

เรียนรู้พื้นฐานการเขียน AVENUE

เรียนรู้พื้นฐานการเขียน AVENUE

คัดแปลงและเรียบเรียงจาก หนังสือ Using AVENUE โดย ESRI

เทคนิคในการเรียนรู้ AVENUE ซึ่งเป็นรูปแบบของคำสั่งในการเรียกใช้งาน ARCVIEW ได้อย่างดี นั้น มีหลายๆ ท่านมีคำถามว่ามีความจำเป็นต้องเรียนรู้การเขียนโปรแกรม (Programming) ไหม ในส่วนของคำตอบนั้นคือ ถ้ามีพื้นฐานบ้างก็จะดี แต่ถ้าไม่มีนั้นเราสามารถที่จะเรียนรู้การใช้คำสั่งจาก ตัวอย่างหลายๆ ตัวอย่าง แล้วคิดว่าตัวอย่างคำสั่งใดบ้าง ที่เหมาะสมกับงานที่เรากำลังจะจัดทำ แล้วเราก็ สามารถที่จะคัดลอก และคัดแปลงคำสั่งเหล่านั้นให้เหมาะสมกับงานที่เรากำลังจะทำ

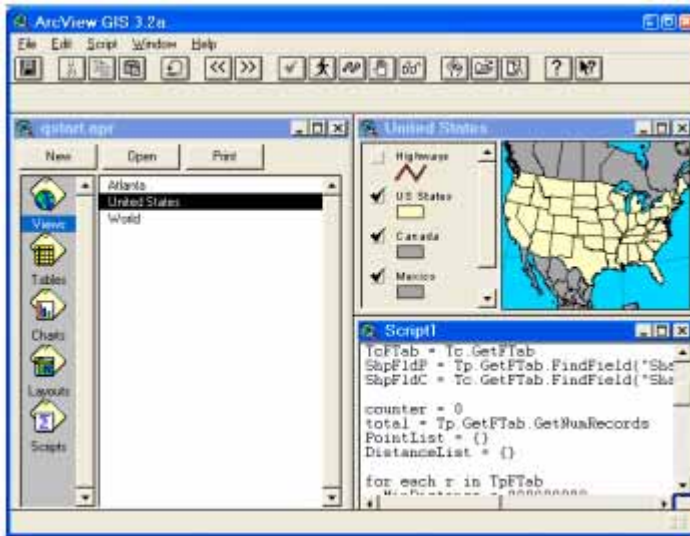
แต่ก่อนที่เราจะสามารถเรียนรู้การเขียนคำสั่ง AVENUE ได้ดีนั้น เราจะต้องมีพื้นฐานความรู้ของ ARCVIEW มาดีในระดับหนึ่งที่น่าสนใจถึงกระบวนการในการใช้งานมาบ้าง ถึงจะสามารถควบคุมวัตถุ (Object) ต่างๆ ที่อยู่บนหน้าต่างของ ARCVIEW นั้นได้อย่างมีประสิทธิภาพ ซึ่งหน้าต่าง ที่ปรากฏในโปรแกรมมักจะเรียกในภาษาอังกฤษว่า Graphic User Interface ในที่นี้อาจจะเรียก ได้ว่า ArcView's user interface นั้นเอง

ถ้าเรามีความรู้ความเข้าใจถึง ArcView's user interface ว่าจะมีลักษณะการทำงานอย่างไร ก็ จะทำให้เราสามารถที่จะเข้าใจว่าเราควรจะใช้คำสั่งใน AVENUE ควบคุมการแสดงผลของแต่ละ Interface ได้อย่างไรนั่นเอง เพราะฉะนั้นถ้าหากเราไม่มีความคุ้นเคยกับการใช้โปรแกรม ArcView เราสามารถหาเอกสาร ตำรา ที่มีการสอน และเรียนรู้ด้วยตัวเอง เพื่อให้คุ้นเคยและเข้าใจ ในการทำงานบน ArcView ได้ดีในระดับหนึ่ง แล้วเราถึงจะพอเข้าใจกระบวนการในการควบคุมการ ทำงานของ Object ต่างๆ ได้

ความหมายของ AVENUE

Avenue เป็นชุดของคำสั่งบน ArcView ที่ใช้ในการควบคุมวัตถุต่างๆ ที่ปรากฏอยู่บนโปรแกรม เนื่องจาก Avenue นี้สามารถทำงานได้บน platform ที่มีโปรแกรม ArcView และ วัตถุ (object) ต่างๆ สามารถใช้งานร่วมกันได้ระหว่าง Avenue และ ArcView เช่น หน้าต่าง (Window) ของ Project, View, Table, Chart, Layout และ Script ต่างๆ นอกจากนี้ยังมีวัตถุย่อยๆ ที่อยู่ภายใน Window ที่ละชนิดที่จะทำงานได้แตกต่างกันออกไปอีก ถ้าเรามีความคุ้นเคย ในการใช้งาน ArcView จะทำให้เราเข้าใจถึงอุปกรณ์แต่ละชนิดที่อยู่บน ArcView ได้ดีขึ้น

หน้าต่างตามปกติของ Graphic User Interface ดังรูปข้างล่าง

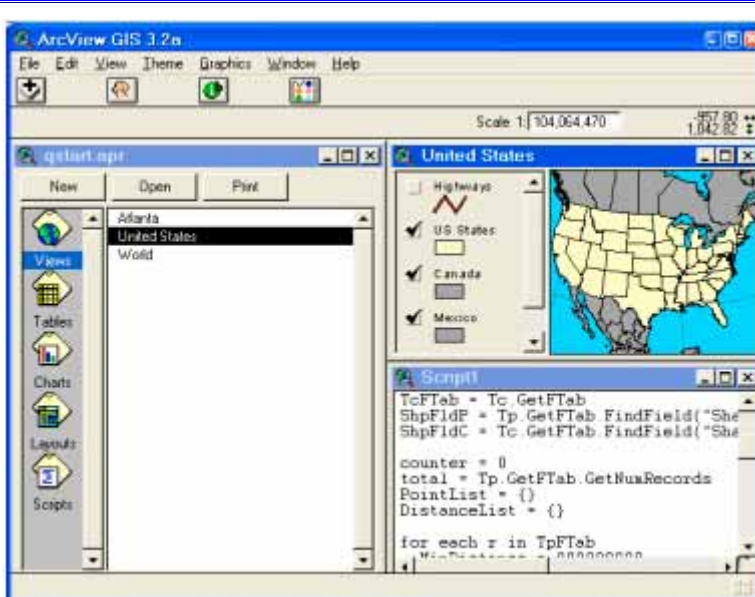


ความสามารถของ AVENUE

เราใช้ Avenue ในการปรับปรุงหรือดัดแปลงรูปแบบของหน้าต่าง (GUI = Graphic User Interface) เช่น สามารถสร้างหรือลบ คำสั่งใน เมนู (menus) หรือปุ่มคำสั่ง (buttons) และ ปุ่มเครื่องมือต่างๆ (tools) ที่ปรากฏบนหน้าต่างแต่ละชนิด ซึ่งไม่เหมือนกัน ขึ้นอยู่กับว่าเป็นหน้าต่างชนิดใด เช่น หน้าต่าง (Window) ของ Project, View, Table, Chart, Layout และ Script

บางครั้งอาจจะใช้ในการสร้างหน้าต่างของการประยุกต์เฉพาะงานของหน่วยงานเรา ที่ต้องการรูปแบบของหน้าต่าง เฉพาะไม่ซ้ำแบบใคร เราก็สามารถใช้ Avenue เขียนเพื่อให้รูปแบบที่แตกต่างกันออกไป เช่น ให้นำหน้าต่าง ไม่ซ้ำซ้อน เพื่อให้ใช้งานง่ายกับ user อาจจะต้องตัดคำสั่ง หรือปุ่มบางอย่างออกไป ให้เหลือเฉพาะงานของเราได้ ที่จำเป็นต่องานประยุกต์นั้นๆ

หน้าต่างที่เปลี่ยนแปลง Graphic User Interface ดังรูปข้างล่าง



ทำความเข้าใจ Object Oriented Programming (OOP)

Object Oriented Programming = "การเขียนโปรแกรมเชิงวัตถุ" หมายถึง การนำวัตถุในโลกแห่งความจริง (real world objects) มาทำการออกแบบจำลอง หรือ Model ให้มีลักษณะคล้ายคลึงกับวัตถุจริงนั้น จะใช้คุณลักษณะของ State และ Behavior

ในการจำลองของ real world objects ด้วย software objects นั้น จะใช้การจำลองวัตถุต่างๆ ด้วย Class และจำลอง State ด้วยตัวแปรของ Class และ Behavior

การเขียนโปรแกรมเชิงวัตถุ นั้น เป็นการเขียนโปรแกรมโดยการใช้หลักการของการออกแบบเชิงวัตถุ ซึ่งจะมีการออกแบบโดยการมอง และวิเคราะห์สิ่งต่างๆ ให้เป็นวัตถุ (Object) ซึ่งประกอบมาเป็นชั้น (Class) การประกอบเป็นชั้น (Aggregation of Objects into Class) และการถ่ายทอดคุณสมบัติระหว่างชั้น (Inheritance) ซึ่งการทำงานดังกล่าว ทำให้การวิเคราะห์และออกแบบโปรแกรมง่ายขึ้น และลดความซับซ้อนของการโปรแกรม และการวิเคราะห์ระบบที่มีความซับซ้อน ตลอดจนการใช้การออกแบบเชิงวัตถุจะช่วยลดเวลาในการพัฒนาโปรแกรม ตลอดจนสามารถจัดการข้อมูลได้อย่างมีประสิทธิภาพกว่าวิธีการแบบดั้งเดิมอย่างมาก

Objects คืออะไร

Objects หรือ วัตถุในที่นี้ ให้หมายถึง สิ่งต่างๆ ที่เราจะทำงานด้วย บนโปรแกรม ArcView เช่น Window ต่างๆ ตัวอย่าง View window ก็จะมีวัตถุอื่นๆ อีก ได้แก่ Themes แผนที่ต่างๆ หรือ Message Box ข้อความบอกข้อมูลในการทำงานหรือปฏิบัติการ ตลอดจนปุ่มคำสั่งต่างๆ นั่นเอง

สิ่งต่างๆ ที่เราทำงานอยู่บนโปรแกรม ArcView คือวัตถุ (object) เมื่อไรก็ตามที่เราได้เปิด View นั้นเรากำลังทำงานกับ View Object และเมื่อเราทำการเพิ่มปุ่มคำสั่ง (button) ใน หน้าต่าง (Interface) นั่นคือเราทำงานกับ Button object

Class จัดการกับ Object ได้อย่างไร

วัตถุใดๆ ที่มีคุณสมบัติคล้ายคลึงกัน ก็จะถูกรวมอยู่ภายใน Class เดียวกัน ในตัวอย่างที่ใน ArcView Project ก็จะมี View Windows หลายหน้าต่าง เป็น View Class นั่นเอง



View เป็น document หนึ่งในโปรแกรม ArcView ที่จะต้องทำงานบน Project แต่ละ View จะได้รับถ่ายทอดคุณสมบัติระหว่างชั้น (Inheritance) จาก document เช่นใน View ต่างๆ จะมีรูปแบบหน้าต่าง (Interface) ที่เป็นแบบเดียวกัน เช่น มีปุ่มคำสั่ง ปุ่มเครื่องมือ หรือเมนูที่เหมือนกัน แต่ในขณะที่แต่ละ View จะมีวัตถุที่อยู่ภายในต่างๆ ก็ได้ เช่น มีแผนที่แต่ละ Layer ไม่เหมือนกัน ก็เป็นไปได้

เพื่อให้เกิดความเข้าใจในการทำงานของโปรแกรมเชิงวัตถุมากขึ้น ให้ทำความเข้าใจลักษณะการทำงานของโปรแกรมเชิงวัตถุดังนี้

ตัวของ Object หรือเรียกว่า Instance เป็นวัตถุที่เก็บข้อมูลเหล่านั้น

กลุ่มของวัตถุที่มีรูปแบบ (Properties) หรือพฤติกรรม (Method) เหมือนๆ กัน จะเก็บไว้ในชั้นเดียวกัน (CLASS)

Instance ต่างๆ ใน Class จะมีข้อมูล หรือ Properties ซึ่งชื่อของ Properties อาจจะมีเหมือนกัน แต่ค่าภายในมีความต่างกัน เช่น ใน Class View จะมี Properties ที่ชื่อว่า NAME ซึ่งใน Instance หนึ่งมีค่าว่า Atlanta ซึ่งในอีก View หนึ่งมีค่าเป็น World ซึ่งค่านี้ทำให้ Object ใน Class เดียวกันมีความต่างกัน

Instance ยังสามารถกำหนดพฤติกรรม (Method หรือ Action) ได้โดยกำหนดใน CLASS ที่เป็นต้นสังกัด เช่น ใน Class VIEW จะมี Method "AddTheme(aTheme)" ซึ่งจะทำหน้าที่ทำงานเมื่อต้องการเพิ่มแผนที่ Theme เข้ามาใน View หรือ Method "DeleteTheme(aTheme)" ทำหน้าที่ทำงานเมื่อต้องการลบแผนที่ที่ออกจาก View เป็นต้น Method เหล่านี้จะเป็นมาตรฐานสำหรับ CLASS ต่างๆ ที่ Object (ในที่นี้หมายถึง View) นั้นเป็นสมาชิกอยู่

CLASS สามารถถ่ายทอด Method และ Properties ลงไปถึง CLASS ที่เป็นสมาชิก (Sub Classes) ต่อลงไปเรื่อยๆ หรือเรียกว่า Inheritance ซึ่งเป็นกรกำหนดชั้น (Classes) และพฤติกรรม (Behavior) ของชั้นต่างๆ ซึ่งสามารถถ่ายทอดลงไปตามลำดับชั้นที่เป็นลูก (Sub Classes) ทำให้ลดขนาดของโปรแกรมลงไปได้มาก (code re-use and code organization)

Methods และ Properties ใน CLASS ต่างๆ จะถูกเขียนทับได้โดยการใช้คุณสมบัติ Overiding โดยที่สมาชิกสามารถกำหนดคุณสมบัติทับลงไปได้

CLASS ใดๆ สามารถถูกถ่ายทอด**คุณสมบัติ**จาก CLASS ที่เป็นแม่ (SuperClass) มากกว่าหนึ่งได้

คุณสมบัติที่สำคัญของ OO คือ การที่สามารถมี Method ที่มีชื่อเดียวกันในหลายๆ CLASS โดยที่ไม่จำเป็นที่จะมีพฤติกรรม หรือการทำงานเหมือนกัน

Class ต่างๆ ใน โปรแกรมเชิงวัตถุ จะมีกระบวนการที่จะสร้าง Instance(Object) ของ Class นั้นๆ โดยการเรียก Constructor Method ซึ่งมักจะเป็นกระบวนการที่ใช้ชื่อเดียวกันกับชื่อของ Class นั้นๆ

Request จัดการวัตถุในการทำงาน

Request จัดการสิ่งต่างๆ ที่เกิดขึ้นใน ArcView เมื่อเราทำการกดปุ่ม หรือเลือกคำสั่งจากเมนู ArcView ก็จะมี Request ไปยังวัตถุ ดังนั้น Request จึงอาจจะเป็นคำสั่งในการสร้างวัตถุใหม่ เช่น เพิ่ม Theme ใหม่เข้ามาใน View สามารถเรียกค้นรายละเอียดที่มีอยู่ในวัตถุต่างๆ เช่น ชื่อของ View ที่ใช้งานอยู่ในปัจจุบัน หรืออาจจะมีคำสั่งในการเปลี่ยนแปลงคุณสมบัติของวัตถุ เช่น เปลี่ยนชื่อของ Theme หรือ View ที่กำลัง Active อยู่

คำสั่งของ Avenue Request โดยทั่วไปจะเริ่มต้นด้วยคำกริยา (Verb) ที่บ่งบอกถึงการกระทำที่โปรแกรมจะทำงาน เช่น GetProject, FindScript, AddDoc และ SetValue ซึ่งคำว่า Get, Find, Add และ Set นั้นเป็นกริยาที่จะกระทำกับ Project, Script, Doc และ Value ซึ่งทำให้ผู้เขียน script เข้าใจ function และ พฤติกรรมของ Avenue Request โดยการจำ keyword ของ กริยา ที่จะถ่ายทอดไปยังชื่อของมัน

คำกริยา	การกระทำ	ตัวอย่าง
Add	เพิ่มวัตถุอื่นๆ ในกลุ่มของวัตถุ	<code>AView.AddTheme</code>
As	แปลงวัตถุหนึ่งไปเป็นอีกวัตถุหนึ่งใน Class ที่ต่างกัน	<code>ANumber.AsString</code>
Can	ตรวจสอบว่าวัตถุสามารถทำคำสั่งที่กำหนดได้	<code>ATheme.CanEdit</code>
Find	ค้นหาวัตถุที่กำหนด และคืนค่าให้กับวัตถุนั้น	<code>AView.FindTheme("Map")</code>
Get	คืนค่าที่อ้างอิงไปยังวัตถุ	<code>AView.GetThemes</code>
Has	แสดงผลถ้าวัตถุมีเงื่อนไขหรือสถานะที่กำหนด	<code>AScript.HasError</code>
Is	คืนค่า True หรือ False	<code>AButton.IsVisible</code>
Make	สร้างวัตถุใหม่	<code>View.Make</code>
Return	สร้างวัตถุใหม่ และคืนค่าไปยังวัตถุ	<code>AFileName.ReturnDir</code>

Set

กำหนดค่าคุณสมบัติให้กับวัตถุ

AButton.SetIcon

ในขั้นต่อไปเราจะเรียนรู้ว่าเราสามารถส่ง Request ไปยัง Object ให้ทำงานตามที่ต้องการ โดยการใส่รูปแบบ Object และ Request ลงในโครงสร้างคำสั่ง Avenue

โครงสร้างของประโยคคำสั่งใน Avenue

รูปแบบของโครงสร้างคำสั่งใน Avenue เราสามารถพิมพ์คำสั่งให้อยู่ในรูปแบบ Syntax :

Object.Request

นั่นคือเราสามารถกำหนดวัตถุ (Object) และตามด้วยจุด (.) และคำสั่ง (Request) ที่จะให้วัตถุทำงาน ตัวอย่างเช่น View.AddTheme เป็นการให้วัตถุเพิ่มแผนที่ชุดใหม่เข้ามาใน View นี้ รูปแบบนี้อาจเรียกว่า Postfix notation เพราะ Request จะตามหลังวัตถุที่จะกระทำ

Object Request Object

เหมือนให้วัตถุสองตัวกระทำการใดๆ กัน เช่น ทางคณิตศาสตร์ ตัวอย่างเช่น $300 + 700$ รูปแบบนี้อาจจะเรียกว่า Infix notation

Request Object

บางครั้งเราอาจจะทำงานในรูปแบบตรรกศาสตร์ มีค่า True หรือ False เราอาจวางค่า Request ไว้หน้า Object ได้ ในส่วนนี้เรียกว่า Prefix notation ตัวอย่างเช่น not someCondition แต่อย่างไรก็ดีอาจจะวางไว้ในรูปแบบของ Postfix notation ได้คือ someCondition.not

เมื่อมีการวางค่าต่าง ที่เราได้ส่งค่า Request ไปยัง Object เราอาจจะให้ค่าผลลัพธ์ไปไว้ในตัวแปรอื่นๆ หรือวัตถุอื่นๆ เช่น

```
theView = av.GetProject.FindDoc("World")
```

นั่นก็คือ ArcView ก็จะตรวจสอบค่า Project ปัจจุบัน และทำการค้นหา Document ว่ามีที่ใด

มีชื่อว่า "World" หรือไม่ แล้วถ้ามีก็คืนค่า "World" ให้กับตัวแปร theView ถ้าไม่มีก็คืนค่า nil ให้กับ theView นั่นเอง

อาจจะพบรูปแบบนี้ในการประมวลผลข้อมูลแบบตัวเลขในทางคณิตศาสตร์ เช่น

BufferDist = 300 + 700

เป็นการนำค่าตัวเลขสองชุดมารวมกัน แล้วคืนค่าใหม่ให้กับตัวแปร BufferDist นั่นเอง รูปแบบนี้อาจจะเรียกว่า Assignment Statement

บางครั้งการที่มีตัวแปรหลายๆ เราอาจจะแยกหรือรวมตัวแปรเหล่านั้นเข้าด้วยกัน โดยอาศัยโครงสร้างของ syntax : Object.Request ให้มีจำนวน Request หลากๆ ระดับเป็น Object.Request.Request.Request ตัวอย่างเช่น

theProj = av.GetProject

theView = theProj.FindDoc("World")

theTheme = theView.FindTheme("Projected population in 2000")

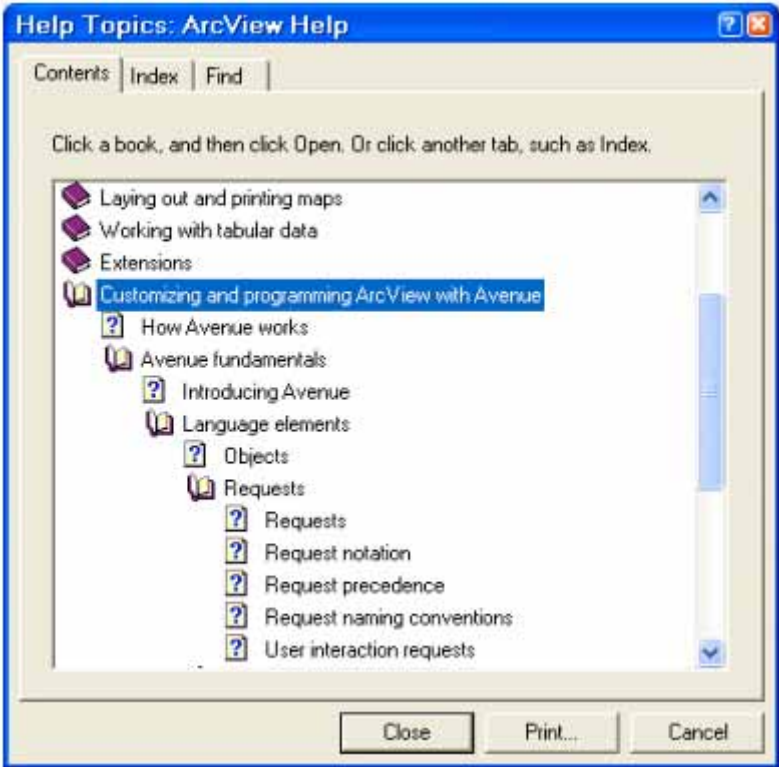
เราอาจจะรวม Request เข้าด้วยกันดังนี้

theTheme = av.GetProject.FindDoc("World").FindTheme("Projected population in 2000")

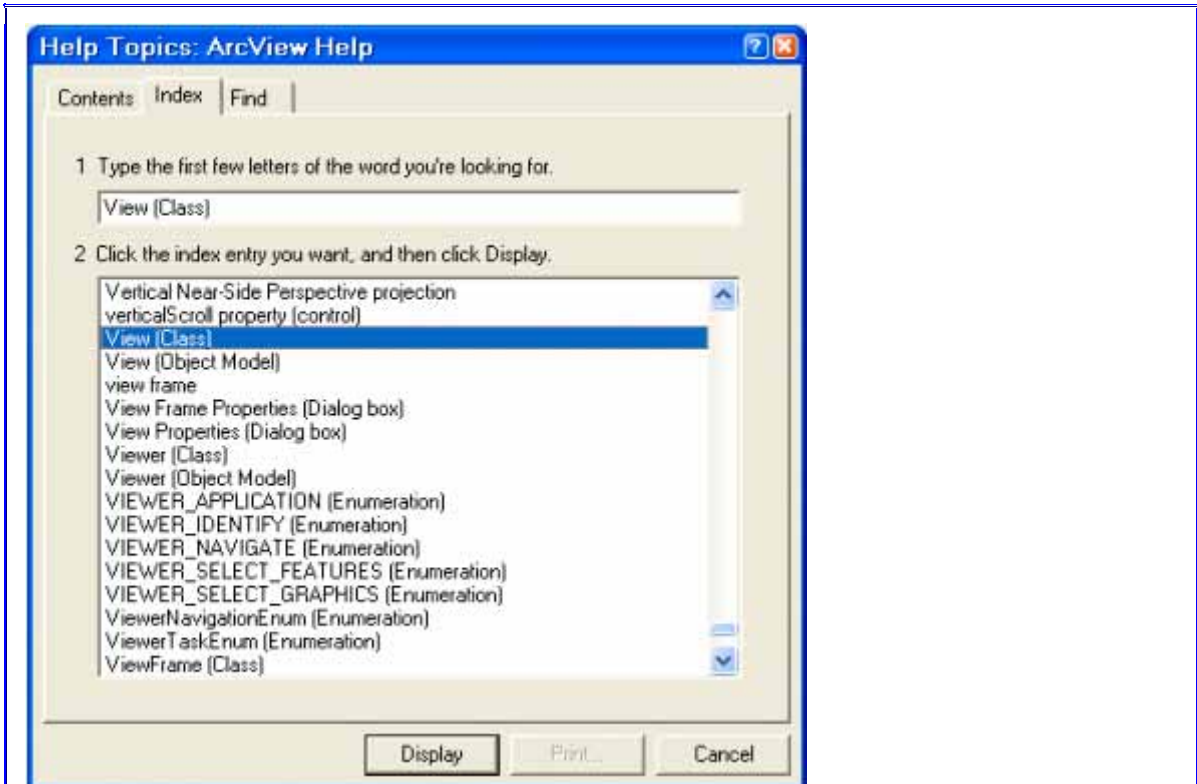
ArcView ก็จะทำกรประมวลผลทุกๆ คำสั่งของโครงสร้าง Object.Request ทีละชุดๆ จากซ้ายไปขวาตามลำดับ โดยคืนค่า Object ให้กับ Request ถัดไปในโครงสร้าง

Help เพื่อการช่วยเหลือในการเรียนรู้ด้วยตนเอง

ในโปรแกรม ArcView จะมีตัวช่วยเหลือในการเรียนรู้การเขียนโปรแกรม Avenue ในโปรแกรมเอง โดยให้เลือกไปที่คำสั่ง Help -> Help Topic เพื่อเรียกแถบที่ชื่อว่า Content แล้วเหลือที่ "Customizing and Programming ArcView with Avenue" ซึ่งจะอ้างอิงในรูปแบบโครงสร้างของการเขียน Avenue ที่ถูกต้อง ทำให้เราเกิดความเข้าใจได้มากขึ้น



หรือถ้าต้องการให้เร็วขึ้นเราอาจจะใช้ Help ช่วยในการเรียกคำสั่งโดยเราอาจจะพิมพ์คำสั่งของ Object ที่เรารู้จักกันเช่น View, Theme, Table, Chart, Layout เป็นต้น ในแถบคำสั่ง Index และจะมีชื่อ Object ตามด้วย class หรือ request ในภายใต้วงเล็บให้ลองคลิกเข้าไปอ่านดู



หรืออาจจะเปิด script ตัวอย่างขึ้นมาดูแล้วถ้าสงสัยคำสั่งใด ให้ลากแถบคำ แล้วเลือกที่ไปปุ่มคำสั่ง Script Help 