General Thoracic Surgery Database
of the
Society of Thoracic Surgeons

Software Specifications

Version 2.07

Table of Contents:

Purpose 3
Data Specifications 3
1. Purpose of the Data Specifications 3
2. Data Version Numbers 3
3. Sequence Number 5
4. Future Upgrades 5
5. Data Specifications Field Descriptions 6
Database Structure 9
Software Specifications 10
1. General Features 10
2. Record Management 11
3. Data Entry 13
4. Field Dependencies 15
5. Data Import 16
6. Record Subsets and Queries 16
7. Reporting 17
8. Data Completeness Checks 18
9. Data Export for Analysis by Users 18
10. Data Export for Harvest to the Data Warehouse 19
11. Customization 21
Appendix 23
1. Consistency Edits 23

Purpose:

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).

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. provides a defined default value if appropriate
   d. accepts only the defined valid values appropriate to each field and ensures that the values are in the correct format
   e. 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 is handled and processed (see Software Specifications below).

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 is as follows:

Demographics Data Version (DemogDataVrsn):

The DemogDataVrsn fields was added as a new field in the 2.07 data specifications. This 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 and has a procedure done, 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, after one or more upgrades have been made to the database software and data specifications. In this case, the original Demographics record is used, which still contains the old data version value, and a new operations record will be created, which will contain the new data version.

Operations Data Version (DataVrsn):

For the 2003 data harvest, all data submitted to the database was collected using one software package that was distributed by the STS and was formatted to the 1.3 version of the data specifications.

In early 2004, the data specifications were upgraded to version 2.06. At that time, the STS software package was modified to record new data records in the 2.06 format. The 2004 data harvest included data in both 1.3 and 2.06 data versions.

In January of 2005, version 2.07 was released to vendors to allow them to develop commercial software packages for collecting General Thoracic data and submitting it to the STS database. It was decided that the shareware software package distributed by the STS will not be upgraded to this new data version. Participants using the STS software will need to purchase one of the commercial packages to continue collecting and submitting data to the STS database.

It was also decided that commercial software packages will handle all data records collected in the past as well as newly created records according to

the 2.07 data specifications. This means that if a participant has their data that was collected under versions 1.3 or 2.06 pulled over into a commercial package, those records will be handled by the new software following the 2.07 specifications. This is a different procedure from how data with different data versions would normally be handled by commercial packages, but this change was necessary to allow participants to carry their old data over to the new software systems.

To enable the commercial software packages to handle the older data, records with a data version of 1.3 or 2.06 need to go through a conversion process. The STS will provide participants with software that will extract the data from the shareware software package, convert it to the new format, change the data version numbers from 1.3 to 1.31 and from 2.06 to 2.061, and save the data in a format that will allow it to be imported into the new commercial packages. The commercial packages will then treat records with data versions 1.31 and 2.061 as if they were version 2.07 records. New records created in the commercial packages will be given a data version of 2.07.

All data records with a procedure date of January 1, 2006 or later will be required to be in the 2.07 format.

3. Sequence Number

   The sequence number field (SeqNo) is provided in the data specifications solely for identifying fields and 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 <u>developers should not use the SeqNo value as a field identifier in any of their programs</u>.

4. 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 after version 2.07, 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.

5. Data Specifications Field Descriptions:

   The data specifications are maintained in a table 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 table for the 2.07 version of the specifications contains the following fields:

   A. 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.

   B. TableName – The name of the table in the export data file where the field is located.

   C. FieldName – 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 FieldName value should never be used to refer to a field in reports, queries or programs.

   D. 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.

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

   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.

- 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.

I. 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 Hospital Postal Code where the value can be automatically inserted by the software after the user has selected the hospital name.

- Calculated – The value is calculated by the software based on values in other fields.

J. Default – The default value in a field if no other value is entered. This field can have the following values:
   - (null/blank = missing) – Most fields are left missing if no value is explicitly entered. A null is used to specify missing.
   - (assigned value) – A specific value that is constant across most records is entered onto every record (such as VendorID).
   - (unique value) – A specific value that is determined by the software that is unique to each record and is entered by the software (such as RecordID).
   - (database assigned value) – A specific value that is derived by the database. This is usually a foreign key value that is used to link an associated record in one table with its primary record in another table (such as RecordID).

K. ValidData – The values that can be accepted for the specified field. This can be a list of values or a numeric range. (See the "Data Entry" section of the "Software Specifications" portion of this document).

L. HarvestCoding – The numerical 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.

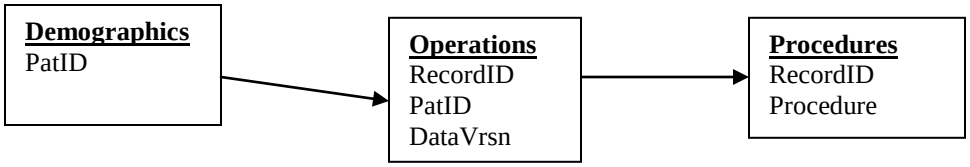
M. ParentField – The short name of the "parent" field on which this field (the "child" field) is dependant. The parent field must contain a value that is specified in the ParentValue field before data can be entered into this field.

N. ParentValue – The list of values the parent field can have before this field can be available for data entry.

O. ChangeFromV2_06 – A description of the changes that were made to the field between data specification versions 2.06 and 2.07.

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 STS data previously collected and maintained in other software products or data versions. (See “Data Import”, below).

   f. The user’s data must be accessible for ad hoc queries either through the software package or by common third party software (e.g. Microsoft Access, Crystal Reports, etc.) If the data is not directly accessible, then the software must provide the ability for the user to export the data in a standard file format which can be queried using common third party query software. (See “Data Export for Analysis by Users”, below). When users are querying their data, grouping records that were created under multiple data version numbers must be invisible to the user. For example, if a user wants to analyze the average patient age in their data for a time period of two years, the fact that their data was recorded under two different version numbers during that period must not require

any additional steps for the user to build the query. We strongly recommend ensuring this by keeping all data in one database regardless of the version number. This requirement is the result of feedback from many frustrated users.

g. Users must be able to select specific records in their database via key fields including patient's name, medical record number, 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.

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. Patient identification number (PatID): The PatID field contains a unique, arbitrary number 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 number, once assigned to a patient, can not be edited or reused if the patient records are ever deleted. In order to avoid issues of patient confidentiality, the PatID value should not be any known identifier such as Social Security Number or Medical Record Number.

b. Demographics Data Version Number (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 operation for a patient. On each Operations record, there are four key fields used for record management:

a. Record identification number (RecordID): The RecordID field contains a unique, arbitrary number 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 can not be edited or reused if the record is ever deleted. In order to avoid issues of patient confidentiality, the RecordID value should not be any known identifier such as Social Security Number or Medical Record Number.

b. Patient identification number (PatID): The PatID field contains the associated 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 identification number (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 “Categorical values specified by user” under the Data Source description in the “Explanation of Data Specification Terms”, 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. Data Version Number (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.

In general, 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. The exceptions to this rule are the data version 1.31 and 2.061. As described in the Data Version Numbers section of the Data Specifications portion of this document above, commercial vendor software packages will handle these records using the version 2.07 rules.

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 export of data to the warehouse, most data fields have a default value, usually null or blank, which indicates that the data is "Missing" (see data specifications). 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 "Valid Data" specified, software must restrict data entries to this set of values. For categorical variables this is expressed as a set of text values; for numerical variables, this is expressed as a numeric range.

d. For all categorical fields, user must select the data value from a pick list of Valid Data values.

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. 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, users must purposely add a value to the list to make it available for selection during data entry. If a user enters a value that is not 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. 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, Diabetes is a parent field to its child Diabetes Control. 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 Diabetes = "Yes" should Diabetes Control be available for data entry. Otherwise, Diabetes Control should be unavailable for data entry.

b. If a parent field contains a "No" value, vendors can choose one of two methods for handling the values in the associated child fields:

   i) set all child field values to Null, or
   ii) set child field values to "No" as is appropriate.

   Note that the STS highly recommends following the first method of setting all child fields to Null.

   Vendors must keep in mind that the first method is required in the export file created for submission to the data warehouse. In other words, regardless of what is in the user's database, the export file must contain Nulls in child fields when the parent is No.

   Also, vendors must notify the STS and the data warehouse if their software will insert No values into child fields when the parent is No. This will allow the warehouse to know that the data received by a site during a data harvest will not look exactly like what the user has in their database.

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 or No depending on the method being used by the vendor as described above. This will avoid the possibility of conflicting information

being left in the data record (for example Diabetes is “No” but Diabetes Control is “Oral”).

5. 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.

    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. <u>This is especially true when importing data that was previously submitted to the data warehouse (i.e. data from another certified software package)</u>. 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. 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.

6. Record Subsets and Queries

    a. Software must allow users to search for Individual records selected by RecordID, PatID or by patient identifiers including

patient name, SSN, medical record #, and surgery date. This is to benefit on site data quality actions and usability of the database.

b. Software should allow groups of records to be selected (e.g. filter function) by multiple fields, which minimally include operation type, surgeon, hospital, date of surgery, date of admission, and date of discharge.

c. Users should be able to name, save, copy and modify record selection criteria.

d. Users should also be able to construct more general queries including field selection, record selection, sorting, and summarizing. It is acceptable if this function is provided by a third party application (e.g. MS Access or Crystal Reports).

7. Reporting

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

a. View and print listing of records (either all records or a selected subset) with basic information such as, but not limited to, RecordID, PatientID, patient name, SSN, operation type, medical record number, date of birth, date of surgery, surgeon, and hospital.

b. Print full record detail, including linked records, on single or multiple selected records.

c. Build, save, copy, and modify more general reports with capability to select fields, record subsets, sorting, and summary statistics. (It is acceptable if this function is provided by a third party application, such as MS Access or Crystal Reports).

d. Incorporate capabilities for graphing the data in reports, including trends over time (it is acceptable if this function is provided by a third party application).

e. 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

f. During the harvest process, participants are required to submit to the warehouse a form listing, for each year included in the data file, the number of patients for each race, the number of patients who were discharged dead, and the total number of operations records included. It is recommended that software developers provide the users with a form that presents this information. However, this form should not try to look exactly like the form used by the warehouse, as the format of the form is subject to change for each harvest.

8. Data Completeness Checking

In the past, the data and software specifications have set requirements stating what action should be taken by the software as part of a data quality check to report missing data. The feedback received from participants has indicated that there is no one set of requirements that will provide all users with the data quality information they will find useful (some specified that there was not enough while others specified there was too much). To address this, the requirement is now only that the software does provide the users with some utility to check and report on the completeness of their data. It is recommended that this utility allow the user to determine which fields are included in the quality control process. They should also be able to use this utility on single records or on a specified group of records. This method will allow users to get data completeness reports that are useful for their own needs.

9. 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 (e.g. text string is stored as a number), the data must be decoded when written to the export file

so that actual values (e.g. full text strings) are contained in 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 build, save, copy, and modify named export configurations.

f. User can control export file naming convention.

10. 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. 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 ...)
```

c. What records are included in the extract file varies depending on the table:
   i) 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.
   ii) For the Procedures table, all records linked to the included Operations records must be included in the export file.
   iii) 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.

d. 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.

e. 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).

f. 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.

g. 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.

h. The values in the harvest file must be the numerical "Harvest Coding" of the data values and not the full text strings.

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

j. The software must create the exported data file using the file naming convention of **XXXXXthr.dat** where “XXXXX” is the 5-digit ParticID for the data contained in the file. The users should not specify the file naming convention. Files not using this naming convention can not be accepted by the automated process at the data warehouse and may be returned to the participant.

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

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

l. 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.

m. 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.

n. On each data record, the fields that are Core and Harvested for the data version specified in the DataVrsn 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.

11. 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 section “f” of “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.

Appendix A: Consistency Edits

The following is a description of the consistency checks that are performed on each participant's data at the data warehouse. These are additional checks that compare variable values for clinical correctness. It is recommended that software developers include these checks in their own quality control routines so that any issues can be resolved before data is submitted to the warehouse. This will also help ensure that the data at the warehouse accurately reflects the data in the participants database.

1. Age
    a. If Age is missing or the current value of Age does not equal the number of months between the DOB and SurgDt divided by 12, then Age is set to this calculated value.
2. MtDate
    a. If MtDate is missing and DischDt is not missing and MtDCStat = Dead, then MtDate is set to equal the DischDt.
3. DischDt
    a. If DischDt is missing and MtDate is not missing and MtDCStat = Dead, then DischDt is set equal to the MtDate.
4. MtDCStat
    a. If MtDCStat is missing and MtDate, AdmitDt, and DischDt all contain valid values and MtDate is between the AdmitDt and DischDt, then MtDCStat is set to "Dead".
    b. If a patient has multiple operations during a single admission (i.e. there are more than one operations record for the patient with the same AdmitDt and DischDt), if any one of those records contains the value of "Dead" in MtDCStat, the value of "Dead" is entered into MtDCStat for any of those records where the value is missing or "Alive".
5. Mt30Stat
    a. If Mt30Stat is missing and MtDate and SurgDt both contain valid values and MtDate is within 30 days of the SurgDt, then Mt30Stat is set to "Dead".

In the following data checks, no updates are made to the existing data because there is no way to determine which of the values is incorrect. Instead, the situation is just reported to the participant through the Data Quality Report.

1. For all records where the surgery date (SurgDt), mortality date (MtDate) and mortality 30-day status all contain valid data, the mortality 30-day status (Mt30Stat) must be Alive if MtDate is greater than 30 days after the SurgDt, or must be Dead if MtDate is less than or equal to 30 days after the SurgDt.

2. The skin incision start time (SIStartT) must be earlier than the skin incision stop time (SIStopT) for all records where the start and stop times are not missing and Multi-Day Operation (MultiDay) is No or missing.
3. The time the patient enters the OR (OREntryT) must be earlier than the time the patient leaves the OR (ORExitT) for all records where where the enter and exit times are not missing and Multi-Day Operation (MultiDay) is No or missing.
4. The time the patient enters the OR (OREntryT) must be earlier than the skin incision start time (SIStartT) for all records where the entry and start times are not missing and Multi-Day Operation (MultiDay) is No or missing.
5. Date values for DOB, AdmitDt, SurgDt, DischDt, and MtDate must be in chronological order. An error is reported if any of the following conditions are true:
    a. SurgDt<AdmitDt
    b. SurgDt>DischDt
    c. MtDate<SurgDt
    d. MtDate<DischDt
    e. AdmitDt>DischDt
    f. DOB>SurgDt