

The HL7 Comprehensive Database

Maintenance

Author: Frank Oemig

|  |  |
| --- | --- |
| email: | HL7@Oemig.de |
| Current Status: | in progress |
| DB-File: | HL7\_92.mdb |
| Date: | 9.01.2017 |

Contents Page

1. Introduction 3

1.1. Conditions for applying the macros 3

2. Extracting information out of the standard 4

2.1. Document “HL7 Extract Macros” 4

2.2. Activities Preparing a new version 4

2.3. Running the macros 4

2.4. Converting to ASCII files 5

2.5. Checking the result 6

2.6. Loading into the Database 6

2.7. Postprocessing 6

3. Using the checking database 7

3.1. Tables 7

3.2. Queries 7

3.3. Importing the data 7

3.4. Checking the data 8

3.5. Manual postprocessing 8

3.6. Uploading the data into the real database 8

4. The “real” HL7 database 10

4.1. Data types (data structures) 10

4.2. Translation to German 10

4.3. Comparison of data elements with the previous version 10

4.4. Comparison of messages with message structures 10

4.5. Generation of HTML files 10

5. Converting the standard into HTML 11

5.1. RTF -> HTML Tool 11

5.2. A database to compute HTML links 11

6. Generating the Product Offerings 13

6.1. Allowed Options 13

6.2. Create Database for the Product 13

6.3. Removing Data 13

6.4. Generation of HTML files 13

6.5. Directory Structure of products 13

7. Appendix A: Document “HL7 Extract Macros.doc” 15

7.1. ExtractMessages 15

7.2. ExtractTables 15

7.3. Tables2Text 15

8. Appendix B: Open Points 16

8.1. Functional Area 16

8.2. Table Items 16

# Introduction

This document is for maintaining the database i.e. extracting the relevant information out of the standard documents and to generate links to the HTML version of the standard.

Therefore this document is for HL7 Inc. use only!

According to the progress made (and will be made) by the Publishing Committee the functionality of the extract macros and the checking database is still under construction. Therefore this document should describe the current status of the work.

## Conditions for applying the macros

The documents must correspond to the current version of the style guide defined by the Publishing Committee!

# Extracting information out of the standard

Besides the formatting macros (style fixer) I provide a separate document to extract the necessary information out of the standard documents: “HL7 Extract Macros.doc”. This separate document allows for an easy access to the macros because they are re­presented like buttons. They aren’t real buttons but they look quite the same except that only double clicks within the area will start the macros.

To extract the relevant information out of the documents a lot of manual steps are required some of which are supported by WinWord macros:

## Document “HL7 Extract Macros”

Additional information is provided in appendix A.

## Activities Preparing a new version

Before running the WinWord macros one has to modify the official documents because some information is not directly available or is wrong.

Because of the fact that the documents follow the official style guide some steps can now be left out.

At least check the information for the following rules:

1. The introductory style for one message structure must be named to „Msg Table Header“! This is the first line. To all subsequent lines the style „Msg Table Body“ must be assigned. This rule is not neccessary any more if the macros of the style fixer are applied.
2. The message tables must contain 3 or 6 columns within all rows! This rule is true for about 98% of the message definitions but causes the extract macro to interrupt. You can check it by marking the complete 1st column (at the top of the column). If in some rows also the second column is highlighted the formatting of the table must be corrected by hand. This is done best by converting it to text and back again with a tabulator character as a delimiter.
3. The heading line must be modified to also contain information about the according event. The latest version of the documents (v2.3.1) already contains this modification. In some places (events “O01”, “O02” and chapter 8) this information must be modified a little bit further. In some cases the third component must be used to specify a separate message structure identifier other than “<message type>\_<event code>”:

<message types> ^ <Event code> ^ <message structure identifier>

1. The movement towards message structure identifiers also allows for a reduction of data. Additional checking for a reuse of message structures is necessary.
2. Since this information is contained in form of a table, too, there is no need to modify the structure information because whether tabs or spaces are used to indent the segments do not matter. The left column is extracted out of the table.
3. Within the left column only valid segment names and some additional characters (like “[“, “]”, “{“, “}”,”<”, “|”,”>” and “?”) are allowed.

## Running the macros

The macros “ExtractTables” and “Tables2Text” creates a set of new documents.

To run the macros the document “HL7 Extract Macros” and the document to extract from should be opened first. The macro “Extract Contents of all Tables to new documents” asks for some additional information first:

A combo box presents the name of all open documents

The pathname where to place the new temporary files

The new version number

The number of the chapter

In a second form the range of tables must be specified.

Then this macro computes a set of new documents:

|  |  |  |
| --- | --- | --- |
| File | Contents | Source Table |
| Event | List of event codes | Message structure table |
| Msg | Message structures for an event | Message structure table |
| MsgStruct | Message structures for a message structure identifier | Message structure table |
| MsgStructID | List of message structure identifier | Message structure table |
| MsgType | List of message types | Message structure table |
| Qry | List of main information for conformance statements | Conformance Statement |
| QryDisplay | List of display lines | Conformance Statement |
| QryInput | List of input parameters | Conformance Statement |
| QryInputComment | List of comments on input parameters | Conformance Statement |
| QryRCP | List of response messages for conformance statements | Conformance Statement |
| Segment | List of segments | Attribute table |
| SegmDat | List of data elements | Attribute table |
| Tab | List of tables | Table table |
| TabVal | List of values for a table | Table table |
| Unknown | all the unknown information | The unidentified tables |

The file “unknown.doc” contains all the tables the information out of which cannot be used elsewhere. Here special attention is necessary. Whenever within this document a table appears which contains relevant information so formatting of the original standard document has to be corrected according to the style guide.

A good idea is to run the macros twice. The first time (on empty or new extract files) to identify and correct possible typos and the second time (into the same set of files) to really extract the information.

## Converting to ASCII files

Finally the computed tables must be converted into plain ASCII text files which can be uploaded into the check database. Therefor the WinWord macro “Tables2Text” converts the tables into single lines with fields delimited by “@”. This macro only allow for converting the files “segmdat.doc”, “tabval.doc” and “QryDisplay.doc”, because the other documents are filled with data in an uploadable format already.

Before saving the files in form of TXT files the lines has to be checked by hand whether original fields also contain end of line delimiters which breaks a logical line into two pieces!

Saving this document to file also requires to them without line breaks. Otherwise additional breaks are inserted which destroy the data.

A good idea is to check the new ASCII files by hand on correct formatting, too.

## Checking the result

When the ASCII files are now loaded into second database “hl7\_check.mdb” additional queries can be applied to identify inconsistent entries. Please see also chapter 3 which explain the checking database a little bit more in detail.

The next table should collect all the checks in general:

|  |  |
| --- | --- |
| Table(s) | Description |
| Segmdat | The same data elements with different descriptions |
| Segmdat | The same data elements with different length |
| Segmdat | The same data elements with different data types |
| Segmdat | The same description with different data elements |
| Msg + Segment | All segments within the message structure are listed in Segment |
| MsgStruct + Segment | All segments within the message structure are listed in Segment |
| Event + Table 0003 | All events are listed in table 0003 and vice versa |
| MsgType + Table 0076 | All message types are listed in table 0076 and vice versa |
| Table + Segmdat | All tables in segmdat are listed in the list of tables |
| Segmdat | Duplicate segment definitions are identical completely. |
| Tab | All table with different descriptions |
| TabVal | All table values with different codes |

The checking queries should result in an empty list. Otherwise some updates onto the data are necessary.

Here you also have to delete all empty records.

## Loading into the Database

Once the data is consistent it can be loaded into the real database. This can be done best by defining append-queries which appends the data to the tables within the database.

## Postprocessing

Finally some manual work is necessary which must be done at the end of the process:

1. Adding the message type for the sending and receiving application to the table “EventMessageTypes”. Here an additional query is helpful which lists all message information without such a reference entry.
2. Check all the defined messages whether they are identical. The corresponding reference information has to be added to the table “EventMessageTypes”. At least not mandatory entries within the table “MsgStructIDs” must be deleted.
3. Define components and subcomponents.
4. Build up the data structure information out of the components. At this point a manual check against chapter 2 must be done.
5. Generate the HTML files. By the help of this process inconsistent usages of brackets within a message description become obvious.
6. Compare the data elements with the ones from the previous version.

At the end all inconsistencies must be eliminated. Referential integrity forces consistency in most cases but not at all. Therefor the queries whose name begin with “xInconsistency” lists the inconsistencies which are not covered by referential integrity.

# Using the checking database

In order to increase the checking possibilities without modifying the contents of the “real” database a second database – called the checking database “hl7\_check.mdb” – is introduced. The extracted data can be uploaded into this database which also contains tables to store the data. This tables exactly look like the ones which are in the real database.

A set of queries allows for checking the integrity of the data with

* Itself and/or
* an arbitrary version within the database.

## Tables

This database contains on the one hand the data extracted out of the documents and references to the real database on the other:

|  |  |  |
| --- | --- | --- |
| Table | Description | Type |
| “\_new” ... | Contains the data expanded out of the documents | Contain data |
| ... | References to the real database: these references must be added by hand | Reference to real database |
| Expand Events | The table “\_new Events” contains event codes which can be compressed like “S01-S12”. This table now provides an extension to these compressed codes. | Contains data |

## Queries

The queries provided fall into six categories:

|  |  |  |
| --- | --- | --- |
| Category | Naming Convention | Description |
| 1 | “Check” ... | Checks the data with itself. The results can be used to identify typos. |
| 2 | “Cross” ... | Checks the data with the data of an arbitrary version within the real database. |
| 3 | “Cross new” ... | Checks the data whether some elements are new. |
| 4 | “Cross old” ... | Checks the data whether data out of the real database is missing. |
| 5 | “x ...” | Additional queries to support the other |
| 6 | “Delete \_new” ... | deletes the complete set of information in the corresponding tables |
| 7 | “xUpload 99” ... | uploads the information into the real database |
| 8 | ... | Queries for supporting reasons. |

The name of the query should describe the purpose of this check.

## Importing the data

The saved ASCII text files from the previous can be used now to be imported into this database.

First of all remove the data from all tables named “\_new ...”!

Next import the data from the ASCII files one by one. Please consider to use “@” as the field delimiter. On account of the correct definition of these fields no further information is necessary.

## Checking the data

Now you can use the queries “Check ...” to check all the new data for consistency!

If one of the queries presents data this is a hint for an inconsistency: You have evaluate the corresponding tables in order to resolve the problem. Whenever a change is necessary this must be documented because it is a change to the standard documents.

At this point I can mention that the efforts of the publishing group take effect: This list of inconsistencies is getting shorter!

## Manual postprocessing

Before an upload of the data into the real HL7 database is made possible some more checks are necessary. These checks must be done by hand because no query can support these tasks.

One task is to check the assignment of messages whether two messages belong together in form of a message sent by the originator of the event and the responder to this event. Therefor the list in “\_new EventMessageType” is too long and must be shortened.

Another task is to check whether some event codes needs an expansion: E.g. “S12-S24, S26” must be expanded to S12, S13, S14, ... S24 and S26. This definition is supported by the additional table. But the corresponding message structure identifiers and their messages need a manual correction. In order to reduce the amount of effort you can make use of the message structure identifier within the message definition: Just add the message structure identifier as the third component within the header:

| Event | Message Type | Message Structure ID | Expand to events: | Chapter |
| --- | --- | --- | --- | --- |
| C01-C08 | CRM | CRM\_C01 | C01, C02, C03, C04, C05, C06, C07, C08 | 7 |
| C09-C12 | CSU | CSU\_C09 | C09, C10, C11, C12 | 7 |
| M01-M06 | MFK  MFN  MFQ  MFR | MFK\_M01  MFN\_M01  MFQ\_M01  MFR\_M01 | M01, M02, M03, M04, M05, M06 | 8 |
| P07, P08 | PEX | PEX\_P07 | P07, P08 | 7 |
| PC1, PC2, PC3 | PPR | PPR\_PC1 | PC1, PC2, PC3 | 12 |
| PC6, PC7, PC8 | PGL | PGL\_PC6 | PC6, PC7, PC8 | 12 |
| PCB, PCC, PCD | PPP | PPP\_PCB | PCB, PCC, PCD | 12 |
| PCG, PCH, PCJ | PPG | PPG\_PCG | PCG, PCH, PCJ | 12 |
| S01-S11 | SRM  SRR | SRM\_S01  SRR\_S01 | S01, S02, S03, S04, S05, S06, S07, S08, S09, S10, S11 | 10 |
| S12-S24, S26 | SIU | SIU\_S12 | S12, S13, S14, S15, S16, S17, S18, S19, S20, S21, S22, S23, S24, S26 | 10 |

## Uploading the data into the real database

This task is supported by queries again. But before executing these queries please make sure that the attached tables reference to correct “real” database!

The queries also require that no duplicate records exist. Otherwise an error during the insertion process will occur. As a result you cannot distinguish between errors on account of duplicate keys and all the other!

| Query | Target table | comment | Current errors |
| --- | --- | --- | --- |
| xUpload 01 Event | Events |  |  |
| xUpload 02 Segments | Segments |  |  |
| xUpload 03 MessageTypes | MessageTypes |  |  |
| xUpload 04 DataTypes | DataTypes |  |  |
| XUpload 05 DataStructures | DataStructures |  |  |
| XUpload 06 Tables | Tables |  |  |
| XUpload 07 DataElements | DataElements |  |  |
| XUpload 08 TableValues | TableValues |  |  |
| XUpload 09 SegmentDataelements | DataElements |  |  |
| XUpload 10 MsgStructIDs | MsgStructID |  |  |
| XUpload 11 MsgStructIDSegments | MSgStructIDSegments |  |  |
| XUpload 12 EventMessageTypes | EventMessageTypes |  |  |
| XUpload 13 EventMessageTypeSegments | EventMessageTypeSegments |  |  |
| xUpload 14 ConfStatements | ConfStatements |  |  |
| xUpload 15 ConfStatemDisplay | ConfStatemDisplay |  |  |
| xUpload 16 ConfStatemInput | ConfStatemInput |  |  |
| xUpload 17 ConfStatemInputComment | ConfStatemInputComment |  |  |
| xUpload 18 ConfStatemRCP | ConfStatemRCP |  |  |

# The “real” HL7 database

Once the data is in the real database, some more manual work is necessary.

## Data types (data structures)

Whenever a data element references a table and uses a data type which is NOT atomic, a new data type (data structure) must be defined. This must be done by hand because there is no table this information can be extracted from.

## Translation to German

## Comparison of data elements with the previous version

## Comparison of messages with message structures

## Generation of HTML files

The generation of HTML files especially for message structures uncovers impaired usage of brackets!

# Converting the standard into HTML

To convert the standard into an HTML form is relative simple. One could do it by saving the documents as a HTML file directly. But doing so one will loose all the numbering information.

The best way keeping this information available is to save the documents as an RTF file out of WinWord.

## RTF -> HTML Tool

This RTF files can now be used as input to a shareware program which can convert every RTF file into an HTML file. It also prepends a contents directory to the top of each file. This is very helpful for creating a file containing links to all information.

Some requirements have to be kept in mind so that this tools runs trough successfully:

If the tools lists some errors when converting the files some titles may be too long: Especially level 4 titles sometimes contain the complete paragraph of information. In such a case the document must be changed to separate the title from the paragraph by an end of line character.

The newly generated files can be used now, but a few manual updates improve the functionality:

1. Each contents directory of all files together make a new HTML file: hl7.html   
   Every link needs some extension: There is only a link within the same document. Therefor the filename must be added. So the reference like “<A HREF = “#HeadingXXX”>” must be modified to “<A HREF = “ChpYY.html#HeadingXXX”>”.
2. At the top of each file links to the other HTML files must be added. Then a user can browse through all files.
3. A lot of information will be generated into the HTML files, but most of the information will not contribute something to contents. This information can be deleted because it decreases the length of file significantly. The next table list some of this information in detail:

|  |  |
| --- | --- |
| Replace | With |
| <a name="\_Ref^?^?^?^?^?^?^?^?^?"></a> |  |
| <a name="\_Toc^?^?^?^?^?^?^?^?^?"></a> |  |
| </p>^a<BR></TD> | </TD> |
| </TR><TR | </TR>^a<TR |
| </p>^a<BR></TH> | </TH> |
| <TD align="left"><p> | <TD align="left"> |
| <TD align="center"><p> | <TD align="center"> |
| Colspan=1 |  |

## A database to compute HTML links

The HL7 database also provides links to the HTML version of the standard. Another database can be used now to compute the necessary set of links.

This database is loaded with the main information contained now within the newly created contents file. Here are also some changes necessary:

|  |  |
| --- | --- |
| Replace | With |
| <HR> |  |
| <hr size=4> |  |
| <UL> |  |
| <LI> |  |
| </UL> |  |
| <H1> |  |
| </H1> |  |
| </A> |  |
| <BR> | Space |
| <A HREF=” |  |
| “> | @ |

Import this new text file into the database. Use the table “Originaldaten” and “@” as the field separator.

After importing the data delete all empty records and all records which do not represent valid data.

Update the reference to the real HL7 database. Make sure that these references points to the current version of the HL7 database:

DataElements

DataStructures

Events

SegmentDataElements

Segments

Tables

Next run the following queries in this order. These queries compute the links to the correct items:

Transfer Data

Set 1 Chapter

Set 2 Main Chapter

Set 3 Data Elements

Set 4 DataItem

Set 5 Event

Set 6 Segments

Set 7 Segments (delete reference to definition)

Finally this database is used to update the chapter and HTL reference information within the HL7 database. The previous listed queries produce a lot wrong entries but on account of updating the information when finding consistent data this doesn´t matter.

UpdateReferences in HL7-DB 1 Events

UpdateReferences in HL7-DB 2 Segments

UpdateReferences in HL7-DB 2b SegmentDataElements

UpdateReferences in HL7-DB 3 DataElements

UpdateReferences in HL7-DB 4 DataStructures

# Generating the Product Offerings

In order to keep the maintenance of this database to a minimum only the master copy of this database is necessary. Every product offered can be generated out of this database.

## Allowed Options

First modify the options (see the user manual) to the desired settings.

## Create Database for the Product

Next create a copy of the master database.

Be sure that the next step is only executed on a copy of the master database!

## Removing Data

Use the form for removing data (FormRemoveVersion). Select the version this database should **not** contain. Press the button “Remove Version”. This function also deletes the information for the preceding or succeeding version.

Repeat this step until the database only contains valid information.

The form, macro and module for removing a version can also be deleted.

## Generation of HTML files

The generated HTML files contains references to other versions if these are also within this database. Therefor the HTML files must be generated separately.

Finally add the files for the buttons (images). The directory with the buttons also contains a file “muster.mic” which can be used by the Microsoft Image Composer to define further buttons for future versions or future functionality.

## Directory Structure of products

The directory structure is as follows:

|  |  |  |
| --- | --- | --- |
| Directory | Version | Contents |
| V2.8.x | 2.8 – v2.8.2 |  |
| Documents |  | Documentation |
| html |  | Generated HTML files |
| Annotations |  |  |
| Images |  |  |
| std28 |  |  |
| std281 |  |  |
| Std282 |  |  |
| PHP-Scripts |  | files for Web-Server |
| All | 2.1 – 2.4 |  |
| Documents |  | Documentation |
| Html |  | Generated HTML files |
| Annotations |  |  |
| Images |  |  |
| std21 |  |  |
| std22 |  |  |
| std23 |  |  |
| std231 |  |  |
| std24 |  |  |
| std25 |  |  |
| std251 |  |  |
| std26 |  |  |
| std27 |  |  |
| std271 |  |  |
| std28 |  |  |
| std281 |  |  |
| Std282 |  |  |
| PHPScripts |  | files for Web-Server |

The main directories represent the complete version with database and HTML files. The subdirectories “html” contain the HTML files only.

# Appendix A: Document “HL7 Extract Macros.doc”

This file contains three VBA-programs for Office97 which can be used to extract the relevant information out of the standard documents. The macros are provided by this extra file which allows for easy access. Hence within the regular text you will find “some things” which look like a button but represent a call to the macros if you double click into this area.

To extract the information you have to start the macros. Afterwards the relevant information is within new WinWord documents but in a form which can be converted by hand in order to upload them into the database:

## ExtractMessages

This macro is a relict of previous days (up to V.2.3). It should extract the message structures out of the documents. Here it relies on the fact that the message structures can be identified by the style “Msg Header”.

I am going to keep this macro because it may be useful when searching for other parts out of the documents. But please be aware that it is not maintained any more.

## ExtractTables

This macros checks the contents of each table and tries to identify it. Once it can decide what it contains – by the help of the format of the upper left cell of the table – it copies the data into the new document and applies simple conversions which will allow for uploading the data.

## Tables2Text

The last macro converts the tables into a format you can save to a file which itself can be uploaded into a database.

# Appendix B: Open Points

This part of this documents should maintain information about open topics. Because this kind of information is not maintained to later versions one can delete it.

## Functional Area

This kind of information is just maintained up to version 2.3.

## Table Items

This information was tried to maintain just for version 2.1.