Contents

Module 1: Introduction ............................................................. 1.1
   Prerequisites .......................................................................................... 1.1
   Housekeeping ......................................................................................... 1.1
   Course Length ......................................................................................... 1.1
   Course Materials ..................................................................................... 1.1
   Course Objectives .................................................................................. 1.2
   What is Scriptwriter Plus™? .................................................................. 1.2
      System Requirements ........................................................................... 1.2
      Features and Benefits .......................................................................... 1.3

Module 2: Scriptwriter Plus™ Run .......................................... 2.1
   Module Objectives .................................................................................. 2.1
   Activate Scriptwriter Plus Run ................................................................. 2.1
   Run a Script from an Icon ........................................................................ 2.2
   Run a Script Using File Name .................................................................... 2.3
   Minimize a Script ..................................................................................... 2.4
   Terminate a Script ................................................................................... 2.4
   Group Windows and Icons ...................................................................... 2.5
      Create a New Group Window .............................................................. 2.5
      Add an Icon ........................................................................................ 2.7
      Save Settings on Exit ........................................................................... 2.7
      Modify Group Windows and Icons ....................................................... 2.8
      Delete Group Windows and Icons ......................................................... 2.9
   Module Review ......................................................................................... 2.10

Module 3: Scriptwriter Plus™ Build - Basics .......................... 3.1
   Module Objectives .................................................................................. 3.1
   Activate Scriptwriter Plus™ Build .......................................................... 3.1
      When to Use ........................................................................................ 3.1
      How to Use .......................................................................................... 3.1
   Window Components Overview .............................................................. 3.2
   Plan a Script Using Online Help .............................................................. 3.4
Module 4: Scriptwriter Plus™ Build – Beyond Basics........ 4.1

Module Objectives ................................................................. 4.1
Conditional Logic...................................................................... 4.1
  When to Use........................................................................ 4.2
  Include Radio Button in Input Window .................................. 4.3
  Include If Statement in Command Window .......................... 4.5
  Example: Conditional Logic Using Three Radio Buttons .... 4.6
  Example: Conditional Logic with an Exact Text Match........ 4.7
Include a Checkbox ............................................................... 4.9
  Exercise – Timatic Script .................................................. 4.10
Build a Template Script .......................................................... 4.11
Browser ................................................................................ 4.12
  Copy Labels between Scripts ............................................ 4.13
Validation .............................................................................. 4.15
  When to Use .................................................................... 4.15
  How to Use ...................................................................... 4.15
  Validate a Label .............................................................. 4.16
Delete Global and Local Labels .............................................. 4.18
Comments ............................................................................ 4.20
Pause and Painted Messages .................................................. 4.21
  Exercise – Hotel, Car and Visa ........................................ 4.22
Notes
Module 1: Introduction

Welcome to the Focalpoint® Scriptwriter Plus™ course. In this class, you will learn the techniques needed to write basic scripts and how to run them.

Prerequisites

Class participants should have a working knowledge of the Apollo® system and Microsoft® Windows® functionality.

Housekeeping

Please make note of the following items:

- Fire Exits
- Restrooms
- Smoking policy
- Breaks
- Communication

Course Length

The course length is four days.

Course Materials

The Scriptwriter Plus Course Book provides areas to take notes, plan scripts and complete exercises. It is designed to be used as a learning tool during class. Once you have completed the Scriptwriter Plus class, use this Course Book as a reference guide back at your office.
Course Objectives

At the end of the course, you will be able to:

- Execute a script using Scriptwriter Plus Run.
- Recognize Scriptwriter Plus Build window components.
- Plan and design scripts for a specific purpose.
- Build an Input window.
- Build a Command window.
- Initiate a looping script.
- Use Browser to locate labels and scripts.

What is Scriptwriter Plus™?

Scriptwriter Plus provides you with the ability to design your own customized interface between the workstation and the Apollo® system. It allows you to standardize the business process, minimize user error and speed up the booking process.

It consists of two components:

- Scriptwriter Plus Build. This is a configurable option that allows you to build customized scripts.
- Scriptwriter Plus Run. This option allows the user to run (play) the scripts in Focalpoint that have been created in Scriptwriter Plus Build.

System Requirements

Hardware

The minimum hardware requirement is:

- Processor – 486DX2 or Pentium 90
- RAM – 16MB
- Hard Disk Space – 4MB
Software

The minimum software requirement is

- Windows 3.x or higher
- Focalpoint 3.0 or higher

Features and Benefits

The Scriptwriter Plus™ product offers many features and benefits. For example:

- Scripts offer a simple guided process for bookings.
- Scripts provide the ability to enforce agency standards, ensuring relevant data is incorporated into the booking.
- Scripts reduce keystrokes and ensure correct formats, including information for back office systems.
- It utilizes some of the Windows™ functionality, such as:
  - Drag and Drop
  - Cut and Paste
  - Multiple Document Interface (MDI)
  - Double Clicking
Notes
Module 2: Scriptwriter Plus™ Run

This module introduces Scriptwriter Plus™ Run and explains how to access and run scripts.

Module Objectives

Upon completion of this module you will be able to:

- Activate Scriptwriter Plus™ Run.
- Run a script from an icon.
- Run a script using the file name.
- Terminate a script.
- Minimize a script.
- Create, modify and delete group windows and icons.
- Save the setting on exit.

Activate Scriptwriter Plus Run

Scriptwriter Plus Run is activated from within the Focalpoint® application. There are several different ways of opening Scriptwriter Plus Run in order to run a script.

When to use

When you need to run any script built using the Scriptwriter Plus Build application.
How to use

Select Scriptwriter Plus™ Run by either:

- Entering ALT + R
- Choosing Scriptwriter Plus Run from the Focalpoint companions menu
- Clicking on the icon on the Focalpoint® Toolbar

The Scriptwriter Plus Run application appears along with any pre-configured groups and icons:

- The group title is CAR
- The script title is Car Availability

**Run a Script from an Icon**

When to use

When scripts are used frequently, icons will have been created so the script can be easily accessed.

How to use

1. Double-click CAR
   The Car Availability script is now running.
Run a Script Using File Name

When to use

When scripts are not used frequently, or while testing scripts, it is possible that an icon will not have been created. In this case, you must locate the script using the path.

How to use

To locate a script using the path, use the following steps.

1. Select File, Run.
2. Highlight AIR.KEY from the file list and click OK.

The running script window appears for you to complete.

Points to note

- The title bar of the script indicates the name of the script being executed including the extension.
- The cursor appears in the first field and progresses from one input field to the next by tabbing.
- There are prompts in the status bar at the bottom of the screen.
- The host entry is automatically transmitted after the final character of the arrival airport field is entered.
Minimize a Script

As with any other Windows® application, you can minimize scripts.

When to use

If you need to delay the running of a script or you need to switch to another application to check something.

How to use

You can use either the keyboard or mouse to minimize the Scriptwriter Plus Run window.

Mouse

- Point to the minimize icon in the top right hand corner of the script window.

Keyboard

1. Enter ALT + Spacebar
2. Enter n (the underlined letter for minimize)

Terminate a Script

Use the Terminate command to terminate the running of a script. This will close the running script window and end communication with the host. You can terminate a script at any point.

When to use

Only one script can communicate with the Galileo® system at a time so if you have more than one script open or when a running script is no longer required, you should stop it.
How to use

You can terminate a script using drop down menus, keyboard shortcut keys or standard Windows® 95 functions.

Using drop down menus
- Select Transaction, Terminate.

Using the keyboard shortcut keys
- Press ALT + T or ALT + F4.

Using standard Windows® 95 functions
- Click on the close icon in the top right hand corner of the running script.

Group Windows and Icons

Group windows function in Scriptwriter Plus™ in exactly the same way as group windows (Windows 3.XX) or folders (Windows '95), in that they collect together scripts of similar functionality. You must create them first in order to add icons which represent specific scripts.

This section covers:
- Creating a new group window.
- Adding an icon.
- Saving settings on exit.
- Modifying group windows and icons.
- Deleting group windows and icons.

Create a New Group Window

Scriptwriter Plus Run gives you the flexibility to create your own group windows. This can help you organize your scripts grouping together scripts with a common theme.
When to use

A large travel agency may want to have several group windows representing different clients or processes. Contained within the groups will be the icons that activate scripts specific to that client or process.

**Example:**

Window 1 = IBM company scripts  
Window 2 = Smith, Kline Beecham company scripts  
Window 3 = Price Waterhouse scripts

How to use

A group window will be created called AIR, then an icon to represent the Air Availability script.

1. Enter ALT + R, to ensure Scriptwriter Plus Run is active  
2. Select Window, New Window  
3. Enter the title for the new window, e.g. AIR  
   It is not necessary to complete the Host field. It will default to Apollo.  
4. Select OK  
5. Size the window as required using either the mouse or keyboard  
   **Mouse**  
   a. Point to a border or corner of the window. The pointer changes to a two-headed arrow.  
   b. Drag the corner or border until the window is the size you want.  
   c. Release the Mouse button.  
   **Keyboard**  
   a. Enter ALT+- (hyphen *not* underscore) to display document control-menu.  
   b. Select Size.  
   c. Use the cursor control keys to enlarge or reduce the size of the window.  
   d. Hit Enter when you have the size you want.

**Note:** There is *no* limit to the number of groups that you can create.
Add an Icon

Each icon represents a script that is run when the icon is selected. You can select which scripts to add to the group windows and configure an icon to represent the scripts.

When to use

After creating the group windows that will contain the icons, you can create the icons themselves. You should create icons for frequently used scripts.

How to use

To add the icon to activate the Air Availability script as demonstrated earlier, use the following steps.

1. Select File, Add Icon.
2. Enter a Description, e.g. Air Availability.
3. Enter the Script Path or use the Browse button to locate the script.
   The location of the scripts depends on the type of Focalpoint installation (i.e. network or local installation).
4. Select Choose Icon.
5. Highlight the name of the file representing the icon you wish to use, then select OK.
   For further information on icons, see Appendix B: Bitmaps.
6. Click on OK to add the icon to the window.
7. Select Window, then Tile, to fill the entire Scriptwriter Plus Run window with the two group windows.

Save Settings on Exit

You must save group windows and icons to retain the settings.

When to use

After creating groups and icons and before closing the Scriptwriter™ Plus Run application, check the Save Settings on Exit option otherwise they will be lost. The system does, however, default to the option being checked so it is only necessary if the option has been changed.
How to use

To Save the settings, use the following steps.
1. Select Window.
2. Ensure Save Setting on Exit has a check next to it.

Practice

Create two group windows and add icons as follows:
• Group window called AIR, add icon for AIR.KEY
• Group window called CAR, add icon for CAR.KEY

Modify Group Windows and Icons

It is possible to change the properties of group windows and icons after you have created them.

When to use

If you have created a group but want to change the name, or if you have created an icon and you need to alter its properties (e.g. path, icon in-use etc.).

How to use

To modify window properties:
1. Highlight the window you want to change.
2. From the Edit menu, choose Window Properties.
3. Enter the new title.
4. Click OK to save the changes.

To modify icon properties:
1. Highlight the icon you want to change.
2. From the Edit menu choose Icon Properties.
3. Amend the Description, Script Path or Icon as required.
4. Click OK to save the changes.
Practice

Change the icon used for the Air Availability script.

Delete Group Windows and Icons

It may be necessary to delete group windows and icons after you have created them.

When to use

- If group windows or icons have been created and they are no longer required.
- If an icon is to be moved from one group to the next, it is necessary to delete and recreate it.

How to use

It is necessary to delete any icons within a group, before you can delete the group itself.

To delete an icon:

1. Highlight the Car Availability icon.
2. Either press the keyboard delete key or select Remove Icon from the File drop down menu.

To delete a group window:

1. Highlight the CAR group.
2. Either press the keyboard delete key or select Delete Window from the Window drop down menu.
Module Review

Using the Course Book or online help, answer the following.

1. Which of the following icons represent the Scriptwriter Plus Run application?
   
   a. 
   
   b. 
   
   c. 
   
   d. 

2. What is the maximum number of group windows that may be created?

______________________________________________________________________________

3. It is possible to change the name of a group window after it has been created. Outline the steps to do this.

______________________________________________________________________________

______________________________________________________________________________

4. What is the default extension given to scripts?
   
   a. .DOC
   
   b. .CPR
   
   c. .VTF
   
   d. .KEY
5. The Window drop down menu appears as follows. Can you predict any problems the user may encounter?

<table>
<thead>
<tr>
<th>Window</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Window</td>
<td>Shift+Enter</td>
</tr>
<tr>
<td>Delete Window</td>
<td>Shift+F5</td>
</tr>
<tr>
<td>Cascade</td>
<td>Shift+F4</td>
</tr>
<tr>
<td>Tile</td>
<td></td>
</tr>
<tr>
<td>Arrange Icons</td>
<td></td>
</tr>
<tr>
<td>Save Settings on Exit</td>
<td></td>
</tr>
<tr>
<td>1 Ar</td>
<td></td>
</tr>
</tbody>
</table>

6. You would like your Scriptwriter Plus™ Run windows to appear as follows, i.e. one above the other. Which drop down menu and command would you choose to do this?

- Travel requirements
- Air

For Help, press F1
7. How many scripts can communicate with the Apollo® system at any one time?

______________________________________________________________________________

8. What is the keyboard shortcut key combination to terminate a script?

______________________________________________________________________________
Module 3: Scriptwriter Plus™ Build - Basics

You can build, modify, and save all scripts using the Scriptwriter Plus Build application.

A script can perform a complete procedure when run, although it is more likely that it will involve more than one script to complete the procedure. This is referred to as ‘Calling’.

This module covers planning and building basic scripts.

Module Objectives

Upon completion of this module you will be able to:

• Activate Scriptwriter Plus Build.
• Describe the window components.
• Plan a script using online help.
• Compile Input and Command windows.
• Utilize the Call command to link scripts together.

Activate Scriptwriter Plus™ Build

When to Use

You must activate Scriptwriter Plus Build when you want to start writing a new script or to check the functionality of an existing script.

How to Use

There are different ways to activate Scriptwriter Plus Build, depending on the version of Microsoft® Windows® you are using and on the Focalpoint® toolbar you have activated.

• From the Windows® 95 Start Button, select Programs, Focalpoint, Scriptwriter Plus, Scriptwriter Plus Build.
• From the Windows® 3.x Program Manager, select the Scriptwriter Plus Build icon from the Focalpoint group.

• From the Focalpoint Toolbar, Select the Scriptwriter Plus Build Icon

Window Components Overview

When Scriptwriter Plus™ Build is activated, an empty script appears which takes focus.

It is important to fully understand the components of the Scriptwriter Plus Build windows in order to efficiently build and amend scripts. Also when reporting problems you need to use the correct component terminology. The Scriptwriter Plus Build windows appear as follows:
<table>
<thead>
<tr>
<th>Callout</th>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Title bar</td>
<td>Bar at top of window that contains name of application and script that is currently active.</td>
</tr>
<tr>
<td>2</td>
<td>Menu bar</td>
<td>Bar containing available menus.</td>
</tr>
<tr>
<td>3</td>
<td>Tool bar</td>
<td>Bar containing standard Microsoft® Windows® icons for New, Open, Save, Cut, Copy, Paste and Print.</td>
</tr>
<tr>
<td>4</td>
<td>Input Window</td>
<td>Upper part of script window. What is created in this area appears to user when script is run.</td>
</tr>
<tr>
<td>5</td>
<td>Command Window</td>
<td>Lower part of script window in which you enter Apollo commands to direct and co-ordinate running of script.</td>
</tr>
<tr>
<td>6</td>
<td>Script border</td>
<td>Within Input window, script border represents actual size of script when it is run.</td>
</tr>
<tr>
<td>7</td>
<td>Toolbox</td>
<td>Toolbox consists of six icons, each representing a type of control. Using mouse, these can be selected and dragged into Input window to create a new field. Six icons are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><img src="Image" alt="Text Control" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td><img src="Image" alt="Edit Control" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td><img src="Image" alt="Radio button Control" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td><img src="Image" alt="Checkbox Control" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td><img src="Image" alt="Frame Control" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td><img src="Image" alt="Bitmap Control" /></td>
</tr>
<tr>
<td>8</td>
<td>Split bar</td>
<td>Horizontal line dividing Input and Command windows. Placing mouse on split bar and dragging it up or down changes proportions of Input and Command windows.</td>
</tr>
<tr>
<td>9</td>
<td>Command bar</td>
<td>Row of buttons situated between Input and Command windows. Each button represents a command that can be used within Command window to control functionality of script. These are also available from Command menu option.</td>
</tr>
<tr>
<td>10</td>
<td>Browser</td>
<td>Tool within Scriptwriter Build that enables you to search and retrieve scripts and labels by specified criteria.</td>
</tr>
<tr>
<td>11</td>
<td>Status Bar</td>
<td>Provides contextual prompt help, displays name of currently active label and gives screen co-ordinates of cursor position.</td>
</tr>
</tbody>
</table>
Plan a Script Using Online Help

Assume the script to be built will check for basic air availability and then sell a seat from that availability.

When to use

Before starting to build scripts, it is essential to plan the script to account for every eventuality. If this is not done and the full implications of writing a script are not recognized, a script which you estimate will take three days to write, could take up to three weeks to write.

How to use

The Scriptwriter Plus™ Build online help has all the prompts needed to plan scripts, so we will refer to this by selecting Help, Contents, Overview, Planning the script.

<table>
<thead>
<tr>
<th>Suggestion from online help:</th>
<th>Possible answer:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Decide exactly what you want the script to do.</td>
<td>Display air availability and sell a seat(s).</td>
</tr>
<tr>
<td>2. Partition functionality of procedure into logical components so that each can be represented by one script. This should give an indication of total number of scripts required to carry out whole procedure and functionality of each.</td>
<td>Two separate scripts will be required. One for availability and one for sell.</td>
</tr>
<tr>
<td>3. Check whether you have any existing scripts that could be used. To find a particular script, use Browser window. If you have a script that is similar to that required, this can be opened and saved with a new name using the Save As command. This script can now be changed.</td>
<td>As yet, no scripts have been written that can be reused. This will come later.</td>
</tr>
<tr>
<td>4. If more than one script is to be used to perform a procedure, you need to establish the &quot;Call&quot; hierarchy of the scripts, i.e. which script is to &quot;Call&quot; another. This helps you to create scripts in a logical order and avoids confusion.</td>
<td>We will create 2 separate scripts first, then come back to this.</td>
</tr>
</tbody>
</table>
### Suggestion from online help:

5. Prepare the details of each script taking into consideration:
   - What is the host entry to be transmitted to the host?
   - What information do you require from the user for the host entry? This will give an indication of what labels are required.
   - Does the user response field require an Edit control, Radio button or Checkbox?
   - Do you want to group any labels within a Frame? (Related radio buttons should be grouped together within a Frame)
   - Do you want to insert a bitmap?

### Possible answer:

- A for availability scripts and 0 for sell script.
- Availability – Date, departure and destination points.
- Sell – How many seats, class, line number
- Edit controls.
- Yes, Text and Edit controls will be grouped together for a more professional appearance.
- Yes.

### Exercise – Use Scriptwriter Plus™ Build Online Help

Use the Scriptwriter Plus Build online help to answer the following:

1. The Scriptwriter Plus Build help Contents page consists of blue boxes which represent sections of Scriptwriter Plus Build help. These are grouped together by subject range. What are the titles of three of the blue boxes?

   ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________

2. In the topic ‘Features of SW+ Build’, there are four paragraphs, what are the titles of the four paragraphs?

   ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________

3. In the topic ‘Overview of SW+ Build’, you are prompted to view another screen via the ‘Click here to view’ link. What is the name of the script used to illustrate the Scriptwriter Plus Build window components when using this link?

   ____________________________________________________________
4. The Help suggests three ways to exit Scriptwriter Plus Build using the keyboard. What are the three ways?

______________________________________________________________________________

______________________________________________________________________________

5. A three-step procedure is recommended to save a script for the first time. What is the first step of the recommended procedure?

______________________________________________________________________________

Input Window

The Input window displays information and response fields to the user when the script is run. The design and layout of the Input window are very important to enable the user to complete it easily.

This section covers:

- Changing the background color and font of the Input window.
- Sizing the script border.
- Including a Frame.
- Including Text controls.
- Including Edit controls and labels.
- Including Bitmaps.

Change Background Color and Font of Input Window

The Input window of each individual script can have a different color and font if required.
When to use

You may want to change the default color and font when you are creating scripts with a continuous theme. This will allow them to be identified quickly.

When you first activate Scriptwriter Plus, the Input window appears.

How to use

To change background color:
1. Select Window
2. Highlight Default Color
3. Select Silver

To change font:
1. Select Window, Default font
The Font dialog box appears.

2. Enter the font details as specified in Appendix A: Style Guidelines; ariel, regular, 8, black

Points to note

- Chosen colors and fonts will be remembered. They are stored within whichever Focalpoint user personalization is being used.
- Only newly created controls will adopt the new default color and font.
- The chosen colors and fonts can be included in a template script. This is covered later in the course.

Size Script Border

The script border is represented by a dotted rectangle in the Input window and determines the size of the script window when run.

When to use

The size of the script border should be determined before creating any other controls. All controls must be placed inside the script border, so the script must be well planned before this stage is reached. It can be resized at any time, but cannot be deleted.
How to use

You can size the script border using either the mouse or keyboard.

**Mouse**
1. Point to the bottom right hand corner of the script border. The pointer changes to a two-headed arrow.
2. Drag the corner until the script border is the size you want (e.g. H15, W74).
3. Release the Mouse button.

**Keyboard**
1. Ensure the cursor is inside the script border.
2. Enter ALT + E, S.
3. Use the cursor control keys to enlarge or reduce the size of the window.
4. Hit Enter when you have the size you want.

Points to note

- The size of the script border appears the right of the Status bar:
  ![Height 22 Width 53]
- The size, in order that the Input window entirely covers an un-zoomed Focalpoint window, should be H15, W74.
- By default the script border is H12, W45. It is possible to make it appear smaller than this but when you actually run the script it will appear as the default minimum size. This is so the status bar can display prompt help and OVR/CAPS/NUM settings.

Include a Frame

You can include Frames in the Input window and can configure them to suit your needs. It is possible to give them a title and you can change the style if you need to.
When to use

There are two main reasons for using a Frame:

- To group together related controls.
  
  You can see an example of this when you view the print dialog box:
  
  - Select File, Print.

  The following dialog box appears with a Frame surrounding the options.

  ![Frame Example](image)

- For appearance to ensure that when running the script it is easy for the user to complete and also for a more professional look.

How to use

To include a Frame:

1. Click the Frame icon in the Toolbox.

2. Drag and drop the icon into the Input window aligning the top left corner of the Frame in the correct position.

3. Size the Frame using either the mouse or keyboard.

   **Mouse**

   a. Point to the bottom right hand corner of the Frame. The pointer changes to a two-headed arrow.
   b. Drag the corner until the Frame is the size you want.
   c. Release the Mouse button.
Keyboard

a. Enter ALT + C, S.
b. Use the cursor control keys to enlarge or reduce the size of the Frame.
c. Hit Enter when you have the size you want.

4. Double click on the highlighted Frame to display the Configure Frame dialog box:

5. Enter ‘Please complete the following…’ in the Text field.

6. Select Style.
   This needs to be done as the Frame text does not adhere to the font defaults set earlier.

7. Enter the font details as specified in Appendix A: Style Guidelines; ariel, 8, italic, maroon.

8. Select OK twice.

Points to note

- You should adjust the Frame to the correct size at this stage. Altering the size of the Frame after introducing other controls can cause problems when trying to highlight individual controls for amendment.
- Click outside the Frame so it isn’t highlighted to see the correct colour.
- It is possible to just hit the spacebar in the Frame Attributes, Text box to create a frame without any title. In Scriptwriter Plus Run, the gap which appears in Build will not show.
Include Text Controls

You can use text controls to display text to the user that is usually information or questions they have to respond to.

When to use

After planning a script you will be aware of the questions you need to ask the user. You can use Text controls to ask these questions and also to include a script title if required.

How to use

To include a Text control as a script title, use the following steps.

1. Click the Text control icon in the Toolbox:
2. Drag and drop the icon to the required position in the Input window.
3. Double click the control.

   The Configure Text Control dialog box appears.

4. Enter ‘Air Availability’ in the Text field.
5. Select Style.

   Note: You need to do this as the Text control for a script title does not adhere to the font defaults set earlier.
6. Enter the font details as specified in Appendix A: Style Guidelines; ariel, 12, black, bold, underline.
7. Click OK twice.
Points to note

- The Label Name field should not be completed in the Configure Text Control dialog box.
- You can also type directly at the cursor position in the Input window to create a new Text control.
- You can activate the ‘align with grid’ function if required by selecting Options, then Align with Grid.
- CTRL + M in conjunction with the keyboard arrow keys will move a Text control, and any other controls, within the script border.

Practice

Create three Text controls to represent the questions to be asked as discussed when planning the script. The completed Input window should appear as follows:
Include Edit Controls and Labels

Edit controls provide a response field for the user to enter the information requested. This information is used to compile the host entry for transmission to the Apollo® system. Labels manage all information passing through an Edit control.

When to use

Whenever you require information from the person executing the script, you must create these storage areas to enter and retain the data.

It is possible to control the type of information the user enters by the use of labels (e.g. alphas or numerics) and the attributes by the use of Edit controls.

How to use

To include an Edit control and label to accept a date, use the following steps.

1. Click the Edit control icon in the Toolbox:
2. Drag and drop the icon to the required position in the Input window.
3. Double click the control.

The Configure Edit Control dialog box appears.
4. Click the Labels button to create the first label for this script. A labels list appears.

4. Click the New button to display the Configure Label dialog box.

5. Complete the Label dialog box as follows:

<table>
<thead>
<tr>
<th>Callout</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Label Name</td>
<td>Mandatory field. Unique name for label (maximum 30 characters). Label names must start with an alphabetic character and subsequent characters may be alphanumeric, numeric, underscore or dollar symbol. When a label name appears in a script, it appears prefixed with script name, followed by # symbol, e.g. CAR_SELL.KEY#vendor.</td>
</tr>
<tr>
<td>2</td>
<td>Input Mask</td>
<td>Mandatory field. This defines format that user’s entry must comply with. Maximum number of characters (without spaces) in input mask is 69. You can use a combination of characters except C which is singular. See separate table for more details.</td>
</tr>
<tr>
<td>Callout</td>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>3</td>
<td>Default Value</td>
<td>Optional field. Use to specify a default value for label. Value entered appears in label when script is run, but can be overridden by user.</td>
</tr>
<tr>
<td>4</td>
<td>Validation Required and Strict Validation</td>
<td>Used to ensure user’s response matches an assigned value. For detailed information, see Module 4.</td>
</tr>
<tr>
<td>5</td>
<td>Global</td>
<td>Optional field. <strong>Global labels</strong> When checked, label is designated as a global. You can use global labels in several different scripts. They also remember what is stored in response field when script is run and display that same data in other scripts using same global label. <strong>Local labels</strong> When <em>not</em> checked, label is designated as local. A local label can only be used by script in which it is created. Data entered into a local label is <em>not</em> retained.</td>
</tr>
</tbody>
</table>

**Input Mask values:**

<table>
<thead>
<tr>
<th>Input Mask</th>
<th>Type</th>
<th>Characters permitted at run time</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Alphabetic</td>
<td>Letter of the alphabet <em>only</em>.</td>
</tr>
<tr>
<td>9</td>
<td>Numeric</td>
<td>Numerals 0-9 are accepted, as well as negative sign (-), decimal point (.) and spaces.</td>
</tr>
<tr>
<td>.</td>
<td>Numeric</td>
<td>Decimal point can be included for arithmetic functions.</td>
</tr>
<tr>
<td>X</td>
<td>Alpha-numeric</td>
<td>Label will accept any character.</td>
</tr>
<tr>
<td>C</td>
<td>Conditional</td>
<td>Usually attached to a Radio button or Checkbox to store true or false values. Can also be used with Edit controls.</td>
</tr>
</tbody>
</table>

7. Click OK when the label details have been completed to return to the SCRIPT1.KEY – Labels list.

8. Ensure the Date label is highlighted, then select Use, to reference the date label in the Configure Edit Control dialog box.

The Labels dialog box will remain on the screen until closed.
The Configure Edit Control dialog box now appears as follows:

The following table explains the use of the control attributes.

<table>
<thead>
<tr>
<th>Callout</th>
<th>Control Attribute</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Must Fill</td>
<td>Choose if user is to enter same number of characters as specified in Input Mask. For example, if Input Mask is AAA, user must enter 3 alphabetic characters.</td>
</tr>
<tr>
<td>2</td>
<td>Must Enter</td>
<td>Choose if user is to enter at least one character in control.</td>
</tr>
<tr>
<td>3</td>
<td>Auto-Transmit</td>
<td>Choose when user has entered all required information. Data will then be used to compile Apollo entry.</td>
</tr>
<tr>
<td>4</td>
<td>Style</td>
<td>Can be used to change font and color of text.</td>
</tr>
</tbody>
</table>

9. Click OK as no attributes are required for the date Edit control.

Practice

Create Edit controls and associated labels for departure and destination airports. Configure both labels as global.
Include Bitmaps

You can include Bitmaps in the Input window to improve the appearance of the script. They do not serve any practical function. The bitmaps used are the same as those used in Scriptwriter Plus Run, so you can obtain a common theme by using the same bitmap in the two applications.

When to use

When you want to add a graphic to the input window.

How to use

To include a Bitmap in an Input window, use the following steps.

1. Click the Bitmap icon in the Toolbox:
2. Drag and drop the icon to the required position in the Input window.
3. Size the bitmap using either the mouse or keyboard. Be careful not to make it too big as it may appear distorted.
4. Double click on the highlighted bitmap to display the Select Bitmap File dialog box.
   
   **Note:** For details on available bitmaps, see Appendix B.
5. Highlight the required bitmap file name and select OK.

Summary: Create an Input Window

When creating an Input window, use the following suggested steps.

1. Plan the script to determine how many scripts will be involved, what questions need to be asked and what the correct Apollo entries are. Use the online help as a prompt.
2. Change the background color and font to reflect requirements or company policy (Only needs to be done initially).
3. Size the script border so it will accommodate any Text, Edit, Frame and Bitmap controls.
4. Include Frame controls.
5. Include Text controls.
6. Include Edit controls and create and attach relevant labels.
7. Include a Bitmap control.

Command Window

The Command window is the partition of the Scriptwriter Plus Build window that controls and co-ordinates the running of the script. When the script is run, the users' inputs transfer to the Command window where they change into a host format and transmit to the Apollo® system. The details entered in the Command window are not visible when the script is run.

This section covers:
- Entering a host table command e.g. SOMs, Transmits, Clear screen.
- Entering a command e.g. Show Window, Call, If.
- Entering a host format and label e.g. HOA, P:CLER/440 123-4576.
- Saving a script.

Enter a Host Table Command

When compiling the Apollo entry, you need ensure the entry transmits correctly. The host table in a script coordinates the transmission of information to the Apollo® system.

When to use

When you need to enter start of message commands, transmits, redirect responses to other windows etc.

How to use

We will first enter a command to clear the screen so any existing data in the Focalpoint windows is removed.

To do this, use the following steps:
1. Position the cursor in the Command window.
2. Select Host from the Command Bar.
The host table appears.

Table description:

<table>
<thead>
<tr>
<th>Column Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td>The wording that appears in the Command window.</td>
</tr>
<tr>
<td>Key</td>
<td>The ‘Shortcut’ key combination which can be entered directly in the Command window eliminating the need to activate the host table.</td>
</tr>
</tbody>
</table>

3. Highlight ‘Clear all windows in session’.

4. Select OK.

The Command window will now appear as follows:

The host table will be used in future to enter SOMs and Transmits.
Points to note

- The Shortcut key combinations are only available after the Host table has appeared at least once while creating a script. This establishes the host being used and is known as keyboard mapping.

Enter a Command

Commands control the functionality of the script. The common commands appear on the Command Bar located between the Input and Command windows and also in the Command menu.

When to use

When you want to enter commands in the Command window. Examples of Commands are Show Window, Call and If.

You will enter the Show Window command into the Command window of the Air script. The Show Window command determines when the Input window appears to the user when the script is run.

How to use

To enter a Show Window command, use the following steps.

1. Ensure the cursor is on the line after the ‘Clear all windows in session’ Host table command.
2. Select Show from the Command bar.

The Command window appears.
Enter a Host Format and Label

The host format is the entry that instructs the Apollo® system on the function it should perform. It has a set order that must be adhered to and can consist of fixed and variable information.

**Fixed information** is the part of the entry that is always the same.

**Variable information** is part of the entry that can change depending on the requirements of the user.

For example in the entry A12JANJFKLHR

<table>
<thead>
<tr>
<th>Availability format</th>
<th>Type of information</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Fixed</td>
</tr>
<tr>
<td>12JAN</td>
<td>Variable</td>
</tr>
<tr>
<td>JFK</td>
<td>Variable</td>
</tr>
<tr>
<td>LHR</td>
<td>Variable</td>
</tr>
</tbody>
</table>

**When to use**

Whenever you need to compile a host entry to be transmitted to the Apollo® system.

**How to use**

*Fixed* information is typed directly into the Command window and displays in blue.

*Variable* information consists of the labels created for the Input window. To include the variable information in the Command window, you must reference the labels that were created when compiling the Input window.

To enter the availability entry into the Command window, use the following steps.

1. Before entering a host format, input a SOM from the host table.
2. Enter ‘A’ for an availability request.
4. Drag and drop the date label to the correct position in the Command window.

**Note:** Use the + to correctly position the label in the Command window. Do not use the mouse pointer or the label graphic.
5. Drag and drop the departure and destination labels in the same manner.
6. Select Host from the Command bar, then RD2 to redirect the response to Window 2.

The Command window appears.

<table>
<thead>
<tr>
<th>Host</th>
<th>While</th>
<th>Show</th>
<th>Call</th>
<th>Assign</th>
<th>Terminate</th>
<th>Comment</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetForm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ShowWindow</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Box 'a' Script1#date Script1#departure Script1#arrival Rd2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Points to note

- It is possible to drag the label from the Edit control in the Input window to the Command window. This is fine if you have a steady hand. If not, you could move the Edit control out of position.
- The maximum number of lines in a Command window should be 300-350. If more than this there may be PC memory problems and the script may not run.

Save a Script

Once you have created the Input and Command windows, you must save the script.

When to use

It is necessary to save and name a script before you can test it in Scriptwriter Plus Run. We recommend, however, that you save the script on a regular basis while building it.
How to use

To save a script, use the following steps.

1. Select File, Save As

   The Save As dialog box appears.

   ![Save As Dialog Box]

Points to note

- The script name has a maximum of 8 characters (even if using Windows 95).
- The extension automatically appended is .KEY however it is useful to only name the ‘master’ script .KEY. This is so it will automatically display when using Scriptwriter Plus Run. You could give other scripts which are of the same ‘suite’ of scripts a different extension e.g. AIR.T1, AIR.T2, AIR.T3 etc.
- The default directory for storing scripts if not using user personalization is \datadir\swplus\scripts. If logging on to Focalpoint using a user name the default directory will be \datadir\users\username\. 

2. Enter AIR, followed by your initials in the File name field, e.g. AIRAP, then select OK.
The Author Page Information dialog box appears. Here you can add additional information, e.g. a Script Title and a Description.

**Points to note**
- The author name is *only* recorded on the Author Page if the user has logged on to Focalpoint® using their name. *If no* name has been entered the field is left blank.
- The Amendment Information displays the date, time, author and reason for the amendment. This is automatically logged each time a script is saved.
- For further information on the Author page refer to online Help, Contents, General Procedures, Viewing the script Author Page.

3. Click OK.

**Practice**

Create a script to sell a seat from the displayed availability.

The entry to sell from availability is:

```
01J4
```
- 1 = Number of passengers
- J = Class of service
- 4 = Line number in the availability display
Call Command

You can use the Call command in a script to execute other scripts. Since building scripts is a modular process, the Call command provides a way to link scripts together.

When to Use

In our case the scripts that have been built (one for air availability and one for air sell), naturally follow on from each other. The air sell script logically follows the air availability script so the availability script will ‘call’ the sell script. The script containing the call command is known as the ‘calling’ script. The script that is specified to run is known as the ‘called’ script.

How to Use

To insert a Call command in the Command window of the air availability script:

1. Open AIR.KEY
2. Place the cursor on the line after the air availability entry
3. Select Call from the Command bar
4. Highlight SELL.KEY
5. Select OK
6. Select Save icon from the Scriptwriter Plus Build toolbar
Call Command Hierarchy

When scripts are called, you must be aware of the use of the Call command as this affects the PC Memory and ultimately the performance of the script.

When a ‘called’ script has finished running, it will naturally return control to the ‘calling’ script:

```
Script 1
Line 1
Line 2
Line 3
CALL SCRIPT 2
Line 4
Line 5
```

Script 2

Line 1
Line 2
Line 3

The order in which the lines will be executed in the previous example is:

```
SCRIPT1 Line 1
SCRIPT1 Line 2
SCRIPT1 Line 3
SCRIPT1 Line 4
SCRIPT1 Line 5
SCRIPT2 Line 1
SCRIPT2 Line 2
SCRIPT2 Line 3
```

Before script 2 has finished executing there will be 2 scripts in the PC’s memory. If script 2 then called in another script; script 3, then script 3 called in another script; scripts 4, the memory will start to fill rapidly.
The following are examples of a well planned script and a poorly planned script.

Example: well planned script

```
Script 1
CALL SCRIPT 2
CALL SCRIPT 3
CALL SCRIPT 4
```

There are *no* more than two scripts in the PC’s memory at any one time. After script 2 has finished running, it is cleared from memory and Scriptwriter Plus automatically returns to the calling script from the point it left. Script 3 will then be called and executed, again being cleared from memory when it has finished running. Finally script 1 calls script 4. After script 4 has finished running, although Scriptwriter Plus will return to script 1 there are no more commands to be carried out, so the procedure is complete.

Example: poorly planned script

```
Script 1
CALL SCRIPT 2
Script 2
CALL SCRIPT 3
Script 3
CALL SCRIPT 4
```

At the final stage, i.e. when script 3 calls script 4, there are 4 scripts in the PC’s memory. Depending on the size of the individual scripts this could cause problems with system performance. You should try to avoid creating a series of scripts that involve more than five levels of hierarchy.

Practice

Insert the Call command into your Air Availability script to bring in the Sell script. Test that it is working satisfactorily.
Exercise – Car Script

Create a script(s) entitled ‘Car Reservation’. It will enable the user to see an availability display, then sell from this display.

In Scriptwriter Plus™ Run, create a group window called Cars and add an icon to represent your script.

1. To enable the sell to take place, there must be a name present. The Apollo entry to do this is:
   \[ \text{N:1SURNAME/FIRSTNAME} \]

2. The Apollo® system entry to check a car availability is:
   \[ \text{CAL12DEC-15DECAMS/ARR-9A/DT-2P} \]

3. The Apollo® system entry to sell from an availability is:
   \[ \text{OA1} \]

Points to note

- All mandatory parts of the entry are in **bold**, all others are variable and up to the user to specify.
- Availability entry:
  - \text{CAL} is the Low to High rate car availability entry
  - \text{ARR} is the Arrival time/pick up time
  - \text{DT} is the drop off time
- Sell entry
  - \text{A} = Column
  - \text{1} = Row
Module Review

Using the course book or online help, answer the following.

1. Which of the following Icons in the Toolbox represents a Text control?
   a. 
   b. 
   c. 
   d. 

2. Can a label be attached to a Text control?

3. Which drop down menu allows you to change the color of the Input window?

4. What is the recommended maximum number of lines that the Command window should not exceed and why?

5. What is the maximum number of characters allowed for a label name?

6. What is the maximum number of characters allowed in an Input mask?

7. What is the difference between a Global and a Local label?

8. What does the ‘split’ option allow you to do?

9. When the Focalpoint® log on screen does not contain a user name, which drive and directory are scripts saved in?
10. When the Focalpoint® log on screen _does_ contain a user name, which drive and directory are scripts saved in?

______________________________________________________________________________

11. What is the shortcut key to enter a zoom command in the Command window?

______________________________________________________________________________

12. What two things are wrong with the Command window of the following script that is written to display air availability?

<table>
<thead>
<tr>
<th>If</th>
<th>While</th>
<th>Show</th>
<th>Call</th>
<th>Assign</th>
<th>Terminate</th>
<th>Comment</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClrScr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Som &quot;a&quot;</td>
<td>AIR.KEY#departure</td>
<td>AIR.KEY#arrive</td>
<td>AIR.KEY#date</td>
<td>Rd2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

______________________________________________________________________________

______________________________________________________________________________

13. What determines the size of the Input window?

______________________________________________________________________________

14. What input mask would you use for an airline code?

______________________________________________________________________________
Module 4: Scriptwriter Plus™ Build – Beyond Basics

Module 3 covered the basic Scriptwriter Plus™ Build commands. There are many others commands available to make full use of the Scriptwriter Plus product.

Module Objectives

Upon completion of this module you will be able to:

• Apply Conditional logic when building a script.
• Build a template script.
• Utilize Browser to locate labels and scripts.
• Validate labels.
• Delete labels.
• Include script builders’ comments.
• Insert a Pause command and Painted message in a script.
• Utilize nested Conditional logic commands.
• Include While and Assign commands in a script.
• Program a looping script so the impact on the PCs memory resources is minimal.

Conditional Logic

Use conditional logic when a script can follow more than one path and it evaluates whether a statement is true or false. It is possible to utilize Scriptwriter Plus functions to determine which path to take, such as:

• Radio buttons and Checkboxes in the Input window
• If and While statements in the Command window
When to Use

When the person running the script must choose from a series of options. For example, which type of fare is required? Is there any preferred air vendor? etc.

All these options have to be evaluated by the script builder and taken into account when writing the script. This is done with Conditional logic.

How to Use

Conditional logic in Scriptwriter Plus™ Run

Run AIR.KEY

The first script checks for air availability. The second script then appears asking if more availability is required. Radio Buttons have been used to evaluate the user’s reply.

Run the script several times selecting different options.

Conditional logic in Scriptwriter Plus™ Build

1. Open AIR.KEY
   This script calls in another script - MOREAIR.KEY
2. Open MOREAIR.KEY
   The following Input and Command windows appear.

   The Input window with two Radio buttons:

   ![Input window with two Radio buttons]

   The Input window has been created using a Frame and two Radio buttons. The Frame is included as described in the previous module and is necessary, in this case, to group together the Radio buttons.
Command window with one If statement:

<table>
<thead>
<tr>
<th>If</th>
<th>While</th>
<th>Show</th>
<th>Call</th>
<th>Assign</th>
<th>Terminate</th>
<th>Comment</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShowWindow</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>If ( MOREAIR.KEY/general )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show &quot;A*&quot; End</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Else</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Call ( MOREAIR2.KEY )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Endif</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Explanation of the above logic:

Use the ‘If’ statement in conjunction with the general label. There are two scenarios:

- The user chooses the Radio button that the general label is attached to. The general label will therefore be positive i.e. chosen and the If statement will evaluate true. The commands following the If condition but before the Else statement will be carried out i.e. A*.

- The user does not choose the Radio button that the general label is attached to. The first part of the If statement will evaluate false and all the instructions between the If and Else statements will be ignored. The commands after the Else statement will be carried out. In this case as only two choices are offered to the user, if the general label is not chosen then the Carrier Specific label must be chosen. A third script (MOREAIR2.KEY) is then called that enables the user to specify a carrier.

Include Radio Button in Input Window

Radio buttons are a way in which you can incorporate Conditional logic into a script to give the user choices.

When to use

When you want to give the user several options from which they must select only one.
How to use

To include a Radio button:

1. Click the Radio button Icon in the Toolbox:

2. Drag and drop the icon to the required position within the Frame.

3. Double click the control to display the ‘Configure Radio Button’ dialogue box and enter the text to accompany the Radio button as follows:

4. Select the Labels button and attach the ‘general’ label to the Radio button control.

   The label has been created using a C input mask.

   Note: For details on the C input mask, see Module 3.

5. Repeat steps 1-4 to configure the ‘View a Carrier Specific Display?’ Radio button.
Include If Statement in Command Window

Use the If command to enter a conditional expression in the Command window.

When to use

To determine what actions should be performed depending on the choice of the user.

How to use

To enter a conditional If statement:
1. Position the cursor after the Show Window command.
2. Select IF from the Command Bar.
   The Enter expression box appears.
3. Select the Labels button and use the ‘general’ label from the labels list.
   Note: Double-clicking the label in the labels list is an alternate way to selecting ‘use’.
4. Click OK to exit the Enter Expression dialogue box.
   The conditional expression will appear in the Command window.
5. Use the keyboard return key to create a space between the If and Else commands.
7. Use the keyboard return key to create a space between the Else and Endif commands.
8. Enter a Call command to call the MOREAIR2.KEY script. The MOREAIR2.KEY script asks the user to enter the carrier required.

Points to note

- The Else statement is automatically entered as part of the command. You can deleted it if required, but once deleted, it you cannot be re-insert it.

Example: Conditional Logic Using Three Radio Buttons

If there had been three choices in the example we have just looked at, then the Command window would have to account for this.

The Input window would appears as follows:

![Input window](image)

It would not be possible to use the Else statement in this case, as if the first option is not chosen, then there is not one, but two other possibilities.

The Command window would appear as follows.

```
If { MOREAIR KEY=general }  
  Call "MOREAIR KEY"
Endi:
If { MOREAIR KEY=Carrier_specific }  
  Call ( MOREAIR2 KEY )
Endi:
If { MOREAIR KEY=seat }  
  Call ( SEAT KEY )
Endi:
```

Separate If, Endif statements account for each of the 3 possibilities. The Else statements have been removed but this is not mandatory, they can remain and not affect the running of the script in any way.
Example: Conditional Logic with an Exact Text Match

This example demonstrates how you can use Conditional logic to determine which path to follow depending on an exact text match. The following example checks for specific city codes.

The Input window would appear as follows:

![Air Availability Window]

The Command window would appear as follows:

```plaintext
BEGIN IF
   IF AIR.KEY#destcity = "BKK" OR AIR.KEY#destcity = "SIN"
   THEN CALL ( HOTEL.KEY )
   ELSE CALL ( BOOK.KEY )
   ENDIF
END IF
```

After an air availability appears, Conditional logic is introduced to determine what happens next depending on the destination city entered by the user.

If the label AIR.KEY#destcity is either BKK or SIN, the expression will evaluate as true and the script HOTEL.KEY is called and run. If the label holds another value, it will evaluate false and the BOOK.KEY is called.

The Expression dialogue box, is completed as follows:

![Enter Expression Dialogue Box]

```plaintext
AIR.KEY#destcity = "BKK" OR AIR.KEY#destcity = "SIN"
```
Points to note

- Type all text, with the exception of labels, directly into the box.
- City codes are in quotation marks indicating that an exact text match is required. The text within the speech marks is case sensitive.
- The label name (AIR.KEY#destcity) is specified in the conditional expression twice, once for each text match.
- For more information on Conditional logic, see ‘Entering Conditional Expressions’ in online help.
- The logical operator OR has been used. Other examples of logical operators are:

<table>
<thead>
<tr>
<th>Logical Operator</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND</td>
<td>Evaluates true if expression to left and right are both true.</td>
</tr>
<tr>
<td>NOT</td>
<td>Evaluates true if value of expression to right is false.</td>
</tr>
<tr>
<td>AND NOT</td>
<td>Compares expressions and evaluates true if these are not equal values.</td>
</tr>
<tr>
<td>OR</td>
<td>Evaluates true if either operand on its left or right is true.</td>
</tr>
<tr>
<td>OR NOT</td>
<td>Compares expression and evaluates true if these are not equal values.</td>
</tr>
</tbody>
</table>

Practice

Create a MORECAR script and incorporate this in the Car script created in Car Script Exercise. It will enable the user to select to view either weekly rates, more availability/vendors or go straight to the Car sell script.

The Apollo® entry to view weekly rates is: **CAU-W**
You will need to use a minimum date range of five days when testing.

The Apollo® entry to view more car types/vendors is: **CAL*PD**
Include a Checkbox

Checkboxes are controls that enable you to give the user a list of options from which any number may be selected.

When to use

Use when multiple choices may be made. Checkboxes are not mutually exclusive, as is the case with Radio buttons. They provide the user complete freedom to choose none, some or all. You could, for example, have a number of Checkboxes representing different unassociated remarks available for use when creating an itinerary and select which ones to include.

How to use

To include a Checkbox in an Input window, use the following steps.

1. Click the Checkbox icon in the Toolbox:
2. Drag and drop the icon to the required position in the Input window.
3. Double click the control.
   - The Configure Checkbox dialogue box appears.

4. Enter the text that will appear next to the Checkbox in the Checkbox Attributes, Text field.
5. Create a label to attach to the Checkbox, using the Labels button.

Points to note

- The labels attached to a checkbox have a Conditional input mask as with Radio buttons.
- As with any other control, you can change the style.

Exercise – Timatic Script

Create a script(s) entitled ‘Timatic’. It will enable the user to check Visa and/or Health requirements.

In Scriptwriter Plus™ Run, create a group window called Timatic and add an icon to represent your script.

The Apollo® system direct request entries are:

- **TI-RV/AAAAA/DEBBB** (to check Visa requirements)
- **TI-RH/EMAAA/DEBBB** (to check Health requirements)

Points to note

- AAA and BBB represent city codes.
- **NA** = Nationality, **DE** = Destination, **EM** = Embarkation
- All mandatory parts of the entries are in bold, all others are variable and up to the user to specify.
- For the visa information, assume *only* a Nationality and Destination are required.
- For the health information assume *only* an embarkation point and Destination are required.
- Either country or city codes *can* be used in the entries. It is more likely, however, that a user will know a city code than a country code.
Build a Template Script

A template script incorporates controls that have the style and design you want all your scripts to follow. It ensures all the scripts have the same look and feel. It is a normal script that you can open and rename.

When to use

When starting to write a new script, if you have a template script you can use this as a starting block.

How to use

Create a script and build your Input window using Frames, Edit and Text controls in the fonts, colors and general style you want all your scripts to adopt.

When you wish to create a new script, open the template script and immediately save it as another name. The template will remain intact for your next script. All you will need to do is configure the individual controls.

Points to note

- Do not include any Global Labels in the Template script as Scripts should never be ‘Saved As...’ when they contain global labels. This is due to the re-referencing of the script name in existing scripts that call in the original.

Practice

Create a Template script with your chosen style. Include at least one of each of the following:

- Frame
- Edit control
- Text control
- Bitmap of your preferred size
- SOM and Show Window command in the Command window
**Browser**

The Browser is a separate window that remains active to help you find and manage your library of scripts. It allows you to:

- View all labels used in the script.
- View host commands in the script.
- View the script call/calling hierarchy.
- Open a script.
- Copy labels to another script.
- Print a script.
- Search and retrieve scripts by specified criteria.

**When to Use**

Use a Browser when you want to confirm which labels have been used in a script or if you want to check which scripts call a particular script.

**How to Use**

You can have the Browser window open while working with a script. Because you *cannot* close the Browser window, you can minimize it to an icon when you *no longer* need it open.

1. Open the Browser by double-clicking the minimized Browser Title bar.

The Browser window will appear.
2. Click the Search Criteria button.
3. Edit the search path to select the drive and directories you want to search. Add or remove paths as required.
4. Click OK to return to the Browser window.
5. Select the Do Search Now button.

   Browser will search for all scripts in the paths you specified. The path and name of all matching scripts are displayed in the Script Name field.
6. Highlight the required script.
7. Select one of the following viewing options:
   - **Labels**
     To display all labels associated with the script. The label name and whether it is a global label or local label appears. You can then ‘export’ the label into another script.
   - **Host Commands**
     To display all Apollo® entries in the selected script.
   - **Call Hierarchy (Down)**
     To display all scripts that are ‘called’ by the selected script until you reach the lowest script on the call hierarchy.
   - **Call Hierarchy (Up)**
     To display all scripts that ‘call’ the selected script. Each level of called scripts is displayed until you reach the top of the call hierarchy.

**Copy Labels between Scripts**

You can use Browser to copy labels between scripts.

**When to use**

- If you need to incorporate a global label in a script which has been created in another script.
- If you have already created a local label in a script and need to create a copy of that label in another script.
How to use

To copy either global or local labels:
1. Locate the script containing the label you wish to copy.
2. Select the ‘Labels’ viewing option.
3. Highlight the label you want to ‘export’ as follows:

- Click Copy.
- Paste the label into either the Command window or Edit control of the script you want to use the label in.

Points to note

- If a local label is copied to another script, a duplicate label is created in the destination script. This label is then a local label to this script and all attributes of the label are retained. You can override the properties of the copied label if required.
- If a global label is imported into a script, a global reference is created in the destination script. You cannot change the properties of the imported label; you must carry out any modifications in the originating script. The status bar displays the owning script name and label name e.g. BOOKING.KEY#date. In the configure label dialogue box of the copied label, the global checkbox is replaced by ‘Defined in BOOKING.KEY’.
- Before you can copy a global label, the path of the owning script must appear in the browser search path.
• It is possible to have multiple scripts visible on screen and to drag and drop labels between them to copy over scripts, instead of using Browser.

Practice

• Open your Template script and immediately save it as another name. Using Browser practice copying Local and Global labels from scripts you have previously created into the new script.

• Use Browser to check the hierarchy, both up and down of scripts you have created.

Validation

You can use Validation ensure that the user gives only a limited selection of answers to a question asked.

When to Use

You can assign label values, which the user's response must match in order to proceed with the running of the script. For example, if creating a script for hotel reservations, you may want to restrict the hotel chains that the user can sell.

How to Use

To establish validation, open a script such as AIR.KEY.

The input panel appears as follows:
Points to note

- There are two levels of validation:
  - Validation Required provides recommended values which the user can override.
  - Strict Validation specifies values to which the user must adhere.

- The Strict Validation boxes, when tabbed to, appear in blue while the Validation boxes are normal background color.

- Strict Validation drop down boxes are immediately next to the input box. Validation drop down boxes are a slight distance from the input box.

- The Comments associated with the Validation appear in the status bar.

It is not immediately obvious that validation exists in a script as it is the labels themselves that are validated. The following instructions show how you can validate a label.

**Validate a Label**

To validate a label, use the following steps.

1. In the Configure label dialogue box, select one or both of the two validation checkboxes.
   - To apply validation, check the ‘Validation Required’ box.
   - To apply strict validation, check both boxes.

2. Choose the Values button
The ‘Valid Values for’ dialogue box appears along with any existing values that have already been entered:

The following table lists the screen components.

<table>
<thead>
<tr>
<th>Callout</th>
<th>Field or button title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enter Valid Values</td>
<td>Current list of values for this label.</td>
</tr>
<tr>
<td>2</td>
<td>Value</td>
<td>What is acceptable for this label when script is run.</td>
</tr>
<tr>
<td>3</td>
<td>Delete</td>
<td>Used to delete a highlighted value in Valid Values box if no longer needed.</td>
</tr>
<tr>
<td>4</td>
<td>Sort List</td>
<td>Sorts list into alphabetical or numeric order. If this button is not used, list will appear to user in order entered. This could be useful e.g. if most common airport used is JFK <em>not</em> EWR. Once sort has been used, there is no ‘unsort’. <strong>Note:</strong> List in above screen has <em>not</em> been sorted.</td>
</tr>
<tr>
<td>5</td>
<td>Comment</td>
<td>Optional description which appears on status bar when script is run.</td>
</tr>
<tr>
<td>6</td>
<td>Apply</td>
<td>Once a value and/or comment has been entered, this updates Valid Values list.</td>
</tr>
<tr>
<td>7</td>
<td>Reset</td>
<td>Clears any text present in Value and Comment boxes.</td>
</tr>
</tbody>
</table>

3. Ensure the next available space is highlighted after the last value in the Valid Values box, i.e. after SFO.

4. Enter IAD (as a value).
5. Enter Washington Dulles (as a comment).
6. Select Apply.
7. Repeat steps 3 to 5 until you have entered all acceptable values.
8. Select Sort List.
9. Select Close, then OK.

Points to note

- You must enter the Input Mask before assigning values to the label. The values entered must conform to the Input Mask.
- When you copy a local label from one script to another, all attributes of the label are retained including any validation entries. You can override the values of the copied label if required.
- If you reference a global label to another script, you cannot change the valid values. You must make any modifications in the originating script.

Practice

- Assign a strict validation to the departure label of your Air script and a validation to the destination label. Use airports of your choice and include a comment to identify each one.
- Assign a strict validation to the name field used in your Car script. The only names applicable for this script are:

<table>
<thead>
<tr>
<th>Name</th>
<th>Job title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alan Bones</td>
<td>Company Director</td>
</tr>
<tr>
<td>Brenda Tidy</td>
<td>Vice President - H.R.</td>
</tr>
<tr>
<td>Charles Ford</td>
<td>Vice President –Marketing</td>
</tr>
<tr>
<td>Colin Maxwell</td>
<td>Vice President – I. T.</td>
</tr>
</tbody>
</table>
Delete Global and Local Labels

You can delete labels within a script providing they are *not* in use.

When to Use

You can delete a label from a script if it has been created in error and is *no* longer required. This ensures that any person looking at the script in Scriptwriter Plus™ Build, has a clear picture of labels being used in the script.

How to Use

To delete a label, use the following steps.

1. Enter CTRL + L to activate the Label list of the MOREAIR2.KEY script. The Labels list appears.

2. Highlight the label to delete, ensuring the In Use column reads No.

3. Select Delete.

Points to note

- Deleting a global label that is used in other scripts is *not* possible until *all* references to that label have been deleted. If trying to do so, the following error appears.
• If you delete a script containing a global label that is used in other scripts, the other scripts utilizing the label will still function correctly. No changes to the label will be possible, however, as this is only possible in the script in which it was created.

Comments

The Comment command allows a script builder to enter any text in the Command window of a script that is relevant to script builders only. The Comment command is completely invisible to the end user.

When to Use

The ability to enter comments in a Command window is particularly useful when testing and developing scripts. Comments are not seen by the user and are not transmitted to the host. You can use them to describe the content and functionality of a particular piece of code for future amendments.

How to Use

To create a comment, use the following steps.

1. In the Command window, position the cursor in the correct position for the comment.
2. Click the Comment button on the Command bar.
3. Enter any relevant text ensuring that if more than one line is needed, the ‘comment symbol’ is repeated on the next line as follows:

Practice

Enter comments in two of the scripts you have created to inform other script builders of your logic.
Pause and Painted Messages

You can use the Pause command within the Command window to automatically pause the script when it is run. A painted message is usually used in conjunction with the Pause command to inform the user why the script has been paused. Although you can use pause and painted message separately, they work best together.

When to Use

If you want to allow the person running the script to access another application or check something in Apollo.

How to Use

To enter a painted message in the Command window:

1. Place the cursor in the desired location.
2. Enter a carriage return command from the host table.
3. Type in the freeform text.
4. Enter a carriage return command from the host table.

To enter a Pause command in the Command window:

1. Place the cursor on the next available line after the painted message.
2. Select Pause from the Command menu.

Example: Command window containing a painted message and Pause.

<table>
<thead>
<tr>
<th>If</th>
<th>While</th>
<th>Show</th>
<th>Call</th>
<th>Assign</th>
<th>Terminate</th>
<th>Comment</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| The departure label and arrival labels have been validated for this script to comply with company policy Ch2 Ch1 Ch3 Chn ShowWindow
|        | 'e'   | AIR KEY#date AIR KEY#departure AIR KEY#destination Ed2 | | | | |
|        | 'c'   | 'This script will be paused before any invoicing details' Ch | | | | |
|        | 'c'   | 'are requested from you. Please check the company invoice' Ch | | | | |
|        | 'c'   | 'policy with head office' Ch Chn | | | | |
| Pause  |       | Call  | ( INVOICE KEY ) | | | | |
| Call1  |       | ( HOTEL KEY ) | | | | | |

Points to note

- The Carriage return command is not required after every line. You will need to test the script and make any cosmetic alterations you think necessary.
- A SOM has been included after the Painted message to ensure any future Apollo entries are not affected.
Practice

Enter a Pause command and a painted message in your Car script to prompt the user to check any special offers currently being offered by Avis before calling in your script which sells a car.

Exercise – Hotel, Car and Visa

You have been asked to create a script to assist a new agent to:

- Enter Unassociated Itinerary Remarks
- Book a hotel room
- Request visa information
- Book a car (depending on destination)

Begin the script by asking ‘What city is the client traveling to?’

Itinerary remarks

The user will input the full city name and this will be used to enter an Unassociated Itinerary Remark. The entry to do this is:

RMU: YOUR INFORMATION FOR SYDNEY

Hotel details

1. To enable the sell to take place, there must be a name present. The Apollo® entry to do this is:
   N: LASTNAME/firstname

2. The Apollo entry to check Hotel index:
   HOI 12JUN-17JUN XXX/MC+CY+RD

3. Followed by a complete availability display:
   HOC + line number, e.g. HOC1

4. Followed by a rules display:
   HOV + line number e.g. HOV1

5. The Apollo entry to sell from availability for the above vendors, using a deposit as a form of guarantee is:
   01INSIDE/G-DPST
Visa details

The option to check visa information is required.

Car details

Finally if the client is traveling to ‘DENVER’, you will call in your car script and display the PNR in Window 2.

Points to note

- All mandatory parts of the entries are in bold, all others are variable and up to the user to specify.
- XXX city code has been used for the Hotel city as this is a training city.

Nested Conditional Logic

Nested Conditional logic is where an If statement is placed within a previous If/Endif statement.

When to Use

A good example of nested Conditional logic is when there are many different choices the user can make. Accounting for every one of differences would involve many single If/Endif statements. Nested Conditional logic accounts for one statement being true and the others relying on it.

How to Use

Nested Conditional logic in Scriptwriter Plus™ Run

Run the NESTFARE.KEY script.

Points to note

- Three Frames have been used for 2 different reasons:
  - Frame around the routing details is for appearance only.
  - Frames around the passenger type and journey type groups are mandatory to group together Radio buttons. This ensures only one of the group is chosen.
- If choosing ‘Senior’, there are 2 further choices to make. The same applies if ‘Companion’ is chosen.
• Defaults have been used when creating the labels for the Radio buttons to ensure at least one of each group is chosen.

Nested Conditional logic in Scriptwriter Plus™ Build

Open the NESTFARE.KEY

Input window

The Input window has been compiled using Text controls, Edit controls, Radio buttons and Frames.

Points to note

• All Radio buttons have a conditional mask, C as seen previously.

• The ‘Companion’ and ‘Round Trip’ Radio buttons have a default value set which ensures that this Radio button is highlighted when the script is run as follows.
Command window

The Command window is compiled using conditional If statements. In this case, however, there are If/Endif statements within other If/Endif statements.

<table>
<thead>
<tr>
<th>If</th>
<th>While</th>
<th>Show</th>
<th>Call</th>
<th>Assign</th>
<th>Terminate</th>
<th>Comment</th>
<th>Host</th>
</tr>
</thead>
</table>
| Show Window
Sen = "D" NESTFARE KEY#DATE NESTFARE KEY#DEPT NESTFARE KEY#DEST
If ( NESTFARE KEY#SENIOR )
  "SENIOR"
  If ( NESTFARE KEY#CHF )
    "CHF"
    If ( NESTFARE KEY#COMPANION )
      "COMPANION"
    Endif
    If ( NESTFARE KEY#ROUNDTRIP )
      "RT"
    Endif
  Endif
If ( NESTFARE KEY#COFF )
  "COF"
  If ( NESTFARE KEY#OVERRIDE )
    "OVERRIDE"
  Endif
  If ( NESTFARE KEY#ROUNDTRIP )
    "ST"
  Endif
Endif
|     |       |      |      |        |           |         |      |

Points to note

- If, Else, Endif statements are automatically indented within the Command window and those which are part of the same statement are always vertically aligned.

- The first If command checks the first selection, i.e. Senior or Companion fare. Nested in that statement is the second choice; One way or Return. The second nested ‘If’ statement is only carried out if the first ‘If’ statement is evaluated true.

- As there are only two choices in the second group, i.e. One Way or Round trip, the Else statement has been used.

- The host entry is ‘built’ depending on options chosen. The transmit is not performed until all the host entry possibilities have been accounted for.

- The main reason for nested conditional logic is to ensure lines of code are not executed unnecessarily, i.e. if the first If statement is not evaluated true, the time and resources are not wasted executing the commands within the If, Else statements.
Exercise – Air Availability with Specific Carriers

Write a script that allows the user to request air availability. You will also give them to option to specify up to three preferred carriers.

Points to note

- Use Edit controls to determine the carriers the user prefers.
- The Apollo® system entry to request availability with specified carriers is:
  A23NOVNYCLON+BA.VS.AA
  It is only possible to compile the correct entry by using nested conditional logic to ensure the correct separators are used.
- You may want to make use of the existing Air Availability script you have.
- All mandatory parts of the entries are in bold, all others are variable and up to the user to specify.

While and Assign Commands

You can use the While and Assign commands separately but they are commonly used in conjunction with each other.

When to Use

While

If you want to instruct a script to repeat a procedure as long as a conditional expression is evaluated true. The procedure should only stop when the condition is evaluated false.

Assign

When you want to allocate a value to a label. The value can be one of the following:

- Constant (a value that is fixed)
- Value of another label
- Result of an expression
How to Use

**While and Assign in Scriptwriter Plus™ Run**

Run the WHILE.KEY script.

**Points to note**

- The script allows the user to create multi segment itineraries. The user inputs the number of segments to be created and as long as the value is greater than 0, the script will continue to run.
- The script can be terminated at any time by entering ALT + T.

**While and Assign in Scriptwriter Plus™ Build**

Open WHILE.KEY

The Command window appears as follows:

```plaintext
If      While      Call      Assign      Terminate     Comment     Host
ShowWindow
While ( WHILE.KEY#numberlabel  >  0 )
Call ( AIR.KEY )
WHILE.KEY#numberlabel := WHILE.KEY#numberlabel - 1
EndWhile
```

**Points to note**

- As an arithmetic function is being performed, numberlabel is compiled using a numeric mask.
- The numeric value entered by the user is not automatically decremented, so this has to be done manually by the assign statement.
- The arithmetic operator > (greater than) is used. Examples of other arithmetic operators can be found in the online help.
To recreate this Command window, use the following steps.

1. Select Show Window from the Command Bar.

2. Select While from the Command Bar and enter the following expression.

   − Use the Labels button to select ‘numberlabel’.
   − The text ‘> 0’ is typed directly into the box.

3. Click OK.

4. Create 2 blank spaces between While and Endwhile.

5. With the cursor in the first blank space, select Call from the Command Bar and locate the AIR.KEY script.

6. With the cursor in the second blank space, select Assign from the Command Bar and enter the Assign expression as follows:

   − Use the Labels button to select ‘numberlabel’.
   − The text ‘- 1’ is typed directly into the box.

7. Click OK.
**Exercise – PRO-files**

Write a script to ask how many Personal Files the user would like to build for a new business account called Infosys.

The Business File, Infosys, has already been built.

The script to build the Personal Files has already been created. It is called PARB.KEY and can be found in the default Scriptwriter Plus Directory on your PC.

The script will commence by asking the user how many Personal Files they wish to create. Providing this value is greater than zero, call in the PARB.KEY script.

**Exercise – IBM Script**

You have been asked to write a script for IBM. There are three IBM employees who regularly travel from New York to London.

**Passenger details:**

- Jan Hobbs - Company Chairperson
- Paul Chambers - Company Director
- Steven Ross - Sales executive

**Air details:**

The clients always travel:

- Outbound on BA112 from JFK to LHR
- Return on BA175 from LHR to JFK
- Jan Hobbs and Paul Chambers travel in ‘F’ class.
- Steven Ross travels in ‘S’ class.

The direct sell entry for an air segments is:

**0BA112F12SEPLHRAMSGK1**

- The person running the script enters the date.
- The class of travel is dependent on the person traveling.
Hotel details:

The clients always stay at the Galileo Plaza, chain code ZQ at city HDQ. The property number is 78482.

- For Jan Hobbs and Paul Chambers book room type A1DCOR.
- For Steven R Ross book room type B1KCOR.

The direct sell entry for the hotel segment is:

**0HHLZQNN112SEP-14SEP78482A1DCOR-1/G-DPST**

- The person running the script enters the dates.
- The room type is dependent on the person traveling.

Separate PNRs are required for each person. Start by asking how many people the reservation is for, e.g. if the reservation is for one person, the script(s) should be run once, if the reservation is for two people, the script(s) should be run twice etc.

### Looping Scripts

Whenever there is a need for a script to be re-run, you can use a looping script.

### When to Use

An example of this would be availability, whether air, car or hotel. In past examples, we have *only* accounted for one additional availability to appear. If the user wanted to display more than one, a looping script would be required.

In this case when a script may be required more than once in a ‘series’ of scripts and that script calls ‘itself’, then this causes problems with PC memory. This is because there is no limit to the number of instances a script can be open, much like other PC applications, e.g. Microsoft® Word.

Avoid the following as MORECAR calls itself in.
How to Use

A looping script in Scriptwriter Plus™ Run

Run AIR.KEY

After the date and the city code information have been entered, the MORELOOP.T2 script keeps displaying until the user is satisfied that they see the segment they require. Only then is the sell script called.

A looping script in Scriptwriter Plus™ Build

Use Browser to see the hierarchy involved in this series of scripts.

<table>
<thead>
<tr>
<th>Call Hierarchy (Down)</th>
</tr>
</thead>
<tbody>
<tr>
<td>air.key</td>
</tr>
<tr>
<td>moreloop.key</td>
</tr>
<tr>
<td>moreloop.t2</td>
</tr>
<tr>
<td>sell.key</td>
</tr>
</tbody>
</table>

Use the following steps.

1. Open AIR.KEY

   This can be done from Browser by double clicking on the script.

<table>
<thead>
<tr>
<th>If</th>
<th>While</th>
<th>Show</th>
<th>Call</th>
<th>Assign</th>
<th>Terminate</th>
<th>Comment</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIR.KEY</td>
<td>ShowWindow</td>
<td>Date AIR.KEY</td>
<td>Departure AIR.KEY</td>
<td>Assign</td>
<td>Moreloop.KEY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MORELOOP.KEY</td>
<td>While</td>
<td>MORELOOP.KEY</td>
<td>MORELOOP.KEY</td>
<td>Y</td>
<td>MORELOOP.KEY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MORELOOP.T2</td>
<td>Call</td>
<td>MORELOOP.KEY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

   This script checks for air availability and calls MORELOOP.KEY

2. Open MORELOOP.KEY

   - MORELOOP.KEY has nothing in the Input window. It sets up a loop using the While statement. As long as the While statement is evaluated true (i.e. the MORELOOP.KEY#loop label equals Y) then the MORELOOP.T2 script is called.
   - The Assign command has been used to allocate a value of ‘Y’ to the loop label. This could be any value at all. It in only used as a marker.
3. Open MORELOOP.T2

![Image of MORELOOP.T2]

- This allows the user to request even more availability if required.
- The *only* time the loop label is changed, to prevent the Moreair.T2 script being called again, is when the user selects the No.

Practice

Add the option to your Air Availability script, to view more than one availability using the above method.
Module Review

Using the course book or online help, answer the following.

1. What is Conditional Logic?

______________________________________________________________________________

2. Give two examples of logical operators.

______________________________________________________________________________

3. What input mask is used with a Radio button?

______________________________________________________________________________

4. The search path of Browser appears as follows. Which drives and directories will be searched?

______________________________________________________________________________
5. You are contemplating deleting a script, but first want to check if any others ‘call’ the script. You edit the search path to look in all the directories storing scripts, then select ‘Do Search Now’. The script you want to delete is shown in the Script Name field, so you highlight it. Which of the four ‘Show’ Radio buttons do you select?

a. Labels
b. Host Commands
c. Call Hierarchy (Up)
d. Call Hierarchy (Down)

6. When running a script, the following input screen appears:

   ![Image of input screen]

   a. What type of validation has been used?

   b. Give two reasons for this decision.
Module 5: Course Review

The course review is divided into two parts:

1. 10 questions on Scriptwriter Plus™ Run and Build
2. Tasks within Scriptwriter Plus™ Build

Part 1

Scriptwriter Plus™ Run

1. You want to change the name of a Group Window. Outline the procedure to do this.

2. What does the keystroke SHIFT + ENTER allow you to do?

Scriptwriter Plus™ Build

3. What does the ‘Align with Grid’ function do?

4. Which command would you use in the Command Window to instruct the script to repeat a procedure as long as a Conditional expression is true?

5. What is the keyboard shortcut entry to move a control in the Input Window?
6. What is the difference between ‘Validation’ and ‘Strict Validation’?

7. How many Local Labels are used in the CarSell.key saved in the default Scriptwriter Plus directory?

8. The Assign command is used when you want to allocate a value to a label. In general terms, what could this value be? (No need to give exact examples).

9. The following example illustrates a conditional expression. What will the result if the person running the script enters JNB as a destination city?

```plaintext
If ( ACITY.KEY#destination - "EKK" OR ACITY.KEY#destination - "SIN" )
   Call ( AIR.KEY )
Else
   Call ( VISA.KEY )
Endif
```
10. The labels dialogue box for a script you are writing are as follows:

<table>
<thead>
<tr>
<th>Label Name</th>
<th>Scope</th>
<th>Mask</th>
<th>Default</th>
<th>In Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>carsize</td>
<td>Local</td>
<td>ppppppp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>date</td>
<td>Local</td>
<td>333aaa</td>
<td></td>
<td></td>
</tr>
<tr>
<td>input</td>
<td>Local</td>
<td>aa39999</td>
<td></td>
<td></td>
</tr>
<tr>
<td>init</td>
<td>Local</td>
<td>aaaaaaa</td>
<td></td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>Local</td>
<td>x99999999</td>
<td></td>
<td></td>
</tr>
<tr>
<td>outdate</td>
<td>Local</td>
<td>999999</td>
<td></td>
<td></td>
</tr>
<tr>
<td>outsource</td>
<td>Local</td>
<td>aa39999</td>
<td></td>
<td></td>
</tr>
<tr>
<td>outroute</td>
<td>Local</td>
<td>aaaaaaa</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- a. Will you be able to delete the label called ‘carsize’?

- b. Will the ‘Indate’ label accept a value of 1JUL?

**Part 2**

You will need to rectify scripts that contain errors. Your aim is to ensure the scripts run correctly.

Five scripts have been loaded onto the hard disk of your PC:

- EVAL.KEY
- AVAIL.KEY
- MORE.KEY
- SELL.KEY
- FINALLY.KEY

Your instructor will tell you which drive and directory they are in.
Notes
Appendix A: Style Guidelines

It is recommended that companies adopt a common style for script design. This ensures certain design decisions are followed by all script builders to provide a common ‘look and feel’ across all scripts.

You may wish to use a particular color, style and design for a particular agency following their corporate image. This could also encompass using a bitmap with their Logo.

Here are some recommended guidelines in style design and color that you may wish to adopt for the Input Window.

The Script border has been sized to height 15 width 74, which, when running a script, will fill a Focalpoint window.
Notes
Appendix B: Bitmaps

The following Icons are loaded and available when Scriptwriter Plus is installed. They can be found in the directory `\DATADIR\BITMAPS`. All have the extension .BMP.

- AERO
- INFO
- BUS
- ISLAND
- CAR
- NOTES
- CONVERT
- PASSPORT
- CURRENCY
- PLANE
- DEFAULT
- SHIP
- DOCUMENT
- THEATRE
- FALL
- TRAIN
Appendix C: Answer Key

Following are the answers to the reviews that are at the end of each module and to Module 5: Course Review.

**Module 2: Scriptwriter Plus Run**

<table>
<thead>
<tr>
<th>Question number</th>
<th>Correct answer</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>There is <em>no</em> Maximum</td>
<td>Any number of group windows can be created providing screen and PC’s memory can accommodate them.</td>
</tr>
</tbody>
</table>
| 3               | 1. Highlight group window to be changed.  
2. Select Edit, Window Properties.  
3. Amend Window title as required. |           |
| 4               | D              |           |
| 5               | Any changes made since Scriptwriter Plus Run was opened will be lost. | As ‘Save Settings on Exit is *not* checked, any changes made to configuration of Scriptwriter Plus windows will *not* be saved. |
| 6               | Window, Tile   |           |
| 7               | One            | Only one central system command can be transmitted/received at a time. |
| 8               | ALT + T  
or  
ALT + F4 |           |
## Module 3: Scriptwriter Plus Build – Basics

<table>
<thead>
<tr>
<th>Question Number</th>
<th>Correct Answer</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Yes, but it Label name field should <em>not</em> be completed.</td>
<td>Labels are <em>not</em> required in text controls as they are there to display static text, <em>not</em> to accept variables.</td>
</tr>
<tr>
<td>3</td>
<td>Window</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>300-350</td>
<td>It prevents bad performance when running scripts</td>
</tr>
<tr>
<td>5</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>A Global label retains its value once it has been entered. It can also be used in script in which it was <em>not</em> created</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Changes proportions of Input and Command windows.</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>G:\DATADIR\SWPLUS\SCRIPT or C:\FP\DATADIR\SWPLUS\SCRIPTS</td>
<td>Path is dependent on whether user is using a turnkey Focalpoint or Platform Independent Focalpoint installation.</td>
</tr>
<tr>
<td>10</td>
<td>G:\DATADIR\USERS\NAME or C:\FP\DATADIR\USERS\NAME</td>
<td>Path is dependent on whether user is using a turnkey Focalpoint or Platform Independent Focalpoint installation.</td>
</tr>
<tr>
<td>11</td>
<td>SHIFT + CTL + Z</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1. No ‘Show Window’ command 2. Incorrect Host entry sequence</td>
<td>Correct sequence for Galileo Host should be: A{date}{departure airport}{arrival airport}</td>
</tr>
<tr>
<td>13</td>
<td>The Script border</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>xx</td>
<td>You need to account for all possible airline codes e.g. UA, AA, 3M, Z9.</td>
</tr>
</tbody>
</table>
## Module 4: Scriptwriter Plus Build – Beyond Basics

<table>
<thead>
<tr>
<th>Question Number</th>
<th>Correct Answer</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Conditional logic is used in Command Window when a script can follow more than one path.</td>
<td>To determine which path to follow, a conditional expression is evaluated and acted upon.</td>
</tr>
<tr>
<td>2</td>
<td>AND, NOT, AND NOT, OR, OR NOT</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>C – Conditional</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>A:</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>a) Strict</td>
<td></td>
</tr>
<tr>
<td></td>
<td>b) Selection box is dark in color and drop down arrow is immediately adjacent to selection box.</td>
<td></td>
</tr>
</tbody>
</table>
Module 5: Course Review

Part 1

<table>
<thead>
<tr>
<th>Question Number</th>
<th>Correct Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1. Select Edit</td>
</tr>
<tr>
<td></td>
<td>2. Select Window Properties.</td>
</tr>
<tr>
<td></td>
<td>3. Enter the new window title.</td>
</tr>
<tr>
<td></td>
<td>4. Click OK.</td>
</tr>
<tr>
<td>2</td>
<td>Add a Group Window</td>
</tr>
<tr>
<td>3</td>
<td>Allows controls to be positioned in alignment with an invisible grid.</td>
</tr>
<tr>
<td>4</td>
<td>While</td>
</tr>
<tr>
<td>5</td>
<td>CTRL + M</td>
</tr>
<tr>
<td>6</td>
<td>Validation – Recommended Values</td>
</tr>
<tr>
<td></td>
<td>Strict Validation – Values the user must adhere to</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>• Constant value (value that is fixed)</td>
</tr>
<tr>
<td></td>
<td>• Value of another label</td>
</tr>
<tr>
<td></td>
<td>• Result of an expression</td>
</tr>
<tr>
<td>9</td>
<td>The VISA.KEY script will be called</td>
</tr>
<tr>
<td>10</td>
<td>a) No – label is still in use</td>
</tr>
<tr>
<td></td>
<td>b) No – only 2 numerics and 3 alphas will be accepted</td>
</tr>
</tbody>
</table>
# Part 2

<table>
<thead>
<tr>
<th>Script Title</th>
<th>Amendment</th>
</tr>
</thead>
</table>
| AVAIL.KEY    | Input window  
  • Amend Date label to read either 99aaa or 9xaaa  
  • Create a new board point text control  
  • Amend off-point label mask to read aaa  
  • Add must enter and auto-transmit control attributes to ‘off-point’ edit control  
Command window  
  • Insert A before labels |
| MORE.KEY     | Input window  
  • Attach the yes label to the ‘Yes’ Radio button  
Command window  
  • Insert transmit or RD2 after A* |
| SELL.KEY     | Input window  
  • Delete auto-transmit from the ‘Number of Passengers’ edit control  
  • Add must enter and auto-transmit control attributes to the ‘From line’ edit control  
Command window  
  • Add Show Window command  
  • Insert ‘Class’ label into sell entry |
Notes