

MapsAlive® User Guide

Drawing Routes

Last updated March 22, 2016

Table of Contents

1	Introduction	3
2	Terminology	4
3	Specifying and Drawing a Route	5
3.1	Route waypoints	5
3.2	Route hotspot	5
3.3	Drawing a route.....	6
4	Route Drawing Example 1.....	7
5	Importing Routes from a Spreadsheet	11
5.1	Segments.....	11
5.2	Routes spreadsheet.....	11
5.3	Non-contiguous routes.....	12
5.4	How to import a routes file.....	12
5.5	Testing imported routes.....	12
6	Route Drawing Example 2 Using Imported Routes.....	14
6.1	The Spreadsheet	15
6.2	The Map	17
6.3	Reversing segment direction.....	18
6.4	HTML and JavaScript	19
	Appendix A – the drawRoute JavaScript API function.....	21

Routes

1 Introduction

What is a Route?

A route is a line drawn on a map through a set of two or more hotspot markers in a connect-the-dots fashion. You can set the line's width, color, opacity, and effects. You can also create gaps in the routes to show that the route travels through something opaque like a tunnel.

To use the route feature, you must have a Plus or Pro Plan.

Who is this guide for?

This guide is for anyone who wants to include routes as part of an interactive map; however, route drawing is an advanced feature and using it requires some basic experience with JavaScript.

How to get more information or assistance

If you need more details, a better explanation, or just a little bit of hand-holding, we are here to help. Please email questions to support@mapsalive.com.

You can find other MapsAlive User Guides at <http://www.mapsalive.com/LearningCenter>.

2 Terminology

MapsAlive

MapsAlive is a web application that runs in your PC or Mac browser with a high-speed internet connection. There is no software to install and you are always using the latest version.

Tour Builder

The MapsAlive Tour Builder is where you create, edit, and publish interactive maps. When you login to your MapsAlive account you can use the Tour Builder to upload map images and photos, enter text, create hotspots, choose layouts, and set the appearance and behavior of your interactive maps.

Tour

A tour is one or more interactive maps and data sheets that are usually related to each other. *Many tours might only have one map. Other tours, like a house tour, might have several floor plans together in one tour to let a prospective buyer explore the house.* You publish your tours when you are ready for other people to use them.

Map

A map is a component of your tour that serves as a container for a single map image, the map's hotspots and hotspot content. You upload a map image and add hotspots to the map using the Tour Builder.

Data sheet

A data sheet contains photos, text and other information related to your tour, but it has no map. A data sheet might be used to display introductory text, contact information, specifications, or similar non-interactive content.

Map Image

A map image can be any image such as a diagram, floor plan, geographic map, or even a photograph. You can think of it as a background layer beneath a layer of hotspots. You upload one map image for each map in your tour.

Hotspot

A hotspot is a location on an interactive map that shows content associated with that location when the mouse moves over it, off of it, or clicks it. The size, shape, and appearance of a hotspot are indicated by a marker.

Hotspot Content

Hotspot content is the information associated with a specific hotspot such as photos, text, and videos. When the mouse moves onto or clicks a hotspot, the content displays. A hotspot can also have an action associated with it.

Marker

A marker can be an image (like an arrow), a visible shape (like a rectangle or polygon), or an invisible shape, that marks the presence of a hotspot on a map. Each hotspot on a map has a marker associated with it. The marker can be unique to one hotspot or the marker can be used by several hotspots.

Tiled versus Popup Content

Tiled content means that your hotspot content displays alongside the map – both the content and the map display within the rectangular canvas area. Popup content is hotspot content that displays in a rectangular area that pops up on top of your map when you mouse over a hotspot on the map.

Canvas

Every tour has a canvas area that displays one map or data sheet. For maps with tiled content, the canvas area contains the map image and the hotspot content. For maps with popup content, the canvas area contains only the map image. For data sheets, the canvas area contains the photo and text for the data sheet.

3 Specifying and Drawing a Route

A route consists of a set of waypoints that can be connected to draw a path from a start point to an end point through a set of intermediate waypoints. The route itself is a special hotspot that is used as a container in which the route's path is drawn. You draw a route by calling a JavaScript function. Each of these elements of a route will be discussed in the following sections.

3.1 Route waypoints

A waypoint in MapsAlive is just a term we use when talking about routes. When we say waypoint we simply mean a marker that a route will be drawn through. Any hotspot's marker can serve as a waypoint except for a special hotspot that is used to draw the route itself.

You indicate the location of a waypoint on the **Map** screen in the Tour Builder by placing a hotspot marker on your map like a dot in a connect-the-dots drawing. MapsAlive uses the center of each waypoint marker as the dot through which a route is drawn.

If you want to see the waypoints that a route is drawn through, use markers that are visible. If you don't want to see the waypoints, use markers that are invisible. For example, if you have a map that uses stars as markers to show the locations of cities, you could draw a route between cities using the visible star markers as waypoints. If you have a floor plan and you want to draw routes from the building entrance to various destinations, you could use invisible markers as waypoints so that you don't clutter up the hallways.

3.2 Route hotspot

In order to draw a route through a set of waypoints you need to add a route hotspot to your map. A route hotspot serves as a container within which the route will be drawn. When you draw a route, you specify the Id of the route hotspot and that hotspot takes on the appearance of the route. You can erase a route by drawing another route using the same route hotspot, or you can use multiple route hotspots so that you can show more than one route at a time.

You create a route hotspot from any existing hotspot by choosing **Hotspot > Advanced Hotspot Options** in the menu and then checking the **Is Route** box.

In order to draw a route, at least one hotspot on your map must have its **Is Route** attribute set.

A route hotspot has all of the attributes of a regular hotspot except that it has no marker and you can't use it as a waypoint in a route. It can, however, have content (like a photo or text) and respond to mouse actions. For example, you can set things up so that when you mouse over a route a photo appears. Or specify that when you click on a route, a JavaScript function gets called. You can also make a route hotspot static so that nothing happens when you mouse over it or click it.

3.3 Drawing a route

A route does not appear on your map until you tell MapsAlive to draw it using the `drawRoute` function of the JavaScript API. You can call this function any time you like, for example, when your map first loads, when a marker is moused over or clicked, or in response to a user action on your web page.

The `drawRoute` function is intended for use by JavaScript programmers, but non-programmers can use the examples in this document to successfully draw routes. Here is an example of what the function looks like when called to draw a route through three hotspots that have Ids of H1, H2, and H3. In the example, the hotspot Id for the route hotspot (see section 3.2 above) is R1.

```
mapsalive.drawRoute("R1", "H1,H2,H3");
```

The `drawRoute` function can take additional optional parameters that allow you to specify the appearance of the route. You can set the line width, color, opacity, and effects such as shadow or glow. To learn more about the `drawRoute` function see Appendix A of this document or the *MapsAlive User Guide for the JavaScript API*.

You can call the `drawRoute` function from your own JavaScript or when a marker mouse event occurs. The marker mouse events are `mouseover`, `mouseout`, and `click`. You specify JavaScript for a hotspot's mouse events on the **Hotspot Actions** screen.

4 Route Drawing Example 1

This section presents a sample tour that draws routes. You can run the tour and download its archive file using the URLs shown below. Or you can follow the steps and create the tour yourself.

Tour: <http://samples.mapsalive.com/20201>

Archive: <http://samples.mapsalive.com/Archive/MapsAliveArchiveForTour20201.zip>

This example draws routes between cities on a world map. When the map first loads, a route from San Francisco to Perth appears. As you move your mouse over a city, the route changes to originate from that city. In the right hand map in the figure below, the route has changed to originate from Mumbai.



Figure 1 - Routes from San Francisco to Perth and from Mumbai to Madrid

Overview of the Steps Involved

Below are the steps necessary to create the tour. Each step is explained in the sections that follow.

1. Create a tour using a Ready Map of the World.
2. Add hotspots for the cities and routes.
3. Add Custom HTML and mouseover handlers to call JavaScript that draws the routes.

Create the tour using a Ready Map of the world

Follow the instructions below to create the tour and map.

1. Choose **New > Tour** in the menu.
2. On the Create New Tour screen click **Continue**.
3. On the New Tour Setup screen:
 - a. Type "Routes Example 1" as the **Tour Name**.
 - b. Choose **Medium** for the tour size.
 - c. Choose **Embedded in another web page**.
 - d. Click **Continue to Map Setup**.
4. On the New Map Setup screen:
 - a. Choose **Popup** for how the content will appear.

- b. Click **Continue**.
5. On the Choose Map Image screen:
 - a. Choose **I want to use a Ready Map**.
 - b. In the Ready Maps Navigator click **World > World Maps > Outline map**.
 - c. Click **OK** when the confirmation dialog appears
 - d. Don't import shapes for this map when the Import Shapes screen appears

Add hotspots for the cities

Follow the instructions below to create five city hotspots.

1. Choose **New > Hotspot** in the menu.
2. On the **Edit Hotspot Content** screen:
 - a. Type "San Francisco" for the Title.
 - b. Type "SFO" for the Hotspot Id.
 - c. Choose the marker named **Star** from the **Marker** dropdown list.
3. Repeat steps 1 and 2 above to add hotspots with these titles and Ids:
 - a. Rio De Janeiro – RIO
 - b. Madrid – MAD
 - c. Mumbai – BOM
 - d. Perth – PER
4. Choose **Map > Go to Map**.
5. On the **Map** screen, place each city's star onto the map:
 - a. Click the city thumbnail below the map or select its name from the list above the map.
 - b. Drag the city's marker to its approximate location on the map.

Add a hotspot for the route

Follow the instructions below to create a route hotspot as was discussed in section 3.2 above.

1. Choose **New > Hotspot** in the menu.
 - a. Type "Route 1" for the Title.
 - b. Type "R1" for the Hotspot Id.
2. Choose **Hotspot > Advanced Hotspot Options**
3. In the **Marker Attributes** section, check the box labeled **Is Route**.

Draw the route

Follow the instructions below to add JavaScript that will draw the route from San Francisco to Perth.

1. Choose **Tour > Custom HTML** in the menu.
2. Enter the code below into the **JavaScript** text box of the **Custom HTML** screen.

```
function maOnMapLoaded()
{
  mapsalive.drawRoute("R1", "SFO, RIO, MAD, BOM, PER");
}
```

3. Click **Tour Preview** to see what happens. The tour should look similar to the left hand map in Figure 1 above.

When the map loads, the `mapOnMapLoaded` function will be called automatically. When it is called, the call to `drawRoute` will be executed. It will draw a route through the hotspot Ids in the order listed.

Things to try:

- Change the route by altering the order of the hotspot Ids.
- Have the route return to San Francisco by adding SFO at the end of the list.
- Create a gap in the route by replacing a comma with a semicolon as will be explained later in section 5.3.
- Add optional parameters to the `drawRoute` call to change the appearance of the route's line. See Appendix A at the end of this document to learn how.

Make the map interactive

Follow the instructions below to make the route change when you mouse over a city.

1. Click **Madrid** in the **Tour Navigator**.
2. Choose **Hotspot > Hotspot Actions** in the menu.
3. Choose **JavaScript** from the **Mouseover Action** dropdown.
4. Enter the code below in the **JavaScript** text box.

```
mapsalive.drawRoute("R1", "MAD, BOM, PER, RIO, SFO");
```
5. Click **Rio De Janeiro** in the **Tour Navigator**
6. Repeat step 3 and use the code below as the **Mouseover Action**.

```
mapsalive.drawRoute("R1", "RIO, SFO, MAD, BOM, PER");
```
7. Click **Mumbai** in the **Tour Navigator**
8. Repeat step 3 and use the code below as the **Mouseover Action**.

```
mapsalive.drawRoute("R1", "BOM, PER, RIO, SFO, MAD");
```
9. Click **Perth** in the **Tour Navigator**
10. Repeat step 3 and use the code below as the **Mouseover Action**.

```
mapsalive.drawRoute("R1", "PER, RIO, SFO, MAD, BOM");
```
11. Click **Tour Preview** and mouse over the cities. When you mouse over Mumbai, the tour should look similar to the right hand map in Figure 1 above. Nothing will happen when you mouse over San Francisco, because you have not added a mouseover handler for it yet.

Draw a second route

This last part of the example demonstrates some advanced features. Follow the instructions below to draw two routes when you mouse over San Francisco.

1. Choose **New > Hotspot** in the menu.
 - a. Type "Route 2" for the Title.
 - b. Type "R2" for the Hotspot Id.
2. Choose **Hotspot > Advanced Hotspot Options**
3. In the **Marker Attributes** section, check the box labeled **Is Route**.
4. Click **San Francisco** in the **Tour Navigator**.
5. Choose **Hotspot > Hotspot Actions** in the menu.
6. Choose **JavaScript** from the **Mouseover Action** dropdown.
7. Enter the three lines of code below in the **JavaScript** text box.


```
mapsalive.drawRoute("R1", "SFO, MAD");
mapsalive.drawRoute("R2", "RIO, PER, BOM", 7, 0x007700, 50);
mapsalive.setMarkerHidden("R2", false);
```
8. Choose **JavaScript** from the **Mouseout Action** dropdown.
9. Enter the single line of code below in the **JavaScript** text box.


```
mapsalive.setMarkerHidden("R2", true);
```
10. Click **Tour Preview** and mouse over San Francisco. The routes you see should look similar to those shown in the figure below. When you mouse off of San Francisco, the green route disappears, but the red route stays visible.

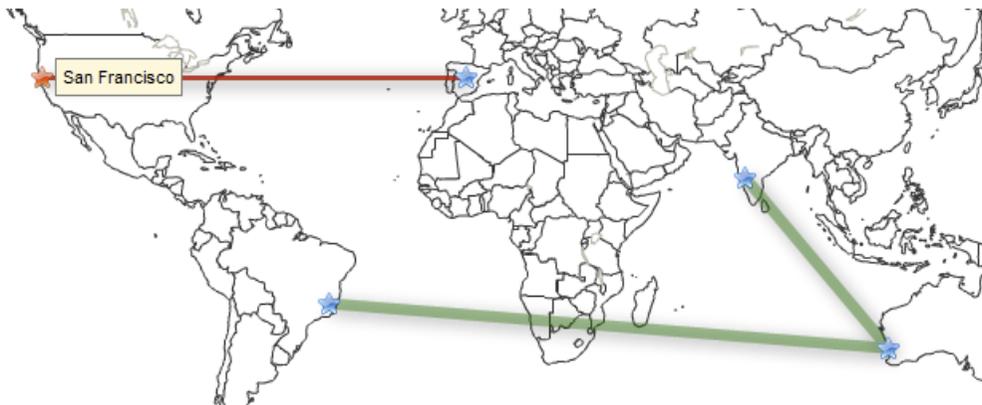


Figure 2 - Map showing two different routes

In the steps above you added a second route hotspot named R2 and used it to draw a 7 pixel wide, green (0x007700), 50% transparent route from Rio to Perth to Mumbai. You also drew a red route from San Francisco to Madrid. You also added logic that makes the green route visible when you mouse over San Francisco and hidden when you mouse out.

This completes the first example. In the next section we'll explain how you can import your routes from a spreadsheet so that you can simply refer to them by name instead of listing each hotspot Id.

5 Importing Routes from a Spreadsheet

The previous sections explained how you draw a route by passing a list of hotspot Ids to the `drawRoute` function. That method works well when you have a small number of simple routes. If your map has dozens or hundreds of routes, you'll want to consider importing your routes.

The Import Routes feature allows you to define and manage large numbers of complex routes in an Excel spreadsheet or CSV file. Each row in the spreadsheet contains a route Id and a set of hotspot Ids that make up the route. When you want to draw an imported route, you refer to it by its route Id like this:

```
mapsalive.drawRoute("R1", "LobbyToWaitingRoom");
```

A row in the spreadsheet can define a complete start-to-end route defined by individual hotspot Ids, or it can combine hotspot Ids with the Ids for route segments as will be explained in the next section.

5.1 Segments

Routes that travel over part of the same path as other routes can make use of segments to avoid duplication of data and make maintenance easier. You can define a segment and then include its Id as part of another route. In a large building there are often primary and secondary corridors that almost everyone walks through to get from point A to point B. These corridors can be defined as segments so that each unique route does not have to define waypoints for corridors over and over again.

5.2 Routes spreadsheet

You import routes from an Excel spreadsheet (or CSV file) containing rows that define routes and segments. The spreadsheet specifies which markers belong to which routes. You identify markers using their Hotspot Id which appears on the **Advanced Hotspot Options** screen. The routes spreadsheet contains one row for each route you want to draw. The columns in the spreadsheet are shown in the table below.

Column	Description
Id	A unique Id for the route. Choose a name that will make it easy to understand what the route is or what its start and end points are.
Route	A list of two or more Hotspot Ids and/or route Ids which describe the route. Normally you use commas to separate the Ids, but you can use a semicolon where there are breaks in the route as explained below in the section on non-contiguous routes. If the waypoints within a segment of the route need to be drawn in reverse order, enclose the segment Id in parenthesis as explained below in the section on reversing segment direction. For example: <code>Wait,P8,(Hall1),Lobb</code> .
Import	Specify True (or TRUE) if the row is for a route that will be drawn on your map. Specify False or leave this column blank to indicate that the row defines a segment used as part of another row's route. Note that a row can be used for both a route that will be drawn and as a segment used by another route. In that case, specify True.

MapsAlive will ignore any other columns in the spreadsheet so feel free to use those for notes. MapsAlive also ignores formatting such as colors and bold text. When you have a lot of routes, it can help to use colors or other formatting to make your spreadsheet easier to read.

5.3 Non-contiguous routes

You can draw a route that has non-contiguous segments by separating the points or segments with semi-colons. The semi-colon tells MapsAlive to pick up the “pen” and move it to the next coordinate before continuing to draw. For example, if you want to draw a route that stops at the edge of a river and then starts again on the other side you can define the points for each part of the route and separate the two groups by a semi-colon like this:

Id	Route	Import
EastSection	H1,H2,H3	
WestSection	H4,H5,H6	
EntireRoute	EastSection;WestSection	TRUE

Figure 3 - Defining non-contiguous routes

When MapsAlive draws the entire route, it will first draw the East section, then pick up the “pen”, move to H4, put down the “pen” and continue drawing the West section. You use a semi-colon instead of a comma anywhere you want to break the route into non-contiguous lines.

5.4 How to import a routes file

After creating the Excel spreadsheet defining the routes, you import it into MapsAlive by choosing **Tour > Import > Routes** in the Tour Builder menu. Then browse for the spreadsheet file and click the **Import** button. During the import process, MapsAlive resolves each Id it encounters by comparing it to the Hotspot Ids present in the tour. If a route references a Hotspot Id that does not exist, MapsAlive will report an error and reject any routes that use that Hotspot Id.

Note that if you have imported routes and then later delete from your map a hotspot that is used in a route, the route will be drawn without going through the missing hotspot.

Anytime you want to modify your routes, simply edit the spreadsheet and import it into MapsAlive again. Whenever you import a routes file, its contents replace any and all existing routes that were previously imported. This differs from importing hotspot content where you can add new hotspots or update a subset of your existing hotspots from imported data.

5.5 Testing imported routes

There are two ways you can test your imported routes. When a tour has routes, the Map screen in the Tour Builder always displays a dropdown list showing the Id of each imported route. You can select a route from the list to have that route drawn as will be shown later in Figure 6.

You can also test your routes from Tour Preview, but to do so you must enable this option by choosing **Map > Advanced Map Options** in the menu and then checking the box labeled **Show Routes List** (it's at the very bottom of the screen). Checking the box will cause a dropdown list to appear in the upper left corner of your map while in Tour Preview. When you select a route Id from the list, the route will be drawn using a dark blue line. The dropdown list will not appear when you run your tour outside of Tour Preview.

6 Route Drawing Example 2 Using Imported Routes

This section presents a sample tour that uses imported routes. You can run the tour, download its archive file, and get its routes spreadsheet using the URLs shown below.

Tour: <http://samples.mapsalive.com/2018g>

Archive: <http://samples.mapsalive.com/Archive/MapsAliveArchiveForTour2018g.zip>

Routes: <http://samples.mapsalive.com/Archive/MapsAliveRoutesForTour2018g.xls>

The sample uses the floor plan for a small hospital having the seven main areas listed below:

- Lobby (Lobb)
- Cafeteria (Cafe) – *has North and South entrances*
- Waiting room (Wait)
- Administration (Admn)
- Pathology (Path)
- Radiology (Radi)
- Emergency (Emrg) – *has East and West entrances*

The tour lets you choose a route starting from either the lobby or the waiting room to the other six rooms. If a room has two public entrances, the route takes you to the closest entrance from the start of the route. Where the route travels through closed doors, a gap appears on either side of the doors. The figure below shows the route from the lobby to the east entrance of the emergency room.



Figure 4 – Route from the Lobby to the East entrance of the emergency room

6.1 The Spreadsheet

The routes for the hospital sample are defined in the spreadsheet shown in the figure below.

	A	B	C
1	Id	Route	Import
2	Hall1	P1,P2,P3	
3	Hall2	P8,P3	
4	LobbToEmrg	Lobb,Hall1,P6,P12;P19,EmrgW	TRUE
5	LobbToPath	Lobb,P1,P14,P15;P10,Path	TRUE
6	LobbToAdmn	Lobb,Hall1,P4,Admn	TRUE
7	LobbToRadi	Lobb,P1,P11,P18,P16,P17,Radi	TRUE
8	LobbToCafe	Lobb,P1,P14,CafeS	TRUE
9	LobbToWait	Lobb,Hall1,P8,Wait	TRUE
10	WaitToEmrg	Wait,P8,P7,P13;P20,EmrgE	TRUE
11	WaitToPath	Wait,Hall2,P2,P15;P10,Path	TRUE
12	WaitToAdmn	Wait,P8,P4,Admn	TRUE
13	WaitToRadi	Wait,P8,P5,P9,Radi	TRUE
14	WaitToCafe	Wait,Hall2,CafeN	TRUE
15	WaitToLobb	Wait,P8,(Hall1),Lobb	TRUE

Figure 5 - Excel spreadsheet defining routes for the hospital sample

Rows 2 and 3 define two segments, Hall1 and Hall2. Hall 1 runs North/South between the cafeteria and the pathology. Hall 2 runs East/West above pathology and emergency. In this simple example, these segments have only a few waypoints, but in a real hospital, a main hall or corridor could have many waypoints. Using a segment not only keeps you from having to specify the same waypoints over and over again in each route that uses the segment, but it also means that you can edit a segment and automatically alter all routes that use it. For example, suppose a detour was required in a corridor as a result of a long term construction project. You could add the detour to the segment (and remove it when the construction was finished) without having to change any of the routes that use that segment.

Rows 4 through 15 define 12 routes, six that start in the lobby and six that start in the waiting room. The Hall1 segment is shared by routes LobbToEmrg, LobbToAdmn, and LobbToWait. The Hall2 segment is shared by WaitToPath and WaitToCafe. Note that some of the waypoints are separated by semicolons instead of commas. The semicolons are used to create gaps where a route goes through closed doors.

Note that the Import column of all rows specifies TRUE except for rows 2 and 3 which define segments. The tour never draws routes using the Hall1 or Hall2 Ids and therefore does not import them into the tour. By telling MapsAlive which routes not to import, you help it optimize the code that it generates for the tour.

6.2 The Map

The figure below shows a screen shot of the Tour Builder's **Map** screen. This is what you see when you are placing waypoint and other markers. In this tour, all of the map's markers are invisible (the waypoint markers and the You Are Here markers), but on the Map screen they appear semi-transparent so that you know where they are.

The Map screen has a dropdown (not shown in the figure) that lets you choose one of the routes imported from the spreadsheet so that you can see it while working on your map. The route is drawn using a dark blue line. In the figure, the route shown in dark blue is LobbToEmrg.



Figure 6 – Annotated screenshot showing what the map looks like on the Tour Builder's Map screen

To help you better understand the spreadsheet, we have annotated the screenshot above with light blue text to label the waypoints that make up the LobbToEmrg route (these labels do not actually appear in the Tour Builder).

The LobbToEmrg route is defined in the spreadsheet (see Figure 5) as:

```
Lobb,Hall1,P6,P12;P19,EmrgW
```

Follow the dark blue line to see how this route is drawn. It starts at the Lobb marker, then travels to P1 which is the start of the Hall1 segment. It leaves the Hall1 segment at P3 and travels to P6 where it makes a right turn to P12. The semicolon following P12 tells MapsAlive to lift the pen and then move to P19 in order to create a gap in the route at the doorway. The route then continues from P19 to EmrgW.

6.3 Reversing segment direction

If the waypoints within a segment of a route need to be drawn in reverse order, enclose the segment Id in parenthesis. You do this when a segment is used for multiple routes that traverse the segment from either direction. The figure below shows why you sometimes need to reverse a segment.



Figure 7 – A correctly drawn route (left) and an incorrectly drawn route (right)

In the figure above, the Hall1 segment shown in orange is made up of waypoints P₁, P₂, and P₃. P₁ is at the bottom of the segment and P₃ is at the top. This sequence is correct for a route that starts from the lobby at the map's lower left and goes north through the segment, but the sequence is backward for a route that needs to traverse the segment in a southerly direction.

The route on the left is drawn starting at the You Are Here marker in the waiting room, through the reversed Hall1 segment, and to the lobby like this:

```
Wait,P8,(Hall1),Lobb.
```

By enclosing Hall1 in parenthesis, the Hall1 path was drawn top to bottom in this sequence: P₃, P₂, P₁.

The route on the right is drawn the same way but without reversing the Hall1 segment:

```
Wait,P8,Hall1,Lobb.
```

As a result, the Hall1 segment's path was drawn bottom to top in its normal P₁, P₂, P₃ sequence. This caused the pen to go from P₈ (the first bend in the path after the You Are Here marker), then cut across the map to P₁, then up to P₃, and then diagonally down to the lobby.

6.4 HTML and JavaScript

The hospital sample tour uses the MapsAlive Custom HTML feature to display a table at the top of the tour along with some instructions as shown in Figure 8 below. You could accomplish the same thing by coding the HTML in a web page that contains the tour. If you import the archive file for this sample you can see the custom HTML by choosing **Tour > Custom HTML** in the menu.

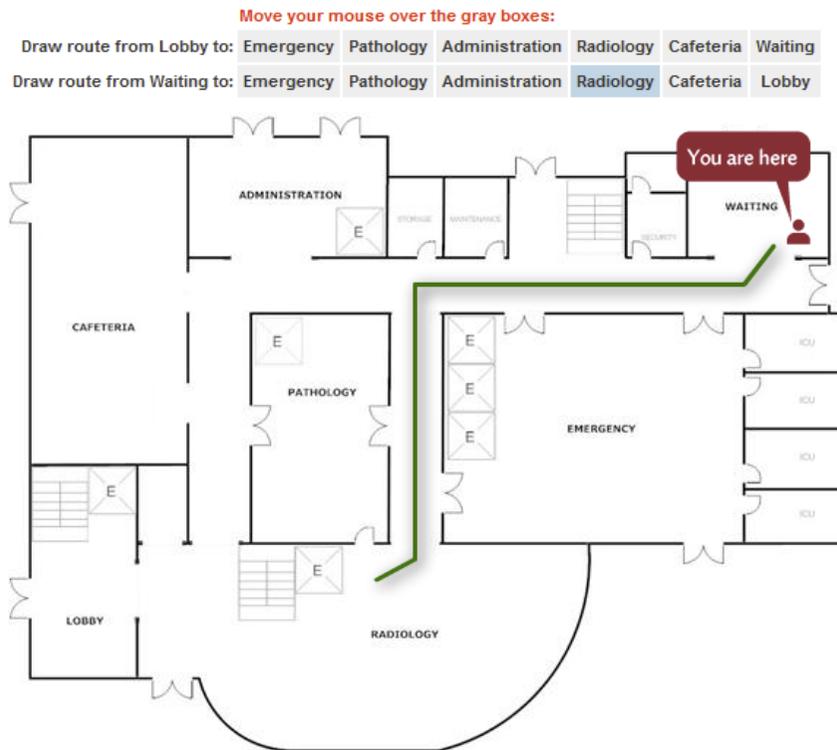


Figure 8 – Screenshot showing what the sample tour looks like when you run it

Each cell in the table at the top of the tour has an `onmouseover` handler that calls a JavaScript function named `showRoute` (see Figure 9 below). Each handler passes the Id of the route displayed in the cell. The `onmouseover` handler for the Emergency cell in the first row looks like this:

```
onmouseover="showRoute('LobbToEmrg');"
```

The `showRoute` function (Figure 10) uses the first four letters of the route Id to determine whether the route originates from the lobby or from the waiting room so that it can show the appropriate You Are Here marker. The `showRoute` function also calls the MapsAlive API function `drawRoute` passing the route Id along with parameters that specify the appearance of the route's line. The call looks like this:

```
mapsalive.drawRoute("Route", routeId, 3, 0x007700, 100, "shadow");
```

Note that in the call above, the first parameter "Route" is the Id of the route hotspot as was explained in section 3.2 above. The second parameter is an Id from the spreadsheet e.g. "LobbToEmrg."

HTML from the Top section of the Custom HTML screen

```
<table class="chart">
  <tr>
    <td/>
    <td colspan="6" style="color:red;">Move your mouse over the gray boxes:</td>
  </tr>
  <tr>
    <td style="text-align:right;">Draw route from <b>Lobby</b> to:</td>
    <td class="box" onmouseover="showRoute('LobbToEmrg');">Emergency</td>
    <td class="box" onmouseover="showRoute('LobbToPath');">Pathology</td>
    <td class="box" onmouseover="showRoute('LobbToAdmn');">Administration</td>
    <td class="box" onmouseover="showRoute('LobbToRadi');">Radiology</td>
    <td class="box" onmouseover="showRoute('LobbToCafe');">Cafeteria</td>
    <td class="box" onmouseover="showRoute('LobbToWait');">Waiting</td>
  </tr>
  <tr>
    <td style="text-align:right;">Draw route from <b>Waiting</b> to:</td>
    <td class="box" onmouseover="showRoute('WaitToEmrg');">Emergency</td>
    <td class="box" onmouseover="showRoute('WaitToPath');">Pathology</td>
    <td class="box" onmouseover="showRoute('WaitToAdmn');">Administration</td>
    <td class="box" onmouseover="showRoute('WaitToRadi');">Radiology</td>
    <td class="box" onmouseover="showRoute('WaitToCafe');">Cafeteria</td>
    <td class="box" onmouseover="showRoute('WaitToLobb');">Lobby</td>
  </tr>
</table>
```

Figure 9 – Custom HTML used to display the tour's route selector table

JavaScript from the JavaScript section of the Custom HTML screen

```
function showRoute(routeId)
{
  // Determine which "You Are Here" marker to show and which to hide.
  var showYouAreHereWaiting = routeId.substr(0, 4) == "Wait";
  mapsalive.setMarkerHidden("YouAreHereLobby", showYouAreHereWaiting );
  mapsalive.setMarkerHidden("YouAreHereWaiting", !showYouAreHereWaiting );

  // Draw a route that is a 3px green line at 100% opacity with a shadow.
  mapsalive.drawRoute("Route", routeId, 3, 0x007700, 100, "shadow");
}
```

Figure 10 – JavaScript function to draw a route when the mouse moves over a route name in the selector table

Appendix A – the drawRoute JavaScript API function

The information below is from the *MapsAlive User Guide for the JavaScript API*.

The drawRoute function draws a line through a set of hotspot markers. The route is drawn through the center point of each marker. Signature:

```
mapsalive.drawRoute  
  (hotspotId, route, lineWidth, lineColor, lineAlpha, effects);
```

Examples:

```
mapsalive.drawRoute("Route", "H1,H2,H3", 3, 0x00CC00, 50, "Shadow");
```

```
mapsalive.drawRoute("Route", "R1", 3, 0x00CC00, 50, "Shadow");
```

Parameters:

Parameter	Description
hotspotId	A hotspot Id that identifies the route marker to use to draw the route. The hotspot must have its Is Route option checked on the Advanced Hotspot Options screen. If the marker was previously used to draw a different route, the call will erase the prior route before drawing the new route.
route	The route parameter must be one of the following: - The name of a route that was imported into MapsAlive from an Excel file. The name must be enclosed in quotes. - A comma-separated list of two or more hotspot Ids enclosed in quotes. The center point of the marker for each hotspot identified in the list will be used as a waypoint along the route. To create a gap in the route, use a semicolon instead of a comma. The semicolon tells the route drawing logic to “pick up the pen” after drawing to the waypoint that precedes the semicolon and to put it down again at the waypoint following the semicolon.
lineWidth	<i>Optional.</i> The width of the route’s line in pixels. Default is 3.
lineColor	<i>Optional.</i> The color to use for the line. Default is 0xff0000.
lineAlpha	<i>Optional.</i> The opacity to use for the line. Default is 100.
effects	<i>Optional.</i> The effects to use on the line. Default is “shadow”. For no effects, provide an empty string “”. To learn about effects see the <i>MapsAlive User Guide for the JavaScript API</i> .