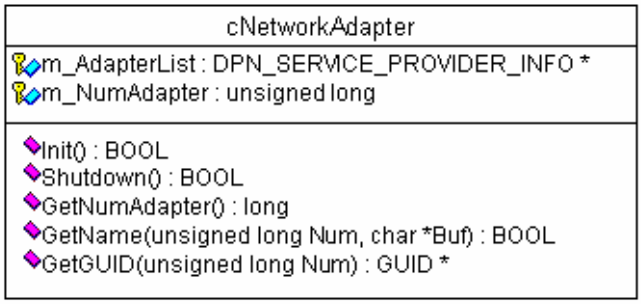
บทที่ 9
เอนจินส่วนเน็ตเวิร์ค

เอนจินส่วนเน็ตเวิร์คจะจัดการติดต่อกับการ์ดเน็ตเวิร์ค โดยจะควบคุมกระบวนการทำงานในการติดต่อสื่อสารกันระหว่างเครื่องคอมพิวเตอร์ และจะจัดการการทำงานในส่วนเซิร์ฟเวอร์ และไคลเอนต์ โดยมีคลาสที่เกี่ยวข้องดังนี้

9.1 คลาส cNetworkAdapter

เป็นคลาสที่จะทำการติดต่อกับการ์ดเน็ตเวิร์ค และจัดการกำหนดลักษณะการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ เพื่อให้สามารถติดต่อสื่อสารกันได้



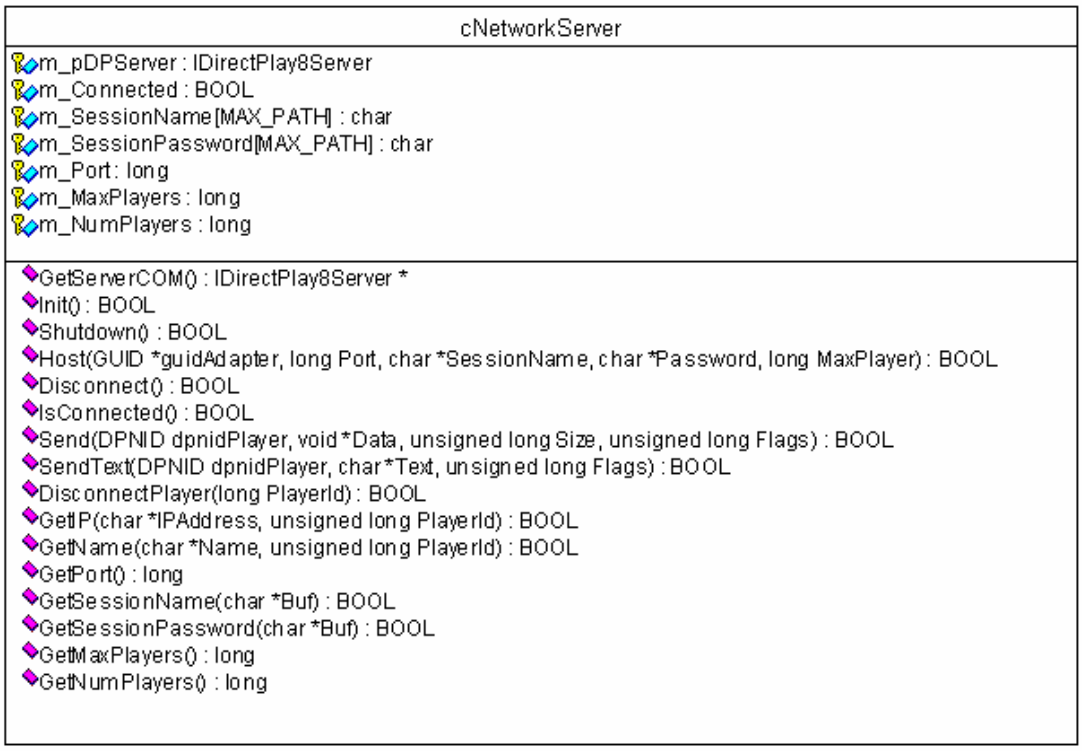
รูปที่ 9-1 คลาสไดอะแกรมของ cNetworkAdapter

คลาสนี้จะทำการติดต่อกับอุปกรณ์เน็ตเวิร์ค โดยผ่านโปรโตคอล TCP/IP ในการใช้งาน การติดต่อกันระหว่างคอมพิวเตอร์นั้น เราจะต้องรู้ GUID โดยเราจะทำการเรียกเมธอด Init() ซึ่งจะทำการกำหนดค่าเริ่มต้นที่จำเป็น และลักษณะการเชื่อมต่อต่างๆ โดยผ่านอินเตอร์เฟสของ DirectPlay ซึ่งเมื่อนำการเรียกเมธอดนี้ ก็จะได้ชื่อของ Adapter และ GUID ซึ่งถ้ามีหลาย Adapter ก็จะทำเก็บจำนวนของ Adapter ที่ตัวแปรชื่อ m_NumAdapter เราจะรู้ GUID ได้โดยใช้เมธอด GetGUID() โดยส่งหมายเลขของ Adapter ที่ต้องการ แล้วจะได้ผลลัพธ์เป็น GUID ของ Adapter ที่ต้องการ และเมื่อใดที่ต้องการเลิกการติดต่อ จะทำการเรียกเมธอด Shutdown() เพื่อทำการยกเลิกการเชื่อมต่อกับระบบเน็ตเวิร์ค

คลาสนี้จะเป็นคลาสหลักที่ใช้เริ่มต้นในการจะทำการติดต่อสื่อสารกันผ่านระบบเน็ตเวิร์ค ซึ่งจำเป็นต้องสร้างขึ้นมาก่อนที่จะทำการกำหนดว่าคอมพิวเตอร์เครื่องใดจะทำหน้าที่เป็นเซิร์ฟเวอร์ หรือไคลเอนต์

9.2 คลาส cNetworkServer

เป็นคลาสที่จะทำหน้าที่เป็นเซิร์ฟเวอร์ ซึ่งจะคอยควบคุมการเชื่อมต่อกันระหว่างไคลเอนต์ ดูแลและจัดการเมสเสจที่เข้าและออก เป็นต้น



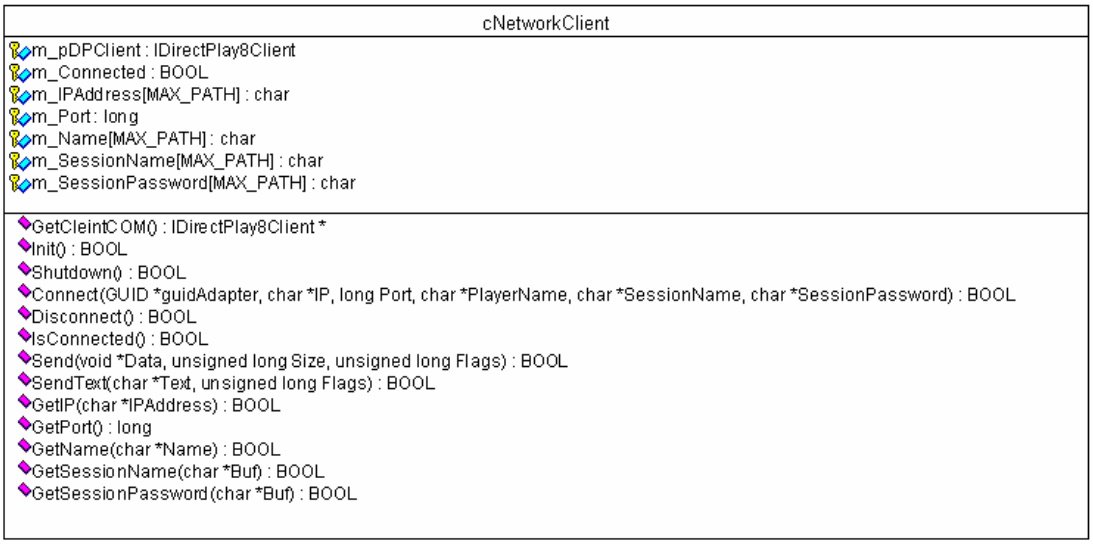
รูปที่ 9-2 คลาสไดอะแกรมของ cNetworkServer

คลาสนี้จะเริ่มใช้งานโดยเรียกเมธอด Init() ซึ่งจะทำการกำหนดค่าเริ่มต้นที่จำเป็นในการใช้งานในการทำงานเป็นเซิร์ฟเวอร์ จากนั้นจะทำการเรียกเมธอด Host() โดยจะส่ง GUID ของ Adapter ที่ต้องการ พอร์ตที่ต้องการใช้ติดต่อ ชื่อและรหัสของ Session ที่ต้องการ ซึ่งเมธอดนี้จะทำการคอยส่งและรับเมสเสจเพื่อทำการประมวลผลเมสเสจที่เข้ามา และทำการทำงานตามลักษณะของเมสเสจที่รับมา เราสามารถตรวจสอบว่ามีการเชื่อมต่ออยู่หรือไม่ โดยใช้เมธอด IsConnected() และเราสามารถส่งเมสเสจไปยังคอมพิวเตอร์เครื่องอื่นได้โดยใช้เมธอด Send() โดยเราจำเป็นต้องรู้ DPNID ของคอมพิวเตอร์ที่จะทำการส่ง ซึ่งเราจะทำรู้ได้จาก Header ของเมสเสจที่รับเข้ามา และเมื่อเราต้องการยกเลิกการทำงานของเซิร์ฟเวอร์ก็จะใช้เมธอด Shutdown() เพื่อยกเลิกการเชื่อมต่อกับคอมพิวเตอร์อื่นๆ

การใช้งานนั้นจะมีการกำหนดจำนวนผู้ใช้ไว้ เพื่อไม่ให้มีผู้ใช้มากเกินที่เซิร์ฟเวอร์จะรับได้ ซึ่งจะช่วยให้ไม่เกิดความล่าช้าในการประมวลผลเมสเสจ และสามารถตอบรับกลับไปยังไคลเอนต์ได้อย่างทันที เรายังสามารถดูจำนวนของผู้ใช้ที่ทำการเชื่อมต่อได้โดยใช้เมธอด GetNumPlayer() ซึ่งจะคืนค่าจำนวนผู้ใช้ที่เชื่อมต่ออยู่ในขณะนั้น

9.3 คลาส cNetworkClient

ทำหน้าที่เป็นไคลเอนต์ ซึ่งจะทำการส่งและรับเมสเสจจากเซิร์ฟเวอร์ โดยจะทำการเชื่อมต่อกับเซิร์ฟเวอร์ เพื่อสื่อสารกับไคลเอนต์อื่นๆ หรือเพื่อต้องการให้เซิร์ฟเวอร์ทำงานให้



รูปที่ 9-3 คลาสไดอะแกรมของ cNetworkClient

คลาสจะทำการเริ่มใช้งานโดยทำการเรียกเมธอด Init() เหมือนคลาส cNetworkServer และจะเริ่มทำการเชื่อมต่อไปยังเซิร์ฟเวอร์โดยใช้เมธอด Connect() ซึ่งจะต้องบอกหมายเลข IP ของเซิร์ฟเวอร์และบอกหมายเลขพอร์ตที่จะทำการเชื่อมต่อ จากนั้นบอกชื่อของ Session ที่ต้องการเข้าร่วมและถ้า Session นั้นมีรหัสในการเข้าร่วมก็ให้ใส่เข้าไปด้วย จากนั้นก็จะทำการเชื่อมต่อไปยังเซิร์ฟเวอร์นั้น ถ้าต้องการตรวจสอบว่ามีการเชื่อมต่ออยู่หรือไม่ จะใช้เมธอด IsConnected() เพื่อช่วยในการตรวจสอบ และถ้าต้องการยกเลิกการเชื่อมต่อกับเซิร์ฟเวอร์ ก็จะทำการเรียกใช้เมธอด Disconnect()

ในการส่งเมสเสจจะใช้เมธอดอยู่ 2 เมธอดก็คือ ถ้าต้องการส่งข้อมูลที่เป็นรูปแบบตามที่ต้องการ ก็จะใช้เมธอด Send() ซึ่งจำเป็นต้องบอกขนาดของเมสเสจด้วย แต่ถ้าต้องการส่งเพียงข้อความอย่างเดียวจะใช้เมธอด SendText() ซึ่งจะทำให้สะดวกต่อการส่งมากกว่าใช้เมธอด Send()

ในการใช้งานคลาส cNetworkServer และคลาส cNetworkClient นั้น จะต้องมีการสืบทอดคลาสดังกล่าวมาเป็นคลาสใหม่ เนื่องจากจำเป็นต้องมีการประมวลผลจัดการกับเมสเสจตามที่ผู้ใช้ต้องการ ดังนั้นการสืบทอดคลาสนั้นจะต้องทำการสร้างเมธอดใหม่ขึ้นมา เพื่อที่จะจัดการกับรูปแบบของเมสเสจที่ผู้ใช้ต้องการ ซึ่งส่วนใหญ่จะสร้างโดยให้มีชื่อเมธอดเป็น Receive() โดยจะรับเมสเสจมา และทำการตรวจสอบว่าเป็นเมสเสจชนิดใด และทำการประมวลผลเมสเสจ จากนั้นอาจทำการตอบสนองเมสเสจหรือทำการอย่างใดอย่างหนึ่งตามที่ผู้พัฒนาต้องการตามชนิดของเมสเสจนั้น