

# Surviving Septic Shock

## How SAS® Helped a Critical Care Nursing Staff Fulfill Its Septic Shock Reporting Requirements

Joe Palmer, OhioHealth, Columbus OH

### ABSTRACT

Sepsis is the result of a body's response to a bacterial or fungal infection and often results in organ damage and subsequent mortality. Starting in October 2015, The Centers for Medicare & Medicaid Services (CMS) requires hospitals to collect and submit data for patients that have severe sepsis and septic shock. Reporting requirements include pinpointing the time in which 2 indicators of Systemic Inflammatory Response Syndrome (SIRS) and 1 indicator of organ failure occur within a 6 hour period. In total, there are 4 indicators of SIRS and 8 indicators of organ failure. In addition, hospitals must report the time in which treatment was started. Manually looking up diagnostic and treatment times in the electronic medical record (EMR) system of a hospital is a lengthy process, even for an experienced nurse. A SAS program was created to query clinical data in a database fed by an electronic medical record system. The program generates an automated report which contains the required diagnosis and treatment times, thus allowing nurses to assess required data in less than 15 minutes per patient.

### INTRODUCTION

Sepsis is a body's immune over-reaction to a bacterial or fungal infection. Widespread inflammation restricts blood flow to organs, resulting in organ failure and potentially death. It is estimated that 2% of all hospital admissions and 10 percent of critical care admissions are due to sepsis. Early diagnosis and treatment of sepsis is a key to survival of this condition. Currently, patients with severe sepsis or septic shock have a 20-30 percent chance of dying, and each hour delay of treatment results in a 7.6 percent reduction in survival<sup>1</sup>. Therefore, rapid diagnosis of sepsis is a fundamental part of treating sepsis.

In October of 2015, the Centers for Medicare and Medicaid Services (CMS) initiated new sepsis reporting requirements for hospitals. Nursing staff are required to collect and record (abstract) information from the medical records of sepsis patients. Each hospital sends patient data to CMS, which tracks the time from diagnosis to treatment. Table 1 includes a list of data elements nurses must abstract from sepsis patients' medical records. CMS requires hospitals to report: 1) the time in which a physician notes a suspected source of infection, 2) Two indications of systemic inflammatory response syndrome (SIRS), and 3) evidence of organ failure. Each of these must occur within 6 hours of each other. In addition, hospitals must report treatment times, as shown in Table 2.

Even if a hospital is reporting data for a sample of their sepsis patients, up to 20-25 patients may have to be abstracted every month. At OhioHealth, the manual process of collecting relevant data takes up to 8 hours per patient because the data are not available in one convenient location within the electronic medical record system (EMR). For instance, Figure 1 is a screen shot of the vitals screen where pulse and respiration are found. Since medical record data are electronic, SAS has the flexibility and power to provide an excellent method of gathering all data into one central location.

### ELECTRONIC DATA COLLECTION

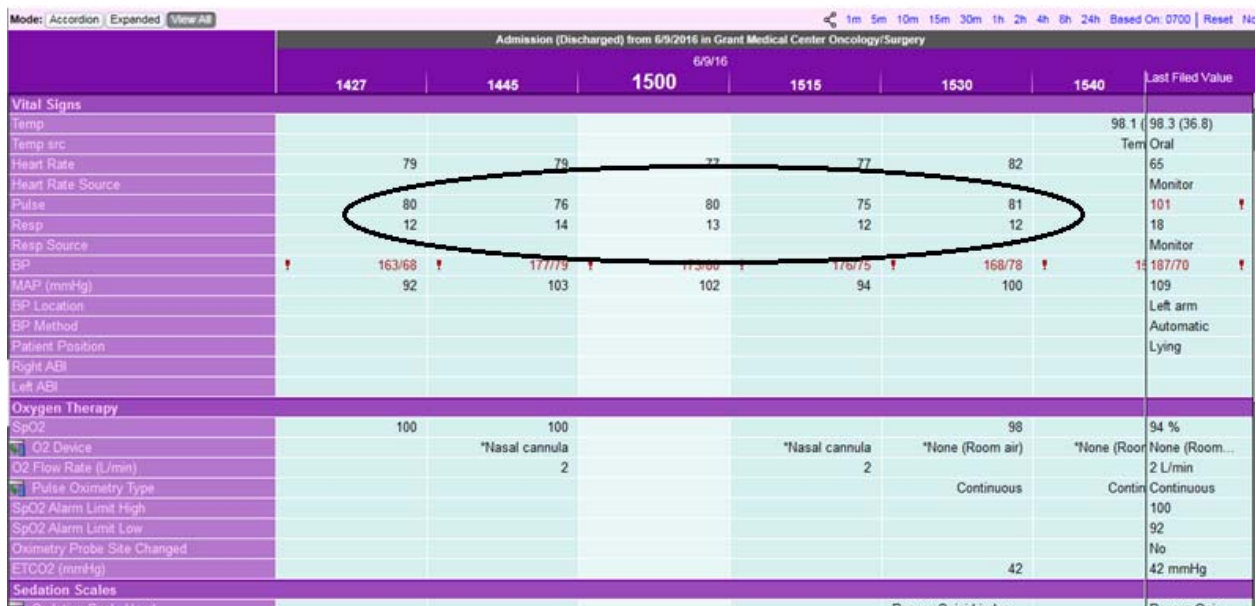
In July of 2014, OhioHealth migrated to EPIC®, an EMR system that includes modules for every aspect of patient care. Seven of OhioHealth's 10 hospitals are currently utilizing EPIC. Every night new data from EPIC flows into a SQL server database called Clarity. The amount of data available in the Clarity database is immense. Every procedure, vital sign, lab result, provider note, patient question/answer and order are recorded in database tables. PROC SQL was used to query the Clarity database for each of the

Source of Infection	Any 2 SIRS criteria	Any 1 Organ Failure
Confirmed or Suspected	Body temperature >38.3 C (100.9 °F) or <36.0 C (98.6 °F)	Systolic blood pressure < 90 or decrease of more than 40 points
	Heart Rate > 90	Mean Arterial Pressure (MAP) < 65
	Respiratory Rate > 20	Creatinine >2.0 or urine output < 0.5mL/kg/hr for 2 hours
	White Blood Cell Count > 12,000 or < 4,000	Bilirubin > 2 mg/dL
		Platelets < 100,000
		Prothrombin International Normalized Ratio (INR) > 1.5 or Activated
		Partial Thromboplastin Time (aPTT) > 60 seconds
		Lactate > 2 mmol/L

**Table 1. Diagnostic Indicators of Sepsis**

Within 3 Hours	Within 6 Hours
Measure lactate	Repeat lactate if level > 2
Obtain Blood Cultures	
Administer Broad Spectrum Antibiotics	
Administer Fluids	

**Table 2. Treatment of Sepsis**



**Figure 1. Screen shot of vitals, including pulse and respiration**

measures in Tables 1 and 2. In order to increase the efficiency of abstraction, only critical values of the various diagnostic indicators were included in the report. Some of these indicators were contained in flowsheets. For example, Output 1 shows code for systolic blood pressure < 90:

```
proc sql;
CREATE TABLE sepsis_bp AS
select a.HSP_ACCOUNT_ID, g.RECORDED_TIME, g.MEAS_VALUE
from epic.v_pat_enc_hsp a
left join epic.v_ip_flwsht_rec f on a.INPATIENT_DATA_ID = f.INPATIENT_DATA_ID
left join epic.v_ip_flwsht_meas g on f.fsd_id = g.fsd_id
left join epic.v_ip_flo_gp_data h on g.FLO_MEAS_ID = h.FLO_MEAS_ID
where a.HSP_ACCOUNT_ID in(&accts) and h.FLO_MEAS_NAME in('BLOOD PRESSURE');
QUIT;

proc sort nodups data=sepsis_bp;
by HSP_ACCOUNT_ID RECORDED_TIME;
run;

data sepsis_bp2; set sepsis_bp;
where meas_value not in('');
format systolic 8.;
systolic = input(substr(meas_value, 1, (index(meas_value, '/')-1)), $10.);
if systolic <90 then output;
drop meas_value;
run;
```

**Output 1. Code to identify abnormal systolic blood pressure**

Note the use of a macro variable **accts** (highlighted in yellow) to limit values to only the current list of sepsis patients. Other indicators are collected from the results from lab tests. Output 2 is the code that queries abnormal values for creatinine:

```
proc sql;
CREATE TABLE sepsis_creat AS
select a.HSP_ACCOUNT_ID, f.RESULT_TIME, f.ORD_VALUE
from epic.v_pat_enc_hsp a
join epic.v_order_proc e on a.PAT_ENC_CSN_ID = e.PAT_ENC_CSN_ID
join epic.order_results f on e.ORDER_PROC_ID = f.ORDER_PROC_ID and
e.PAT_ENC_CSN_ID = f.PAT_ENC_CSN_ID
join epic.clarity_component g on f.component_id = g.component_id
join epic.v_order_proc_2 h on e.ORDER_PROC_ID = h.ORDER_PROC_ID
where a.HSP_ACCOUNT_ID in(&accts) and g.name in('CREATININE') and
f.LAB_STATUS_C in(3);
QUIT;

proc sort nodups data=sepsis_creat;
by HSP_ACCOUNT_ID RESULT_TIME;
run;

data sepsis_creat2; set sepsis_creat;
where ORD_VALUE not in('', '<0.20');
format creat 8.2;
creat = input(compress(ORD_VALUE, '<>'), $10.);
if creat >2 then output;
run;
```

**Output 2. Code to identify abnormal creatinine levels**

Treatment times are obtained from the pharmacy orders within the Clarity database. The code in Appendix 1 shows how data for several types of antibiotics are collected and organized into different antibiotic types. Appendix 1 also includes code to pull the administration time of fluids.

## SUMMARIZATION AND OUTPUT OF DATA

Once all of the necessary data is collected from the Clarity database, they are summarized to fit into one EXCEL® spreadsheet per patient. Each spreadsheet contains a header with the Medical Record Number, patient name, account number, discharge disposition (alive or expired), and admit date/time of the patient. Following the header is a table with the following columns:

1. Measure: The name of the data element, e.g. Blood Pressure, Heart Rate, Antibiotic (ATB), etc.
2. Measurement: The value for each data element
3. Date and Time Collected: Date/time of collection for lab values such as blood cultures, lactate or white blood cell counts
4. Date and Time Recorded/Med Admin Start: For vitals, this is the time of the measurement and for labs, the time a specimen was taken. For antibiotics and fluids, this is the time treatment started.
5. Med Admin End Date/Time: The time fluids ended

Summarization of SIRS and organ failure criteria required separation of dates and times into distinct fields. This was also the case for the treatment data. The code to summarize the patient data is in Appendix II.

In order to make the report creation process as efficient as possible, a macro is used to run the output code for each distinct patient. This way, the lists are created dynamically and without manual intervention. PROC REPORT and tagsets are used to output the spreadsheets to Excel (see Appendix III). The sheet\_interval property is set to "none" (highlighted in yellow) to allow a table to be placed under each patient header. Each spreadsheet is dynamically created by first counting the number of distinct patients into a macro variable **patcnt**:

```
proc sql noprint;
select count(*)
into :patcnt
from pt_list_hdr1;
quit;
%put &patcnt;
```

**Output 3. Count each distinct patient into a macro variable called patcnt**

Next, the code creates a list of medical record numbers and reads them into macro variables **pat1** - **patcnt**:

```
proc sql noprint;
select cats("pat",trim(put(count(*),8.)))
into :patcnt
from pt_list_hdr1;
quit;
%put &patcnt;

PROC SQL noprint;
select pt_name_mrn
into :pat1 - :&patcnt
from pt_list_hdr1;
quit;
```

**Output 4. Create a list of macro variables, each with a medical record number**

Finally, the following code runs the PROC REPORT macro from Appendix III for each medical record number:

```
%macro runall;
%do i=1 %to &patcnt;
%xpwt("&&pat&i");
%end;
run;
%mend;

%runall;
```

Output 5. Run the output macro for each medical record number

## FINAL PRODUCT

Figure 2 shows a partial example of an Excel spreadsheet created for one patient (de-identified):

SIRS and Organ Failure Critical Resuns							
MRN: 333333333							
Patient Name: DOE, JANE							
Account: 777777777							
Discharged Alive							
13AUG16:11:41							
Detail							
Measure	Measurement	Date Collected	Time Collected	Recorded/Med Admin Start	Time Recorded/Med Admin Start	Med Admin End Date	Med Admin End Time
ATB-Ampicillin	Monotherapy	.	.	07/10/2016	16:00	.	.
ATB-Piperacillin	Monotherapy	.	.	07/10/2016	22:23	.	.
ATB-Piperacillin	Monotherapy	.	.	07/08/2016	6:51	.	.
ATB-Vancomycin	Poly-Glycopeptides	.	.	07/10/2016	22:51	.	.
Blood Culture	Negative	07/08/2016	0:15	07/13/2016	9:58	.	.
Blood Culture	Negative	07/08/2016	0:59	07/13/2016	9:58	.	.
Blood Culture	Negative	07/16/2016	2:45	07/21/2016	9:58	.	.
Blood Culture	Negative	07/16/2016	14:51	07/21/2016	17:58	.	.
Blood Pressure	89	.	.	07/08/2016	12:00	.	.
Blood Pressure	85	.	.	07/11/2016	4:00	.	.
Blood Pressure	86	.	.	07/11/2016	5:00	.	.
Blood Pressure Drop	98	.	.	07/10/2016	20:15	.	.
CR--Lactated Ringers Bolu	154208356	.	.	07/08/2016	8:37	07/08/2016	9:37
CR--Lactated Ringers Bolu	154298851	.	.	07/08/2016	18:46	07/08/2016	19:46
CR--Lactated Ringers Bolu	155564475	.	.	07/16/2016	6:48	07/16/2016	7:03
CR--NAACL IV Bolus1000 mL	49723084	.	.	07/10/2016	19:41	07/10/2016	20:41
CR--NAACL IV Bolus1000 mL	49723116	.	.	07/10/2016	21:13	07/10/2016	22:01
CR--NAACL IV Bolus1000 mL	154183212	.	.	07/08/2016	0:44	07/08/2016	0:44
CR--NAACL IV Bolus1000 mL	154208353	.	.	07/08/2016	5:32	07/08/2016	5:32
CR--Unknown	154208361	.	.	07/08/2016	12:01	07/08/2016	12:01
Creatinine	2.45	.	.	07/10/2016	20:05	.	.
Heart Rate	122	.	.	07/10/2016	18:02	.	.
Heart Rate	118	.	.	07/10/2016	19:45	.	.
Heart Rate	114	.	.	07/10/2016	21:13	.	.
Heart Rate	109	.	.	07/10/2016	21:58	.	.
Heart Rate	110	.	.	07/10/2016	22:15	.	.
Heart Rate	112	.	.	07/10/2016	22:30	.	.
Heart Rate	109	.	.	07/10/2016	22:46	.	.
Heart Rate	110	.	.	07/10/2016	23:00	.	.
Heart Rate	106	.	.	07/10/2016	23:15	.	.
Heart Rate	117	.	.	07/10/2016	23:45	.	.
Heart Rate	120	.	.	07/10/2016	23:55	.	.
Heart Rate	120	.	.	07/08/2016	0:00	.	.
Heart Rate	130	.	.	07/08/2016	0:15	.	.
Heart Rate	113	.	.	07/08/2016	5:00	.	.
Heart Rate	114	.	.	07/08/2016	5:00	.	.
Heart Rate	118	.	.	07/08/2016	5:30	.	.
Heart Rate	117	.	.	07/08/2016	5:30	.	.

Figure 2: Excel spreadsheet created for a sepsis patient

Notice the header section identifying the patient by medical record number, patient name, account number, disposition, and admission date/time. Also, notice that the measures are sorted alphabetically first, then Recorded/Med Admin Start second. Each nurse has her/his own method of using their spreadsheet. Usually, the spreadsheet is sorted by Recorded/Med Admin Start first, then color coded to

identify measures of SIRS, organ failure, and treatments. Figure 3 is an example of a spreadsheet after it has been sorted and color-coded by a nurse:

Measure	Measurement	Date Collected	Time Collected	Recorded/Measured Admin Start	Time Recorded/Measured Admin Start	Med Admin End Date	Med Admin End Time
Heart Rate	114			05/22/2016	22:15		
CR--NAACL IV Bolus500 mL	146913766			05/22/2016	23:00	05/23/2016	0:00
Heart Rate	104			05/22/2016	23:00		
source	ed doc			5/22/2016	23:08		
INR	4			05/22/2016	23:43		
Lactate	1.2	05/22/2016	22:45	05/22/2016	23:53		
ATB-Cefepime	Monotherapy			05/23/2016	0:00		
Heart Rate	110			05/23/2016	0:39		
Heart Rate	105			05/23/2016	1:00		
ATB-Vancomycin	Poly-Glycopeptides			05/23/2016	1:21		
Heart Rate	95			05/23/2016	3:00		
source	h&P			5/23/2016	4:40		
Heart Rate	92			05/23/2016	6:45		
Heart Rate	113			05/23/2016	9:15		
Heart Rate	104			05/23/2016	9:16		
source	hms progress notes			5/23/2016	10:54		
INR	4.3			05/23/2016	11:14		
Heart Rate	103			05/23/2016	12:32		
Heart Rate	92			05/23/2016	20:28		
Mean Arterial Pressure	64			05/23/2016	20:28		
Heart Rate	95			05/24/2016	0:27		
Heart Rate	93			05/24/2016	4:20		
INR	3.7			05/24/2016	5:43		
Heart Rate	92			05/24/2016	9:54		
source				5/24/2016	10:06		
Heart Rate	92			05/24/2016	11:34		
Heart Rate	95			05/24/2016	15:35		
Heart Rate	94			05/24/2016	19:57		
Heart Rate	95			05/24/2016	23:28		
Heart Rate	92			05/25/2016	4:17		
INR	2.6			05/25/2016	5:52		
ATB-Ceftriaxone	Monotherapy			05/25/2016	15:00		
Heart Rate	104			05/25/2016	15:00		
Heart Rate	93			05/26/2016	0:18		
INR	1.9			05/26/2016	5:04		
Heart Rate	103			05/26/2016	8:00		
Heart Rate	109			05/26/2016	9:48		
Blood Culture	Negative	05/22/2016	22:45	05/28/2016	9:59		
Blood Culture	Negative	05/23/2016	10:40	05/28/2016	13:58		
Blood Culture	Negative	05/23/2016	10:48	05/28/2016	13:58		

Figure 3. Sepsis spreadsheet sorted and coded by diagnosis/treatment category

### CONCLUSION: WHAT SAS HAS ACCOMPLISHED

The new sepsis reporting requirements set forth by CMS in October of 2015 caused considerable consternation among an already overworked nursing staff at OhioHealth. The hospital system had considered hiring contract staff to perform this task, which would have cost \$150,000 a year and potentially resulted in less than accurate measurement collection. However, implementation of the SAS-based report has made that unnecessary. The report itself takes about 30 minutes to generate for all hospitals, and the time necessary to abstract data is at most 15 minutes per patient, down from 8 hours.

In the end, this is not an example of efficient programming or a novel approach to solving a problem. Instead, this is just another example of how the power and flexibility of SAS can be used to save time and money for a corporation or organization.

### REFERENCES

1. Miller, R.R., L. Dong, N. C. Nelson, S. M. Brown, K. G. Kuttler, D. R. Probst, T. L. Allen and T. P. Clemmer. 2013. Multicenter implementation of a severe sepsis and septic shock treatment bundle. American Journal of Respiratory and Critical Care Medicine 188: 77-82.

### ACKNOWLEDGEMENTS

Thank you to Babette Rosselot, Erin Janiak, Michelle Woodham and Kirk Lafler for providing feedback for this paper.

## TRADEMARK INFORMATION

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.

© indicates USA registration.

## CONTACT INFORMATION

Name: Joe Palmer

Company: OhioHealth

Email: joseph.palmer@ohiohealth.com

### Appendix I: Code to pull antibiotics and fluids

```
proc sql;
create table atb as select distinct a.hsp_account_id, b.MedicationID, d.name,
c.TakenTime
from epic.v_pat_enc a join epic.v_rxi_order_med b on a.PAT_ENC_CSN_ID =
b.PAT_ENC_CSN_ID
join epic.v_rxi_mar c on b.OrderMedID = c.OrderMedID
join epic.v_rxi_erx d on b.MedicationID = d.MedicationID
where a.hsp_account_id in(&accts) and c.TakenTime > 0 and c.action = 'New
Bag' and
(d.name like '%Doribax%' or d.name like '%Doripenem%'
or d.name like '%Eratopenem%' or d.name like '%Invanz%'
or d.name like '%Imipenem%' or d.name like '%Cilastatin%'
/* long list of antibiotics */
or d.name like '%Oxacillin%' or d.name like '%Penicillin%'
or d.name like '%Ertapenem%');
quit;

/* get first dispensed time for each antibiotic */
proc sort data=atb;
by hsp_account_id MedicationID TakenTime;
run;

data atb2; set atb;
by hsp_account_id MedicationID;
if first.MedicationID;
/* organize into names and antibiotic types */
format atb_name $30. atb_type $30.;
if index(propcase(name), 'Doribax') > 0 then do;
atb_name = 'ATB-Doribax'; atb_type = 'Monotherapy'; end;
else if index(propcase(name), 'Doripenem') > 0 then do;
atb_name = 'ATB-Doripenem'; atb_type = 'Monotherapy'; end;
else if index(propcase(name), 'Eratopenem') > 0 or
index(propcase(name), 'Ertapenem') > 0 then do;
atb_name = 'ATB-Eratopenem'; atb_type = 'Monotherapy'; end;
else if index(propcase(name), 'Invanz') > 0 then do;
atb_name = 'ATB-Invanz'; atb_type = 'Monotherapy'; end;
else if index(propcase(name), 'Imipenem') > 0 then do;
atb_name = 'ATB-Imipenem'; atb_type = 'Monotherapy'; end;
else if index(propcase(name), 'Cilastatin') > 0 then do;
atb_name = 'ATB-Cilastatin'; atb_type = 'Monotherapy'; end;
/* etc */
```

```

else if index(propcase(name), 'Oxacillin') > 0 then do;
atb_name = 'ATB-Oxacillin'; atb_type = 'Poly-Penicillins'; end;
else if index(propcase(name), 'Penicillin') > 0 then do;
atb_name = 'ATB-Penicillin'; atb_type = 'Poly-Penicillins'; end;
drop name MedicationID;
run;

/* Crystalloid Fluid */
proc sql;
create table cr_fluid as select distinct a.hsp_account_id, b.OrderMedID,
f.TakenTime, f.dose, f.doseunit, e.StartDate, e.EndDate, b.medicationid

from epic.v_pat_enc a join epic.v_rxi_order_med b
on a.PAT_ENC_CSN_ID = b.PAT_ENC_CSN_ID
join epic.v_patient c on a.pat_id = c.pat_id
join epic.v_rxi_order_med e on b.OrderMedID = e.OrderMedID
join epic.v_rxi_mar f on e.OrderMedID = f.OrderMedID
where e.orderstatus in('Completed') and b.medicationid in(400291, 400296,
144789, 4318) and a.hsp_account_id in(&accts) and
f.action in('New Bag', 'Given', 'Bolus');
quit;

data cr_fluid1; set cr_fluid;
format cr_type $25.;
if medicationid in(400291) then cr_type =
'CR--NACL IV Bolus' || compress(dose) || ' ' || compress(doseunit);
else if medicationid in(400296) then cr_type = 'CR--Lactated Ringers Bolus';
else cr_type = 'CR--Unknown';
drop medicationid dose doseunit;
run;

proc sort data=cr_fluid1;
by hsp_account_id TakenTime;
run;

/* vassopressors */
proc sql;
create table vaso as select distinct a.hsp_account_id, e.OrderDisplayName,
d.name, f.TakenTime
from epic.v_pat_enc a join epic.v_rxi_order_med b on
a.PAT_ENC_CSN_ID = b.PAT_ENC_CSN_ID
join epic.v_patient c on a.pat_id = c.pat_id
join epic.v_rxi_erx d on b.MedicationID = d.MedicationID
join epic.v_rxi_order_med e on b.OrderMedID = e.OrderMedID
join epic.v_rxi_mar f on e.OrderMedID = f.OrderMedID
where (
d.name like '%Norepinephrine%' or d.name like '%Levophed%' or
d.name like '%Epinephrine%' or d.name like '%Adrenalin%' or
d.name like '%Phenylephrine%' or d.name like '%Neosynephrine%' or
d.name like '%Dopamine%' or d.name like '%Inotropin%' or
d.name like '%Vasopressin%' or d.name like '%Petressin%')
and a.hsp_account_id in(&accts) and f.action not
in('Paused', 'Canceled Entry', 'Missed')
order by c.pat_mrn_id, a.hsp_account_id, e.StartDate;
quit;

```



```

data vaso1; set vaso;
/* organize into names and antibiotic types */
format vaso_type $25.;
if index(uppercase(name), 'NOREPINEPHRINE')>0 then
vaso_type = 'Vaso--Norepinephrine';
else if index(uppercase(name), 'LEVOPHED')>0 then
vaso_type = 'Vaso--Levophed';
else if index(uppercase(name), 'EPINEPHRINE')>0 then
vaso_type = 'Vaso--Epinephrine';
else if index(uppercase(name), 'ADRENALIN')>0 then
vaso_type = 'Vaso--Adrenalin';
else if index(uppercase(name), 'PHENYLEPHRINE')>0 then
vaso_type = 'Vaso--Phenylephrine';
else if index(uppercase(name), 'NEOSYNEPHRINE')>0 then
vaso_type = 'Vaso--Neosynephrine';
else if index(uppercase(name), 'DOPAMINE')>0 then
vaso_type = 'Vaso--Dopamine';
else if index(uppercase(name), 'INOTROPIN')>0 then
vaso_type = 'Vaso--Inotropin';
else if index(uppercase(name), 'VASOPRESSIN')>0 then
vaso_type = 'Vaso--Vasopressin';
else if index(uppercase(name), 'PETRESSIN')>0 then
vaso_type = 'Vaso--Petressin';
drop orderdisplayname name;
run;

```

## Appendix II: Summarization of diagnostic and treatment data

```

/* Need 2 SIRS */
data sirsl; set sepsis_temp_f2(in=a) sepsis_heart_rate2(in=b)
sepsis_respiration2(in=c) sepsis_wbc2(in=d);
format sirscriterial $25.;
if a then sirscriterial = 'Temperature';
else if b then sirscriterial = 'Heart Rate';
else if c then sirscriterial = 'Respiration';
else if d then sirscriterial = 'WBC';
format sirscresult_dt1 datetime14. coll_dt datetime14.;
if d then sirscresult_dt1 = result_time;
else sirscresult_dt1 = RECORDED_TIME;
coll_dt = SPECIMN_TAKEN_TIME;
drop RECORDED_TIME result_time SPECIMN_TAKEN_TIME;
run;

/* Need 1 Organ Failure */
data of1; set sepsis_bp2(in=a) sepsis_bp4(in=b) sepsis_map2(in=c)
sepsis_creat2(in=d) sepsis_urine3(in=e) sepsis_bilirub2(in=f)
sepsis_platelet2(in=g) sepsis_inr2(in=h) sepsis_aptt2(in=i)
sepsis_lactate2(in=j) sepsis_intub2(in=k);
format of_criteria $25.;
if a then of_criteria = 'Blood Pressure';
else if b then of_criteria = 'Blood Pressure Drop';
else if c then of_criteria = 'Mean Arterial Pressure';
else if d then of_criteria = 'Creatinine';
else if e then of_criteria = 'Urine Volume';
else if f then of_criteria = 'Bilirubin';
else if g then of_criteria = 'Platelets';

```

```

else if h then of_criteria = 'INR';
else if i then of_criteria = 'aPTT';
else if j then of_criteria = 'Lactate';
else if k then of_criteria = 'Intubated';
format of_result_dt datetime14. coll_dt datetime14.;
if a or b or c or k then of_result_dt = RECORDED_TIME;
else of_result_dt = result_time;
coll_dt = SPECIMN_TAKEN_TIME;
drop RECORDED_TIME result_time ORD_VALUE SPECIMN_TAKEN_TIME;
run;

/* Treatment */
data bundle; set blood_culture2(in=a) atb2(in=b) cr_fluid1(in=c)
vasol(in=d);
format bd_criteria $35.;
if a then bd_criteria = 'Blood Culture';
if b then bd_criteria = atb_name;
if c then bd_criteria = cr_type;
if d then bd_criteria = vaso_type;
format bd_result_dt datetime14. coll_dt datetime14. bd_end_dt datetime14.;
bd_result_dt = RESULT_TIME;
coll_dt = SPECIMN_TAKEN_TIME;
if a then bd_result_dt = RESULT_TIME;
else if b or c or d then bd_result_dt = TakenTime;
if c then do;
    bd_end_dt = enddate;
    bd_measure = OrderMedID;
end;
drop result_time SPECIMN_TAKEN_TIME atb_name cr_type vaso_type
TakenTime OrderMedID;
run;

/* bring together SIRS, Organ Failure, Bundles and Alerts */
data pt_list1; set sirs1(rename=(sirs_result_dt1 = result_dt
sirs_criterial = criteria)) of1(rename=(of_result_dt =
result_dt of_criteria = criteria)) bundle(rename=(bd_result_dt =
result_dt bd_criteria = criteria)) alert1(rename=(NOTE_TIME =
result_dt));
format measure $30.;
if temp_c not in(.) then measure = left(put(temp_c,8.2));
else if heart_rate not in(.) then measure = left(put(heart_rate,8.));
else if respiration not in(.) then measure = left(put(respiration,8.));
else if wbc_count not in(.) then measure = left(put(wbc_count,8.2));
else if systolic not in(.) then measure = left(put(systolic,8.));
else if map not in(.) then measure = left(put(map,8.));
else if creat not in(.) then measure = left(put(creat,8.2));
else if urine not in(.) then measure = left(put(urine,8.));
else if bilirubin not in(.) then measure = left(put(bilirubin,8.1));
else if platelets not in(.) then measure = left(put(platelets,8.));
else if inr not in(.) then measure = left(put(inr,8.1));
else if aPTT not in(.) then measure = left(put(aPTT,8.1));
else if lactate not in(.) then measure = left(put(lactate,8.1));
else if urine not in(.) then measure = left(put(urine,8.));
else if intub not in(.) then measure = left(put(intub,8.));
else if culture_result not in('') then measure = culture_result;
else if atb_type not in('') then measure = atb_type;
else if bd_measure not in('') then measure = left(bd_measure);

```

```

format rdate date9. rtime time6.;
rdate = datepart(result_dt);
rtime = timepart(result_dt);
format cdate date9. ctime time6.;
cdate = datepart(coll_dt);
ctime = timepart(coll_dt);

format edate date9. etime time6.;
edate = datepart(bd_end_dt);
etime = timepart(bd_end_dt);

drop temp_c heart_rate respiration wbc_count systolic map creat urine
bilirubin platelets inr aPTT lactate urine culture_result atb_type intub
result_dt coll_dt bd_end_dt bd_measure;
run;

```

### Appendix III: Output Macro

```

%macro xpirt(pat);
ods tagsets.excelxp options (sheet_name=&pat sheet_interval='none');
title1 j=1 color=blue bcolor=white bold height=18pt
"SIRS and Organ Failure Critical Results" ;

proc report noheader data=pt_list_hdr3 nowd split='\ ' headline headskip
style (header)=[fontsize=14pt font_weight=bold color=white]
style(report)=[fontsize=12pt]
style(column)=[font=(Arial, 12pt)];
where pt_name_mrn in(&pat);
column measure;
define measure/display style(column)={cellwidth=3in};
run;
title1;

title2 j=1 color=white bcolor=blue bold height=15pt "Detail";
proc report data=pt_list_dtl nowd split='\ ' headline headskip
style (header)=[fontsize=14pt font_weight=bold color=blue background =
white bordercolor = blue]
style(report)=[fontsize=12pt]
style(column)=[font=(Arial, 12pt)];
where pt_name_mrn in(&pat);
column criteria measure cdate ctime rdate rtime edate etime lac_high
lac_high2;
define criteria/display 'Measure' style(column)={cellwidth=3in};
define measure/display 'Measurement' style(column)={cellwidth=2in};
define cdate/display 'Date Collected' format=mmddy10. style(column)=
{cellwidth=1.5in};
define ctime/display 'Time Collected' format=time6. style(column)=
{cellwidth=2in};
define rdate/display 'Date Recorded/Med Admin Start' format=mmddy10.
style(column)={cellwidth=1.5in};
define rtime/display 'Time Recorded/Med Admin Start' format=time6.
style(column)={cellwidth=2in};
define edate/display 'Med Admin End Date' format=mmddy10. style(column)=
{cellwidth=1.5in};
define etime/display 'Med Admin End Time' format=time6. style(column)=

```

```
{cellwidth=2in};
define lac_high / noprint;
define lac_high2 / noprint;
compute lac_high2;
  if lac_high = 'N' then lac_high2 = 1;
  if lac_high2 = 1 then do;
    call define('_C1_', 'STYLE', 'style={color=blue}');
    call define('_C2_', 'STYLE', 'style={color=blue}');
    call define('_C3_', 'STYLE', 'style={color=blue}');
    call define('_C4_', 'STYLE', 'style={color=blue}');
    call define('_C5_', 'STYLE', 'style={color=blue}');
    call define('_C6_', 'STYLE', 'style={color=blue}');
  end;
endcomp;
run;

title2;
%mend;
```