ABSTRACT

This paper details the creation of a ready-to-publish document with multiple data sources and differing page layouts of the same type of information based on page fit. Each set of information is placed on one, two, or three pages depending on fit, and requires three calls to the REPORT procedure using three different data sets. This solution does not involve ODS DOCUMENT. Instead, we calculate how much horizontal and vertical space the report requires and use the macro language to execute code conditionally. This allows us to dynamically determine which page layout to apply, including setting or suppressing page breaks with ODS STARTPAGE=. This solution makes extensive use of inline formatting as well. The output was produced as an RTF file using SAS® University Edition running on a Microsoft Windows 10 machine.

INTRODUCTION

This is a part of a simplified railroad simulation. This reporting challenge is to create a passenger railroad timetable using a vintage US timetable as the template. There are three distinct pieces of information required: the train schedules themselves, as shown in Figure 1:

![Figure 1: Train Schedule Template](image)

The second piece is the explanation of the symbols and letters within the schedules (Figure 2:)

![Figure 2: Symbols and Letters](image)
Figure 2: Reference Mark Template

The last piece is a list of what type(s) of cars and amenities are available on each train (Figure 3.)

![Reference Mark Template](image)

Figure 3: Accommodations and Amenities Template

Railroads had to print millions of these, so a key requirement was to use as little paper as possible, necessitating the tiny print. The modern method of timetable distribution is via PDF files or the web, so we used a larger, non-proportional font. There is some post-processing with Visual Basic to change the font in the RTF file generated by SAS due to the limited selection of fonts available in SAS University Edition.

PAGE LAYOUTS AND EXAMPLES

The timetable document uses four distinct layouts, determined on the fly.

1. Both directions in one table spanning a page, reference marks/services on same page, shown in Appendix 1.
2. Both directions in two tables, reference marks and services on same page, shown in Appendix 2.
3. One direction on a page with services on the same page, shown in Appendix 3.
4. One direction on a page with services on a separate page, shown in Appendix 4.
**DATA SOURCES**

The document uses data from seven datasets, several of which are created from Excel workbooks. Microsoft Excel was chosen as the user interface for this project because of its general availability. The SAS datasets and their descriptions are in Table 1:

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RR.SERVICES</td>
<td>One record per train, contains train number, train name, how often the train runs, what types of equipment, and how many of each type are used for this train.</td>
</tr>
<tr>
<td>RR.TT_REFMARKS</td>
<td>Standard time table reference marks.</td>
</tr>
<tr>
<td>RR.TT_NOTES_OUT</td>
<td>Timetable reference marks for outbound trains. One record per train, per station.</td>
</tr>
<tr>
<td>RR.TT_NOTES_IN</td>
<td>Timetable reference marks for inbound trains. One record per train, per station.</td>
</tr>
<tr>
<td>RR.TTBASE_OUT</td>
<td>Actual schedule of outbound trains. One record per station per train. Each train’s schedule is a variable in the dataset, and schedules are maintained as SAS time values.</td>
</tr>
<tr>
<td>RR.TTBASE_IN</td>
<td>Actual schedule of inbound trains. One record per station per train. Each train’s schedule is a variable in the dataset. Order of stations is reversed from RR.TTBASE_OUT, schedules are maintained as SAS time values.</td>
</tr>
<tr>
<td>RR.TRAINS2ADD</td>
<td>Extra trains running on “special service” days. Contains train numbers, one record per “special service” date.</td>
</tr>
<tr>
<td>RR.TRAINS2RMV</td>
<td>Trains that will not run on “special service” days. Contains train numbers, one record per “special service” date.</td>
</tr>
</tbody>
</table>

**Table 1 : Data Sources for Timetable**

**CHOOSING A PAGE LAYOUT**

All three sections are required for a complete timetable page. The pages are arranged by route, and we loop through the routes individually with a macro %DO loop. We have set the width of each train schedule column at .65 inches. Given a page width of 8.5 inches, and accounting for margins (.5 inches) and the mandatory STATION column (1.25 inches,) the horizontal space available for train columns is 6.75 inches. Therefore, a maximum of eight schedule columns will fit horizontally on each page (Figure 4):
Figure 4: Maximum Eight Train Columns Across a Page

It took some trial and error to determine that 62 lines will fit on a page vertically. Although each train’s schedule section is displayed vertically, some routes have more than eight trains. PROC REPORT does a good job breaking columns into groups, so we tested to find the maximum number of schedule section columns (7) PROC REPORT will print before breaking by the ID variables. We have to multiply the number of station lines (STACNT, Example 1) by that (SVCMULT, Example 1) to calculate how many lines the schedule section will occupy. We add the number of lines required for the services (SVCLINES) and reference mark sections (REFLINES), and include some slack to ensure we don’t put more information on a page than will fit. Overestimating the number of lines is better than underestimation. Example 1 shows the relevant code to calculate vertical page fit:

```
svcmult = ceil(nservices/7); /* # of grouped schedules on the page */
/* Vertical page length calculation */
pagelen = (svcmult*stacnt)+reflines+svclines+5;
```

Example 1: Calculating Vertical Page Fit

We store those calculations in macro variables, and use them to execute ODS STARTPAGE conditionally, shown in Example 2:

```
/* Approx. 62 lines/pg @ 8.5PT Calibri */
%IF &pagesize LE 62 %THEN %DO;
  ODS RTF STARTPAGE=NO;
  TITLE "#BYVAL2 Services";
%END;
%ELSE %DO;
  ODS RTF STARTPAGE=NOW;
%END;
```

Example 2: Conditional Page Breaks in ODS

Example 3 demonstrates how this method is also used to execute conditional LINE statements in a COMPUTE block, which cannot be done with using a simple IF statement within the COMPUTE block itself:
Example 3: Conditional LINE statement in a COMPUTE BLOCK

Example 3 creates service headers and sub-headers. If services for a route are on their own dedicated page, then the service header is provided by the TITLE statement in Example 2, and does not need to be repeated. Otherwise, the header is created and displayed in lines 4 and 5. The second condition in lines 8-13 only prints a service sub-header followed by a blank line when a sub-header exists.

IT ALL BEGINS WITH THE ODS TEMPLATE

None of the default SAS ODS templates match our desired data display, so we have to create our own by modifying a standard template. The Calibri font was ultimately chosen as our timetable font for its readability at smaller font sizes, and the 8.5-point size will accommodate our data in the minimum number of pages, while somewhat maintaining legibility. The custom template code is in Example 4:
1. ODS PATH work.templat(UPDATE) sashelp.tmplmst(READ);
2. PROC TEMPLATE;
3. DEFINE STYLE timetbl; PARENT=styles.printer;
4. REPLACE BODY FROM DOCUMENT / LEFTMARGIN=.25in
   rightmargin=.25in TOPMARGIN=.25in bottommargin=.25in;
5. STYLE SYSTITLEANDFOOTERCONTAINER FROM SYSTITLEANDFOOTERCONTAINER /
   CELLPADDING = 0;
6. STYLE SYSTEMTITLE / FONT=(Calibri) FONTSIZE=11pt FONTWEIGHT=bold;
7. STYLE SYSTEMFOOTER / FONT=(Calibri) FONTSIZE=8.5pt;
8. STYLE TABLE FROM TABLE / CELLPADDING=0pt CELLPACING=1pt
9. BORDERCOLOR=light gray BORDERBOTTOMCOLOR=light gray
10. BORDERWIDTH=1pt FONTSIZE=8.5pt;
11. STYLE HEADER / FONT=(Calibri) FONTSIZE=10pt FONTWEIGHT=bold;
12. STYLE PAGENO /FONT=(Calibri) FONTSIZE=8.5pt FONTWEIGHT=medium;
13. STYLE DATA / FONT=(Calibri) FONTSIZE=8.5pt TEXTALIGN=right;
14. STYLE USERTEXT FROM USERTEXT / FONT=(Calibri) FONTSIZE=11pt
    FONTSIZE=10pt FONTSIZE=8.5pt FONTWEIGHT=medium;
15. END;
16. RUN;
17. ODS ESCAPECHAR='~'; /* Necessary for in-line formatting! */

Example 4: The TIMETBL ODS Template

We use the standard ODS template STYLES.PRINTER as our basis (line 3.) Our template:

- Defines page margins to maximize the printable area of the page while maintaining the
  ability to print the output on paper.
- Modifies the font size and selection for each of the ODS objects as we require.
- Changes the amount of padding around the title and footer cells.
- Changes the border and cell spacing parameters of the table.

CREATING THE SCHEDULE SECTION

Figure 5 shows a partial schedule section from a page of our timetable. There are twelve
trains in this direction on this particular route, automatically split into two groups of similar
lines by PROC REPORT. It is color-coded to show how the table is assembled from our
database:

<table>
<thead>
<tr>
<th>Days of Operation</th>
<th>Train Name</th>
<th>Miles</th>
<th>1056</th>
<th>358</th>
<th>1058</th>
<th>1060</th>
<th>362</th>
<th>360</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Flyer Service</td>
<td>Blackbird Service</td>
<td>Flyer Service</td>
<td>Blackbird Service</td>
<td>Blackbird Service</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Mo-Fr</td>
<td>Mo-Fr</td>
<td>Daily</td>
<td>Daily</td>
<td>Holidays</td>
<td>Daily</td>
</tr>
<tr>
<td>Perry</td>
<td>00p</td>
<td>5:00AM</td>
<td>5:05AM</td>
<td>6:00AM</td>
<td>7:00AM</td>
<td>9:00AM</td>
<td>11:00AM</td>
<td></td>
</tr>
<tr>
<td>Clarkson</td>
<td>42p</td>
<td>5:44AM</td>
<td>6:00AM</td>
<td>6:58AM</td>
<td>7:58AM</td>
<td>10:15AM</td>
<td>1:05PM</td>
<td></td>
</tr>
<tr>
<td>Pottstown</td>
<td>81p</td>
<td>5:58AM</td>
<td>6:20AM</td>
<td>6:58AM</td>
<td>7:58AM</td>
<td>10:15AM</td>
<td>1:05PM</td>
<td></td>
</tr>
<tr>
<td>Jamestown</td>
<td>134p</td>
<td>7:05AM</td>
<td>11:00AM</td>
<td>11:15AM</td>
<td>11:51AM</td>
<td>2:41PM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chestnut Hill</td>
<td>191p</td>
<td>7:15AM</td>
<td>7:56AM</td>
<td>8:15AM</td>
<td>9:15AM</td>
<td>11:51AM</td>
<td>2:41PM</td>
<td></td>
</tr>
<tr>
<td>Greenwood</td>
<td>216p</td>
<td>8:18AM</td>
<td>12:13PM</td>
<td>3:03PM</td>
<td>3:41PM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Allegheny</td>
<td>258p</td>
<td>8:56AM</td>
<td>12:51PM</td>
<td>3:41PM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Columbia</td>
<td>291p</td>
<td>8:32AM</td>
<td>9:45AM</td>
<td>9:32AM</td>
<td>10:32AM</td>
<td>1:40PM</td>
<td>4:30PM</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5: Data Sources for the Schedule Section
The light blue in Figure 5 indicates the data come from the schedule datasets, TTBASE_IN and TTBASE_OUT, and green, the SERVICES dataset. Any relevant reference marks (®, ⃝) to be added to the days of operation cells are assigned on the fly. Yellow indicates the data have been assembled from the schedule and timetable notes (TTNOTES_IN/OUT) datasets.

Vertical ordering is done by using a non-printing variable, and the ordering values for train number, name, and days of operation are defined so they will be the first three lines in every schedule, in that order. While the schedules themselves are maintained as SAS time values, the train name and “Days of Operation” values are character variables. Since we need to add reference marks to the schedule (for example, the "D" on the “South Columbia” line in Figure 5.), all of the displayed columns except for the mileage are character variables. The CATS() and PUT() functions create the data for each of the yellow cells.

As for the report itself, the train numbers are column labels; the actual variables used are temporary, only created for the timetable. The “~{newline}” metacharacter is used where necessary to break the train name in order to fit the column width defined in the PROC REPORT step. In-line formatting centers and bolds the train name, centers the “Days of Operation” cells, and bolds the PM times in accordance with US railroad tradition.

Each PROC REPORT that creates a schedule section must be created dynamically, because the stations and trains vary by route and the variable names are not known in advance. As noted earlier, a macro loop runs each route individually. First, we create a temporary dataset for the schedule cells by merging the schedule and timetable notes datasets from our database. We can then use DICTIONARY.COLUMNS to determine the names of the columns to print. In Example 5, &&RID&I is the macro loop index and resolves to the route number:

```
1. PROC SQL NOPRINT;
2. SELECT STRIP(PUT(COUNT(name),3.)), STRIP(PUT((COUNT(name))/2,3.))
   INTO :nsvcs, :dtrns
3. FROM dictionary.columns
4. WHERE libname EQ 'WORK' AND memname EQ "TT&&rid&i" AND
   SUBSTR(name,1,1) EQ 'T';
5. SELECT SCAN(name,1,'_') INTO :trn1-:trn&nsvcs
6. FROM dictionary.columns
7. WHERE libname EQ 'WORK' AND memname EQ "TT&&rid&i" AND SUBSTR(name,1,1)
   EQ 'T';
8. QUIT;
```

**Example 5: Getting Column Names Dynamically Using Dictionary Tables**

The first SQL query in Example 5 (lines 1-4) obtains the number of trains in the route (&nsvcs), and the number of trains in each direction (&dtrns), which are used to determine if the schedule will need to be split by direction for printing. We create the dynamic list of column names in the second query (lines 5-8.) Each column name corresponds to a train schedule in the given route, and the list of macro variables is generated each time the main macro loop executes. The character variable displayed by PROC REPORT is created in a DATA step combining the reference mark for each combination of train and station with the schedule time. The in-line formatting for the PM times is also done in the same DATA step. Finally, we add the character schedule dataset to the train name and days of operation records, and are now ready to execute the PROC REPORT for the schedule section.

Three different PROC REPORT steps are coded to create the schedule section; which one is used depends on how the information fits on the page. We’ve already calculated the horizontal and vertical space needed for the schedule information and stored that in macro
variables. The first test is whether both directions will fit horizontally. If so, then we can use the bidirectional layout (Appendix 1), shown in Example 6:

```
1. %IF &nsvcs LE 8 %THEN %DO;
2. OPTIONS ORIENTATION=PORTRAIT CENTER;
3. PROC REPORT DATA=txx&rid&i NOWD OUT=trace MISSING SPLIT='\'
4. BY lineID routeID RouteName;
5. COLUMNS sbrk lineID routeID routeName milepost
   ('Read Down' &outvnam) outbd_tt_miles dparout Station
dparin ('Read Up' &invnam);
6. DEFINE lineID / ORDER NOPRINT;
7. DEFINE routeID / ORDER NOPRINT;
8. DEFINE routeName / ORDER NOPRINT;
9. DEFINE milepost / NOPRINT ORDER ORDER=INTERNAL;
10. DEFINE outbd_tt_miles / ORDER ORDER=INTERNAL 'Miles' MISSING
    STYLE=[PADDINGRIGHT=2PT];
11. DEFINE dparout / ' ' STYLE=[TEXTALIGN=L PADDINGLEFT=1PT
    PADDINGRIGHT=2PT];
12. DEFINE Station / ' ' ID ORDER ORDER=INTERNAL
    STYLE=[CELLWIDTH=1.25IN TEXTALIGN=C FONTWEIGHT=BOLD];
13. %DO k=1 %TO &nsvcs;
14.  DEFINE &trn&k / STYLE(COLUMN)=[CELLWIDTH=.65IN];
15.  %END;
16.  DEFINE dparin / ' ' STYLE=[TEXTALIGN=L PADDINGLEFT=2PT
    PADDINGRIGHT=2PT];
17.  DEFINE sbrk / NOPRINT;
18.  COMPUTE lineID;
19.  IF sbrk EQ ' ' THEN
20.    CALL DEFINE(_ROW_,'STYLE',
        'STYLE=[BORDERBOTTOMSTYLE=HIDDEN]');
21.  ENDCOMP;
22.  RUN;
```

**Example 6: PROC REPORT Code for Bidirectional Schedule Layout**

A maximum of eight trains in both directions will fit across one page, and that is tested in line 1. Bidirectional schedules are centered on the page (CENTER option, line 2.) The %DO loop at line 13 creates a DEFINE statement for each schedule column using the list of macro variables created in the second query from Example 5. The macro variables &OUTVNAM and &INVNAM (line 5, Example 6) contain the list of outbound and inbound schedule variables, respectively. These are created using the SQL procedure to put the variable list into a single macro variable (Example 7.)

```
1. PROC SQL NOPRINT;
2. SELECT scan(_trainid,1,'_') INTO :outvnam SEPARATED BY ' ' 
3. FROM outbd
4. WHERE routeID EQ &&rid&I;
5. ORDER BY strttm;
6. QUIT;
```

**Example 7: Using PROC SQL to Create a List of Variables as a Single Macro Variable**

Finally, we don’t want a border to show when the value of STATION is repeated. Since it’s an ID column, PROC REPORT will only print the first occurrence of each value, and it won’t be actionable in a COMPUTE statement. We have to create a dummy variable SBRK in the dataset, which is blank if it is not the first occurrence of STATION. We use that in the
COMPUTE block in lines 18-21 of Example 6 to prevent the printing of the border with the BORDERBOTTOMSTYLE style attribute. If we changed the border color to match the background instead, there would be tiny gaps in the vertical cell lines. Yes, it is a minor issue, but the challenge was to reproduce the template as closely as possible. A magnified view demonstrates the difference in Figure 6:

![Figure 6: BORDERCOLOR=WHITE (top) vs. BORDERSTYLE=HIDDEN (bottom)](image)

The other schedule layouts require printing one direction at a time, so there is a similar PROC REPORT for each direction. Although the column and display order differs from Example 6, and they work on different schedule datasets created for each direction, we still use the macro variables &INVNAM and &OUTVNAM and the macro variable list we created in Example 5 to specify the columns and generate a DEFINE statement for each variable. We test to see if both directions will fit on a single page, and execute ODS STARTPAGE=NOW if they don’t.

**CREATING THE REFERENCE MARK SECTION**

Reference marks fall into two categories. "Standard" reference marks are maintained in the TTNOTES datasets and are linked to a specific train at a specific station. "Service exceptions" are days where a train runs in addition to its regularly scheduled days, or does not run even though it is scheduled to run. For example, a train scheduled to run Monday through Friday might not run on Thanksgiving, even though it is a Thursday. Similarly, a train scheduled to run only on holidays may also run on the Saturday before Thanksgiving, because more people than usual are traveling that day. Some exceptions don’t have the same date every year, as in the Thanksgiving examples above. Therefore, the actual dates of service exceptions have to be calculated for each calendar year. For convenience, the actual dates and train numbers of each service exception have been compiled for the next ten years and stored in the TRAINS2ADD/RMV datasets. The process to associate service exceptions with their actual calendar dates is beyond the scope of this paper.

We do not need to put all ten years of service exception dates in the timetable. Our timetables are effective for the six-month periods of January 1 through June 30, or July 1 through December 31, unless there is a schedule or service change during the six-month period. In that case, the start date is adjusted to the date of the change plus one week. %LET tt_frame=%SYSFUNC(TODAY());
1. PROC SQL NOPRINT;
2. SELECT MAX(MAX(lastchg)+7,INTNX('SEMIYEAR',&tt_frame,0,'B')) as effdt,
3.     INTNX('SEMIYEAR',&tt_frame,0,'E') as effend,
4.     PUT(CALCULATED effdt,worddate18.) as effdate,
5.     PUT(CALCULATED effend,worddate18.) as effendstr,
6.     CATX(' ~{unicode 2014} ',CALCULATED effdate,
7.     CALCULATED effendstr)
9. FROM (SELECT DATEPART(modate) as lastchg
10.     FROM dictionary.tables
11.     WHERE libname EQ 'RR' AND memname IN ('SERVICES', 'TTNOTES_IN',
12.        'TTNOTES_OUT','TTBASE_IN','TTBASE_OUT')
13.     )
14. ;
15. QUIT;

Example 8 shows how effective dates are calculated:

17. %LET tt_frame=%SYSFUNC(TODAY());
18. PROC SQL NOPRINT;
19. SELECT MAX(MAX(lastchg)+7,INTNX('SEMIYEAR',&tt_frame,0,'B')) as effdt,
20.     INTNX('SEMIYEAR',&tt_frame,0,'E') as effend,
21.     PUT(CALCULATED effdt,worddate18.) as effdate,
22.     PUT(CALCULATED effend,worddate18.) as effendstr,
23.     CATX(' ~{unicode 2014} ',CALCULATED effdate,
24.     CALCULATED effendstr)
26. FROM (SELECT DATEPART(modate) as lastchg
27.     FROM dictionary.tables
28.     WHERE libname EQ 'RR' AND memname IN ('SERVICES', 'TTNOTES_IN',
29.        'TTNOTES_OUT','TTBASE_IN','TTBASE_OUT')
30.     )
31. ;
32. QUIT;

Example 8: Calculating Effective Dates

The macro variable &TT_FRAME allows us to select any six-month period starting on January 1 or July 1. The default is TODAY(), which yields the current one, but we can pick a date in any future six-month period. The beginning and end of the six-month period are calculated with the INTNX() function (lines 4 and 5), using the adjustment parameters “B”, and “E”, respectively.

The subquery in lines 12 through 16 obtain the modification dates for the datasets that would have an effect on the timetable, and the latest modification date is used in line 4 of the main query (MAX(lastchg)). If that date (plus seven days) is later than the start date of the six-month period, then the start date is set to that value. The end date of the six-month period remains the same, regardless of the start date. All of these are stored in macro variables.

The start and end dates are SAS date values, and used with a WHERE=option and BETWEEN clause to filter TRAINS2ADD/RMV. Example 9 is the code that creates the reference marks for additional trains:
Example 9: Generating Reference Marks and Text for Service Exceptions
Line 3 filters the TRAINS2ADD dataset so we only have the service exceptions for the timetable’s effective dates. Each time this DATA step iterates, the exception date is formatted and appended onto a long character variable (lines 7 and 8.) Once all exception dates for a given train have been processed, we add introductory text and send it to a dataset (lines 10-16.) Since multiple trains can have the same exception dates, one reference mark is used for all trains with the same exception dates (for example, the “☺” displayed in Figure 7.) We sort the exception date text with the NODUPKEY option to provide unique values.

The next step is to generate a list of Unicode reference marks. We create an easily recognized distinction by having service removals use the Enclosed Numerics Unicode block, and additional services (Example 9, lines 22-31) use the Enclosed Alphanumerics Unicode block. We merge that Unicode list with the unique exception list text (lines 32-38) to associate a reference mark with exception text. The last step (lines 39-45) is to merge the reference marks with the train numbers by the exception text, providing the train-to-reference mark link used in the schedule section, as seen in Figure 7.

**Figure 7: Reference Marks Appended to Days of Operation**

Putting the reference marks together is one task; displaying them is another. We chose a three-column format to minimize the vertical space needed to print the reference marks and text.

Reference marks are always printed on the same page as the schedule because it simplifies coding, so ODS STARTPAGE=NO is used before the PROC REPORT that generates the reference mark section. If we chose to add another possible layout (schedule on one page, reference marks and services on a separate page), we’d have to test page fit and execute ODS STARTPAGE=NOW before displaying the reference mark section. If the reference mark section were on its own page, ODS STARTPAGE=NO for the services section would also be executed. Otherwise, the reference marks would always be on one page, and the services section on the next page.

What appears to be three columns is actually six; the reference mark is in a separate column from the text because it looks nicer when using a proportional font. We rely on PROC REPORT to wrap the reference mark text, giving us the result in Figure 8:

**Figure 8: The Reference Marks Section**

**CREATING THE SERVICES (ACCOMMODATIONS AND AMENITIES) SECTION**

The final piece of the timetable is the services information. This is the most complicated piece of the timetable, as a great deal of data preparation is necessary to create the services text for display. For brevity, we will focus on the creation of the service description, and not the formatting of the entire section.
The SERVICES dataset contains information specific to each train in the simulation, including the train name, quantity, and types of equipment. It is maintained as an Excel spreadsheet. A SAS program reads the spreadsheet, creates the SERVICES dataset, and produces an updated Excel workbook that can be updated. The workbook file has the creation date in its name, so it is easy to roll back erroneous changes. Table 2 is an excerpt from the SERVICES dataset (some variable names have been altered to fit):

<table>
<thead>
<tr>
<th>Train No</th>
<th>trainName</th>
<th>reserved</th>
<th>Bag</th>
<th>LH Coach</th>
<th>BC Coach</th>
<th>SLP106</th>
<th>SLP14</th>
<th>Diner</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The White Star</td>
<td>Y</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>The White Star</td>
<td>Y</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>The Capitol</td>
<td>Y</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>The Capitol</td>
<td>Y</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2: Excerpt from SERVICES Dataset

Formatting is only one aspect of the services section. Many questions need to be answered in the process: what train numbers are on each route? Do all the trains have the same services? How many unique services are on this route? Is there specific text associated with a specific train or trains? Let’s look at how the data from the SERVICES dataset are used to answer these questions and create service descriptions.

The first question is whether a general rule applies to this train requiring standard introductory text. As an example, if the train number is 10 or less, the text “All-reserved, extra fare train. Reservations required, fare supplement charged at booking.” should be displayed as the first item in the services text. If special text is present, the ~{newline} metacharacter separates it from the remaining service information. Next, we process the text for each type of car on the train. Each type of car has a standard short description that is also the label for the variables in the SERVICES dataset. Each type also has a full description with in-line formatting. The full description was created specifically for use in the services section of timetables, as shown in Table 3:

<table>
<thead>
<tr>
<th>Car Type Short Description</th>
<th>Car Type Description for Timetables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baggage Car</td>
<td>~S={fontweight=bold}Checked baggage service available.~S={}</td>
</tr>
<tr>
<td>Long-haul Coach</td>
<td>~S={fontweight=bold}Coaches. ~S={}Reclining seats, leg rests.</td>
</tr>
<tr>
<td>Dome Coach</td>
<td>~S={fontweight=bold}Dome Coaches. ~S={}Reclining seats, leg</td>
</tr>
<tr>
<td></td>
<td>rests on lower level, unreserved scenic viewing on upper level.</td>
</tr>
<tr>
<td>10-6 Sleeper</td>
<td>~S={fontweight=bold}Sleeping Cars. ~S={}Roomettes and double</td>
</tr>
<tr>
<td></td>
<td>bedrooms.</td>
</tr>
<tr>
<td>4-4-2 Sleeper</td>
<td>~S={fontweight=bold}Sleeping Cars. ~S={}Suites and double</td>
</tr>
<tr>
<td></td>
<td>bedrooms.</td>
</tr>
<tr>
<td>Full Service Diner</td>
<td>~S={fontweight=bold}Dining Car. ~S={}Full dining service.</td>
</tr>
</tbody>
</table>

Table 3: Sample Car Type Labels and Descriptions

The information in Table 3 is stored in one of the simulation’s core datasets. At initiation of the simulation, both the short and full descriptions are created as permanent formats in the simulation’s format library. We use the link between the labels for the variables in the
SERVICES dataset and these formats to obtain the description text. The code fragment in Example 10 demonstrates how this works:

```plaintext
1. ARRAY itsrvc{*} Baggage HSCoach LHCochar ARCoach BCCoach DMCoach SLP106 SLP14 SLP442 Dinette Diner TWDiner DMDiner HLSlounge Lounge DMlounge DinerLounge DOMlounge RoundObsLounge FlatObsLounge MobileOffice TGV1 TGV2 TGV3;
2. DO i=1 to DIM(itsrvc);
3. chk = VLABEL(itsrvc{i});
4. IF itsrvc{i} GT 0 THEN DO;
5. cartyp = PUT(chk,$svcdesc.);
6. IF itsrvc{i} EQ 1 THEN DO;
7. cartyp = TRANWRD(cartyp,'Coaches','Coach');
8. cartyp = TRANWRD(cartyp,'Cars','Car');
9. END;
10. svcdesc = CATS(svcdesc,cartyp,'~{newline}');
11. END;
12. END;
13. svcdesc = CATS(svcdesc,"~{newline}");
```

Example 10: Creating Services Text from SERVICES Dataset

First, we place the relevant variables from SERVICES into an array, allowing us to loop through each of the passenger car types. Line 3 gets the label of the variable, and we test to see if that particular type is in the train (line 4.) Line 5 assigns the formatted version of the variable label. Lines 6 through 9 change the full description from plural to single when there is only one of a certain type. Line 10 adds the text for this equipment and a line feed (`~{newline}`) to the service description text. That causes the next car type description to be displayed on the next line in the timetable. We insert another blank line to the end of the text (line 14) after all car types have been processed. Example 11 shows the actual service description text the program creates for trains 1 and 12:

<table>
<thead>
<tr>
<th>Train #</th>
<th>Service Description Text (svcdesc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td><del>S={fontweight=bold}All-reserved, extra fare train. Reservations required,</del>{newline}fare supplement charged at booking.<del>S={}</del>{newline}<del>S={fontweight=bold}Sleeping Cars. <del>S={}</del>{newline}Roomettes and double bedrooms.</del>{newline}<del>S={fontweight=bold}Sleeping Cars. <del>S={}</del>{newline}Roomettes.</del>{newline}~S={fontweight=bold}Sleeping Car. <del>S={}Suites and double bedrooms.</del>{newline}<del>S={fontweight=bold}Dining Car. <del>S={}</del>{newline}</del>{newline}~S={fontweight=bold}Dining Car. <del>S={}</del>{newline}<del>S={fontweight=bold}Lounge Car. <del>S={}</del>{newline}<del>S={fontweight=bold}Mobile Office Car. <del>S={}</del>{newline}</del>{newline} Internet and business suites.</del>{newline}</td>
</tr>
<tr>
<td>#12</td>
<td>~S={fontweight=bold}All-reserved train. Reservations required.<del>S={}</del>{newline}~S={fontweight=bold}Checked baggage service available.<del>S={}</del>{newline}<del>S={fontweight=bold}Coaches. <del>S={}</del>{newline}Reclining seats, leg rests.</del>{newline}<del>S={fontweight=bold}Sleeping Cars. <del>S={}</del>{newline}Roomettes and double bedrooms.</del>{newline}~S={fontweight=bold}Dining Car. <del>S={}</del>{newline}<del>S={fontweight=bold}Lounge Car. <del>S={}</del>{newline}<del>S={fontweight=bold}Mobile Office Car. <del>S={}</del>{newline}</del>{newline} Internet and business suites.</del>{newline}</td>
</tr>
</tbody>
</table>
**Example 11: Actual Service Description Text**

When we display the service description text using ODS RTF and PROC REPORT, we get Result 1 (the train numbers have been added to the table for easy reference to Example 11):

<table>
<thead>
<tr>
<th>Train #</th>
<th>Actual Display of Service Description Text (svcdesc)</th>
</tr>
</thead>
</table>

**Result 1: Actual Service Description Text as Generated by the RTF File**

As with reference marks, we collapse all service description text into unique values, as it is possible for multiple trains within a route to have the same accommodations and amenities. From there, intricate SAS code compiles all combinations of service description text, train name, and train number before determining how to format the services section of the timetable. How many lines the services section occupies on the page is calculated at the end of this process, which yields the data source for the PROC REPORT step that generates the services section. This is all done in DATA steps because we want the data formatted properly for display, and the PROC REPORT step can be simplified. In general, for such intricate formatting, the less work done inside PROC REPORT, the easier it is to debug and make adjustments.

**POST-PROCESSING AND OTHER DISCOVERIES**

It was noted earlier that some post-processing was required to create the output in its final form. It depends on the fonts available on your SAS platform. In this case, we started on Windows SAS, and the Calibri font was selected for its readability at small sizes. However, Courier is the default replacement font on SAS University Edition when the specified font is not available. I created a one-click Visual Basic macro solution to change the Courier font to Calibri throughout the entire timetable document, including titles, headers, and footers, because the global find/replace capacity in Microsoft Word only changes the font in the main body of this document. The Visual Basic code to do this is provided in Appendix 5.

Ultimately, in keeping with modern railroad practice, we want to create our timetable as a PDF file, and it is a two-step process: generate the timetables as an RTF file, change the font in Microsoft Word, and then create the PDF from the modified RTF document.
Interestingly, when we used Windows SAS, rendering issues occurred when creating the timetable directly via the PDF destination, so this has always been a two-step process of converting an RTF file into a PDF.

CONCLUSION

Base SAS and the basic capabilities of ODS and PROC REPORT successfully met the many challenges in reproducing this historical template. This particular exercise demonstrated just how powerful and comprehensive the reporting facilities in SAS are. In-line formatting can give you control over the display of your data at the individual character level if necessary. Metacharacters give you spacing control within a report cell. You even have access to Unicode characters if needed. Using macros for data-driven conditional execution of code, you can direct your data into multiple PROC REPORT statements, creating a myriad of layout possibilities. You can even suppress the display of border lines within a table.

The availability of SAS University Edition makes it possible for anyone to work with the power of SAS outside the boundaries of work or school. The idea of a railroad simulation began with an offhand comment. I was able to turn it into a way to grow and maintain my SAS, database design, and problem-solving skills. This highly specialized report is merely one aspect of the entire simulation, which has multiple opportunities for skill enhancement and development.

Finally, are there other ways to have done this in SAS? As with many things in SAS, I’m sure there are alternate approaches to generate similar output. ODS DOCUMENT, ODS LAYOUT, or using the DATA step with ODS probably could produce acceptable results, and possibly with less data manipulation. I may explore those alternatives in the future as a way to develop those skills.

ACKNOWLEDGMENTS

Thanks to Sharon Hamrick from SAS Technical Support for solving the unwanted table border problem, and to SAS Technical Support in general for their assistance over the years.

RECOMMENDED READING

- Carpenter’s Complete Guide to the SAS® REPORT Procedure

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:
Derek Morgan
mrdatesandtimes@gmail.com

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.

Other brand and product names are trademarks of their respective companies.
Appendix 1: Both Directions in One Table, Reference Marks/Services on Same Page

Columbia-Mt. Kennedy

<table>
<thead>
<tr>
<th>Read Down</th>
<th>Read Up</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>111</strong></td>
<td><strong>112</strong></td>
</tr>
<tr>
<td>The Mountaineer</td>
<td>The Pioneer</td>
</tr>
<tr>
<td>The Cherry Blossom</td>
<td>The Palisades</td>
</tr>
<tr>
<td><strong>115</strong></td>
<td><strong>114</strong></td>
</tr>
<tr>
<td><strong>117</strong></td>
<td><strong>116</strong></td>
</tr>
<tr>
<td>The Mountaineer</td>
<td>The Pioneer</td>
</tr>
<tr>
<td>The Cherry Blossom</td>
<td>The Palisades</td>
</tr>
<tr>
<td><strong>Miles</strong></td>
<td><strong>Days of Operation</strong></td>
</tr>
<tr>
<td>Daily</td>
<td>Daily</td>
</tr>
<tr>
<td>5:20A</td>
<td>7:02A</td>
</tr>
</tbody>
</table>

Reference Symbols:
- **D** Stops only to discharge passengers.
- **Y** Sleeping cars may be occupied 1 hour before departure.
- **R** Sleeping cars may be occupied at 10pm.
- **X** Sleeping cars may be occupied until 6am.
- **W** WILL NOT operate 1/1, 1/19, 5/4, 5/24.
- **M** WILL operate 1/1, 1/20, 4/12, 5/4, 5/23, 5/25.
- **B** WILL operate 1/1, 1/20, 4/12, 5/4, 5/23, 5/25.
- **G** WILL operate 1/1, 1/20, 4/12, 5/4, 5/23, 5/25.
- **V** WILL operate 1/1, 1/20, 4/12, 5/4, 5/23, 5/25.

Columbia-Mt. Kennedy Services

- **The Mountaineer - Trains 111 and 112**
  - Checked baggage service available.
  - Coaches: Reclining seats, leg rests.
  - Dinette: Light meals, snacks and beverages.
  - Lounge Car: Sandwiches, snacks, and beverages.

- **The Pioneer - Trains 113 and 114**
  - Checked baggage service available.
  - Coaches: Reclining seats, leg rests.
  - Dinette: Light meals, snacks and beverages.
  - Lounge Car: Sandwiches, snacks, and beverages.

- **The Cherry Blossom - Trains 115 and 116**
  - Checked baggage service available.
  - Coaches: Reclining seats, leg rests.
  - Dinette: Light meals, snacks and beverages.
  - Lounge Car: Sandwiches, snacks, and beverages.

- **The Palisades - Trains 117 and 118**
  - Checked baggage service available.
  - Coaches: Reclining seats, leg rests.
  - Dining Car: Full dining service.
  - Lounge Car: Sandwiches, snacks, and beverages.

Effective Dates: January 1, 2020 — June 30, 2020
## Appendix 2: Both Directions in Two Tables, One Page

### Perry-Phillips Bay

#### Westbound

<table>
<thead>
<tr>
<th>Train Name</th>
<th>Days of Operation</th>
<th>401</th>
<th>411</th>
<th>403</th>
<th>413</th>
<th>405</th>
<th>407</th>
<th>409</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mo-Tu (6)</td>
<td>Daily</td>
<td>Daily</td>
<td>Daily</td>
<td>SaSuHo</td>
<td>Mo-Tu (6)</td>
<td>Daily</td>
<td>Daily</td>
</tr>
<tr>
<td>Perry</td>
<td>Dp 0 5:15A</td>
<td>6:00A</td>
<td>8:00A</td>
<td>10:00A</td>
<td>5:20P</td>
<td>7:00P</td>
<td>8:30P</td>
<td></td>
</tr>
<tr>
<td>Oakdale</td>
<td>Dp 30 5:1A</td>
<td>6:36A</td>
<td>8:36A</td>
<td>10:36A</td>
<td>5:56P</td>
<td>7:36P</td>
<td>9:06P</td>
<td></td>
</tr>
<tr>
<td>Miller's Hook</td>
<td>Dp 110 7:23A</td>
<td>8:08A</td>
<td>10:08A</td>
<td>12:08P</td>
<td>7:28P</td>
<td>9:08P</td>
<td>10:38P</td>
<td></td>
</tr>
</tbody>
</table>

#### Eastbound

<table>
<thead>
<tr>
<th>Train Name</th>
<th>Days of Operation</th>
<th>406</th>
<th>408</th>
<th>410</th>
<th>414</th>
<th>402</th>
<th>412</th>
<th>404</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mo-Tu (6)</td>
<td>Daily</td>
<td>Daily</td>
<td>SaSuHo</td>
<td>Daily</td>
<td>Daily</td>
<td>Daily</td>
<td></td>
</tr>
<tr>
<td>Phillips Bay</td>
<td>0 Dp 5:15A</td>
<td>6:05A</td>
<td>9:45A</td>
<td>5:30P</td>
<td>5:30P</td>
<td>7:30P</td>
<td>8:20P</td>
<td></td>
</tr>
</tbody>
</table>

### Reference Symbols

- **D** Stops only to discharge passengers.
- **Y** Sleeping cars may be occupied 1 hour before departure.
- **W** Will not operate 1/1, 1/19, 5/4, 5/24.
- **J** Will not operate 1/1, 1/20, 4/11, 5/4, 5/25.
- **Q** Will not operate 1/19, 1/20, 5/4, 5/24.
- **A** Will also operate 1/1, 1/2, 2/20, 5/4, 5/25.
- **G** Will also operate 1/20.
- **F** Will operate 1/1, 1/4, 1/14, 4/14, 5/6, 5/25, 5/29.
- **R** Stops only to receive passengers.
- **C** Will operate 1/1, 1/20, 4/11, 5/4, 5/23, 5/25.

### Perry-Phillips Bay Services

**Kingfisher Service - Trains 401 through 414**

**Coaches.** Reclining seats, leg rests.

**Lounge Car.** Sandwiches, snacks, and beverages.

---

Page 27

Effective Dates: January 1, 2020 — June 30, 2020
## Appendix 3: One Direction on Page, Services on Same Page

### Columbia-Perry Westbound

<table>
<thead>
<tr>
<th>Train Name</th>
<th>Days of Operation</th>
<th>Read Down</th>
<th>1051</th>
<th>1053</th>
<th>361</th>
<th>351</th>
<th>353</th>
<th>355</th>
<th>357</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Flyer Service</td>
<td>Flyer Service</td>
<td>Blackbird Service</td>
<td>Blackbird Service</td>
<td>Blackbird Service</td>
<td>Blackbird Service</td>
<td>Blackbird Service</td>
<td>Blackbird Service</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mo-Fri</td>
<td>Holidays</td>
<td>Daily</td>
<td>Daily</td>
<td>Daily</td>
<td>Daily</td>
<td>Mo-Fri</td>
<td></td>
</tr>
<tr>
<td>Columbia</td>
<td>Dp 0</td>
<td>5:05A</td>
<td>9:06A</td>
<td>10:10A</td>
<td>11:50A</td>
<td>3:30P</td>
<td>4:20P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Allegheny</td>
<td>Dp 33</td>
<td>9:44A</td>
<td>10:54A</td>
<td>12:34P</td>
<td>4:14P</td>
<td>5:04P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pottstown</td>
<td>Dp 210</td>
<td>7:36A</td>
<td>8:36A</td>
<td>12:24P</td>
<td>1:34P</td>
<td>6:54P</td>
<td>7:44P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clarkson</td>
<td>Dp 249</td>
<td>1:00P</td>
<td>2:10P</td>
<td>3:05P</td>
<td>7:30P</td>
<td>8:20P</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Columbia-Perry Eastbound

<table>
<thead>
<tr>
<th>Train Name</th>
<th>Days of Operation</th>
<th>Read Down</th>
<th>363</th>
<th>1055</th>
<th>1057</th>
<th>359</th>
<th>1059</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Blackbird Service</td>
<td>Flyer Service</td>
<td>Flyer Service</td>
<td>Blackbird Service</td>
<td>Flyer Service</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mo-Fri</td>
<td>Holidays</td>
<td>Daily</td>
<td>Daily</td>
<td>Daily</td>
<td></td>
</tr>
<tr>
<td>Columbia</td>
<td>Dp 0</td>
<td>5:30P</td>
<td>6:45P</td>
<td>6:35P</td>
<td>7:00P</td>
<td>7:40P</td>
<td></td>
</tr>
<tr>
<td>South Columbia</td>
<td>Dp 9</td>
<td>R5:52P</td>
<td>R6:52P</td>
<td>R7:22P</td>
<td>R7:57P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Allegheny</td>
<td>Dp 33</td>
<td>6:14P</td>
<td></td>
<td>7:44P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Greenwood</td>
<td>Dp 75</td>
<td>6:53P</td>
<td></td>
<td>8:23P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chestnut Hill</td>
<td>Dp 100</td>
<td>7:16P</td>
<td>8:16P</td>
<td>8:46P</td>
<td>8:56P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jamestown</td>
<td>Dp 157</td>
<td>8:05P</td>
<td></td>
<td>9:35P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clarkson</td>
<td>Dp 249</td>
<td>9:30P</td>
<td></td>
<td>11:00P</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Reference Symbols
- **D** Stops only to discharge passengers.
- **Y** Sleeping cars may be occupied 1 hour before departure.
- **X** Sleeping cars may be occupied until 8am.
- **R** Stops only to receive passengers.
- **Z** Sleeping cars may be occupied at 10pm.
- **W** Will NOT operate 1/1, 1/19, 5/4, 5/24.
- **W** Will NOT operate 1/1, 1/20, 4/11, 5/4, 5/25.
- **W** Will NOT operate 1/19, 1/20, 5/4, 5/24.
- **W** Will also operate 1/1, 1/20, 5/4, 5/24.
- **W** Will also operate 1/20, 5/4, 5/24.

### Columbia-Perry Services

**Blackbird Service** - Trains 351 through 364
- **Coaches.** Reclining seats, leg rests.
- **Lounge Car.** Sandwiches, snacks, and beverages.

**Flyer Service** - Trains 1051 through 1060
- **Coaches.** Reclining seats, leg rests.
- **Business Class.** Reclining seats, leg rests, extra legroom.
- **Lounge Car.** Sandwiches, snacks, and beverages.

Effective Dates: January 1, 2020 — June 30, 2020
## Phillips Bay-Englewood Beach

### Northbound

<table>
<thead>
<tr>
<th>Train Name</th>
<th>Miles</th>
<th>The Islander</th>
<th>The Beachcomber</th>
<th>The Meteor</th>
<th>The Boardwalk Limited</th>
<th>The Clipper</th>
<th>The Twilight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Days of Operation</td>
<td></td>
<td>Daily</td>
<td>Daily</td>
<td>Daily</td>
<td>Daily</td>
<td>Daily</td>
<td>Sa-Th</td>
</tr>
<tr>
<td>Englewood Beach</td>
<td>0-Dp</td>
<td>6:30P</td>
<td>12:15P</td>
<td>3:15P</td>
<td>R8:15P</td>
<td>9:00P</td>
<td>111:30P</td>
</tr>
<tr>
<td>Chester</td>
<td>51-Dp</td>
<td>7:32A</td>
<td>1:17P</td>
<td>4:17P</td>
<td>10:02P</td>
<td>12:32A</td>
<td></td>
</tr>
<tr>
<td>Doylestown</td>
<td>136-Dp</td>
<td>8:50A</td>
<td>2:35P</td>
<td>5:35P</td>
<td>1:55A</td>
<td>3:11A</td>
<td></td>
</tr>
<tr>
<td>Sebring</td>
<td>195-Dp</td>
<td>9:45A</td>
<td>3:30P</td>
<td>6:30P</td>
<td>R8:57P</td>
<td>12:07A</td>
<td>2:45A</td>
</tr>
<tr>
<td>Orono</td>
<td>261-Ar</td>
<td>10:51A</td>
<td>4:36P</td>
<td>7:36P</td>
<td>9:58P</td>
<td>1:11A</td>
<td>3:51A</td>
</tr>
<tr>
<td>Redmond</td>
<td>315-Dp</td>
<td>11:46A</td>
<td>5:31P</td>
<td>8:31P</td>
<td>4:46A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logan</td>
<td>449-Dp</td>
<td>1:42P</td>
<td>7:27P</td>
<td>10:27P</td>
<td>6:42A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Livingston</td>
<td>519-Dp</td>
<td>2:53P</td>
<td>8:36P</td>
<td>11:38P</td>
<td>5:02A</td>
<td>7:53A</td>
<td></td>
</tr>
<tr>
<td>Lincoln</td>
<td>588-Dp</td>
<td>4:00P</td>
<td>9:45P</td>
<td>12:45A</td>
<td>R1:59A</td>
<td>6:06A</td>
<td>9:00A</td>
</tr>
<tr>
<td>Franklin</td>
<td>677-Dp</td>
<td>5:22P</td>
<td>11:07P</td>
<td>2:07A</td>
<td>7:28A</td>
<td>10:22A</td>
<td></td>
</tr>
<tr>
<td>Nicetown</td>
<td>755-Dp</td>
<td>5:46P</td>
<td>12:35A</td>
<td>3:35A</td>
<td>11:50A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hunting Park</td>
<td>780-Dp</td>
<td>7:15P</td>
<td>1:00A</td>
<td>4:00A</td>
<td>9:13A</td>
<td>12:15P</td>
<td></td>
</tr>
<tr>
<td>Fairmount</td>
<td>815-Dp</td>
<td>7:49P</td>
<td>1:34A</td>
<td>4:34A</td>
<td>12:49P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bryansville</td>
<td>975-Dp</td>
<td>10:08P</td>
<td>5:55A</td>
<td>6:53A</td>
<td>D7:01A</td>
<td>12:01P</td>
<td>3:08P</td>
</tr>
<tr>
<td>Wilson</td>
<td>1053-Dp</td>
<td>11:22P</td>
<td>5:07A</td>
<td>8:07A</td>
<td>4:22P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Springfield</td>
<td>1150-Dp</td>
<td>1:02A</td>
<td>6:47A</td>
<td>9:47A</td>
<td>2:17P</td>
<td>6:02P</td>
<td></td>
</tr>
<tr>
<td>Greenville</td>
<td>1179-Dp</td>
<td>1:37A</td>
<td>7:23A</td>
<td>10:23A</td>
<td>6:37P</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Reference Symbols

- **D** Stops only to discharge passengers.
- **Y** Sleeping cars may be occupied 1 hour before departure.
- **S** Will operate 1/1, 1/19, 5/4, 5/24.
- **W** Will operate 1/1, 1/20, 4/11, 5/4, 5/25.
- **M** Will operate 1/1, 1/20, 4/12, 5/4, 5/24, 5/25.
- **O** Will operate 1/1, 1/20, 4/12, 5/4, 5/23, 5/25.
- **F** Will operate 1/1, 1/20, 5/4, 5/23, 5/25.
- **E** Will operate 1/1, 1/20, 5/4, 5/23, 5/25.
- **R** Stops only to receive passengers.
- **Z** Sleeping cars may be occupied at 10pm.
- **X** Sleeping cars may be occupied until 6am.
- **1** Will operate 1/1, 1/20, 5/4, 5/23, 5/25.
- **2** Will operate 1/1, 1/20, 5/4, 5/23, 5/25.
- **3** Will operate 1/1, 1/20, 5/4, 5/23, 5/25.
- **4** Will operate 1/1, 1/20, 5/4, 5/23, 5/25.
- **5** Will operate 1/1, 1/20, 5/4, 5/23, 5/25.
- **6** Will operate 1/1, 1/20, 5/4, 5/23, 5/25.

### Effective Dates

- January 1, 2020 — June 30, 2020

---

Page 7
Phillips Bay-Englewood Beach Services

The Boardwalk Limited - Trains 3 and 4
All-reserved, extra fare train. Reservations required.
Checked baggage service available.
Coaches. Reclining seats, leg rests.
Sleeping Cars. Roomettes and double bedrooms.
Dining Car. Full dining service.
Lounge Car. Sandwiches, snacks, and beverages.
Mobile Office Car. Internet and business suites.

The Islander - Trains 33 and 34
Checked baggage service available.
Coaches. Reclining seats, leg rests.
Sleeping Car. Roomettes and double bedrooms.
Dining Car. Full dining service.
Lounge Car. Sandwiches, snacks, and beverages.

The Meteor - Trains 37 and 38
Checked baggage service available.
Coaches. Reclining seats, leg rests.
Sleeping Cars. Roomettes and double bedrooms.
Dining Car. Full dining service.
Lounge Car. Sandwiches, snacks, and beverages.

The Clipper - Trains 31 and 32
Checked baggage service available.
Coaches. Reclining seats, leg rests.
Sleeping Cars. Roomettes and double bedrooms.
Dining Car. Full dining service.
Lounge Car. Sandwiches, snacks, and beverages.

The Beachcomber - Trains 35 and 36
Checked baggage service available.
Coaches. Reclining seats, leg rests.
Sleeping Car. Roomettes and double bedrooms.
Sleeping Car. Roomettes.
Dining Car. Full dining service.
Lounge Car. Sandwiches, snacks, and beverages.

The Twilight - Trains 39 and 40
Checked baggage service available.
Coaches. Reclining seats, leg rests.
Sleeping Cars. Roomettes and double bedrooms.
Dining Car. Full dining service.
Lounge Car. Sandwiches, snacks, and beverages.

Effective Dates: January 1, 2020 — June 30, 2020
Sub CouriertoCalibri()
    Application.ScreenUpdating = False
    Dim objSingleWord As Range
    Dim objDoc As Document

    Set objDoc = ActiveDocument

    With objDoc
        For Each objSingleWord In .Words
            If objSingleWord.Font.Name = "Courier" Then
                objSingleWord.Font.Name = "Calibri"
            End If
        Next
    End With

    Call fixHdFt
End Sub

Sub fixHdFt()
    Application.ScreenUpdating = False
    Dim Rng2 As Range, Sctn As Section, HdFt As HeaderFooter
    With ActiveDocument
        For Each Rng2 In .StoryRanges
            Call FndRepRng(Rng2)
        Next
        For Each Sctn In .Sections
            For Each HdFt In Sctn.Headers
                With HdFt
                    If .LinkToPrevious = False Then
                        Call FndRepRng(HdFt.Range)
                    End If
                End With
            Next
            For Each HdFt In Sctn.Footers
                With HdFt
                    If .LinkToPrevious = False Then
                        Call FndRepRng(HdFt.Range)
                    End If
                End With
            Next
        Next
    End With
End Sub

Sub FndRepRng(Rng2 As Range)
    With Rng2.Find
        .ClearFormatting
        .Text = ""
        .Font.Name = "Courier"
    With .Replacement
        .ClearFormatting
        .Text = ""
        .Font.Name = "Calibri"
    End With
    .Forward = True
    .Wrap = wdFindContinue
    .Format = True
    .Execute Replace:=wdReplaceAll
End With
End Sub