Global Historic Climate Network - Daily database to create a new data structure (requires networking features in MATLAB 6.5+).

- Data Turbine Channel Data (WWW): Opens a GUI dialog for retrieving the latest channel data for a specified source on a Data Turbine server (requires networking features, Java 1.6+, and the DTMatlabTK software library in the MATLAB path)

- [Custom file filter]: Loads data and metadata using a custom filter to create a new data structure (note that custom filters are MATLAB functions that are registered using the dialog invoked by the 'Options|Add/Edit Import Filters' command, and filters can be added, renamed or deleted at any time to update toolbox capabilities)

Import Metadata: Loads column descriptor and general metadata from another data structure or named template stored in 'imp_templates.mat' and updates descriptors for existing data columns that match template entries. Unmatched columns are not updated and units of matching columns are only

updated if existing units are blank or 'unspecified' (warnings are issued for each unmatched column or units conflict).

Options include:

- Existing Data Structure: Imports column descriptors and general metadata from another data structure saved to disk

- Standard Template: Imports column descriptors and general metadata from a named template stored in 'imp_templates.mat' -- see Edit|Metadata Functions|Edit Metadata Templates)

Batch Import Files: Opens a GUI dialog for batch processing all or selected files in a directory using a specified import filter (defined using Misc > Add/Edit Import Filters) and arguments. GCE Data Structure files are generated in a specified directory, named according to each imported data file (e.g. mydata.txt imported as mydata.mat).

Batch Export Files: Opens a GUI dialog for batch exporting all or selected GCE Data Structures in a directory as delimited text files with various format options. This is the same dialog as Export Data/Metadata > Text File (ASCII) > Standard Text File, except export is performed for files on disk and not data structures in memory.

Join Data Sets: Loads data from an existing data structure stored on disk or in another open editor window and integrates it with the current structure to create a composite data set. Columns from each data set are aligned by joining data records on values in one or more compatible "key" columns, resulting in a derived data set containing key columns and one or more output columns chosen from each original data set. All standard relation join types are supported (i.e. inner, left outer, right outer, and full outer) for flexibility in handling unmatched records. Metadata are also meshed by performing a field-by-field comparison and concatenating any fields with content differences to preserve all information in both data sets.

Options include:

- Manual Key Selection: opens a GUI dialog for manually selecting key columns, join options, and output columns for the integrated data set.

- Automatic Date/Time Join: opens a GUI join dialog then identifies compatible date/time columns in each data set and automatically adds corresponding join conditions, simplifying integration of two time-series data sets.

Merge Data Sets: Loads data from an existing data structure stored on disk or in another open editor window and integrates it with the current structure by concatenating records. Columns are automatically aligned by matching on name, datatype, and units to ensure compatibility, and any unmatched columns are retained with records offset as necessary with null values. All records from the second structure can be added to the end ('Append New Data') or beginning ('Prepend New Data') of the current structure. Additionally, time series data sets can be merged by date based on date/time information stored in each data set to create a single data set with records from the older

data set ('Time-Series Merge (overwrite older records)') or newer data set ('Time-Series Merge (add newer records)') with overlapping date/time values automatically removed to form a continuous time-series. Metadata are also meshed by performing a field-by-field comparison and concatenating any fields with content differences to preserve all information in both data sets.

Save Data Structure: Saves the current structure to disk as a MATLAB binary data file (*.mat).

- Standard File: saves the structure as the variable 'data', overwrititing any existing variables if the .mat file already exists in the specified directory

- Named Variable: saves the structure as a user-specified variable, retaining any other variables if the .mat file already exists in the specified directory

Export Data/Metadata: Exports structure data and metadata in various file formats, including:

- Text File: options for exporting data and metadata in various text formats

  - Standard Text File (*.txt/*.csv): opens a GUI dialog for exporting the data set in various delimited ASCII text file formats with multiple file header, metadata, and QA/QC flag formatting options

  - XML/HMTL File: options for exporting data, QA/QC flags and selected metadata in various XML and HTML implementations, including Keyhole Markup Language (KML) format (requires geographic coordinate columns)

  - LTER ClimDB/HydroDB File: opens a GUI dialog for exporting the current data set in the ClimDB exchange format to create a ClimDB/HydroDB harvest file

- Toolbox ASCII Import File: Saves the documentation and attribute metadata and optionally data values as a text file specially formatted for automatic parsing by imp_ascii.m; these commands can be used to generate headers to simplify parsing of commonly-imported data formats or serializing metadata and data for archiving and subsequently reloading into the Toolbox.

- MATLAB File: exports the data set and metadata as variables in a standard MATLAB *.mat binary file. Note that if a metadata style is specified, files will contain a padded character array of formatted metadata in a separate 'metadata' variable

  - Struct Variable: exports data values as individual numeric or cell arrays named according to the data set column name, along with encoded integer QA/QC flag columns

  - Individual Array Variables: exports data values as individual numeric or cell arrays named based on the data set column, along with encoded integer QA/QC flag columns

  - Matrix Variable: exports data values in a single n row x m col numeric matrix variable ('data'), with QA/QC flag columns and any string columns encoded as integers. Cell arrays listing column names ('columns'), column units ('units'), column descriptions ('descriptions') are also included as variables

- EML Data Package: opens a GUI dialog for exporting the data structure as a delimited text file accompanied by an XML metadata file compliant with the Ecological Metadata Language (EML) 2.2 schema. The EML file will contain data set and attribute metadata, as well as physical file download descriptors, suitable for depositing data packages in an archive.
- Copy Structure to Workspace: copies the existing structure to the base MATLAB workspace as the variable 'data' for command-line processing using the GCE Data Toolbox

- Copy Columns to Workspace: copies all or selected data columns to the base MATLAB workspace as variables named according to the column names for running standard MATLAB scripts and functions from the command line

- Copy Structure to Search Engine: copies the structure to the GCE Data Search Engine search results list for data set management or integration with data sets returned from searches

- Move Structure to Search Engine: copies the structure to the GCE Data Search Engine search results list for data set management or integration with data sets returned from searches, then closes the editor window

Clone Data Structure: opens all or selected columns of the active data structure in a new Dataset Editor window to create a cloned data structure (e.g. for experimenting with a toolbox feature or for making temporary edits prior to export)

Clear Data: Clears all data from the current editor window for creating or loading a new data structure.

Close Window: Closes the current editor window and returns to the start-up screen (if specified) or another open window.

Exit MATLAB: Closes all toolbox windows and exits the MATLAB session (with data save warning).

IV. Edit Menu

This menu contains general functions for editing the content of GCE Data Structures. All operations are performed on the current structure, unlike functions in the Tools menu which generally create derived structures that are opened in new editor instances (see below).

View/Edit Title: Displays the title of the data structure (edits are applied both to the structure title field and 'Title' field of the 'Dataset' metadata category).

View/Edit Data: Opens a separate GUI data "grid" application for viewing and editing data set values stored in the structure. Values are displayed according to the datatype and precision settings for the column, and new entries are validated using these settings (i.e. excess digits are rounded to the column precision). All value changes and row inserts and deletions are logged to the processing history field when the data is returned to the structure editor using the 'File|Return Data' menu command).

Add Data Columns: Adds data columns to the current structure, as follows:

- Add/Update Column(s) from Workspace: Opens a list of compatible variables in the base MATLAB workspace to add to the data set (i.e. numeric or cell arrays with one scalar dimension and length matching the number of rows in the current data set). If 'Update Existing Columns' is specified, selected variables with names matching a existing columns will be used to update corresponding column values, otherwise all selected columns will be added to the data set with provisional metadata descriptors.

- Add Calculated Columns: Opens a GUI application for creating one or more calculated columns based on user-defined mathmatical expressions involving other data columns, constants, and MATLAB functions.

- Add Empty Integer Column: Adds a column of NaNs with datatype 'integer' and precision 0

- Add Empty Floating-Point Column: Adds a column of NaNs with datatype 'floating-point' and precision 2 (note that this column can be converted to 'exponential' by modifying the 'Data Type' selection and precision without data loss)

- Add Empty Text Column: Adds a column of empty strings with datatype 'string'

Copy Data Columns: Copies all or selected columns in the data set, appending "_copy" to the to the original column names. All column metadata properties other than name will be retained in the duplicated columns.

- All Columns: Copies all data set columns and appends them to the end of the data set

- Selected Column(s): Copies only selected columns and appends them to the end

Convert Column Data Types: Converts values in all or selected columns to the specified data type, transforming values and updating column data type, numerical type and precision as appropriate. The specific MATLAB expression used for the conversion is documented in the structure processing history. Note that conversions from numeric-to-string or string-to-numeric will cause any QA/QC flag criteria defined to be cleared to prevent syntax errors; however, if any flags are assigned the criteria 'manual' will be added to lock the flags and prevent their removal on recalculation. Also note that conversions of string columns assigned the variable type 'date or time' to any numeric format will be performed using a type-specific filter to generate MATLAB serial date values (i.e. from formatted date/time strings). Similarly, string columns with variable type 'geographic coordinates' will be converted to decimal degrees (i.e. from formatted latitudes or longitudes in degrees, degrees minutes, degrees minutes seconds formats, with our without special symbol characters and hemisphere codes). If conversions fail for all values in a column then the values will not be updated and a warning will be displayed. Also, requested conversions to the data type already assigned will be skipped.

Encode Text Columns As Integers: Converts all text columns to coded integer columns by replacing unique string values in each column with a unique integer and documenting the code values in the metadata (Data/ValueCodes).

- Update Existing Code Definitions: Reconciles the new code definitions with any existing definitions in the metadata for each coded column (e.g. if column 'Species' was already encoded with definitions 'Species: A1 = Spartina alterniflora, A2 = Spartina cynosuroides', the definitions would be updated to reflect the new integer assignments "Species: 1 = Spartina alterniflora, 2 = Spartina cynosuroides")

- Append to Existing Code Definitions: Adds new code definitions to any existing definitions in the metadata, preserving all prior definitions (useful if other metadata fields contain references to original alphanumeric code values)

Decode Coded Columns: Decodes values in coded columns based on code lists in the metadata (Data/ValueCodes), and adds categorical text columns containing code definitions. The decoded text columns are named based on the original column with "_Decoded" appended. Note that non-code columns with be ignored regardless of selection option.

- All Coded Columns: Decodes all columns with a Variable Type of "code"

- Selected Column(s): Decodes only the selected coded columns(n)

Set Precision As Displayed: Rounds or truncates trailing digits from all or selected numeric data columns to set precision equal to the display precision specified in the metadata (note that all numeric data is stored internally in double-precision floating-point format regardless of data type or precision specified).

- Round to Nearest Digit: Sets precision by rounding the nearest terminal digit (e.g. 12.3456789 precision 1 --> 12.3)

- Round Up (Ceiling): Sets precision by rounding the terminal digit up (e.g. 12.3456789 precision 1 --> 12.4)

- Round Down (Floor): Sets precision by rounding the terminal digit down (e.g. 12.3456789 precision 1 --> 12.3)

- Truncate (Fix): Sets precision by truncating trailing digits (e.g. 12.3456789 precision 3 --> 12.345)

Coalesce Multiple Columns: Coalesces data in two or more compatible columns by filling in missing values (empty strings or NaN) in the first-selected column with valid values from subsequence columns sequentially. This command can be used to consolidate data in duplicate or equivalent data columns resulting from merge or join operations (e.g. data offsets resulting from merging data sets with differences in column name or units spelling). Note that data types and units must match, and columns will be coalesced in list order regardless of selection order, so make sure to arrange columns appropriately before performing this operation.

- Delete Coalesced Columns: Specifies that columns should be deleted after successfully combining values, retaining only the first-selected column

- Retain Original Columns: Specifies that all original columns should be retained, and only the first column will be updated

Concatenate Column Values: Concatenates (combines) values in two or more selected columns to generate a single text column, with values optionally separated by a character string. Note that any numeric columns will automatically be converted to strings based on their data type and precision settings. The derived column will be added to the data set after the last selected column, and will be named by concatenating the original column names separated by underscores (e.g. Year_Month_Day when concatenating Year, Month and Day columns).

Split Column Values: Splits values in a selected string column using a specified delimiter to create multiple string columns.

Encode Text Columns as Integer: Replaces values in selected text columns with unique integers and documents the codes in the metadata.

Decode Coded Columns: Replaces values in all or selected columns with variable type 'code' with code definitions in the metadata.

Sort Records: Opens a GUI dialog for sorting data records (rows) by values in one or more specified columns.

Search/Replace Data Values: Opens a GUI dialog for searching and replacing text or numeric values in a selected data column.

- Search/Replace Text: Replaces matched text in a specified a string data column based on a case-sensitive or insensitive full or partial match (using regular expression matching).

- Search Replace Number: Replaces matched values (or NaN) in a specified numeric column with the specified number (or NaN)

Interpolate Missing Values: Opens a GUI dialog for filling in gaps in a data set using various one-dimensional interpolation methods. Interpolation requires a floating-point or exponential X data column that steadily increases or decreases without duplication, such as a serial date or distance column. Interpolation can be performed for the entire data set or for data grouped by values in one or more columns.

Calculate Missing Values: Opens a GUI dialog for filling in gaps in a data set using a valid MATLAB expression or scalar value. Expressions can reference values in other data columns or any function available in the MATLAB search path.

Correct for Sensor Drift: Opens a GUI dialog for correcting for sensor drift in data set columns by applying constant, linearly-varing or custom-weighted corrections to values over a specified range of dates, optionally flagging the revised values.

Remove Duplicate Records: Searches for records with identical values in all columns ('All Columns Duplicated' option) or just non-data/non-calculation columns (Non-data Columns Duplicated option), and deletes all but the first occurrence. Note that the structure will first be sorted by values in all or non-data columns, resp.

Remove Empty Records: Removes records (rows) from the data structure which contain only null values (NaN or '') in all columns ('All Columns Empty' option) or data/calculation columns ('All Data Columns Empty' option).

Remove Leading/Trailing Blanks: Trims insignificant white-space from all or selected text columns in the data set.

Remove Empty Columns: Removes all empty columns from the structure (i.e. numeric columns containing only NaN and text columns containing only empty strings).

Unit Conversion Functions: Performs batch unit conversions on data columns using predefined unit conversion tables, as follows:

- Convert English Units to Metric: Applies English to metric unit conversions by matching column units to a table of conversions (see below)

- Convert Metric Units to English: Applies metric to English unit conversions

- View/Edit Unit Conversions: Opens the master table of unit conversions into the data structure editor, allowing the user to view, edit, add and delete specific conversions and equations used by the manual and automatic conversion routines

- View/Edit English<->Metric Conversions: Opens the English-Metric unit matching table in the data structure editor, allowing the user to view, edit, add and delete matches used by the automatic English/Metric conversion routines above. Note that corresponding unit conversion parameters must also be present in the master conversion table (including reciprocal conversions).

Date Functions: Adds new datetime columns to the current structure, calculated from existing datetime columns, as follows:

- Convert Date/Time Format: Converts the format of selected date/time column(s) to a specified numeric serial date or text format. Note that all columns must be of variable type 'datetime', and that string columns must be in a date/time format recognized by the MATLAB datenum() function. Also note that if a time format is specified for a column containing dates, the date portion will be lost.

- Date Components from Date Column: Adds Year, Month, Day, Hour, Minute and Second columns (as necessary) calculated from a formatted text date or numeric serial date column named Date (i.e. with Variable Type 'datetime' and Data Type 'f' or 's')

- Date From Date Components: Adds a formatted date or serial date column calculated from Year, Month, Day, Hour, Minute, Second columns (not all required, but must be named as above and with variable type of 'datetime')

- Year Day from Date/Time Column(s): Adds YearDay column (days since January 1 00:00 of the same year) calculated from numerical or string date/time column(s)

- Expand Date Gaps (time series data): Fills in missing date/time records to complete a partial time-series data set with even date-time intervals (i.e. monotonic dates). Records for data and calculation columns will be filled with appropriate empty values (NaN or ''). If 'Replicate Values in Non-data Columns' is specified, then records for non-data/calculation columns (e.g. categorical, text, coord) will be replicated when values before and after the gap match, otherwise will be filled with empty values.

Geographic Functions: Adds new geographic coordinate columns calculated by transforming coordinates in existing structure columns, as follows:

- Calculate UTM from Latitude/Longitude: Adds columns UTM_Zone, UTM_Easting, UTM_Northing and UTM_Hemisphere calculated from columns Latitude and Longitude (in decimal degrees with Data Type 'f', Variable Type 'coord', Number Type 'continuous') using the specified ellipsoid datum.

- Calculate Latitude/Longitude from UTM: Adds columns Latitude and Longitude calculated from columns UTM_Zone, UTM_Easting, UTM_Northing and UTM_Hemisphere (if present). UTM_Easting and UTM_Northing must be in units 'm', and have Data Type of 'f' and Variable Type 'coord'. UTM_Zone must contain integer zone numbers (Data Type 'd') or zone plus hemisphere (Data Type 's', e.g. '17N') and also have a Variable Type of 'coord'. If UTM_Zone does not contain hemisphere values and a UTM_Hemisphere column with Data Type 's' and Variable Type 'coord' is not present, a value of 'N' indicating northern hemisphere will be assumed.

- Lookup Study Sites for Coordinates: Adds a column of site codes calculated by matching values in geographic coordinate columns to sites in the GCE geographic database based on position relative to site bounding polygons. - 'Any Site Type' compares coordinates against all GCE site polygons - 'Only Marsh Sites' compares coordinates against only marsh site polygons - 'Only Transects' compares coordinates against only transect polygons

- Lookup Coordinates for Stations/Locations: Adds geographic coordinate columns to a data structure by matching station or location codes to entries in the GCE geographic database - 'Latitude/Longitude' adds Latitude and Longitude coordinates in decimal degrees - 'UTM (WGS84)' adds UTM easting, northing, zone and hemisphere columns

- Lookup Stations/Locations from Coordinates: Uses geographic distance calculations to look up registered stations or location names for GPS coordinates in the data structure (Latitude m). Lookups can optionally be constrained to a specific type of location. (Note: geographic reference data stored in 'gce_locations.mat' are used for this function, which can be edited to add or remove locations. If coordinates are successfully matched, 'Location' and 'Location_Offset' columns will be added to the structure containing the lookup results.

- Lookup Transect Distances from Coordinates: Uses geographic polygon inclusion and Thalweg line proximity analyses to look up transect distances (i.e. distance upstream from the line of demarcation) for GPS coordinates in the data structure (Latitude and Longitude columns in degrees, or UTM_Zone plus UTM_Easting and UTM_Northing columns in m). Lookups can be optionally be constrained to a specific transect to improve performance. (Note: geographic reference data for this function are stored in the data files 'mapping/thalweg_bnd.mat' and 'mapping/thalweg_ref.mat', which can be extended at any time to add or remove transects and corresponding Thalweg reference transects). If coordinates are successfully matched, columns 'Transect' and 'Transect_Distance' will be added to the structure containing the lookup results.

Q/C Flag Functions: Perform various operations based on QA/QC flag arrays in the structure, as follows:

- View/Edit Q/C Flag Definitions: Opens the list of Q/C flag codes defined in the data structure metadata into a GUI dialog for editing. The data anomalies text in the metadata is also displayed for editing.

- Recalculate Q/C Flags: Forces recalculation of Q/C flags based on the flagging criteria defined for each column (useful prior to manual Q/C flag assignment using 'Manual QA/QC')

- Copy Q/C Flags to Dependent Columns: Opens a GUI dialog for copying composite flags from one or more data columns to add to or overwrite flags in one or more dependent columns (i.e. propagates flags and adds 'manual' to the criteria to prevent automatic recalculation).

- Lock Q/C Flags (Disable Auto-update): Adds the string 'manual' to the Q/C criteria for all columns, only columns with variable types of 'data' or 'calculation', or selected columns (as specified on the submenu) to prevent automatic reanalysis and updating of flags following data structure changes

- Unlock Q/C Flags (Restore Atuo-update): Removes 'manual' strings from the Q/C criteria for all columns, columns with variable types of 'data' or 'calculation', or selected columns (as specified on the submenu) to restore automatic reanalysis and updating of flags following data structure changes

- Convert Q/C Flags to Text Columns: Converts all or selected flag tokens in the 'flags' field of the structure to individual text columns named Flag_[Column Name], as specified.

- Convert Q/C Flags to Numeric Columns: Generates coded integers representing all or selected flag tokens in the 'flags' field of each column (with 0 = no flag), adds the integer columns to the structure (named Flag_[Column Name]), and adds the list of flag codes to the metadata.

- Convert Text Columns to Q/C flags: Converts all text columns named Flag_[Column Name] to QA/QC flags for the corresponding data columns and deletes the flag columns (note: this is primarily used in conjunction with 'Convert Q/C Flags to Text Columns' to manually assign and edit value flags for multiple columns at the same time and synchronize the changes back to the data column flag arrays). Existing flags are retained or overwritten as specified, and criteria for the respective data columns are set to 'manual' to override automatic flag calculation.

- Remove Data with Q/C Flags: Converts flagged values to nulls (NaN/empty string) or deletes rows containing any flagged values, as specified.
  - Selectively Remove Flags: opens a GUI dialog to select a subset of data columns and defined Q/C flags for removal/row deletion
  - Null All Flagged Values: removes all values assigned any flag in any column
  - Delete All Rows With Any Flagged Values: deletes all rows containing any value assigned any Q/C flag in any column

- Remove Q/C Flag Assignment: Opens a GUI dialog for selectively removing Q/C flags assigned to one or more data columns, optionally locking flag criteria after updating to prevent reassignment of flags by automatic Q/C criteria evaluation. Note that data values are not affected by this operation.

- Replace Q/C Flag Assignment: Opens a GUI dialog for performing a string search/replace on Q/C flags assigned to data values in a specified column (e.g. to support a different controlled vocabularly). Flag criteria are automatically locked if any flags are revised to prevent reassignment by automatic Q/C criteria evaluation. Note that data values are not affected by this operation.

Taxonomic Functions: Perform operations based on taxonomic information in data columns.

- Lookup ITIS TSN: Matches taxonomic names in the selected column to terms in the Integrated Taxonomic Information System (ITIS) database and adds a column of taxonomic serial numbers (TSN) corresponding to each name. Note that if the column contains text or numeric codes (i.e. Variable Type = code), the code values will be resolved based on definitions in the metadata automatically prior to querying ITIS. Options include:
  - Scientific Name (Selected Column): Matches values in the selected column to scientific names in ITIS (species binomial or any other taxonomic level)
  - Common Name (Selected Column): Matches values in the selected column to common names in ITIS

Undo Changes: Undo changes since the last load/import or edit operation.

V. Metadata Menu

This menu contains commands for viewing and editing metadata (data set documentation), as well as automatically generating metadata based on evaluation of data table values (e.g. study dates from date/time columns, study sites from geographic coordinates, data anomalies from flagged and missing values, etc.). Like commands in the 'Edit' menu, all operations are performed on the active data structure (i.e. functions do not return new structures in separate editor windows).

View Processing History: Displays the entire processing history of the data structure in memory in a scrolling list box for inspection.

View/Edit Metadata: Opens a separate GUI application for viewing and editing the contents of the metadata array stored in the 'metadata' field of the structure (i.e. general metadata information organized by categories and fields and formatted using style templates to create formatted documentation).

View Formatted Metadata: Generates formatted metadata in one of the specified styles stored in 'metastyles.mat' and displays the text in a scrolling list box for inspection (note that file format fields will be blank - use the 'Export Data/Metadata | Delimited ASCII Text' to generate complete metadata as a stand-alone file or data file header).

Add/Update Study Date Metadata: Automatically updates study date metadata (Study|BeginDate and Study|EndDate) based on date/time information present in the data set.

Add/Update Site/Station Metadata: Automatically updates site descriptors in the metadata based on values in a data column containing site codes, station codes, location codes or geographic coordinates by looking up corresponding site information stored in the file 'gce_coords_all.mat' or 'gce_locations.mat' (cached versions of geographic information tables from the GCE Metabase).

Add/Update Q/C Flag Definitions, Data Anomalies: Opens the list of Q/C flag codes and data anomalies text defined in the data structure metadata into a GUI dialog for editing.

Document Flagged Values as Anomalies: Generates plain text summaries of all flagged values for each parameter, optionally grouped by date using the specified date and separator format.

Document Flagged and Missing Values as Anomalies: Generates plain text summaries of all flagged and missing values for each parameter, optionally grouped by date using the specified date and separator format.

Automatically Assign Numerical Descriptors: Automatically assigns data types, numerical types, and precisions for numerical columns based on analysis of magnitude and significant digits.

Generate Code Definition Dataset: Generates a data set summarizing codes and code definitions for all or selected coded columns in the current data structure. Columns that are not designated as Variable Type = "code" will be ignored. Metadata fields to copy to the summary data set can be determined automatically or specified manually using a list dialog.

VI. Tools Menu

This menu contains links to external GUI applications and dialogs for performing specialized operations on data structures. Commands in the 'Tools' menu generate derived structures or other analysis products (e.g. plot figures) and do not alter the active structure in memory.

Plotting: Create various plots of data in the current structure.

- 2D Line/Symbol (multiple Y): Create multiple-Y vs. X symbol/line plots, optionally restricting values by values in one specified column (inline query).

- 2D Line/Symbol (single Y, split by groups): Create a Y vs. X symbol/line plot of after grouping rows by values in a specified column. One line segment is produced per group after optionally restricting values by values in another specified column (inline query).

- Map Plot: Plot data values on a map as symbols, text labels, or color-mapped patches (requires georeference columns with variable type 'coord')

- Vertical Profile Contour Plot: Generates a 3-D color contour plot of a selected variable versus distance and depth (requires multiple depth profiles collected multiple distances along a transect)

Filtering: Filter records in the current data structure to create a subset

- Filter/Subset Data by Column Values (Query Builder): Opens a GUI application for building a custom query string to select a subset of data rows meeting the query criteria. The resulting subset structure is opened in a new editor window.

Statistics: Perform various statistical analyses on structure data

- View Column Statistics: Displays automatic column statistics in a scrolling textbox.

- Column Statistics Report: Opens a dialog for generating a custom statistical report in various delimited ASCII text formats.

- Statistics for Grouped Data: Opens a GUI dialog for generating a summary data set by grouping/aggregating records based on values in one or more specified grouping columns and then calculating relevant descriptive statistics for records within each aggregate for all specified data columns. The resulting derived data set contains the grouping columns and multiple statistical result columns for each specified data column, and is opened in a separate editor window to permit additional customization and analysis.

- Statistics for Binned Data: Opens a GUI dialog for generating a summary data set by assigning records to bins based on values in a bin data column (e.g. Depth) using the specified bin range and interval. Records can also be grouped by values in one or more additional columns, then relevant descriptive statistics are calculated for records within each bin for all specified data columns. The resulting derived data set contains bin columns (bin top, bin middle and bin mean), grouping columns (if specified), and multiple statistical result columns for each specified data column, and is opened in a separate editor window to permit additional customization and analysis.

- Statistics for Date/Time Intervals: Opens a GUI dialog for generating a summary data set for a specifed date/time interval (yearly, monthly, daily, hourly), optionally grouping by values in one or more non-date/time columns. Relevant descriptive statistics are calculated for records within each

date/time interval for all specified data columns. The resulting derived data set contains date/time columns, grouping columns (if relevant) and multiple statistical result columns for each specified data column, and is opened in a separate editor window to permit additional customization and analysis.

Transformation: Transform the current data structure to create a derived data set

- Split Compound Data Series: Opens a dialog for splitting a compound data series based on values in a string or integer column (e.g. parameter or site codes) and joining each individual series to create a table with separate data columns for each series group (e.g. split a merged data set containing hydrodraphic data from multiple sites to form a table with individual salinity and temperature columns for each site, and records aligned by date and time)

- Normalize Multiple Related Columns: Opens a dialog for normalizing a data set by merging multiple related columns to form composite parameter name and parameter value columns with records in other specified columns repeated for each original group of parameters (e.g. normalizes a data set containing a repeating group of data columns where each column represents observations for a specific date, location, or species so that the data can be summarized and filtered by data, location or species, resp.)

- Statistical Data Reduction: Generate reduced data sets using statistical analysis and aggregation alogithms

- Group/Aggregate Data: see Statistics > Statistics for Grouped Data (above)

- Bin Data: see Statistics > Statistics for Binned Data (above)

- Date/Time Resampling: see Statistics > Statistics for Date/Time Intervals (above)

Specialized: Perform specialized operations on specific types of data structures.

- Top/Bottom Values (Vertical Profiles): Opens a GUI application for creating a data structure containing only top and bottom records from a vertical profile based on values in a designated depth or pressure column, after optionally grouping records by values in one or more data columns.

- Combine CTD Surveys with Overlapping Times (<=4km apart): Combines CTD surveys within cruises to make multi-leg or multi-vessel surveys contiguous, based on overlapping ending date/time of the earlier leg and starting date/time of the latter leg, and minimum terminal distances of <= 4km (i.e. allowing for reversed orientations). Requires a CTD data set with data columns 'Cruise' and 'Survey', plus appropriate temporal and geospatial data columns.

- Create CTD Station Dataset for Mapping: Generates a data set containing geographic coordinates and station codes at a specified distance interval from CTD station location information stored in 'ctd_stations.mat'

Data Harvesting: Manage automated (timed) data harvesters

- Start Harvesters: start all active harvesters defined in 'harvest_timers.mat' database

- Stop Harvesters: stop all or specified harvesters currently running in MATLAB

- List Harvesters: list all harvesters currently running in MATLAB

- Edit Harvesters: open 'harvest_timers.mat' in the Data Editor grid to revise or define harvesters

Data Merge Tool: Opens a dialog for selecting multiple data structures to load and merge into a single structure (both data and metadata are combined).

Data Search Engine: Opens or switches focus to the GCE Data Search Engine dialog

Mapping Toolbox: Opens a MATLAB map figure created using the GCE Mapping Toolbox for customization, data mapping, or geographic analysis

VII. Misc Menu

This menu contains various commands for managing the lists of import filters, metadata styles and metadata templates displayed on Data Editor menus.

Add/Edit Import Filters: Opens the list of data import filters stored in the file 'imp_filters.mat' into a data grid, allowing filters displayed in 'File | Import Data' to be added or edited (changes will be applied to all open editor windows).

Add/Edit Metadata Templates: Opens a GUI dialog (ui_template) for adding and editing named metadata templates stored in 'imp_templates.mat' and used by various toolbox functions to apply boilerplate metadata based on matching column names and units to template entries (changes will be applied to all open editor windows).

Add/Edit Metadata Styles: Opens a GUI dialog (ui_metastyle) for adding and editing named metadata style definititions stored in 'metastyles.mat', which are used to generated formatted metadata in various user-specified styles.

Update Registration: Opens a GUI dialog (ui_gce_register) for adding or updating user name, affiliation and contact information for retrieve data from the GCE-LTER data catalog via the Search Engine

Update Geographic Databases: Opens the Point Locations database ('geo_locations.mat') or Site Polygons database ('geo_polygons.mat') for editing in the Data Editor application

VIII. Window Menu

This menu contains commands for managing data editor window titles and positions and switching between active editor windows when working simultaneously with multiple data sets.

New Window: Opens a new (empty) editor window for loading and working with another structure.

Arrange: Re-arranges all active (non-minimized) editor windows on the main console. - Cascade: moves the active window to the top left of the screen and cascades all other open windows (i.e. offsets their positions to the right and down) - Tile: moves the active window to the top left of the screen and tiles as many other open editor windows across the screen as will fit. Any remaining windows will be stacked under the bottom right-most window position - Stack: stacks all active editor windows at the top left of the screen

Choose: Opens a GUI dialog for choosing an active editor window to display, then sets the focus to that window

Rename Editor Window: Allows the Data Structure Editor window to be assigned a descriptive name to aid operations involving multiple editor windows. - Use Structure Title: rename window based on the title of the active data structure - Custom Name: opens a GUI dialog for entering or editing the window title - Reset Window Name: resets the window name to the default "Data Structure Editor"

IV. Help Menu

This menu contains links to Data Editor documentation and information on the entire toolbox package.

View Documentation: Displays data structure editor help text in the documentation viewer.

About the GCE Data Toolbox: Opens the GCE Data Toolbox startup screen and task launcher.

GUI Applications - Join Data Reference

I. Overview

The "Join Data" dialog allows two data sets to be integrated into a single composite data set containing columns derived from both parent data sets, for example:

> Data Set A columns: Station, Year, Month, Day, Hour, Minute, Temp_Air, Precipitation
>
> Data Set B columns: Site, Yr, Mo, Dy, Hr, Mn, Temp_Water, Salinity, Pressure, Depth
>
> Join Parameters:
> Key columns (A=B): Year=Yr, Month=Mo, Day=Dy, Hour=hr, Minute=Mn
> Selected output columns from A: Temp_Air
> Selected output columns from B: Temp_Water, Salinity
>
> Joined Data Set columns: Year, Month, Day, Hour, Minute, Temp_Air, Temp_Water, Salinity

Note that in order to be joined the two parent data sets must contain compatible "key" columns so that rows from each data set can be matched and aligned appropriately. Key columns can differ in name, as in the example, but data types, variable types (i.e. semantic types) and units must match for all paired columns. Also the combination of key column values must be unique for each data set in order to perform the join. These conditions are more stringent than join operations performed by most relational databases using Structured Query Language in order to prevent joins based on matching columns of different semantic types and to avoid inappropriate duplication of data values that may affect statistical analyses (e.g. if one data set includes duplicate values and the other does not).

In most cases successfully joining two data structures requires knowledge about the data column characteristics and table layouts, so reviewing the data set metadata before proceeding is strongly recommended.

II. Instructions

1) Define Key Columns and Options for Joining Structures

- Choose the first pair of key columns by selecting one column from Structure A and one from Structure B in the respective column lists, and clicking the "Add >" button to add the join condition to the list

- Repeat this process for addition key column pairs, proceeding in order from the most general/inclusive columns to the most specific/exclusive (e.g. Year -> Month -> Day rather than Day -> Month -> Year)

- Unnecessary columns in the join list (i.e. those added after a unique combination of key columns is achieved) will not be used in the join, but will be included as output columns in the joined data set

- Choose the "Rows to Return" option for each structure, specifying "Only matching" to include only rows with key column values matching the other structure or "All rows" to include all rows regardless of key matches (i.e. if "Only matching" is specified for both A and B, an "Inner" join is performed that omits any rows that cannot be matched between structures, whereas if "All rows" is specified an "Outer" join is performed and all rows in both structures will be returned, with NaN/blank values used to pad rows in columns that cannot be matched to the other data set)

- After all key columns are specified, press the "Choose Output Columns" button to generate or refresh the output column lists in section 2

- If desired, check the "Remove duplicated records in key columns" option to force the join to succeed even if duplicate data values are present within combinations of key columns (i.e. only the first instance of each unique key column combination is returned). Note that this option can produced biased results, so other data reduction techniques, such as aggregation, binning, or date/time resampling, should be performed instead whenever possible to make the two data sets comparable before joining them. For example, if you want to join a time-series data set with a daily time step to a data set with an hourly time step, use the 'View/Edit' button to open the hourly data set in an editor window, then use the 'Tools > Transformation > Statistical Data Reduction > Date/Time Resampling' command to resample the data set and generate daily summary values for each selected parameter. After resampling, use the 'File > Return Data' command from the editor window to return to the join dialog and re-choose join columns and output columns for the updated data set.

2) Choose Output Columns for the Joined Structure

- Add desired columns to the "Selected" lists by individually selecting columns from the corresponding "Available Columns" list and pressing the "Add >" button or double clicking on the column name

- Remove columns from the "Selected" lists by selecting the column name and pressing the "< Del" button to return them to the corresponding "Available Columns" list

- Note that at least one output column must be selected from each data structure in order to perform a join

- Optionally choose a column prefix to prepend to the name of each output column, distinguishing which data set it was derived from or to prevent column name conflicts (e.g. if both data sets include a column "Salinity", specifying "Site1_" and "Site2_" prefixes would produce output columns "Site1_Salinity" and "Site2_Salinity"

3) Perform the Join

- After choosing the key columns, specifying join options, and choosing output columns, press the "Proceed" button to create the composite data set

- If the join is successful, the new data set will be opened in a new data editor window, otherwise a specific error message will be displayed

- If the "Close dialog after performing the join" option is checked the dialog will be closed automatically after a successful join

GUI Applications - GCE Data Search Engine

I. Introduction

The GCE Data Search Engine is a GUI application for performing metadata-based searches to find GCE Data Structures that meet user-specified thematic, temporal, and geospatial criteria. In order to support searching, data structures are first analyzed using a combination of metadata and data mining techniques to generate an optimized search index. Data structures stored in MATLAB files in any number of local directories can be indexed, and indices can be saved and re-used to speed subsequent searches. Pre-generated indices of public data sets in the GCE Data Catalog and GCE Data Portal can also be downloaded and merged with local indices to support simultaneous searches of local and web-based data holdings.

After each search, data sets matching the specified criteria are added to a cumulative match list, allowing queries to be combined to produce composite result sets. Data sets in the matched list can be be viewed, edited and analyzed using a variety of GCE Data Toolbox programs. Multiple data sets can also be batch copied or exported in various ASCII text or MATLAB formats with user-selectable metadata formats and QA/QC flag options. When data sets in web directories are selected for any operation, the corresponding data files are automically downloaded from the GCE web site and cached for future use.

II. Starting the Program

The search engine program can be started several different ways:

1) using the "Search Engine" button on the GCE Data Toolbox startup screen
2) using the "GCE Data Search Engine" option on the Data Editor Tools menu
3) typing "ui_search_data" on the MATLAB command prompt

The search engine window is divided into three panes; the top pane contains the list of indexed paths, search fields and controls, the middle pane contains the query history, and the lower pane the cumulative list of matched data sets. The query history window can optionally be hidden to enlarge the lower pane if desired using the "Hide Query History" command on the Options menu (default on computers with low resolution screens).

III. Building and Managing Search Indices (top pane)

In order to search for data sets, a search index must first be loaded or created. Users running MATLAB R13 (6.5) or higher can download a pre-generated index of all public GCE data sets using the "GCE Web Index File (WWW)" command on either the "File/Load Search Index" or "File/Merge Search Index" menus. GCE Data Structures stored in MATLAB files in local directories can also be indexed at any time using the "Add" button to the right of the "Indexed Directories" list --the newly created index will automatically be merged with any existing index, updating any duplicate entries. If the "Subdirectories" option is checked, all MATLAB files in child directories will also be indexed at the same time.

Once a search index is loaded or created, all indexed paths will be listed along with the number of data structures each contains. Paths and all corresponding data sets can also be selectively removed from the index at any time by pressing the "Remove" button. The index can also be refreshed to remove missing files, re-index modified files, and index newly added files in each directory by pressing the "Refresh" button. If GCE web entries are included in the index, an updated web index will also be downloaded and incorporated at the same time.

IV. Performing Searches (top pane)

After an index is loaded or created, the index can be queried by filling in any number of criteria in the search fields, selecting appropriate search options, and pressing the "SEARCH" button. The label "(multiple)" is displayed alongside text fields and lists that can accept multiple entries (see individual field descriptions for details), otherwise only individual entries are allowed.

Most text searches (e.g. general metadata fields, author, keywords, species names) are based on partial string matches, so wildcard characters (*) are implicit and should not be included or they will be regarded as literal text. Also, note that negative criteria can be specified in any text field (i.e. "-" preceding the text, e.g. "-ctd") to filter out data sets matching that text criteria. For multiple entry fields, positive and negative criteria can be mixed in any order (e.g. keywords = "-ctd, -sonde, salinity")

Each Search Criteria field is described below:

1. General Metadata Fields: Text in various general metadata fields, such as Title, Abstract, and Methods, can be searched by selecting a metadata field using the drop-down menu and entering search text in the adjacent edit box. The option "Any Text Contains" can also be specified to search for the text in any of the listed metadata fields. The text comparison type is listed after each metadata field, e.g. "Contains" (partial string match), "Begin With" (beginning string match) or "Is" (exact string match). For partial string matches, either single words or complete phrases can be entered (e.g. "crab abundance" or "abund"). Note that spaces or commas in the search string will be considered part of the search phrase, so including multiple terms in metadata search fields will not generally produce useful results.

2. Date Fields: Data set study periods (i.e. temporal coverages) can be searched by selecting the "Date Range" option and filling in one or both adjacent edit boxes to specify minimum starting date and maximum ending date criteria, resp. Alternatively, the "Contains Date" option can be specified to search for data sets with study periods that include a specific date. Any valid date/time format recognized by MATLAB can be used (see documentation for "datenum" and "datestr"); entries will automatically be converted to DD-MMM-YYYY or DD-MMM-YYYY HH:MM:SS format as appropriate (e.g. 25-Oct-2004 10:30:00).

3. Author Name: Principal data set authors can be searched in the same manner as general metadata fields. Partial or complete author names can be entered (e.g. "Wade M. Sheldon", "Sheldon" or "shel").

4. Keywords: Data set key words can be searched by entering one or more partial of complete words in the Keywords field, separating multiple words with commas. Unlike general metadata

and author searches, keyword searches are based on beginning string matches, so searches on "plant" will match "plant", "plants", and "plant biomass", but not "marsh plants". This distinction is made to avoid unexpected matches when keywords contain search terms, such as matching "Spartina" on searches for "par". Leading or trailing spaces or spaces after commas will be ignored.

5. Species Names: Taxonomic names defined in metadata for any coded species columns (e.g. "Plant_Species" or "Species", typically species name binomials or genus spp.) can be searched in the same manner as general metadata (e.g. "alterniflora" will match "Spartina alterniflora"). Multiple comma-delimited search terms can be specified. Note that only data sets containing coded species name columns typically contain taxonomic index entries, so key word searches on species names may also be useful.

6. Data Columns: Data set columns can be searched by selecting one or more entries in the corresponding Data Columns list. Multiple listbox selection techniques vary by computer platform, but dragging with the mouse will generally select contiguous list rows and holding down the "Ctrl" or "Option" key while clicking will select non-contiguous rows. The total number of columns selected is displayed on the left side of the list for reference. (Note that the column list is dynamically generated from data set entries in the current search index, so removing indexed paths will clear selections for any columns no longer present when the list is regenerated).

7. Study Site: The data set geographic coverage can be queried by specifying a nominal GCE study site from the "Study Site" drop-down menu. When data sets are indexed, detailed geospatial analyses are performed to augment site references in the data set metadata, consequently study site indices are very comprehensive. For example, index entries for hydrographic surveys will include all relevant GCE transect codes plus codes for all marsh sites intersected by the surveys (i.e. based on geographic coordinates within GCE site polygons). Site criteria can be used to modify bounding box criteria (below), or vice versa.

8. Bounding Box: Geographic data set coverage can also be queried by entering one or more bounding latitudes and longitudes or complete bounding box limits, or by dragging a bounding box on a map (opened by pressing the "Map" button). During indexing, data set geographic coverage is determined based on limits of all longitude and latitude values present in the data set (or reprojected from WGS84 UTM coordinates), or by pooling bounding box limits of indexed study sites if no georeference columns are available. Data set bounding boxes therefore vary in accuracy, and should be used with caution if geospatial coverage criteria is critical (e.g. specify "Longitude", "Latitude", "UTM_Easting", etc. as data column criteria to limit results to georeferenced data sets). If one or more bounding box values are omitted, NaN (not a number) values are substituted for missing bounds in the query statement, and no restriction is placed on the corresponding bounds (i.e. equivalent to +/-180 longitude or +/-90 latitude).

9. Bounding Box modifiers: the following options determine the type of bounding box search performed when any bounds are specified:

a. "Datasets enclosed by bounds" searches for data sets with overall bounding longitudes and latitudes entirely enclosed within the specified limits (i.e. traditional bounding box search)

b. "Datasets overlapping bounds" searches for data sets that reference study sites within or overlapping the specified boundaries; site lookups are used instead of overall data set bounding box searches to limit the potential for spurious matches to large survey data sets whenever bounding boxes within the GCE domain, but excluding all GCE sampling areas, are specified. Regardless of the match type option setting (see below), data sets referencing any enclosed or overlapping site will be matched.

10. Search Options: the following options can be specified to tailor the search:

    a. "Match all criteria" and "Match any criteria" specify the match type of the overall query. If "Match all" is selected, only data sets matching all criteria will be returned, otherwise data sets matching any one criteria will be returned (if only one criteria is specified there is no functional difference between these options)

    b. "Case sensitive text searches" specifies whether all text searches should consider case (checked) or not (unchecked). Text search fields affected by this setting include general metadata, author, keyword, species, data column, study site.

    c. "Save new queries to history list" specifies whether new (successful) queries are saved to the query history list or not (disabled if the query pane is hidden)

11. Clear Bounds: clears all bounding box entries

12. Clear All: clears all search criteria selections (resets the form)

13. SEARCH: searches all data sets in the current index using the specified search criteria and options, and returns matched data sets to the list of "Cumulative Search Results", omitting any matches that duplicate existing matches (an informational box will be displayed if no data sets are matched or all matches are duplicates).

V. Working with Stored Queries (middle pane)

If the Query History pane is visible and the "Save new queries to history list" option is checked, each successful query statement is added to the query history list unless it duplicates a prior saved query. Saved queries can be parsed to fill in Search Criteria fields at any time using the "Load Query" button, and all or individual saved queries can be deleted using the "Clear All" or "Clear Query" buttons, resp.

This query history feature allows users to build up lists of standard queries, and then re-load search criteria for editing or re-run past queries on new or updated search indices to identify additional data sets of interest.

VI. Working with Search Results (bottom pane)

After every successful query, new data sets matching the specified criteria are added to the Cumulative Search Results list. All information necessary to retrieve the corresponding data set is stored along with each entry, so search results are completely independent from search indices. In fact, result sets can be generated over multiple sessions using any number of different index files.

The following information is displayed for each data set in the list: [location]/ [accession] - [title] (period: [start date] - [end date]), where: [location] = file location (local or web) [accession] = data set accession, if defined (otherwise [no accession] or portal) [title] = title of the data set [start date] = data study start date (mm/dd/yyyy) [end date] = date study end date (mm/dd/yyyy)

Double clicking on any entry with the mouse loads the corresponding data set and displays its metadata information in the currently selected style (set using the Options/Metadata Style menu). If the data set resides on the GCE web server, the user registration information entered on initial program startup (or updated using the "Tools/Update GCE Registration") is used to retrieve the corresponding file from the GCE server. Web-based files are cached locally in the "search_webcache" subdirectory of the GCE Data Toolbox the first time they are retrieved, then the cached copy is used for all subsequent analyses to minimize network file access.

The following buttons to the right of the listbox can be used to manage the list items and tailor the result set, or to perform other operations with the selected data sets:

"Sort" sorts the data sets by location, accession, and title

"Select All" selects all data sets for batch operations

"Select None" clears all selections

"Remove" deletes all selected data sets from the match list

"Remove All" deletes all data sets from the match list

"View/Edit" opens the specified data set(s) using the GCE Data Editor application for customization or analysis

"Plot X/YY" opens a dialog for generating line/scatter plots for the selected data set

"Plot Groups" opens a dialog for generating line/scatter plots split by a categorical variable

"Map Plot" opens a dialog for generating map plots of geo-referenced data

"Summary" displays a statistical summary of the selected data set

Note that button states are toggled on or off in accordance with the list selections, so buttons for tasks that do not support multiple data sets are automatically disabled when more than one data set are selected.

In addition to the task buttons, varous operations can also be performed on selected data sets using commands under the "File" menu, as follows:

"Copy Data Sets" copies all selected data sets to a specified local directory (in data structure format)

"Export Data Sets" exports data sets and metadata to a specified local directory in various Text and MATLAB file formats (see section VII for a discussion of metadata styles, text file layouts and QA/QC flag handling).

"Join Data Sets" allows two data sets to be "joined", based on matching common values in one or more "key" columns, to create a composite data set containing columns from both structures (this command opens a GUI dialog for choosing key columns and output columns for the join). Keys can either be manually defined ("Manual Key Selection"), or determined automatically to join structures based on supported date/time columns ("Automatic Date/Time Join") to simplify integration two time-series data sets.

"Merge Data Sets" allows two or more data sets to be "merged" (i.e. concatenated) to create a single data set. Note that columns will automatically be matched based on column name and column attributes (i.e. data type, units), and columns that cannot be matched will be "staggered" in the composite data set and padded with missing values for proper alignment

VII. Customizing Program Options

The "Options" menu includes several user-selectable options that affect all data set operations, as specified below:

1. Metadata Style: This menu allows the style used to display and export metadata to be specified, as follows:

    a. "GCE Standard" style (default) lists all data set descriptors except for project, and includes a compact table of attribute descriptors

    b. "LTER FLED" style lists complete metadata in a verbose numbered outline style

    c. "Abbreviated" style omits all metadata except for title, author and data descriptors

    d. "None" omits metadata from exported data sets, however "GCE Standard" will be used to view metadata interactively

2. Q/C Flag Options: This menu allows preferences to be set for handling QA/QC-flagged values present in data sets when the data sets are exported in standard MATLAB or text formats. Tables of flags are stored along with the data table in GCE Data Structures to support flag editing and reanalysis, but this arrangement is not supported in these other file types. Q/C flag options include:

a. "Retain all flagged values" retains all flagged values but instantiates flags as data columns alongside any column with any flag codes assigned. Flag columns are coded text (text formats) or integer columns (MATLAB formats) named according to the corresponding data column (e.g. "Flag_Salinity" for flags on column "Salinity"), and flag descriptions are automatically documented in the data set metadata prior to export.

b. "Remove all flagged values" converts all flagged values to missing (" or NaN or text or numeric data columns, resp.) prior to export, and documents the conversion in the metadata.

c. "Removes values flagged 'I' (invalid)" converts all values flagged as 'I' (invalid) to missing, and instantiates any other value flags as data columns as described above.

d. "Delete rows with flagged values" deletes all data records containing any flagged values prior to exporting the data, documenting the deletions in the metadata

e. "Delete rows with values flagged 'I' (invalid)" deletes data records containing any values flagged 'I' (invalid), documenting the deletions in the metadata, and instantiates any other value flags as data columns as described above.

3. Text File Header: Specifies the header style and metadata format to use when exporting data sets in ASCII text format, as follows:

   a. "Metadata Header" prepends metadata in the selected style (see section VII.A) to the data table, and also includes column names, column units, and column variable types above each data column

   b. "Brief Header (separate metadata)" only includes column names, column units, and column variable types above each data column as a file header (metadata will be saved as a separate text file named according to the data file with an extension "-meta.txt").

   c. "Column Titles (separate metadata)" only includes column names above each data column as a file header (metadata will be saved as a separate text file named according to the data file with an extension "-meta.txt").

   d. "None (separate metadata)" saves the data table with no header (metadata will be saved as a separate text file named according to the data file with an extension "-meta.txt").

4. Metadata Merge Option: Specifies the option for merging metadata when multiple data sets are merged using one of the "File/Merge Data Sets" commands, as follows:

   a. "Merge All" merges all metadata fields, concatenating any non-duplicated field contents to form composite metadata

   b. "Merge None" does not merge any metadata (only metadata content from the first structure is retained)

   c. "Merge Selected Sections" only merges metadata fields specified using a pick list of all available fields

5. Flag Merge Option: Specifies the handling of QA/QC flag criteria and flags when merging multiple data sets

   a. "Lock Flags" locks all flags prior to merging data sets by adding the flag criterion "manual" to each field to prevent automatic recalculation

   b. "Do Not Lock Flags" retains all original flag criteria, causing flags to be regenerated for columns in the merged structure (except where already set to "manual")

6. Dataset Name Option: Specifies the option for automatically creating a data set name column when merging multiple data sets using any "File/Merge Data Sets" command.

   a. "No Dataset Name Column on Merge" does not add a data set name column prior to merging multiple data sets

   b. "Add Dataset Name Column on Merge" adds a column "DataSetName" to each structure prior to merging, so that the name of the corresponding data set (i.e. filename without extension) is listed for each record in the combined data set

   c. Auto-Save Workspace Option: Specifies whether to automatically save the current workspace (i.e. all results and settings) each time the search engine dialog is closed

   d. "Yes" saves all results and settings as 'search_default.mat' in the 'search_indices' sub-directory of the toolbox

   e. "No" does not automatically save the workspace on exit (the user will be prompted to save any results manually)

7. Auto-Add Integrated Datasets: Specifies whether data sets from 'File|Join Data Sets' and 'File|Merge Data Sets' operations are automatically added to the cumulative search results list for data file management or serial data integrations. Note that data sets can also be manually added to list at any time from the data set editor window using the 'File|Export Data/Metadata|Copy Structure to Search Engine' or 'File|Export Data/Metadata|Move Structure to Search Engine' commands.

   a. "Yes" specifies that integrated data sets are automatically added after creation (note: this option may result in a short delay before editor window commands work as MATLAB saves the file and updates the search engine match list)

   b. "No" specifies that data sets are not added

8. Auto-Delete Temp File Option: Specifies whether to automatically delete temporary files when they are removed from the search results list

    a. "Yes" specifies that temporary files will be deleted on removal

    b. "No" specifies that temporary files will be retained until manually deleted using the "File/Clear Temporary Files" menu option

9. Hide/Show Query History: Toggles display of the query history pane

VIII. Saving and Restoring Workspaces

All program options, the entire search index, search field selections, query history and cumulative search results can be saved as a workspace file, and reloaded at any time using the "File/Save Workspace" and "File/Load Workspace" menu commands. This allows the user to create and manage custom search environments for specific tasks.

Appendix I -- Data Structure Specification

GCE Data Structure Specification (version 1.1, May 2001)

FieldContent	Type	Description
version	n x 1 cell array of strings	GCE Data Structure version (used for validation)
title	character array	Title of the overall data set
metadata	n x 3 cell array of strings	Parseable array of general metadata information (variable-length array of metadata category names, field names, field values)
datafile	n x 2 cell array	Names and sizes of data files added to the structure
createdate	character array	Date and time the data structure was created
editdate	character array	Date and time the structure was last edited
history	n x 2 cell array of strings	Processing history of the data structure (list of dates and operations performed)
name	1 x m cell array of strings	Name of each data column
description	1 x m cell array of strings	Description of each data column
units	1 x m cell array of strings	Units for values in each data column
datatype	1 x m cell array of strings	Physical data type of each column (i.e. storage type, e.g. 'f' for floating-point, 'e' for e by ponential, 'd' for decimal/integer, 's' for string)
variabletype	1 x m cell array of strings	Variable type of each column (i.e. basic semantic type), i.e. 'data' = measured data value 'calculation' = calculated data value 'nominal' = categorical value (e.g. name, species, site) 'logical' = Boolean or true/false value (e.g. 0 or 1, yes or no) 'datetime' = date and/or time value 'ordinal' = order or positional value 'code' = coded value 'coord' = geographic coordinate value (e.g. latitude or longitude) 'text' = free text (e.g. notes, comments)
numbertype	1 x m cell array of strings	Numerical type of each data column (e.g. 'continuous', 'discrete', 'angular', 'none')
precision	1 x m array of integers	Decimal places to display for each column
values	1 x m cell array	Data values (each cell contains a matching "column" of data, as an n x 1 numerical array or n x 1 cell array of strings)
criteria	1 x m cell array of strings	Flag criteria e by pressions for each column, which are evaluated by the 'dataflag' function to generate QA/QC flags
flags	1 x m cell array	Arrays of flag characters (each cell is empty or contains an n x m character array of flags matching the corresponding value array)