

CT07:



Patient Narratives – Speeding Up Narrative Generation Using ODS Document Stores

Nev Cope

Standards, Automation and Specialist Programming (SASP), Clinical Data and Insights (CDI)
Development Operations, R&D, AstraZeneca, Cambridge, UK



13–16 November
ICC Belfast

EU 2022

The Clinical Data
Science Conference

Narratives

- Mandated by ICH-E3 guidelines (Section 12.3.2), which state that deaths, SAEs or other significant Adverse Events should be documented in a patient narrative
- Programs used to generate narratives typically get copied from one study to another for many years.
- This can lead to both programming stagnation and execution inefficiencies
- Both of these outcomes had been reached within the AstraZeneca programming teams, and so I was tasked with investigating ways to improve narrative generation across different therapeutic areas.

Please note, all patient data displayed in this presentation is simulated.



What are they?

- A summary of a patient's data in either compact tables, or full programmatically-generated sentences.
- The actual layout of a narrative can vary between TAs
 - The TAs at AstraZeneca each have separate requirements for narratives
- For example:
 - CVRM = 13 tables,
 - R&I = 2 tables and 22 sentences
 - Oncology = 19 tables and 41 sentences

Protocol: D#####
<Safety/All> Population

<USUBJID> Narrative

<Table 1>

Protocol: DXXXXXXXXXX	Subject Enrolment Code: E#####
Country: France	Age/Sex/Race/Ethnicity: 48 years/Male/White/Not Hispanic or Latino
Investigational Product and Dosing Regimen:	Adverse Event Preferred Term: <preferred term> (<category>) Angioedema (SAE) Anaphylactic reaction (Anaphylactic reactions) Hypersensitivity (SAE) Anaphylactic reaction (Anaphylactic reactions) <Sort by STUDYID, USUBJID, AESTDTC, AEENRTPT, AEENDTC AEDECOD and only display the Adverse Events triggered by a narrative category with the categories displayed in parenthesis following the adverse event>

<Table 2>

Adverse Events <Provide a complete chronological list of only adverse events triggering a narrative>

Preferred Term	Event Start Date	Event Stop Date	Related (Yes/No)	Seriousness (Yes/No)	Action Taken with IP	Outcome	Severity
Adverse Event 1	07-Nov-2017	09-Nov-2017	No	Yes	Dose not changed	Recovered/resolved	Moderate
Adverse Event 2	28-Nov-2017	02-Dec-2017	No	Yes	Dose not changed	Recovered/resolved	Severe
Adverse Event 3	05-Jan-2018	05-Jan-2018	No	Yes	Dose not changed	Recovered/resolved	Mild
Adverse Event 4	15-Feb-2018	16-Feb-2018	No	Yes	Not applicable	Recovered/resolved	Moderate

Paragraph 1 includes sentence 1

<Sentence 1>

Subject <Subject ID>, a <Age>-<Age units>-old <race> <gender> from <Country>, with <Disease Under Study (DUS)> diagnosed on <Date DUS First Diagnosed>, was screened in <Study ID> study on <Date of Informed Consent>.

Paragraph 2 includes sentence 2

<For sentence 2, display the code for either 2a or 2b depending on the data for each subject>

<Sentence 2>

The subject

- <If randomised to blinded treatment, display the following: 'was randomised to '>
- <If randomised to open label treatment, display the following: 'initiated treatment with '>

<Sentence 2a><If the randomisation date and the first IP dose date are the same, display the following (similar to event-based narrative)> <Investigational product dose and dosing regimen> and received the first dose on <dosing start date>.

<Sentence 2b><If the randomisation date and the first IP dose date are different, display the following> <Investigational product dose and dosing regimen> on <date randomised> and received the first dose on <dosing start date>.

Paragraph 3 includes sentence 3

<Sentence 3>

The subject's medical history included <derived term> <continue to list all (current, past and other) medical history followed by a comma without start/stop dates unless otherwise specified. Prior to the last value, display 'and' and the last medical history value> < If there is no medical history, display 'The subject had no medical history;'>.

Paragraph 4 includes sentence 4

<Sentence 4>

The subject's surgical history included <derived term>, <continue to list all surgical history followed by a comma without start dates. Prior to the last value, display 'and' and the last surgical history value>. <If there is no past surgical history, display 'The subject had no surgical history.'>



Legacy Programs

- The sequential nature of the output structure may seem straightforward to begin with, but on closer scrutiny of the templates complications emerge.
- The programming complexity was increased by having to mix tabular and textual elements on a page
- The 3 different TA code libraries that I was involved in reviewing were maintained independently, sometimes at the study level
- Results tended to be stored in separate data sets, one for each table/sentence. Then hard-coded PROC REPORT calls sequentially generated each output element while writing to an ODS destination, one subject at a time.
- This was slow and unwieldy.
If only there was a better way....?



Introducing – ODS Document Stores

- Early versions of SAS had limited options for output, mostly just flat text
- V8.0 introduced the full production version of the Output Delivery System (ODS)
- ODS allows SAS to route your output to different multiple file types (RTF, PDF, etc)
- V9 (?) Lets you capture your output in a new native SAS format: the ODS Document Store.
- Document Store entries can be manipulated and then output to any ODS destination



ODS Document Store Contents

```
ods document name=work.demostore;  
proc report data=sashelp.class;  
  column name sex age  
           height weight;  
  define name      / display;  
  define sex       / display;  
  define age       / display;  
  define height    / display;  
  define weight    / display;  
run;  
ods document close;
```



	S			
Name	x	Age	Height	Weight
Alfred	M	14	69	112.5
Alice	F	13	56.5	84
Barbara	F	13	65.3	98
Carol	F	14	62.8	102.5
Henry	M	14	63.5	102.5
James	M	12	57.3	83
Jane	F	12	59.8	84.5
Janet	F	15	62.5	112.5
Jeffrey	M	13	62.5	84
John	M	12	59	99.5
Joyce	F	11	51.3	50.5
Judy	F	14	64.3	90
Louise	F	12	56.3	77
Mary	F	15	66.5	112
Philip	M	16	72	150
Robert	M	12	64.8	128
Ronald	M	15	67	133
Thomas	M	11	57.5	85
William	M	15	66.5	112



Document Store Contents

```
ods output properties=results;  
proc document name=work.demostore;  
    list / levels=all;  
quit;
```



Listing of: \Work.Demostore\ Order by: Insertion Number of levels: All		
Obs	Path	Type
1	\Report#1	Dir
2	\Report#1\Report#1	Dir
3	\Report#1\Report#1\Report#1	Table

NOTE:

The document store contains internal directories...



Document Store entry with a BY (1)

```
proc sort data=sashelp.class
  out=work.class;
  by name;
run;

ods document name=work.demostore;
  proc report data=work.class;
    by name;
    column name sex age height weight;
    define name / display;
    define sex / display;
    define age / display;
    define height / display;
    define weight / display;
  run;
ods document close;

ods output properties=results;
proc document name=work.demostore;
  list / levels=all;
quit;
```

Listing of: \Work.Demostore\
Order by: Insertion
Number of levels: All

Obs	Path	Type
1	\Report#1	Dir
2	\Report#1\ByGroup1#1	Dir
3	\Report#1\ByGroup1#1\Report#1	Dir
4	\Report#1\ByGroup1#1\Report#1\Report#1	Table
5	\Report#1\ByGroup2#1	Dir
6	\Report#1\ByGroup2#1\Report#1	Dir
7	\Report#1\ByGroup2#1\Report#1\Report#1	Table
8	\Report#1\ByGroup3#1	Dir
9	\Report#1\ByGroup3#1\Report#1	Dir
10	\Report#1\ByGroup3#1\Report#1\Report#1	Table
11	\Report#1\ByGroup4#1	Dir
12	\Report#1\ByGroup4#1\Report#1	Dir
13	\Report#1\ByGroup4#1\Report#1\Report#1	Table
14	\Report#1\ByGroup5#1	Dir
15	\Report#1\ByGroup5#1\Report#1	Dir
16	\Report#1\ByGroup5#1\Report#1\Report#1	Table
17	\Report#1\ByGroup6#1	Dir
18	\Report#1\ByGroup6#1\Report#1	Dir
19	\Report#1\ByGroup6#1\Report#1\Report#1	Table
20	\Report#1\ByGroup7#1	Dir
21	\Report#1\ByGroup7#1\Report#1	Dir
22	\Report#1\ByGroup7#1\Report#1\Report#1	Table
23	\Report#1\ByGroup8#1	Dir
24	\Report#1\ByGroup8#1\Report#1	Dir
25	\Report#1\ByGroup8#1\Report#1\Report#1	Table



Manipulate the Document Store (code)

```
data exec;
  set work.class end=eof;
  bygroup=_n_;
  length exec $200;
  if _n_=1 then do;
    exec="proc document name=work.demostore;";
    output;
    exec="make Boys; make Girls;";
    output;
  end;
  exec=catx(" ", "obpage", cats("\Report#1\ByGroup", bygroup, "#1\Report#1\Report#1 / delete;"));
  output;
  exec=catx(" ", "rename", cats("\Report#1\ByGroup", bygroup, "#1\Report#1\Report#1"), "to", name, ";");
  output;
  if sex="M" then exec=catx(" ", "move", cats("\Report#1\ByGroup", bygroup, "#1\Report#1\Report#1", name), "to \Boys#1;");
  else if sex="F" then exec=catx(" ", "move", cats("\Report#1\ByGroup", bygroup, "#1\Report#1\Report#1", name), "to \Girls#1;");
  output;
  if eof then do;
    exec="delete \Report#1;";
    output;
    exec="run;";
    output;
  end;
run;

data _null_;
  set exec;
  call execute(exec);
run;
```

Create a Boys and Girls folder

Remove page breaks

Move to relevant folder

Delete old folders

Rename entry to child's name

Listing of: \Work.Demostore\
Order by: Insertion
Number of levels: All

Obs	Path	Type
1	\Report#1	Dir
2	\Report#1\ByGroup1#1	Dir
3	\Report#1\ByGroup1#1\Report#1	Dir
4	\Report#1\ByGroup1#1\Report#1\Report#1	Table
5	\Report#1\ByGroup2#1	Dir
6	\Report#1\ByGroup2#1\Report#1	Dir
7	\Report#1\ByGroup2#1\Report#1\Report#1	Table
8	\Report#1\ByGroup3#1	Dir
9	\Report#1\ByGroup3#1\Report#1	Dir
10	\Report#1\ByGroup3#1\Report#1\Report#1	Table
11	\Report#1\ByGroup4#1	Dir
12	\Report#1\ByGroup4#1\Report#1	Dir
13	\Report#1\ByGroup4#1\Report#1\Report#1	Table
14	\Report#1\ByGroup5#1	Dir
15	\Report#1\ByGroup5#1\Report#1	Dir



Manipulate the Document Store (executed)

```
proc document name=work.demostore;  
  make Boys;  
  make Girls;  
  obpage \Report#1\ByGroup1#1\Report#1\Report#1 / delete;  
  rename \Report#1\ByGroup1#1\Report#1\Report#1 to Alfred;  
  move \Report#1\ByGroup1#1\Report#1\Alfred to \Boys#1;  
  
  obpage \Report#1\ByGroup2#1\Report#1\Report#1 / delete;  
  rename \Report#1\ByGroup2#1\Report#1\Report#1 to Alice;  
  move \Report#1\ByGroup2#1\Report#1\Alice to \Girls#1;  
  <AND SO ON...>  
  delete \Report#1;  
run;
```

Listing of: \Work.Demostore\
Order by: Insertion
Number of levels: All

Obs	Path	Type
1	\Boys#1	Dir
2	\Boys#1\Alfred#1	Table
3	\Boys#1\Henry#1	Table
4	\Boys#1\James#1	Table
5	\Boys#1\Jeffrey#1	Table
6	\Boys#1\John#1	Table
7	\Boys#1\Philip#1	Table
8	\Boys#1\Robert#1	Table
9	\Boys#1\Ronald#1	Table
10	\Boys#1\Thomas#1	Table
11	\Boys#1\William#1	Table
12	\Girls#1	Dir
13	\Girls#1\Alice#1	Table
14	\Girls#1\Barbara#1	Table
15	\Girls#1\Carol#1	Table
16	\Girls#1\Jane#1	Table
17	\Girls#1\Janet#1	Table
18	\Girls#1\Joyce#1	Table
19	\Girls#1\Judy#1	Table
20	\Girls#1\Louise#1	Table
21	\Girls#1\Mary#1	Table



Generate Output

```
*--- Output Boys ---;
ods rtf file="<PATH>/boys.rtf" style=Harvest;
  proc document name=work.demostore;
    replay Boys / levels=all;
  quit;
ods rtf close;

*--- Output Girls ---;
ods rtf file="<PATH>/girls.rtf" style=Ocean;
  proc document name=work.demostore;
    replay Girls / levels=all;
  quit;
ods rtf close;
```

Name	Sex	Age	Height	Weight
Alfred	M	14	69	112.5

Name	Sex	Age	Height	Weight
Henry	M	14	63.5	102.5

Name	Sex	Age	Height	Weight
James	M	12	57.3	83

Name	Sex	Age	Height	Weight
Jeffrey	M	13	62.5	84

Name	Sex	Age	Height	Weight
John	M	12	59	99.5

Name	Sex	Age	Height	Weight
Philip	M	16	72	150

Name	Sex	Age	Height	Weight
Robert	M	12	64.8	128

Ronald	M	15	67	133
--------	---	----	----	-----

Name	Sex	Age	Height	Weight
Thomas	M	11	57.5	85

Name	Sex	Age	Height	Weight
William	M	15	66.5	112

Name	Sex	Age	Height	Weight
Alice	F	13	56.5	84

Name	Sex	Age	Height	Weight
Barbara	F	13	65.3	98

Name	Sex	Age	Height	Weight
Carol	F	14	62.8	102.5

Name	Sex	Age	Height	Weight
Jane	F	12	59.8	84.5

Name	Sex	Age	Height	Weight
Janet	F	15	62.5	112.5

Name	Sex	Age	Height	Weight
Joyce	F	11	51.3	50.5

Name	Sex	Age	Height	Weight
Judy	F	14	64.3	90

Name	Sex	Age	Height	Weight
Louise	F	12	56.3	77

Name	Sex	Age	Height	Weight
Mary	F	15	66.5	112



Summary

- Document Stores are a way to “capture” ODS output to a SAS file
- Items in the store can be manipulated and reordered
- All, or parts, of the Document Store can then be replayed to any ODS destination, with any ODS style as normal
- Can save time on very large projects where you want to report the data once and then rearrange/combine the output



AZ Narratives

- Create data for each element
- Output each element
 - PROC REPORT (tables)
 - PROC ODSTEXT (sentences)
- Manipulate output
- Write to master Document Store
- Generate outputs



Example

Subject 1 – Table 1

Subject 2 – Table 1

Subject 3 – Table 1

Subject 4 – Table 1

Subject 5 – Table 1

Subject 1 – Table 2

Subject 2 – Table 2

Subject 3 – Table 2

Subject 4 – Table 2

Subject 5 – Table 2

Subject 1 – Table 3

Subject 2 – Table 3

Subject 3 – Table 3

Subject 4 – Table 3

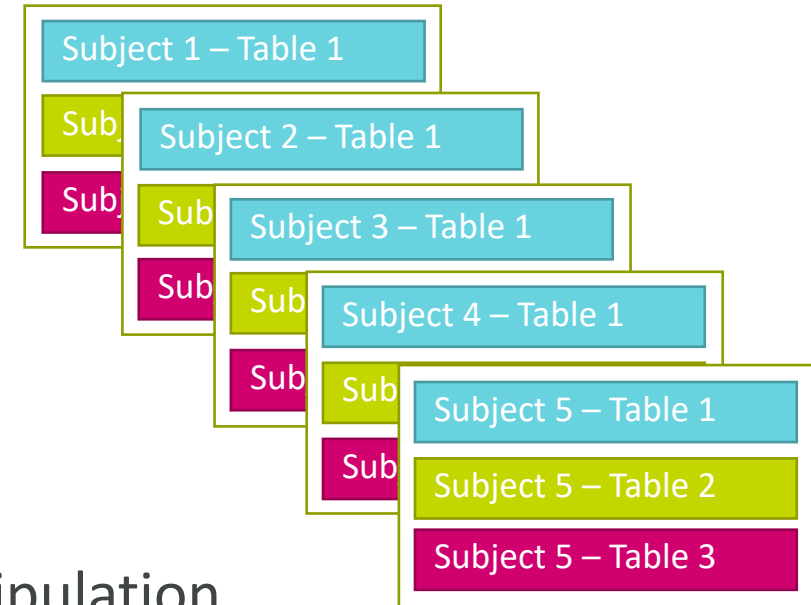
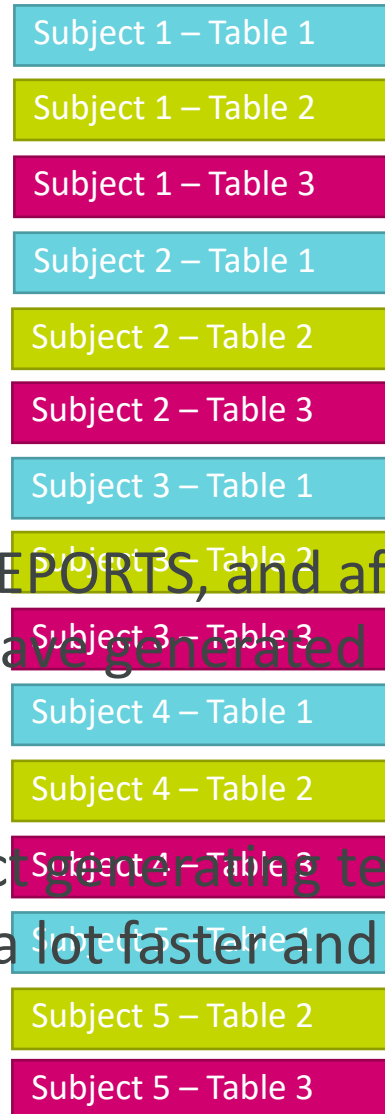
Subject 5 – Table 3

- Arrange data by element, i.e.
 - All Table 1 info in a data set
 - All Table 2 info in a data set
 - All Table 3 info in a data set
- PROC REPORT, with BY, for each element
 - Capturing in Document Stores
- Rearrange document stores to be by **Subject**, not element



Example

- Then REPLAY each subject to a separate RTF file
- So I started with 3 PROC REPORTS, and after manipulation of the Document Stores, have generated 5 RTF files
- On a large narrative project generating tens of tables/sentences over hundreds of subjects this made things a lot faster and easier!



Questions?



Confidentiality Notice

This file is private and may contain confidential and proprietary information. If you have received this file in error, please notify us and remove it from your system and note that you must not copy, distribute or take any action in reliance on it. Any unauthorized use or disclosure of the contents of this file is not permitted and may be unlawful. AstraZeneca PLC, 1 Francis Crick Avenue, Cambridge Biomedical Campus, Cambridge, CB2 0AA, UK, T: +44(0)203 749 5000, www.astrazeneca.com

