

Basic SIMMOD Study Creation Tutorial

Featuring Visual SIMMOD

Copyright:

AirportTools

March 2005

All Rights Reserved.

Introduction

This tutorial is designed to help you build your first SIMMOD study.

After completion of this tutorial you will have learned:

- Where to start when creating a SIMMOD study.
- Minimum study data requirements.
- The sequence of steps to create a new SIMMOD study.
- The basics of SIMMOD data management and editing
- Running a SIMMOD study.

Keys to SIMMOD Success

- Time management. Do not underestimate the time to create a SIMMOD study. Even the most seasoned veterans find a new SIMMOD study challenging.
- Thorough knowledge of how the airport (and/or airspace) operates is critical.
- Willingness to read the documentation. SIMMOD is a very complex data driven model, it is critical to spend time reviewing and understanding the available documentation. Some of the most critical documents are as follows:
 - Flat-files documentation. The inputs to SIMMOD are defined in this document. You must become acquainted with the SIMU01, SIMU03, SIMU07, and SIMU09 files via this documentation. Available online at:
<http://www.airporttools.com/jsimmod/Documentation/input/flat/index.html>
 - Tips and tricks documentation. Available online at:
<http://www.airporttools.com/jsimmod/Documentation/other/tips/index.html>
 - Debugging SIMMOD. Available online at:
<http://www.airporttools.com/jsimmod/Documentation/other/howto/tng/index.html>
 - SIMMOD Outputs. Available online at:
<http://www.airporttools.com/jsimmod/Documentation/output/simu26/index.html>

Keys to SIMMOD Success

- Communications are critical. You must be able to interpret the study requirements, possibly from a third party client, and translate them into SIMMOD solutions.
- Interaction with others. Similar to the above communications bullet, you should take advantage of the wealth of free knowledge available from other SIMMOD users via the [European SIMMOD Users Group](#), the [North American SIMMOD Users Group](#), and the [Yahoo SIMMOD Group](#).
- Training courses if necessary. A few firms make SIMMOD training courses available to the public several times a year. Sign up for one, they are well worth it since they are usually taught by experts in the field.
- Obtain SIMMOD support if necessary. Most of the SIMMOD tools vendors offer support contracts. And while they probably will not be willing to perform your study for you they generally can offer expert advise when asked in a concise manner.

Document Guide

The following conventions are used within this document:

GTE A Text which should be typed in by the user.

File//Print//GIF A menu sequence in Visual SIMMOD.

Ok A button which should be pressed in Visual SIMMOD.



An idea or a thought which should be considered.



A note providing extra important descriptive material.



A critical message to which users must adhere.



Usage of the left mouse button is recommended for the task.



Usage of the middle mouse button is recommended for the task.



Usage of the right mouse button is recommended for the task.



Object in list is new.



Object in list has been edited.



Object in list has been marked for deletion.

Document Guide (continued)

The sections contained within this document include:

- Gathering SIMMOD data recommendations.
- Creation of a new SIMMOD study in Visual SIMMOD.
- Basic Visual SIMMOD usage instructions:
 - Setting graphics layer's visibilities.
 - Zooming and panning.
 - Saving a map graphics scene.
 - AutoCAD DXF file format conversion to QGF graphics file format.
 - Permanently displaying QGF graphics files on the map.
 - Understanding database entry errors.
- SIMMOD static (airfield, airspace structures) data creation.
- SIMMOD events (flights, etc.) data creation.
- Running SIMMOD.

Gathering SIMMOD Data

For the purposes of this tutorial, no additional data beyond this document is required from you. But, for your own study, you should consider gathering the following data before beginning a new study:

- Airport/facility directories.
- CAD drawings of relevant airports' runway and taxiway layouts, specifically DXF formatted files.
- Airway maps.
- Proposed runway/taxiway/gate/concourse/airspace structure details.
- Airline flight schedules.
- Airline fleet mixes.
- Historical wind direction conditions.
- Operations counts by runway.
- Runway operating procedures, e.g. Time between operations on related runways
- Departure queue locations and capacities.
- Taxiway operations, e.g. Locations, passing rules, speeds, directionality.
- Gate operations, e.g. Locations, blocking/blocked situations, power or pushback, max size of aircraft allowed.
- Staging areas and the conditions when they are used.
- Deicing areas and the conditions when they are used.

General Program Usage Note

Visual SIMMOD, and this tutorial, depends upon usage of items which are either drawn, displayed, or edited upon Visual SIMMOD's map graphics capability.

Thus, the first portion of this tutorial will focus on getting the user familiar with how to:

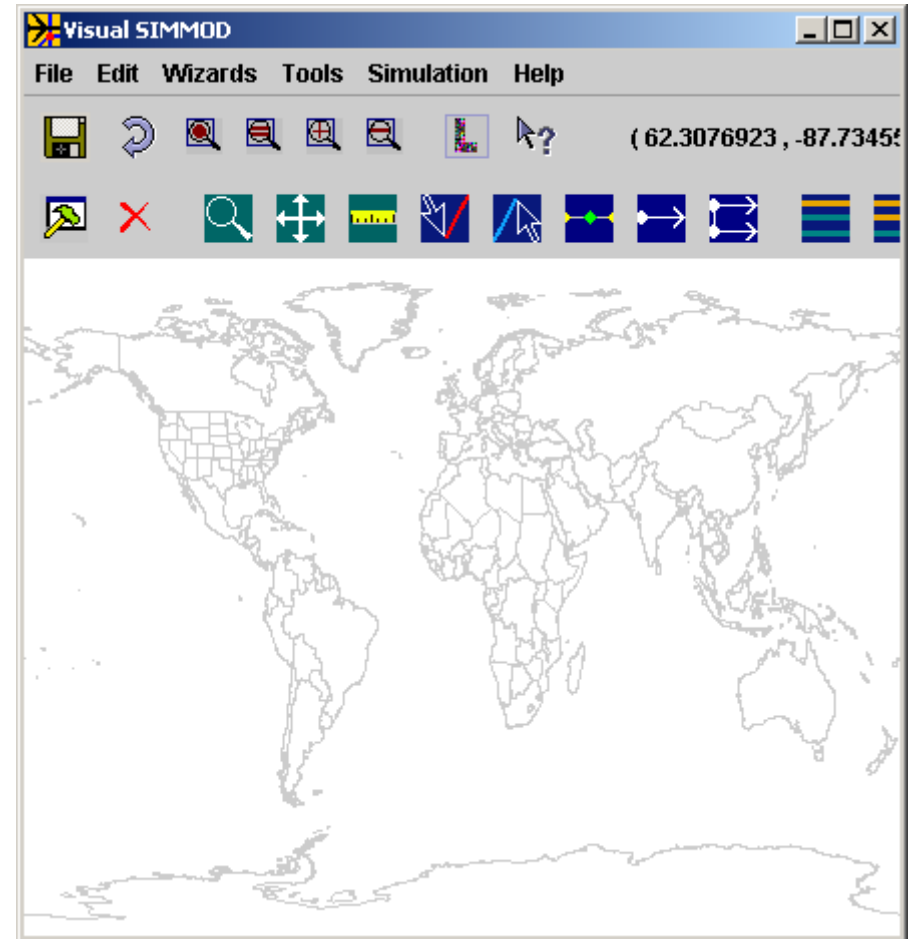
- Create a new SIMMOD application directory.
- Turn the visibility of the map graphics on or off.
- Zoom and pan in the map graphics.
- Save a particular scene within the map graphics.

In the remainder of this tutorial the window to the right is referred to as the “Network Builder”.

The Network Builder is the main Visual SIMMOD window and it's purpose is to display relevant map graphics. Layers such as coastlines, CAD drawings, and SIMMOD study data are displayed on the map graphics.

The Network Builder also allows the editing of SIMMOD data via usage of the mouse.

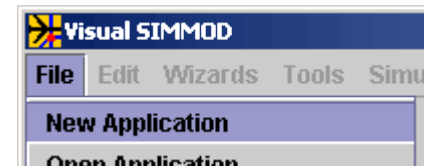
Lastly, the Network Builder's menu system allows access to the other tools within Visual SIMMOD.



Creating a New SIMMOD Study

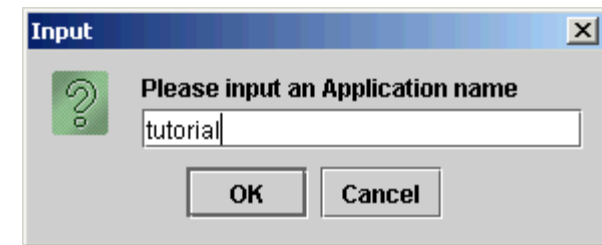
To create a new SIMMOD study you should first open the Visual SIMMOD program. After the program loads you should execute:

File//New Application



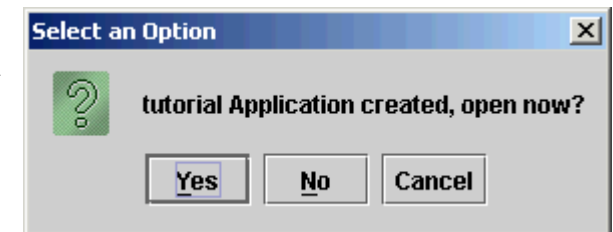
After issuing the above command an input dialog will open requesting you supply a unique study identifier. For this tutorial please type in the following:

tutorial



Then press the **OK** button.


Visual SIMMOD will now create a new Application called “tutorial” in the Applications subdirectory where Visual SIMMOD is installed. A popup window will ask you if you want to open the “tutorial” application. Select the **Yes** button to continue.





When naming SIMMOD studies most users will want to establish a naming convention which is both short and descriptive. For example, if your airport identifier is JFK a possible naming convention might be “JFK_East” for east wind conditions and “JFK_North” for north wind conditions.

Setting Layer Visibilities

When Visual SIMMOD creates a new application all of the graphics layers have their visibilities turned on and a random color assigned to them. Having everything visible simultaneously can prove to be detrimental to productivity. Thus, this page describes how to turn off (and on) the visibility of graphics layers.

To set a layer's visibility in Visual SIMMOD you should first click the “Set Attributes” button  on the top row of the toolbar.

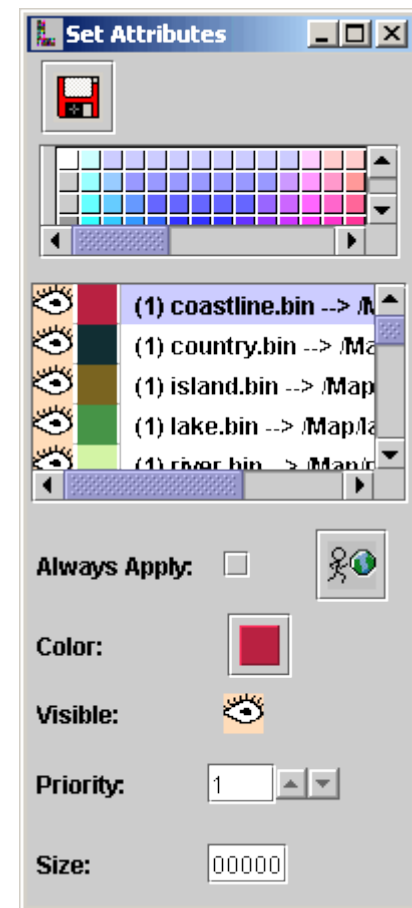
As shown at the right, the Set Attributes window will be opened. You can pick graphics layers from the list to modify. When chosen, the layer's attributes will be displayed in the lower portion of the window.

Clicking on the  button in the lower portion of the window will toggle the visibility of the item to the off condition. To view the results you may need to press the  button.

It is recommended that you toggle the visibility off for all layers except for the following items, which should remain on:

- Coastline.bin
- AIRSPACE LINKS
- AIRFIELD LINKS
- AIRSPACE NODES
- AIRFIELD NODES
- SELECTED OBJECTS


To save your attribute settings press the  button.




Zooming and Panning

One of the most fundamental skills you will need to master before proceeding is how to zoom and pan in the Network Builder's map graphics window.

Visual SIMMOD has been optimized to work best with a three button mouse. For those without a three button mouse Visual SIMMOD can zoom and pan via just the left mouse button.

! Please note: If during this tutorial you cannot locate where you are on the map graphics you can press the  button to immediately take you back to a map of the world.

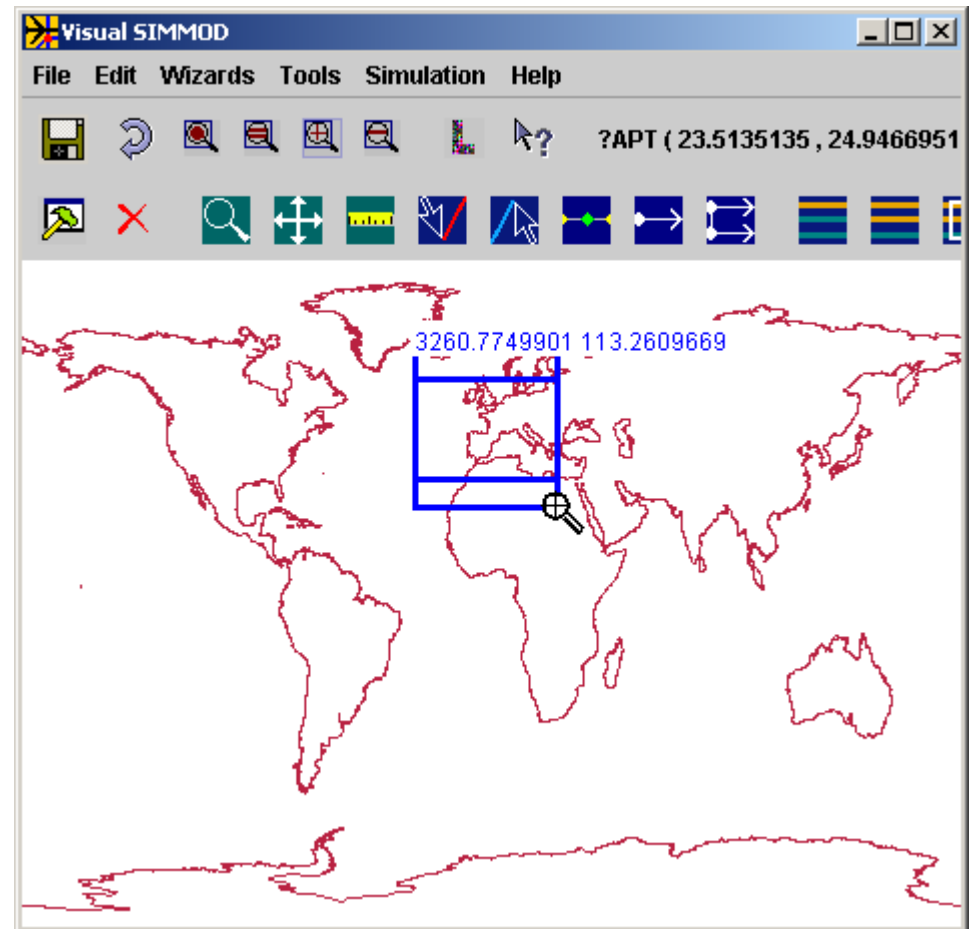
 Users with a three button mouse may press the middle mouse button on the Network Builder map graphics to begin the zoom process.

While holding down the middle mouse button, a rectangle should be drawn around the interest area.

It is important that the user select from the upper-left to the lower-right as shown in the example.


When the box is correct releasing the mouse performs the actual zoom.

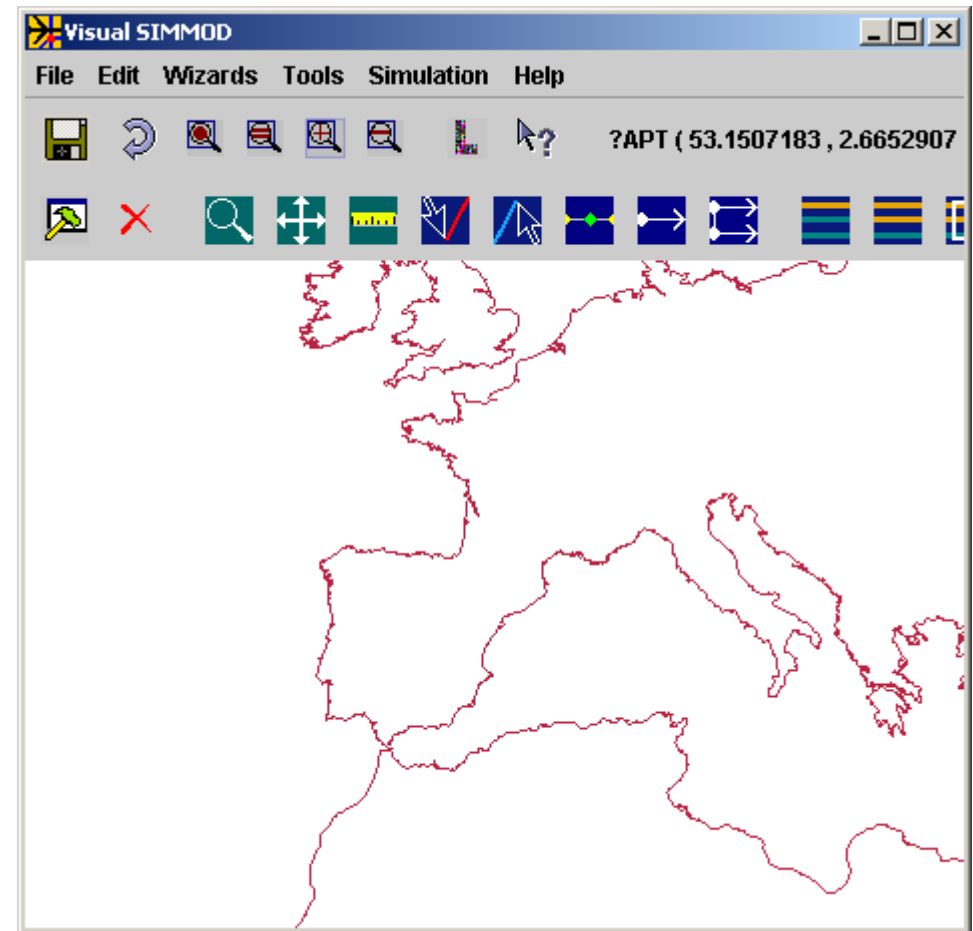
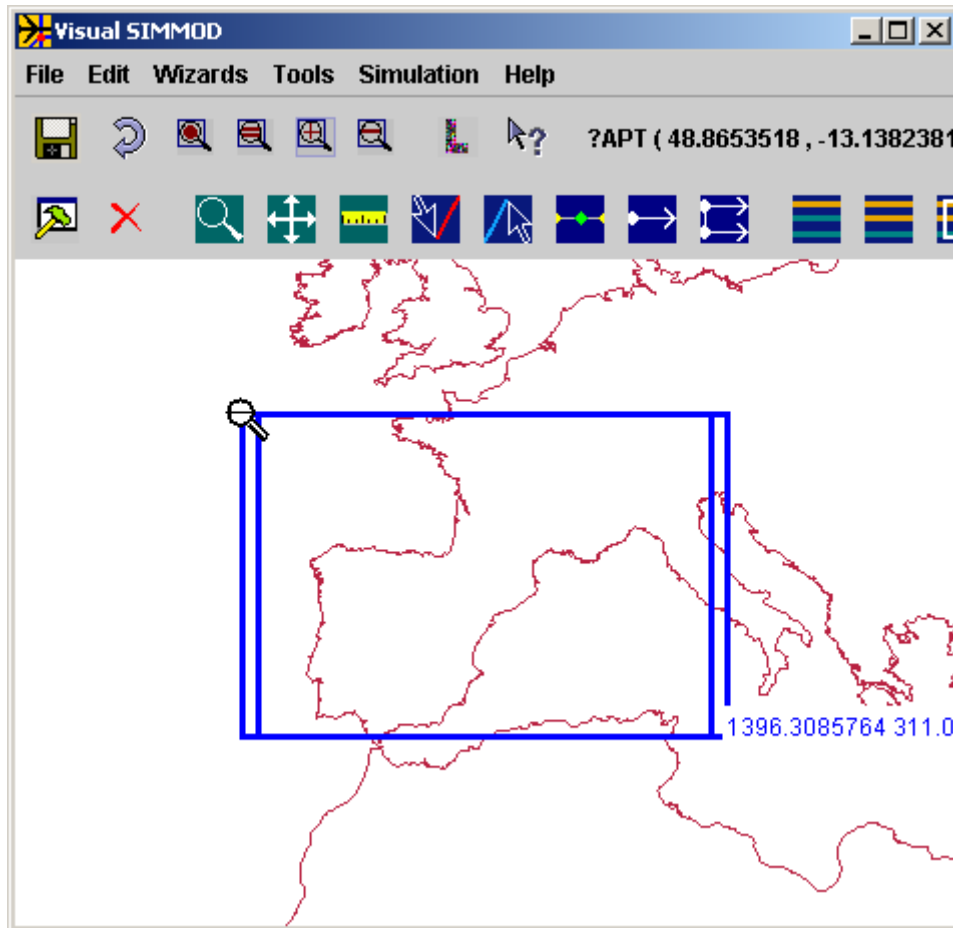
The snapshot on the following page displays the results of the example shown to the right.



Zooming and Panning (continued)


As seen to the right, the zoom operation has resulted in most of Southern Europe and some of Northern Africa becoming more visible.

 To zoom out, the user need only press the middle mouse button again and while holding it down drag the mouse from the lower right to the upper left.

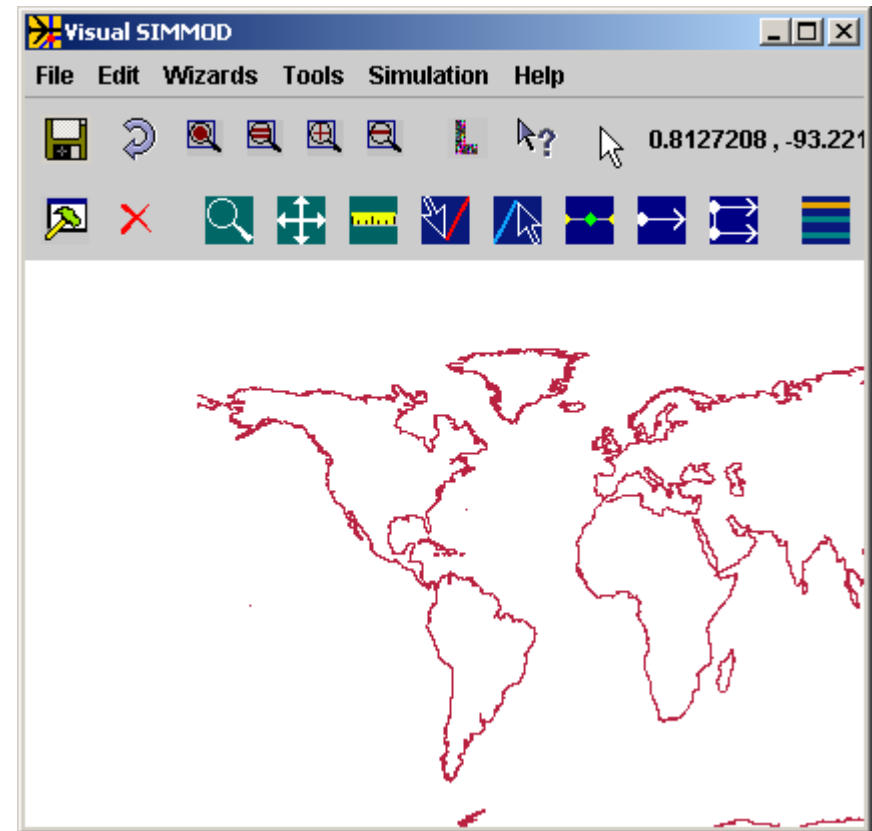
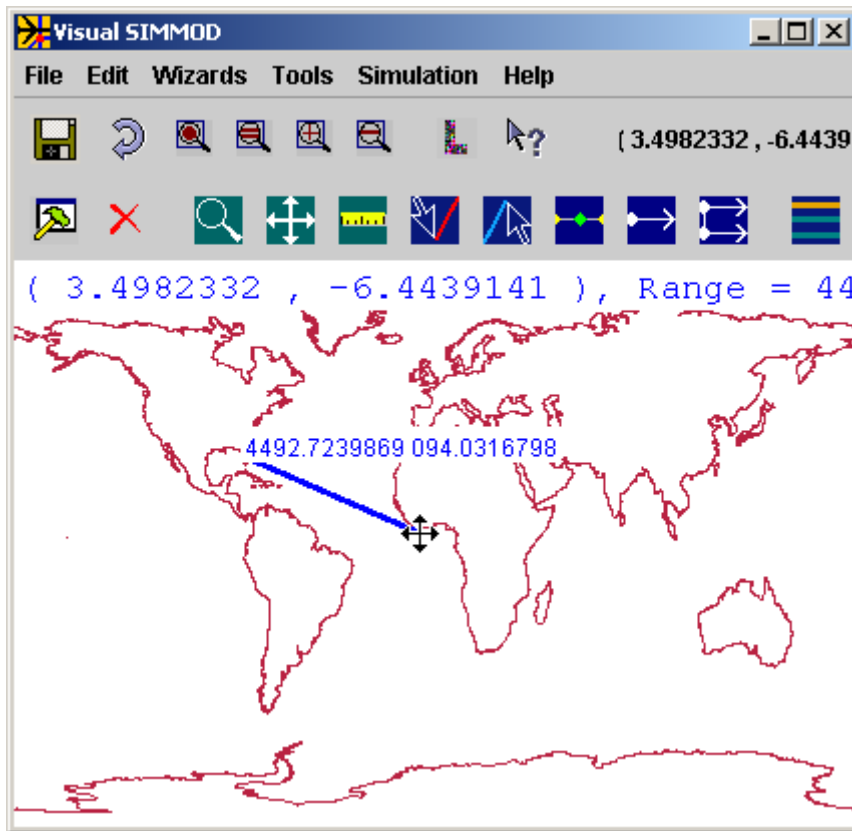



An example of a zoom-out operation is shown to the left:

Zooming and Panning (continued)

For those who have only a two button mouse you may press the  button to start the zoom process and then use the left mouse button with the previously described procedures for three button mouse users.


Panning in Visual SIMMOD is accomplished by placing the mouse over the map graphic you would like to pan to a new location, clicking down the right mouse button, dragging a line to where the map should be positioned and then releasing the mouse. The before and after illustrations below demonstrate panning:



For those who have only a two button mouse you may press the  button to start the panning process and then use the left mouse button with the just described procedures for three button mouse users.

Setting Scenes

It is expected that users will want to have a set of map graphics scenes which can be recalled to speed map viewing. For example, users may want saved scenes that can take them immediately to their runways, gate areas, particular airspace fixes, and other graphical elements.

To save a scene press the  button on the Network Builder window. After pressing the button a popup will open allowing you to enter a new scene name. Enter **Default** and press the **OK** button to save the current scene.



The scene name *Default* is a special scene name. If, when initially executing, Visual SIMMOD detects that a scene named Default exists then that scene will be displayed on initial program loadup.

Scenes can be recalled by pressing the  button.

Importing a CAD Drawing

When initially building a SIMMOD study it is often useful to be able to import a CAD drawing as a background layer which can then serve as a template for tracing purposes.

Visual SIMMOD has the ability to import an AutoCAD DXF formatted drawing. DXF drawings' coordinate systems can be scaled, rotated, and translated from their original size and position by Visual SIMMOD as well.



Included with this tutorial you should have received a sample DXF file called tutorial.dxf. This sample DXF file will serve as the template for the airport which will be built in this tutorial. You should locate the tutorial.dxf file now before proceeding.

After locating the tutorial.dxf file it is suggested that you make a copy of it in the graphics subdirectory of your Visual SIMMOD installation. Depending on your choice of installation directories, on Microsoft Windows, this directory is normally:

`c:\AirportTools\JSIMMOD\graphics`


During the import process Visual SIMMOD converts the basic geometry information in the DXF file into a file format called a QGF file.

The QGF file format is documented on the AirportTools website and you may edit it by hand if necessary using your favorite ASCII text editor.

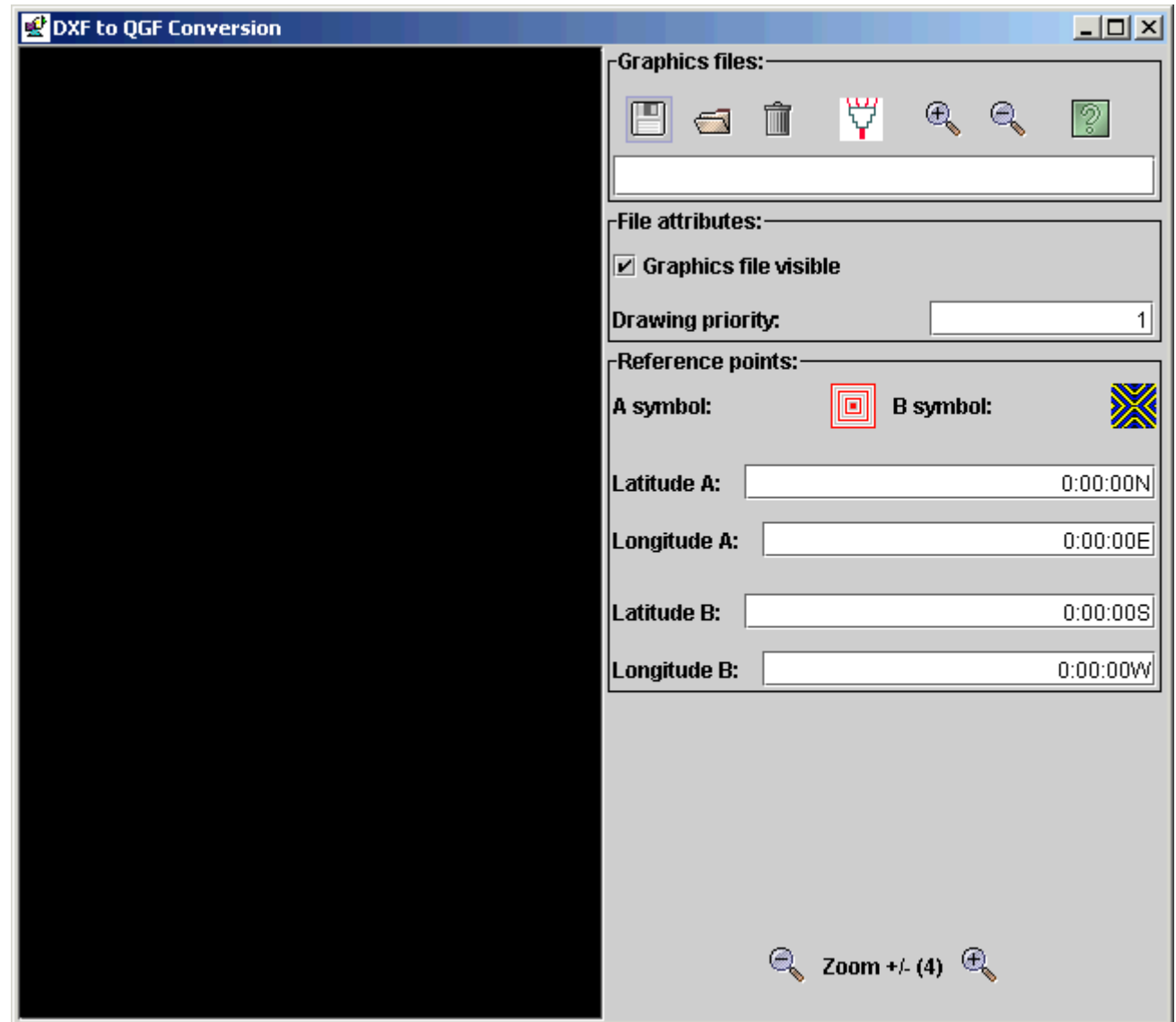
Importing a CAD Drawing (continued)

The first step in importing a DXF file into Visual SIMMOD is to open the “DXF to QGF Conversion” window via the menu commands:

File//DXF to QGF Conversion (Improved)

The DXF to QGF Conversion window is shown to the right. If you properly placed your tutorial.dxf file in the graphics subdirectory then it should be listed when you press the  button

Please click on the “tutorial.dxf” entry to signify to the program that this is the file which will be converted to a QGF file.



Importing a CAD Drawing (continued)

After opening the tutorial.dxf CAD drawing it should be displayed in the graphics window.

Locate the reference point A in the graphics window and drag it to the lower left portion of the CAD drawing as shown to the right.

Next locate the reference point B and drag it to the lower left portion of the CAD drawing as shown to the right.

Next, in the Latitude A field type [27.00000572](#).

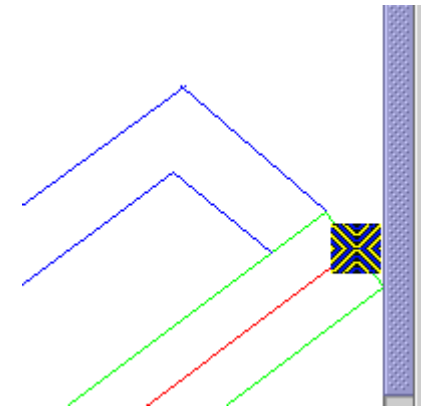
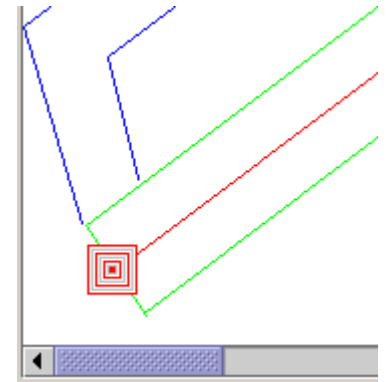
In the Longitude A field type [-80.99997108](#).

In the Latitude B field type [27.00662549](#).

In the Longitude B field type [-80.98023327](#).

Note: In any of the above fields you may use the format `##:##:##.##N(SEW)`.

Next press the  button to convert the DXF file to QGF format.



Importing a CAD Drawing (continued)





Assuming your inputs matched the specified inputs the drawing will be located approximately between Tampa Bay and Miami in Florida of the far Southeastern United States.

After the convert dxf button has been pressed the layer should look similar to the snapshot to the right.

If the drawing looks somewhat skewed you may resize the Network Builder to correct the skew.

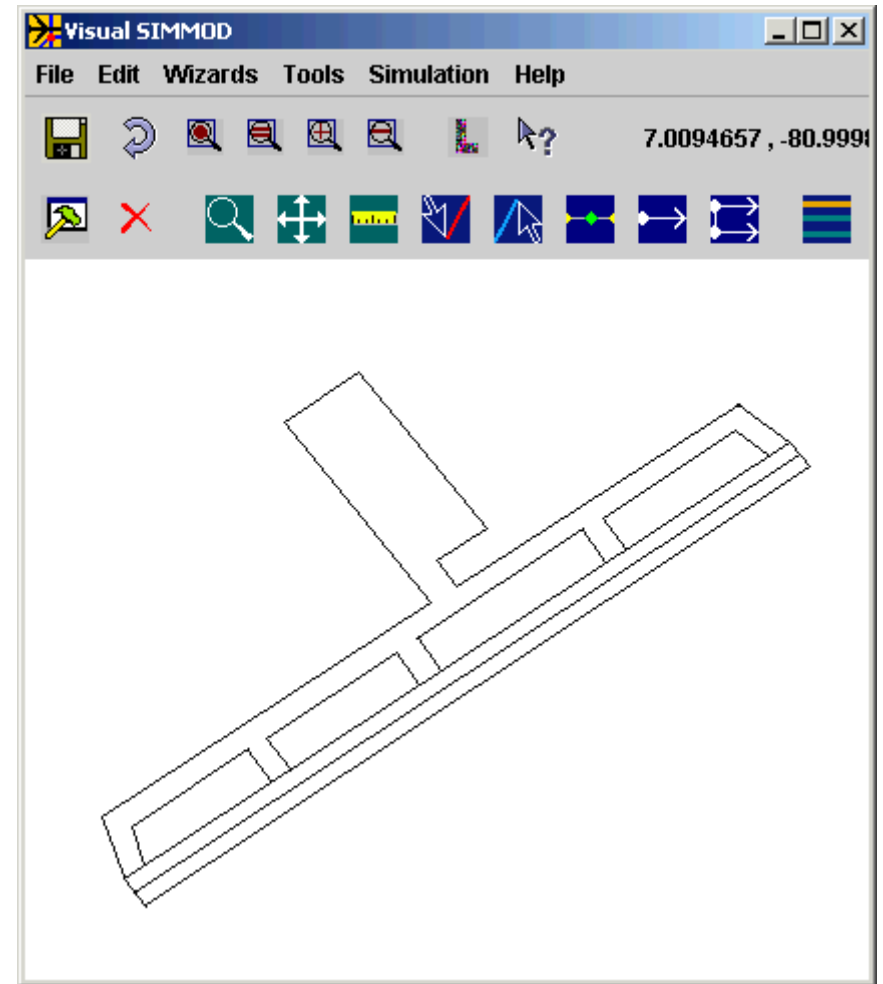
You may need to zoom and pan in order to locate the display of the airport drawing.

 After zooming in and confirming the conversion process went properly you may want to set the current view as the special “Default” scene.

 Please note, the import process as just described merely prepares the QGF to be used as a layer within Visual SIMMOD. When you exit Visual SIMMOD and then re-enter the program the just converted DXF file will no longer be displayed.



The next section of the tutorial describes how to specify which QGF layers are displayed and which aren't.

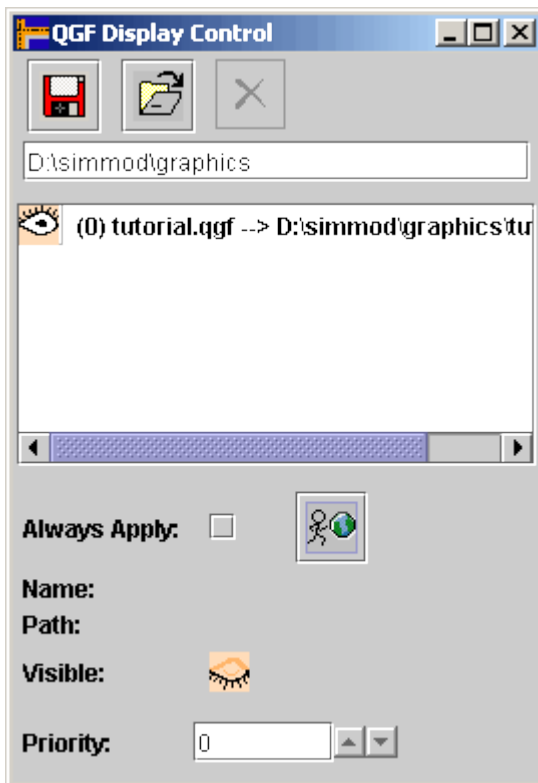
You may close the DXF to QGF Conversion window.




QGF Display Control

To specify which QGF files to display as layers you need to execute the **File//QGF Display** menu item. The QGF Display Control window is shown to the right without any QGF files assigned.

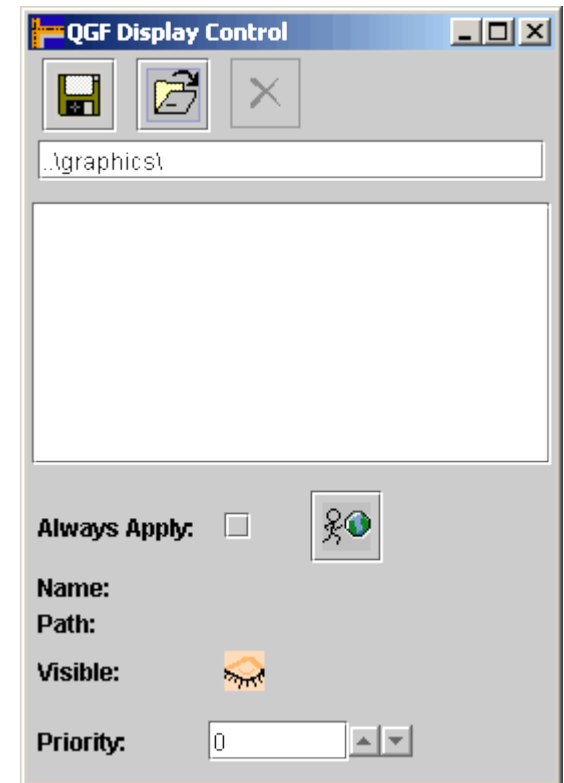
To assign a QGF to be displayed press the  button. Locate the tutorial.qgf file and open it. Press the  button to refresh the Network Builder's map graphics.



After the tutorial.qgf file has been assigned to be displayed the QGF Display Control will appear as shown on the left.

Before exiting the QGF Display Control you should press the  button to save your settings.

You may close the QGF Display Control window.



Understanding Database Errors

Before proceeding to enter actual SIMMOD data a description of Visual SIMMOD's errors is needed.

Visual SIMMOD includes a database system which manages your SIMMOD inputs for you. Before SIMMOD input data is stored to disk the database performs internal checks which verify that the data to be written is correct.

An example might be a check to confirm that the length of a taxiway link is greater than zero.

If erroneous data is sent to the database then the database detects it, tells Visual SIMMOD what the error is, and then Visual SIMMOD passes this error message on to the user.

The database error message is very detailed and is designed more for expert users. But, since you may make a mistake and cause a database error message to be displayed it is worthwhile to explain the relevant portions of such messages to you.

A sample error is shown to the right:

There are four separate parts to the error message.



The first portion of the error message extends up to, but not including, the text “exception 121”. The first part merely indicates which database system has had an error. Visual SIMMOD only includes one database system but it is conceivable that multiple database systems could be in use at the same time.

You may ignore the first portion of the message.

Understanding Database Errors (continued)

The second part of the error message is the “exception 121” portion. This is merely a quick reference number for a programmer or database administrator to know where to look for a problem.

You may ignore the second portion.



The third part of the error message is the “APT_ID not in DEGENERATE_AIRPORT” portion. This is the user-friendliest part of the error message. This portion was written by a human being for another human being to read. You should pay close attention to this portion of the error message.

Unfortunately the database system only allows the third portion of the message to be eighty characters long. Thus, sometimes very creative efforts had to be made by the person writing the message to describe the problem.

In the above case, the message simply is indicating that the supplied APT_ID (JFK in this case) cannot be found in the table called DEGENERATE_AIRPORT.

The fourth portion of the error message is the computer code (SQL statement for those computer nerds among us) that describes what Visual SIMMOD wanted the database to do. The fourth portion of the error message is also valuable and generally should be examined for the nature of the problem.

In the above example, the fourth portion indicates that the database should insert into the AIRPORT table three pieces of information. One of the pieces is the airport identifier APT_ID which is set to JFK.

Understanding Database Errors (continued)

But, as we earlier established from the third portion of the error message, the APT_ID given, JFK, is not in the DEGENERATE_AIRPORT table.

The solution to this particular problem would be to insert JFK into the DEGENERATE_AIRPORT table and re-try the process that created our error.



Most of the Visual SIMMOD editing tools perform checks on the data before it is even sent to the database system for storage. If an error is found by one of these “Pre” checks then a more user-friendly message is displayed for the user.

Ultimately, the database system has the final say as to whether the data is correct or not.

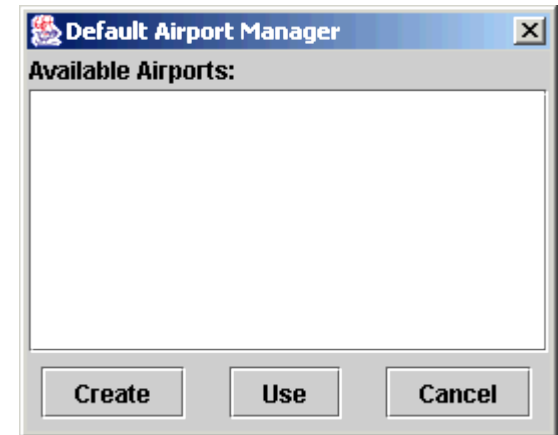
In general, the database does not check your SIMMOD data's overall logical structure. The database merely checks the validity of items.

Airport Definition

The first step in the creation of SIMMOD study data is the definition of your airport. All airfield objects such as gates, taxiways, and runways must be associated with a distinct airport. For example, Visual SIMMOD requires the definition of an airport before a gate can be created, since the gate must be assigned to a pre-existing airport.

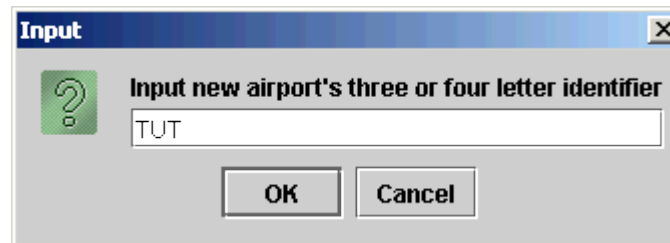
To create an airport you should execute the menu tree *Edit//Set Current Airport*.

As displayed to the right the Default Airport Manager tool will open after the menu tree is executed. Press the *Create* button to input a new airport.

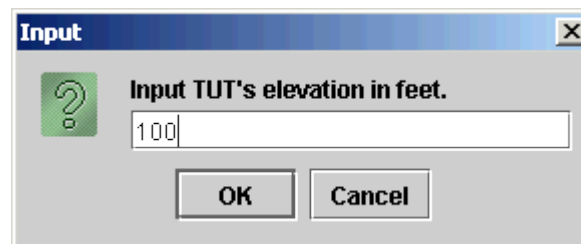


When the Create button is pressed a sequence of three popup windows will open asking you for basic information about the airport. You will be asked for the airport's three or four character LOCID, it's elevation in feet, and a brief description of the airport, usually the name it is known by in the community.

Enter **TUT** as the LOCID (airport identifier) and press the *OK* button:

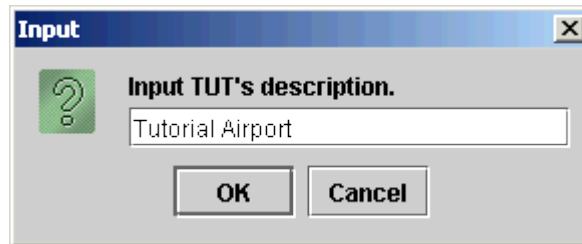


Enter **100** for the elevation and press the *OK* button:



Airport Definition (continued)

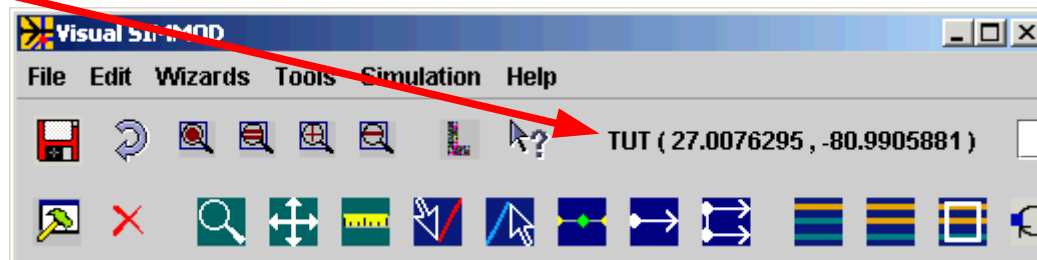
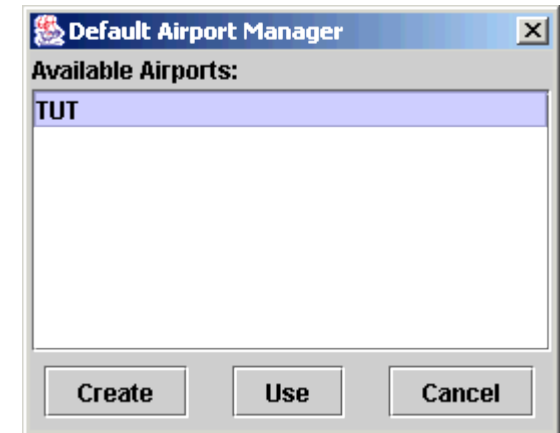
Enter **Tutorial Airport** for the description and press the **OK** button.



After creation of the TUT airport you should select it from the list as shown to the right.

Once TUT is selected you should press the **Use** button.

After pressing the Use button the Default Airport Manager will close and the TUT airport LOCID will be displayed on the Network Builder as shown below:



If you decide to close the Network Builder don't forget to press the Save button before exiting otherwise your newly created airport will not be saved.

Subsequent sessions will require you to pick a current airport before any Network Builder editing of airfield objects can be performed. In those sessions simply execute the **Edit/Set Current Airport** menu items, pick the airport you want to modify, and then press the **Use** button.

Taxiway Creation

Visual SIMMOD makes creation of airfield structures such as taxiways a simple task. Assuming you have a background QGF layer (previously created from a DXF file) the creation of taxiways involves tracing over the background layer using your mouse.

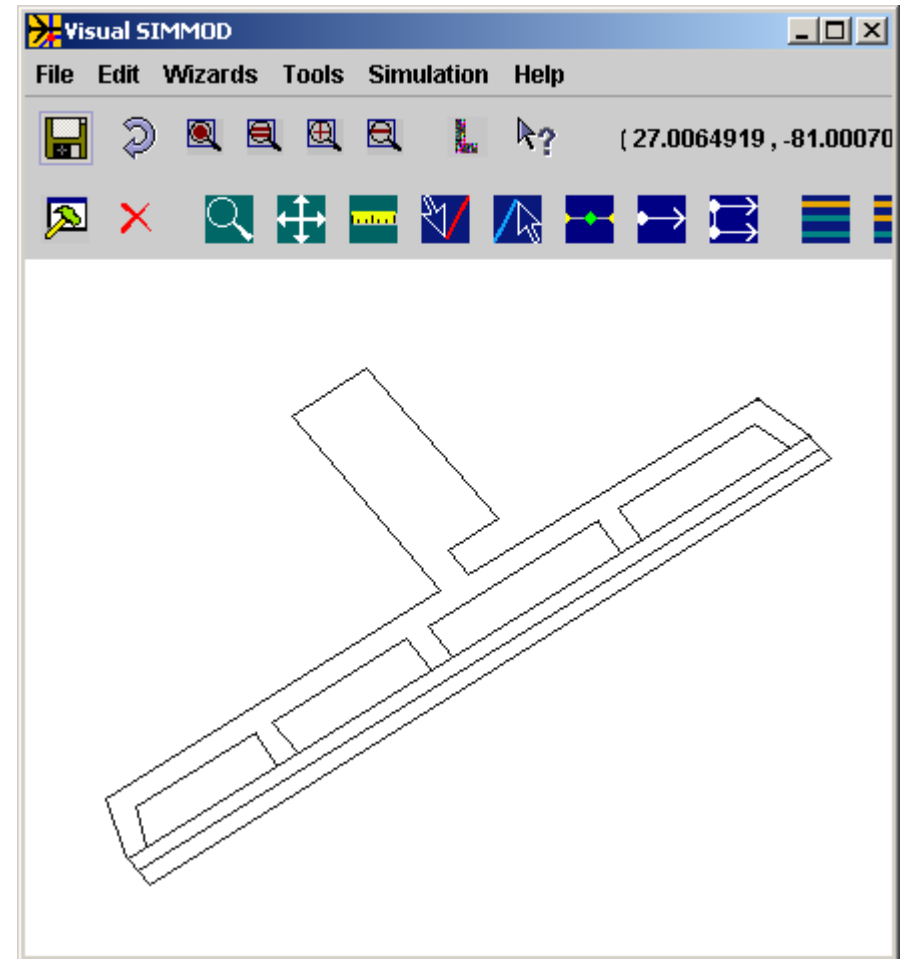
As you trace over the background layer Visual SIMMOD creates SIMMOD links and nodes with default attribute values for each. You are also given the opportunity to modify the attribute values as each link or node is created.

Before beginning the creation of taxiways you should zoom and pan the scene to something similar to the scene shown at the right:


If you do not remember how to zoom and pan please review the earlier sections in this tutorial.



Note that the window to the right is far smaller than normal usage dictates, we suggest you make the Network Builder window much larger. We had to make the window smaller than normal for this document.



Taxiway Creation (continued)

To add an airfield link press the  button.

The next step in taxiway creation is to trace over a section of taxiway with the mouse.

In the example shown the left mouse button was pressed on the taxiway in the upper right of the map graphics and then dragged to the first intersection.

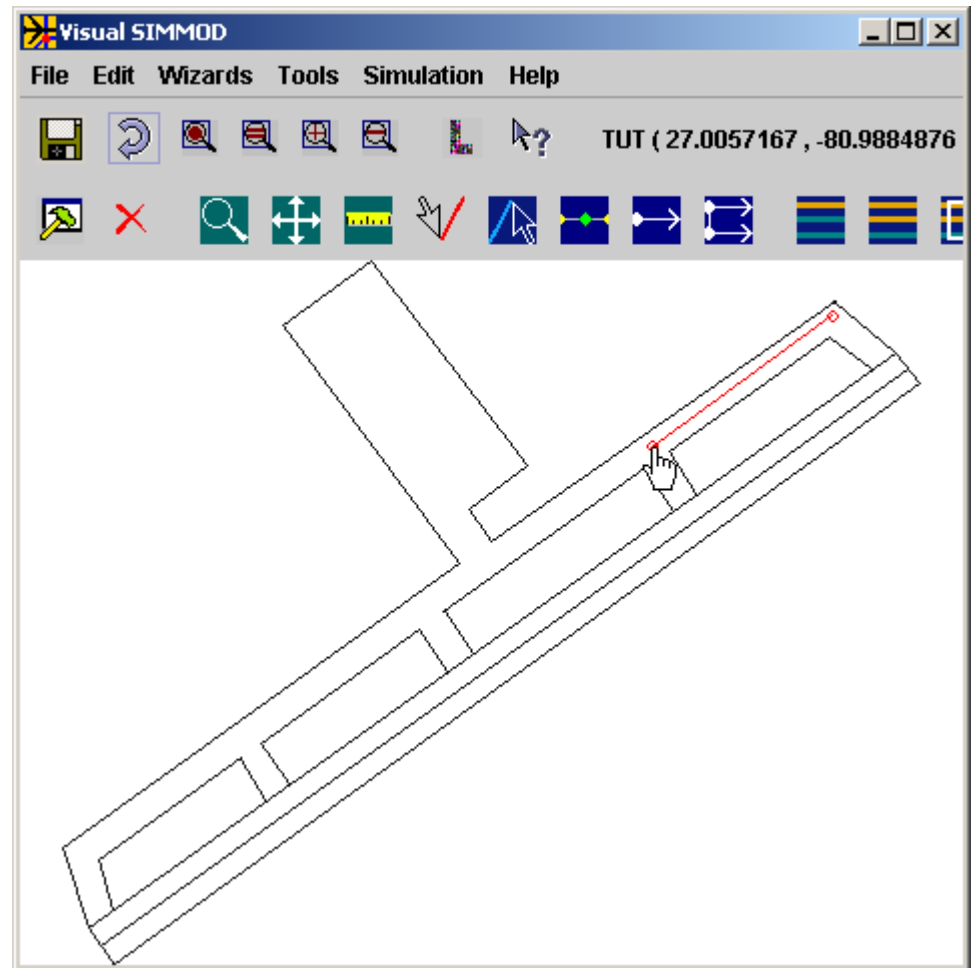
 Do NOT extend the link beyond the intersection.

The left mouse button was then released and a set of three new SIMMOD objects were created.

Two new airfield nodes were created, as depicted by circles (your colors may vary from those shown in this tutorial) on the map. In SIMMOD terminology these new airfield nodes are referred to as the initial and final nodes.

Also, a new airfield link was created. The airfield link extends from the initial node to the final node.

A window which allows expert users to modify the attributes of the link and nodes just created will also have opened. It is recommended that you make this window an icon for the remainder of the tutorial otherwise it will reopen each time a new link is created.

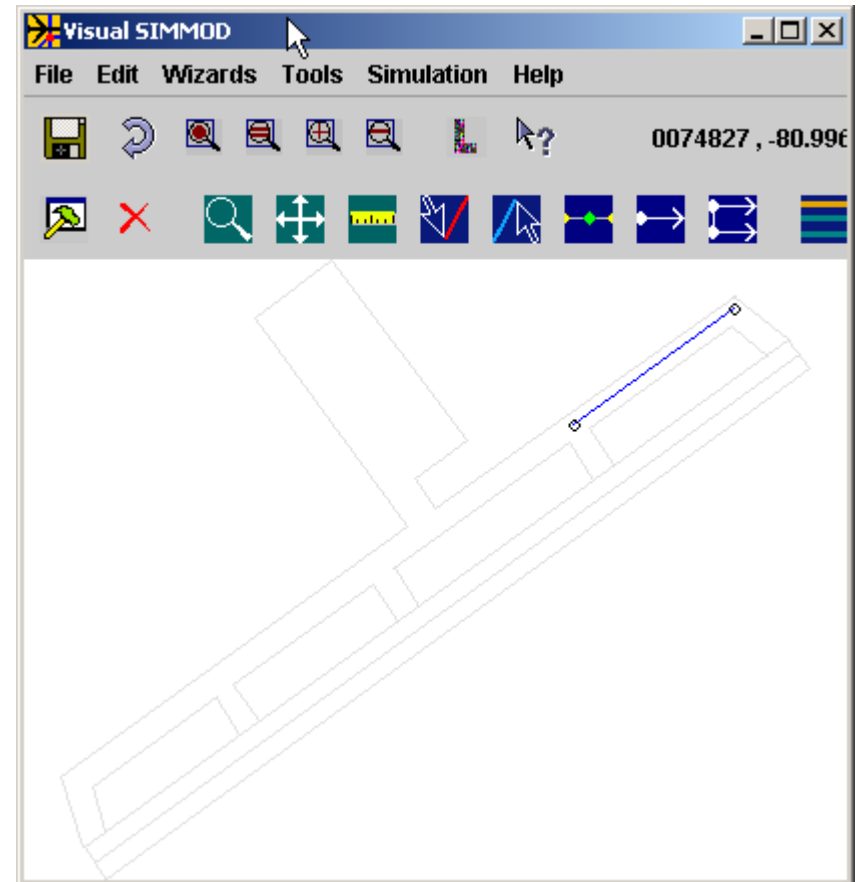
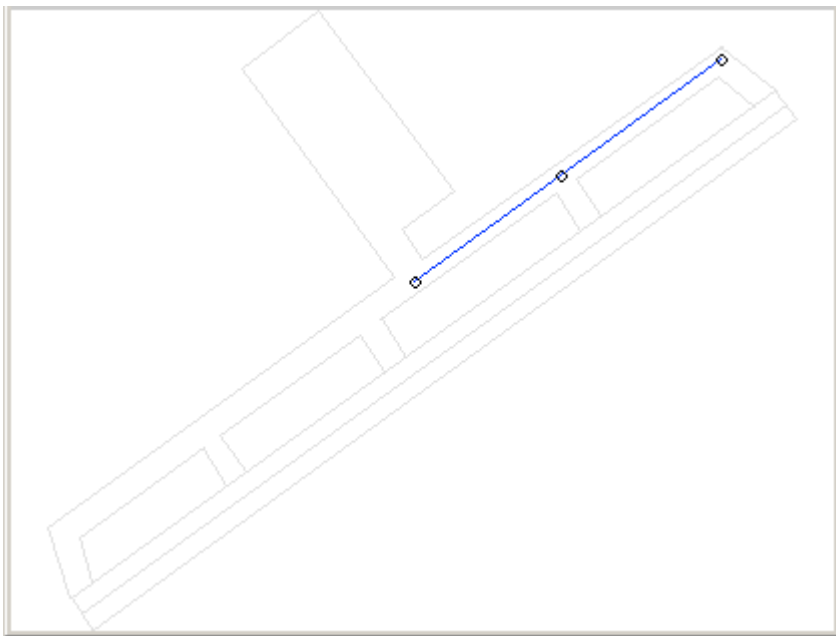


Taxiway Creation (continued)

- ! Do NOT attempt to create airfield links over the runway portion of the background layer at this time. Creation of runways is a specialized task that is very similar to creation of taxiways, but runways require special handling and those requirements will be covered later in the tutorial.

The final result of the add airfield link operation is shown to the right. The nodes are depicted in black and the new link is depicted in blue.

To add a second link position the mouse directly over the final node which was just created, press the left mouse key, drag the mouse to the target location and release the left mouse button.



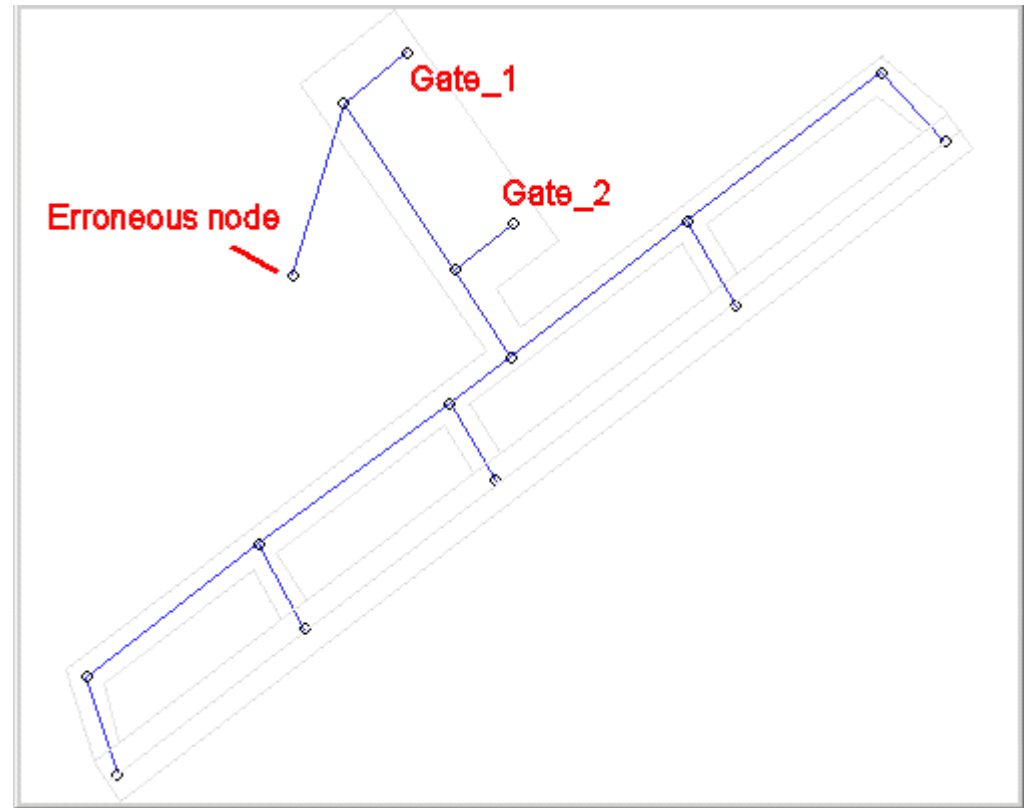
The result of the creation of a second link can be seen in the snapshot to the left. Notice that the link only extends from intersection to intersection.

Taxiway Creation (continued)

It is important to note that you should build links (and therefore place nodes) at locations on the airfield where aircraft may interact with each other.

Taxiway crossings, runway exits, runway crossings, staging areas, gates, and deicing areas are all locations on airfields that a SIMMOD link should end at and upon which a SIMMOD node should be positioned.

Node locations can also be described as distinct locations where aircraft may interact, such as one aircraft having to wait for another to cross in front.



You may now finish the creation of links and nodes at the TUT airport. We have finished the task and the snapshot above shows the results.



You may have noticed the link coming off the pad that does not end at any apparent airport location. We intentionally created this error to be able to demonstrate link and node editing techniques in the next section.




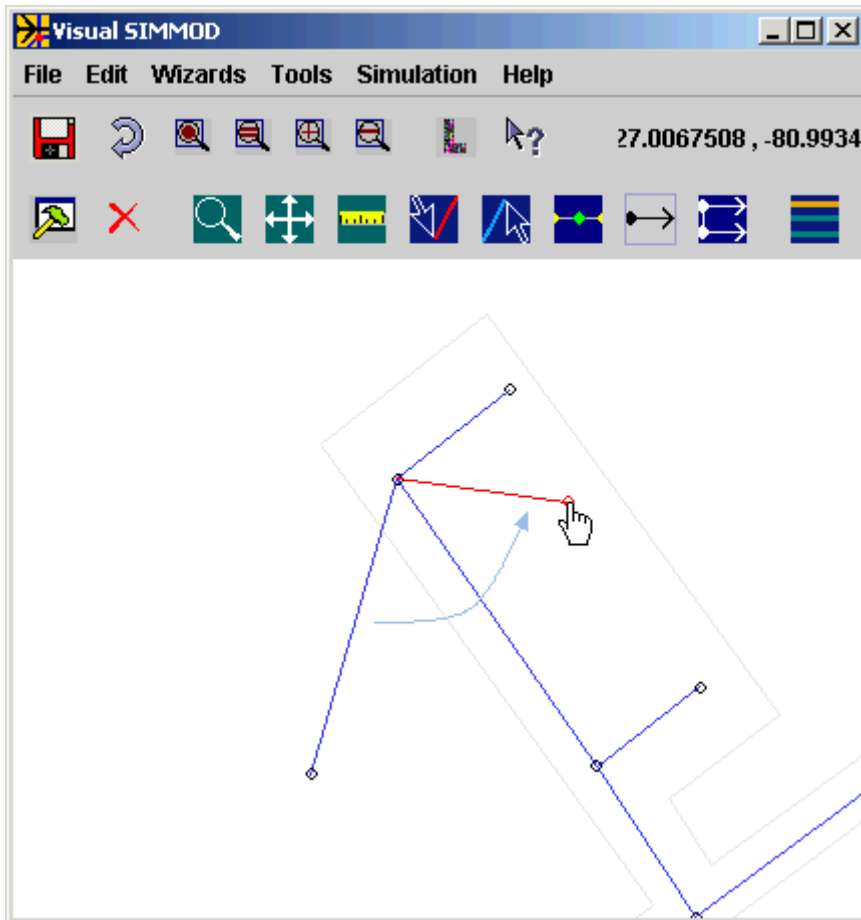
Also, note that no runway links have been created yet. The intersections and end-points for the runway have been designated, as shown. Runway link creation will occur in a later section of the tutorial.

Moving Nodes

Visual SIMMOD makes moving a node from one location to another easy. The procedure is described below:

In the previous example an intentional error was made whereby an erroneous link and node extended from the apron area to a non-existent airport location.

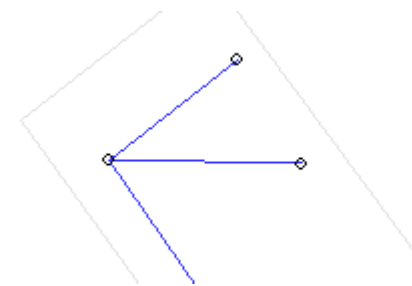
To move the node from the erroneous position to a correct position first press the  button.



Then press the left mouse button on the erroneous node, drag the node to the new location, and then release the mouse button.

The map graphics will update with the node in the correct location and the link extending to it properly.


The result of the move operation shown at the left is shown below:




Selecting Links and Nodes


A task that all users of Visual SIMMOD will repeatedly perform is the selection (highlighting) of links and nodes.

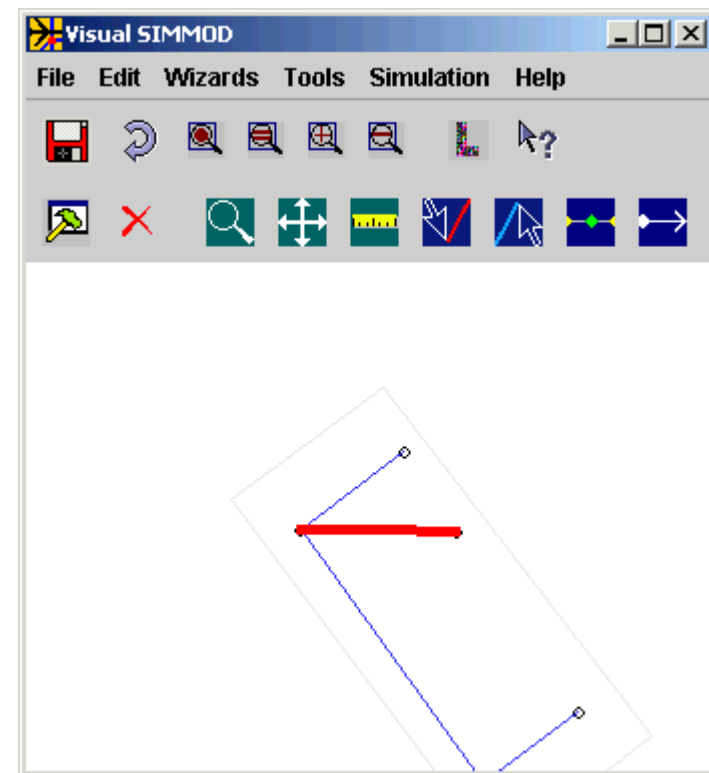
! It is important that you learn this task since much of the functionality of Visual SIMMOD depends upon your ability to designate upon which links or nodes you want a particular operation to be performed.

To select a link or node first press the  button. You can now select one link or node on the map graphics. Press the left mouse button on a link or a node. Assuming you clicked on a link it will change color (SELECTED OBJECTS color in the Set Attributes Tool) to represent the fact that it is selected and may be referenced via other functionality.


An example of a selected link (in red) is shown to the right:

Multiple links or nodes may be selected if you first select the  button. Pressing the  button allows a box to be drawn around links or nodes to select them.

To deselect (unhighlight) all items press the  button.



Editing Links and Nodes

To edit selected links or nodes press the  button. A Generic Table Editor will open for each category of link or node (airfield node, airspace node, airfield link, or airspace link).

Link and node attributes, such as capacity and direction of travel, may be edited now. It is beyond the scope of this tutorial to detail all of the link and node attributes.

Visual SIMMOD creates links and nodes with generic attributes that allow for the free flow of aircraft through the airfield and airspace. It is left up to the user in the study design to assign more restrictive attributes to the links and nodes.

All Generic Table Editor windows open inside the Workbench window. Any resizing of a Generic Table Editor window is remembered from one Visual SIMMOD session to another.

If a Generic Table Editor window is left open when the Workbench window is closed then the next time the Workbench window is opened the Generic Table Editor Window will still be open.

It is possible to have two Generic Table Editors open for the same table.


A more thorough review of the Generic Table Editor will be presented later in this tutorial.

You may close the Workbench window now.

Deleting Links and Nodes

The following describes the procedure for deleting a link or node.

 This description is for reference only, **do not** delete any links or nodes at this time.

To delete selected links or nodes press the  button on the Network Builder.

If a dependency exists for a link or a node, such as a link belongs to a runway, then a window will popup allowing you to review which links and nodes have a dependency.

You must eliminate all dependencies before you can delete a parent link or node.

Runway Creation

The runway creation process is similar to creation of taxiways except for one small but significant difference as required by the SIMMOD engine.

The SIMMOD engine requires runway links and nodes to be ordered in the runway's primary direction.



For runway links, aircraft traveling in the runway's primary direction must always move from the initial node to the final node. This is a SIMMOD simulation engine requirement.

For normal airfield links aircraft are permitted to move along the link from the initial node to the final node or from the final node to the initial node.

As noted earlier in this tutorial all links have an initial node and a final node. It is possible, and universally common, for two links to share the same node, i.e. they are connected links.

For a single shared node and two connected links the following situations arise:

- The first link may have the shared node as it's initial node and the second link may have the shared node as it's initial node. This is not valid for runways.
- The first link may have the shared node as it's initial node and the second link may have the shared node as it's final node. This is not valid for runways.
- The first link may have the shared node as it's final node and the second link may have the shared node as it's initial node. This is valid for runways.
- The first link may have the shared node as it's final node and the second link may have the shared node as it's final node. This is not valid for runways.

Runway Creation (continued)


For aircraft traveling in the opposite direction on a runway they must always move from the final node to the initial node. This is a SIMMOD engine requirement.

When creating airfield links Visual SIMMOD always makes the first node entered (where the mouse is initially pressed) the initial node of the link and the last node entered (where the mouse is released) the final node of the link.

Thus, to create a runway using Visual SIMMOD you should confirm the primary direction of the runway and create the airfield links accordingly.

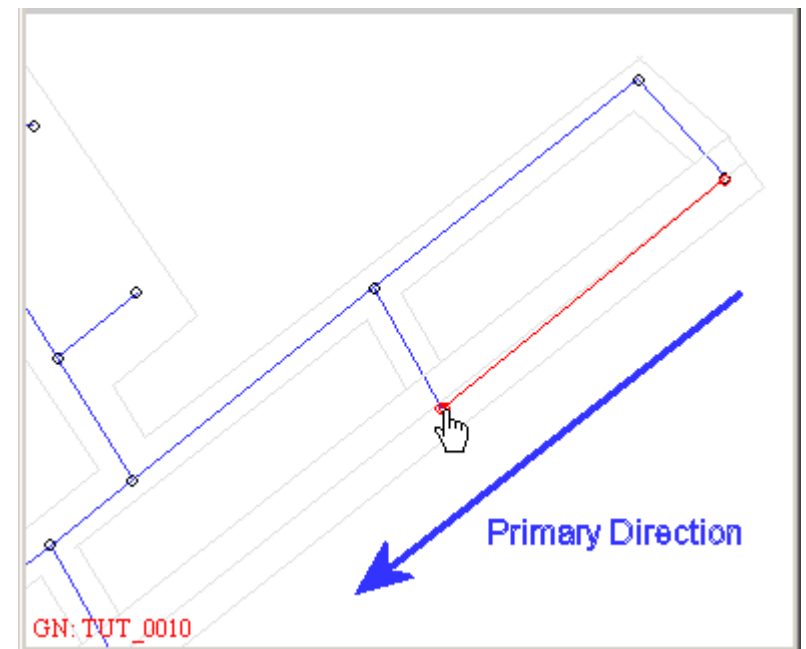
The snapshot to the right has been augmented with a large blue arrow and text tag to show the primary direction of travel.

It can be seen from the snapshot that the mouse was pressed on the initial node at the end of the runway and is about to be released on the second runway node.

You should duplicate the link as shown. Don't forget to press the  button first.

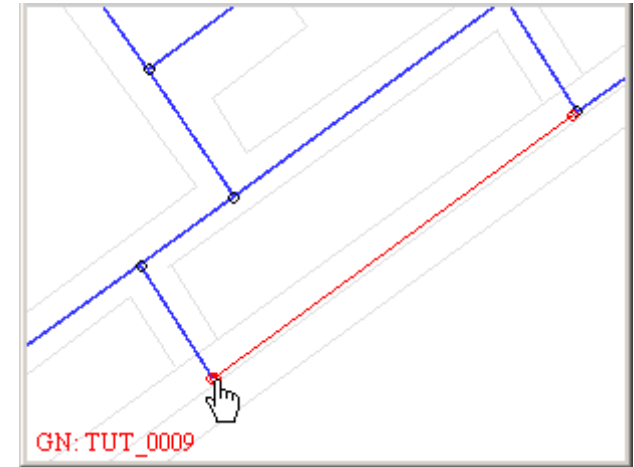
The newly created link is not a runway link yet, you will assign it to a runway in a later step. But, you can note that its initial node and final node will be shared with normal taxiway links.


The final node will also be shared with the next created runway link.

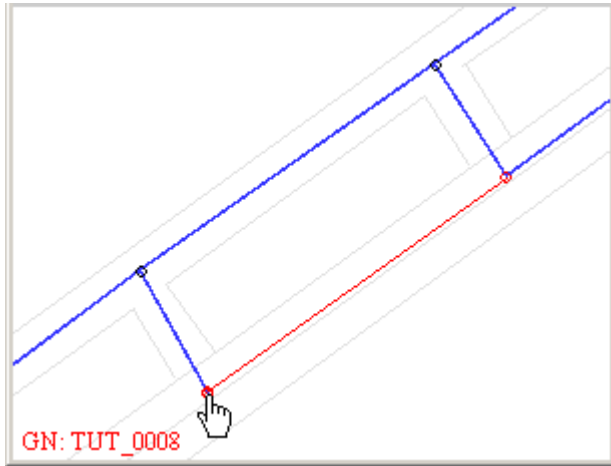


Runway Creation (continued)

The second runway link should start at the second runway node (TUT_0010) and extend to node the third runway node (TUT_0009). Create the link now.



 Your node names may not match those in this tutorial due to the link creation order. If the links and nodes are in the proper physical locations there shouldn't be any issues caused by differing link and node names.

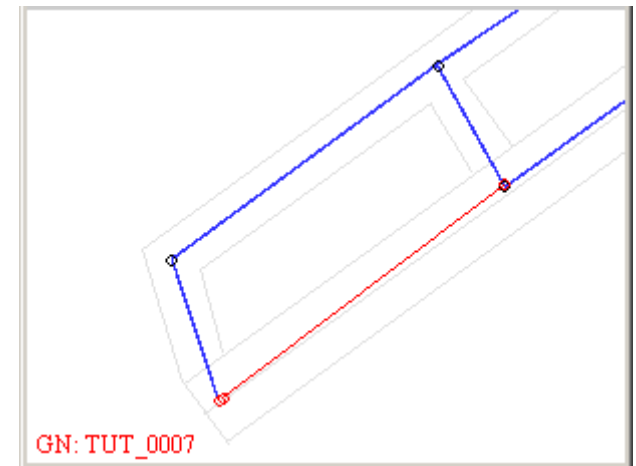


The third runway link should start at the third runway node (TUT_0009) and extend to the fourth runway node (TUT_0008). Create the link now.

The fourth runway link should start at the fourth runway node (TUT_0008) and extend to node the fifth runway node (TUT_0007). Create the link now.

After completion of the above the following links exist:

Link Identifier	Initial Node	Final Node
TUT_0011~TO~TUT_0010	TUT_0011	TUT_0010
TUT_0010~TO~TUT_0009	TUT_0010	TUT_0009
TUT_0009~TO~TUT_0008	TUT_0009	TUT_0008
TUT_0008~TO~TUT_0007	TUT_0008	TUT_0007



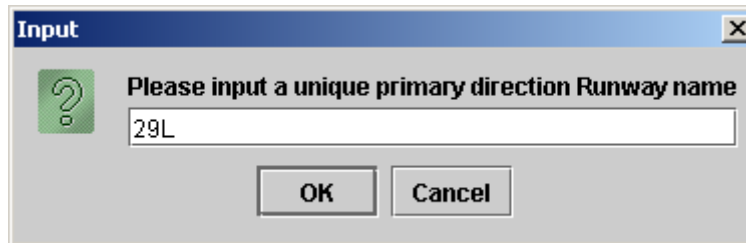
Runway Creation (continued)

The next step in runway creation is to open the Runway Editor tool from the Network Builder's main menu. To open the Runway Editor execute the *Tools//Runway Editor* menu tree now.

The Runway Editor with no data is shown to the right.

Create a new runway by pressing the  button.

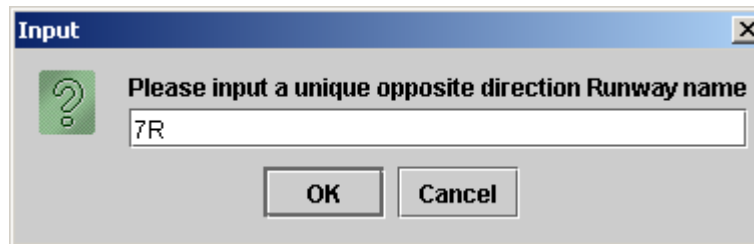
Type **29L** into the popup window as shown to assign the primary direction identifier for this runway.



An 'Input' dialog box with a question mark icon. The text reads: 'Please input a unique primary direction Runway name'. The text input field contains '29L'. There are 'OK' and 'Cancel' buttons at the bottom.

Press **OK**.

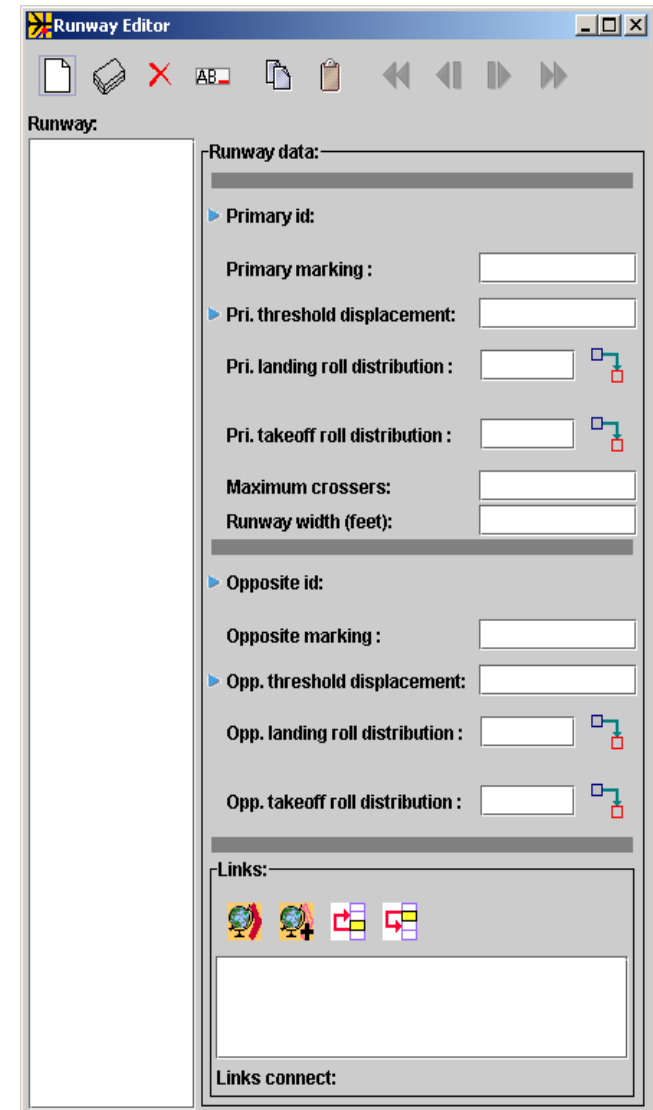
Type **7R** into the popup window as shown to designate the opposite direction identifier for this runway.



An 'Input' dialog box with a question mark icon. The text reads: 'Please input a unique opposite direction Runway name'. The text input field contains '7R'. There are 'OK' and 'Cancel' buttons at the bottom.

Press **OK**.

The Runway Editor should now show the 29L runway in its list of runways. It will have a symbol next to it indicating a new runway.



Runway Creation (continued)



Note that SIMMOD will provide inconsistent results when provided a runway identifier that evaluates to an integer number. Thus, if your runway identifier is simply 14 then you should prepend the airport's identifier to the runway identifier. Thus, if, in our example, we had a runway called 14 we would have used TUT_14.

Or rather, if the runway identifier is an integer number and you want to display only that number, you can provide this number in the primary or opposite direction marking field.

Visual SIMMOD can display the runway markings using an exclusive built-in font which matches exactly the FAA's required lettering dimensions for runways. This field's data can later be exported to the APECS Animator for display using the same font.

In our example runway you should type in the following for runway 29L/7R:

Primary marking: 29L

Pri. threshold displacement: 0

Maximum crossers: 4

Runway width (feet): 150

Opposite marking: 7R

Opp. threshold displacement: 0




Notice that after typing in the above values the icon to the left of the 29L changes to an edited symbol.

The last task remaining is to assign the four links previously created to the runway.

While keeping the Runway Editor open, temporarily switch windows to the Network Builder.

Runway Creation (continued)

To make sure no extraneous links are assigned to the runway press the  button on the Network Builder.


Next press the  button to indicate that you want to enable multiple selections.

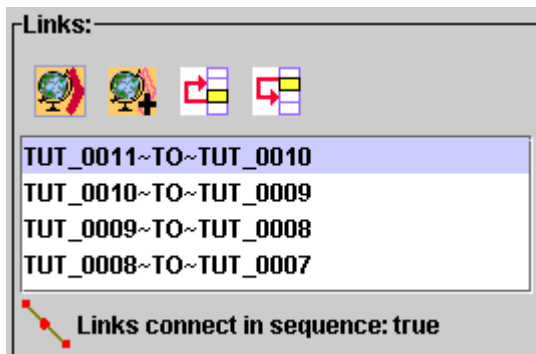
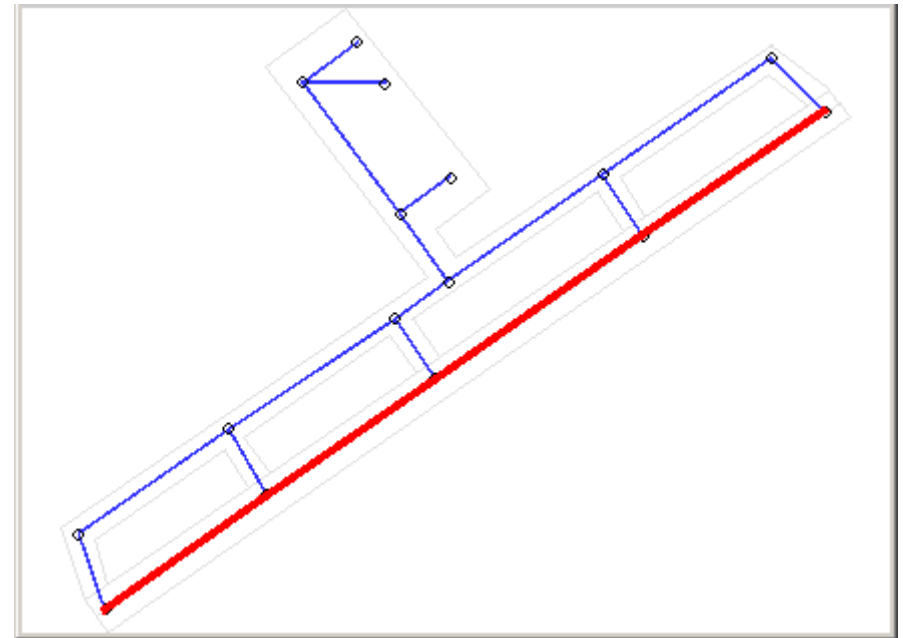


Move the mouse slowly over each of the four links created for the runway. Notice how the name of the link is displayed as the mouse travels over it. This feature lets you know when you can select a particular item. It also helps prevent the accidental selection of incorrect items.

Click the left mouse button on each of the four new runway links. Your map graphics should look similar to the snapshot shown at the right:

After highlighting the four links you are now ready to assign these four links to runway 29L (and 7R as well, by default).

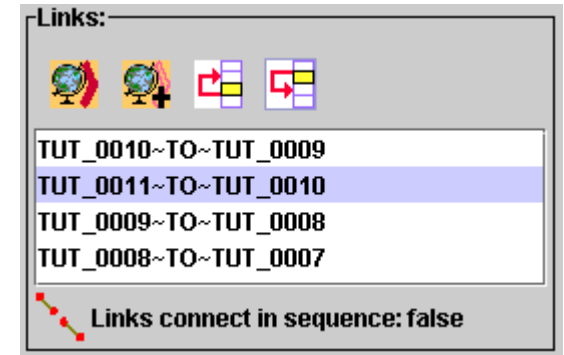
On the Runway Editor assign the four links by pressing the  button.





If the links are correctly built then the message “Links connect in sequence: true” will be displayed at the bottom of the Runway Editor.

Runway Creation (continued)


If the links are not correctly built then the message “Links connect in sequence: false” will be displayed at the bottom of the Runway Editor.






To correct the problem, first select the link you think is causing the problem in the Links: list and then you can press the  or the  button to try to correct the problem by rearranging the links. If the problem is corrected the message will change to say that the links do connect in sequence properly.

If you cannot rearrange the links to correct the problem then one or more of your links does not connect to the other links in the previously described initial node to final node (which should be an initial node on the next link as well) to final node, etc. sequence.

It is recommended that if you have the above issue, and cannot correct it by resequencing the links, that you delete the four runway links and start the runway creation process again.

If your links are sequenced properly then you need only press the  button to write your new runway to the database.

 Note that pressing the write to database button () does not save your data, you need to press the  button on the Network Builder to actually save your changes for use in a different session.

You may close the Runway Editor.

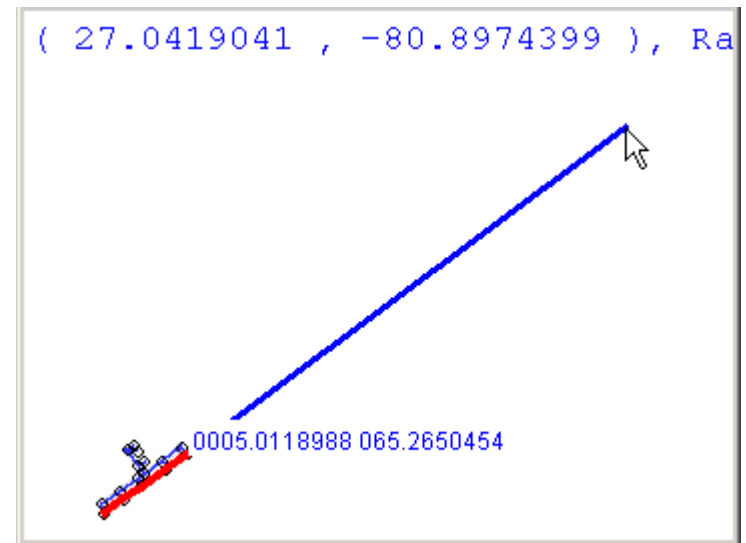
Route Creation



Airspace routes in SIMMOD are a sequence of links that connect to form a path. Aircraft fly from node to node via links along a route. Usually routes are either arrival routes (last node is physically located at an airport) or departure routes (initial node is physically located at an airport).

A link may belong to multiple routes, which also means that routes can either converge at a particular node or they may diverge at a particular node.

Creation of routes using Visual SIMMOD is very similar to the creation of runways. The requirement of aircraft must always travel from initial to final nodes on runways is also a requirement for routes. Thus, the approach taken to building routes is similar.


To begin the route creation process zoom out so that you can view portions of the map that are at least five nautical miles in distance from the TUT airport.



You can measure distances on the map graphics by pressing the  button and then dragging the mouse between your points of interest. Press the  button a second time to remove the markers that remain after a measurement. Measurements less than three nautical miles are shown in feet, otherwise in nautical miles.

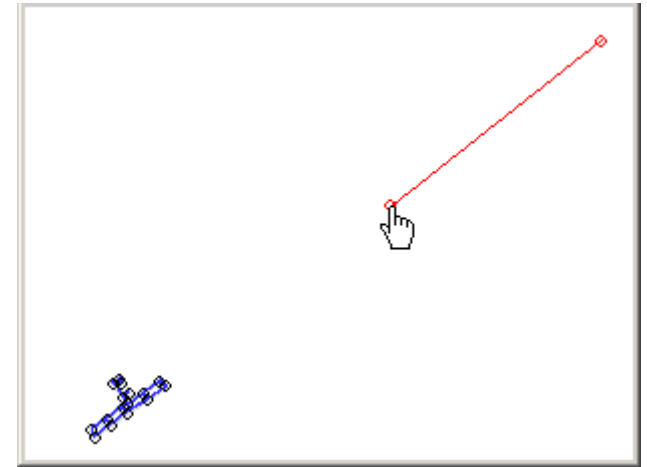
The first route to create is an arrival route. The goal is to build a route that approaches the TUT airport from about four miles out. An airspace link of approximately two miles long will be built first. The initial node on this airspace link will be about four miles from the TUT airport and the final node will be about two miles from TUT airport.

Route Creation (continued)

To add an airspace link press the  button. Then click and hold down the left mouse button at a location about four miles away from runway 29L's touchdown location. Drag the left mouse button closer to runway 29L and release it when the mouse is approximately halfway to the runway.

An example of adding an airspace link is shown to the right:

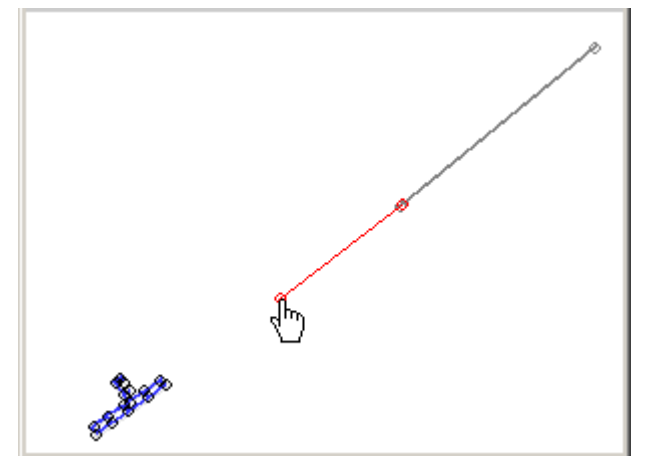
When the mouse is released the links should look similar to the snapshot below (your colors may vary):



The next step is to add a second link.

The important thing to remember when adding this second link is that you should position the mouse over the final node of the first link. When the mouse is properly positioned the name of the final node will appear at the lower left of the map graphics.

When positioned correctly, press the left mouse button, drag the mouse halfway to the runway again and release the mouse button:



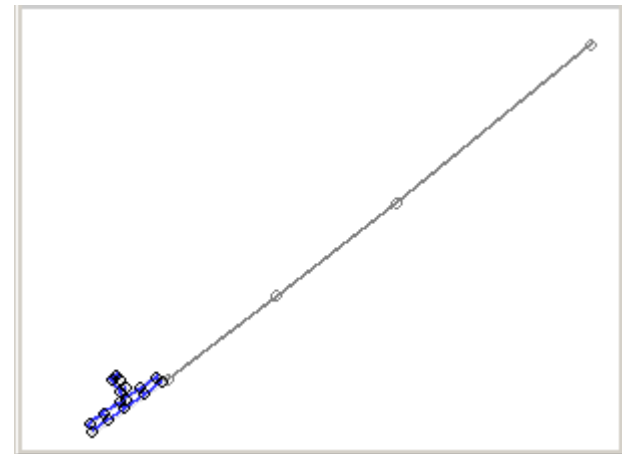
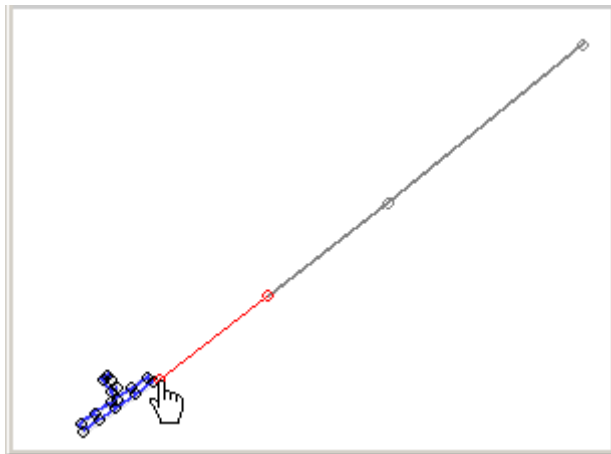
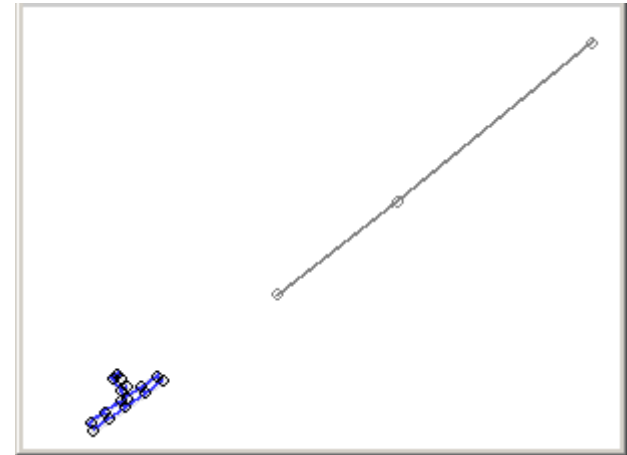
Route Creation (continued)

The result of adding a second link is shown to the right:

Adding a third link which ends the arrival route at the airport is the obvious next step.

You may add the third link in the same manner as the second link was created, making sure that the initial node of the third link is the final node of the second link.

It is not necessary to attempt to position the third link's final node directly on top of an airfield node. An airspace link cannot have an airfield node as its final node. The final airspace node of the third link can sit on top of an airfield node or it can be slightly askew since it can be adjusted quite easily at a later time. The process of adding the third link and the result is shown below:




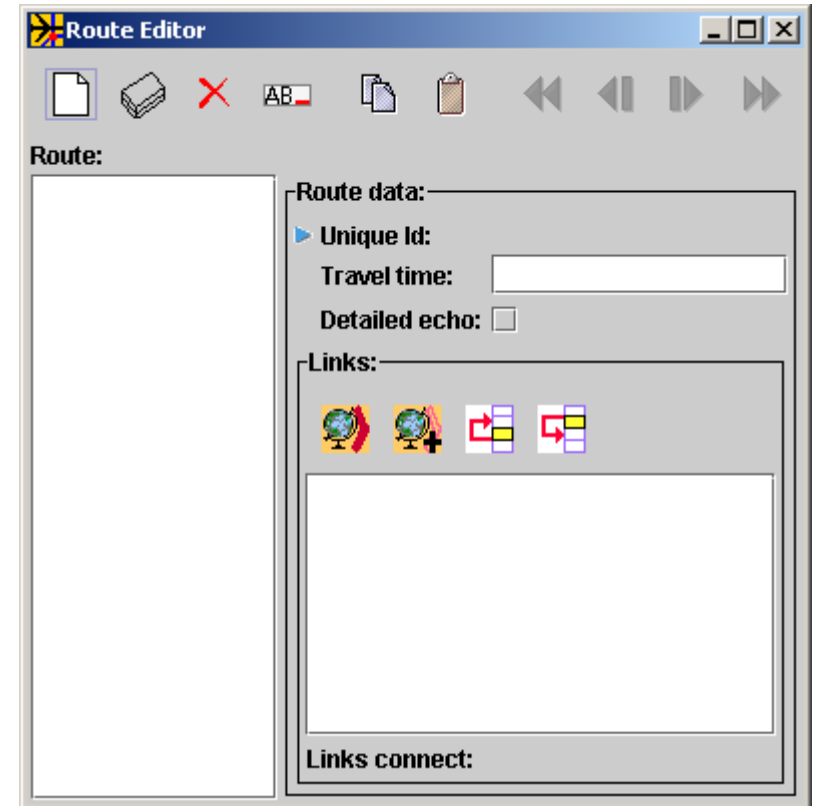
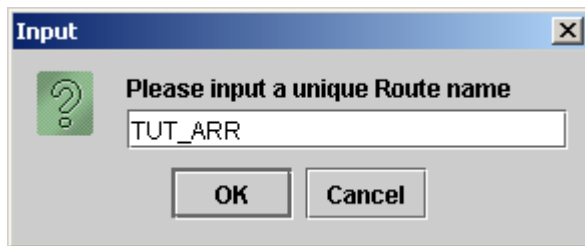
Route Creation (continued)

After creation of the three airspace links as shown the next step in route creation is to open the Route Editor.

You may open the Route Editor by executing the **Tools//Route Editor** menu command in the Network Builder.

The Route Editor is shown to the right:

To create a new route press the  button. A popup window will appear that will allow you to type in the name of a new unique route identifier. Type in **TUT_ARR** as the route identifier. Then press the **OK** button.




Upon pressing the OK button the new route will be placed in the list of known routes as a new route. You are now ready to finish the definition of the route.

It is not necessary to provide a travel time for the TUT_ARR route since travel times in this context are related to bank turnarounds and this tutorial will not cover banking issues. If you wish to receive a detailed report concerning a route's usage then you may optionally check the "Detailed echo" checkbox.

Route Creation (continued)

The last task remaining is to assign the newly created airspace links to the TUT_ARR route.

To make sure that no extraneous links are assigned to the route first press the  button on the Network Builder.


Next press the  button to indicate that you want to enable multiple selections.

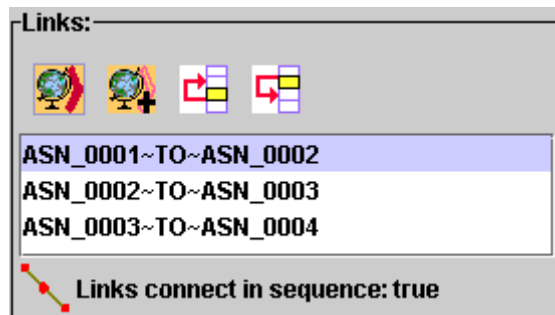
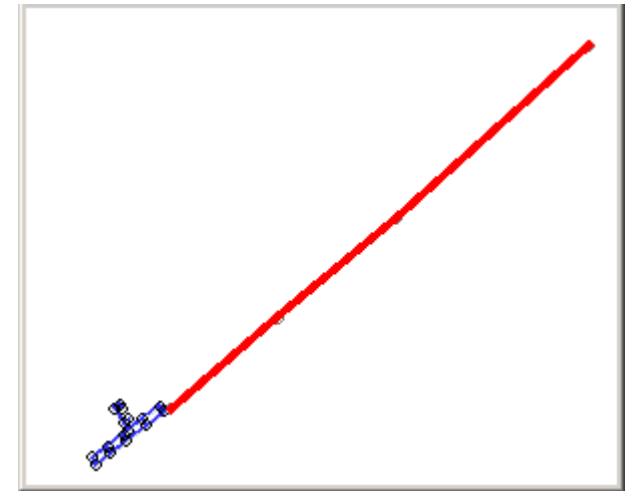


Move the mouse slowly over each of the three links created for the route. Notice how the name of the link is displayed as the mouse travels over it. This feature lets you know when you can select a particular item. It also helps prevent the accidental selection of incorrect items.

Click the left mouse button on each of the three new route links. Your map graphics should look similar to the snapshot at the right:



After highlighting the three links you are now ready to assign these links to route TUT_ARR.

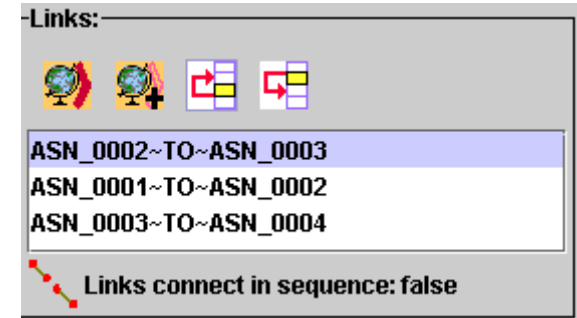
Assign the three links by pressing the  button on the Route Editor. If the links are correctly built then the message “Links connect in sequence: true” will be displayed at the bottom of the Route Editor.



Route Creation (continued)


If the links are not correctly built then the message “Links connect in sequence: false” will be displayed at the bottom of the Route Editor.




To correct the problem, first select the link you think is causing the problem in the Links: list and then you can press the  or the  button to try to correct the problem by rearranging the links. If the problem is corrected the message will change to say that the links do connect in sequence properly.



If you cannot rearrange the links to correct the problem then one or more of your links does not connect to the other links in the previously described initial node to final node (which should be an initial node on the next link as well) to final node, etc. sequence.


It is recommended that if you have the above issue, and cannot correct it by resequencing the links, that you delete the three links and start the route creation process again.

If your links are sequenced properly then you need only press the  button to write your new route to the database.

 Note that pressing the write to database button () does not save your data, you need to press the  button on the Network Builder to actually save your changes for use in a different session.

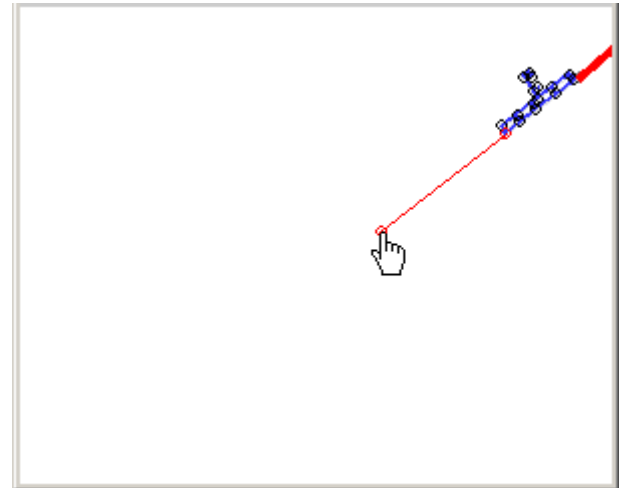
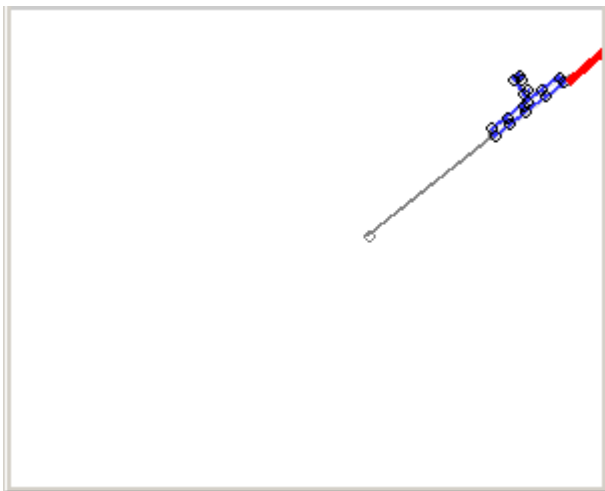
Route Creation (continued)

The next route to create is a departure route. The goal is to build a route that departs the TUT airport up to about four miles out. An airspace link of approximately two miles long will be built first. The initial node on this airspace link will be at the departure end of runway 29L and the final node will be about two miles from TUT airport.

To add an airspace link press the  Button on the Network Builder. Then click and hold down the left mouse button at a location over runway 29L's final node. Drag the left mouse button in a direction away from runway 29L and release it when the mouse is approximately two miles from the runway.

An example of adding an airspace link is shown to the right:

When the mouse is released the link should look similar to the snapshot below (your colors may vary):

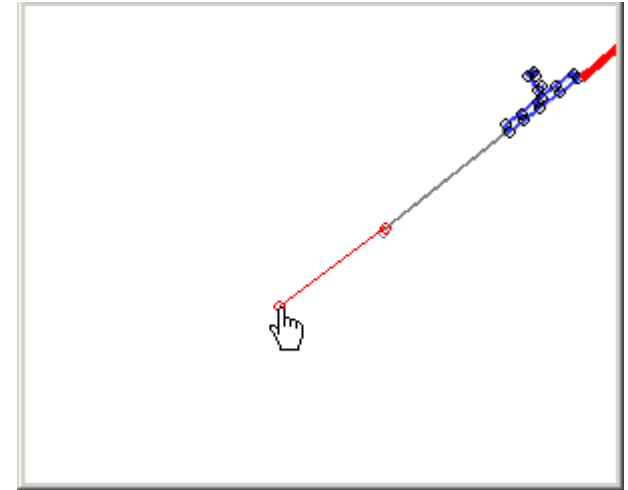


Route Creation (continued)

The next step is to add a second link.

The important thing to remember when adding this second link is that you should position the mouse over the final node of the first departure link. When the mouse is properly positioned the name of the final node will appear at the lower left of the map graphics.

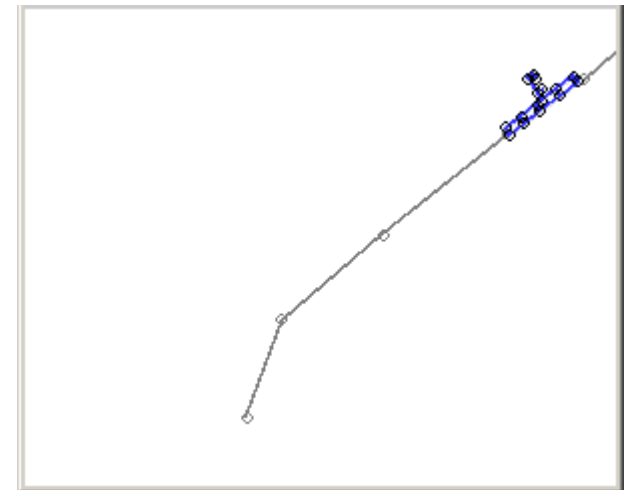
When positioned correctly, press the left mouse button, drag the mouse about one nautical mile and release the mouse button:



The final drawing step is to add a third link.

Once again, it is important to remember when adding this third link that you should position the mouse over the final node of the last departure link added.

When positioned correctly, press the left mouse button, drag the mouse about one nautical mile and release the mouse button:




The result of adding the two additional links is shown to the right:

Route Creation (continued)

Just as in the creation of the arrival route previously, after creation of the three airspace links the next step in route creation is to open the Route Editor.

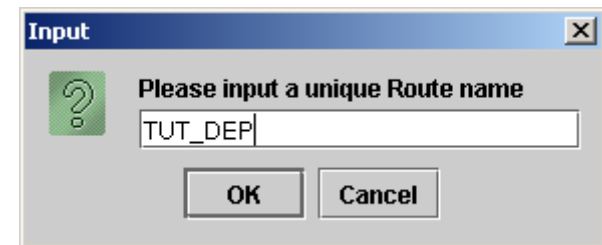
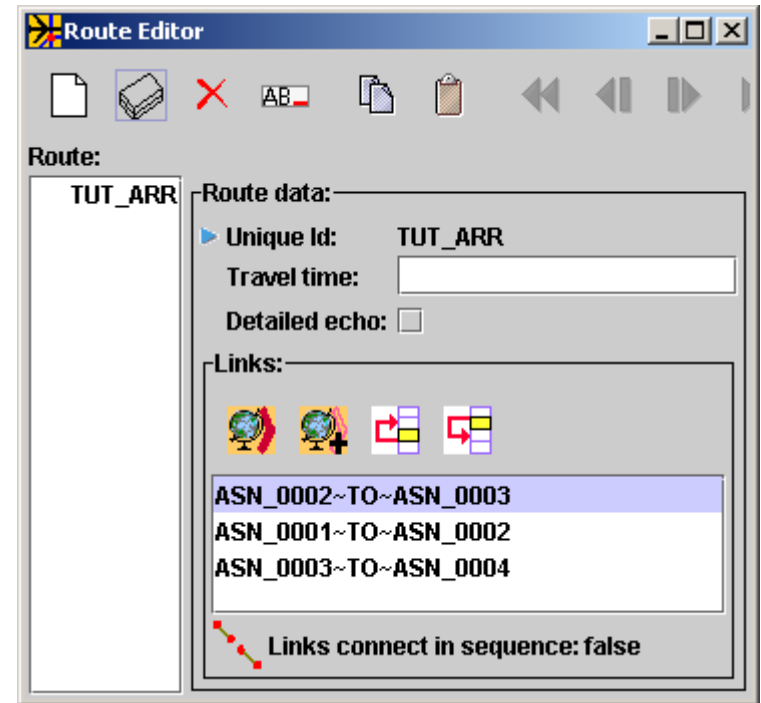
If the Route Editor is no longer open you may open it by executing the **Tools//Route Editor** menu command in the Network Builder.

The Route Editor will now show the previously created TUT_ARR route.

To create a new route press the  button. A popup window will appear that will allow you to type in the name of a new unique route identifier. Type in **TUT_DEP** as the route identifier. Then press the **OK** button.

Upon pressing the OK button the new route will be placed in the list of known routes as a new route. You are now ready to finish the definition of the route.

It is not necessary to provide a travel time for the TUT_DEP route since travel times in this context are related to banking turnarounds and this tutorial will not cover banking issues. If you wish to receive a detailed report concerning a route's usage then you may optionally check the “Detailed echo” checkbox.



Route Creation (continued)

It should be noted that route TUT_DEP has been assigned the same links which belong to route TUT_ARR. Although this is not what is desired it is not an error condition. Visual SIMMOD just tries to make the creation of routes that have similar features to the currently selected route (in this case it was route TUT_ARR) easier by cloning the currently selected route (excepting the identifier of course).

The next step is to assign the departure links just created to route TUT_DEP.

To make sure that no extraneous links are assigned to the route first press the  button.

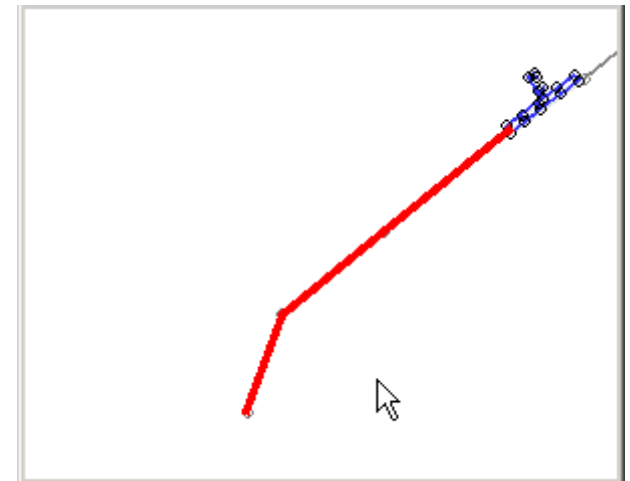
Next press the  button to indicate that you want to enable multiple selections.




Move the mouse slowly over each of the three links created for the route. Notice how the name of the link is displayed as the mouse travels over it. This feature lets you know when you can select a particular item. It also helps prevent the accidental selection of incorrect items.

Click the left mouse button on each of the three new route links.
Your map graphics should look similar to the snapshot at the right:



After highlighting the three links you are now ready to assign these links to route TUT_DEP.



Route Creation (continued)

Assign the three links by pressing the  button on the Route Editor. If the links are correctly built then the message “Links connect in sequence: true” will be displayed at the bottom of the Route Editor.




If the links are not correctly built then the message “Links connect in sequence: false” will be displayed at the bottom of the Route Editor.

To correct the problem, first select the link you think is causing the problem in the Links: list and then you can press the  or the  button to try to correct the problem by rearranging the links. If the problem is corrected the message will change to say that the links do connect in sequence properly.

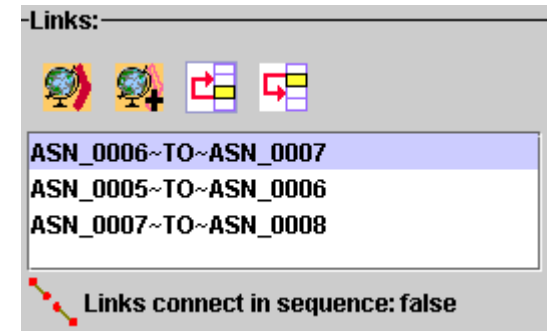
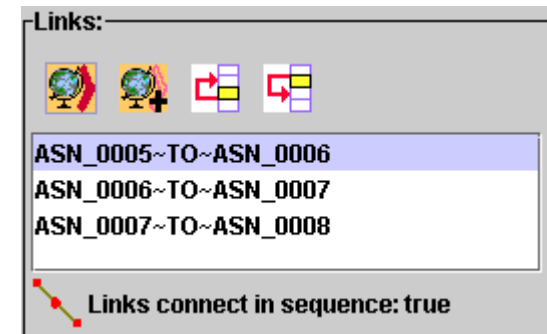
If you cannot rearrange the links to correct the problem then one or more of your links does not connect to the other links in the previously described initial node to final node (which should be an initial node on the next link as well) to final node, etc. sequence.

It is recommended that if you have the above issue, and cannot correct it by resequencing the links, that you delete the three links and start the route creation process again.

If your links are sequenced properly then press the  button to write your new route to the database.

 Note that pressing the write to database button () does not save your data, you need to press the  button on the Network Builder to actually save your changes for use in a different session.

You may close the Route Editor now.




Interface Node Creation

Interface nodes are the airspace nodes which are physically located at, or over, runway ends. They represent the final nodes of arrival routes and the initial nodes of departure routes.

When combined with SIMMOD procedures data, the interface nodes tell SIMMOD where the airspace network connects with the airfield network. The procedures data will be discussed later in this tutorial. Interface nodes are a requirement before procedures data can be properly defined.

Assigning interface nodes in Visual SIMMOD involves opening the Airports Editor tool, selecting the nodes from the Network Builder's map graphics and then assigning them via the Airports Editor tool.

To begin selecting interface nodes first press the  button on the Network Builder's toolbar to remove all previous selections.

Open the Airports Editor and begin assigning the nodes. To open the Airports Editor execute the *[Tools//Airports Editor](#)* menu tree.

Next click on the previously created TUT airport in the list on the left.


While keeping the Airports Editor open change your window to the Network Builder.

You may need to zoom in to the airport area in a manner similar to the snapshot on the following page.


Interface Node Creation (continued)

Next press the  button on the Network Builder's toolbar to enable the selection of multiple items.

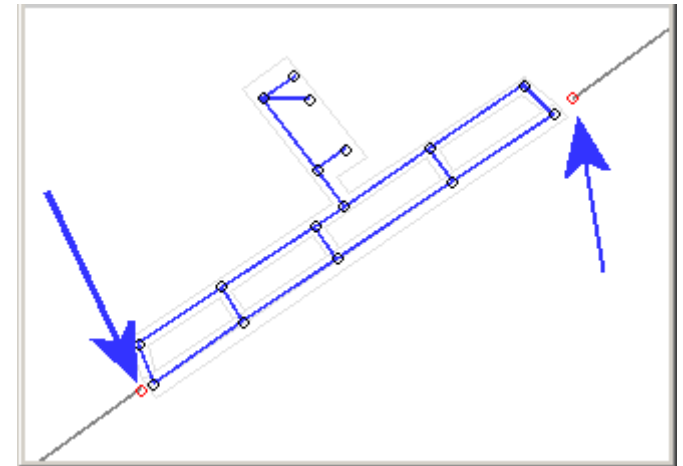
Click the left mouse button on the last node of the previously defined arrival route to select it.

Then click the left mouse button on the initial node of the departure route which was previously defined. Be careful to not select any airfield nodes. If you do press the  button and select again.

The selected nodes are shown in red in the snapshot at the right:

To assign the selected nodes to the TUT airport press the  button on the Airport Editor. The two nodes will be added to the spreadsheet at the lower right of the tool.

The Node Gap, Separation Increase, and Increase Types fields are optional and will be left blank in this tutorial. We recommend you review the SIMMOD AIRPORTS card flatfile documentation to determine if you will need to provide this data in your study.



Press the  button to write the modified TUT airport data (with two added interface nodes) to the database.

You may close the Airport Editor at this time.

Generic Table Editor/Airline Creation

Before gates or flights can be properly constructed in Visual SIMMOD it is necessary to create airlines.

To create airlines you should use the Generic Table Editor. The Generic Table Editor is a tool designed to allow editing of user-accessible database tables within the Visual SIMMOD system.

Based upon a spreadsheet design, users add, edit, and delete rows of database tables using the Generic Table Editor. Sophisticated queries can be made, find and replace operations performed, sorting of columns, and copy and paste tasks can be made as well.

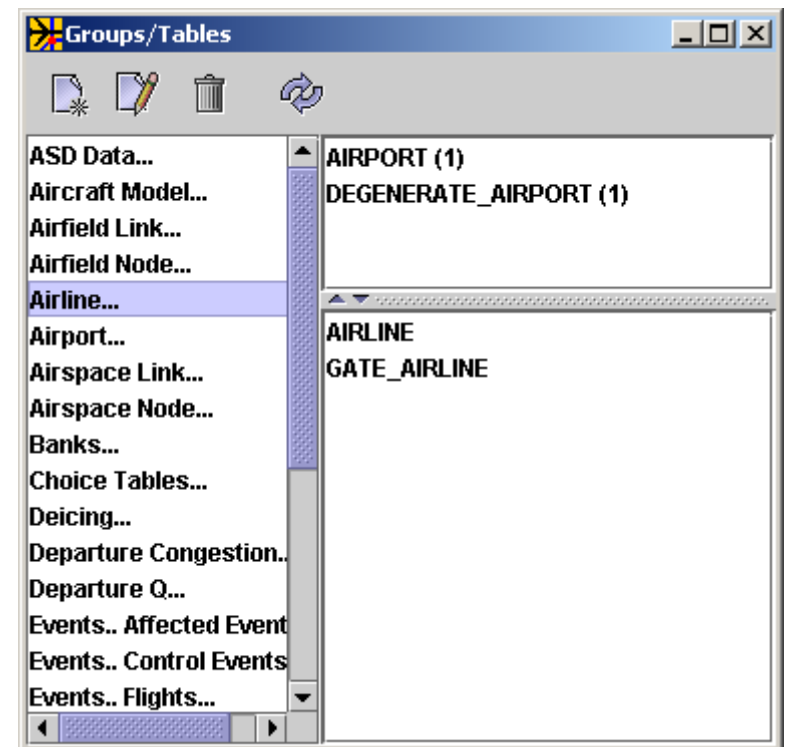
To edit the AIRLINES table in Visual SIMMOD you should first execute the *Edit/Edit Tables* menu item. The Groups/Tables Tool will open.

The Groups/Tables Tool, as shown to the right, organizes the tables into sets of tables which are related in some manner.

The Groups (related table sets) are listed in the left column. In this case choose the “Airline...” grouping.


Upon choosing the Airline... grouping the tables belonging to that grouping will be displayed in the lower right list. Choose the AIRLINE table from the list to begin editing it.

The upper right list displays previously accessed tables (favorites) sorted by how many times they have been accessed. This tool allows you to manually set the number of access times per table for greater flexibility.



Generic Table Editor/Airline Creation (continued)


Clicking on the AIRLINE table opens a Workbench window with the AIRLINE Generic Table Editor within it.




To add a new record press the  button.

 ALN_ID	 DESCRIPTION	 ALN_TYPE
------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------

Click the mouse on the ALN_ID column and begin typing in your data. In our example please use DZZ, Dazaz, and COMMERCIAL in the ALN_ID, DESCRIPTION, and ALN_TYPE fields respectively.

 ALN_ID	 DESCRIPTION	 ALN_TYPE
DZZ	Dazaz	COMMERCIAL

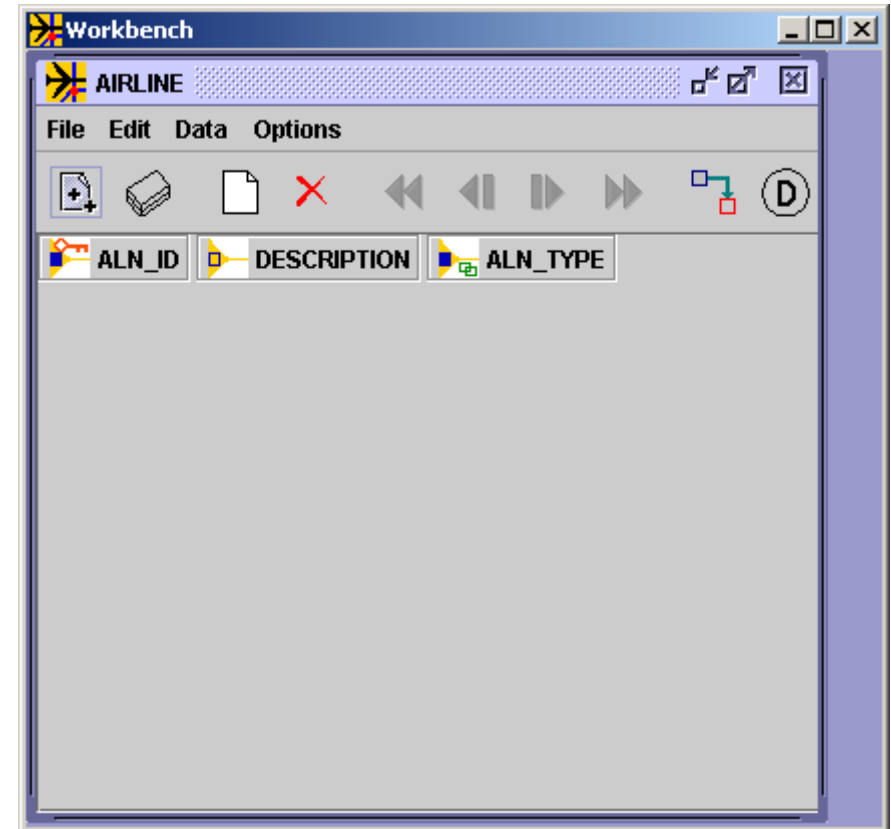
To add a second record press the  button again.

 ALN_ID	 DESCRIPTION	 ALN_TYPE
DZZ	Dazaz	COMMERCIAL
FLH	FlyHigh	COMMERCIAL


When the second record is added it contains the same contents as the last record selected. This behavior is a time-saving feature. But, you must modify the values of the key field(s) since duplication of key fields by two or more records is not allowed.


A key field is depicted by a red key in the icon of the field's header.

In our example we have modified the ALN_ID and DESCRIPTION fields to FLH and FlyHigh respectively.





Generic Table Editor/Airline Creation (continued)


Press the  button to write the new, edited, and deleted AIRLINES data to the database.


You may press the  button to review the full specifications of the Generic Table Editor. Some of the more important capabilities are summarized below.

 Retrieve all of the records currently stored in the database.

 Mark selected records for deletion. You may undo these marked records via the *Edit//Restore Selected Records* menu item.

 Undo All / Undo One / Redo One / Redo All buttons.

 Link to a field's possible parent values. When you press this button you will be presented with possible values that this field may contain from the field's parent table.

 Retrieve/set default values. Some fields may have a default value. You may press this button to retrieve the default value or to set the default value for the field.

Edit//Cut - Perform a cut from the spreadsheet. You may paste this data to a third-party spreadsheet.

Edit//Copy - Perform a copy from the spreadsheet. You may paste this data to a third-party spreadsheet.

Edit//Paste - Perform a paste from other Generic Table Editor windows or from a third-party spreadsheet.

Data//Sort - Sort the spreadsheet. You may also click on a column's header to sort just that column.

Data//Query – Open a window which allow you to make sophisticated queries on the database.

You may close the Groups/Tables window, the Generic Table Editor window, and the Workbench window.

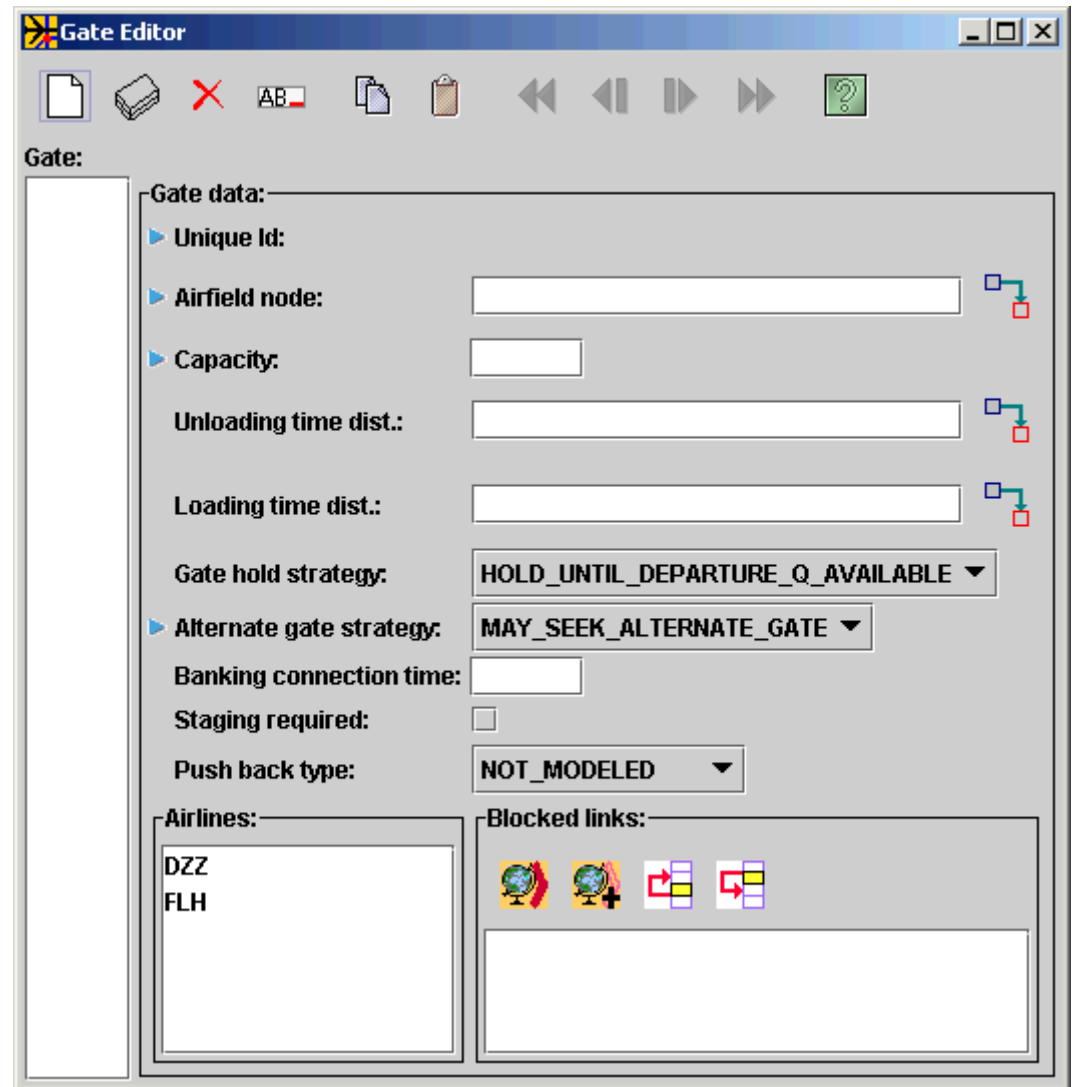
Gate Creation

The creation of gates within Visual SIMMOD is slightly more complex than the creation of runways and routes because there are more attributes associated with gates than with runways or routes.

Most of the gates' attributes are optional though and this tutorial, for the sake of brevity, will not cover the optional attributes.

In SIMMOD a gate exists at a specified airfield node. It has a capacity, airlines that can park at it, links which are blocked when it is occupied, and both loading and unloading time distributions associated with it.

To begin the gate creation process first open the Gates Editor via the **Tools//Gate Editor** menu item. An empty Gate Editor is shown to the right.



Create the first gate by pressing the  button.

Supply the name **DZZ_GATE** and press the **OK** button.


Gate Creation (continued)

Next, go to the Network Builder and zoom to the gate pad area of the map graphics.

Deselect all items on the map by pressing the  button.

Now enable single airfield node selections by pressing the  button.

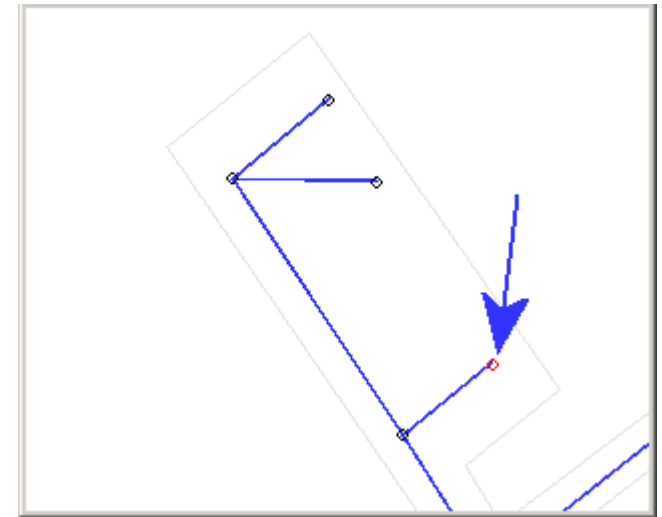
Select the airfield node on the pad closest to the runway by clicking on it once with the left mouse button. The node after having been selected is shown highlighted in red in the snapshot to the right:

On the Gate Editor now press the  button to the right of the Airfield Node field to open a window which will allow you to pick the highlighted node from a list. Pick the node you highlighted on the map graphics and press the **Ok** button. You have now assigned the DZZ_GATE to that node.

Change the required capacity field to indicate that the gate can accommodate up to 100 aircraft.

The required Alternate Gate Strategy field allows you to decide whether aircraft which want to use the gate, and find a capacity exceeded condition, must wait for the gate to become available or whether they can immediately seek an alternate gate. Since this tutorial will have just one DZZ gate select MUST_WAIT for this gate attribute.


Select DZZ from the Airlines list to assign which airlines may use this gate.



Gate Creation (continued)

- ! At least one blocked link must be supplied for each gate created. A blocked link is a link which is blocked when a departing aircraft is traversing the link.
- ! SIMMOD requires the initial blocked (first in the list) link must be the first link off of the node which is assigned to the gate. The sequence of all of the other links is not important.

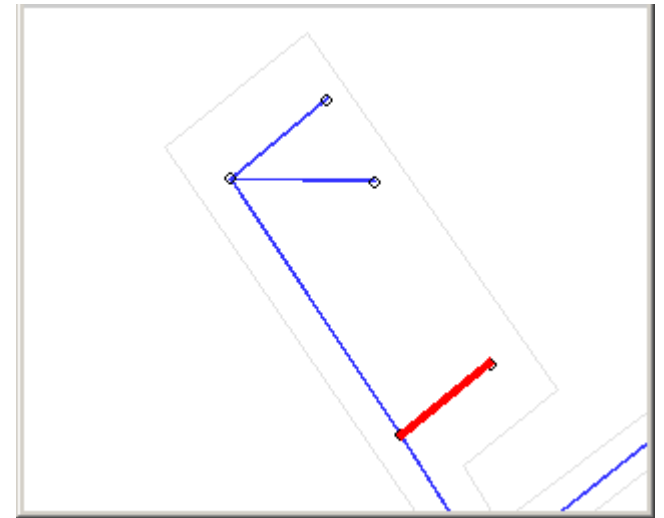
While keeping the Gate Editor open switch to the Network Builder.


To assign a blocked link press the  button to clear all selections.

Next zoom in to the pad area on the Network Builder's map graphics.


Now enable single airfield link selections by pressing the  button.

Select the link attached to the DZZ_GATE's airfield node by clicking on it with the left mouse button. The result of this operation is shown to the right:



Switch to the Gate Editor and press the  button on the Blocked Links list to assign the link to the gate.

You have now created the DZZ_GATE which can be used by the airline DZZ. The next step in this tutorial is to create a gate for the airline FLH.

Begin the gate creation process again by pressing the  button.

Supply the name **FLH_GATE** and press the OK button.

A screenshot of an 'Input' dialog box. The title bar says 'Input'. The main text says 'Please input a unique Gate name'. Below this is a text input field containing 'FLH_GATE'. At the bottom are 'OK' and 'Cancel' buttons.

Input

Please input a unique Gate name

FLH_GATE

OK Cancel

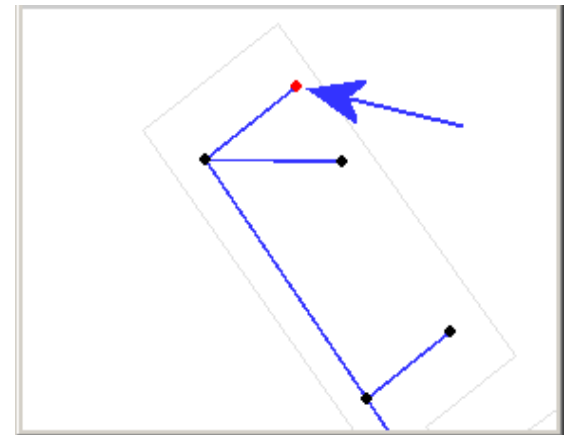
Gate Creation (continued)


Next, switch to the Network Builder and zoom to the gate pad area of the map graphics.

Deselect all items on the map by pressing the  button.

Now enable single airfield node selections by pressing the  button.

Select the airfield node on the gate pad furthest from the runway by clicking on it once with the left mouse button. The selected node is shown highlighted in red in the snapshot to the right:



On the Gate Editor now press the  button to the right of the Airfield Node field to open a window which will allow you to pick the highlighted node from a list. Pick the node you highlighted on the map graphics and press the **Ok** button. You have now assigned the FLH_GATE to that node.

Change the required capacity field to indicate that the gate can accommodate up to 100 aircraft.

The required Alternate Gate Strategy field allows you to decide whether aircraft which want to use the gate and find a capacity exceeded condition must wait for the gate to become available or whether they can immediately seek an alternate gate. Since this tutorial will have just one FLH gate select MUST_WAIT for this gate attribute.

Select FLH from the Airlines list to assign which airlines may use this gate.

Deselect DZZ since we do not want DZZ aircraft parking at FLH's gate..

Gate Creation (continued)

- ! At least one blocked link must be supplied for each gate created. A blocked link is a link which is blocked when a departing aircraft is traversing the link.
- ! SIMMOD requires the initial blocked (first in the list) link must be the first link off of the node which is assigned to the the gate. The sequence of all of the other links is not important.

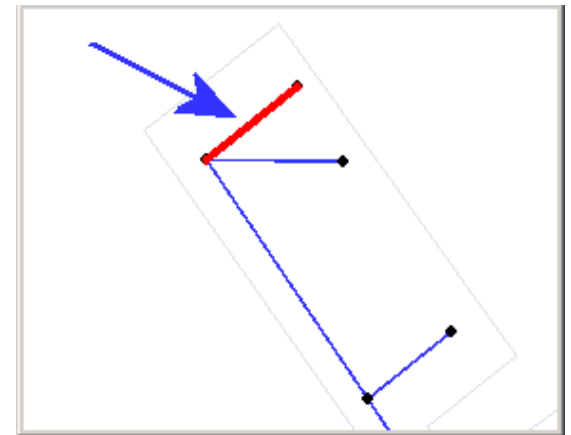
While keeping the Gate Editor open switch to the Network Builder.

To assign a blocked link press the  button to clear all selections first.

Next zoom in to the pad area on the Network Builder's map graphics.


Now enable single airfield link selections by pressing the  button.

Select the link attached to the FLH_GATE's airfield node by clicking on it with the left mouse button. The result of this operation is shown to the right:



Press the  button on the Blocked Links list to assign the link to the gate.

You have now created the FLH_GATE which can be used by the airline FLH.

Press the  button to write your Gates to the database.

You may close the Gates Editor.

Departure Queue Creation

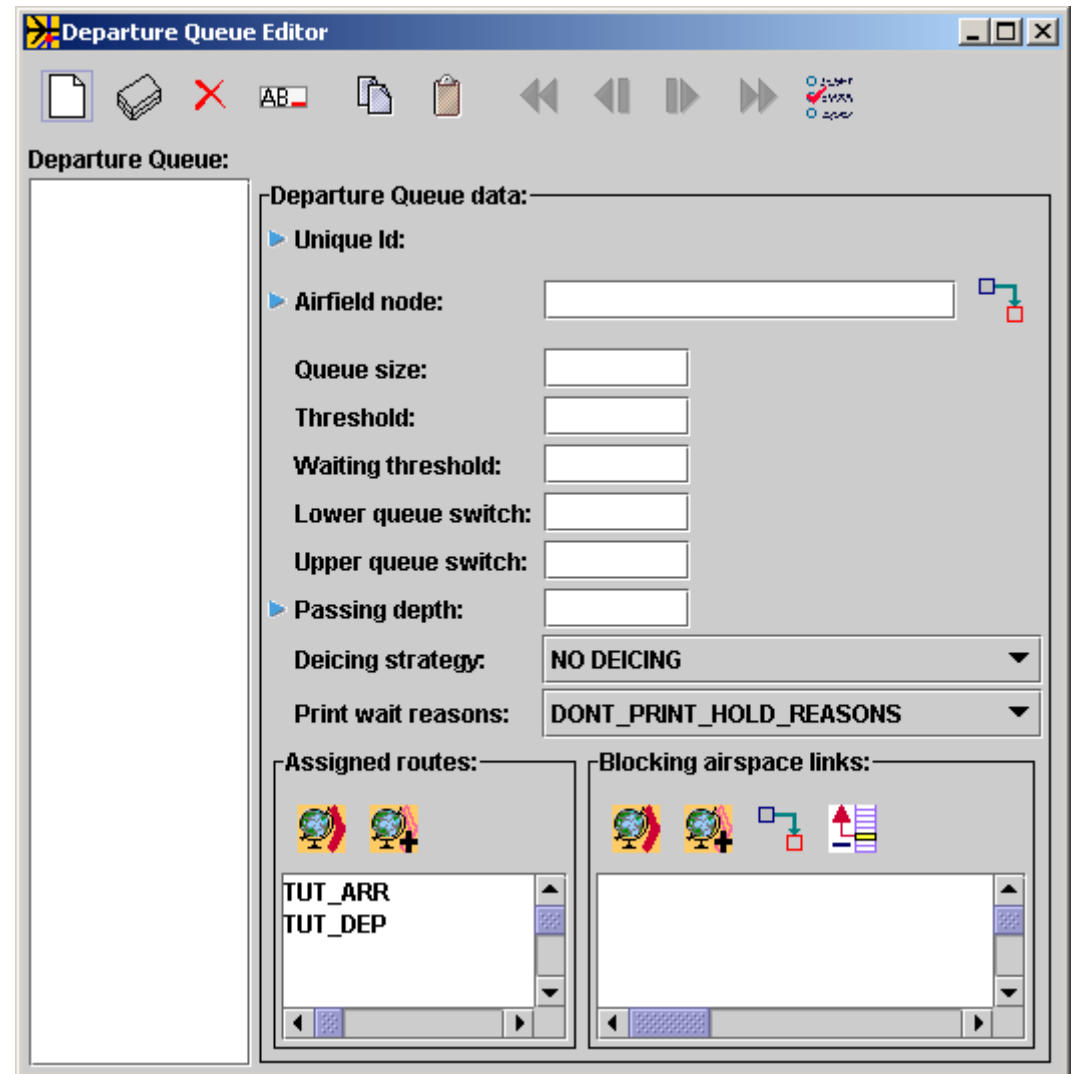
Departure queues in SIMMOD are airfield nodes where aircraft wait to enter a runway for takeoff. Multiple departure queues may be located at the same airfield node.

! It is possible for departure queues to be located on a runway node (normally the initial node of the runway) or on a node which is one or more links off the runway.

If the departure queue is located on a node off the runway then SIMMOD requires that you supply a taxipath (covered later in this tutorial) which provides an optimal path to the runway.

Departure queues have a capacity and may allow passing.

SIMMOD also makes provisions for passing depth, maximum waiting times, and defining airspace links which can block departure queues.

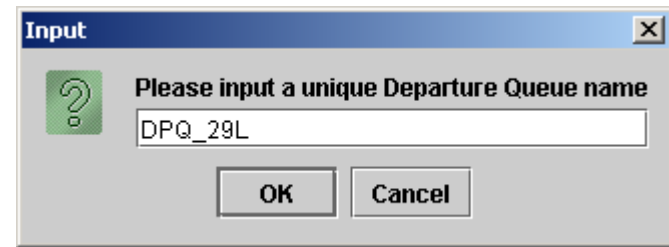


To begin the departure queue creation process first open the Departure Queue Editor via the **Tools//Departure Queue Editor** menu item on the Network Builder. An empty Departure Queue Editor is shown above.

Departure Queue Creation (continued)

Create the departure queue by pressing the  button.

Supply the name DPQ_29L and press the **OK** button.

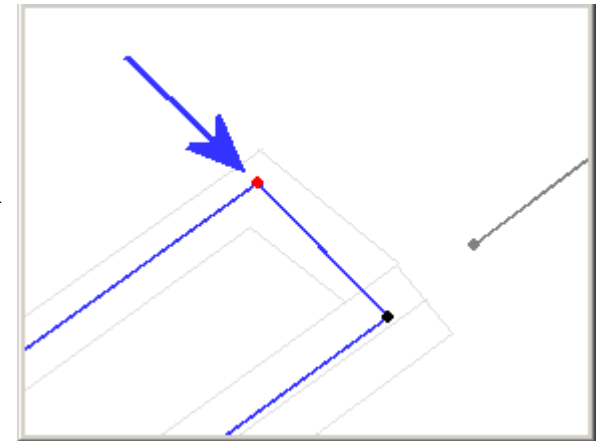



Next, go to the Network Builder and zoom to the initial end of runway 29L on the map graphics.

Deselect all items on the map by pressing the  button.

Now enable single airfield node selections by pressing the  button.

Select the airfield node one link off the runway's arrival end by clicking on it once with the left mouse button. The node after having been selected is shown highlighted in red in the snapshot to the right:



On the Departure Queue Editor now press the  button to the right of the Airfield Node field to open a window which will allow you to pick the highlighted node from a list. Pick the node you highlighted on the map graphics and press the **Ok** button. You have now assigned the DPQ_29L queue to that node.

Change the Queue Size field to indicate that the queue can accommodate up to 3 aircraft.


The Passing depth field determines how many aircraft in the queue a particular aircraft may pass if the procedures dictate that that particular aircraft could takeoff without any conflicts in separations.



If the passing depth value is left as zero then no passing may occur in the queue.

Departure Queue Creation (continued)


Double click TUT_DEP from the Routes list to assign which departure routes utilize this queue.

You may also highlight any link(s) on the departure route and then press the  button to assign the route to this newly created queue.

In this tutorial there will be no blocking airspace links therefore it is not necessary to assign any.

As noted earlier, if a departure is not located on a runway node then SIMMOD requires the user to supply a taxipath which will provide an optimal path to the runway node where the runway roll should begin.

The DPQ_29L queue requires just such a taxipath and in the next section of the tutorial the taxipath will be created.

You may press the  button to save your new departure queue to the database.

You may now close the Departure Queue Editor.

Taxipath Creation

Taxipaths in SIMMOD provide a means for the user to “help” the simulation determine the path that an aircraft should take when taxiing from one location on the airfield to another location.

On a more technical level, taxipaths are one or more airfield links which have zero cost assigned to them when SIMMOD is evaluating whether to use them in a potential optimal taxipath.

When determining the airfield path, an aircraft should take, SIMMOD automatically computes an optimal path between two locations (nodes) to minimize the travel time. This generalization normally works quite well since aircraft are normally instructed in real life to take the shortest path between two points.

However, for Departure Queues, we want to specify the path that should always be taken. We also may need to do this in some other real life special cases, where the shortest path is not necessarily what would actually be assigned to the aircraft to traverse. To model these special cases it is necessary to provide a taxipath which effectively manipulates the optimal taxipath calculation such that the physically shortest path is not chosen and the special case airfield links are chosen.



Taxipaths need not be consecutive airfield links, or rather they may not be connected at all. They also need not specify an entire path from one location to another, they merely need to include enough links to “guide” the SIMMOD optimal taxipath algorithm to choose the taxipath links rather than the optimal taxipath links.

Visual SIMMOD provides a tool called the Taxipath Editor which can be used to quickly construct taxipaths for a SIMMOD study.

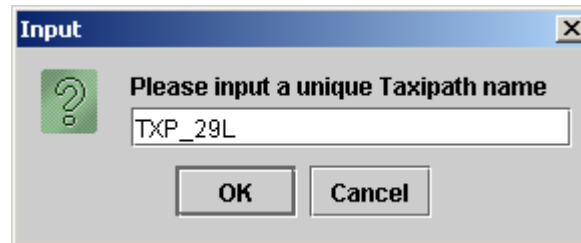
As noted earlier in the discussion of Departure Queues a taxipath between a departure queue and a runway is necessary when the departure queue does not share a node with the runway. The following example shows how to create this required taxipath. Other taxipaths can be created in the same manner.

Taxipath Creation (continued)

To define a taxipath begin by executing the *Tools//Taxipath Editor* menu item on the Network Builder. An empty Taxipath Editor is shown to the right.

Create a new taxipath by pressing the  button.

Supply the name **TXP_29L** and press the **OK** button.



Next, go to the Network Builder and zoom to the initial end of runway 29L on the map graphics.

Deselect all items on the map by pressing the  button.

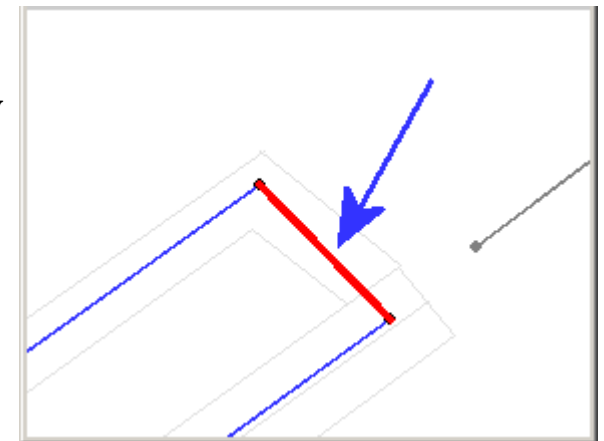
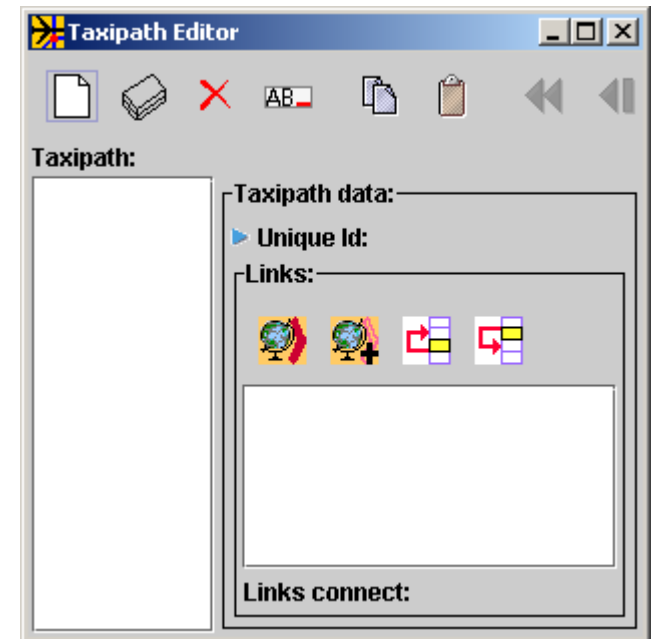
Now enable single airfield link selections by pressing the  button.

Select the airfield link between the departure queue and the runway end by clicking on it once with the left mouse button. The link after having been selected is shown highlighted in red in the snapshot to the right:

On the Taxipath Editor press the  button to assign the link to the TXP_29L taxipath.

Press the  to write the new taxipath to the database.

You may now close the Taxipath Editor.



Procedures Creation

Procedures in SIMMOD define the timing of aircraft operations on related runways.

More specifically, procedures inputs determine for a particular set of related runways separation distances and times for arrival/arrival, arrival/departure, departure/arrival, and departure/departure operations.

SIMMOD procedures are quite complex and require information such as which runways mutually affect each other during operations, the departure queues involved, aircraft groups involved, distance separations between related procedures, and time separations between related procedures.

Procedures are a **required** SIMMOD input.

In this tutorial only one runway is being developed therefore the procedures will define the time between subsequent flight operations on the 29L runway.

An example would be a departure flight following another departure flight. The procedures data which will be developed will dictate how long in seconds that a following departure on 29L must wait after a leading departure has taken off.

The procedures data will also tell SIMMOD how far in nautical miles that a leading departure on 29L must have flown before a following departure on 29L is allowed to depart.

The above separation examples will also be contingent on the aircraft grouping to which the aircraft belongs. Visual SIMMOD predefines a set of aircraft and groups of aircraft which have similar performance characteristics so that you can develop your studies faster.

Procedures Creation (continued)

To begin procedures development execute the **Tools//Procedures Editor** menu item. An empty Procedures Editor is shown to the right:

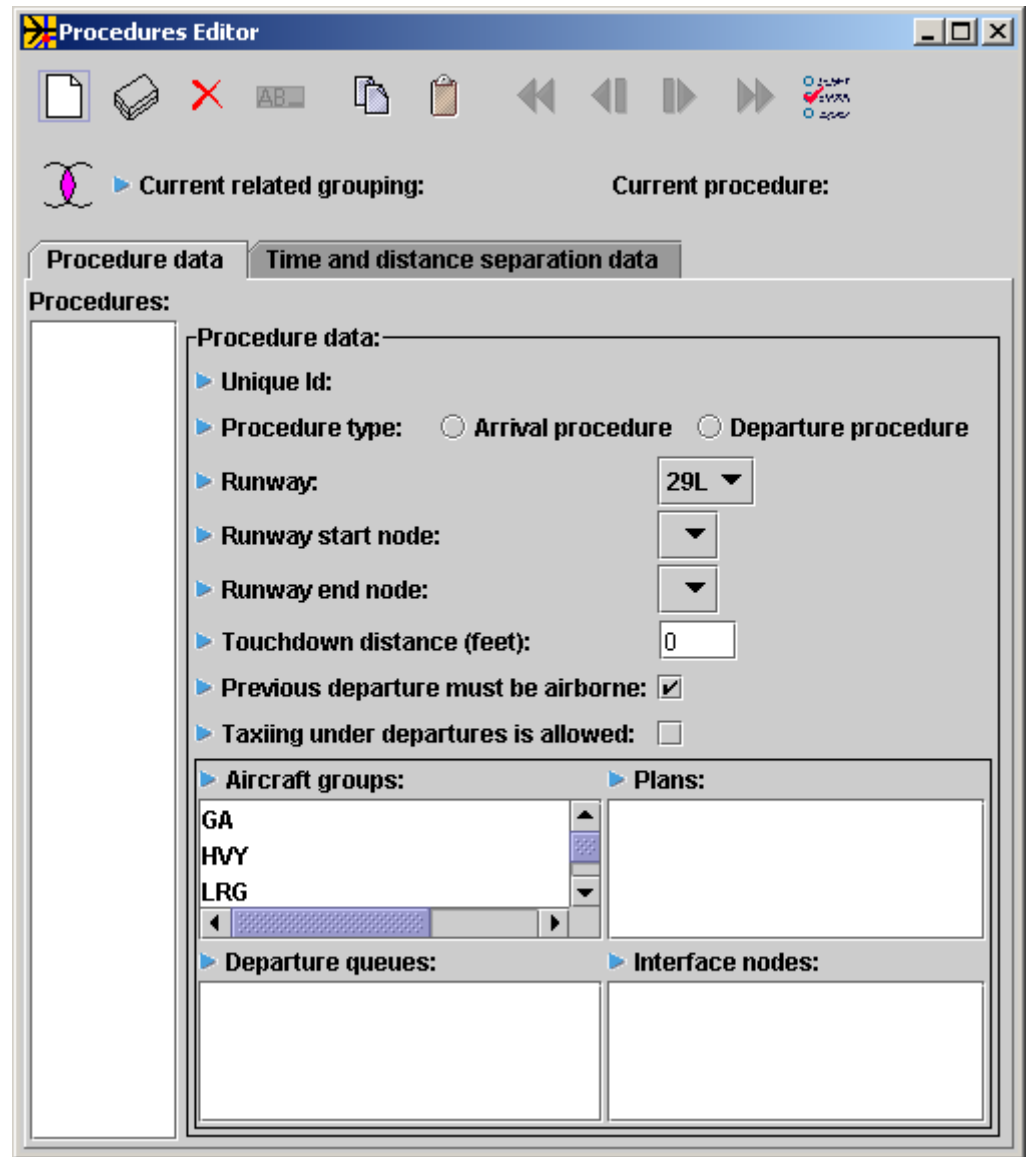
Please note the “Current related grouping” label. A related grouping is defined as a set of procedures that represent runways which have contingent flight operations with each other.

It is to be expected that separate airports within the same SIMMOD study would dictate separate related groupings.

Quite often at a particular airport the various runway operations are co-dependent and all of the procedures can be assigned to one related grouping.

But, it is certainly possible that a particular airport can have a set of runways which are widely spaced (or non-associated by some other factor) from another set of runways and the two sets do not interact in terms of one set delaying operations for another set.

In the above case two related groupings for a single airport would be appropriate, but rare.



Procedures Creation (continued)

Upon opening the Procedures Editor if Visual SIMMOD detects that a related grouping has not yet been created it will automatically create one called GROUP_1. You must activate this by pressing the  button.

The Related Procedures Groups Editor allows you to create another related grouping if required. Groupings can be deleted as well.


In this tutorial only one grouping is needed therefore you should click on the GROUP_1 grouping and then press the *Use* button.

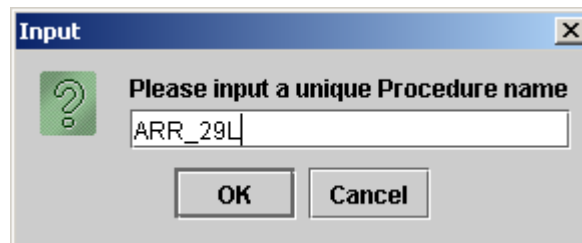
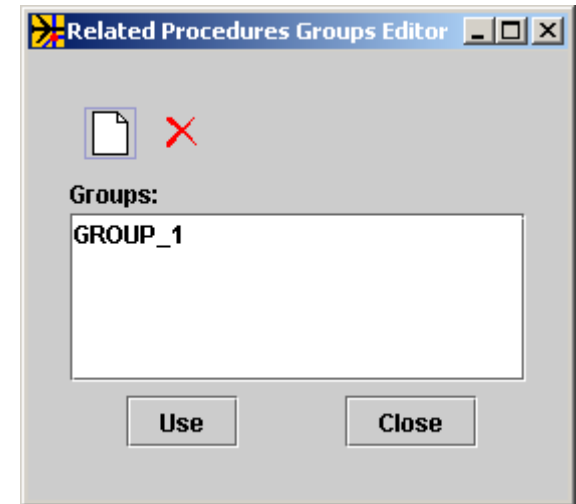
The Procedures Editor should now indicate that the Current related grouping is “GROUP_1”.

To keep the tutorial simple only two procedures will be defined.

An arrival procedure will define the time and distance separations for arrival aircraft on runway 29L.

A departure procedure will define the time and distance separations for departing aircraft on runway 29L.

Begin the procedures development by creating the arrival procedure for runway 29L. Press the  button on the Procedures Editor to create the procedure. Type **ARR_29L** in the window which opens and then press the *OK* button.



Procedures Creation (continued)

The ARR_29L procedure should now appear in the list of procedures to the left side of the editor.

Click on the “Arrival procedure” radio button to confirm the Procedure type.

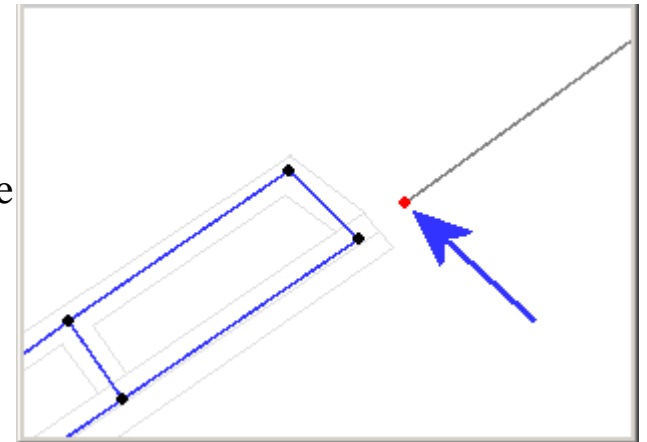
Choose Runway 29L as the runway to assign to this procedure.

You may skip down to the Plans list and choose PLN_01. Plans will be discussed later in this document.




Visual SIMMOD automatically creates a single plan for you when you open the Procedures Editor, assuming a plan doesn't already exist.

Next choose from the Interface nodes list the airspace node which is the last node on the TUT_ARR route, or rather, located nearest to the touchdown end of runway 29L as shown in the snapshot to the right.

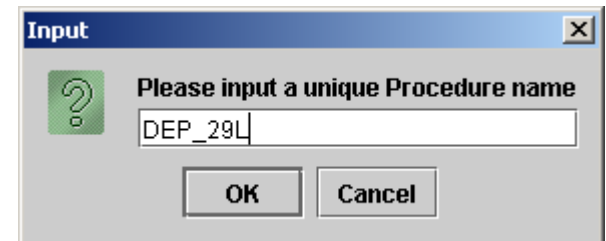


The next step in procedures creation is to create the departure procedure.

Press the  button on the Procedures Editor to create the procedure. Type **DEP_29L** in the window which opens and then press the **OK** button.

The DEP_29L procedure should now appear in the list of procedures to the left side of the editor.

Click on the “Departure procedure” radio button to confirm the Procedure type.



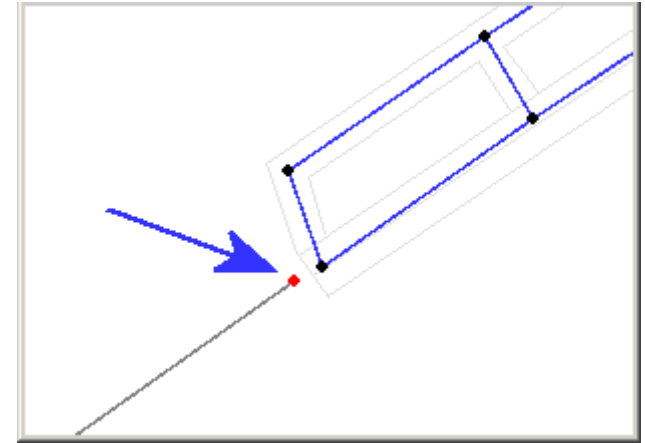
Procedures Creation (continued)

Choose Runway 29L as the runway to assign to this procedure.

You may skip down to the Plans list and choose PLN_01 from the list.

Choose the associated departure queue next, which in this case is DPQ_29L.

Next choose from the Interface nodes list the airspace node which is the first node on the TUT_DEP route, or rather, located nearest to the takeoff end of runway 29L as shown in the snapshot to the right.



! Since Visual SIMMOD duplicated the ARR_29L arrival procedure for you when it created the DEP_29L procedure you must deselect the arrival interface node (ASN_0004) from the current departure procedure.

To deselect the arrival interface node (ASN_0004) click on it once with the left mouse button. You will be asked if you want to remove the interface node from the Procedure. You should choose the answer **Yes**.

! Important note: You may have noticed that the list of aircraft groupings on the Procedures Editor and that all of the groupings have been assigned to both the arrival and departure procedures. This assignment of aircraft groupings is quite normal, but SIMMOD gives you the option to assign one (or more) aircraft grouping(s) to a particular procedure.

Thus, if this tutorial had chosen to have a procedure per aircraft grouping then there would have been a need to define four arrival procedures and four departure procedures, or eight procedures in total. A sample procedure identifier in that case would be ARR_HVY_29L.

Procedures Creation (continued)

You now have two procedures created belonging to the GROUP_1 related grouping. Visual SIMMOD has automatically created time and distance separation data for you. An explanation of how to edit the separation data follows:

To edit the separation data click the “Time and distance separation data” tab. You should see something similar to the snapshot on the right:

The spreadsheet to the left side displays the procedures in the first column and the aircraft groupings in the second column.

The spreadsheet in the middle displays the time (in seconds) that the procedure shown in the column header must be separated by if an aircraft of the grouping shown in the left side spreadsheet is executing the procedure shown in the left side spreadsheet.

The spreadsheet on the right displays the distance (in nautical miles) that the procedure shown in the column header must be separated by if an aircraft of the grouping shown in the left side spreadsheet is executing the procedure shown in the left side spreadsheet.

The screenshot shows the 'Procedures Editor' window with the following data:

Procedure data		Time and distance separation data			
PRC_ID	ASG_ID	ARR_29L	DEP_29L	ARR_29L	DEP_29L
ARR_29L	GA	60	60	2.00	2.00
	HVY	60	60	2.00	2.00
	LRG	60	60	2.00	2.00
	SML	60	60	2.00	2.00
DEP_29L	GA	60	60	2.00	2.00
	HVY	60	60	2.00	2.00
	LRG	60	60	2.00	2.00
	SML	60	60	2.00	2.00

Procedures Creation (continued)

In the example shown to the right if a LRG aircraft has just completed procedure ARR_29L any other aircraft wanting to execute procedure ARR_29L must wait until the LRG aircraft has been on the ground for 90 seconds.

PRC_ID	ASG_ID	ARR_29L	DEP_29L	ARR_29L	DEP_29L
ARR_29L	GA	60	60	0.00	0.00
	HVY	60	60	0.00	0.00
	LRG	90	60	0.00	0.00
	SML	60	60	0.00	0.00
DEP_29L	GA	60	60	2.00	2.00
	HVY	60	120	2.00	2.00
	LRG	60	60	2.00	6.00
	SML	60	60	4.00	2.00

If a HVY aircraft has just completed procedure DEP_29L any other aircraft wanting to execute procedure DEP_29L must wait until the HVY aircraft has been airborne for 120 seconds.


When an arrival executes ARR_29L, the subsequent aircraft arrivals or departures (i.e. execution of ARR_29L or DEP_29L) do not depend upon the distance that the original arrival has traveled, since most likely the aircraft will be taxiing. Thus, it is normal to enter zeros in these circumstances. In logic separate from procedures SIMMOD does ensure that departures do not begin their roll until the arrival exits the runway.

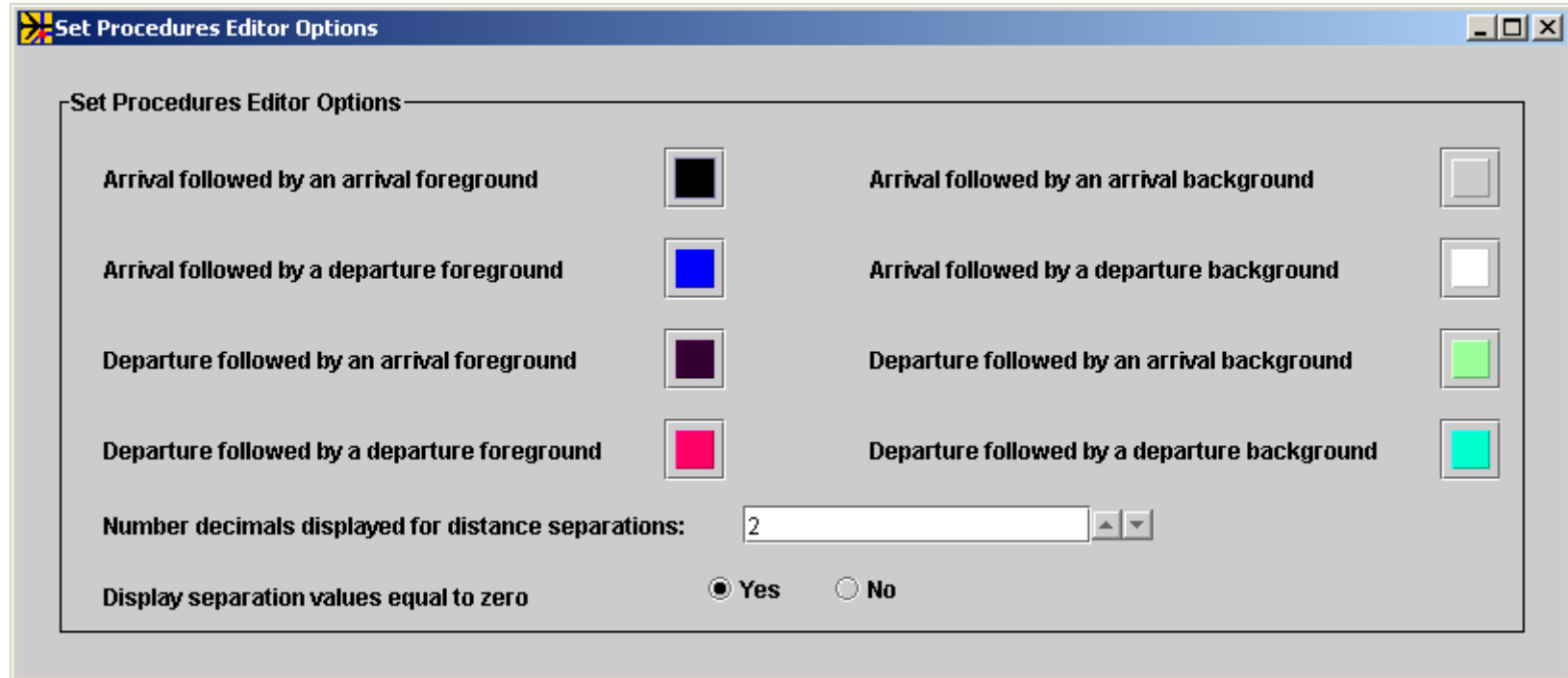
If a SML aircraft has just completed procedure DEP_29L any other aircraft wanting to execute ARR_29L must wait until the SML aircraft has flown 4.00 nautical miles away from the runway end.

The above procedure combination can normally be ignored since arrival operations normally are not contingent on the distance a departure has flown. It is shown here for the purposes of illustration only. It is suggested that you use the value zero instead.

If a LRG aircraft has just completed procedure DEP_29L any other aircraft wanting to execute DEP_29L must wait until the LRG aircraft has flown 6.00 nautical miles away from the runway end.

Procedures Creation (continued)

To change colors, decimals displayed, or zero values displayed in the Procedures Editor's spreadsheets press the  button:



You may now close the Set Procedures Editor Options window.

Press the  to write the new procedures to the database.

You may now close the Procedures Editor.

DSD Path Creation

Direction-of-travel conflicts are one of the most common SIMMOD problems which must be solved. When a taxiway is shared by aircraft which traverse it in either direction and that taxiway is too small for opposite direction passing a deadlock (gridlock) situation may occur in the simulation.

In the example shown, arrival aircraft (red) traveling to their gate traverse a link (in red) which is shared with departure aircraft (green).

When a departure aircraft is traversing the highlighted link all arrival aircraft should wait for the departure aircraft to finish taxiing upon the highlighted link.

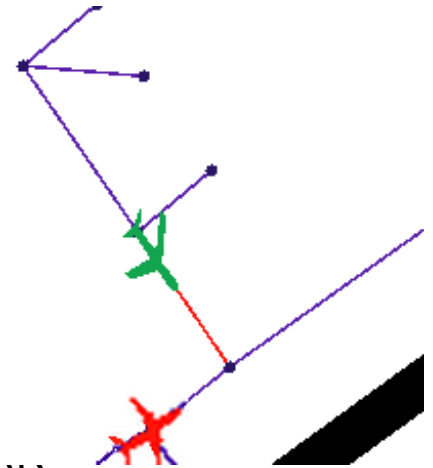
Normally SIMMOD would not make either aircraft wait. They would reach the link and depending on the link attributes (such as “no passing”) both aircraft would be placed into an infinite hold, thus creating a gridlocked condition and the simulation would crash.

When a DSD Path is defined, SIMMOD will anticipate any head-on conflicts. SIMMOD computes when aircraft will be traversing each DSD Path link.

If a head-on conflict will occur between two aircraft during taxi operations SIMMOD will allow the first aircraft to arrive at a DSD Path link to enter the DSD Path links unimpeded.

The SIMMOD will compute appropriate hold times for any additional aircraft desiring to enter the DSD Path links while the first aircraft is traversing the DSD Path links. The additional aircraft are permitted to enter the DSD Path when they no longer conflict with another aircraft.

The hold times computed for subsequent aircraft is always sufficient to allow the first aircraft to proceed unimpeded through the defined DSD Path links.



DSD Path Creation (continued)

If no conflict will occur, SIMMOD will allow more than one aircraft to enter a set of DSD Path links simultaneously. Thus, two aircraft traveling in the same direction will be permitted to enter the DSD Path simultaneously because the first aircraft never conflicts with the second.

The analyst may also supply a capacity value to the DSD Path if desired. SIMMOD will limit the number of aircraft in the DSD Path to the number of aircraft specified by the capacity value.

If the analyst supplies a capacity of one to the DSD Path then only one aircraft will ever be permitted to enter the DSD Path's links. All other aircraft, whether they create a conflict or not, will be held outside the DSD Path's links.

The creation of a DSD Path is very similar to the creation of a taxipath. The analyst decides which links will have conflicts and then assigns those links to a unique DSD Path.

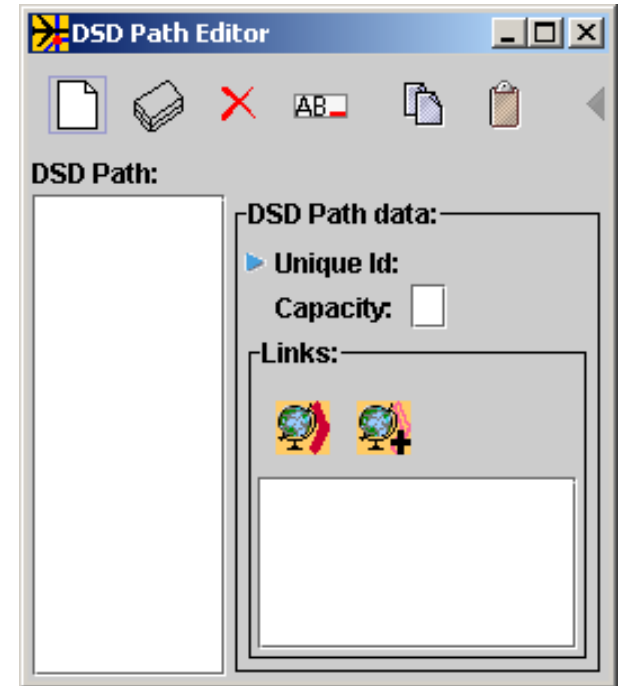
It is worth noting that the analyst should NOT assign a link to more than one DSD Path. Nor should the analyst assign a link which may be blocked for other reasons, such as gate link blocking, to a DSD Path.

The reason is that when SIMMOD computes the delay times for aircraft entering a DSD Path it assumes no other holding will change the conditions within the DSD Path. If an aircraft backing out of a gate blocks an aircraft currently traversing a DSD Path then all aircraft previously assigned to wait to enter the DSD Path will have incorrect release from holding times.

When those holding aircraft release and begin moving they may encounter the original aircraft which was attempting to traverse the DSD Path. And a head-on conflict may occur, despite the definition of a DSD Path.

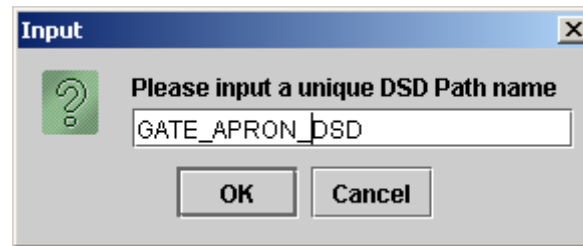
DSD Path Creation (continued)

To define a DSD Path begin by executing the *Tools//DSD Path Editor* menu item on the Network Builder. An empty DSD Path Editor is shown to the right.



Create a new DSD Path by pressing the  button.

Supply the name **GATE_APRON_DSD** and press the **OK** button.



Next, go to the Network Builder and zoom to the gate apron area on the map graphics.

Deselect all items on the map by pressing the  button.

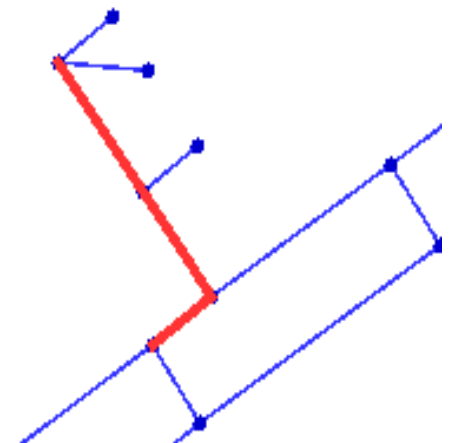
Now enable multiple airfield link selections by pressing the  button.

Select the three airfield links by clicking on each of them with the left mouse button. The links after having been selected are shown highlighted in red in the snapshot to the right:

On the DSD Path Editor press the  button to assign the links to the GATE_APRON_DSD path.

Press the  to write the new DSD Path to the database.

You may now close the DSD Path Editor.



Static Data Summary

Congratulations! You have now completed the requirements for the static portion of your SIMMOD tutorial study.

So far you have learned how to:

- Create a new study directory.
- Turn on and off graphics layer's visibilities.
- Save/retrieve a particular map graphics scene.
- Zoom and pan in the map graphics.
- Import a DXF file as a background layer.
- Import/turn on/off QGF background layers.
- Create/select an airport.
- Create an airfield link/node network.
- Move/edit/delete a node and/or link.
- Create a runway
- Create an airspace link/node network.
- Create an airspace route.
- Create an interface node.
- Work with the Generic Table Editor
- Create an airline.
- Create a gate.
- Create a departure queue.
- Create a SIMMOD taxipath.
- Create procedures.
- Create a DSD Path

Events Creation

SIMMOD events provide the dynamic element of your study. Events fall into the following basic categories:

- Flight events such as arrivals and departures.
- Static data modification events, such as dynamically setting node attributes.
- Program trace events such as turning on specific program debugging capabilities.

In addition to providing access to all of the SIMMOD events, Visual SIMMOD augments the usage of events by providing a unique mechanism for managing your events.

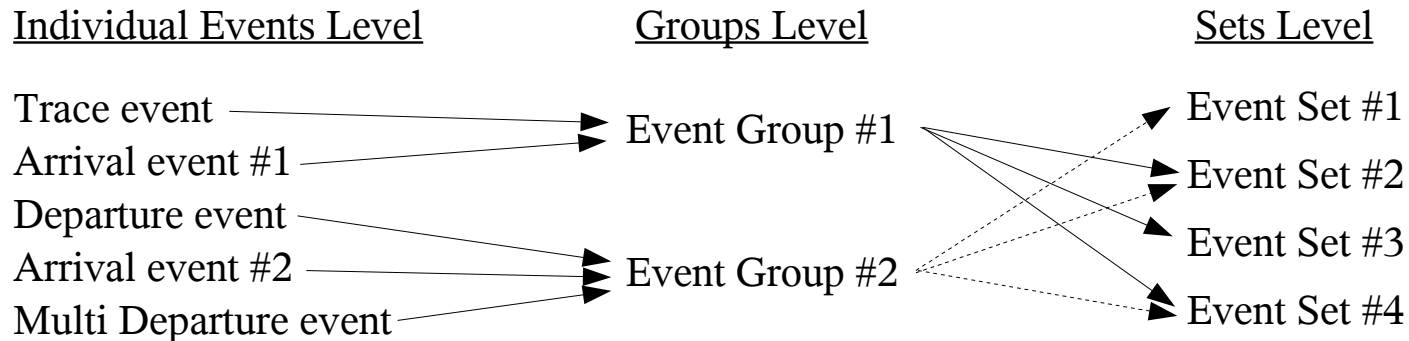
Visual SIMMOD provides the ability to group events into logical sets which can then be easily applied (or turned off) as needed to allow quick changes to how the study is run. Some examples are as follows:

- A set of arrival and departure events of a particular experimental aircraft model that can be enabled or disabled to quickly show before and after effects of adding a new aircraft model.
- A set of arrival and departure events which utilize a particular runway which can be enabled or disabled to show the effects of opening or closing the same runway.
- A logical set of debugging traces that can be enabled together when trying to determine the source of a problem.

It is also possible to apply time ranges when event groups will be active. Thus, allowing even greater granularity in the definition of events.

Events Creation (continued)

In Visual SIMMOD all individual events must belong to a particular Event Group. And all event groups must belong to at least one Event Set. It is possible for an Event Group to belong to more than one Event Set. The illustration below describes the possible relationships:



At each level the user can turn on or off whether the item will be enabled in their next SIMMOD simulation.

Turning off an Event Group means that any Individual Events assigned to it will not be supplied as an input to SIMMOD.

Turning off an Event Set means that any of its assigned child Event Groups will not be supplied as an input to SIMMOD, assuming the child Event Groups aren't assigned to any Event Sets that are still turned on.

For users who do not wish to utilize the event management functionality of Visual SIMMOD it is possible to create one event grouping and assign all individual events to that grouping. Then one event set should be created and the one grouping assigned to this set.

To help you create and manage Event Sets and Groups Visual SIMMOD provides the Event Management Editor. This tool is described next in the tutorial.

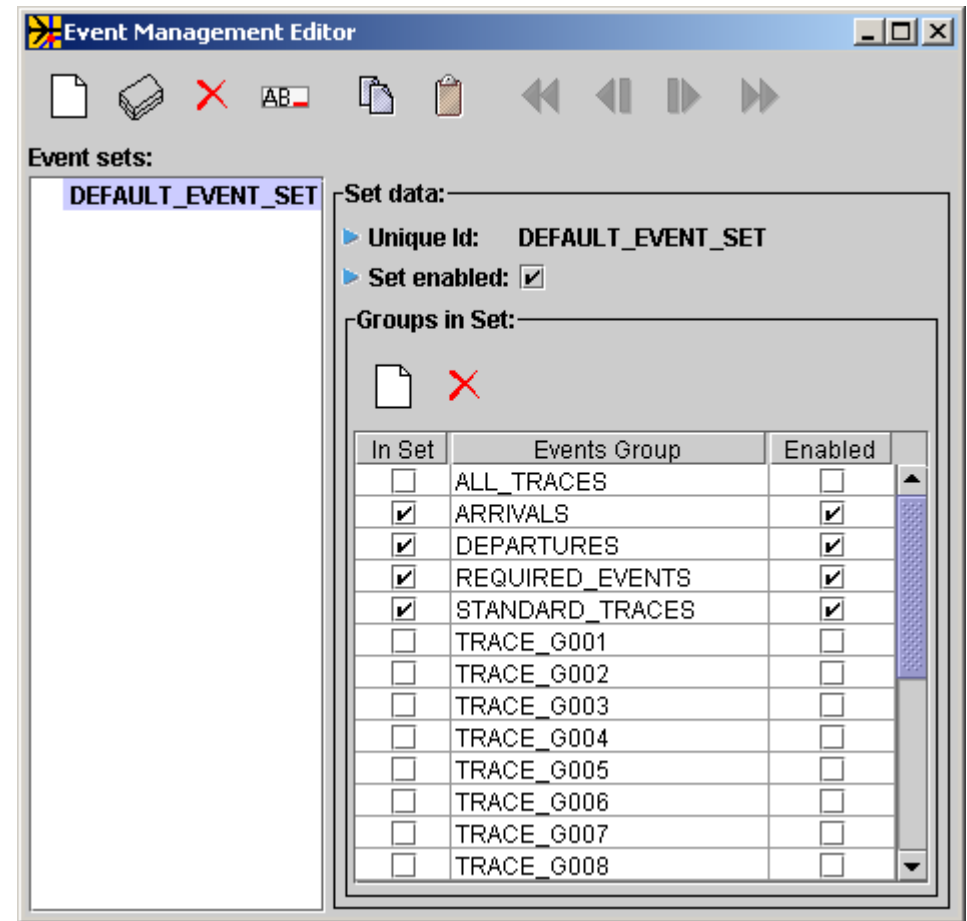
Event Management Editor

The Event Management Editor provides the ability to create and delete event groups as well as event sets.

An Event Management Editor is shown to the right. To open the Event Management Editor execute the ***Tools//Events//Event Management Editor*** menu item.

During the creation of a new study Visual SIMMOD creates a default event set (**DEFAULT_EVENT_SET**) and several default event groups. This set and groups are shown in the snapshot to the right. Click on the default set to view the groupings.

As you create events, such as an arrival flight, you must assign them to an event group. If an event doesn't logically belong in the default groups provided then it is suggested that you create a new grouping and assign it to this new grouping.



New event sets may be created as needed using this tool via the toolbar buttons at the top of tool.

Groupings may be assigned to a set by clicking the set identifier then on the checkbox in the “In Set” column. Groups may be disabled for a particular set by toggling the checkbox in the “Enabled” column.

You may close the Event Management Editor.

SIMMOD Events

The next section of the tutorial will cover the creation of individual SIMMOD events. The events which are described are necessary for the correct execution of SIMMOD.

As noted earlier each individual SIMMOD event must belong to an Event Group which itself must belong to at least one Event Set.

The individual events which are described in the remainder of this document are as follows:

- SETPLAN Event – Designates the PLAN and the associated runways under which SIMMOD is to execute.
- ENDSIM Event – Designates when the simulation should end. If not supplied the simulation may run indefinitely.
- TRACE Events – TRACE events direct SIMMOD to output specific simulation statistics, debugging output, and various log files.
- ARRIVAL Flight Events – Designates arrival flights for SIMMOD to execute.
- EMPLANE Flight Events – Designates departure flights for SIMMOD to execute.

SETPLAN Event


SIMMOD has the capability to model multiple airport plans during the modeled day. Visual SIMMOD creates a default plan called PLN_01 for you during the initial Application creation process.

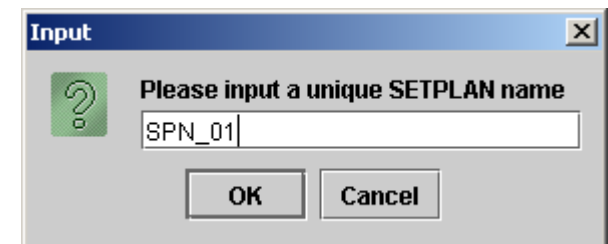
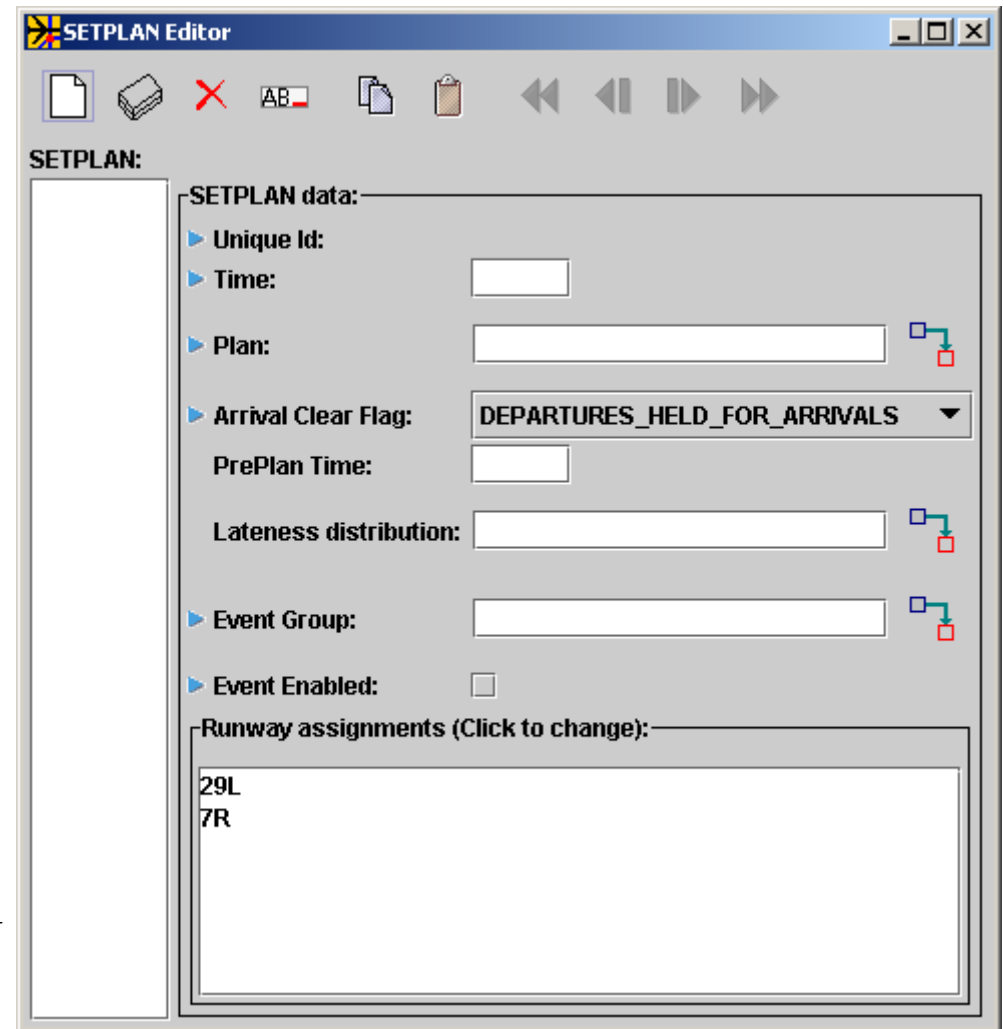
For differing airport operating conditions, such as differing prevailing winds, you may create multiple plans.

To indicate which plan is currently in effect SIMMOD requires a SETPLAN event.

The SETPLAN event indicates the time the plan goes into effect, which plan is being run, and the runways which are active during the plan.

Visual SIMMOD provides a tool called the SETPLAN Editor to ease editing of the SETPLAN data. To open a SETPLAN Editor execute the **Tools//Events//SetPlan** Editor menu item.

Begin the SETPLAN development by pressing the  button on the SetPlan Editor. Type **SPN_01** in the window which opens and then press the **OK** button.



SETPLAN Event (continued)

The first SETPLAN record of the day normally should be assigned a time of 00:00:00. If necessary type **00:00:00** in the Time field.

The plan in this case will be PLN_01. Type **PLN_01** in the Plan field.

The Arrival Clear Flag may be set to **NO_CONSTRAINT** since in this tutorial there will only be one plan thus there is no need for departures to wait for all arrivals to land before taking off.

Type **REQUIRED_EVENTS** into the Event Group field since all SIMMOD studies should have at least one SETPLAN event.

In this case select only the **29L** runway from the assigned runways list since it is the primary direction runway and it would not be logical to have the 7R opposite direction runway enabled at the same time.

To write the newly created SETPLAN press the  button.

You may now close the SETPLAN Editor window.

END.SIM Event


The END.SIM event tells SIMMOD when to end the simulation. It is a requirement that you specify at least one END.SIM event otherwise the simulation will never finish.

There should never be more than one enabled END.SIM event record. You may have multiple END.SIM records, but all of them, except the active one, should be disabled.



No enabled events should ever occur after an enabled END.SIM event.

To specify an END.SIM event you should execute the **Edit//Edit Tables** menu item. Choose the **Events... Control Events...** grouping at the left and then the **END_SIM** table to the right.

Press the  button to retrieve all of the currently stored END_SIM events.

To speed SIMMOD study development Visual SIMMOD creates a default END.SIM event for you at time 30.0 (30 hours into simulation time). You may choose to keep this END.SIM event, edit this event, or create a new END.SIM event using the techniques described earlier for editing data using the Generic Table Editor.

You may close the END_SIM's Generic Table Editor window.

TRACE Events

TRACE events direct SIMMOD to log to a file a particular aspect of activities which are happening within a running SIMMOD simulation. Some examples include logging of:

- The beginning of a takeoff roll.
- A gate reassignment.
- Sector boundary crossings
- Aircraft being placed in a queue for deicing.

All traces are referred to by their array index number within SIMMOD. Thus, there is a trace number 52 and a trace number 146. TRACE numbers are added to new versions of SIMMOD when the need arises, thus you should be aware that the “maximum” trace number can and does change occasionally.

A majority of the current SIMMOD TRACE numbers are between 0 and 450.

Most SIMMOD traces are logged to the SIMU04 output file. A few traces are significant enough that they are logged to their own distinct files. A notable example is trace number 302 which directs SIMMOD to print the log file (SIMU26 history file) of a flight's activities in a format suitable for reading with an animation or reporter program.

The [SIMMOD Flat Files Documentation](#) is the definitive guide to the trace numbers and their meanings. It is recommended that you consult that document when you need explanation of a particular trace number.


A majority of Traces are toggled such that they are either on or off. As with all SIMMOD events, traces have an associated time when they are to be invoked.

You may specify a trace to turn on at 7:00:00am, to turn off at 7:30:00am, and to turn back on at 9:30:00pm.

TRACE Events (continued)

Upon creation of a new Application Visual SIMMOD builds several TRACE event groupings automatically for you. The most important of these is the STANDARD_TRACES event grouping. This grouping includes traces that a majority of users need to perform SIMMOD studies. Included are traces to log route statistics, aircraft movements, and accumulated undelayed and delayed travel statistics.

Another important TRACE event grouping which is automatically created by Visual SIMMOD is the ALL_TRACES grouping. The ALL_TRACES grouping includes all of the SIMMOD traces.

 **Caution: The ALL_TRACES grouping should only be used when trying to determine the nature of a problem occurring within SIMMOD. When enabled, extremely large volumes of logging information will be printed to the SIMU04 file. Depending on the complexity of the simulation even large hard drives could be completely filled by this capability.**


SIMMOD traces one through twenty are special group traces. When toggled on they actually turn on multiple other “related” traces. An example is trace number 5. When toggled on SIMMOD automatically turns on traces 79 through 86 which are traces to log node holding activities by flights.


Visual SIMMOD automatically creates an event grouping for each of the one through twenty traces. The event groupings are named GRP_TRACE_001, GRP_TRACE_002, and so on. You should exercise caution using these trace groupings since they too can create large volumes of debugging messages.

To add your own trace events to SIMMOD's event stream you should use Visual SIMMOD's Generic Table Editor.

Begin by executing the *Edit/Edit Tables* menu item. Then select the *Events... Control Events...* grouping from the left column and then the *TRACE* table from the lower right column.

TRACE Events (continued)

Press the  button to display the currently stored traces, if any.

To add a new trace press the  button. The new trace will appear at the bottom of the spreadsheet. If you had previously pressed the mouse on an existing trace then a duplicate of that trace will appear on the new line. Otherwise a blank line will appear.

Type **MY_TRACE** in the TRC_ID column. This field must contain a unique value.

Type **7.0** in the EVENT_TIME column. This specifies that the event will take place at 7:00:00am.

Type **TRC_NUM_000163** in the TRC_TYPE column. Trace number 163 indicates that you want detailed debugging information about node arrival/departures.

Type **ON** in the ON_OFF_FLAG column. This flag turns on or off the trace at the specified time. Use OFF to turn the trace off.

Type **ALL** in the ITERATION_FLAG column. The value ALL indicates that this trace will be executed in every iteration of the study. Other options include: NEVER and ONE. If ONE is chosen then you must supply a valid integer iteration number in the ITERATION_NUM field.

Type **STANDARD_TRACES** in the EVG_ID column. This column stores the trace's event group assignment.

Type **T** in the EVENT_ON_TF column. By specifying T (true) in this column you have indicated to Visual SIMMOD that you want this trace printed to the SIMMOD events file (SIMU09). Use F (false) to prohibit printing this trace to the events file.

TRACE Events (continued)

To store your newly created trace press the  button.

When SIMMOD is run with your trace turned on all node arrivals and departures will be logged to the SIMU04 file. You can locate the 163 trace entries by searching for **-163**).

You may close the TRACE table's Generic Table Editor window.

ARRIVAL Flight Events

Most SIMMOD studies will include several, possibly hundreds, arrival events at the modeled airports.

Before attempting to create an arrival event you will need to know the arrival flight's airline, the aircraft model used, the arrival route it is expected to traverse, and when the flight is expected to appear at the outermost node of the arrival route.

To obtain the outermost node arrival time you will need to perform a calculation similar to the following:

	OAG gate arrival time	8:00:00am
less	Average taxi-in duration	6 minutes
Result	Runway touchdown time	7:54:00am
less	Nominal route travel time	22 minutes
Result	SIMMOD arrival event injection time	7:32:00am

In most cases a standard taxi-in duration is sufficient. It is difficult to compute a nominal taxi-in duration since flights may take a different runway exit and be assigned to a different gate per each successive iteration run by SIMMOD.


Nominal route travel times can be obtained from the SIMMOD simulation by first inserting the OAG arrival time in the ARRIVAL event's time field. Then make a normal SIMMOD run (discussed later in this tutorial). After SIMMOD has run a SIMU38 file will have been produced which contains the nominal route travel time by flight. You can then subtract the indicated route travel time (presented in the SIMU38 file in both decimal hours and in seconds) as shown in the example above.

ARRIVAL Flight Events (continued)

To add your own arrival events to SIMMOD's event stream you should use Visual SIMMOD's Generic Table Editor.

Begin by executing the *Edit/Edit Tables* menu item. Then select the *Events... Flights...* grouping from the left column and then the *FLIGHTS* table from the lower right column.

Press the  button to display the currently stored flights, if any.

To add a new arrival flight press the  button. The new flight will appear at the bottom of the spreadsheet. If you had previously pressed the mouse on an existing flight then a duplicate of that flight will appear on the new line. Otherwise a blank line will appear.

Type *MY_ARRIVAL* in the FLD_ID column. This field must contain a unique value. Usage of the flight number in this field is not recommended since numeric values may be misinterpreted by SIMMOD and in some instances flight number's may be duplicated during a particular days schedule.

Type *8.0* in the EVENT_TIME column. This specifies that the event will take place at 8:00:00am.

Type *A* in the A_D_FLAG column. This specifies that the flight is an arrival flight.

Type *DZZ* in the ALN_ID column. This specifies the flight's airline.

Type *932* in the FLT_NUMBER column. This column specifies the flight number information for the flight and it may be safely duplicated in other rows.

ARRIVAL Flight Events (continued)

Type **737** in the ACM_ID field. This is the aircraft model field.


Type **TUT_ARR** in the RTE_ID field. This field specifies the route.

Type **ARRIVALS** in the EVG_ID field. This specifies that this event belongs to the ARRIVALS event grouping.

Type **T** in the EVENT_ON_TF field. This specifies that this particular event should be printed to the SIMU09 file.



Note: The FLIGHTS table is a unique table in that it supports two SIMMOD input cards (ARRIVAL and EMPLANE cards). Thus, there are many fields that are appropriate only to arrivals and others that only appropriate to departures. And, there are many fields which are common to both arrivals and departures in the FLIGHTS table as well. Lastly, as with so many other SIMMOD inputs there are several fields which are optional. Thus, many fields are left blank in the above described arrival event.

To store your newly created arrival press the  button.

Given the above information, when SIMMOD is run, a 737 belonging to airline DZZ will be injected onto the TUT_ARR route for an arrival operation. Because no gate was specified the aircraft will be randomly assigned to a gate to which airline DZZ has permission to park. And because no taxipath was specified the aircraft will be assigned an optimal taxipath from it's runway exit point to the eventually assigned gate.

EMPLANE (Departure) Flight Events


In SIMMOD, EMPLANE (Departure) events are similar to arrival events. The time of the event, the airline, the aircraft model, and the route must be supplied for an EMPLANE event. Additionally, EMPLANE events require an origin airport. And some users may want to supply a lateness distribution.

A lateness distribution specifies a distribution of probabilities and associated time durations. SIMMOD takes a random draw from this probability distribution and determines the time duration that the flight will be late departing.

To add your own emplane events to SIMMOD's event stream you should use Visual SIMMOD's Generic Table Editor.

Begin by executing the *Edit/Edit Tables* menu item. Then select the *Events... Flights...* grouping from the left column and then the **FLIGHTS** table from the lower right column.

Press the  button to display the currently stored flights, if any.

To add a new emplane flight press the  button. The new flight will appear at the bottom of the spreadsheet. If you had previously pressed the mouse on an existing flight then a duplicate of that flight will appear on the new line. Otherwise a blank line will appear.

Type **MY_EMPLANE** in the FLD_ID column. This field must contain a unique value. Usage of the flight number in this field is not recommended since numeric values may be misinterpreted by SIMMOD and in some instances flight number's may be duplicated during a particular days schedule.

Please remember to hit enter after typing in a cell to insure that all data is properly entered.

Type **9.0** in the EVENT_TIME column. This specifies that the event will take place at 9:00:00am.

EMPLANE Flight Events (continued)

Type **D** in the A_D_FLAG column. This specifies that the flight is an departure flight.

Type **FLH** in the ALN_ID column. This specifies the flight's airline.

Type **1599** in the FLT_NUMBER column. This column specifies the flight number information for the flight and it may be safely duplicated in other rows.

Type **757PW** in the ACM_ID field. This is the aircraft model field.

Type **TUT_DEP** in the RTE_ID field. This field specifies the departure route which will be traversed.

Type **TUT** in the APT_ORIG_ID field. This field specifies the origin airport which is needed by SIMMOD to determine the gate to assign to the flight.

Type **DEPARTURES** in the EVG_ID field. This event belongs to the DEPARTURES event grouping.

Type **T** in the EVENT_ON_TF field. This specifies that this event should be printed to the SIMU09 file.


To store your newly created emplane press the  button.

Given the above information, when SIMMOD is run, a 757PW belonging to airline FLH will be assigned to a random gate at 9:00:00am. Since no taxiway was specified the flight will proceed to the active runway's departure queue via an optimal taxipath and take off upon runway availability. After takeoff the plane will fly the TUT_DEP route. After flying the TUT_DEP route the plane will be ejected from SIMMOD.

You may now close the FLIGHTS Generic Table Editor window and the Workbench window.

Running SIMMOD

If you have supplied Visual SIMMOD with the information as described in the previous pages of this tutorial then you are now ready to run the SIMMOD model.

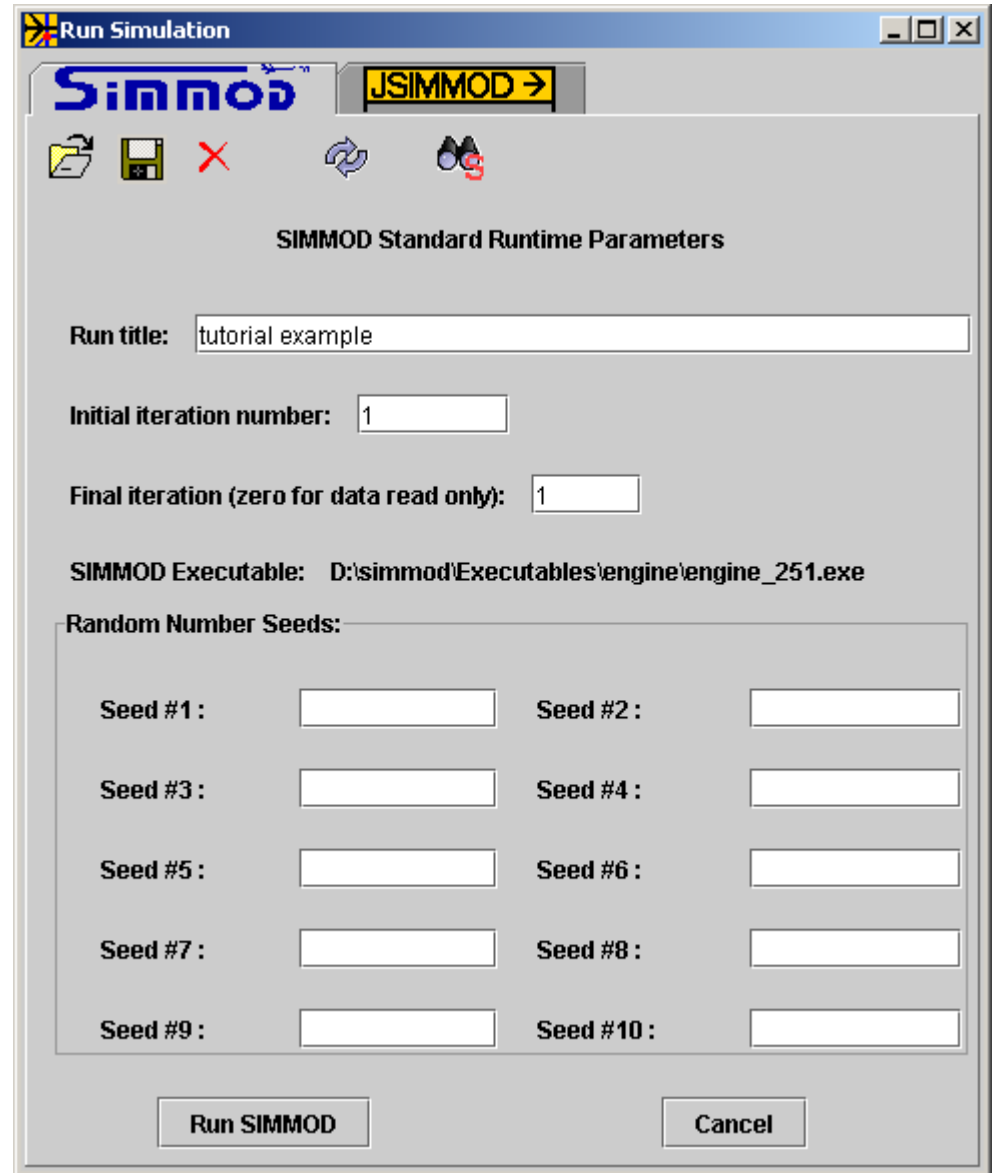
Before you attempt to run the SIMMOD model it is advisable that you save your work. Press the  button on the Network Builder to save all of your previous inputs.

To run the SIMMOD model begin by executing the *Simulation//Run Simulation* menu item on the Network Builder. The Run Simulation tool will open and appear similar to the snapshot on the right.

In the snapshot on the right you will notice the two tabs at the top of the window. By selecting the SIMMOD tab you are specifying that you want to run the SIMMOD model.

By selecting the JSIMMOD tab you could run JSIMMOD instead.

Please choose the SIMMOD tab in this instance.



In the example shown a run title has been supplied. Please type **tutorial example** in the Run title field.

Running SIMMOD (continued)

The data supplied in the Run title field will be placed into the RUNDATA file which is read by SIMMOD. The run title will then be printed in many of the output files produced by SIMMOD for identification purposes.


From a single set of inputs SIMMOD can be directed to run multiple iterations. SIMMOD refers to iterations by number. The initial iteration number can be specified in the Initial iteration number field. In this case type **1** in the field.

If you want SIMMOD to run multiple iterations then you should type in the Final iteration field an integer value greater than what was placed in the initial iteration number field. The positive difference between the two fields plus 1 is how many iterations SIMMOD will produce. Typing a zero in this field will cause SIMMOD to read the simulation inputs you have prepared, check them for correctness, and then to exit without executing any simulation.

In this case type **1** in the Final iteration field.

When SIMMOD initially starts its run it will by default create ten random number streams with each stream having a random number seed which may be affected by the inputs supplied to the program. Thus, if an input is changed, the random number seeds may change and your results may vary.


To insure a particular set of random number seeds, rather than accepting a possibly changing set of seeds, you can supply your own set of seeds. You can supply these seeds by filling in the fields in the Random Number Seeds section.

Visual SIMMOD can generate a set of random number seeds for you by pressing the  button and supplying a single integer value when requested. Subsequently, if you supply the same single seed using this method you will get the same ten random number seeds.

Running SIMMOD (continued)

If multiple iterations are run, iterations beyond the first will receive their own unique set's of random number seeds. These unique sets will be dependent on the set used in iteration number one and they are repeatable, assuming you supplied a set for the first iteration.

The user may save and retrieve the simulation parameters by pressing the save and load buttons on the Run Simulation window.

Additionally, by pressing the  button the user can pick a particular SIMMOD simulation version to run.

To run SIMMOD press the *Run SIMMOD* button at the bottom of the Run Simulation window.

If your database has not been saved Visual SIMMOD will ask you do you want to save it. Answer Yes or Ok.

Then a Supplemental Simulation Results Window will open. In this window will be any error or warning messages from Visual SIMMOD concerning problems with your input data.

At this time Visual SIMMOD reads the database, organizes your data, looks for any errors, and prints your data in the SIMMOD required formats.

Any error messages should be investigated. If an error is detected by Visual SIMMOD all further processing will stop and SIMMOD will not be run.

A couple of error messages concerning Windsets normally occur but they may be safely ignored.

If no errors are reported then SIMMOD is executed. The execution of the model occurs in the Simulation directory of the current application. Thus, all output files are written to this directory.

Running SIMMOD (continued)

Some of the most important files to be found in the Simulation subdirectory of your SIMMOD Application are as follows:

RUNDATA – Contains your study's title, the starting/ending iteration values, and the random number seeds.

SIMU01 – Contains the study's input aircraft model definitions. Visual SIMMOD provides you with a large number of models as a default.

SIMU02 – Contains various echoes of input data read by SIMMOD as well as any warning or error messages printed during the input read process.

SIMU03 – Contains the study's input airspace definitions.

SIMU04 – Contains trace messages printed by SIMMOD during the running of the simulation. May also contain error messages printed by SIMMOD when problems are encountered during the running of the model.

SIMU07 - Contains the study's input airfield definitions.

SIMU09 - Contains the study's input events.

SIMU10 – Standard report containing average travel and delay times for both airspace and airfield aspects of the study.

SIMU14 – Contains statistics gathered about each flight during the running of the model.

Running SIMMOD (continued)

SIMU18 – Contains route travel time statistics.

SIMU26 – Contains a node-by-node history of each flight's movements.

SIMU36 – Contains a listing by flight of information useful for input to the INM model.

SIMU38 – Contains nominal route travel times.

SIMU40 – Contains link/node information presented in a format that is comma-delimited and more readily processed by third party programs.

SIMU44 – Contains supplemental statistics concerning the operations of the various flights.

The SIMMOD required naming convention is that odd number files are input files and even numbered files are output files. SIMMOD will only read the odd numbered files as noted above. It is not possible to rename them.

You can edit the odd numbered files using an ASCII text editor. It is recommended that you keep these types of edits small and only for quick “test” cases.

Visual SIMMOD writes a batch file called `run_simmod.bat` which you can use to run SIMMOD outside of the Visual SIMMOD graphical user interface.

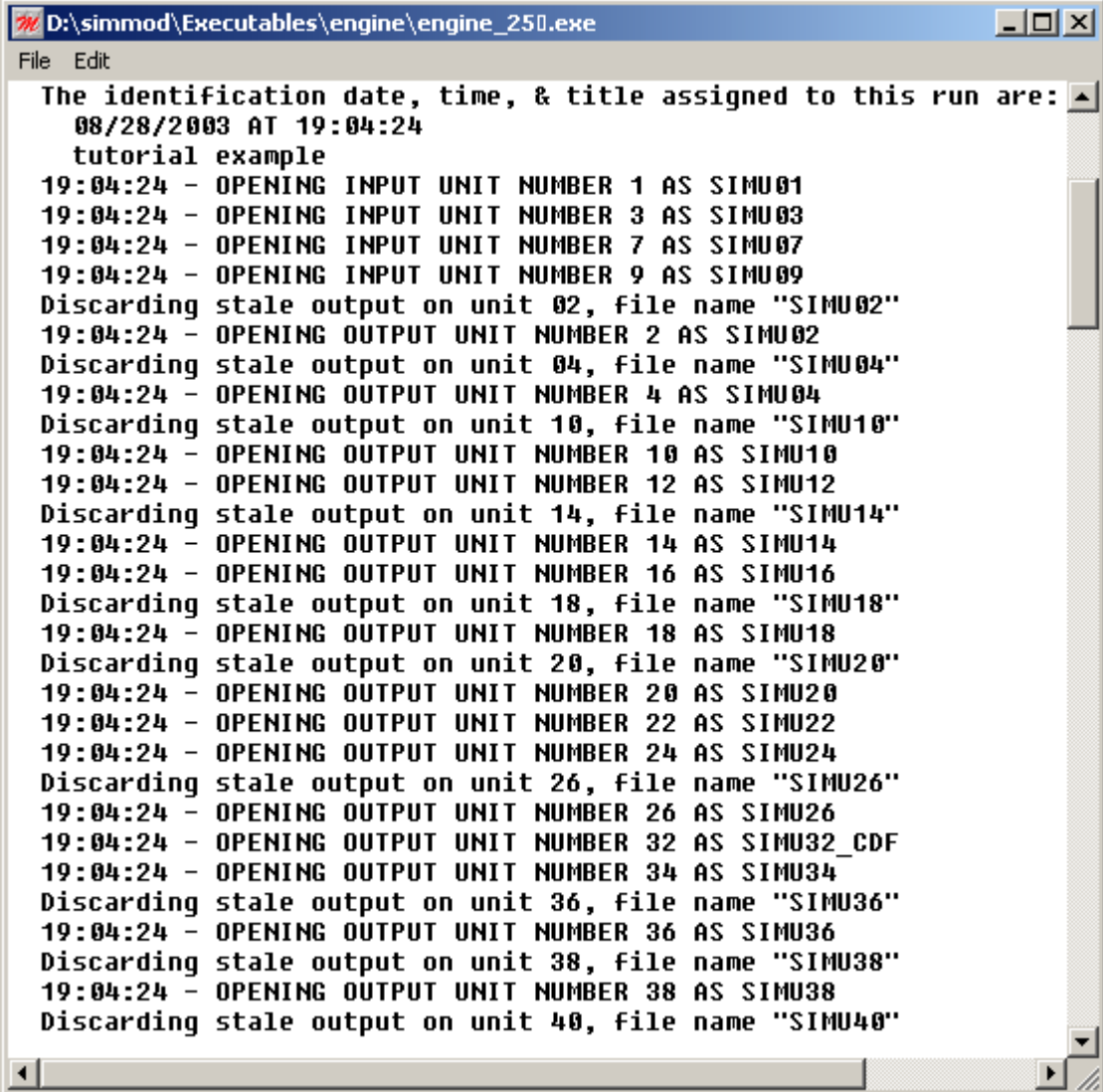
Running SIMMOD (continued)

The snapshot to the right displays SIMMOD running. In general, the echoes displayed are as follows:

- SIMMOD version number.
- A year 2000 disclaimer.
- The title of the run as supplied by you.
- The opening and closing of files.
- Random seed values.
- Input data processing progress messages.
- Linking of input data progress messages.
- Warning and error counts.
- Static data printing echoes.
- Simulation progress messages

After SIMMOD has executed successfully all of the output files will have been written.

You can immediately open the APECs Animator to view the simulation results as stored in the SIMU26 file.



```
D:\simmod\Executables\engine\engine_250.exe
File Edit
The identification date, time, & title assigned to this run are:
 08/28/2003 AT 19:04:24
tutorial example
19:04:24 - OPENING INPUT UNIT NUMBER 1 AS SIMU01
19:04:24 - OPENING INPUT UNIT NUMBER 3 AS SIMU03
19:04:24 - OPENING INPUT UNIT NUMBER 7 AS SIMU07
19:04:24 - OPENING INPUT UNIT NUMBER 9 AS SIMU09
Discarding stale output on unit 02, file name "SIMU02"
19:04:24 - OPENING OUTPUT UNIT NUMBER 2 AS SIMU02
Discarding stale output on unit 04, file name "SIMU04"
19:04:24 - OPENING OUTPUT UNIT NUMBER 4 AS SIMU04
Discarding stale output on unit 10, file name "SIMU10"
19:04:24 - OPENING OUTPUT UNIT NUMBER 10 AS SIMU10
19:04:24 - OPENING OUTPUT UNIT NUMBER 12 AS SIMU12
Discarding stale output on unit 14, file name "SIMU14"
19:04:24 - OPENING OUTPUT UNIT NUMBER 14 AS SIMU14
19:04:24 - OPENING OUTPUT UNIT NUMBER 16 AS SIMU16
Discarding stale output on unit 18, file name "SIMU18"
19:04:24 - OPENING OUTPUT UNIT NUMBER 18 AS SIMU18
Discarding stale output on unit 20, file name "SIMU20"
19:04:24 - OPENING OUTPUT UNIT NUMBER 20 AS SIMU20
19:04:24 - OPENING OUTPUT UNIT NUMBER 22 AS SIMU22
19:04:24 - OPENING OUTPUT UNIT NUMBER 24 AS SIMU24
Discarding stale output on unit 26, file name "SIMU26"
19:04:24 - OPENING OUTPUT UNIT NUMBER 26 AS SIMU26
19:04:24 - OPENING OUTPUT UNIT NUMBER 32 AS SIMU32_CDF
19:04:24 - OPENING OUTPUT UNIT NUMBER 34 AS SIMU34
Discarding stale output on unit 36, file name "SIMU36"
19:04:24 - OPENING OUTPUT UNIT NUMBER 36 AS SIMU36
Discarding stale output on unit 38, file name "SIMU38"
19:04:24 - OPENING OUTPUT UNIT NUMBER 38 AS SIMU38
Discarding stale output on unit 40, file name "SIMU40"
```

Final Thoughts

AirportTools hopes you found this tutorial useful. We certainly enjoyed bringing it to you. It is always nice to know that another person has joined the ever growing community of SIMMOD users.

As you continue to use SIMMOD you will discover that this tutorial has only scratched the surface of what this amazing model can do.

We highly encourage you to promote SIMMOD in your work. It is a well thought-out model with over twenty years of development work put into it.

Thank you for your support!