Oracle SOA Suite 11g R1 PS5

SOA Suite for healthcare integration Series

Creating a Canonical HL7 v2 Message Model

michael@czapski.id.au

September 2012

Table of Contents

Introduction

Each different domain, covered by HL7 v2 standard, has a set of message definitions which support message exchanges for a particular domain. Most of the message definitions for a domain share certain segments. Many of the segments are optional and perhaps not used in a particular messaging environment.

In any but the simplest of HL7 messaging environments there will be multiple sources and multiple destinations of HL7 messages. It is very unlikely that all, or even a majority of these, will use exactly the same HL7 message structures in terms of versions, optional/mandatory segments, extension Z segments, and so on.

It is common for integrating specialists, if the tooling which they use permits, to develop one or few generalised message structures which can be used to represent more than one distinct message from a domain they work with. These are typically called canonical models. The overriding purpose is to standardise the message payload being passed around between integration components, enabling reuse and reducing complexity. The Canonical Message model (CMM) works hand-in-glove with the enterprise architecture in which transformation to/from the CMM is performed at the edges of the integration domain, standardizing as much as possible, payload structure within the integration domain.

This article works through the mechanics of deriving a Canonical Message Model for the series of articles on SOA Suite for healthcare integration using the "Oracle SOA Suite for healthcare integration" tooling. It contains an abridged version of the article "Healthcare Enterprise – IT Architecture Building Blocks – Canonical Message Model for a HL7 Enterprise", available at http://blogs.czapski.id.au/2010/10/healthcare-enterprise-%e2%80%93-it-architecture-building-blocks-canonical-message-model-for-a-hl7-enterprise, but adds new material related to testing the canonical structure against sample data and modifying the structure to accommodate data idiosyncrasies.

Brief Rationale

In any but the simplest of HL7 messaging environments there will be multiple sources and multiple destinations of HL7 messages. It is very unlikely that all, or even a majority of these, will use exactly the same HL7 message structures in terms of versions, optional/mandatory segments, extension Z segments, and so on. These differences necessitate transformation of messages from/to formats used by source/target systems, in extreme cases leading to the virtual point-to-point integration model, negating most benefits of having a modern integration infrastructure.

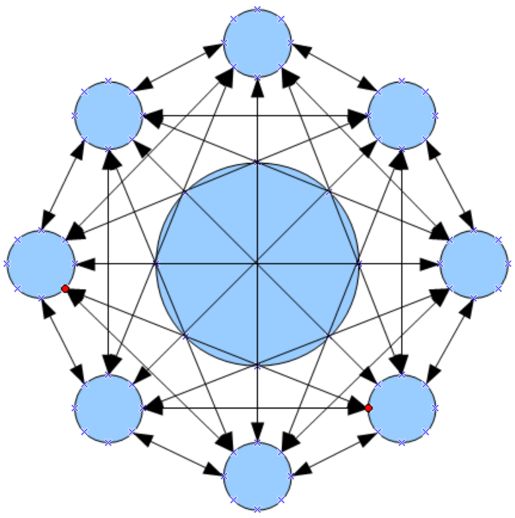


Illustration 1: Point-to-point message transformation

A sensible approach to dealing with these kinds of issues, and a key component of the Enterprise Architecture, is Canonical Message Model (CMM).

The Canonical Message Model works hand-in-glove with the enterprise architecture in which transformation to/from the CMM is performed at the edges of the integration domain. This ultimately leads to the reduction in the number of transformations to the number of external touch points. This also tightly couples endpoints and their transformations, localising change domains and insulating the rest of the enterprise integration infrastructure from the changes in endpoints.

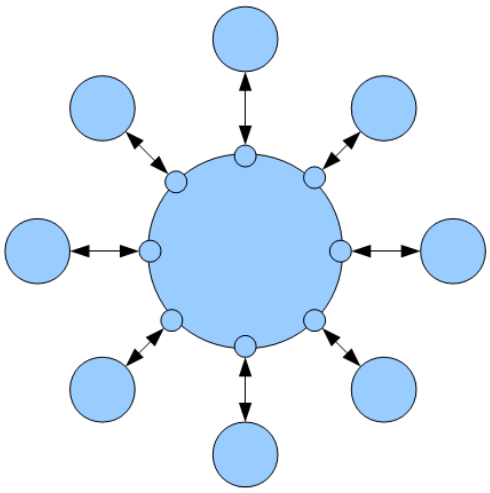


Illustration 2: Using Canonical Message Model

As a consequence of this design integration solutions need only be concerned with message routing.

To accommodate all possible data of interest to the enterprise the CMM will have to be constructed in such a way that it is inclusive of all data.

It is likely that an enterprise will have multiple business domains, for example Patient Administration, Patient Billing, Catering, and so on, which while dealing with patient information in some manner are so different in nature that it is impractical to attempt to derive a canonical message model that would include data from all. This may lead to multiple canonical message models, one for each business domain. This is a valid approach and likely to be more robust then attempting to derive a single model for all business domains in the enterprise.

For the remainder of this article let's assume that the following architectural decisions were made:

1. A canonical message model will be developed to support all HL7-capable clinical message sources and destinations in the enterprise
2. The canonical message model will be based on the HL7 version 2.3.1 standard
3. The canonical message model will include all HL7 v2.3.1 segments that occur in any of the messages provided by source systems or required by destination systems
4. The canonical message model will use the XML representation for HL7 v2.3.1 messages
5. Transformation between endpoint-specific HL7 messages and canonical messages will take place at the edge of the integration domain (at the endpoint)

For the remainder of this article let's assume that the following major technical decisions were made:

1. Oracle “SOA Suite for healthcare integration” will be used to implement the canonical message model and the enterprise integration infrastructure which will use it
2. Oracle “SOA Suite for healthcare integration” functionality will be used to integrated source and destination systems and transform HL7 v2.x delimited messages into their XML equivalents
3. Mediator components will be used to transform HL7 v2.x XML messages into canonical message model messages

Implied technical decisions are:

1. Oracle Document Editor will be used to develop and maintain the common message model artefacts
2. Oracle "SOA Suite for healthcare integration" runtime will be configured to:
3. Perform transparent transformation between HL7 v2.x delimited and their corresponding HL7 v2.x XML formats
4. Handle functional acknowledgements
5. Handle payload persistence and message tracking
6. Handle KPI collection and display for all HL7 endpoints
7. Handle broadcast of a single message to multiple receivers

Modelling the Canonical Message

Let's assume that two systems in the enterprise use the following HL7 messages:

Type and Trigger	*HL7 Version*	*Description*
ADT A01	V2.3.1	Admin/Visit Notification
ADT A03	V2.3.1	Discharge/End Visit
ADT A08	V2.3.1	Update Patient Information
ADT A17	V2.3.1	Swap Patients

Table 1 HL7 v2.x Delimited messages in the Enterprise

There might be (and very likely will be) many more systems and many more messages. This example illustrates a method which will hold for any number of systems and messages.

With the objective of deriving a single message structure that will include all segments in the corresponding HL7 v2.3.1 messages let's prepare a table with Message and Event Types as columns and HL7 v 2.3.1 segments as rows, with assistive symbology where "?" means 0 or 1 (optional), "*" means 0 or more (optional repeating) and "-" means no such segment in this event type. No special symbol means "required".

	A01	*A03*	*A08*	*A17*
1	MSH	MSH	MSH	MSH
2	EVN	EVN	EVN	EVN
3	PID	PID	PID	PID
4	PD1?	PD1?	PD1?	PD1?
5	NK1*	-	NK1*	-
6	PV1	PV1	PV1	PV1
7	PV2?	PV2?	PV2?	PV2?
8	DB1*	DB1*	DB1*	DB1*
9	OBX*	-	OBX*	OBX*
10	AL1*	-	AL1*	-
11	DG1*	DG1*	DG1*	-
12	DRG?	DRG?	DRG?	-
Group PROCEDURE*				
13A	PR1	PR1	PR1	-
13B	ROL*	ROL*	ROL*	-
13C	GT1*	-	GT1*	-
Group INSURANCE*				
14A	IN1	-	IN1	-

14B	IN2?	-	IN2?	-
14C	IN3*	-	IN3*	-
15	ACC?	-	ACC?	-
16	UB1?	-	UB1?	-
17	UB2?	-	UB2?	-
18	-	OBX*	-	-
19	-	-	-	PID
20	-	-	-	PD1?
21	-	-	-	PV1
22	-	-	-	PV2?
23	-	-	-	DB1*
24	-	-	-	OBX*
25	-	Z01?	Z01?	-

Table 2: Segment-level comparison

Inspection of columns A01 and A08 reveals that the same segments are used in both and that they are a large superset of the segments used in other event types. To construct a structure that would cater for A01, A08 and A03 requires addition of an optional, repeating OBX segment (row 18). To construct a structure that would also cater for A17 would require addition of PID, PD1, PV1, PV2, DB1 and OBX segments. The PID and PV1 segments, which in A17 are required, would have to be made optional to allow A01, A03 and A08 to be correctly parsed. This would mean that a malformed A17 with a missing PID or PV1 segment would be accepted as valid. We are prepared to live with that for the sake of the benefits of a canonical message model.

One of our systems sends messages with a custom Z01 segment. This segment's structure is shown in Table 3.

Sequence	*Length*	*Data Type*	*Element Name*
1	1	IS	Original Gender
2	1	IS	Current Gender
3	26	TS	Date of Change
4	1	IS	Legal Gender

Table 3: Z01 Segment composition

The segment is optional, since not very many people change their gender, but all components are required, since when they do all the required pieces of information are known.

Knowledge of our systems tells us that no system ever sends segments other than MSH, EVN, PID and PV1, and Z01. This is a grossly simplifying statement. It is unlikely that this will be the case in real systems but for the sake of brevity in this article it is a reasonable simplification. The method will hold no matter which and how many segments are included. It is likely, in fact, that an enterprise would include all optional segments in the canonical message model if for no other reason than to ensure that a system that one day gets upgraded, and starts including one of the optional segments, does not break the model.

With this simplification, however, our message will consist of HL7 segments listed in Table 3.

CMM
MSH
EVN

PID
PV1
PID?
PV1?
Z01?

Table 4: Canonical Message Model HL7 Segments

Analysis aimed at deriving a canonical message model would then proceed deeper to inspect fields, components and subcomponents, and determine whether and what might need to be done to restrict, relax or modify rules that might apply to them, and include or exclude any in/from the model. We will stop at the segment level for the purpose of this article.

Building the CMM Structure

We will use the Oracle Document Editor to build the CMM structure and produce the external forms of it for use in our integration work.

In our example A01 and A08 both have the same set of segments, rows 1 through 17 in Table 2. We will need to add an optional repeating OBX segment to account for the requirements of A03 (row 18 in Table 2). This will address standard structures for A01, A03 and A08. To account for the requirement of the A17 we need to add PID, PD1, PV1, PV2, DB1 and OBX segments, shown in rows 19 through 24 in Table 2. Finally, we will add the custom Z01 segment.

- ☐ Start the Oracle Document Editor.
- ☐ Click New Document. Expand HL7 → 2.3.1 → Event A01: ADT/ACK and select ADT: ADT message node.

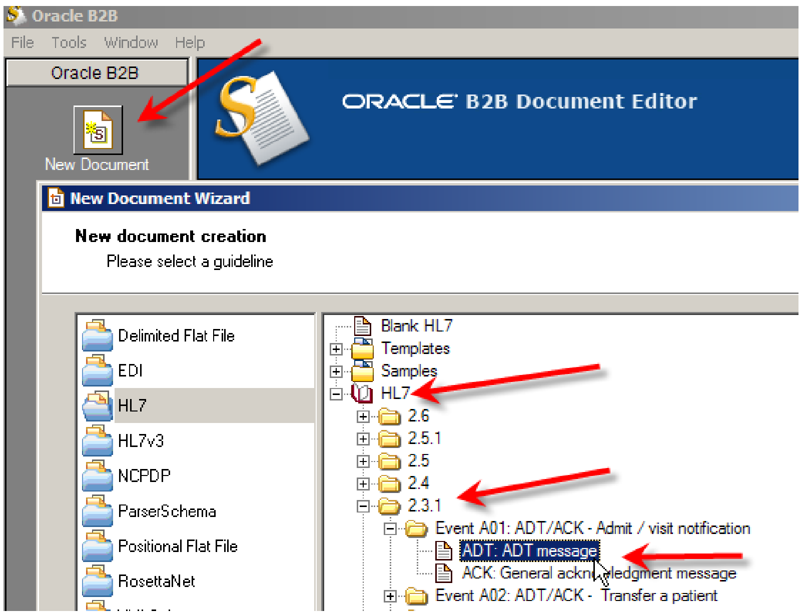
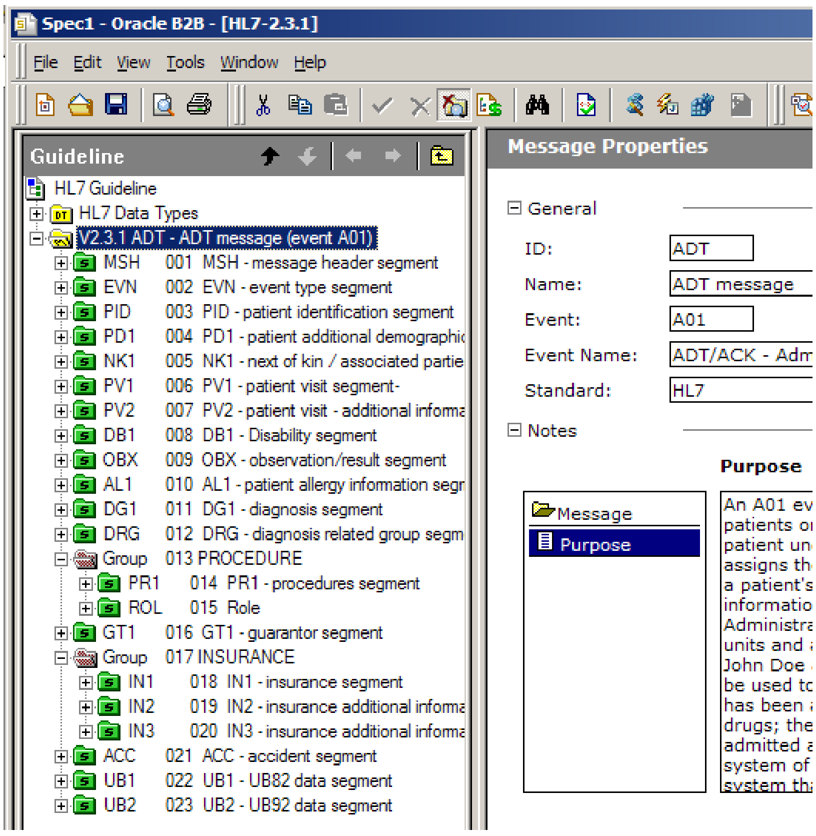


Illustration 3: Choose standard ADT A01 message to modify

The Spec / Guideline has all the segments needed for A01 (because this is A01) and A08 (because both have the same structure).


Spec1 - Oracle B2B - [HL7-2.3.1]
File Edit View Tools Window Help
Guideline
Message Properties

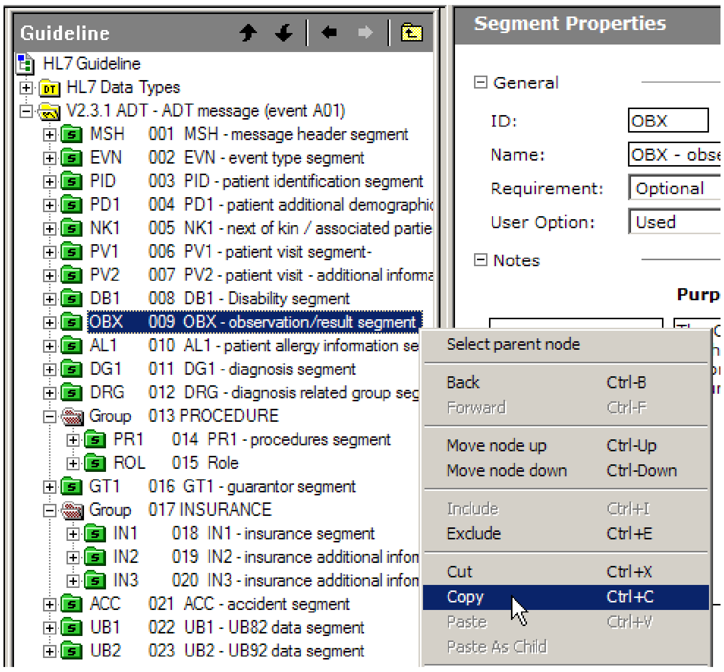
```
HL7 Guideline
HL7 Data Types
V2.3.1 ADT - ADT message (event A01)
  MSH    001  MSH - message header segment
  EVN    002  EVN - event type segment
  PID    003  PID - patient identification segment
  PD1    004  PD1 - patient additional demographic
  NK1    005  NK1 - next of kin / associated partie
  PV1    006  PV1 - patient visit segment-
  PV2    007  PV2 - patient visit - additional informa
  DB1    008  DB1 - Disability segment
  OBX    009  OBX - observation/result segment
  AL1    010  AL1 - patient allergy information segr
  DG1    011  DG1 - diagnosis segment
  DRG    012  DRG - diagnosis related group segm
  Group  013 PROCEDURE
    PR1    014  PR1 - procedures segment
    ROL    015  Role
  GT1    016  GT1 - guarantor segment
  Group  017 INSURANCE
    IN1    018  IN1 - insurance segment
    IN2    019  IN2 - insurance additional informa
    IN3    020  IN3 - insurance additional informa
  ACC    021  ACC - accident segment
  UB1    022  UB1 - UB82 data segment
  UB2    023  UB2 - UB92 data segment
```

```
General
  ID:           ADT
  Name:         ADT message
  Event:        A01
  Event Name:   ADT/ACK - Adm
  Standard:     HL7
Notes
                Purpose
  Message       An A01 ev
  Purpose       patients oı
                patient un
                assigns th
                a patient's
                informatio
                Administra
                units and
                John Doe
                be used tc
                has been
                drugs; the
                admitted a
                system of
                system th
```

Illustration 4: Standard A01 structure

Let's now add an optional, repeating OBX segment to the end of the structure (append) to accommodate the requirement of the A03 structure.

☐ Right-click an OBX segment in the A01 message and choose Copy.

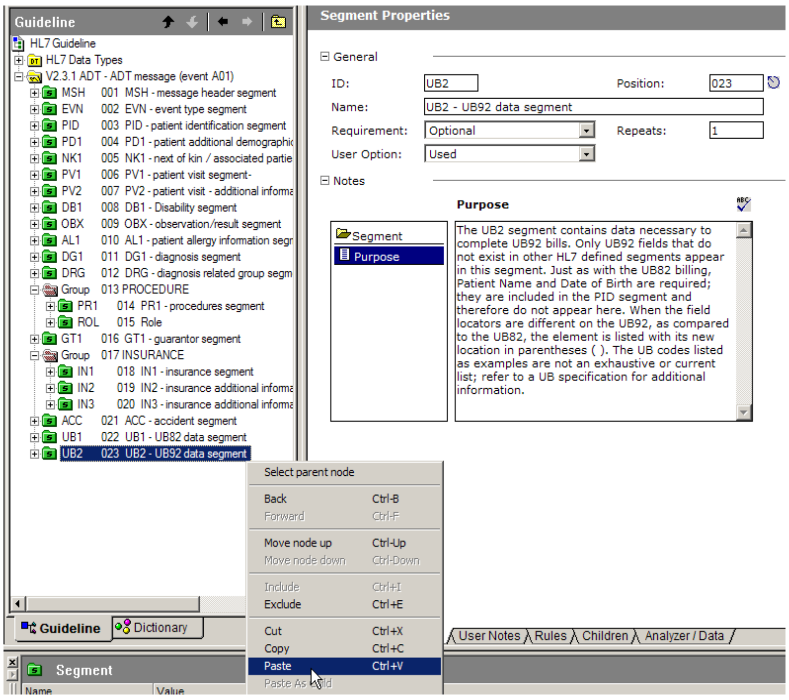


```
HL7 Guideline
HL7 Data Types
V2.3.1 ADT - ADT message (event A01)
  MSH    001  MSH - message header segment
  EVN    002  EVN - event type segment
  PID    003  PID - patient identification segment
  PD1    004  PD1 - patient additional demographic
  NK1    005  NK1 - next of kin / associated partie
  PV1    006  PV1 - patient visit segment-
  PV2    007  PV2 - patient visit - additional informa
  DB1    008  DB1 - Disability segment
  OBX    009  OBX - observation/result segment
  AL1    010  AL1 - patient allergy information se
  DG1    011  DG1 - diagnosis segment
  DRG    012  DRG - diagnosis related group seg
  Group  013 PROCEDURE
    PR1    014  PR1 - procedures segment
    ROL    015  Role
  GT1    016  GT1 - guarantor segment
  Group  017 INSURANCE
    IN1    018  IN1 - insurance segment
    IN2    019  IN2 - insurance additional infor
    IN3    020  IN3 - insurance additional infor
  ACC    021  ACC - accident segment
  UB1    022  UB1 - UB82 data segment
  UB2    023  UB2 - UB92 data segment
```

```
General
  ID:            OBX
  Name:          OBX - obse
  Requirement:   Optional
  User Option:   Used
Notes
                 Purp
```

Illustration 5: Copy OBX segment

☐ Right-click on the last node of the structure (UB2) and choose “Paste”.



```
Guideline
HL7 Guideline
HL7 Data Types
V2.3.1 ADT - ADT message (event A01)
  MSH   001 MSH - message header segment
  EVN   002 EVN - event type segment
  PID   003 PID - patient identification segment
  PD1   004 PD1 - patient additional demographic
  NK1   005 NK1 - next of kin / associated partie
  PV1   006 PV1 - patient visit segment-
  PV2   007 PV2 - patient visit - additional informa
  DB1   008 DB1 - Disability segment
  OBX   009 OBX - observation/result segment
  AL1   010 AL1 - patient allergy information segr
  DG1   011 DG1 - diagnosis segment
  DRG   012 DRG - diagnosis related group segm
  Group 013 PROCEDURE
    PR1   014 PR1 - procedures segment
    ROL   015 Role
  GT1   016 GT1 - guarantor segment
  Group 017 INSURANCE
    IN1   018 IN1 - insurance segment
    IN2   019 IN2 - insurance additional informa
    IN3   020 IN3 - insurance additional informa
  ACC   021 ACC - accident segment
  UB1   022 UB1 - UB82 data segment
  UB2   023 UB2 - UB92 data segment
```

Segment Properties

General

ID: UB2 Position: 023

Name: UB2 - UB92 data segment

Requirement: Optional Repeats: 1

User Option: Used

Notes

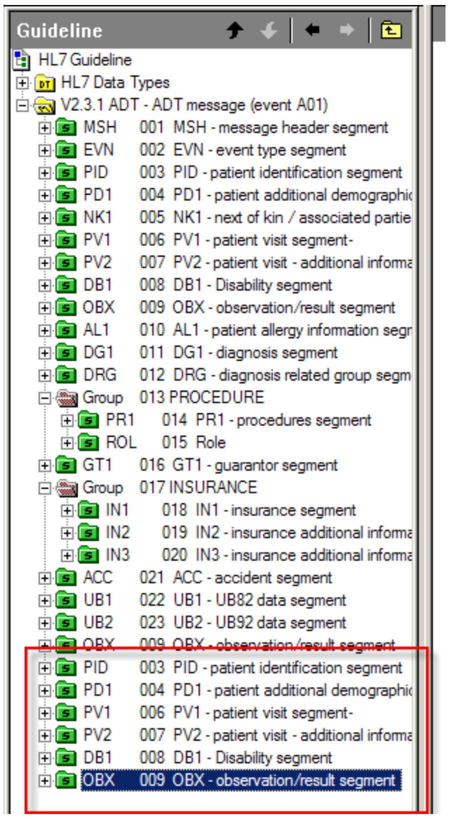
Purpose

The UB2 segment contains data necessary to complete UB92 bills. Only UB92 fields that do not exist in other HL7 defined segments appear in this segment. Just as with the UB82 billing, Patient Name and Date of Birth are required; they are included in the PID segment and therefore do not appear here. When the field locators are different on the UB92, as compared to the UB82, the element is listed with its new location in parentheses ( ). The UB codes listed as examples are not an exhaustive or current list; refer to a UB specification for additional information.

Illustration 6: Append OBX to the end of the structure

Now the message reflects the requirements of the standard A01, A03 and A08 messages.

☐ Using the same copy/paste process let's append PID, PD1, PV1, PV2, DB1 and OBX segments.

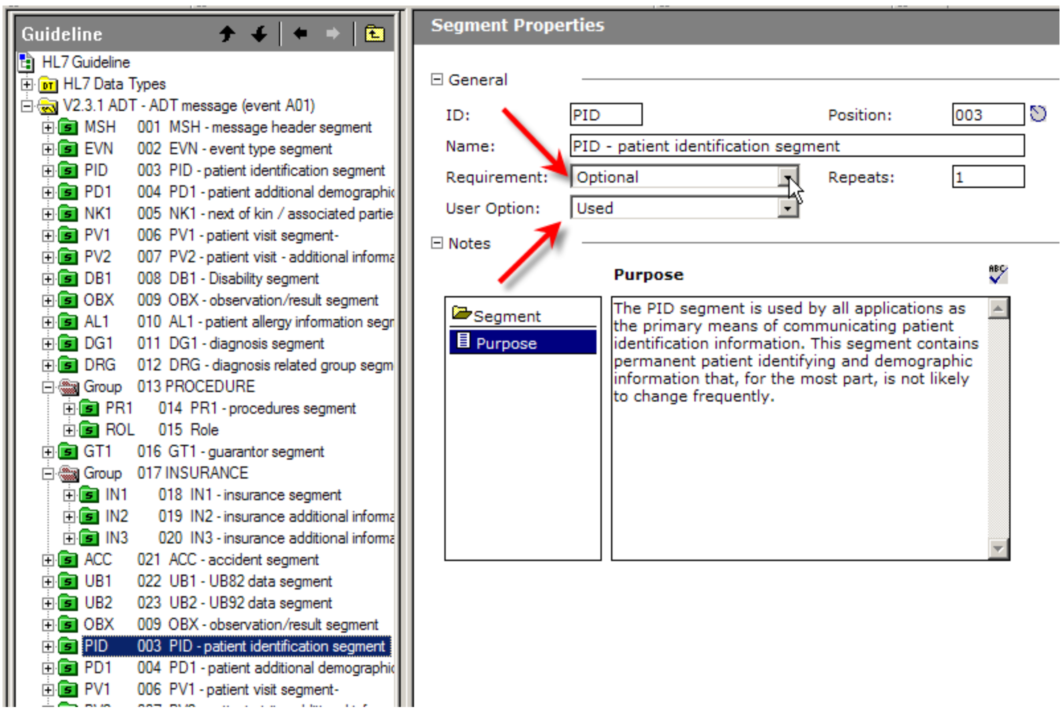


```
Guideline
HL7 Guideline
HL7 Data Types
V2.3.1 ADT - ADT message (event A01)
  MSH   001 MSH - message header segment
  EVN   002 EVN - event type segment
  PID   003 PID - patient identification segment
  PD1   004 PD1 - patient additional demographic
  NK1   005 NK1 - next of kin / associated partie
  PV1   006 PV1 - patient visit segment-
  PV2   007 PV2 - patient visit - additional informa
  DB1   008 DB1 - Disability segment
  OBX   009 OBX - observation/result segment
  AL1   010 AL1 - patient allergy information segr
  DG1   011 DG1 - diagnosis segment
  DRG   012 DRG - diagnosis related group segm
  Group 013 PROCEDURE
    PR1   014 PR1 - procedures segment
    ROL   015 Role
  GT1   016 GT1 - guarantor segment
  Group 017 INSURANCE
    IN1   018 IN1 - insurance segment
    IN2   019 IN2 - insurance additional informa
    IN3   020 IN3 - insurance additional informa
  ACC   021 ACC - accident segment
  UB1   022 UB1 - UB82 data segment
  UB2   023 UB2 - UB92 data segment
  OBX   009 OBX - observation/result segment
  PID   003 PID - patient identification segment
  PD1   004 PD1 - patient additional demographic
  PV1   006 PV1 - patient visit segment-
  PV2   007 PV2 - patient visit - additional informa
  DB1   008 DB1 - Disability segment
  OBX   009 OBX - observation/result segment
```

Illustration 7: Additional segments for A17

☐ The new segments only occur in A17s so let's make sure all are optional. The copied

PID and PV1 are required. Select PID segment and change Required to Optional and Must Use to Used.



Segment Properties

General

ID: PID

Position: 003

Name: PID - patient identification segment

Requirement: Optional

Repeats: 1

User Option: Used

Notes

Purpose

The PID segment is used by all applications as the primary means of communicating patient identification information. This segment contains permanent patient identifying and demographic information that, for the most part, is not likely to change frequently.

Guideline
HL7 Guideline
HL7 Data Types
V2.3.1 ADT - ADT message (event A01)
MSH 001 MSH - message header segment
EVN 002 EVN - event type segment
PID 003 PID - patient identification segment
PD1 004 PD1 - patient additional demographic
NK1 005 NK1 - next of kin / associated partie
PV1 006 PV1 - patient visit segment-
PV2 007 PV2 - patient visit - additional informa
DB1 008 DB1 - Disability segment
OBX 009 OBX - observation/result segment
AL1 010 AL1 - patient allergy information segr
DG1 011 DG1 - diagnosis segment
DRG 012 DRG - diagnosis related group segm
Group 013 PROCEDURE
PR1 014 PR1 - procedures segment
ROL 015 Role
GT1 016 GT1 - guarantor segment
Group 017 INSURANCE
IN1 018 IN1 - insurance segment
IN2 019 IN2 - insurance additional informa
IN3 020 IN3 - insurance additional informa
ACC 021 ACC - accident segment
UB1 022 UB1 - UB82 data segment
UB2 023 UB2 - UB92 data segment
OBX 009 OBX - observation/result segment
PID 003 PID - patient identification segment
PD1 004 PD1 - patient additional demographic
PV1 006 PV1 - patient visit segment-
Segment
Purpose

Illustration 8: Make the second PID Optional and Used

- ☐ Repeat the process for the second PV1.

Let's now create the Z01 segment.

- ☐ Right-click the last segment in the structure (second OBX). Choose Insert Node → Insert → Segment.

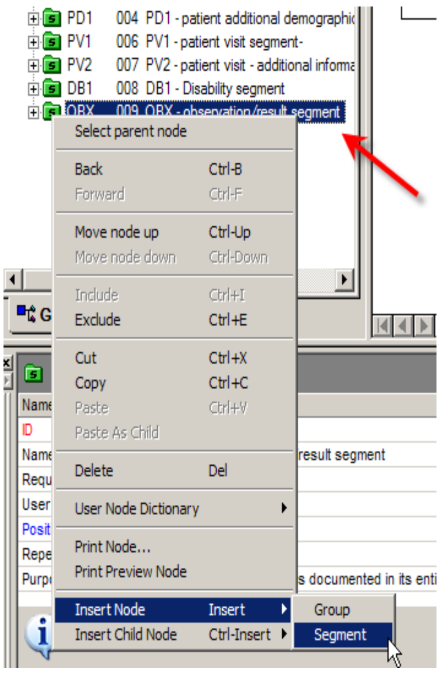
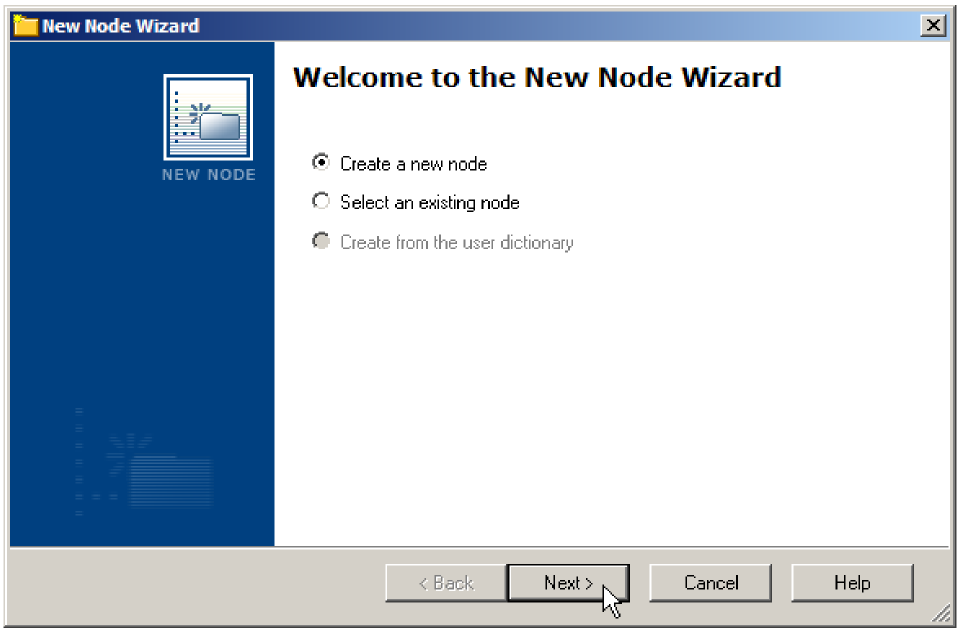


Illustration 9: Insert Segment

- ☐ Choose "Create a new node" and click Next.

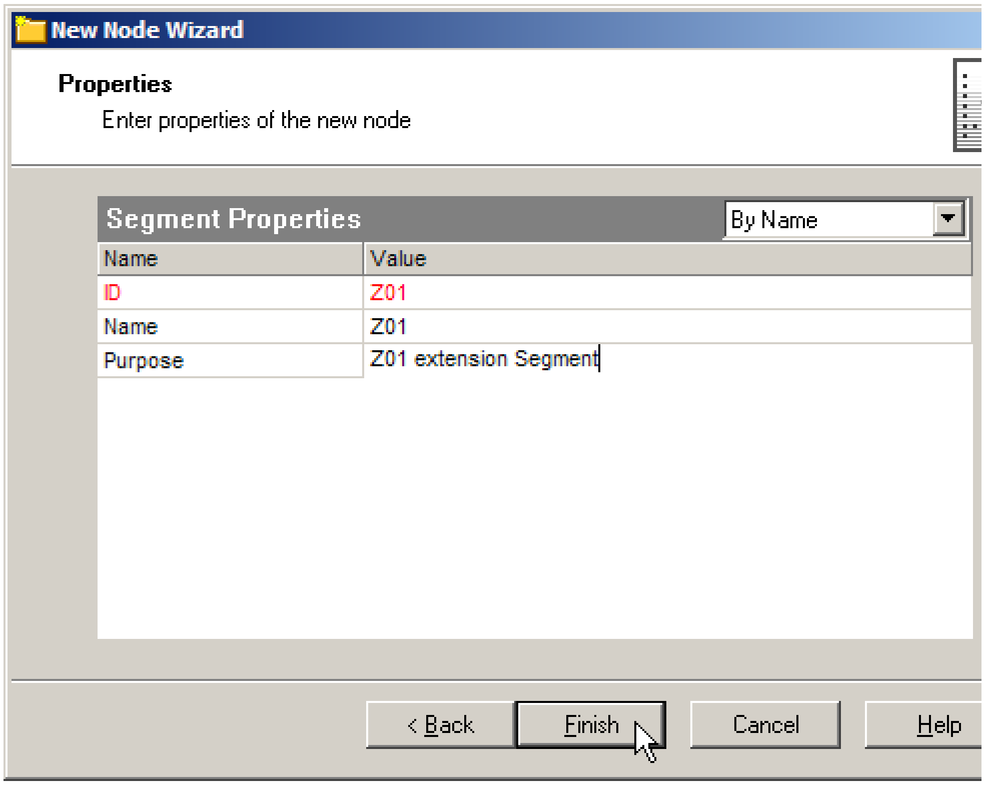


Welcome to the New Node Wizard

[✓ Create a new node]
Select an existing node
Create from the user dictionary

Illustration 10: Create a new node

☐ Enter ID: Z01, Name: Z01 and Purpose and click Finish.



Properties

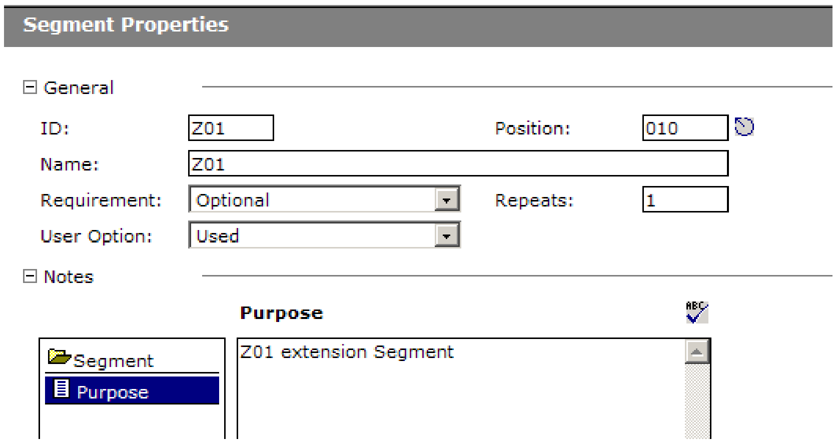
Enter properties of the new node

Segment Properties

Name	Value
ID	Z01
Name	Z01
Purpose	Z01 extension Segment

Illustration 11: Configure segment details

☐ Change Requirement and User Option to Optional and Used. This makes the segment optional.


Segment Properties

General

ID:	Z01	Position:	010
Name:	Z01		
Requirement:	Optional	Repeats:	1
User Option:	Used		

Notes

Purpose

Z01 extension Segment

Illustration 12: Make this segment optional

Recall the structure of the Z01 segment.

Sequence	Length	Data Type	Element Name
1	1	IS	Original Gender
2	1	IS	Current Gender
3	26	TS	Date of Change
4	1	IS	Legal Gender

We will add these components one at a time.

☐ Right-click on the name of the segment and choose Insert Child Node → Field.

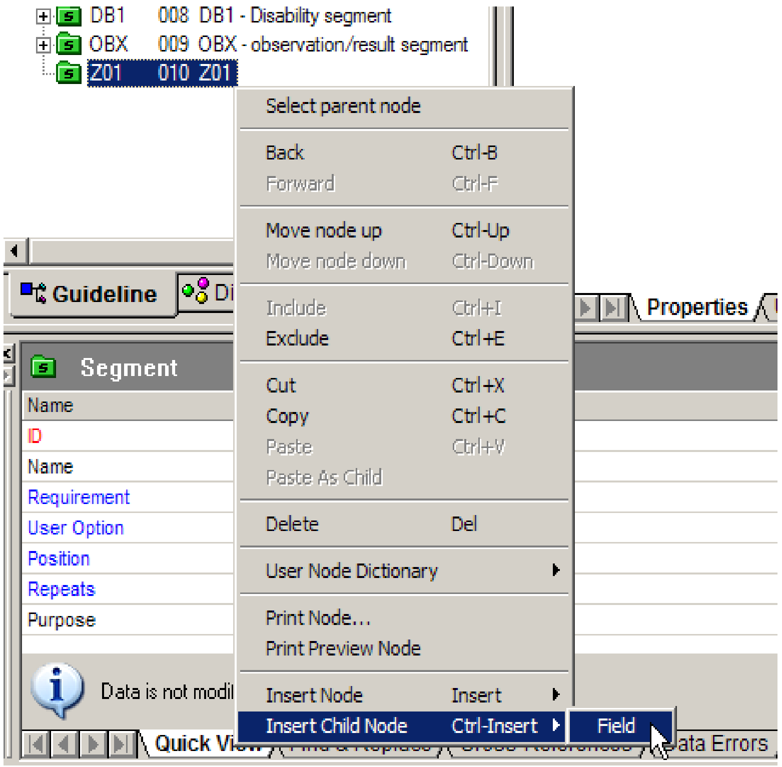
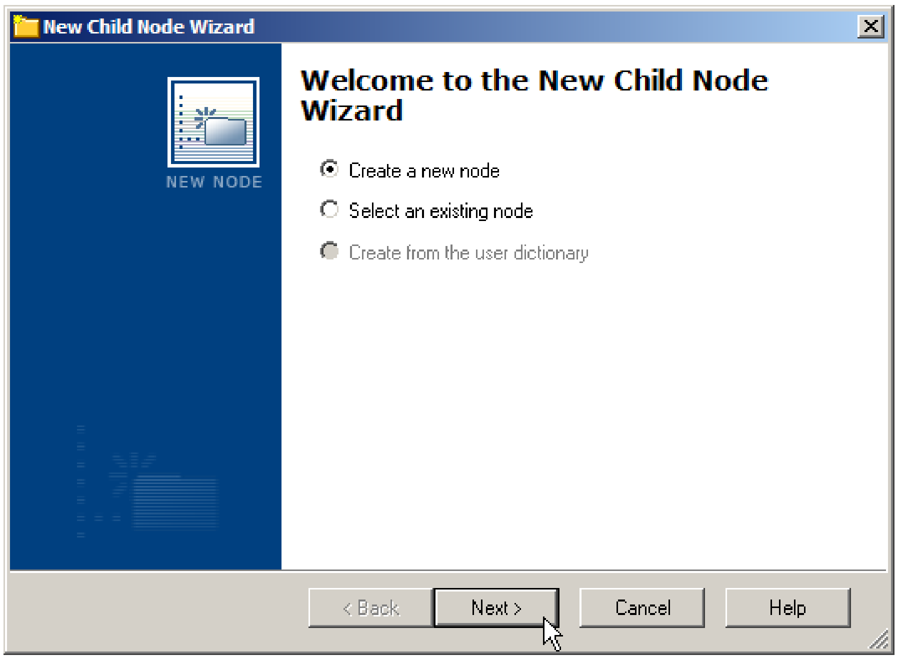


Illustration 13: Insert field as a child of the segment

☐ Accept "create a new node" and click Next.

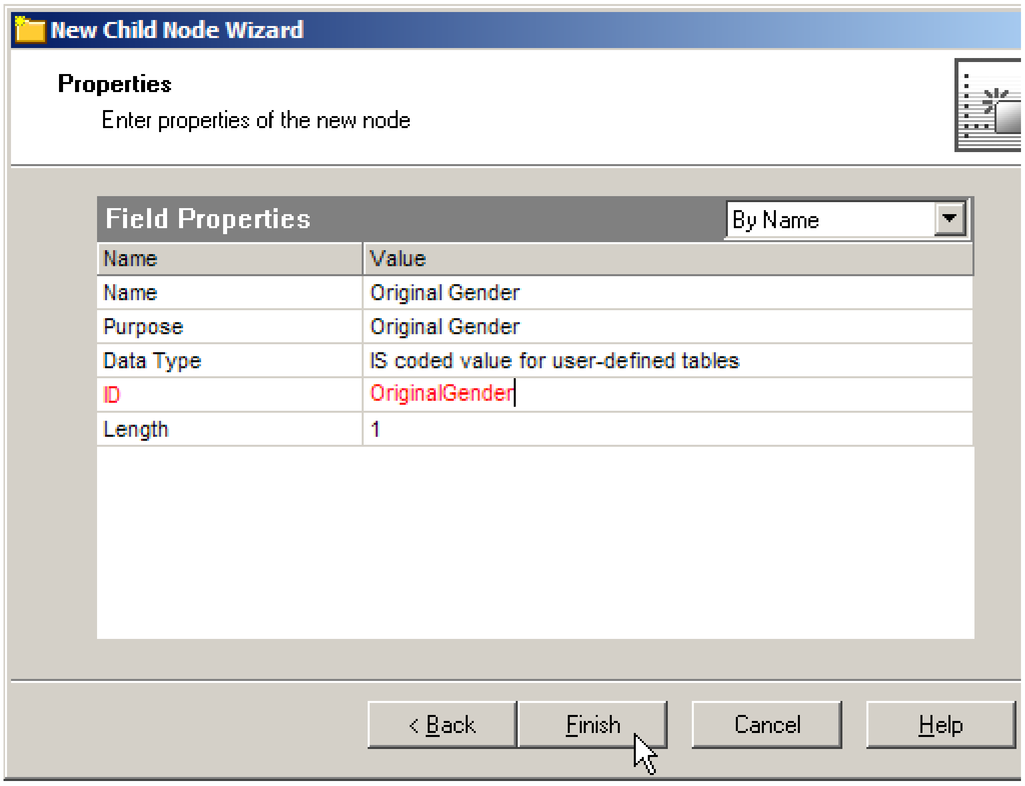


Welcome to the New Child Node Wizard

[✓ Create a new node]
Select an existing node
Create from the user dictionary

Illustration 14: Create a new node

☐ Provide values for Name, Purpose, Data Type (from a drop down menu), ID and Length and click Finish.



Properties

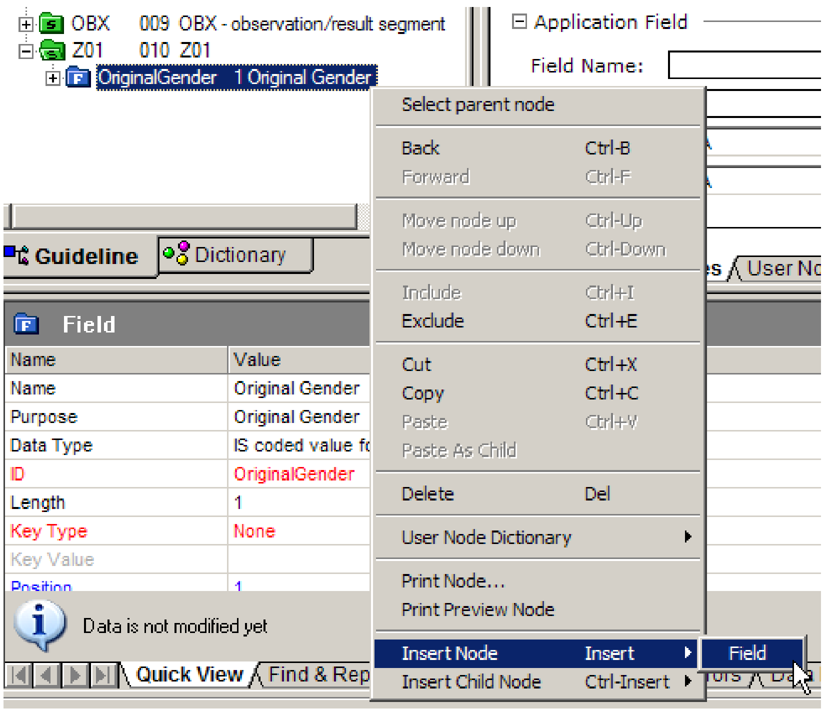
Enter properties of the new node

Field Properties — By Name

Name	Value
Name	Original Gender
Purpose	Original Gender
Data Type	IS coded value for user-defined tables
ID	OriginalGender
Length	1

Illustration 15: Defined Original Gender field

☐ Right-click on the name of the new field and choose Insert Node → Field.

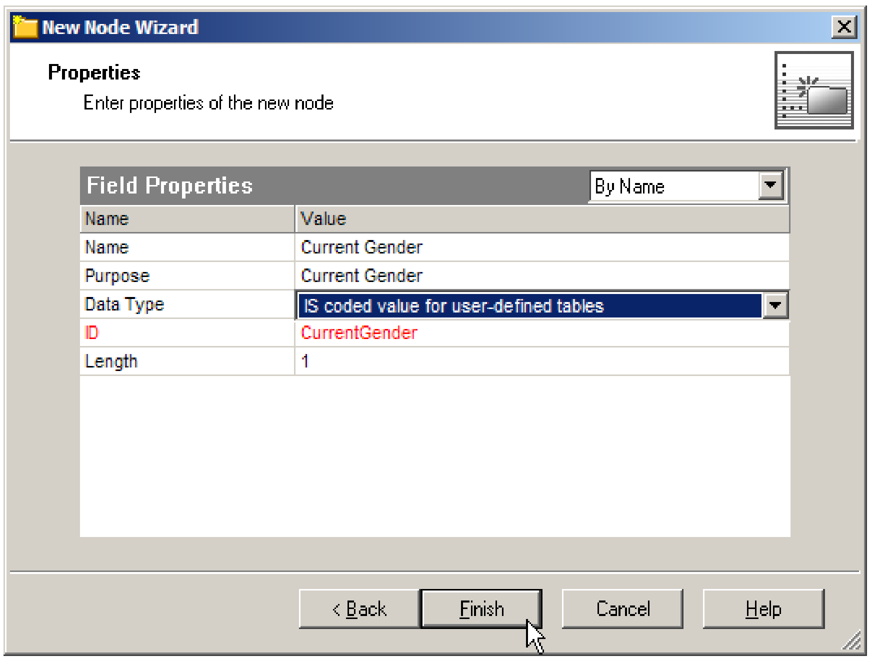


Field

Name	Value
Name	Original Gender
Purpose	Original Gender
Data Type	IS coded value fo
ID	OriginalGender
Length	1
Key Type	None
Key Value	
Position	1

Illustration 16: Add another field

☐ Provide values for Name: Current Gender, Purpose: Current Gender, Data Type: IS (from drop down menu), ID: CurrentGender and Length: 1. Click Finish.

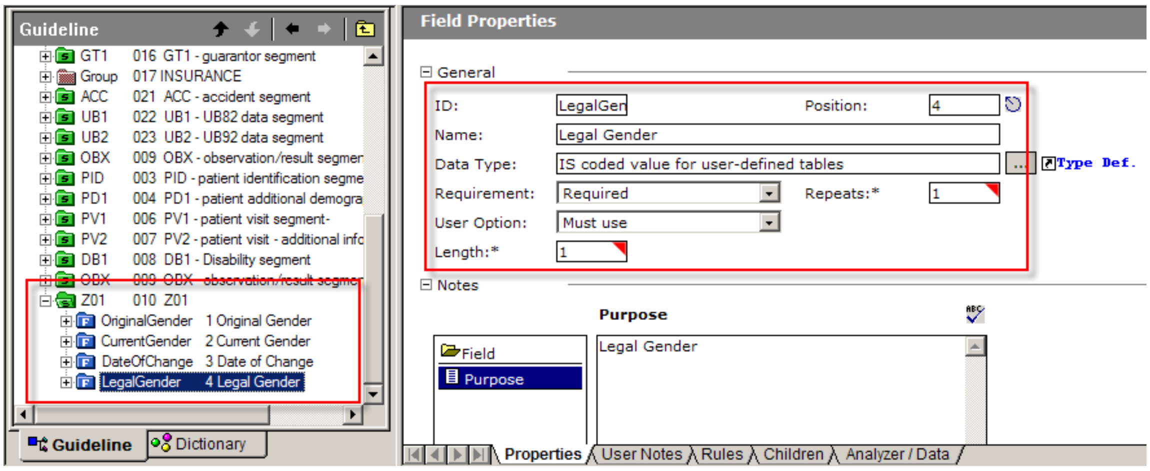


Field Properties — By Name

Name	Value
Name	Current Gender
Purpose	Current Gender
Data Type	IS coded value for user-defined tables
ID	CurrentGender
Length	1

Illustration 17: Configure Current Gender field

☐ Repeat the process for the remaining two fields – Date of Change and Legal Gender.


Guideline

GT1	016 GT1 - guarantor segment
Group	017 INSURANCE
ACC	021 ACC - accident segment
UB1	022 UB1 - UB82 data segment
UB2	023 UB2 - UB92 data segment
OBX	009 OBX - observation/result segmer
PID	003 PID - patient identification segme
PD1	004 PD1 - patient additional demogra
PV1	006 PV1 - patient visit segment-
PV2	007 PV2 - patient visit - additional infc
DB1	008 DB1 - Disability segment
OBX	009 OBX - observation/result segme
Z01	010 Z01
OriginalGender	1 Original Gender
CurrentGender	2 Current Gender
DateOfChange	3 Date of Change
LegalGender	4 Legal Gender

Field Properties

General

ID: LegalGen
Position: 4
Name: Legal Gender
Data Type: IS coded value for user-defined tables
Requirement: Required
Repeats:* 1
User Option: Must use
Length:* 1

Notes

Purpose

Legal Gender

Illustration 18: Completed structure

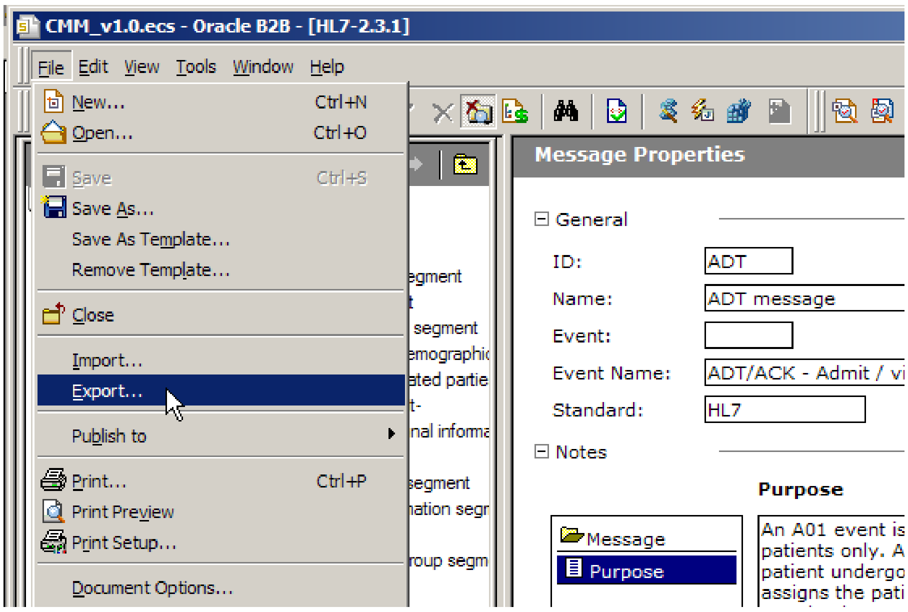
☐ Note that by default new fields are Required and Must Use. Modify as needed for your fields.

☐ Save the structure.

This structure/guideline/spec is a customisation of the ADT A01 structure.

Once the structure is ready we can save the ECS file and export the XSD file. The former is used in Oracle SOA Suite B2B for message validation (optional) and conversion between HL7 v2 Delimited and XML (in either direction). The latter is used for integration between the Oracle SOA Suite B2B and other parts of the SOA Suite.

☐ Pull down the File menu and choose Export.


CMM_v1.0.ecs - Oracle B2B - [HL7-2.3.1]

Message Properties

General

ID: ADT
Name: ADT message
Event:
Event Name: ADT/ACK - Admit / vi
Standard: HL7

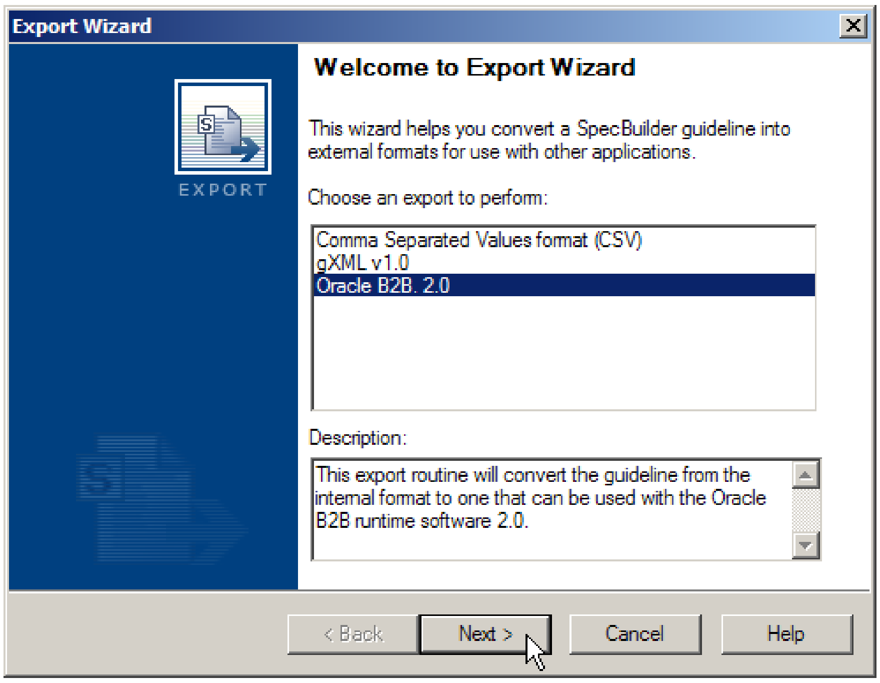
Notes

Purpose

An A01 event is
patients only. A
patient undergo
assigns the pati

Illustration 20: Trigger Export Wizard

☐ Choose Oracle B2B 2.0 and click Next.


Export Wizard

Welcome to Export Wizard

This wizard helps you convert a SpecBuilder guideline into external formats for use with other applications.

Choose an export to perform:

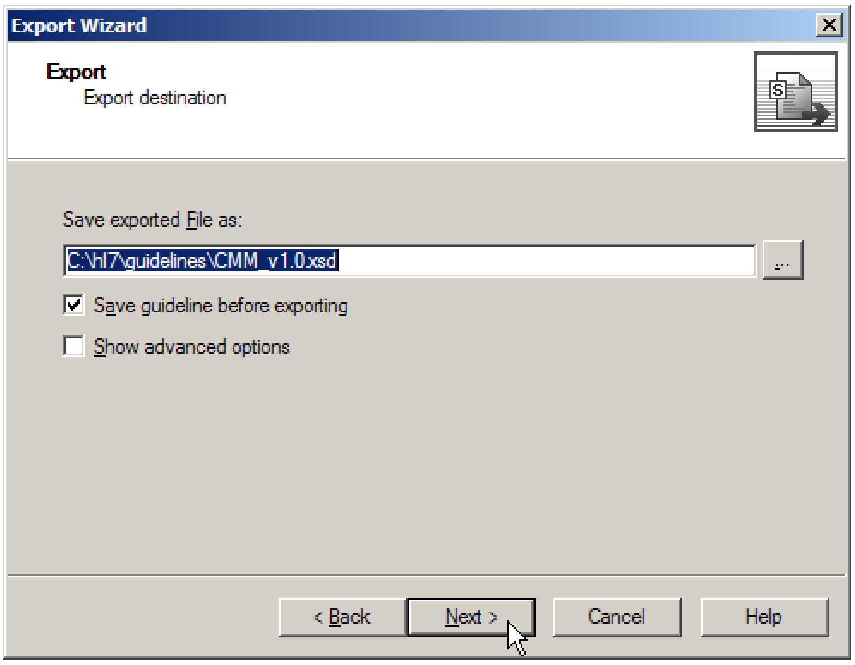
Comma Separated Values format (CSV)
gXML v1.0
Oracle B2B. 2.0

Description:

This export routine will convert the guideline from the internal format to one that can be used with the Oracle B2B runtime software 2.0.

Illustration 21: Coose to export to Oracle   format

☐ Check “Save guideline before exporting”, change the name to an appropriate name and click Next.


Export Wizard

Export
Export destination

Save exported File as: C:\hl7\guidelines\CMM_v1.0xsd

☑ Save guideline before exporting

☐ Show advanced options

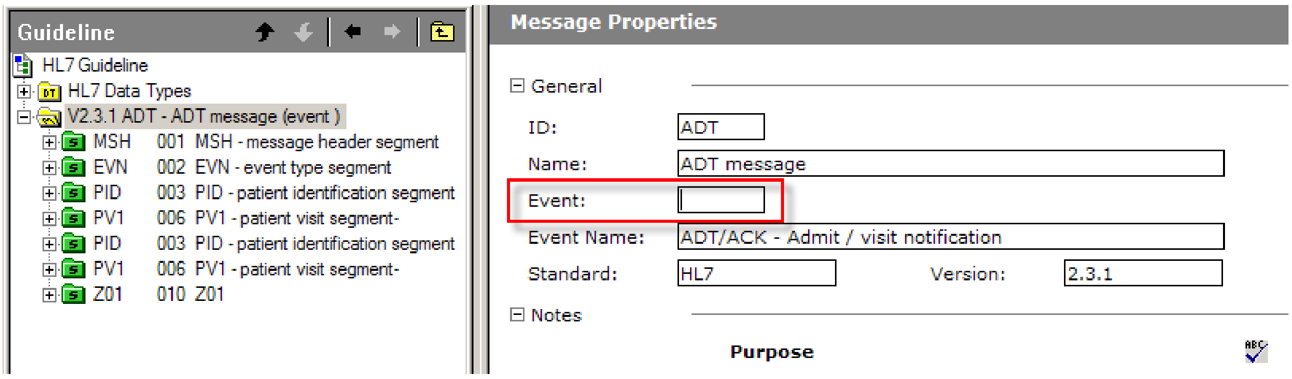
Illustration 22: Name the file and choose to save guideline

Two new files will appear in the file system – CMM_v1.0.ecs (EDIFECS Guideline) and CMM_v1.0xsd (XML Schema Document).

Remove unused Segments

The canonical message model structure which we created includes all segments from the A01/A08 message and all extra segments from 17 and the Z segment. We stated earlier that our messages will only ever use MSH, EVN, PID, PV1 and Z01 segments. Let's delete all other segments and save/export the structure.

- ☐ Open the CMM_v1.0.ecs file and delete all segments except the ones mentioned, by right-clicking the segment and choosing Delete. Once done save and export the modified files.
- ☐ While at it, click the root node of the structure and clear the Event field to eliminate Event Type dependency.


Guideline
HL7 Guideline
HL7 Data Types
V2.3.1 ADT - ADT message (event)
MSH 001 MSH - message header segment
EVN 002 EVN - event type segment
PID 003 PID - patient identification segment
PV1 006 PV1 - patient visit segment-
PID 003 PID - patient identification segment
PV1 006 PV1 - patient visit segment-
Z01 010 Z01
Message Properties
General
Notes
Purpose

ID:	ADT		
Name:	ADT message		
Event:			
Event Name:	ADT/ACK - Admit / visit notification		
Standard:	HL7	Version:	2.3.1

Illustration 23: Minimalist CMM

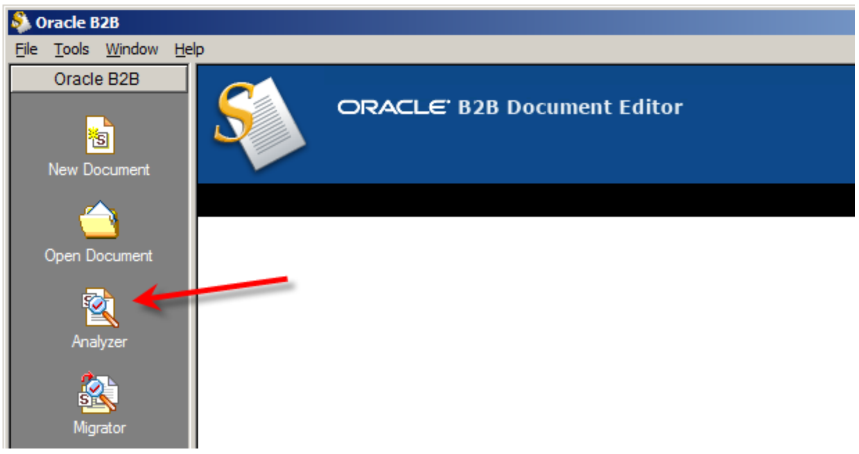
Analyse and Customize the Structure

The canonical message model must accommodate sample data in such a way that a sample message can be validated using the structure. To ensure this we will use data analysis functionality of the Oracle Document Editor to validate sample A01 and A03 messages, and review and correct errors.

Analyze using an A01 Message

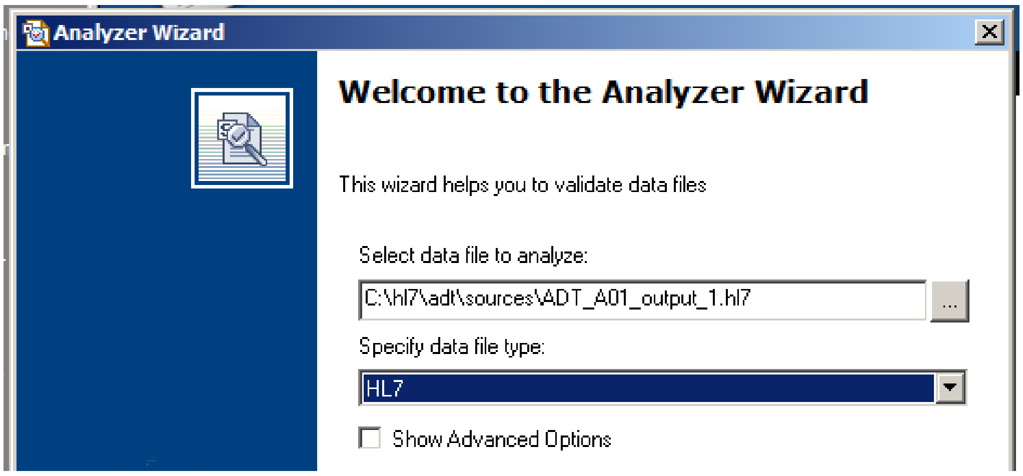
We will use a sample A01 message to see whether the message structure accommodates the message without errors. In short, we will attempt to validate the sample message using the message structure as the template.   The steps are discussed in detail below.

- ☐ Start the Oracle Document Editor. Click on the Analyzer button to start the Data Analyzer



☐ Select sample file named **ADT_A01_output_1.hl7**

☐ Change Data File Type to HL7 and click Next



Analyzer Wizard

Welcome to the Analyzer Wizard

This wizard helps you to validate data files

Select data file to analyze:

C:\hl7\adt\sources\ADT_A01_output_1.hl7

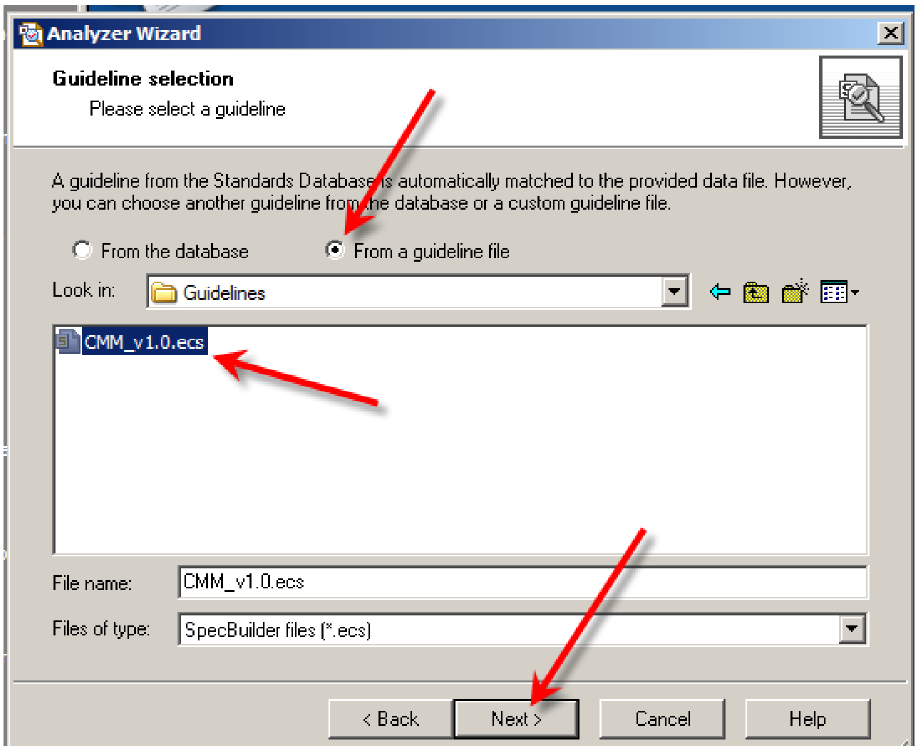
Specify data file type:

HL7

☐ Show Advanced Options

☐ Check the "From a guideline file" radio button

☐ Choose the CMM_v1.0.ecs guideline file and click Next.



Analyzer Wizard

Guideline selection

Please select a guideline

A guideline from the Standards Database is automatically matched to the provided data file. However, you can choose another guideline from the database or a custom guideline file.

○ From the database ● From a guideline file

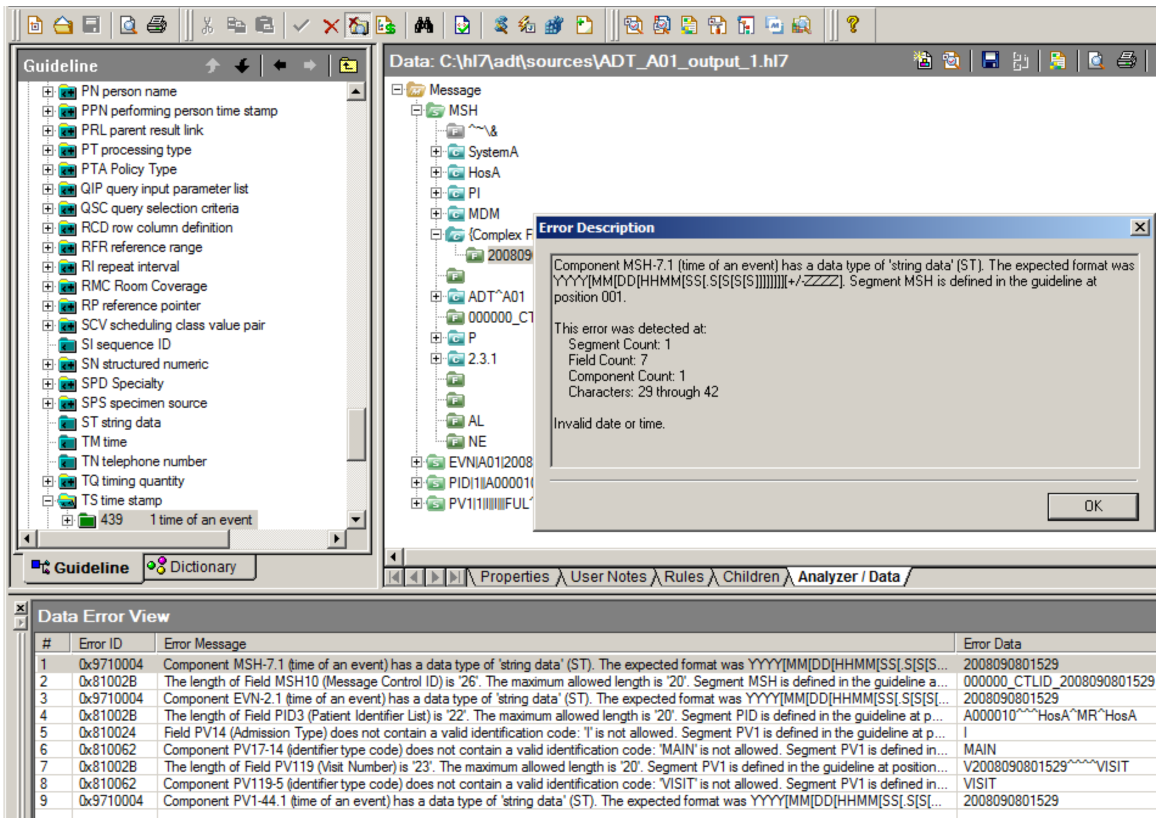
Look in: Guidelines

CMM_v1.0.ecs

File name: CMM_v1.0.ecs

Files of type: SpecBuilder files (*.ecs)

☐ Inspect error messages to see what kinds of issues the analyzer is reporting for the A01 sample message.



Error Description

Component MSH-7.1 (time of an event) has a data type of 'string data' (ST). The expected format was YYYY[MM[DD[HHMM[SS[.S[S[S[S]]]]]]]][+/-ZZZZ]. Segment MSH is defined in the guideline at position 001.

This error was detected at:
Segment Count: 1
Field Count: 7
Component Count: 1
Characters: 29 through 42

Invalid date or time.

OK

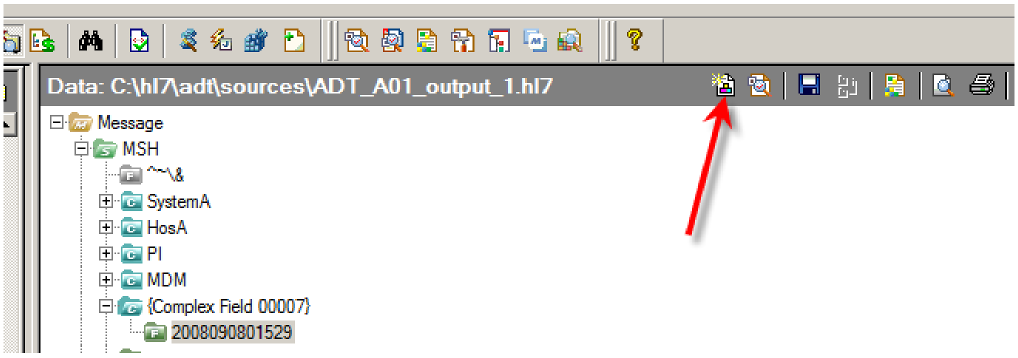
Data Error View

#	Error ID	Error Message	Error Data
1	0x9710004	Component MSH-7.1 (time of an event) has a data type of 'string data' (ST). The expected format was YYYY[MM[DD[HHMM[SS[.S[S[S...	2008090801529
2	0x81002B	The length of Field MSH10 (Message Control ID) is '26'. The maximum allowed length is '20'. Segment MSH is defined in the guideline a...	000000_CTLID_2008090801529
3	0x9710004	Component EVN-2.1 (time of an event) has a data type of 'string data' (ST). The expected format was YYYY[MM[DD[HHMM[SS[.S[S[S[...	2008090801529
4	0x81002B	The length of Field PID3 (Patient Identifier List) is '22'. The maximum allowed length is '20'. Segment PID is defined in the guideline at p...	A000010^^^HosA^MR^HosA
5	0x810024	Field PV14 (Admission Type) does not contain a valid identification code: 'I' is not allowed. Segment PV1 is defined in the guideline at p...	I
6	0x810062	Component PV17-14 (identifier type code) does not contain a valid identification code: 'MAIN' is not allowed. Segment PV1 is defined in...	MAIN
7	0x81002B	The length of Field PV119 (Visit Number) is '23'. The maximum allowed length is '20'. Segment PV1 is defined in the guideline at position...	V2008090801529^^^^VISIT
8	0x810062	Component PV119-5 (identifier type code) does not contain a valid identification code: 'VISIT' is not allowed. Segment PV1 is defined in...	VISIT
9	0x9710004	Component PV1-44.1 (time of an event) has a data type of 'string data' (ST). The expected format was YYYY[MM[DD[HHMM[SS[.S[S[...	2008090801529

The issues fall into the following categories:

1. Data format errors – for date/ time fields in which data does not agree with expected format – MSH-7.1, EVN-2.1, PV1-44.1
2. Compound field length errors – the total length allowed for a compound field is insufficient to accommodate the combined lengths of constituent component – MSH-10, PID-3, PV1-19
3. Field value not in a list of values – the code is not in the code set defined for the field – PV1-4, PV1-7.14, PV1-19.5

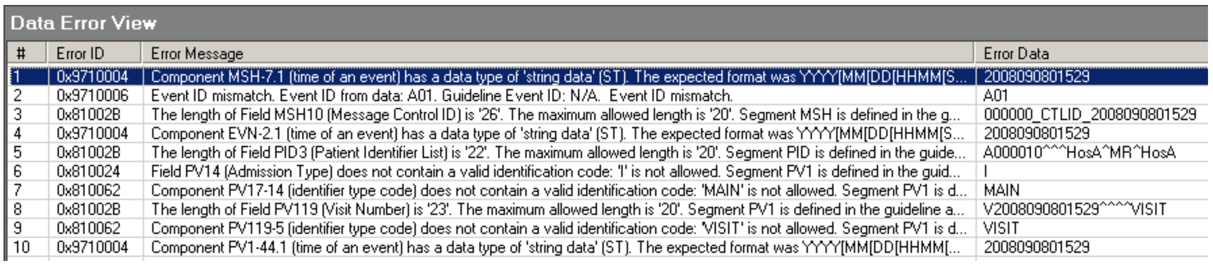
☐ Click on the New Data button to analyze the A03 sample



Analyze using the A03 sample

☐ Select sample file named **ADT_A03_output_1.hl7** and click Next

☐ Inspect error messages to see what kinds of issues the analyzer is reporting for the A03 sample message.



Data Error View

#	Error ID	Error Message	Error Data
1	0x9710004	Component MSH-7.1 (time of an event) has a data type of 'string data' (ST). The expected format was YYYY[MM[DD[HHMM[S...	2008090801529
2	0x9710006	Event ID mismatch. Event ID from data: A01. Guideline Event ID: N/A. Event ID mismatch.	A01
3	0x81002B	The length of Field MSH10 (Message Control ID) is '26'. The maximum allowed length is '20'. Segment MSH is defined in the g...	000000_CTLID_2008090801529
4	0x9710004	Component EVN-2.1 (time of an event) has a data type of 'string data' (ST). The expected format was YYYY[MM[DD[HHMM[S...	2008090801529
5	0x81002B	The length of Field PID3 (Patient Identifier List) is '22'. The maximum allowed length is '20'. Segment PID is defined in the guide...	A000010^^^HosA^MR^HosA
6	0x810024	Field PV14 (Admission Type) does not contain a valid identification code: 'I' is not allowed. Segment PV1 is defined in the guid...	I
7	0x810062	Component PV17-14 (identifier type code) does not contain a valid identification code: 'MAIN' is not allowed. Segment PV1 is d...	MAIN
8	0x81002B	The length of Field PV119 (Visit Number) is '23'. The maximum allowed length is '20'. Segment PV1 is defined in the guideline a...	V2008090801529^^^^VISIT
9	0x810062	Component PV119-5 (identifier type code) does not contain a valid identification code: 'VISIT' is not allowed. Segment PV1 is d...	VISIT
10	0x9710004	Component PV1-44.1 (time of an event) has a data type of 'string data' (ST). The expected format was YYYY[MM[DD[HHMM[...	2008090801529

As before, the issues fall into the following categories:

- ☐ Data format errors – for date/ time fields in which data does not agree with expected format – MSH-7.1, EVN-2.1, PV1-44.1
- ☐ Compound field length errors – the total length allowed for a compound field is insufficient to accommodate the combined lengths of constituent component – MSH-10, PID-3, PV1-19, PV1-36
- ☐ Field value not in a list of values – the code is not in the code set defined for the field – PV1-4, PV1-7.14, PV1-19.5, PV1-36

In addition, we have an issue with the Event ID, which does not match the Event ID in the guideline. We will ignore this particular issue for the present.

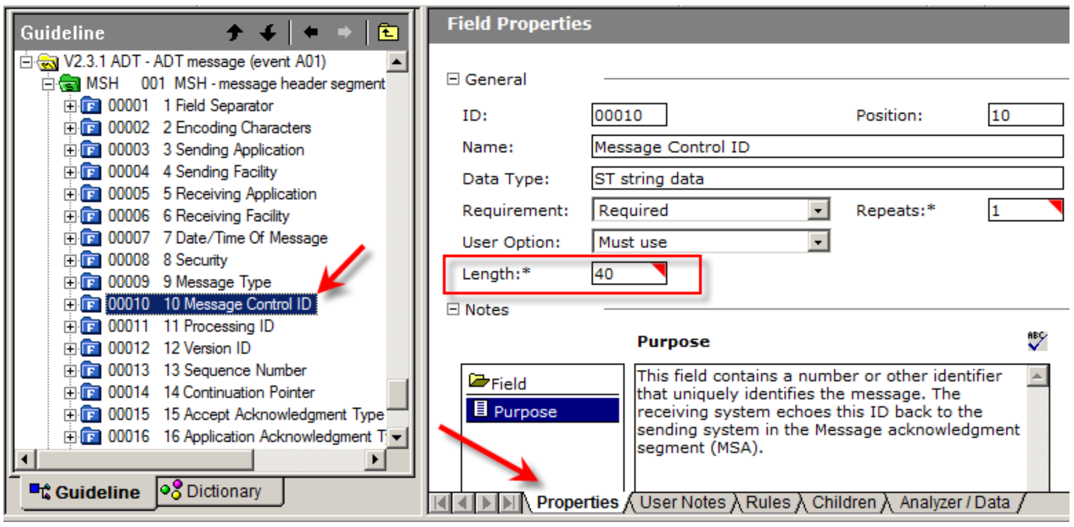
Modify Message Structure to fix A03 sample message errors

We know that the sample A01 and A03 messages are not valid according to the canonical message structure which we developed so far. We will modify the canonical message structure to accommodate sample messages and will validate them again. The steps are discussed in detail below.

Leave Error relating to "Event ID Mismatch" alone – we will not deal with it at present

Fix Error "The length of Field MSH10 (Message Control ID) is '27'. The maximum allowed length is '20'. Segment MSH is defined in the guideline at position 001."

- ☐ Click corresponding Error line in the Data Error View pane
- ☐ Click MSH-10 field in the Guideline pane
- ☐ Click the Properties Tab in the Field Properties pane and change Length property from 20 to a value equal to or greater than 27 – say 40



Guideline

V2.3.1 ADT - ADT message (event A01)
MSH 001 MSH - message header segment
00001 1 Field Separator
00002 2 Encoding Characters
00003 3 Sending Application
00004 4 Sending Facility
00005 5 Receiving Application
00006 6 Receiving Facility
00007 7 Date/Time Of Message
00008 8 Security
00009 9 Message Type
00010 10 Message Control ID
00011 11 Processing ID
00012 12 Version ID
00013 13 Sequence Number
00014 14 Continuation Pointer
00015 15 Accept Acknowledgment Type
00016 16 Application Acknowledgment T

Field Properties

General

ID: 00010
Position: 10
Name: Message Control ID
Data Type: ST string data
Requirement: Required
Repeats:* 1
User Option: Must use
Length:* 40

Notes

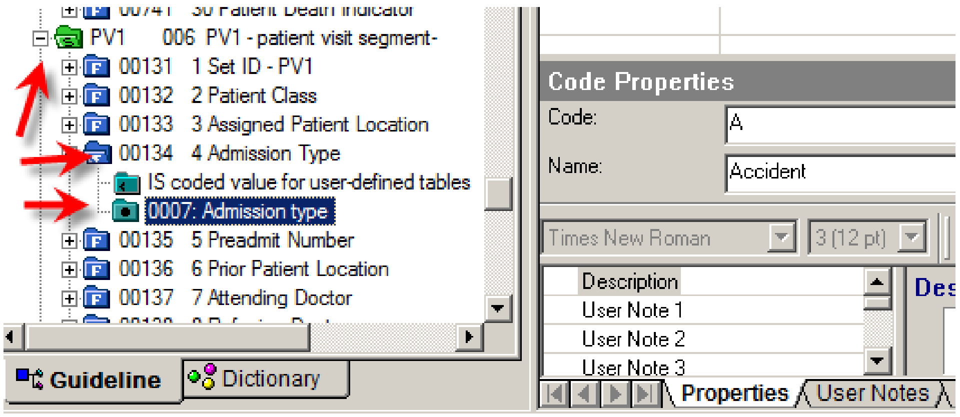
Purpose

This field contains a number or other identifier that uniquely identifies the message. The receiving system echoes this ID back to the sending system in the Message acknowledgment segment (MSA).

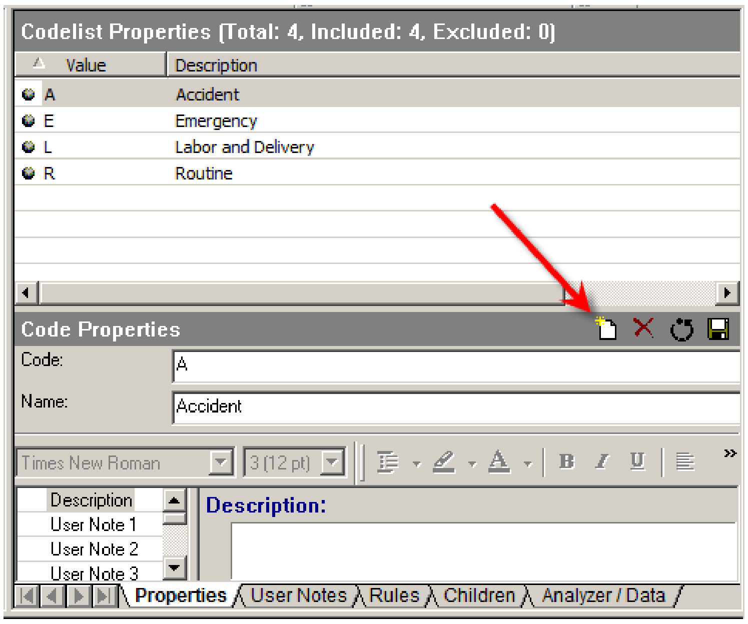
- ☐ Fix Error "The length of Field PID3 (Patient Identifier List) is '22'. The maximum allowed length is '20'. Segment PID is defined in the guideline at position 003"
- ☐ Click corresponding Error line in the Data Error Pane
- ☐ Click PID-3 field in the Guideline pane
- ☐ Click the Properties Tab in the Field Properties pane and change Length property from 20 to a value equal to or greater than 27 – say 40

Fix Error "Field PV14 (Admission Type) does not contain a valid identification code: 'I' is not allowed. Segment PV1 is defined in the guideline at position 006."

- ☐ Click corresponding Error line in the Data Error Pane
- ☐ Expand PV1-4 structure
- ☐ Select 0007 Admission Type

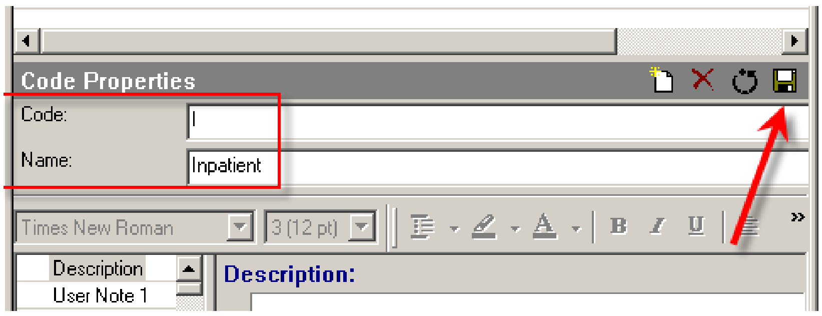


- ☐ Click "New Code" button in the Codelist Properties pane


Codelist Properties (Total: 4, Included: 4, Excluded: 0)

Value	Description
A	Accident
E	Emergency
L	Labor and Delivery
R	Routine

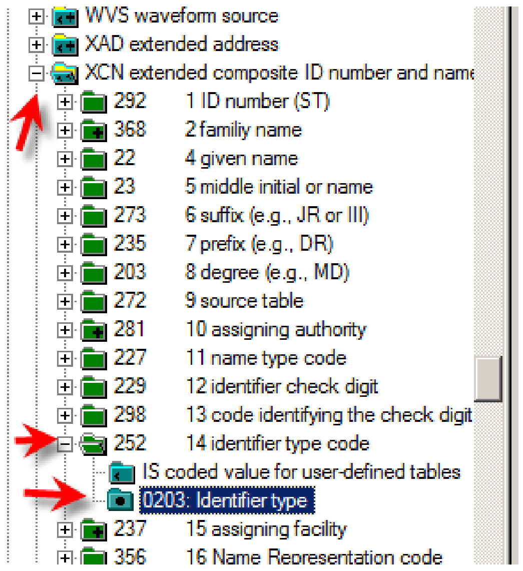
- ☐ Enter Code value of "I" and Name value of "Inpatient" and click the Save button



Code: I
Name: Inpatient

Fix Error "Component PV17-14 (identifier type code) does not contain a valid identification code: 'MAIN' is not allowed. Segment PV1 is defined in the guideline at position 006."

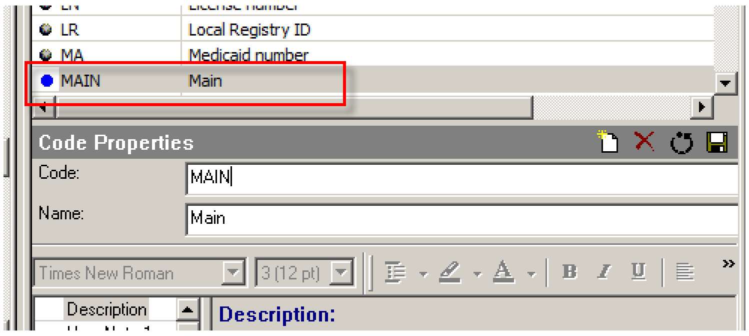
- ☐ Click corresponding Error line in the Data Error Pane
- ☐ Expand XCN structure



WVS waveform source
XAD extended address
XCN extended composite ID number and name

292	1 ID number (ST)
368	2 familiy name
22	4 given name
23	5 middle initial or name
273	6 suffix (e.g., JR or III)
235	7 prefix (e.g., DR)
203	8 degree (e.g., MD)
272	9 source table
281	10 assigning authority
227	11 name type code
229	12 identifier check digit
298	13 code identifying the check digit
252	14 identifier type code
	IS coded value for user-defined tables
	0203: Identifier type
237	15 assigning facility
356	16 Name Representation code

- ☐ Select 0203 Identifier Type
- ☐ Click "New Code" button in the Codelist Properties pane
- ☐ Enter Code value of "MAIN" and Name value of "Main" and click the Save button



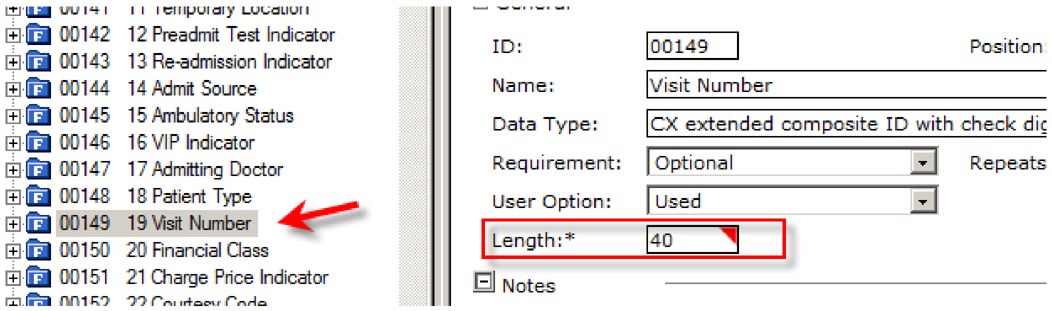
LR	Local Registry ID
MA	Medicaid number
MAIN	Main

Code: MAIN
Name: Main

Fix Error "The length of Field PV119 (Visit Number) is '23'. The maximum allowed length is '20'. Segment PV1 is defined in the guideline at position 006."

- ☐ Click corresponding Error line in the Data Error Pane
- ☐ Click PV1-19 field in the Guideline pane

- ☐ Click the Properties Tab in the Field Properties pane and change Length property from 20 to a value equal to or greater than 23 – say 40

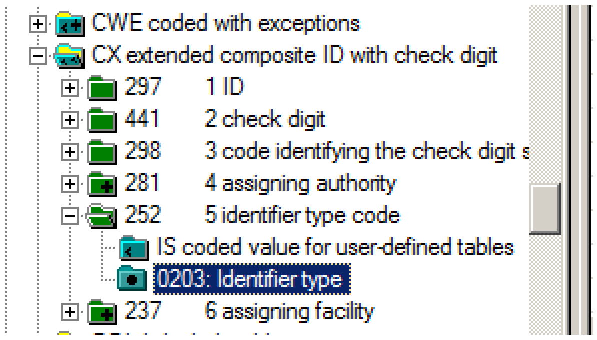


00142 12 Preadmit Test Indicator
00143 13 Re-admission Indicator
00144 14 Admit Source
00145 15 Ambulatory Status
00146 16 VIP Indicator
00147 17 Admitting Doctor
00148 18 Patient Type
00149 19 Visit Number
00150 20 Financial Class
00151 21 Charge Price Indicator

ID: 00149
Name: Visit Number
Data Type: CX extended composite ID with check dig
Requirement: Optional
User Option: Used
Length:* 40

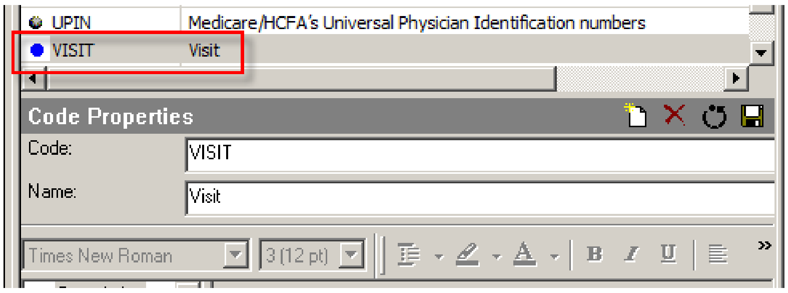
Fix Error "Component PV119-5 (identifier type code) does not contain a valid identification code: 'VISIT' is not allowed. Segment PV1 is defined in the guideline at position 006."

- ☐ Click corresponding Error line in the Data Error Pane
- ☐ Expand CX structure



CWE coded with exceptions
CX extended composite ID with check digit
297 1 ID
441 2 check digit
298 3 code identifying the check digit s
281 4 assigning authority
252 5 identifier type code
IS coded value for user-defined tables
0203: Identifier type
237 6 assigning facility

- ☐ Select 0203 Identifier Type
- ☐ Click "New Code" button in the Codelist Properties pane
- ☐ Enter Code value of "VISIT" and Name value of "Visit" and click the Save button

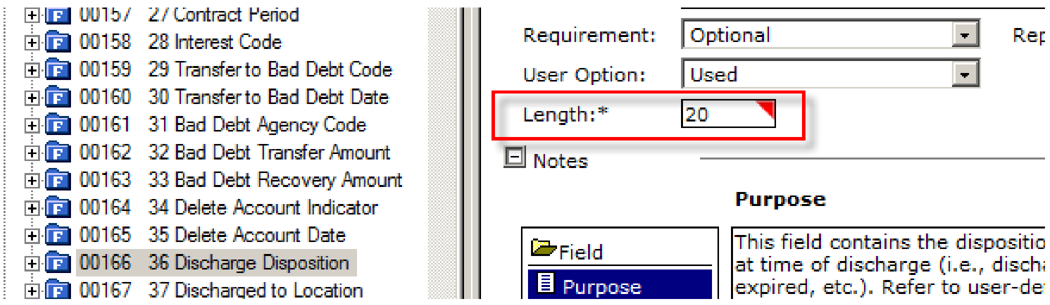


UPIN	Medicare/HCFA's Universal Physician Identification numbers
VISIT	Visit

Code Properties
Code: VISIT
Name: Visit

Fix Error "The length of Field PV136 (Discharge Disposition) is '9'. The maximum allowed length is '3'. Segment PV1 is defined in the guideline at position 006."

- ☐ Click corresponding Error line in the Data Error Pane
- ☐ Click PV1-36 field in the Guideline pane
- ☐ Click the Properties Tab in the Field Properties pane and change Length property from 20 to a value equal to or greater than 3 – say 20


00157 27 Contract Period
00158 28 Interest Code
00159 29 Transfer to Bad Debt Code
00160 30 Transfer to Bad Debt Date
00161 31 Bad Debt Agency Code
00162 32 Bad Debt Transfer Amount
00163 33 Bad Debt Recovery Amount
00164 34 Delete Account Indicator
00165 35 Delete Account Date
00166 36 Discharge Disposition
00167 37 Discharged to Location

Requirement: Optional
User Option: Used
Length:* 20

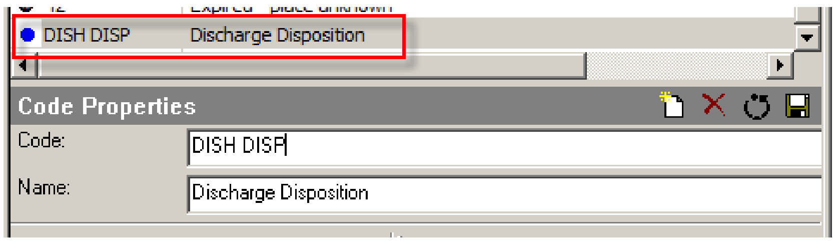
Notes

Purpose

This field contains the dispositio
at time of discharge (i.e., discha
expired, etc.). Refer to user-de

Fix Error “Field PV136 (Discharge Disposition) does not contain a valid identification code: 'DISH DISP' is not allowed. Segment PV1 is defined in the guideline at position 006.”

- ☐ Click corresponding Error line in the Data Error Pane
- ☐ Expand PV1-36 field in the Guideline pane
- ☐ Select 0112 Discharge Disposition
- ☐ Click “New Code” button in the Codelist Properties pane
- ☐ Enter Code value of “DISH DISP” and Name value of “Discharge Disposition” and click the Save button


DISH DISP Discharge Disposition

Code Properties

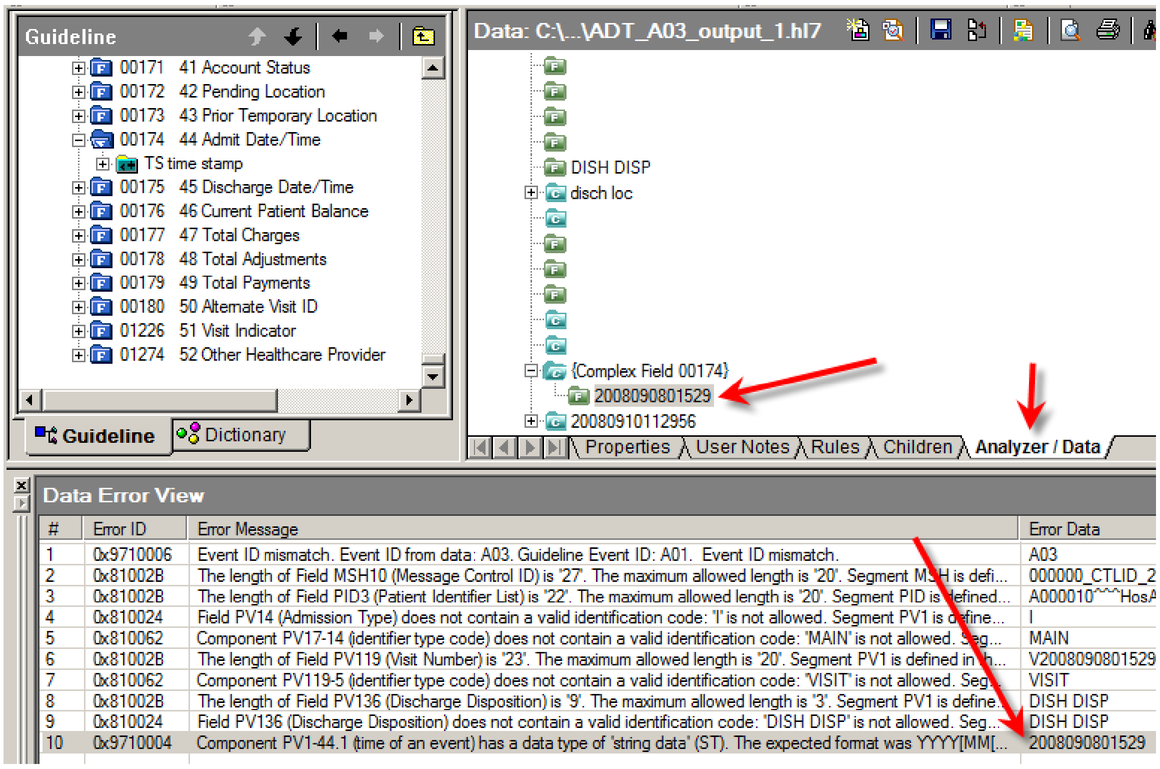
Code: DISH DISP

Name: Discharge Disposition

Fix Error “Component PV1-44.1 (time of an event) has a data type of 'string data' (ST). The expected format was YYYY[MM[DD[HHMM[SS[.S[S[S[S]]]]]]]][+/-ZZZZ]. Segment PV1 is defined in the guideline at position 006.”

Observe that the alleged date/time string is “2008090801529”, which is 13 characters in length. The date/time string is not valid. Rather than fixing the data we will change the data type from TS – Timestamp to ST – String

Note that there are 2 PV1 segments in the canonical message structure – PV1-14 field will have to be fixed in both so the procedure needs to be followed for both


Guideline

- 00171 41 Account Status
- 00172 42 Pending Location
- 00173 43 Prior Temporary Location
- 00174 44 Admit Date/Time
  - TS time stamp
- 00175 45 Discharge Date/Time
- 00176 46 Current Patient Balance
- 00177 47 Total Charges
- 00178 48 Total Adjustments
- 00179 49 Total Payments
- 00180 50 Alternate Visit ID
- 01226 51 Visit Indicator
- 01274 52 Other Healthcare Provider

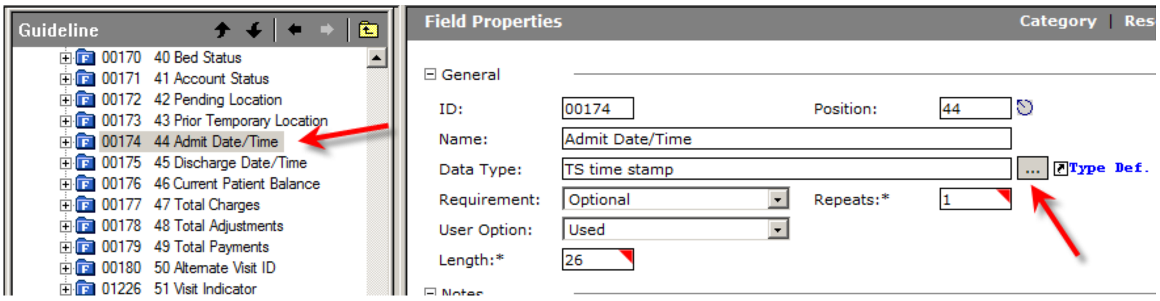
Data: C:\...\ADT_A03_output_1.hl7

- DISH DISP
- disch loc
- {Complex Field 00174}
  - 2008090801529
- 20080910112956

Data Error View

#	Error ID	Error Message	Error Data
1	0x9710006	Event ID mismatch. Event ID from data: A03. Guideline Event ID: A01. Event ID mismatch.	A03
2	0x81002B	The length of Field MSH10 (Message Control ID) is '27'. The maximum allowed length is '20'. Segment MSH is defi...	000000_CTLID_2
3	0x81002B	The length of Field PID3 (Patient Identifier List) is '22'. The maximum allowed length is '20'. Segment PID is defined...	A000010^^^HosA
4	0x810024	Field PV14 (Admission Type) does not contain a valid identification code: 'I' is not allowed. Segment PV1 is define...	I
5	0x810062	Component PV17-14 (identifier type code) does not contain a valid identification code: 'MAIN' is not allowed. Seg...	MAIN
6	0x81002B	The length of Field PV119 (Visit Number) is '23'. The maximum allowed length is '20'. Segment PV1 is defined in th...	V2008090801529
7	0x810062	Component PV119-5 (identifier type code) does not contain a valid identification code: 'VISIT' is not allowed. Seg...	VISIT
8	0x81002B	The length of Field PV136 (Discharge Disposition) is '9'. The maximum allowed length is '3'. Segment PV1 is define...	DISH DISP
9	0x810024	Field PV136 (Discharge Disposition) does not contain a valid identification code: 'DISH DISP' is not allowed. Seg...	DISH DISP
10	0x9710004	Component PV1-44.1 (time of an event) has a data type of 'string data' (ST). The expected format was YYYY[MM[...	2008090801529

- ☐ Click corresponding Error line in the Data Error View pane
- ☐ Select Field PV1-44 in the Guideline pane
- ☐ Click Properties Tab in Data pane
- ☐ Click "Select data type" button (ellipsis button)


Guideline

- 00170 40 Bed Status
- 00171 41 Account Status
- 00172 42 Pending Location
- 00173 43 Prior Temporary Location
- 00174 44 Admit Date/Time
- 00175 45 Discharge Date/Time
- 00176 46 Current Patient Balance
- 00177 47 Total Charges
- 00178 48 Total Adjustments
- 00179 49 Total Payments
- 00180 50 Alternate Visit ID
- 01226 51 Visit Indicator

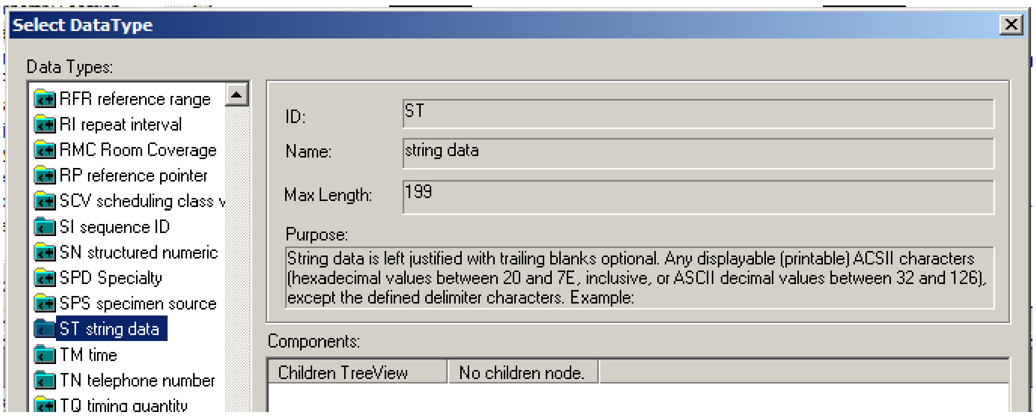
Field Properties

General

ID:	00174	Position:	44
Name:	Admit Date/Time		
Data Type:	TS time stamp		
Requirement:	Optional	Repeats:*	1
User Option:	Used		
Length:*	26		

Notes

- ☐ Choose ST – String Data from the Data Types list and click OK


Select DataType

Data Types:

- RFR reference range
- RI repeat interval
- RMC Room Coverage
- RP reference pointer
- SCV scheduling class v
- SI sequence ID
- SN structured numeric
- SPD Specialty
- SPS specimen source
- ST string data
- TM time
- TN telephone number
- TQ timing quantity

ID:	ST
Name:	string data
Max Length:	199

Purpose:
String data is left justified with trailing blanks optional. Any displayable (printable) ACSII characters (hexadecimal values between 20 and 7E, inclusive, or ASCII decimal values between 32 and 126), except the defined delimiter characters. Example:

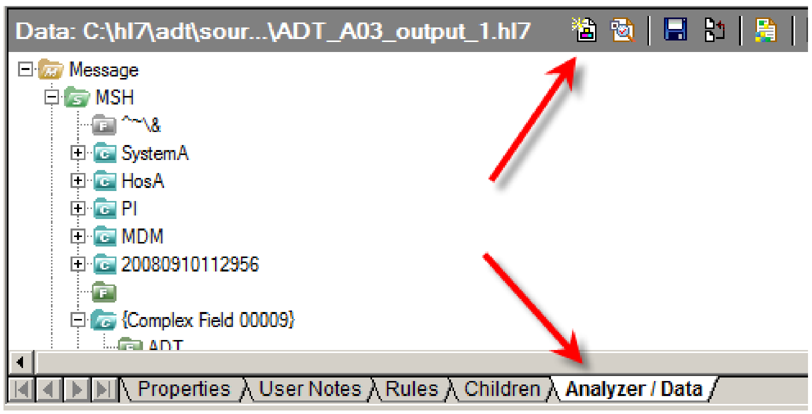
Components:

Children TreeView	No children node.

- ☐ Save the modified guideline

Analyze again using the A03 Sample Message

- ☐ Click on the "Analyzer / Data" Tab then on the "New Data" button to re-start the Data Analyzer

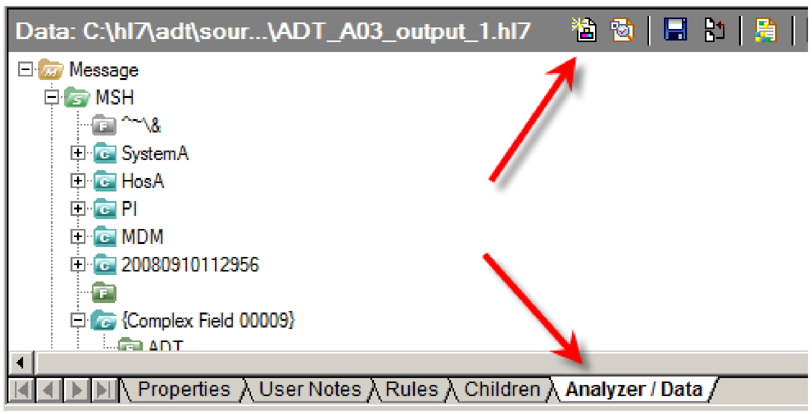


- ☐ Select ADT_A03_output_1.hl7 sample file and click Next

Inspect error messages to see what kinds of issues the analyzer is reporting for the A03 sample message. The only issue reported should be "Event ID mismatch. Event ID from data: A03. Guideline Event ID: A01.". We will ignore this.

Analyze again using the A01 Sample Message and Fix Errors

- ☐ Click on the "New Data" button in the Data pane to re-start the Data Analyzer

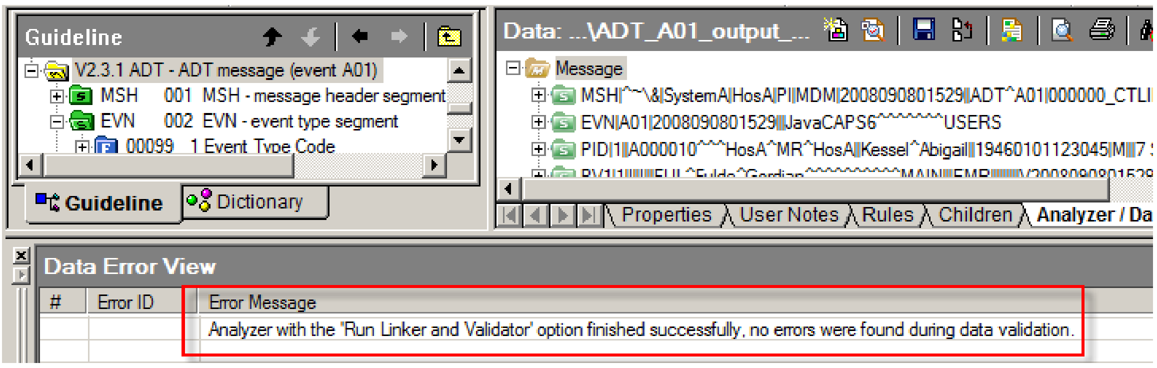


- ☐ Select ADT_A01_output_1.hl7 sample file and click Next

Inspect error messages. The analyzer is reporting Timestamp-related errors for the A01 sample message fields MSH-7.1 and EVN-2.1.

Fix these errors by changing data type for these fields from TS (Timestamp) to String (ST) as discussed above.

Analyze again to make sure all errors are fixed


Guideline
V2.3.1 ADT - ADT message (event A01)
MSH 001 MSH - message header segment
EVN 002 EVN - event type segment
00099 1 Event Type Code
Guideline
Dictionary
Data: ...\ADT_A01_output_...
Message
MSH|^~\&|SystemA|HosA|PI|MDM|2008090801529||ADT^A01|000000_CTLI
EVN|A01|2008090801529|||JavaCAPS6^^^^^^USERS
PID|1||A000010^^^HosA^MR^HosA||Kessel^Abigail||19460101123045|M|||7
Properties
User Notes
Rules
Children
Analyzer / Da

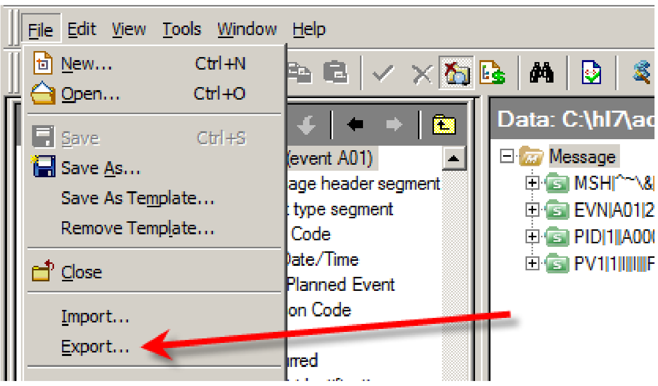
Data Error View

#	Error ID	Error Message
		Analyzer with the 'Run Linker and Validator' option finished successfully, no errors were found during data validation.

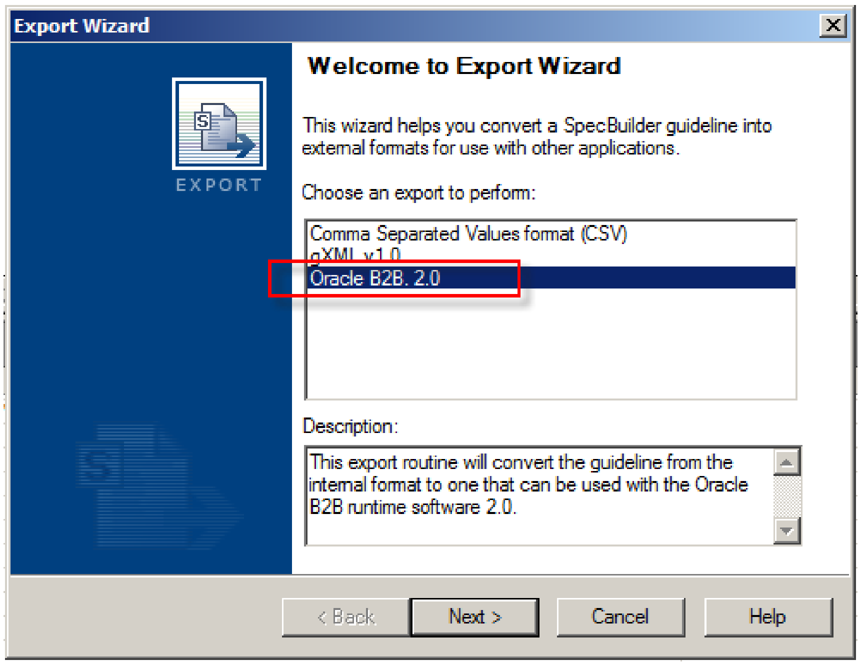
☐ Save guideline file

Export CMM in ECS and XSD forms

☐ Pull down the File menu and chose "Export"



☐ Choose "Oracle B2B" and click Next


Export Wizard

Welcome to Export Wizard

This wizard helps you convert a SpecBuilder guideline into external formats for use with other applications.

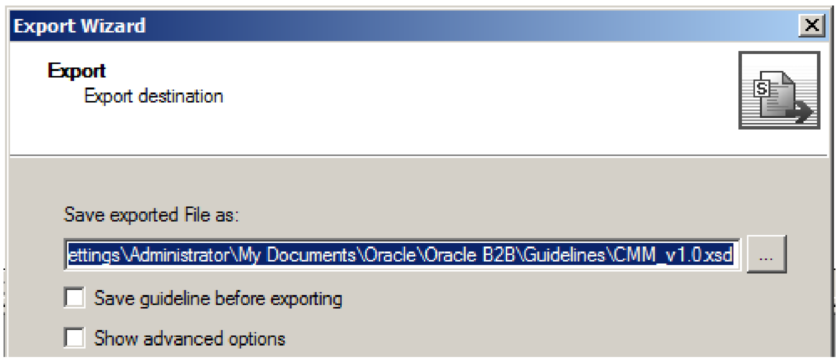
Choose an export to perform:

- Comma Separated Values format (CSV)
- gXML v1.0
- Oracle B2B. 2.0

Description:

This export routine will convert the guideline from the internal format to one that can be used with the Oracle B2B runtime software 2.0.

☐ Accept the default name and location, which will be the same location as the ECS file and the same name but with the .xsd extension, by licking Next.


Export Wizard

Export

Export destination

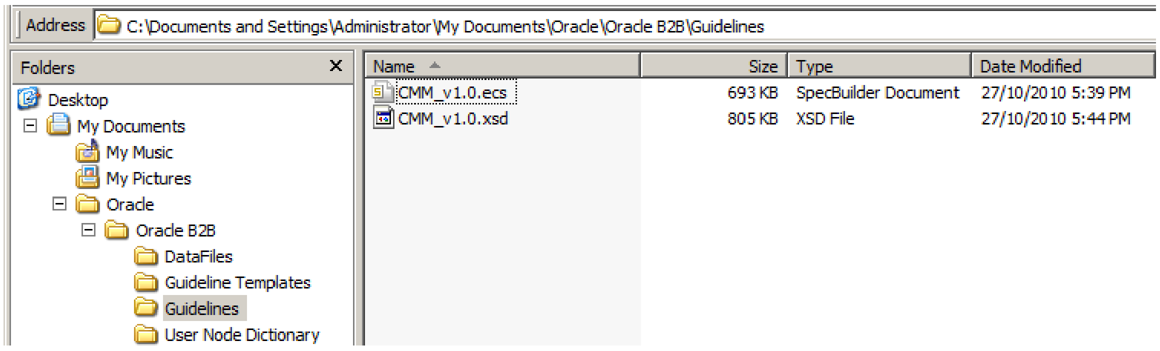
Save exported File as:

ettings\Administrator\My Documents\Oracle\Oracle B2B\Guidelines\CMM_v1.0.xsd

☐ Save guideline before exporting

☐ Show advanced options

- ☐ Click Finish
- ☐ Close all Oracle Document Editor windows, saving ECS file if necessary


Address C:\Documents and Settings\Administrator\My Documents\Oracle\Oracle B2B\Guidelines

Folders
- Desktop
  - My Documents
    - My Music
    - My Pictures
    - Oracle
      - Oracle B2B
        - DataFiles
        - Guideline Templates
        - Guidelines
        - User Node Dictionary

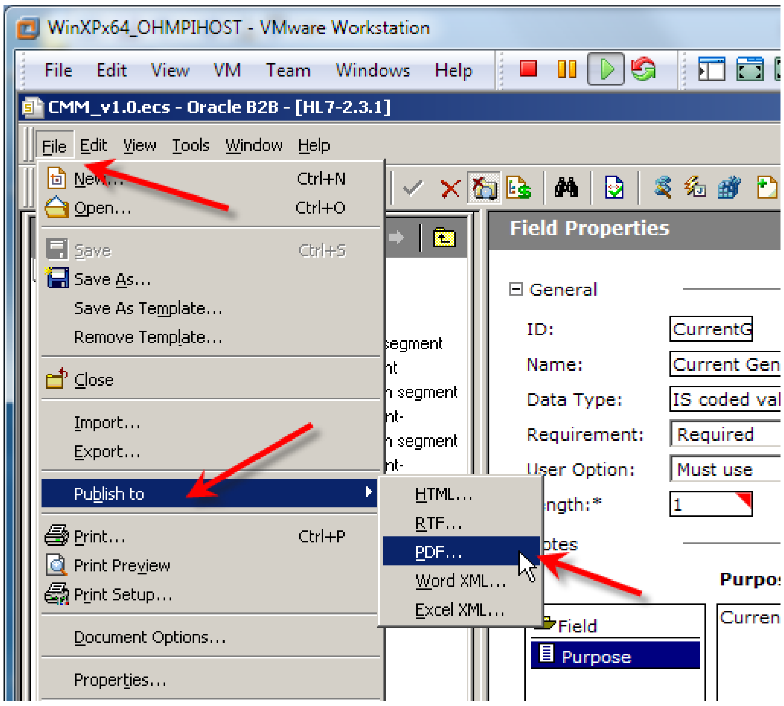
Name	Size	Type	Date Modified
CMM_v1.0.ecs	693 KB	SpecBuilder Document	27/10/2010 5:39 PM
CMM_v1.0.xsd	805 KB	XSD File	27/10/2010 5:44 PM

We are done modifying our message structure. The canonical message model is ready and available for use in ECS and XSD forms. This message structure can be used to work with our sample A01 and A03 messages.

Generate Message Structure Documentation

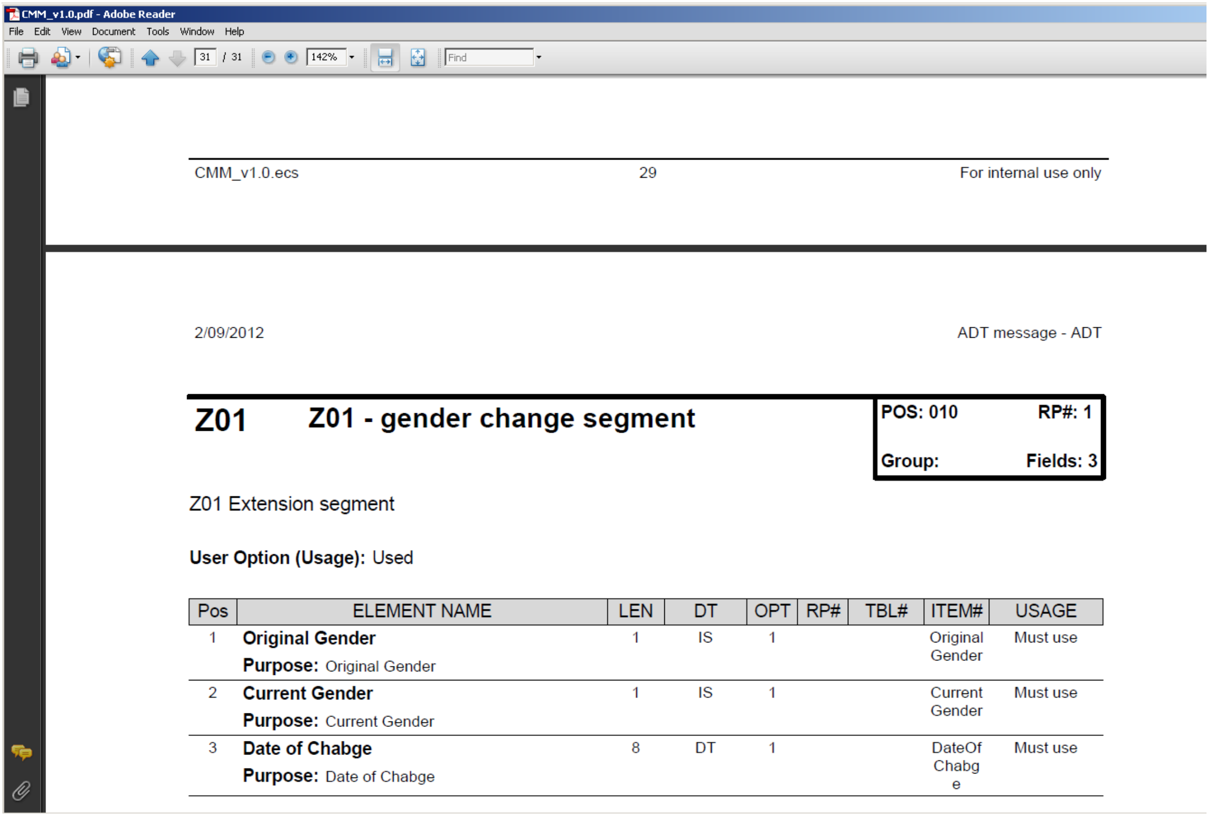
Oracle Document Editor allows message structure to be extensively documented, with notes on segments, fields, components and subcomponents, which we saw but ignored as we worked through the steps in the previous section. We can now use the Oracle Document Editor functionality to produce message structure reports which we can provide to interface developers and other whoi have a need to know the details of message structures they might be using.

- ☐ With the CMM_v1.0.ecs open in the Oracle Document Editor pull down the “File” menu, select “Export to” and click “PDF…”



☐ Navigate to the location where you want the PDF file to be stored, give it a name, for example "CMM_v1.0.pdf" and click "Save"

☐ Inspect the PDF file to see what it contains, noting that the Z segment section is pretty sparsely populated, compared to other sections, because we did not provide documentation of the segment and its component fields as we developed it. We hopefully will be more conscientious next time ☺


CMM_v1.0.pdf - Adobe Reader

CMM_v1.0.ecs 29 For internal use only

2/09/2012 ADT message - ADT

Z01 **Z01 - gender change segment**

POS: 010 **RP#: 1**
Group: **Fields: 3**

Z01 Extension segment

User Option (Usage): Used

Pos	ELEMENT NAME	LEN	DT	OPT	RP#	TBL#	ITEM#	USAGE
1	**Original Gender** **Purpose:** Original Gender	1	IS	1			Original Gender	Must use
2	**Current Gender** **Purpose:** Current Gender	1	IS	1			Current Gender	Must use
3	**Date of Chabge** **Purpose:** Date of Chabge	8	DT	1			DateOf Chabg e	Must use

☐ Publish the CMM_v1.0 to "HTML…" and "Excel XML…" and inspect the results if you have the tools to do so

- ☐ Close all Oracle Document Editor windows – we are done

Summary

This article worked through the mechanics of deriving a Canonical Message Model for the series of articles on SOA Suite for healthcare integration using the “Oracle SOA Suite for healthcare integration” tooling. It contains an abridged version of the article “Healthcare Enterprise – IT Architecture Building Blocks – Canonical Message Model for a HL7 Enterprise”, available at http://blogs.czapski.id.au/2010/10/healthcare-enterprise-%e2%80%93-it-architecture-building-blocks-canonical-message-model-for-a-hl7-enterprise, but added new material related to testing the canonical structure against sample data and modifying the structure to accommodate data idiosyncrasies.