HAPI TestPanel

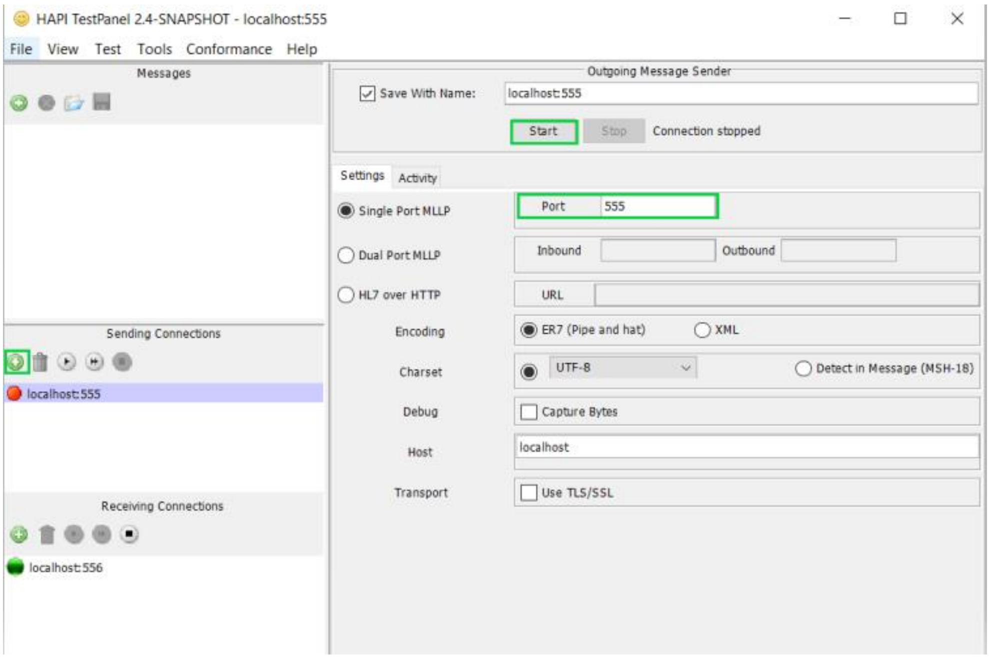
1. What is HAPI

HAPI is an open-source HL7 application programming interface, object-oriented HL7 2.x parser for Java. This page assumes at least a basic familiarity with HL7 and it's terminology, for more information visit the page www.hl7.org. The HAPI TestPanel is a full featured HL7 message editor, transmitter and receiver. TestPanel is free software, distributed under the MPL/GPL. TestPanel is also a demonstration of the many features of HAPI, the best HL7 programming API available for Java.

2. How to use HAPI

2.1 Getting started

Once the application has started, populate it with test data by entering the "Test" menu and clicking "Populate TestPanel with Sample Message and Connections". This creates a receiving connection (a connection which receives messages) on a randomly assigned port. It also creates a sending connection (a connection which sends messages) to that same port, so that you can send a message to yourself. It then creates a sample message for you to send. To send a message to yourself choose the receiving connection by clicking on it's name (near the bottom left of the window). When the settings dialog opens, click "start". Then, choose the message to send by clicking on the message in the message list (near the top left of the window). Now, click send. To send a message to a specific destination select the green button on the left panel. Select the port number and start the connection.



HAPI TestPanel 2.4-SNAPSHOT - localhost:555

☑ Save With Name: localhost:555

Connection stopped

● Single Port MLLP — Port: 555

○ Dual Port MLLP — Inbound: — Outbound:

○ HL7 over HTTP — URL:

Encoding: ● ER7 (Pipe and hat) ○ XML

Charset: ● UTF-8 ○ Detect in Message (MSH-18)

Debug: □ Capture Bytes

Host: localhost

Transport: □ Use TLS/SSL

Sending Connections: localhost:555

Receiving Connections: localhost:556

2.2 Message validation

The HAPI library enables us to build different types of message validation functionality. They are basic message validation, custom message validation and validation by the use of conformance profiles.

The validation performs basic data type checks, such as making sure that NM datatype fields do not contain text, TS datatype fields contain valid dates, etc. To enable or disable this feature, click on the "Validate" selector, and choose either "No Profile/Validation" or "Default HAPI Validation". When validation is enabled, you will see any errors in the column to the left of the value column. You can test this out by enabling validation and changing the MSH-7 (Timestamp) value so that it contains letters within the timestamp. Hovering over the red "X" will give more information about what is wrong.

The basic message validation functionality that HAPI offers by default, ensures that any HL7 message that we are either sending or receiving conforms to some basic rules around both length as well as optionality permitted around things such as segment groups, segments and primitive data types as specified by the HL7 message standard. However, this alone will not be sufficient as no two environments are ever the same and always require site specific customizations. Specific workflows in many healthcare environments always require additional data constraints to be in place, and these constraints need to be applied over and beyond what the HL7 2.x standard specifies. This is where custom message validation and conformance profiles come in. Custom message validation is basically an enhancement to the basic validation offered by HAPI, and is simply enabled through the extensibility offered by the message validation framework itself. It helps us bolt-on any additional rules for validation during message processing by overriding the default behaviour offered by message validation-related classes available within the library. They can for example help constrict the definition of a message specification in a manner that removes the optionality constructs and the general rules around the processing of a HL7 2.x message within a specific clinical workflow. The parties involved here can then codify these rules into their message exchange systems. Conformance profiles, known as "Message Profiles" in HL7 2.x allow sharing of any stringent message specifications involving specific workflows to an open registry for everyone to use.

2.3 Conformance generator

Conformance profiles are special XML-based files which constrain the standard HL7 message definitions, allowing you to specify maximum field lengths, required or optional fields, cardinalities, etc. This is useful both to constrain/remove unwanted elements from a message type, as well as to extend a message type by adding custom segments.

2.3.1 Add a profile group

TestPanel arranges profiles in what it calls a "Profile Group". A profile group is a collection of one or more conformance profiles. Within a profile group, each profile is marked as applying to specific message types, so that for instance you could create a profile group with one profile for ADT^A01 messages and another profile for ORU^R01. To create a profile group, choose "Profiles and Tables..." from the Conformance menu. This opens the profiles dialog. Click on "New Profile Group" to create a new profile group.

Select the newly created group, and click on "Add Profile" to add a conformance profile to the group. The allows you to select an XML based conformance profile, and then assign it to one or more message types. To apply your new profile group to a message (or a collection of messages), open the messages in the main message editor, and choose your new profile group from the "Validate" picker, near the top right of the window. If there are any problems with your message, you will now see them highlighted.

2.3.2 How to create a message profile

Message profiles are a precise method of documenting message constraints, using a standard XML syntax defined by HL7. The static definition is an exhaustive specification for a single message. The static definition is based on a message structure defined in the HL7 Standard. The message code, trigger event, event description, role (Sender or Receiver) and, if applicable, the order control code will be provided. A complete static definition shall be defined at the message, segment, and field levels. A static definition is compliant in all aspects with the HL7-defined message it profiles. However, the static definition may define additional constraints on the standard HL7 message. A static definition explicitly defines: Segments, segment groups, fields and components usage rules, cardinalities, length information, value sets and coding systems. To build a segment use the tag <Segment> and inside segments are the fields with the tag <Field>, inside fields are the sub-fields with the tag <Component> and inside that ones are the sub-sub-fields with tag <SubComponent>. Each one of these elements can have different types of attributes, below are defined the possible attribute for each Xml element when building a conformance profile.

Static definition attributes

Attribute	Meaning
msgType	Message type code, Refer to HL7 Table 0076
eventType	Event type, Refer to HL7 Table 0003
msgStructID	Message structure, Refer to HL7 Table 0354
orderControl	Order control code, Refer to HL7 Table 0119
eventDesc	Event description
identifier	Each static definition must have a unique identifier when registered
role	Sender or Receiver

Segment attributes

Attribute	Meaning
impNote	Some important notes
description	Description about the element or restriction
reference	Reference to the section in the HL7 standards
predicate	Elements assigned with a declared conditional usage have an associated predicate that is computable and based on other elements in the message to provide specifiers with a simple language that affords consistency within and across HL7 v2 implementation guides, the readability of a natural language, and be machine computable.
name	Name of the element
longName	Complete name of the segment (e.g. MSH segment has Message header as longName)
usage	Determines whether the element must be present, can be present, or must not be present in a message instance for the sender
min	Minimum number of repetitions
max	Maximum number of repetitions

Field/sub-field/sub-sub-field attributes

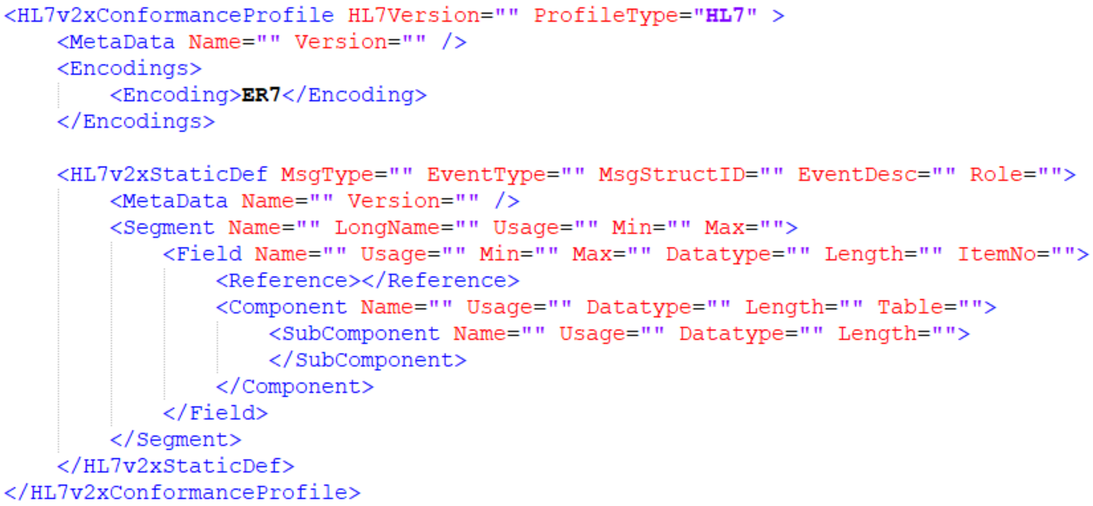
Attribute	Meaning
impNote	Some important notes
description	Description about

reference	Reference to the section in the HL7 standards
predicate	Elements assigned with a declared conditional usage have an associated predicate that is computable and based on other elements in the message to provide specifiers with a simple language that affords consistency within and across HL7 v2 implementation guides, the readability of a natural language, and be machine computable.
name	Name of the field
longName	Complete name of the field
usage	Determines whether the element must be present, can be present, or must not be present in a message instance for the sender
dataType	Data type of the value
length	Maximum length of the value
constantValue	Constant value of the field according HL7 specifications
table	Code of the HL7 table according to standards

Field Attributes

Attribute	**Meaning**
itemNo	Identifies the field occupying in the segment
min	Minimum number of repetitions of the field
max	Maximum number of repetitions of the field

Message profile template:



```
<HL7v2xConformanceProfile HL7Version="" ProfileType="HL7" >
    <MetaData Name="" Version="" />
    <Encodings>
        <Encoding>ER7</Encoding>
    </Encodings>

    <HL7v2xStaticDef MsgType="" EventType="" MsgStructID="" EventDesc="" Role="">
        <MetaData Name="" Version="" />
        <Segment Name="" LongName="" Usage="" Min="" Max="">
            <Field Name="" Usage="" Min="" Max="" Datatype="" Length="" ItemNo="">
                <Reference></Reference>
                <Component Name="" Usage="" Datatype="" Length="" Table="">
                    <SubComponent Name="" Usage="" Datatype="" Length="">
                    </SubComponent>
                </Component>
            </Field>
        </Segment>
    </HL7v2xStaticDef>
</HL7v2xConformanceProfile>
```

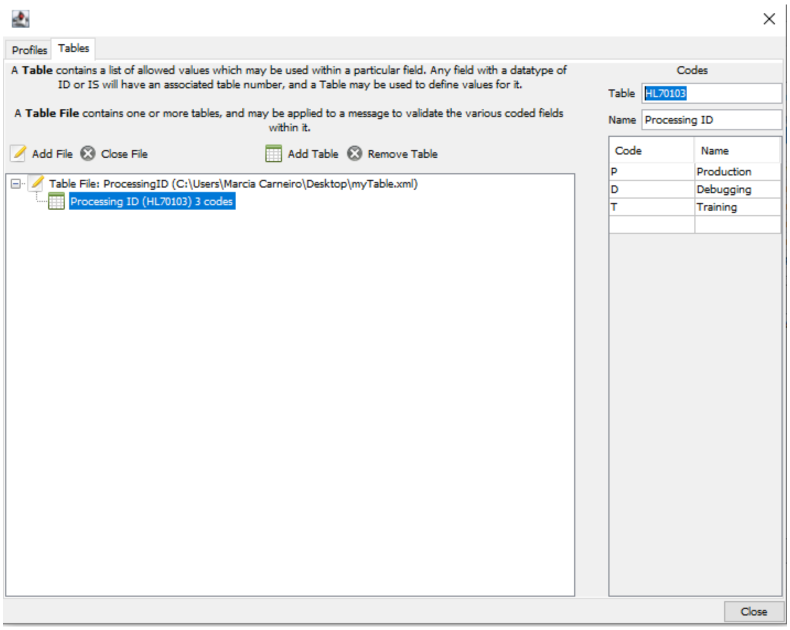
For more information on how to build message profiles according to HL7 standards, visit the page http://www.hl7.eu/refactored/profile.html.

2.3.2 Tables

Tables are stored in table files, and a table file contains one or more tables. TestPanel allows you to define tables from which coded values are drawn. To create a table file, choose "*Profiles and Tables...*" from the *Conformance* menu, and navigate to the *Tables* tab. Click on "*Add File*" and create a new file. The Xml file added from the path must have the following tags:

```
<tableFile>
   <id> id of the file</id>
</tableFile>
```

Now, create a new table. HL7 tables are generally represented by four digits. For example, component 1 of a CE such as MSH-11 "Processing Id" is table 0103. To create a validation table, add the prefix HL7, as shown in the screenshot.



A **Table** contains a list of allowed values which may be used within a particular field. Any field with a datatype of ID or IS will have an associated table number, and a Table may be used to define values for it.

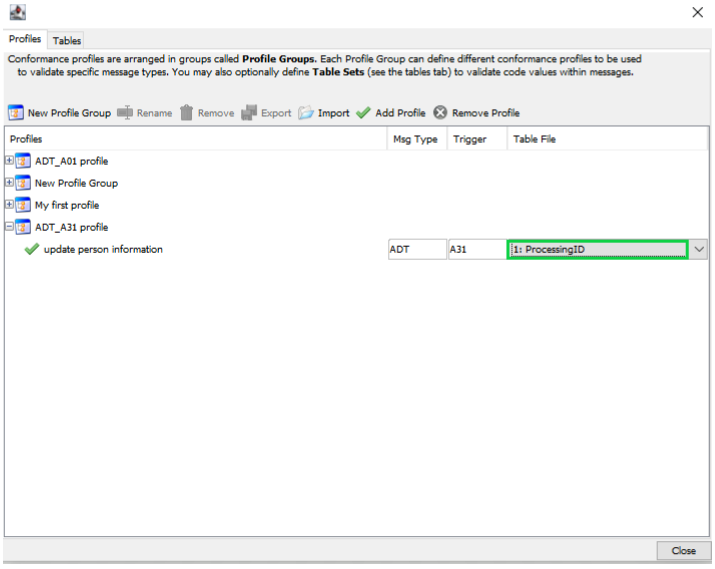
A **Table File** contains one or more tables, and may be applied to a message to validate the various coded fields within it.

Table File: ProcessingID (C:\Users\Marcia Carneiro\Desktop\myTable.xml)
Processing ID (HL70103) 3 codes

Table: HL70103
Name: Processing ID

Code	Name
P	Production
D	Debugging
T	Training

Once you have created your tables, return to the *Profiles* tab, and assign your table file to the Profile Group by clicking on the Table file column and selecting the created table.



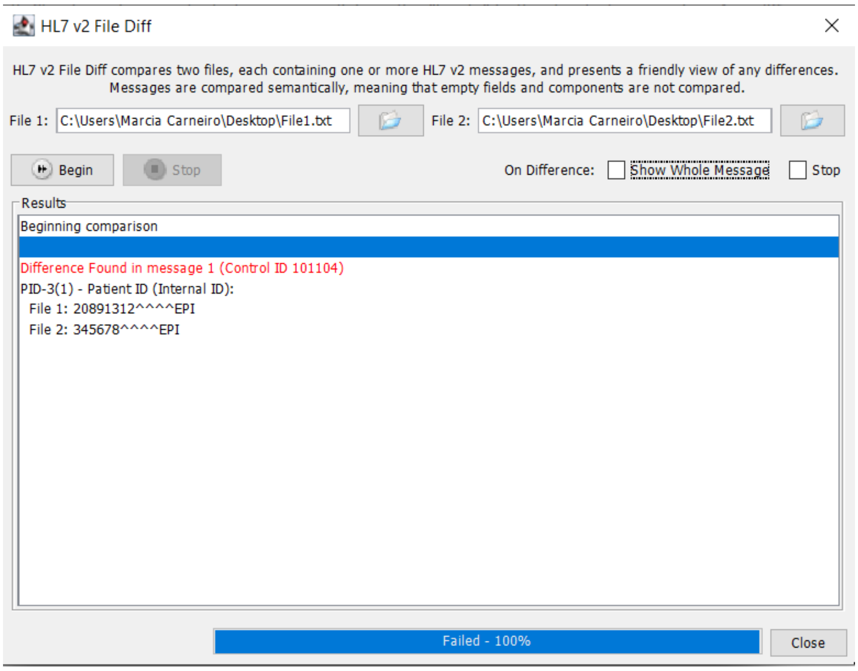
Conformance profiles are arranged in groups called **Profile Groups.** Each Profile Group can define different conformance profiles to be used to validate specific message types. You may also optionally define **Table Sets** (see the tables tab) to validate code values within messages.

Profiles	Msg Type	Trigger	Table File
ADT_A01 profile			
New Profile Group			
My first profile			
ADT_A31 profile			
✓ update person information	ADT	A31	1: ProcessingID

3. Compare HL7 message files

The HL7 v2 File Diff can be used to compare two files, each containing raw HL7 messages, for differences. Each message is parsed, and the parsed message is then compared at a segment-by-segment and field-by-field level. This means that messages are compared to see if they are semantically identical as opposed to performing a rigid byte level comparison.

To access the file diff, start the Test Panel, and choose "HL7 v2 File Diff" from the tools menu. This will bring up the diff dialog. In this dialog, two files (file 1, and file 2) must be selected. These files must contain raw HL7 messages. Note that any lines beginning with hash (#) will be treated as comments and ignored. When both files have been selected, you may click on the "begin" button to start comparing. Any differences found will be highlighted.



HL7 v2 File Diff compares two files, each containing one or more HL7 v2 messages, and presents a friendly view of any differences. Messages are compared semantically, meaning that empty fields and components are not compared.

File 1: C:\Users\Marcia Carneiro\Desktop\File1.txt

File 2: C:\Users\Marcia Carneiro\Desktop\File2.txt

On Difference: □ Show Whole Message □ Stop

```
Beginning comparison

Difference Found in message 1 (Control ID 101104)
PID-3(1) - Patient ID (Internal ID):
  File 1: 20891312^^^^EPI
  File 2: 345678^^^^EPI
```

Failed - 100%

4. Helpfull links

https://saravanansubramanian.com/hl7tutorials/

https://www.ihe.net/

http://www.hl7.org/