General Thoracic Surgery Database
# of the
Society of Thoracic Surgeons

Software Specifications

Version 5.21.1

Updated: 2/16/2021

Note: Some portions of this document are highlighted in blue. Although it is critical for the success of the developer's software that all of the information in this document be understood and followed, the highlights are used to point out areas that have changed since previous versions or areas of extreme importance to the functionality of the software.

Table of Contents:

Purpose of Software Specifications:

The purpose of this document is to describe the features that are required to exist in software approved by The Society of Thoracic Surgeons (STS) for the collection and submission of General Thoracic Surgery data. The STS is making an effort to set minimum standards for the software to be used by its members, while allowing enough flexibility so that developers can produce competitive features for the members' benefit.

The intended audience for this document is the software developers who are designing and maintaining the code used by participants to collect and submit data to the STS database. This information will be essential for developers working for vendors who will distribute their software to many members as well as developers working for an individual member designing a package to be used only by themselves (Participant Generated Software).

Note: All software used to collect data to be submitted to the STS Data Warehouse must go through an approval process before data will be accepted into the national database. Developers must also have a signed contract on file with the STS before the approval process can begin.

Since the functionality of the software will revolve around the data specifications, this document will start by providing some information about the specifications.

Data Specifications:

1. Purpose of the Data Specifications

   The data specifications describe the data fields that are required to exist in approved software. It details the field names, definitions, dependencies, acceptable values, the harvest codes associated with those values, etc. Software developers should use the data specifications to ensure their software:

   a. includes all core fields in the application (see description of core fields below)
   b. follows the defined field dependency rules (see description of Parent / Child relationships below)
   c. accepts only the defined valid values appropriate to each field and ensures that the values are in the correct format
   d. provides the user with appropriate warnings or error messages for unusual or missing values.

2. Data Version Numbers

As medicine, technology and interest in research areas change, the data specifications have and will change to collect the additional and more detailed information. A Data Version number is assigned by the STS to each official version of the data specifications. This number will play a key role in how the data are handled and processed (see Software Specifications below).

Note the change in the format of the data version value between v2.41 and v5.21.1. The change was made to make the version number reflect the database and the year it was initially released. Version numbers will now have three levels. The first level (5) will indicate the General Thoracic Surgical Database. The second level (21) indicates the year of the mandatory start date for this version (in this case, 2021). The third level (1) indicates the release number of this version of the specifications. Although every effort is made not to make changes to the specifications once they have been released, in the event that minor changes are needed, the third level will be used to reflect that change and ensure the correct version of the documentation is being used. All three levels should always be used when referencing a specific data version (such as the value placed in the DataVrsn field).

The data version number that is assigned to and followed by a specific operation record is dependent on the date of surgery for that record. When users create a new operations data record, the software must first prompt the user for the surgery date. The format of the operation record that is created must meet the specifications that are appropriate for the version that is accepted for that date. This process will ensure that all records in the national database for procedures performed during a specific time period will follow the same data version, regardless of when the record was created.

For the General Thoracic Database, there are two data version fields: one for the Demographics table (DemogDataVrsn) and one for the Operations table (DataVrsn). A description of these fields follows:

Demographics Data Version (DemogDataVrsn)

The DemogDataVrsn field is necessary because the data version followed to create the patient’s demographics record may not match the version followed to create all of their associated operations records. For example, when a new patient comes to a facility, a Demographics record is created for the patient and an Operations record is created for the operation. In this case, the two data version values will be the same. The same patient can come back to this facility years later to have another procedure performed. By this time, one or more upgrades may have been made to the data specifications.

In this case, the original Demographics record is used which still contains the old DemogDataVrsn value, and a new operations record will be created, which will contain the newest DataVrsn value.

There is no correlation between the value of the demographics data version number and the operation record's data version number. For example, it is possible to have a Demographic record with DemogDataVrsn value of 2.41 associated with an operation record with a DataVrsn value of 5.21.1. It is also possible, if a site is entering data retrospectively, to have a Demographic record with a DemogDataVrsn value of 5.21.1 associated with an Operations record with a DataVrsn value of 2.41.

Operations Data Version (DataVrsn):

The following table defines which version of the data specifications should be followed with surgery dates in the specified time periods:

Surgery date	Data Specifications
Any date up to September 30, 2008	v2.07
October 1, 2008 – December 31, 2008	v2.07 or v2.081
January 1, 2009 – December 31, 2011	v2.081
January 1, 2012 – December 31, 2014	v2.2
January 1, 2015 – June 30, 2018	v2.3
July 1, 2018 – June 30, 2021	v2.41
July 1, 2021 – Present	v5.21.1

Please note that as of 2020 when IQVIA became the Data Warehouse, participants can submit only Operation records with a data version of 2.3 or later and only these records will be accepted. Also note that, since newer Operation records can be associated with older Demographic records, all versions of Demographic records are still being accepted.

3. Sequence Number

The sequence number field (SeqNo) is provided in the data specifications solely for sorting fields within the specific version of the data specification database. They are not intended as a permanent identifier for individual fields as a number assigned to a field in one version of the data specifications might be assigned to a different field in another version. Because of this, it is highly recommended that developers should not use the SeqNo value as a field identifier in any of their programs.

4. Fields Required For Record Inclusion

Starting with version 2.081, specific fields are required to contain data for the entire record to be included in the analysis performed at the data warehouse. These fields are indicated in the data specifications by the field “RequiredForRecordInclusion” set to “Yes”. All of these fields must contain valid data for the entire record to be accepted into the data warehouse. If any one of these fields is missing data (while taking appropriate parent/child relationships into account), the entire record will be excluded from the analysis.

Which specific fields, and the total number of fields that are required for record inclusion will be different in each data version, and each version of the data specifications should be reviewed to gather this information.

Several of the required fields are part of the “check all that apply” fields to collect the patient’s race information (the specific fields depend on the Data Version). Only one of these fields must contain a value of “Yes” to meet the “required for inclusion” criteria. All of the other fields must contain data for the record to be included.

It is highly recommended that software developers provide their users with reports indicating which, if any, records do not meet these criteria. If this is included as part of a larger data quality report, this information should be in a separate section to help emphasize the importance of this issue.

5. Future Upgrades

As the need arises, new versions of the data specifications will be distributed by the STS. In the interest of keeping major software upgrades and testing down to a minimum, the STS does not expect to upgrade the specifications more frequently than once every other year. Developers should anticipate these upgrades and design their software in such a way that the new versions can be incorporated with minimal software changes and that records created under different data versions will be handled properly, as described below.

When upgrades are made to the data specifications, the older versions of the data will not be converted to a newer version. Instead, the software will be required to handle data in all of the valid data versions as described below.

6. Data Specifications Field Descriptions:

The data specifications are maintained in tables in an Access database to allow the information to be cut and pasted, sorted and reported on in a variety of ways and to make incorporating the information into applications easier. The 5.21.1 version of the specifications contains the following tables and fields:

Table name: tblThoracicDataSpecsV5_21_1

A. SectionName – The title of the section of the DCF in which the field can be found

B. SectionSeqNo - The order number of the section of the DCF where the field is located.

C. SeqNo – An arbitrary number (sequence number) used for ordering the fields within a specific version of the data specifications. The ordering of the numbers is set to loosely follow the order in which the fields appear on the DCF. As described above, the value for one field can change from one version of the specifications to the next. The values, therefore, should never be used in any reports, queries or programs to refer to a specific field.

D. LongName – The longer and more descriptive name of the field. In most cases, the FieldName does not change from one version of the specifications to the next, but they do change in some instances. Because of this, the LongName value should never be used to refer to a field in reports, queries or programs.

E. ShortName – The short programmatic name assigned to the field. The ShortName value should be used in all reports, queries and programs to refer to a given field as this value will not change from one version of the specifications to another.

F. Core – This field contains a value of Yes or No to define whether or not the field should be available to the users for data entry. These values have the following meanings:
   - Yes = Field must be available to the users for entering data for records following this version of the data specifications and the field must be included in the data files exported for submission to the STS database that contain records following this data version.
   - No = Field is not required to be available to the users for entering data for records following this version of the data specifications. Whether or not the field is included in data files exported for submission to the STS database depends on the Harvest value described below and on what other data versions are being included in the data extract. (See the “Data Export for Harvest to the Data Warehouse” section of the Software Specifications below.)

G. Harvest – This field contains a value of Yes, No or Optional to define whether or not the data for this field is included in the export file to be submitted to the data warehouse. (See the “Data Export for Harvest to the Data Warehouse” section of the Software Specifications below for more details about the contents of the submitted files.) The values for this field have the following meanings:

- Yes – Data from this field must be included in the data file for all records following this version of the data specifications.
- No – Data from this field must not be included in the data file for all records following this version of the data specifications.
- Optional – The individual users determine whether or not the data from this field is included in the data file. By default, the software should treat this as a Yes and include the data in the extract. The users must explicitly state that they do not want the data for this field included.

H. Format – The format in which the values for the field should be collected. The options for this field are:

- Date in mm/dd/yyyy format: Date values only with the month specified as a 2-digit numeric value, day specified as a 2-digit numeric value, and year specified as a 4-digit numeric value, all with leading zeros when appropriate.
- Time in 24 hour hh:mm format: Time values only with the hours specified as a 2-digit numeric value (in 24-hour format), and the minutes specified as a 2-digit numeric value, both with leading zeros when appropriate.
- Integer: Numeric values with no decimal place.
- Real: Numeric values with at least one decimal point.
    - Note that the fields defined with this format should store and display A MINIMUM of the same number of decimal places defined in the LowValue and HighValue parameters for this field. Additional decimal places can be displayed and ideally the software will store and display whatever number of decimal places the user enters.
- Text: Value can contain any alphanumeric characters.
- Text (categorical values specified by STS): Values displayed to the user are the text descriptions defined in the data specifications harvest codes table. The values submitted to the Data Warehouse are the Harvest Codes

defined in the data specifications.

- Text (categorical values specified by user): Values displayed to the user and submitted to the Data Warehouse come from a list maintained by the user (see item “g” under the “3. Data Entry” section of the “Software Specification” below).
- Multi-Select: This format is similar to the “Text (categorical values specified by STS)” format in that values displayed to the user are the text descriptions defined in the data specifications harvest codes table. However, for fields with a multi-select format, users can select more than one choice. The values submitted to the Data Warehouse are a comma-delimited list of the harvest codes associated with each choice indicated by the user. **(Please do not include any spaces between the choices).**

  For example, if the user enters 72 for Patient Age, selects “White”, “Black/African American”, and “Asian” for Race – Multi-Select, and enters “General Hospital” for Hospital Name, that portion of the submitted data record would look like this:

  ```
  …|Age|RaceMulti|HospName|…
  …|72|1,2,3|General Hospital|…
  ```

  Developers should also be aware that multi-select fields can be the parent of other fields. The child field should be enabled for data entry if ANY of the choices selected in the multi-select parent field are in the ParentHarvestCodes list for the child field. The child field should be disabled if NONE of the choices selected in the multi-select parent field are in the ParentHarvestCodes list for the child field.

I. DBTableName – The name of the table in the export data file where the field is located. (See “Data Export for Harvest to the Data Warehouse” section of the Software Specifications below).

J. DataSource – This field defines how the data is entered into the field. The options for this field are:

- User – The user enters the value, otherwise it is left missing (null).
- Automatic – The software automatically inserts a value for every record. This is usually assigned to administrative fields that must contain a value, such as the DataVrsn field.
- Automatic or User – The value can be entered automatically by the software based on values in other

fields or the user can enter the value manually. This usually applies to fields such as Participant ID where all data entered by a user is for one participant so the value can be automatically entered, or a user might enter data for more than one participant so the value should be entered by the user.

- Calculated - The value is calculated by the software based on values in other fields. For v5.21.1 there is only one calculated field: Calculated BMI (CalculatedBMI). The formula that must be used for this calculation is defined in Appendix A below.
- Lookup – The software automatically inserts a value after looking up the information kept in a table maintained by the user (for example, HospStat is filled in based on which HospName value is selected). For v5.21.1 the following table defines which fields are used drive the values contained in the lookup fields:

Driving field	Lookup field
Surgeon's Name (Surgeon)	Surgeon's National Provider Identifier (SurgNPI)
Hospital Name (HospName)	Hospital Region (HospStat)
Hospital Name (HospName)	Hospital Postal Code (HospZIP)
Hospital Name (HospName)	Hospital National Provider Identifier (HospNPI)

K. Definition – A description of the information to be collected in the data field.

L. LowValue – The lowest valid value that can be accepted for the specified field. This is used only in fields that accept numeric values. If field values are submitted to the Data Warehouse that are less than LowValue, the entire data record will be rejected.

M. HighValue – The highest valid value that can be accepted for the specified field. This is used only in fields that accept numeric values. If field values are submitted to the Data Warehouse that are greater than HighValue, the entire data record will be rejected.

N. UsualRangeLow - The lowest value that is likely to be entered by the user. If the user enters a value that is below this number, but still greater than or equal to the value defined in LowValue, the value should be accepted, but the user should be given a message that the value they entered is unusually low and that they should verify the value.

O. UsualRangeHigh - The highest value that is likely to be entered by the user. If the user enters a value that is above this number, but still less than or equal to the value defined in HighValue, the value should be accepted, but the user should be given a message that the value they entered is unusually high and that they should verify the value.

P. ParentLongName – The long name of the “parent” field on which this field (the “child” field) is dependent. The parent field must contain a value that is specified in the ParentValue field before data can be entered into this field.

Q. ParentShortName – The STS short name of the parent field.

R. ParentValue – The list of values the parent field must have before this field can be available for data entry. This field may also contain general instructions instead of specific values, such as “Is Not Missing”.

S. ParentHarvestCodes – A bar-delimited list of the harvest codes associated with the values listed in the ParentValue field. This field may also contain general instructions instead of specific values, such as “Is Not Missing”.

T. FieldStatus – The status of the field as it relates to the previous version of the data specifications. This field has one of three values:

- New – Field is new to this data version and did not exist in the previous version of the specifications.
- Dropped – Field was a core field in the previous version of the data specifications, but is not a core field in this version.
- Continued – Field was a core field in the previous version of the data specifications and is also a core field in this version. These fields may or may not have undergone changes between the two versions.

U. RequiredForRecordInclusion – This field contains a value of Yes or No and defines whether a value must be present in the field for the entire data record to be included in the analysis performed at the Data Warehouse.

Table name: tblThoracicDataSpecsV5_21_1_HarvestCodes

A. DbTableName - The name of the table in the export data file where the field is located.

B. ShortName – The short programmatic name assigned to the field.

C. HarvestCode – The code that is assigned to each choice in the valid data. These are the values that are used in the exported data file that is submitted to the Data Warehouse.

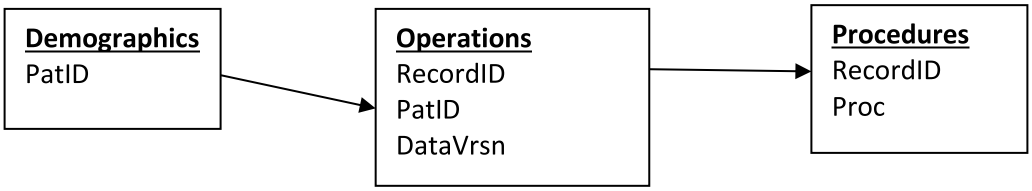
D. Description – The text description of the choice. This is the value the user sees while doing data entry.

E. DisplayOrder – The order in which the choices are displayed to the user for this field.

F. Definition – The official definition of the specified choice for this field. Note that not all choices will have a definition.

Database Structure:

The General Thoracic database has a relational structure made up of three tables: Demographics, Operations, and Procedures. The following diagram depicts how the tables relate to each other. (Note, each table contains more fields than are depicted here. Only the fields involved in the table relationships are included in this diagram.)



The contents of each table and how they are linked to each other is described below:

Demographics: This table contains one record for each patient. The Patient ID (PatID) field is the primary key for each record to link to the associated Operations records.

Operations: This table contains one record for each operation performed on a patient. There can be many operation records in this table for each one patient record in the Demographics table. The Patient ID (PatID) field is the foreign key field linking this table to the Demographics table. The RecordID field is the primary key for each operation record linking it to the Procedures table.

Procedures: This table can contain multiple records for every patient in the Demographics table, one for each procedure performed. It can also contain multiple records for each Operations record since more than one procedure can be performed during a single operation. The RecordID field is the foreign key linking each record to the associated record in the Operations table.

For more information about the tables and records, see the "Record Management" section of the "Software Specifications" below.

Software Specifications:

It is not the intention of the STS to regulate the algorithms and methodologies the developers use to produce their software. However, there are specific features and functionalities that are needed in the software to allow data to be collected and submitted in a uniform format and to enable the warehouse to communicate with the members about individual records and data items. The purpose of this section is to describe those features and functions.

1. General Features

   The approved software must have the following minimum features:

   a. A user-friendly interface that can be used on a current personal computer operating system.

   b. Allow users to be able to view and select the actual data values for each field. If the data is coded internally, user should, by default, view the non-coded values.

   c. Ensure all date values are year 2000 compliant having a 4-digit year format.

   d. The STS General Thoracic database has a relational structure composed of 3 tables. Regardless of the method of internal storage, the software must be able to export the data into one data file in the format specified by the STS. (See "Data Export for Harvest to the Data Warehouse", below.)

   e. Software must accept and integrate data previously collected and maintained in other software products or data versions. (See "Data Import", below).

   f. Users must be able to select specific records in their database via key fields including patient's name, Patient ID (PatID), and the Record ID (RecordID). The data warehouse communicates specific record problems with the sites using the PatID and RecordID fields. Therefore, these specific fields must be visible to the user and labeled with these field names.

   g. Software must execute a specified list of data quality checks when a user attempts to save a record and not allow the user to save the record if any of the conditions are true. See "Data Quality And Completeness Checks" below.

2. Record Management

Demographics

Each record in the Demographics table of the database describes one patient. On each Demographics record, there are two key field used for record management:

a. Operations Table Patient Identifier (PatID): The PatID field contains a unique, arbitrary alphanumeric value to uniquely and permanently identify each patient in the Demographics table and is required for each Demographics record. Regardless of the number of operations or admissions to the hospital, only one Demographics record should exist for each patient. The value, once assigned to a patient, cannot be edited or reused if the patient records are ever deleted. In order to avoid issues of patient confidentiality, the PatID value should not include any known identifier such as Social Security Number or Medical Record Number.

   Beginning with version 2.2 of the data specification, the values generated by the software for the PatID field must be a combination of a vendor specific code followed by an alphanumeric value that makes the identifier unique. The vendor-specific code will consist of three characters and will be assigned to each vendor by the STS. The codes will be in a format similar to "V01". For example, the software will generate a PatID value of V01000001 for the first demographic record and V01000002 for the second demographic record. The purpose of this feature is to allow sites to move their data from one version of a software package to another, or from one vendor package to another, and maintain the referential integrity of their data records.

b. Demographics Table Data Version (DemogDataVrsn): The DemogDataVrsn field contains the data specifications version number under which the record is created. The value is automatically entered into the record by the software at the time the record is created. Note that the DemogDataVrsn value may not always match the DataVrsn value in Operations records that are associated with this Demographics record.

Operations

Each record in the Operations table of the database describes one trip to the Operating Room (OR) for a patient. On each Operations record, there are four key fields used for record management:

a. Operations Table Record Identifier (RecordID): The RecordID field contains a unique, arbitrary alphanumeric value to uniquely and permanently identify each operation in the Operations table and is required for each Operations record. A new Operations record should be created for each operation or operating room visit and linked to the patient via the PatID field (see PatID below). Once the RecordID is assigned to a record, it cannot be edited or reused if the record is ever deleted. In order to avoid issues of patient confidentiality, the RecordID value should not contain any known identifier such as Social Security Number or Medical Record Number.

   Beginning with version 2.2 of the data specification, the values generated by the software for the RecordID field must be a combination of a vendor specific code followed by an alphanumeric value that makes the identifier unique. The vendor-specific code will consist of three characters and will be assigned to each vendor and Participant Generated Software site by the STS. The codes will be in a format similar to "V01". For example, the software will generate a RecordID value of V01000001 for the first operations record and V01000002 for the second operations record. The purpose of this feature is to allow sites to move their data from one version of a software package to another, or from one vendor package to another, and maintain the referential integrity of their data records.

b. Operations Table Patient Identifier (PatID): The PatID field contains the patient identification number from the Demographics table that identifies the corresponding patient. For patients with multiple operations or hospital admissions, the same identifying PatID should be used to link these operations to the appropriate patient. The PatID is required for each Operations record. See PatID under Demographics (above) for more information.

c. Participant ID(ParticID): Each group of surgeons collecting and entering data into a database for submission to the STS is assigned a 5-digit ParticID number by the STS. In most cases, all data being entered into a database will be for one participating group, in which case all records will have the same value in this field. In these situations, the developer can have the software enter the value into the record automatically for the user.

   In some situations however, more than one participating group will be entering their data into a single database. In these situations, the user should select the appropriate ParticID value from a drop down list (see item "g" under the "3. Data Entry" section of the "Software Specification" below).

The developer should consult with the users to determine how many participants will be entering data into a single database and adjust the programs accordingly. In either case, a value for ParticID is required and the software should ensure one exists for every Operations record.

d. Version Of STS Data Specification (DataVrsn): The DataVrsn field contains the data specifications version number under which the record is created. The value is automatically entered into the record by the software at the time the record is created.

The data version value that is assigned to a record is determined by the surgery date of the procedure. Once a data version value has been assigned to a record, the value should never be changed by the software or the user.

Once an Operations record is created and it has been assigned a data version number, that record will always follow the rules defined by that version of the data specifications. If the software is upgraded to follow a newer version of the data specifications, when a user selects a record for editing that has an older data version number, the software must follow the older data specification rules for editing that record. This includes controlling which fields are available to the user, which values are available for each field and the appropriate parent/child dependencies.

For more information about the DataVrsn field, see the “Database Structure” section above.

Note that the DataVrsn value on this table will also define which fields and values are available to the user for any records in the Procedures table that are associated with this Operations record.

Procedures

Each record in the Procedures table of the database describes one procedure for an operation. Each operation record can be linked to multiple procedure records. On each Procedures record, the RecordID field is the key field used for record management. The RecordID field in the Procedures table contains the RecordID value from the associated record in the Operations table. For patients with multiple procedures for a given operation, the same identifying RecordID would be used to link these procedures to the appropriate operation. The RecordID is required for each Procedure record. See RecordID under Operations for more information.

3. Data Entry

The software must have the following features to control the data being entered by the users:

a. For data entry purposes the site and vendor may choose to institute internal codes for "Missing" values. As the site drives the needs for this feature, the STS data specifications do not define standard codes for "Missing" values during data entry. If a site applies data entry "Missing" codes, the harvest process must include a step that maps the missing code to the STS specification for "Missing" values (null or blank). Note: zero is never used to indicate missing data.

b. The user should always be able to delete entered data, and return the field's value to the null or blank "Missing" value.

c. For any field having specific values or a range of acceptable values defined, the software must restrict data entries to this set of values. For categorical variables this is expressed as a set of harvest codes and descriptions and the user must select from a pick list of these values. For numerical variables, this is expressed as a valid numeric range defined as a LowValue and HighValue, and the user must enter a value on or between the specified limits. If the user enters a value that is not one of the harvest codes or is outside of the defined range, the user must be given an error message that the value is invalid and the invalid value must not be stored in the database. It is recommended that the software allow the user to enter any value and then check it for validity, such as when the user leaves the field. Stopping a user while they are typing could cause erroneous values to be left in the data. For example, if a field should accept a numerical value between 1 and 100 and the user intends to enter the value 110, stopping the user from entering the third digit could cause the value of 11 to be left in the field (which is technically valid but not the intended value).

d. Where a numeric variable has a UsualRangeLow and UsualRangeHigh specified, if the user attempts to enter a value that is outside of that range but still inside the LowValue / HighValue range, the software must warn the user that the value entered is valid but is outside of the expected range.

e. Documentation including data definitions and help should be easily accessible to the user, preferably on-line.

f. Some categorical text fields are designed to have data values controlled by the user. These fields are identified in the data specifications by having a Format = "Text (categorical values specified by User)". This applies primarily to a few site-specific fields such as hospital name and surgeon name. The user should be able to maintain the pick list of valid data for these fields including the ability to add, change, or delete list elements. During data entry, the user should be able to enter only values that are in this pick list.

The process of maintaining the list should be separate from the data entry process. In other words, if a user doing data entry needs to enter a value that is not already on the list, they should not be able to add the value to the list without first exiting out of entering data for that case. If a user attempts to enter a value during data entry that is not already on the list, it should be rejected and not automatically added to the list. The idea here is to avoid the possibility of users entering "free text" which causes unacceptable data quality issues at the warehouse.

It is important that the vendor support the site's ability to control these fields. Items in the user list should not have more than one choice for the same entity. For example, the hospital names "General Memorial Hospital" and "GMH" should not represent select choices for the same hospital.

4. Importing data from other data sources

Although the data many participants are entering into their STS certified software may be gathered from another electronic data system at their site (such as an EMR), it is strictly against STS policy for vendors to provide the users with the means to import this data automatically. It is not practical for the STS to certify the mapping of data from each site's EMR to the STS data specifications, which would be required to ensure the integrity of the overall STS database.

For the General Thoracic database there is only one exception to this policy:

a. The following data fields can be imported from an Admission/Discharge/Transfer (ADT) system:

LongName	ShortName

Medical Record #	MedRecN
Patient's First Name	PatFName
Patient Middle Name	PatMName
Patient's Last Name	PatLName
Social Security Number Known	SSNKnown
Social Security Number	SSN
Permanent Street Address	PatAddr
Patients Permanent City	PatCity
Patients Permanent Region	PatRegion
Country	PatientCountry
Postal Code	PostalCode
Date Of Birth	DOB
Age At Time Of Surgery	Age
Gender	Gender
Race Documented	RaceDocumented
Race - Multi-Select	RaceMulti
Hispanic Or Latino Ethnicity	Ethnicity
Admission Date	AdmitDt
Date Of Surgery	SurgDt
OR Entry Time	OREntryT
OR Exit Time	ORExitT
Hospital Discharge Date	DischDt
Mortality Date	MortDate

5. Field Dependencies

Field dependencies exist where one field (“parent” fields) controls whether or not one or more other fields (“child” fields) can contain data. Child fields are indicated in the specifications by having their immediate parent field named in the "Parent Field" section of their specification. For example, Creatinine Level Measured is a parent field to its child Last Creatinine Level. The following guidelines must be followed to handle dependent fields:

a. If the data value of a parent field indicates that no data should be in its dependent fields, then those dependent fields should be skipped or unavailable on the data entry screen. In the example above, only if Creatinine Level Measured = "Yes" should Last Creatinine Level be available for data entry. Otherwise, Last Creatinine Level should be unavailable for data entry.

b. If a parent field indicates that no data should be in its dependent field, vendors must set all child fields to Null. **Note that in prior versions of the Software Specifications, vendors had the option of setting**

child field values to "No" provided those fields were set to Null during data extract. This has caused parent/child issues to appear in site data, so this practice is no longer acceptable.

c. If a parent field is originally set to "Yes", then values can be entered into its child fields. If the record is subsequently edited by the user and the parent value is changed to "No", **the values in the child fields must be automatically changed to Null**

d. Note that starting with v5.21.1 some parent fields might be multi-select fields and can contain multiple values. In these instances, the ParentHarvestCodes will be defined with "contains(XYZ)" where "XYZ" is a comma-delimited list of harvest codes. If one or more of the listed values were selected in the parent field, then the child field should be enabled and available for data entry. If NONE of the listed values were selected in the parent field (or the parent field is Null), then the child field should be disabled, not available for data entry, and cleared of any values.

6. Data Quality and Completeness Checks

The software must provide the users with a utility for checking the accuracy and completeness of their data that includes the following features:

A. Data quality checks can be run during data entry and/or on demand for groups of records as specified by the user. This utility produces a data quality report indicating which records and fields failed the data checks. This report is used by the site data manager to review and potentially repair the data.

B. Certified software must contain a utility for checking and reporting on data completeness. This utility must include the following features:

   i) The user must be able to identify in a list the fields that they want to have checked for completeness. The user should be able to select just one field, all fields, or any number of fields desired (by default, the utility should report on ALL fields). It is recommended that user should be able to save the selected list so as not to have to go through the selection process again the next time data quality is being checked.

   ii) The utility should report on individual records or groups of records (recommend grouping by surgery date range) as specified by the user.

   iii) The utility must take into consideration dependent fields when checking for completeness. For fields defined as "child" fields of a "parent" field, the child is considered missing only if the parent is answered "Yes" (or in a way that would allow the user to enter data into the child field) and the child field contains no data. Following this guideline will restrict reporting missing data to only those situations where data is clinically expected.

C. Certified software must run quality control checks at the time the user attempts to save a new or modified record. The quality checks to be run are defined in the associated document "STS General Thoracic On-Save Quality Checks". If any of the checks are met (i.e., return a value of True) the user must be presented with the appropriate error message and the record should not be saved. The user should not be able to leave the record until the issue has been corrected or all changes have been voided.

7. Data Import

a. Software must be able to import data in standard file formats from third party applications. At a minimum, this must include delimited, ASCII text files. Other common formats (e.g., Excel or MS Access) are also recommended. **This functionality is to only be used on a one-time basis.** For example, this utility should only be used when a user first purchases a new certified software package and wants to import the data they had been collecting up to that time in a different package. Once the old data has been imported into the new package, all future data should be entered directly into the new package via the data entry screens and no additional data should be imported. Using the import feature to regularly import data (such as from a hospital's Electronic Medical Record (EMR) or Electronic Heath Record (EHR)) so that it can be exported in the STS format for submission to the Data Warehouse is strictly against the STS policies and may affect the certification status of the vendor.

b. Data that is imported will require controlled conversion to an acceptable STS data version. The conversion process must include reviewing the data for consistency with the STS data (i.e., mapping the categorical values in the imported data to the appropriate STS values). The site data manager and software vendor hold responsibility for the accuracy (both clinical definition and harvest format) of all imported data harvested to the warehouse. The software will assign to each imported record the STS data version number to which the data is converted. The warehouse will handle data according to the STS data version number on each observation in a harvest file regardless of whether it was created in the software's data entry utility or imported from another source.

c. Special consideration is needed for the values in the PatID and RecordID fields when importing data. This is especially true when importing data that was previously submitted to the data warehouse (i.e., data from another certified software package). PatID and RecordID values must never change once they are assigned to a record. The software developers and data managers must ensure that the values in the imported data do not change in the conversion process, and that they do not cause duplication of values with any existing records. If the PatID and RecordID values being imported contain the vendor codes for the previous vendor, keeping the old values as they are will help maintain this data integrity. In other words, do not change the vendor codes in the records being imported.

d. Developers must also ensure that new records created after the data has been imported are not assigned RecordID or PatID values that already exist in the data. If data is to be imported that would cause a conflict in this manner, the software developer and or data manager must contact the data warehouse to determine what steps need to be taken.

8. Record Subsets and Queries

Software must allow users to search for Individual records selected by RecordID, PatID or by patient identifiers including patient name and surgery date. This is to help users with their data quality actions and to increase the usability of the database.

9. Reporting

Software should provide the users with reporting abilities that can do the following:

a. Data harvest procedure provides the site with a report documenting the following:
   i) whether or not the extract completed successfully
   ii) number of records extracted
   iii) time frame of the data extracted (by date of surgery)
   iv) date the data extraction was performed
   v) name of the person who performed the data extraction

10. Data Export for Analysis by Users

The software must allow users to export their data for their own use in the following manner:

a. Software must be able to export data in standard file formats suitable for transfer into third party applications. This must include at a minimum delimited, ASCII text, and optionally other common formats such as Excel and Access. Developers should keep in mind that sites may need to export their data for reasons other than the STS data harvests.

b. User should be able to choose whether an export includes all data or selected records and fields.

c. If data is coded for internal storage (i.e., text string is stored as a number), the data must be decoded when written to the export file so that actual values (full text strings) are contained in export file. The user can decide which format should be used for each export file.

d. Export files must have short field names in the first header row in the same order as the data in subsequent rows.

e. User can control export file naming convention.

11. Data Export for Harvest to the Data Warehouse

As one of the key reasons for having certified software, the software must allow users to export their data for submission to the STS data warehouse following these exact guidelines:

a. The user must be able to specify the records to be exported for harvest by using range limits for the surgery date.

b. Note that users must not be able to select individual records for exporting (for example, by RecordID value). It is acceptable for users to extract data for one specific day by specifying the same date value for the beginning and ending dates of the requested export time period. If there is only one record with that surgery date, then it is acceptable that the export file would contain only that one record. However, users must not be able to pick only one record for export as this would cause other records at the Data Warehouse for that date to be deleted from the database.

c. The data for each of the three tables must be exported to one file precisely as indicated here:

i) The harvest data file will be a bar (a.k.a. “pipe”) delimited ASCII text file. It will not, however, be a “standard flat file” since different records will contain different numbers of fields

ii) The data of each relational table will be included in the file. Each table will be separated by one line containing a "table delimiter" followed by another line with the table's header record. The table delimiter is defined as three asterisks followed by the table name (e.g., ***Demographics). The header record consists of the field (short) names in the order in which they appear on each data record, separated by the bar delimiter.

iii) Each of the three tables should be present in the data file in the order specified below. The resulting data file should be in the following format:

```
***Demographics
(... header record for the Demographics table ...)
(... data records for the Demographics table ...)
***Operations
(... header record for the Operations table ...)
(... data records for the Operations table ...)
***Procedures
(... header record for the Procedures table ...)
(... data records for the Procedures table ...)
```

d. What records are included in the extract file varies depending on the table:

iv) For the Operations table, all records with a surgery date between the starting and ending dates specified by the user must be included in the export file.

v) For the Procedures table, all records linked to the included Operations records must be included in the export file.

vi) For the Demographics table, ALL database records must be extracted, regardless of whether or not they are linked to the included Operations records. This is necessary due to the nature of thoracic operation procedures and the "rolling harvest" procedures used at the data warehouse. This method ensures that patients with multiple admissions to the hospital over several years are captured correctly over each data harvest.

e. Only a single harvest file for each participant can be submitted to the warehouse for processing. Participants may submit repeatedly during a harvest, but each submission must consist of only one file.

f. The extracted file must contain data for only one ParticID. If the site's database contains data for more than one participant, all of which is to be submitted to the warehouse, the software must extract the data for each ParticID into separate data files each with an appropriate file name (see below).

g. The harvest file must include all fields, and only those fields, defined in the data specifications where Core is ”Yes” and Harvest is “Yes" or “Optional” for all STS data versions within the harvest file. Fields with Core=”No” or Harvest=”No” and site-specific or custom fields must not be included in the export file.

h. Fields that are defined as Core is “Yes” and Harvest is “Optional” must be included in the data file. What is “optional” is whether or not the field contains data. By default, the software should include all data for optional fields. If the user specifies that an optional field should not be included, the data file will include the field but every record will contain a blank (null) in that field. This is necessary for the warehouse to be able to tell the difference between a field being left out by mistake and a site opting not to include that data.

i. The values in the harvest file must be the “Harvest Coding” of the data values and not the full text strings.

j. A harvest report should be produced whenever a data harvest is performed (see “Reporting”, above)

k. **Note that starting with data version 5.21.1, the naming convention to be used for extract files has changed.** The software must create the exported data file using the file naming convention of XXXXXthr_YYYYMMDDhhmm.dat where:

- XXXXX is the 5-digit ParticID for the data in the file
- YYYY is year portion of the date on which the extract was created
- MM is the 2-digit numeric month portion of the date on which the extract was created (with leading zeros as needed)
- DD is the 2-digit numeric day portion of the date on which the extract was created (with leading zeros as needed)
- hh is the 2-digit numeric hour portion of the date on which the extract was created (using a 24-hour clock with leading zeros as needed)
- mm is the 2-digit numeric minute portion of the date on which the extract was created (using a 24-hour clock with leading zeros as needed)

For example, if participant 12345 creates an export file on September 6, 2020 at 3:05 pm (local time), the extract file must be named “12345thr_202009061505.dat”.

The user must not specify the file naming convention. Files not using this naming convention will not be accepted by the automated process at the Data Warehouse.

l. When records from more than one data version are being exported for an STS data harvest, the file must adhere to the following format:

  - All data records for a single participant must be exported into one and only one data file.

  - For each of the three tables included in the export file, there must be only one "header" record containing the STS short field names in the same sequence as the data fields in subsequent rows.

  - Every data record for each table in the file must contain the same fields which will consist of a superset of the Core, Harvested fields from all included data versions.

  - On each data record, the fields that are Core and Harvested for the data version specified in the DataVrsn (or DemogDataVrsn for Demographics data) field will contain data values as available and appropriate. The fields that are not Core or not Harvested for that data version will contain nulls (blanks). When the data is being processed by the warehouse, only the fields appropriate for the data version specified on the record will be included.

12. Customization

It is up to the developer's discretion as to whether or not the users will have the ability to add customized fields to their software and database. If the user will have this ability, the following items must be considered:

a. In no case can the field names, short field names, or categorical data values specified by the STS be customized or modified by the users. (Please note however in the STS specifications that users can build the categorical data values for certain fields, see "Data entry", above.)

b. Fields added by users must not be included in the data file exported for submission to the STS data warehouse.

c. Developers should make clear to the potential users whether users can add custom fields themselves, or if they will require contracted work by the developer.

d. It should be possible for users of customizable software to import custom fields that they might have created in a previous database or software package.

e. Most importantly, developers who allow users to add customized fields must keep in mind that software upgrades will be necessary from time to time as new versions of the data specifications become available. It is the developer's responsibility to handle how a user's customization is incorporated when their software is being upgraded.

13. Suggested features

Although the data specifications require the patient's height and weight to be collected in centimeters and kilograms, in some instances the information available to the data manager entering this data is in inches and pounds. Many users have requested a means for converting values from imperial units to metric units and it is suggested that developers provide this in their software. This is not a required feature and, if it is implemented, it must not interfere in any way with the specified height in centimeters and weight in kilograms fields and the imperial values must not be included in the submissions to the Data Warehouse.

Appendix A: Calculation of BMI (CalculatedBMI):

Starting with version 5.21.1, software must be able to calculate the Body Mass Index (BMI) score for each patient. The results from this calculation are entered by the software into the field “Calculated BMI” (CalculatedBMI). The value of this score is calculated using the values entered by the user into the fields “Weight (kg)” (WeightKg) and “Height (cm)” (HeightCm) using the following formula:

$$\text{CalculatedBMI} = \text{WeightKg} / ((\text{HeightCm}/100))^2$$