GMS DESIGN DOC

TSSDS Interface

Last Modified: 3/31/00
Other programs affected: SMS, WMS?
Document: I:\docs\gms\design\other\TSSDS\tssds.doc
Resource File: I:\docs\gms\design\other\TSSDS\script1.rc

Introduction

We have been contracted by WES to design a set of tools for GMS that will allow users to import data from a database that has been built to TSSDS standards. TSSDS stands for Tri-Service Spatial Data Standards. It is an organization that controls the standards that military organizations should follow when designing databases. Refer to the "Statement of Work" document for more information.

The interface will be built in stages. The first stage will focus on building a strong foundation that will be able to support all the functionality that is ultimately desired. It will include a general purpose query builder and the ability to import borehole, scatter point, and coverage point data. The design should be flexible enough for multiple GMS data types.

TSSDS Standards

The TSSDS standards can be installed from a CD and viewed on a computer. Here are a few helpful notes on using the standards. Table/attribute summaries can be printed from the Print menu. To see what table a foreign key comes from, the "Join Relations" feature can be used while browsing by structure. The Parent/Child relationships for a table can be viewed from the "Joins" tab in the "Tables" dialog while browsing by structure. "I" means integer, "S" means single precision, "D" means double precision, "C" means character, "R" means real.

Figure 1 is an example of the parent/child relationships between several tables. The parents are above the children. The first line in the box is the table name. Below that is the long table name, and below that is a list of some of the pertinent information stored in the table. As Figure 1 indicates, data in the database are divided into multiple tables. It will usually be necessary to look at several tables in order to import the data into GMS. The figure also indicates that it is not always obvious how the data in the database correspond to the data we need for GMS. Our design needs to be flexible enough to handle these conditions.

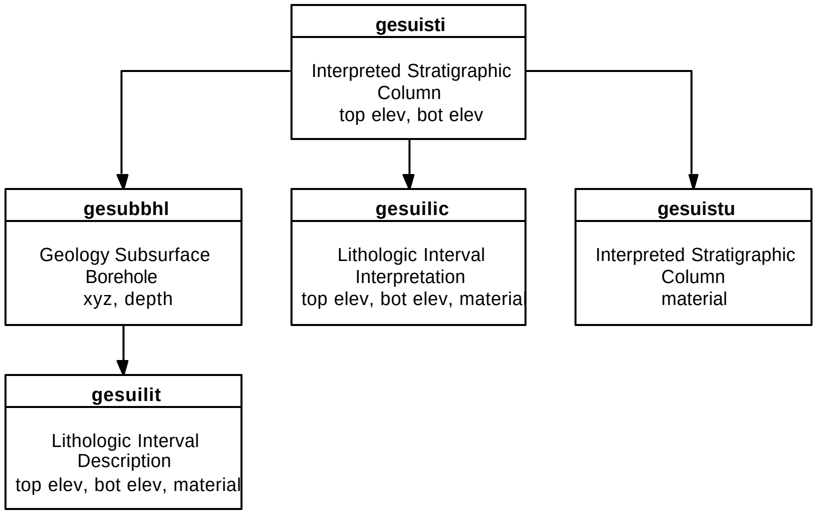

Figure 1. Parent/child relationships between different tables in a TSSDS database.

Import Wizard

A wizard will be used to import data from a TSSDS database. The wizard has 5 steps, which are outlined below.

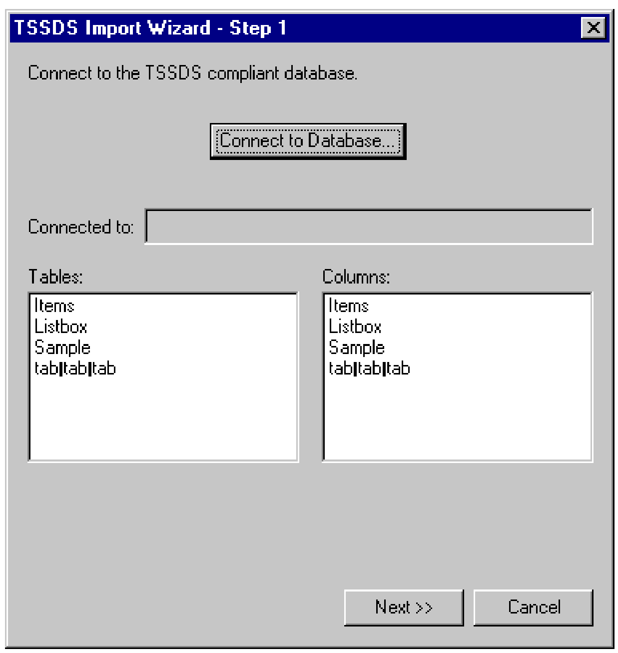

Figure 2. Import Wizard Step 1.

The first step in the wizard allows the user to establish an ODBC connection to a database. The "Connect to Database" button brings up the standard ODBC dialog which lets the user select their database. Once connected, the "Connected to" field will show the name or the path and filename to the database. The tables found in the database will be shown on the left and the columns for the highlighted table will be shown on the right.

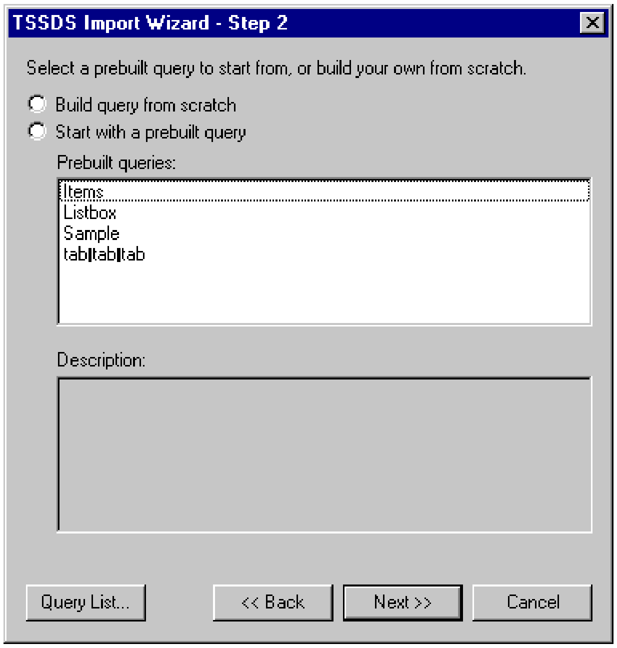

Figure 3. Import Wizard Step 2

Step 2 allows the user to select a pre-built query or design their own from scratch. If they elect to use a pre-built query, the available queries will be displayed in the middle part of the dialog. A description of the highlighted pre-built query will be displayed in the bottom portion of the dialog. The "Query List" button will be described later.

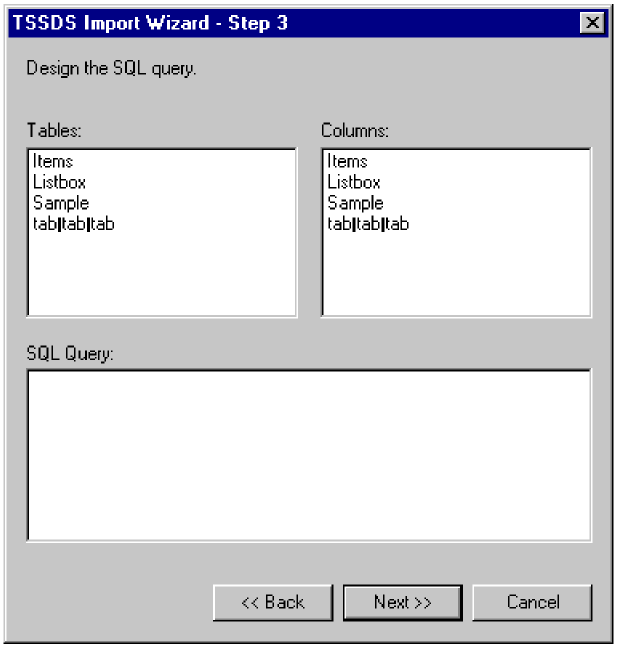

Figure 4. Import Wizard Step 3

Step 3 shows the SQL query in the bottom of the dialog and lets the user modify it as they wish. If the user has elected to build a query from scratch, the query window will be empty. Otherwise the query window will be initialized with the query selected in Step 2. To aid the user, the available tables are listed on the left and the columns of the highlighted table are listed on the right, as in Step 1.

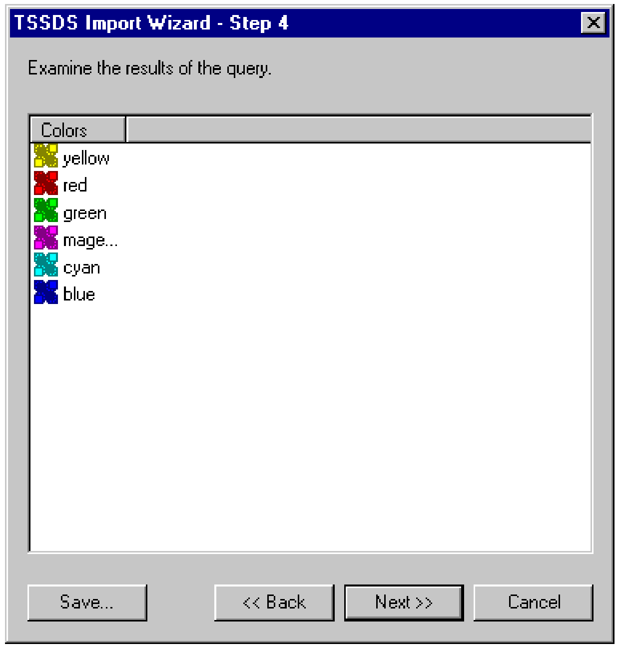

Figure 5. Import Wizard Step 4

Step 4 displays the results of the query from Step 3. This will be displayed as a table with rows and columns. If the user has modified a pre-built query or designed their own from scratch and would like to add it to the list of pre-built queries, they can select the Save button. This will bring up a dialog (not shown) allowing them to give a name and description to the SQL query. The query will thereafter show up in Step 2. The “Query List” button in Step 2 can be used to bring up a dialog (not shown) allowing the user to delete pre-built queries from the list, or reset the list of pre-built queries to the defaults.

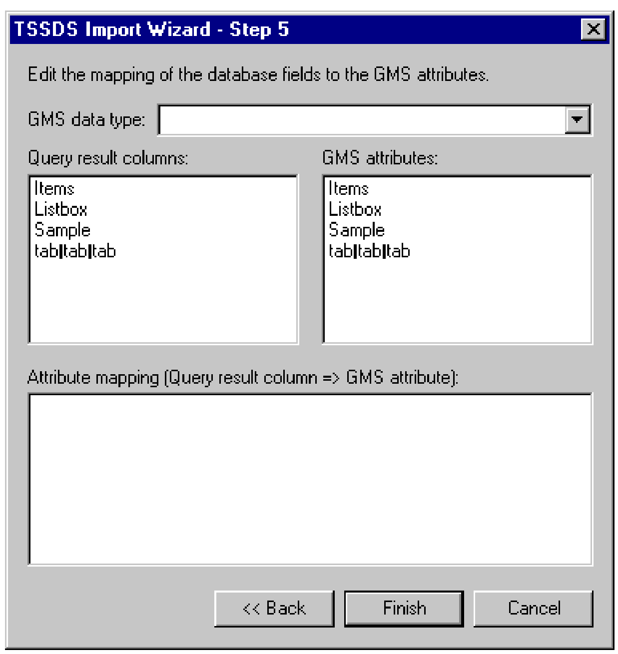

Figure 6. Import Wizard Step 5

Step 5 allows the user to map the fields (columns) in the query result to a GMS attribute. At the top of the dialog a pull-down list displays the GMS data types that are available to import the data to. The following table shows what may be in the list:

Observation Coverage Points
MODFLOW Coverage Points
2D Scatter Points
3D Scatter Points
Boreholes
…

This list may grow in later iterations.

Prebuilt Queries

Observation wells – water level – imwliglv

The following is an example of a possible pre-built query. The information came from examining the TSSDS database design. The name of the query would be something like "Observation wells – water level – imwliglv." The description would be something like "Imports water level data stored in the "imwliglv" table (Long Name: FM – Improvement Wells Groundwater Level Data)."

The SQL query is as follows:

SELECT imwlimas.mnwell_id, gesubbhl.coord_x, gesubbhl.coord_y, gesubbhl.coord_z, imwliglv.wl_date, imwliglv.wl_time, ([imwliglv].[wl_melev] – [imwliglv].[wl_static]) AS Elevation, imwliglv.wlelev_u_d

FROM (gesubbhl INNER JOIN imwlimas ON gesubbhl.subbhl_id = imwlimas.subbhl_id) INNER JOIN imwliglv ON imwlimas.mnwell_id = imwliglv.mnwell_id;

The SQL query is built from the following three tables in the TSSDS database:

Table Name: imwliglv
Table Long Name: FM – Improvement Wells Groundwater Level Data
Description: This table contains groundwater level data representing a single groundwater level measurement.

Attribute Name	Data Type	Char Length	Definition
wlelev_id			Primary Key. Begin with wlevel (e.g., wlevel2 for groundwater level measurement record no. 2).
mnwell_id	C	20	Foreign Key. Used to link the record to the applicable WELL record (imwlimas).
wl_date	I		Date that the fluid level measurement in the well was taken. Format for date is YYYYMMDD
wl_time	I		Time that the fluid level drawdown measurement in the well was taken. Format for time is HHMMSS (24 hour clock)
wl_melev	D		Elevation of the datum from which the groundwater level measurements and sounding are made. The datum is the top of casing elevation fro water wells.
wl_static	D		Depth to groundwater. Depths are measured relative to the distance below the measuring point datum elevation.
wlelev_u_d	C		Unit of measure for depths and elevations.

Table Name: imwlimas Table Long Name: FM – Improvement Wells Data Description: This table is a parent table for wells.			
Attribute Name	Data Type	Char Length	Definition
mnwell_id	C	20	Primary Key. The unique identification number of a well. Used to track individual item data records and to link to other data records. Begin with well (e.g., well2 for well no. 2).
subbhl_id	C	20	Foreign key. Used to link the record to the borehole. (gesubbhl)

Table Name: gesubbhl **Table Long Name:** Geology Subsurface Borehole **Description:** This table contains data about a borehole or boring.			
Attribute Name	**Data Type**	**Char Length**	**Definition**
mnwell_id	C	20	Primary Key. The unique identification number of a well. Used to track individual item data records and to link to other data records. Begin with well (e.g., well2 for well no. 2).
subbhl_id	C	20	Foreign key. Used to link the record to the borehole. (gesubbhl)
coord_x	D		The x component of individual coordinate point.
coord_y	D		The y component of individual coordinate point.
coord_z	D		The z component of individual coordinate point.

The results of the query are shown in the following table. Note that the query combines data from the 3 tables above into one table. This means some of the data in the query result are duplicated, but the query result is never saved back to the database so file size is not an issue. Having all the data in one table makes the interface simpler and makes it much easier to do the attribute mapping. Also note that the wl_melev (datum) and wl_static (depth) fields in the database are combined in the query to one field called Elevation.

If the user wanted to limit the imported data to samples collected between a certain beginning and ending time, or to wells located within a certain bounding box, this could be done by adding additional restrictions to the SQL query.

mnwell_id	coord_x	coord_y	coord_z	wl_date	wl_time	Elevation	wlelev_u_d
1	0	0	0	19991225	103000	50	ft
1	0	0	0	19991226	93000	55	ft
1	0	0	0	19991227	110000	65	ft
2	10	0	0	19991225	110000	190	ft
2	10	0	0	19991226	100000	180	ft
2	10	0	0	19991227	113000	175	ft

The attribute mapping would be as shown in the following table. GMS needs a Measurement name and will get this from the name of the query result column that is mapped to the Measurement attribute. If the user wanted to import more than one measurement type at once, we could handle this by listing "Measurement 1", "Measurement 2" etc. in the GMS attribute list. As the data are imported, GMS will first search for an existing observation point with the given name. If no match is found, GMS will create a new one point. Then the measurement data associated with that point will be assigned to it. Any information that GMS needs that does not come from the database (is not mapped or simply isn't available) will be initialized to default values.

Query result column	=>	GMS attribute
mnwell_id	=>	Observation point name
coord_x	=>	X coordinate
coord_y	=>	Y coordinate
coord_z	=>	Z coordinate
wl_date	=>	Date
wl_time	=>	Time
Elevation	=>	Measurement
wlelev_u_d	=>	Units

Observation Wells – concentration data – ehefmres

It is likely that people will want to import their concentration data into GMS. The concentration data could be obtained by measuring the concentration of a certain contaminant in an observation well. Figure 7 shows a diagram of some database tables that appear to be involved with saving this sort of data. Exactly how these tables are used in practice is not entirely clear. When it is clear, an appropriate query could be designed.

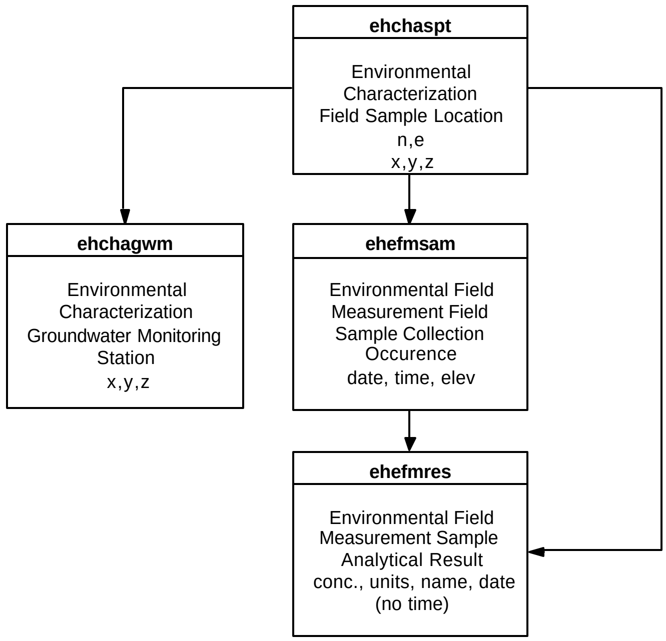

Figure 7. Tables involved in storing concentration measured at observation wells.

Scatter Point Data

In GMS, scatter point sets can have data sets associated with them. These data sets can have one or more time steps. For transient data sets, each time step is associated with a given time. Conceptually, this corresponds to the following table:

Scatter Point ID	Value at Time = 0	Value at Time = 1	Value at Time = 2	Value at Time = 3
1	490	500	480	500
2	200	190	180	180
3	300	320	310	300
…				

The data in a TSSDS database will not likely correspond with this format because different points are usually sampled at different times, or each point has it’s own time series. This corresponds well with GMS observation coverage points or MODFLOW coverage wells, but not with GMS scatter points unless the data are steady state and there is only one time step per point. Thus, importing data from a TSSDS database into GMS as scatter points presents an additional challenge.

In order to overcome this, it will be necessary to build a tool for converting observation points or MODFLOW well points into a scatter point set. This is currently not possible in GMS, although it is possible to go the other way and convert scatter points into observation points. In order to convert observation points into scatter points, the user will need to define the times at which they want to create the time steps for the data set that will be created. This interface can be fairly similar to the Stress Periods dialog in the MODFLOW interface. We could make this part of the import wizard, but by making it separate it is more general and more powerful. Once the times are defined, GMS will use the time series from the individual points to interpolate a value. These values will become the transient data set for the scatter points.

Questions

There are a number of questions that have been raised in designing this interface. Before we finalize the design, these should be answered.

1. Can we get a portion of a TSSDS database for testing?

2. Can we talk to individuals who are familiar with GMS and with a TSSDS database? We need to know what kind of data is being worked with in the real world – which database tables are most frequently used and how.

3. Is there a graphical diagram showing the relationships of the various tables in the TSSDS standards?

4. Several tables (gesubbhl for example) have coord_x, coord_y, and coord_z fields, and a coord_id field which is a coordinate stored in another table. Which should we use?

5. In the gesuilit table, the description for the inttop_ele and intbot_ele fields says "Values deeper than mean seal level are positive." These seems backwards. Same for the gesuilic table, stra_begin and stra_end fields.

6. In lftopsrv and gdsriprj, it looks like xyz survey data is not stored in the database but in external files. Only the paths to the files are stored in the database. If so, we already have an interface to import this type of data.