

IBM

@server

iSeries

การจัดระบบงาน

เวอร์ชัน 5 รีลีส 3







@server

iSeries

การจัดระบบงาน

เวอร์ชัน 5 รัสเซีย 3

**หมายเหตุ**

ก่อนใช้ข้อมูลและผลิตภัณฑ์ที่สนับสนุนนี้, ต้องแน่ใจว่าได้อ่านข้อมูลใน “ประกาศ”, ในหน้า 91.

**พิมพ์ครั้งที่สี่ (สิงหาคม 2548)**

การพิมพ์ครั้งนี้ใช้ได้กับเวอร์ชัน 5, ซีรีส์ 3, โมดิฟิเคชัน 0 ของ IBM Operating System/400 (หมายเลขผลิตภัณฑ์ 5722-SS1) และใช้กับซีรีส์และโมดิฟิเคชันถัดจากนี้ไป จนกว่าจะมีการระบุเป็นอย่างอื่นในการพิมพ์ครั้งใหม่. เวอร์ชันนี้ไม่สามารถรันบนรุ่น reduced instruction set computer (RISC) ได้ทุกรุ่น รวมทั้งไม่สามารถรันบนรุ่น CISC.

© ลิขสิทธิ์ของ International Business Machines Corporation 1998, 2005. สงวนลิขสิทธิ์ทั้งหมด.

# สารบัญ

การจัดการระบบงาน . . . . .	1	งาน . . . . .	35
สิ่งที่เพิ่มมาใหม่ใน V5R3 . . . . .	2	Job description . . . . .	35
พิมพ์หัวข้อนี้ . . . . .	3	งานแอคทีฟและ inactive . . . . .	36
การจัดการงานประจำวัน . . . . .	4	งานที่แอคทีฟ . . . . .	36
การมอนิเตอร์ activity ระบบ . . . . .	6	งานที่ inactive . . . . .	36
การทำงานกับสถานะระบบ . . . . .	6	ประเภทของงาน . . . . .	36
การจัดการงานและ thread . . . . .	7	งานแบบ autostart . . . . .	37
การกำหนดตารางการทำงาน . . . . .	8	งานแบบแบตช์ . . . . .	37
เปรียบเทียบตารางเวลางาน . . . . .	8	งานสื่อสาร . . . . .	38
การค้นหางานบนเซิร์ฟเวอร์ iSeries . . . . .	10	งานโต้ตอบ . . . . .	38
การกำหนดสถานะของงาน . . . . .	13	งาน Prestart . . . . .	39
การดูสถิติประสิทธิภาพการทำงานของงาน . . . . .	14	งาน reader และ writer . . . . .	39
การเรียกดูข้อมูลที่มีความสัมพันธ์กันของงาน . . . . .	16	งานระบบย่อย . . . . .	39
การสิ้นสุดการทำงาน . . . . .	16	งานระบบ . . . . .	40
การปฏิบัติงาน . . . . .	17	งานเซิร์ฟเวอร์ . . . . .	44
การดู threads รั้นภายใต้งานเฉพาะ . . . . .	19	ข้อความ . . . . .	45
การดูคุณสมบัติ thread . . . . .	19	คุณสมบัติของงาน . . . . .	46
การลบหรือจบการทำงานของ thread . . . . .	19	สิทธิในการใช้งานที่เหมาะสม . . . . .	48
การจัดการคิวงาน . . . . .	20	Job control special authority (*JOBCTL) . . . . .	48
การดูงานบนคิวงาน . . . . .	20	การดึงพริ้นเตอร์เอาต์พุตออกมา . . . . .	48
การเปลี่ยนระดับความสำคัญของงานภายในคิวงาน	20	สถิติการทำงานที่ผ่านมา . . . . .	48
การย้ายงานไปยังคิวงานอื่น . . . . .	21	สถานะโดยละเอียด . . . . .	49
การจัดการระบบย่อย . . . . .	24	สิ้นสุดการทำงาน . . . . .	49
การมอนิเตอร์ระบบย่อย . . . . .	24	รายละเอียด: การปฏิบัติงานแอคทีฟ . . . . .	50
การดูงานในระบบย่อย . . . . .	25	อ็อบเจกต์ที่ถูกล็อก . . . . .	51
การเริ่มใช้งานระบบย่อย . . . . .	25	บันทึกการใช้งาน . . . . .	52
หยุดระบบย่อย . . . . .	25	thread . . . . .	53
การจัดการพูลหน่วยความจำ . . . . .	26	การดำเนินการของ thread . . . . .	53
การมอนิเตอร์จำนวนงานในพูลหน่วยความจำ . . . . .	26	คุณสมบัติ thread . . . . .	54
การมอนิเตอร์จำนวนระบบย่อยโดยใช้พูลหน่วยความ		สิทธิในการใช้งานที่ต้องการสำหรับ thread . . . . .	55
จำ . . . . .	28	Thread Control . . . . .	55
ตรวจสอบการใช้พูลหน่วยความจำ . . . . .	28	ชนิด thread . . . . .	55
การเปลี่ยนขนาดของพูลหน่วยความจำ . . . . .	29	สถานะ thread . . . . .	55
การจัดการบันทึกการใช้งาน . . . . .	31	คิวงาน . . . . .	56
การเข้าถึงบันทึกการทำงานสำหรับงานแอคทีฟ, รวม		รายการเรียงลำดับ . . . . .	57
ถึงงานเซิร์ฟเวอร์ . . . . .	31	วิธีการทำงานของคิวงาน . . . . .	57
การแอ็กเซสพริ้นเตอร์เอาต์พุต . . . . .	32	ระบบย่อย . . . . .	58
การจัดการเอาต์พุตคิว . . . . .	32	Subsystem description . . . . .	59
การดูเอาต์พุตคิวบนระบบ . . . . .	33	ระบบย่อยที่มีมาพร้อมกับระบบ . . . . .	70
ย้ายเอาต์พุตระหว่างและภายในเอาต์พุตคิว . . . . .	33	ระบบย่อยที่ผู้ใช้กำหนดเอง . . . . .	71
เคลียร์เอาต์พุตคิว . . . . .	34	คุณสมบัติของระบบย่อย . . . . .	72
โครงสร้างของระบบ . . . . .	34	วงรอบชีวิตของระบบย่อย . . . . .	73
		สิ่งที่เกิดขึ้นเมื่อระบบย่อยเริ่มทำงาน . . . . .	74

พุลหน่วยความจำ . . . . .	75
activity level ของพุลหน่วยความจำ . . . . .	75
ประเภทของพุลหน่วยความจำ . . . . .	76
พุลหน่วยความจำพื้นฐาน . . . . .	77
พุลหน่วยความจำของเครื่อง . . . . .	77
พุลแบบใช้งานร่วมกันทั่วไป . . . . .	77
เอาต์พุตคิว . . . . .	77
แอ็ททริบิวต์ของเอาต์พุตคิว . . . . .	79
ลำดับของไฟล์ . . . . .	79
สถานะของพริเตอร์เอาต์พุต . . . . .	80
วิธีการทำให้งานเสร็จสมบูรณ์ . . . . .	81
งานคืออะไร . . . . .	81

สิ่งที่เกิดขึ้นก่อนงานเข้าสู่ระบบ . . . . .	82
วิธีที่การทำงานจะเข้าสู่ระบบ . . . . .	82
การทำงานถูกประมวลผลอย่างไร . . . . .	83
งานออกจากระบบอย่างไร . . . . .	84
การจัดการการแก้ไขปัญหาทางาน . . . . .	84
งานเกิดหยุดการทำงาน . . . . .	84
งานของเงินประสบปัญหาทำงานช้า . . . . .	86
ข้อมูลที่เกี่ยวข้องกับการจัดการระบบงาน . . . . .	87
<b>ภาคผนวก. ประกาศ. . . . .</b>	<b>91</b>
เครื่องหมายการค้า . . . . .	93
ข้อกำหนดและเงื่อนไขในการดาวน์โหลดและพิมพ์หนังสือนี้ . . . . .	94

---

## การจัดการระบบงาน

การจัดการระบบงานเป็นองค์ประกอบพื้นฐานที่สำคัญภายในระบบปฏิบัติการของเซิร์ฟเวอร์ iSeries<sup>(TM)</sup>. ฟังก์ชันของการจัดการระบบงานเป็นพื้นฐานของการประมวลผลงาน, การรันงาน, และการจบงานที่ผ่านเข้ามาในระบบบนเซิร์ฟเวอร์ iSeries. ไม่ว่าคุณจะรันงานเป็นแบทช์ต่างๆ สัปดาห์ละครั้ง หรือจะเป็นการเรียกใช้แอปพลิเคชันประจำวัน (เช่น Lotus Notes<sup>(R)</sup>) ก็ตาม, การจัดการระบบงาน จะช่วยในการบริหารจัดการงานและอ็อบเจกต์ต่างๆ ที่รันบนระบบของคุณ. นอกจากนี้ ยังรองรับคำสั่งและฟังก์ชันภายในที่จำเป็นในการควบคุมการปฏิบัติการของระบบ และจัดสรรรีซอร์สให้กับแอปพลิเคชันเมื่อต้องการอีกด้วย.

เซิร์ฟเวอร์ iSeries จะถูกเชื่อมต่อและพร้อมใช้งาน. ผู้ใช้งานส่วนใหญ่ไม่จำเป็นต้องเปลี่ยนค่ากำหนดดีฟอลต์แต่ประการใด. อย่างไรก็ตาม, หากคุณต้องการจะปรับแต่งการจัดการระบบงานให้เหมาะกับบริษัทของคุณ, คุณจะต้องทำความเข้าใจกับคำศัพท์และหลักการต่างๆ ที่เกี่ยวข้อง รวมถึงวิธีการที่ส่วนประกอบเหล่านี้จะทำงานร่วมกันเพื่อให้ เซิร์ฟเวอร์ iSeries ทำงานได้ประสิทธิภาพสูงสุด.

» ในอีกทางหนึ่ง, คุณสามารถทำงานกับส่วนต่างๆ ขององค์ประกอบของการจัดการระบบงานโดยการใช้งาน iSeries Navigator ที่ทำงานบนเว็บ. โดยจะยอมให้คุณทำงานกับฟังก์ชันการจัดการระบบงานโดยใช้เว็บเบราว์เซอร์. สำหรับข้อมูลเพิ่มเติม, ให้ดูใน การทำงานบนเว็บของ iSeries Navigator. <<

ไม่ว่าคุณจะเชี่ยวชาญในการใช้เซิร์ฟเวอร์ iSeries หรือเพิ่งหัดใช้ก็ตาม, บทความต่อไปนี้จะช่วยให้คุณเข้าใจภาพการทำงานของ การจัดการระบบงานได้อย่างง่ายดาย. หัวข้อนี้มี entry point ต่างๆ, เพื่อให้คุณสามารถเลือกจุดเพื่อเริ่มต้นเรียนรู้เกี่ยวกับการจัดการระบบงาน.

### ช่วงชีวิตของงาน

ลองติดตามดูงานตลอดวงจรชีวิตของงาน ในโครงสร้างการจัดการระบบงาน — โดยใช้กราฟิกแบบโต้ตอบเพื่อนำคุณไปยังข้อมูลที่ยังรายละเอียดเกี่ยวกับการจัดการ ระบบงานที่มากขึ้น.

### “การจัดการงานประจำวัน” ในหน้า 4

เรียนรู้งานประจำวันที่คุณสามารถดำเนินการ เพื่อจัดการงานอย่างมีประสิทธิภาพด้วย iSeries Navigator และระยะเวลาอันเหมาะสมที่จะทำงานเหล่านั้น. จากการตรวจสอบ บันทึกการใช้งานไปจนถึงการมอนิเตอร์การทำงานของระบบ, คุณจะได้เรียนรู้ภารกิจที่สำคัญประจำวันซึ่งเกี่ยวข้องกับ การจัดการระบบงาน.

### “โครงสร้างของระบบ” ในหน้า 34

เรียนรู้จักกับคำศัพท์และหลักการที่เกี่ยวข้องกับการจัดการระบบงาน (รวมถึง งาน, คิวงาน, ระบบย่อย, และพูลหน่วยความจำ) ที่คุณสามารถใช้เพื่อจัดการงานบนเซิร์ฟเวอร์ iSeries.

### “วิธีการทำให้งานเสร็จสมบูรณ์” ในหน้า 81

เรียนรู้จักวิธีการทำให้งานเสร็จสมบูรณ์บนเซิร์ฟเวอร์ iSeries ของคุณ. จัดเตรียมคิวงาน, จัดสรรหน่วยความจำให้กับระบบย่อย และ เรียนรู้ว่าจะเกิดอะไรขึ้นหลังจากงานรันเสร็จแล้ว.

### “การจัดการการแก้ไขปัญหางาน” ในหน้า 84

เรียนรู้วิธีการจัดการแก้ไขปัญหที่เกิดขึ้นกับงานผ่านทาง iSeries Navigator.

โปรดดูที่หัวข้อ “สิ่งที่เพิ่มมาใหม่ใน V5R3” สำหรับข้อมูลใหม่และข้อมูลที่เปลี่ยนแปลงไปและโปรดดูหัวข้อ “พิมพ์หัวข้อนี้” ในหน้า 3 หากคุณต้องการพิมพ์หัวข้อนี้ทั้งหมดเป็น PDF.

“ข้อมูลที่เกี่ยวข้องกับการจัดการระบบงาน” ในหน้า 87

คู่มือของ IBM<sup>(R)</sup> ประกอบด้วยรายละเอียดทางเทคนิค, ความรู้, และ “วิธีการทำ” ต่างๆ.

---

## สิ่งที่เพิ่มมาใหม่ใน V5R3

ใน V5R3, มีฟังก์ชันใหม่ๆ เพิ่มเติมลงไปยังส่วนประกอบการจัดการระบบงานใน iSeries<sup>(TM)</sup> Navigator. คุณลักษณะและฟังก์ชันใหม่เหล่านี้ถูกรวมเป็นโครงสร้างการจัดการระบบงาน, ดังนั้น คุณจึงสามารถเลือกจุดที่คุณต้องการเริ่มศึกษาเกี่ยวกับส่วนประกอบการจัดการระบบงาน: ช่วงชีวิตของงาน (กราฟิกแบบโต้ตอบ), จัดการงานประจำวัน, โครงสร้างเซิร์ฟเวอร์ iSeries, และวิธีการทำงานให้เสร็จสมบูรณ์. ส่วนต่างๆ เหล่านี้มีระดับความยากในการทำความเข้าใจเรื่องการจัดการระบบงานหลายระดับด้วยกัน. ไม่ว่าคุณจะเชี่ยวชาญในการใช้เซิร์ฟเวอร์ iSeries หรือเพิ่งหัดใช้ก็ตาม, บทความต่อไปนี้จะช่วยให้คุณเข้าใจภาพการทำงานของจัดการระบบงานได้อย่างง่ายดาย.

มีการปรับปรุงฟังก์ชันและขั้นตอนการทำงานในการจัดการระบบงาน. ด้านล่างนี้คือรายการของฟังก์ชันในการจัดการระบบงาน รวมถึงสิ่งที่ได้ปรับปรุงใหม่ใน V5R3.

### การใช้งานผ่านเว็บ

คุณสามารถทำงานกับบางรายการของ การจัดการระบบงาน ผ่านทาง iSeries Navigator โดยใช้เว็บ. คุณสามารถใช้เว็บเบราว์เซอร์กับรายการต่อไปนี้:

- “การจัดการงานและ thread” ในหน้า 7
- “การจัดการระบบย่อย” ในหน้า 24
- “การจัดการเอาต์พุตคิว” ในหน้า 32

### “การทำงานกับสถานะระบบ” ในหน้า 6

- มีการยกเลิกความสามารถในการใช้ไดอะล็อก **Configure Logical Partitions** โดยตรงจาก System Status เพื่อลดข้อจำกัดทางฮาร์ดแวร์. คุณยังสามารถตั้งค่า โลจิคัลพาร์ติชัน ได้สองทางทั้งนี้ขึ้นกับฮาร์ดแวร์ของคุณ. ถ้าคุณใช้เครื่องรุ่น 8xx หรือเก่ากว่านั้น, คุณสามารถตั้งค่า โลจิคัลพาร์ติชันผ่าน iSeries Navigator โดยเลือกที่ระบบของคุณ → **Configuration and Service** → **Logical Partitions**. สำหรับกรณีอื่น, คุณสามารถตั้งค่า โลจิคัลพาร์ติชันโดยใช้ Hardware Management Console สำหรับ eServer.
- สามารถระบุข้อมูลเพิ่มเติมด้าน ประเภทตัวประมวลผล ตามความเหมาะสม. ทั้งนี้ขึ้นกับ configuration ของฮาร์ดแวร์คุณ, คุณสามารถกำหนดประเภทของตัวประมวลผลเป็น dedicated, shared-capped, หรือ shared-uncapped.
- สามารถระบุเปอร์เซ็นต์การใช้งานของพูลตัวประมวลผลร่วมของระบบ.
- สามารถระบุเปอร์เซ็นต์การใช้งานของ uncapped CPU capacity, ถ้าฮาร์ดแวร์นั้นสนับสนุนตัวประมวลผลแบบ shared-uncapped.

### งาน

- “รายละเอียด: การปฏิบัติงานแอ็คทีฟ” ในหน้า 50 เพิ่มเติม:



การสนับสนุนด้านการเปิดไฟล์ได้ขยายขอบเขตโดยรวมเอาความสามารถในการทำงานกับ library object ของงาน และ file system object เข้ามา. ก่อนหน้า V5R3, คุณสามารถทำงานกับ library object ของงานเท่านั้น .

- คุณสมบัติงานเพิ่มเติม:

สามารถดูวันและเวลาโลคัลของงานที่หน้า **Date/Time**.

สามารถดูข้อมูลเขตเวลาของงานที่หน้า **Date/Time** .

สามารถดูค่า **Offset from coordinated universal time (UTC)** ของงานที่หน้า **Date/Time**.

“การเรียกดูข้อมูลที่มีความสัมพันธ์กันของงาน” ในหน้า 16 ที่หน้า **Resources**.

### ข้อความบันทึกการใช้งาน

- มีการปรับปรุงด้านประโยชน์ใช้สอยของข้อความในบันทึกการใช้งาน, โดยการเพิ่มฟิลด์, **From user**, ที่แทนค่าโปรไฟล์ของผู้ส่งข้อความเข้าไป.

### “Subsystem description” ในหน้า 59

- มีการอัปเดตระบบย่อย QSYSWRK และ QUSRWRK เพื่อสนับสนุนการย้าย Electronic Service Agent ไปที่ระบบปฏิบัติการพื้นฐาน. นอกจากนี้, ยังมีการอัปเดตระบบย่อยดังกล่าวเพื่อให้สอดคล้องกับการปรับปรุงด้านฟังก์ชันคลัสเตอร์ในซอฟต์แวร์ iSeries.

### บันทึกจากประสบการณ์

- บันทึกจากประสบการณ์, โดยนักพัฒนาของ IBM<sup>(R)</sup>, เป็นการบันทึกจากประสบการณ์ในการลงมือปฏิบัติจริงในสภาพแวดล้อมและโซลูชันที่ใช้จริง. ใช้บันทึกนี้ในการเรียนรู้จากประสบการณ์ของนักพัฒนาของ IBM<sup>(R)</sup> โดยวิธีการปฏิบัติที่เน้นเฉพาะโซลูชันของ iSeries<sup>(TM)</sup>, อันสมบูรณ์แบบด้วยคำแนะนำในทุกขั้นตอน. สำหรับหัวข้อที่เกี่ยวกับการจัดการระบบงาน ในบันทึกนี้, ดูที่ “ข้อมูลที่เกี่ยวข้องกับการจัดการระบบงาน” ในหน้า 87.

### วิธีการดูว่ามีอะไรใหม่หรือมีอะไรที่เปลี่ยนแปลงไป

เพื่อช่วยให้คุณสามารถดูได้ว่าการเปลี่ยนแปลงทางเทคนิคที่ใดบ้าง, ข้อมูลนี้ใช้:

- รูปภาพ **»** เพื่อทำเครื่องหมายว่าข้อมูลใหม่หรือข้อมูลที่ถูกเปลี่ยนแปลงเริ่มต้นที่ใด.
- รูปภาพ **«** เพื่อทำเครื่องหมายว่าข้อมูลใหม่หรือข้อมูลที่ถูกเปลี่ยนแปลงสิ้นสุดที่ใด.

**»** หากต้องการหาข้อมูลอื่นเกี่ยวกับสิ่งใหม่ๆ หรือสิ่งที่เปลี่ยนแปลงในรีลีสนี้, ให้อ่าน บันทึกถึงผู้ใช้. **«**

---

## พิมพ์หัวข้อนี้

เพื่ออ่านหรือดาวน์โหลดเวอร์ชัน PDF ของเอกสารนี้, เลือก การจัดการระบบงาน (ขนาดประมาณ 660 กิโลไบต์).

คุณสามารถอ่านหรือดาวน์โหลดหัวข้อที่เกี่ยวข้องเหล่านี้:

- **»** การจัดเวลางานขั้นสูง **«**
- ค่ากำหนดของระบบ



### รายละเอียดอื่นๆ

คุณสามารถดูหรือพิมพ์คู่มือ V4R5 Work Management ในแบบ PDF ได้:




- คู่มือ V4R5 Work Management  (ประมาณ 2720 KB หรือ 573 หน้า)

## การบันทึกไฟล์ PDF

ในการเซฟไฟล์ PDF ไว้ที่เวิร์กสแตชันของคุณเพื่อใช้สำหรับดูหรือพิมพ์:

1. ให้คลิกขวาที่ PDF บนเบราว์เซอร์ของคุณ (คลิกขวาที่ลิงก์ด้านบน).
2.  คลิก **Save Target As...** กรณีที่คุณใช้ Internet Explorer. คลิก **Save Link As...** กรณีที่คุณใช้ Netscape Communicator.  

3. เปลี่ยนไดเรกทอรีไปยังไดเรกทอรีที่คุณต้องการเซฟไฟล์ PDF.
4. คลิก **Save**.

## ดาวน์โหลด Adobe Acrobat Reader

 คุณต้องการ Adobe Acrobat Reader ในการอ่านหรือพิมพ์ไฟล์ PDF. คุณสามารถดาวน์โหลดโปรแกรมได้จาก เว็บไซต์ของ Adobe ([www.adobe.com/products/acrobat/readstep.html](http://www.adobe.com/products/acrobat/readstep.html))  .

---

## การจัดการงานประจำวัน

หน้าที่สำคัญของผู้ควบคุมระบบหรือผู้บริหารระบบ คือ การดูแลให้เซิร์ฟเวอร์ทำงานโดยราบรื่น. นั่นหมายถึงการคอยการตรวจสอบ, จัดการ และตรวจตราดูว่า งาน, คิวงาน, ระบบย่อย, พูลหน่วยความจำ, เอาต์พุตคิว และบันทึกข้อความต่างๆ ทำงานเป็นปกติดี.

หัวข้อในส่วนนี้จะอธิบายภารกิจของการจัดการระบบงานรายวันประเภทต่างๆ รวมถึงภารกิจอื่นๆ อีกที่อาจจำเป็นต้องทำบนเครื่องเซิร์ฟเวอร์ iSeries ของคุณ. ในหัวข้อย่อยแต่ละอันจะบอกถึงความสำคัญของภารกิจแต่ละอย่าง, รวมถึงวิธีการทำงานให้ภารกิจสำเร็จอีกด้วย.

“การมอนิเตอร์ activity ระบบ” ในหน้า 6

การตรวจสอบระบบเป็น activity ประจำวันที่สำคัญมาก. คุณสามารถตรวจสอบให้เสร็จสมบูรณ์ได้หลายวิธี, เช่นโดยใช้ iSeries Navigator และ iSeries Navigator Management Central. ภารกิจที่คุณจะได้เรียนรู้ในหัวข้อย่อยนี้คือ:

- การทำงานกับสถานะระบบ
- มอนิเตอร์ดูการทำงานของระบบ
- การทำงานกับมอนิเตอร์

“การจัดการงานและ thread” ในหน้า 7

ไม่ว่าคุณจะต้องรายงานสถานะของงานชิ้นหนึ่งๆ หรือ thread หรือ คอยตรวจสอบการทำงานของงาน หรือ thread , คุณก็สามารถหาคำตอบส่วนใหญ่ที่คุณต้องการได้อย่างง่ายดายผ่านทาง iSeries Navigator. ภารกิจที่คุณจะได้เรียนรู้ในหัวข้อย่อยนี้คือ:

- การกำหนดตารางการทำงาน
- การค้นหางานที่อยู่บนเซิร์ฟเวอร์ iSeries
- การกำหนดสถานะของงาน

- การดูสถิติประสิทธิภาพการทำงานของงาน
- >> การดูข้อมูลข้างเคียง<<
- การสิ้นสุดการทำงาน
- สิ่งที่ได้ปฏิบัติงานกับงานนั้น
- การดู threads วันภายใต้งานเฉพาะ
- การดูคุณสมบัติ thread
- การสิ้นสุด thread

#### “การจัดการคิวงาน” ในหน้า 20

คิวงานเป็นองค์ประกอบสำคัญในวงรอบชีวิตของงานแบบเบตซ์. คิวงานช่วยควบคุม อัตราการเข้าสู่ระบบย่อยของงานแบบเบตซ์. ภารกิจที่คุณจะได้เรียนรู้ในหัวข้อย่อยนี้คือ:

- การดูงานบนคิวงาน
- การเปลี่ยนระดับความสำคัญของงานภายในคิวงาน
- การย้ายงานไปยังคิวงานอื่น

#### “การจัดการระบบย่อย” ในหน้า 24

เนื่องจากงานรันอยู่บนระบบย่อย, จึงจำเป็นที่จะต้องตรวจสอบ activity ของระบบย่อยเพื่อป้องกันปัญหาที่อาจมีผลกระทบต่อความสามารถในการรันงาน. ภารกิจที่คุณจะได้เรียนรู้ในหัวข้อย่อยนี้คือ:

- การตรวจสอบระบบย่อย
- การดูงานในระบบย่อย
- สตาร์ทระบบย่อย
- การหยุดการทำงานของระบบย่อย

#### “การจัดการพูลหน่วยความจำ” ในหน้า 26

พูลหน่วยความจำจะจัดสรรหน่วยความจำให้กับระบบย่อยเพื่อให้งานสามารถรันไปได้. ดังนั้น จึงเป็นสิ่งสำคัญที่งานจะต้องได้หน่วยความจำอย่างเพียงพอเพื่อให้สามารถรันได้โดยสมบูรณ์. ภารกิจที่คุณจะได้เรียนรู้ในหัวข้อย่อยนี้คือ:

- การตรวจสอบจำนวนงานในพูลหน่วยความจำ
- การตรวจสอบจำนวนของระบบย่อยในพูลหน่วยความจำ
- ตรวจสอบการใช้พูลหน่วยความจำ
- การเปลี่ยนขนาดของพูลหน่วยความจำ

#### “การจัดการบันทึกการใช้งาน” ในหน้า 31

บันทึกการใช้งานประกอบด้วยข้อมูลต่างๆ เกี่ยวกับ request ที่ส่งเข้ามาในงาน เช่น คำสั่งที่ใช้ในงาน, คำสั่งที่ใช้ในโปรแกรม, และข้อความต่างๆ. ภารกิจที่คุณจะได้เรียนรู้ในหัวข้อย่อยนี้คือ:

- การเข้าถึงบันทึกการทำงานสำหรับงานแอ็คทีฟ, รวมถึงงานเซิร์ฟเวอร์
- การเข้าใช้งานพรินเตอร์เอาต์พุต

“การจัดการเอาต์พุตคิว” ในหน้า 32

เอาต์พุตคิวจะช่วยคุณจัดการพรินเตอร์เอาต์พุตที่สร้างขึ้นเมื่อสิ้นสุดการทำงาน. จึงสำคัญที่ต้องเข้าใจวิธีการรักษาเอาต์พุตคิวของคุณอย่างมีประสิทธิภาพเพื่อเอาต์พุตแบบพิมพ์ของท่านจะได้รับการประมวลผลอย่างรวดเร็ว. การกิจที่คุณจะได้เรียนรู้ในหัวข้อย่อหน้านี้คือ:

- ดูเอาต์พุตคิวบนระบบ
- เคลียร์เอาต์พุตคิว
- ย้ายเอาต์พุตระหว่างและภายในเอาต์พุตคิว

## การมอนิเตอร์ activity ระบบ

การมอนิเตอร์ activity ระบบเป็นภารกิจประจำวันอันหนึ่งที่มีความสำคัญของผู้บริหารระบบ. การมอนิเตอร์การไหลของงานผ่านระบบเป็นข้อมูลชิ้นหนึ่งที่สมควรได้รับการมอนิเตอร์อยู่เป็นประจำทุกวัน. IBM ได้จัดเตรียมทูลต่างๆ เพื่อช่วยให้คุณมอนิเตอร์การทำงานของระบบคุณตั้งแต่การตรวจสอบระบบพื้นฐานโดยใช้สถานะระบบไปจนถึงการมอนิเตอร์ระบบขั้นสูงด้วย Management Central.

“การทำงานกับสถานะระบบ”

ใน iSeries Navigator, หน้าต่างสถานะระบบจะช่วยให้คุณสามารถดูและแอ็กเซสฟังก์ชันระบบต่างๆ บนระบบในตำแหน่งที่สะดวกหนึ่งตำแหน่ง.

การจัดการประสิทธิภาพการทำงาน iSeries

ฟังก์ชัน Management Central ใน iSeries Navigator มีการมอนิเตอร์ระบบที่รวบรวมและแสดงข้อมูลประสิทธิภาพการทำงานเป็นเวลาจริง ซึ่งคุณสามารถตรวจสอบและแก้ปัญหาการทำงานของระบบได้.

การทำงานกับมอนิเตอร์

มอนิเตอร์งานและเซิร์ฟเวอร์ของคุณ, message queues, ทำการเปลี่ยนแปลงกับไฟล์ที่เลือกไว้, และการทำธุรกรรมระหว่างองค์กร.

## การทำงานกับสถานะระบบ

เพราะถูกออกแบบตามต้นฉบับของหน้าจอ Work with System Status (WRKSYSSTS) ในอินเตอร์เฟซที่ใช้อักขระ ไดอะล๊อก System Status จึงมีวิธีที่ง่ายและรวดเร็วในการตรวจสอบสถานะของระบบ. จาก Management Central คุณสามารถตรวจสอบฟังก์ชันต่างๆ ได้อย่างละเอียดผ่านทาง การมอนิเตอร์ระบบ.

ฟังก์ชันอื่นๆ ที่คุณสามารถใช้งานได้จากหน้าต่างสถานะระบบคือ:

- ดูการใช้งาน CPU
- ดูจำนวนรวมของงาน, งานแอ็กทีฟ, และจำนวนงานสูงสุดที่อนุญาตบนระบบ
- ดูจำนวนของ “thread” ในหน้า 53 แอ็กทีฟบนระบบ
- ดูอัตราส่วนของแอดเดรส (ถาวรและชั่วคราว) ที่ใช้บนระบบ
- ดูเนื้อที่ว่างดิสก์ทั้งหมด
- ดูความจุของดิสก์พูลระบบและการใช้งาน
- ดูจำนวนของตัวประมวลผลบนระบบของคุณ
- >> ดูประเภทของตัวประมวลผลว่าเป็นชนิด dedicated, shared-capped, หรือ shared-uncapped (ถ้าฮาร์ดแวร์นั้นสนับสนุน)<<

- >> ดูเปอร์เซ็นต์การใช้งานพูลตัวประมวลผลร่วมของระบบ<<
- >> ดูเปอร์เซ็นต์การใช้งานของ shared-uncapped CPU capacity ของระบบ (ถ้าฮาร์ดแวร์นั้นสนับสนุน)<<

หมายเหตุ: มีหน้าProcessorsจำนวนสามหน้า โดยขึ้นอยู่กับประเภทของระบบ iSeries ที่คุณมี. คุณอาจดูข้อมูลว่าด้วยตัวประมวลผลเพิ่มเติมได้โดยขึ้นอยู่กับ configuration ของระบบคุณ:

- ระบบที่ไม่มีพาร์ติชัน
- ระบบที่มีพาร์ติชัน, ตัวประมวลผลเฉพาะ
- ระบบที่มีพาร์ติชัน, ตัวประมวลผลที่ใช้งานร่วมกัน

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการทำโลจิคัลพาร์ติชันบนระบบ iSeries, ให้อูที่โลจิคัลพาร์ติชัน.

- ให้อูหน่วยความจำรวมของระบบ
- ดูหน่วยเก็บชั่วคราวที่ถูกใช้งาน
- ดูขนาดหน่วยเก็บชั่วคราวปัจจุบันที่ถูกใช้งานและขนาดสูงสุดที่ถูกใช้งานนับตั้งแต่การรีสตาร์ทระบบครั้งล่าสุด
- แอ็กเซสงานแอ็คทีฟ
- แอ็กเซสงานและค่ากำหนดของระบบหน่วยหน่วยเก็บ
- แอ็กเซสดีสก์พูล
- แอ็กเซสพูลหน่วยความจำแอ็คทีฟ

คุณสามารถแอ็กเซสไดอะล็อก System Status ได้จากโพลเดอร์ System หรือโพลเดอร์ Work Management ภายใน iSeries Navigator.

วิธีการดึงสถานะระบบจากโพลเดอร์System:

1. ใน iSeries Navigator, ให้อูแดก My Connections.
2. คลิกขวาที่การเชื่อมต่อที่คุณต้องการทำงานด้วยและเลือก System Status.

วิธีการดึงสถานะระบบจาก การจัดการระบบงาน folder:

1. ใน iSeries Navigator, ให้อูขยาย Work Management.
2. คลิกเมาส์ขวาที่ Work Management และเลือก System Status.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการกิจอื่นๆ ที่คุณสามารถทำเสร็จสมบูรณ์โดยใช้สถานะระบบ, ให้อูที่คำอธิบาย iSeries Navigator.

## การจัดการงานและ thread

เนื่องจากชุดงานส่วนใหญ่ที่ทำอยู่บนระบบจะอยู่ในรูปแบบของชิ้นงาน (job) และ thread , ดังนั้น จึงเป็นสิ่งสำคัญที่คุณจะเป็นต้องค้นหา, ติดตาม และจัดการงานภายในระบบของคุณให้เป็น.

ดูหัวข้อย่อยด้านล่างเพื่อเป็นข้อมูลในการจัดการงานและ thread:

- >> “การกำหนดตารางการทำงาน” ในหน้า 8<<
- “การค้นหางานบนเซิร์ฟเวอร์ iSeries” ในหน้า 10
- “การกำหนดสถานะของงาน” ในหน้า 13

- “การดูสถิติประสิทธิภาพการทำงานของงาน” ในหน้า 14
- **»** “การเรียกดูข้อมูลที่มีความสัมพันธ์กันของงาน” ในหน้า 16 **«**
- “การสิ้นสุดการทำงาน” ในหน้า 16
- “การปฏิบัติงาน” ในหน้า 17
- “การดู threads รั้นภายใต้งานเฉพาะ” ในหน้า 19
- “การดูคุณสมบัติ thread” ในหน้า 19
- “การลบหรือจบการทำงานของ thread” ในหน้า 19


สำหรับข้อมูลเพิ่มเติมเกี่ยวกับภารกิจต่างๆที่คุณสามารถทำกับงานและ thread ได้, ให้ดูที่คำอธิบาย iSeries Navigator .

สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับงานและประเภทของงานบนเซิร์ฟเวอร์ iSeries, ให้ดูหัวข้อ “งาน” ในหน้า 35. สำหรับข้อมูลโดยละเอียดเกี่ยวกับ thread, ให้ดูที่ “thread” ในหน้า 53.

## การกำหนดตารางการทำงาน

**»** การกำหนดตารางการทำงานในระบบ iSeries, สามารถทำได้ที่ Management Central Scheduler, หรือที่ OS/400 scheduler, และที่ Advanced Job Scheduler.

สำหรับข้อมูลของตัวกำหนดตารางทำงานแต่ละชนิด, ให้ดูที่หัวข้อต่อไปนี้:

- Management Central Scheduler  
ให้ใช้ตัวกำหนดตารางการทำงานนี้กับงานที่อยู่ใน Management Central.
- Advanced Job Scheduler  
ให้ใช้ตัวกำหนดตารางการทำงานนี้กับงานที่อยู่ในเซิร์ฟเวอร์ iSeries. ตัวกำหนดตารางการทำงานนี้เป็น plug-in ของส่วนอินเตอร์เฟสใน iSeries Navigator และมีความเสถียรสูงกว่า OS/400 scheduler.
- OS/400 scheduler  
ตัวกำหนดตารางการทำงานนี้เหมาะกับงานพื้นฐานทั่วไป. ซึ่งคุณสามารถใช้งานผ่านทางเซสชันของอิมูเลเตอร์แบบ 5250 เท่านั้น. โดยไม่สามารถใช้ผ่านทาง iSeries Navigator. สำหรับข้อมูลเพิ่มเติม, ให้ดูที่ Job Scheduler for OS/400  .

สำหรับข้อมูลเพิ่มเติม, ให้ดูหัวข้อต่อไปนี้:

### “เปรียบเทียบตารางเวลางาน”

ดูที่หัวข้อนี้เพื่อช่วยตัดสินใจว่าคุณสมบัติของตัวกำหนดตารางการทำงานแบบใดที่สำคัญ. จากนั้น, คุณจะสมารถตัดสินใจได้ว่าตัวกำหนดตารางการทำงานใดที่เหมาะสมกับคุณ.

การกำหนดตารางการทำงานของงานหรือ task ด้วย Management Central Scheduler  
เปรียบเทียบวิธีการใช้ OS/400 Management Central scheduler กับ Advanced Job Scheduler.



**เปรียบเทียบตารางเวลางาน:** เมื่อทำการเลือกผลิตภัณฑ์ตารางเวลางาน, คุณต้องพิจารณาความหลากหลายของคุณลักษณะพิเศษต่างๆ. ต่อไปนี้เป็นรายการของคุณลักษณะพิเศษที่ต้องพิจารณาเมื่อมีการตัดสินใจว่าจะใช้ตารางเวลางานอย่างไร:

### การกำหนดตารางเวลางานแบบอัตโนมัติ

- ความยืดหยุ่นในการกำหนดตารางเวลางาน
- การประมวลผลงานที่ไม่ตั้งใจ (หรือ ตั้งใจ) ตลอด 24 ชั่วโมงต่อวัน, 7 วันต่อสัปดาห์, พร้อมด้วยการประยุกต์ทั้งหมดให้เข้ากับตารางเวลางานที่คุณตั้งไว้
- ส่วนขยายตามธรรมชาติของระบบปฏิบัติการ iSeries
- การควบคุมโดยสมบูรณ์ของงานที่ส่งเข้ามาว่าอย่างไร, เมื่อไร, และที่ไหน
- งานส่วนขยายขึ้นอยู่กับ อีอบเจ็กต์ (การเกิดขึ้นของไฟล์หรือเรียกคอร์ตภายในฟิสิคัลไฟล์), activity หรือ inactivity ของงานอื่น, หรือสถานะของสาย, คอนโทรลเลอร์, หรือ ระบบย่อย
- เสริมสร้างฟังก์ชันการจัดการปฏิทิน, รวมถึงปฏิทินการเงิน และปฏิทินวันหยุด
- การทำงานหลายอย่างต่อวัน

#### พารามิเตอร์ระบบ และพารามิเตอร์ที่ผู้ใช้กำหนด

- วันที่ปัจจุบัน, วันที่ส่งงาน, วันที่ก่อนหน้า, และเวลาปัจจุบัน สามารถส่งผ่านเข้าไปในแอ็พพลิเคชันโปรแกรมได้
- ค่าพารามิเตอร์ที่ผู้ใช้กำหนดสามารถสร้าง, เปลี่ยน, และส่งผ่านเข้าไปในแอ็พพลิเคชันโปรแกรมได้

#### การพยากรณ์ Workload/history

- พยากรณ์ตารางเวลาทำงานทั้งหมดให้วัน สัปดาห์หน้า, เดือนหน้า, หรือวันหน้า
- Optimize ความต้องการในการผลิต
- การแทรกข้อมูลในอดีต และบันทึกการทำงานของการปฏิบัติงาน Advanced Job Scheduler ทั้งหมด

#### การจัดการเน็ตเวิร์ก

- เราสามารถตั้งค่างานบนเซิร์ฟเวอร์ iSeries ใดๆในเน็ตเวิร์กเพื่อรันบนเซิร์ฟเวอร์ iSeries อื่นๆบนเน็ตเวิร์ก
- จัดเตรียมข้อมูลงานในอดีตที่เสร็จสิ้นของงานบนระบบที่มีการส่งงาน
- เราสามารถส่งกลุ่มงานและงานส่วนเสริมผ่านเข้าไปยังเน็ตเวิร์กได้

#### การจัดการ และการจำแนกรายงาน

- การจัดเส้นทาง, การมอนิเตอร์, และการควบคุม รายงานเอาต์พุตทั้งหมดที่สร้างโดย Advanced Job Scheduler หรือระบบปฏิบัติการ OS/400
- การจำแนกไฟล์ที่สูญหายไปยังเอาต์พุตคิวหลายอัน หรือไปยังระบบบริโมตพร้อมด้วยตัวเลือกแบนเนอร์เพจ
- เราสามารถทำสำเนาสูญเอาต์พุต หรือส่งไปยังผู้ใช้ใดๆบนเน็ตเวิร์กของระบบ iSeries

#### การรักษาความปลอดภัย

- การรักษาความปลอดภัยของระบบ iSeries ที่ใช้งานอยู่อาจจะทำให้เกิดประโยชน์ภายใน Advanced Job Scheduler
- ให้ระบุบุคคลในองค์กรของคุณที่มีสิทธิในการตั้งค่าหรือเปลี่ยนแปลงข้อมูลเกี่ยวกับงานที่กำหนดตารางเวลา
- เราสามารถระบุสิทธิในการใช้งานสำหรับ ฟังก์ชันส่วนบุคคลของ Advanced Job Scheduler หรือสำหรับงานที่ระบุเฉพาะก็ได้

#### Graphical user interface

- ความสามารถในการเลือกและคลิก เมื่อมีการกำหนดตารางเวลาของงาน
- การจัดการงาน

- การคงไว้ของส่วนเสริม
- แทร็กกำหนดตารางเวลาการปฏิบัติงาน และ ข้อมูลบันทึกการทำงาน

### คุณลักษณะพิเศษของคีย์อื่นๆ

- หลายคำสั่งทำงาน
- Definition สำหรับงาน LDA (Local Data Area)
- มอนิเตอร์คอนโซลเพื่อรันงานในสภาวะจำกัด
- ตรวจสอบรันไทม์สูงสุดสำหรับแต่ละงาน
- อินเตอร์เฟซโดยตรงกับระบบการเพจแบบข้อความของบริษัทที่ให้บริการอื่นๆ
- การส่งงาน และการมอนิเตอร์ของโปรซีเดอร์ System/36
- บทบัญญัติสำหรับเอกสารคู่มือออนไลน์แบบเต็มของแต่ละงาน
- ปุ่มสัมผัสเคอร์เซอร์ที่ขยายข้อความคำอธิบายบนจอแสดงผลทั้งหมด

ให้ดู Advanced Job Scheduler กับ ตารางเวลามาตรฐาน OS/400  สำหรับการเปรียบเทียบข้อมูล.

### การค้นหางานบนเซิร์ฟเวอร์ iSeries

การทำความเข้าใจเรื่องการค้นหางานบนเซิร์ฟเวอร์ iSeries มีความสำคัญอย่างยิ่ง. เพราะไม่ว่าอย่างไรก็ตาม, บางครั้งคุณอาจต้องการข้อมูลบางอย่างจากเฉพาะบางงานขึ้นมาได้. ใน iSeries Navigator, คุณสามารถ **Find** งานทั้งหมด หรือคุณสามารถบีบการค้นหาลงเพื่อใช้ฟังก์ชัน **Include...** ตามด้วย Find ก็ได้. ฟังก์ชัน **Include...** จะอนุญาตให้คุณใส่เงื่อนไขในการแสดงผลใน iSeries Navigator. ตัวอย่างเช่น, แทนที่จะต้องทำการค้นหางานเป็นจำนวนหลายร้อยงาน, คุณสามารถรัน **Include...** เพื่อแสดงผลของงานบางประเภทได้. หรือ, คุณสามารถ เลือกแสดงผลเหล่านั้นด้วยการระบุ job user ID เป็นการเฉพาะ.

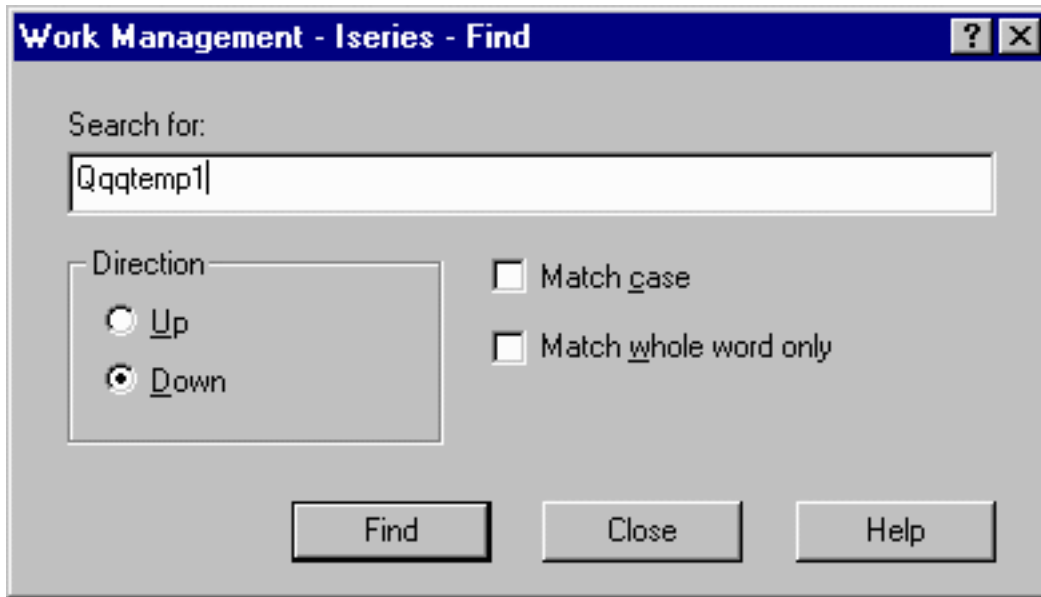
และหากมองในเรื่องของประสิทธิภาพการทำงานแล้ว, ในกรณีที่คุณมีงานเป็นจำนวนมากอยู่บนระบบ, ขอแนะนำให้ใช้ฟังก์ชัน **Include...** เพื่อบีบจำนวนงานที่จะต้องค้นหา. เพราะการค้นหาให้ทั่วงานทั้งระบบที่มีอยู่เป็นจำนวนมากนั้น จะทำให้การทำงานของระบบช้าลง.

**หมายเหตุ:** คุณสามารถใช้เมนูบาร์ **Find** และ **Include...** ได้ทั่วการจัดการระบบงานที่คุณต้องการค้นหา. คุณยังสามารถใช้ทูลต่างๆ ในการค้นหาคิวงาน, ระบบย่อย, และพูลหน่วยความจำในลักษณะเดียวกันอีกด้วย. แต่อย่าลืมว่าคุณจะต้องคลิกไปบนพื้นที่ๆ ต้องการค้นหาก่อนที่จะใช้ทูลเหล่านี้ค้นหาต่อไป.

หากต้องการค้นหางานโดยใช้ฮ็อพชัน **Find (Ctrl+F)**, ให้ทำดังต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → **server-name** → **Work Management** → **Active Jobs**.
2. เลือก **Edit** → **Find (Ctrl+F)**.
3. ในฟิลด์ข้อความ **Search for:**, ให้พิมพ์ job ID ที่คุณต้องการค้นหา (เช่น Qqtemp1). จะค้นหาของคุณในคอลัมน์งานทั้งหมด.





4. คลิก **Find**. ในกรณีที่พบ, iSeries Navigator จะไฮไลต์งานดังกล่าวให้เห็น.  
หมายเหตุ: อย่าลืมว่าชื่องานต้องคั่นถึงขนาดตัวพิมพ์เล็กใหญ่ด้วย เมื่ออยู่ภายในเครื่องหมายคำพูด (เช่น, "MyJob").  
แต่หากชื่องานไม่ได้อยู่ในเครื่องหมายคำพูด, ก็ไม่ต้องคั่นขนาดตัวพิมพ์เล็กใหญ่.

ในการจำกัดข้อมูลที่จะแสดงโดยใช้ **Include...**; ให้ทำดังนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → *server-name* → **Work Management** → **Active Jobs** or **Server Jobs**.

- จากเมนู View , ให้เลือก Customize this View, และ Include. โดอะล็อกซ์ Include จะปรากฏขึ้น.

**Active Jobs - Include**

**Include**

Job:

Name: All

User: All Browse...

Number: All

Type

Autostart  Reader

Batch  Subsystem

Communications  System

Interactive  Writer

Prestart

Select All

Subsystem: All Browse...

Current user: All Browse...

OK Cancel Help ?

- ในโดอะล็อกซ์ Include, เลือกอ็อพชั่นที่คุณต้องการในการค้นหา.
- คลิก OK. จากจุดนี้, ให้ใช้ Find เพื่อทำการแสดงงานที่ต้องการค้นหาอย่างเฉพาะเจาะจง.

สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับงาน, ให้อูที่ “งาน” ในหน้า 35.

## การกำหนดสถานะของงาน

การมอนิเตอร์งานจะช่วยให้คุณรู้ว่างานดังกล่าวกำลังทำอะไรอยู่. สถานะของงานเป็นข้อมูลชิ้นสำคัญที่คุณสามารถใช้ดูว่างานชิ้นหนึ่งๆ กำลังทำงานอะไรอยู่. คุณสามารถค้นหาสถานะงานได้อย่างสะดวกใน iSeries Navigator.

หากต้องการตรวจสอบสถานะของงานแอคทีฟหรืองานเซิร์ฟเวอร์, ให้ทำดังนี้:

1. ใน iSeries Navigator, ให้ขยาย My Connections → server-name → Work Management → Active Jobs or Server Jobs.

หมายเหตุ: คุณสามารถดูสถานะของงานจากที่ใดก็ได้จากโพลเดอร์ของ Work Management ที่คุณได้อีกเสเข้าไปในงานนั้น.

2. ให้อูที่คอลัมน์ “สถานะโดยละเอียด” ในหน้า 49 เพื่อดูสถานะของงาน (เช่น Waiting for event, Waiting for time interval, หรือ Waiting for dequeue).

Job Name	Subsystem	Detailed Status	Type	Run Priority
Qserver	Qserver	Waiting for event	Autostart	20
Qneosoem	Qsyswrk	Waiting for time interval	Autostart	50
Qzbsvbm	Qsyswrk	Waiting for event	Autostart	50
Qgldpube	Qsyswrk	Waiting for dequeue	Autostart - Server	50
Qgldpuba	Qsyswrk	Waiting for signal	Autostart - Server	50
Qsysacd	Qctl	Waiting for event	Batch	10
Qqtemp1	Qsyswrk	Waiting for dequeue	Batch	20
Qpasvrp	Qsyswrk	Waiting for dequeue	Batch	50
Qqtemp2	Qsyswrk	Waiting for dequeue	Batch	20
Qqtemp1	Qsyswrk	Waiting for dequeue	Batch	20
Qqtemp2	Qsyswrk	Waiting for dequeue	Batch	20
Qpwfservsd	Qserver	Waiting for select	Batch - Server	20
Qneosoem	Qsyswrk	Waiting for time interval	Batch	50
Qneosoem	Qsyswrk	Waiting for time interval	Batch	50
Q1psch	Qsyswrk	Delayed 60 seconds	Batch	50
Qprfsynch	Qsyswrk	Waiting for dequeue	Batch	50
Qnetwork	Qsyswrk	Waiting for dequeue	Batch	50
Qmsf	Qsyswrk	Waiting for dequeue	Batch	35
Qpasvrs	Qsyswrk	Waiting for time interval	Batch	30
Qpasvrs	Qsyswrk	Waiting for time interval	Batch	30
Qtcpip	Qsyswrk	Waiting for dequeue	Batch - Server	20
Qtcpmonit	Qsyswrk	Waiting for dequeue	Batch	10
Qappctcp	Qsyswrk	Waiting for time interval	Batch	20
Qrwtlstn	Qsyswrk	Waiting for select	Batch - Server	20
Qdirsv	Qsyswrk	Waiting for signal	Batch - Server	50
Qtldp01498	Qsyswrk	Waiting for dequeue	Batch - Server	25
Qtldp01514	Qsyswrk	Waiting for time interval	Batch - Server	25
Qypssrv	Qsyswrk	Waiting for dequeue	Batch - Server	10
Qzlsserver	Qserver	Waiting for event	Batch - Server	20
Qtsmtprsvd	Qsyswrk	Waiting for select	Batch - Server	35
Qtmsnmp	Qsyswrk	Waiting for dequeue	Batch - Server	35
Qvtelnet	Qsyswrk	Waiting for time interval	Batch - Server	20
Qtvdevice	Qsyswrk	Waiting for time interval	Batch - Server	20

สำหรับข้อมูลอย่างละเอียด, โปรดดู “สถานะโดยละเอียด” ในหน้า 49.

## การดูสถิติประสิทธิภาพการทำงานของงาน

ประสิทธิภาพการทำงานเป็นสิ่งสำคัญกับผู้ใช้เซิร์ฟเวอร์ iSeries เนื่องจากงานหนึ่งงานที่รันอย่างด้อยคุณภาพจะส่งผลกระทบต่องานอื่นๆ บนระบบไปด้วย. การดูงานที่มีปัญหาจะทำให้คุณป้องกันปัญหาของประสิทธิภาพการทำงานก่อนที่จะเกิดขึ้นได้.

หน้าต่าง “สถิติการทำงานที่ผ่านมา” ในหน้า 48 จะให้คุณมอนิเตอร์การใช้งาน CPU ของงาน, disk I/O (ฮาร์ดไดรฟ์อินพุต/เอาต์พุต), อัตรา page fault, เวลาเฉลี่ยของการตอบกลับ, และจำนวนของ interactive transaction. คุณสามารถเลือกอีพซันในหน้าต่างนี้เพื่อรีเฟรชสถิติเหล่านี้ด้วยมือหรือด้วยการตั้งเวลา.

ในการแสดงสถิติการทำงานที่ผ่านมา, ให้ทำตามนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → *server-name* → **Work Management** → **Active Jobs**.

**หมายเหตุ:** คุณสามารถตรวจสอบประสิทธิภาพของงานได้จากการจัดการระบบงานทุกที่. สามารถแสดงผลไดอะล็อก **Elapsed Performance Statistics** จากแท็บ **Performance** ของหน้าคุณสมบัติ **Job**.

2. คลิกขวาที่งานที่คุณต้องการแสดงผลสถิติการทำงาน,และเลือก Details → Elapsed Performance Statistics.

Job: Qpwfserv User: Quser Number: 090863

Elapsed time: 00:00:00

Performance statistics calculated over elapsed time

CPU:

Percentage:	0.0	%
Time:	0	milliseconds
Database percentage:	0.0	%
Database time:	0	milliseconds

Disk I/O rate:

Synchronous:	0.0	per second
Asynchronous:	0.0	per second

Disk I/O count:

Synchronous:	0
Asynchronous:	0

Page fault rate:

Average response time:	0	seconds
------------------------	---	---------

Interactive transactions:

	0
--	---

Refresh Now Timed Refresh... Reset Statistics Close Help ?

คุณสามารถรีเฟรช, รีเซ็ต, และตั้งกำหนดเวลาของสถิติการทำงานเพื่อให้รีเฟรชโดยอัตโนมัติได้.

**หมายเหตุ:** คุณสามารถดูสถิติการทำงานที่ผ่านมาของงานได้มากกว่าหนึ่งงาน ในเวลาเดียวกัน โดยเปิดหน้าต่างหลายๆ อันพร้อมกัน. สิ่งนี้จะช่วยให้คุณมอนิเตอร์งานจำนวนมากที่มีปัญหาได้พร้อมๆ กัน. หน้าต่างแต่ละอันจะแสดงข้อมูลของงานๆ หนึ่งเท่านั้น.

สถิติการทำงานที่ผ่านมาคือวิธีการดูประสิทธิภาพการทำงานในขณะที่งานนั้นย้ายไปตามที่ต่างๆ ของระบบ. อีกวิธีหนึ่งในการดูงานบนระบบคือดูผ่านทางโพลเดอร์ Management Central. คุณสามารถมอนิเตอร์งานใน Management Central, รวมถึงการทำงาน และข้อความของระบบ. สำหรับข้อมูลเรื่องการมอนิเตอร์งาน, ให้ดูที่ การมอนิเตอร์ Management Central.

## การเรียกดูข้อมูลที่มีความสัมพันธ์กันของงาน

▶ แต่ละงานที่อยู่บน iSeries ประกอบด้วยข้อมูลที่มีความสัมพันธ์กันของหน่วยความจำและตัวประมวลผล. ข้อมูลที่มีความสัมพันธ์กันอธิบายว่า ไม่ว่า threads จะมีความสัมพันธ์ต่อกลุ่มของตัวประมวลผลและหน่วยความจำเดียวกันกับ initial thread หรือไม่ก็ตามเมื่อมันเริ่มต้นทำงาน. มันยังคงระบุถึงระดับของระบบที่พยายามจะรักษาไว้ซึ่งความสัมพันธ์ระหว่าง threads และ เซ็ตย่อยของรีซอร์สระบบที่ได้กำหนดไว้แล้ว. นอกจากนี้, ข้อมูลที่มีความสัมพันธ์กันระบุ โดยไม่คำนึงถึงว่า งานนี้ได้จัดกลุ่มรวมกับงานอื่น ดังนั้นมันจะมีความสัมพันธ์กับเซ็ตย่อยของรีซอร์สของระบบเดียวกัน.

โดยการรวม threads เข้าด้วยกันที่จะแบ่งใช้งานชุดของข้อมูลทั่วไปในแหล่งเก็บข้อมูลหลัก, การแคชของระบบของคุณและ อัตราการเข้าถึงข้อมูลหน่วยความจำจะดีขึ้น.

เพื่อที่จะเรียกดูข้อมูลที่มีความสัมพันธ์กัน, ให้ทำดังต่อไปนี้ให้สมบูรณ์:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → *server-name* → **Work Management** → **Active Jobs**.

**หมายเหตุ:** คุณสามารถดูข้อมูลที่มีความสัมพันธ์กันของงานจากตำแหน่งใดก็ได้ภายใน การจัดการระบบงาน ที่ซึ่งคุณสามารถเรียกดูงานได้.

2. คลิกขวานงานที่คุณต้องการเรียกดู, และเลือก **Properties**.
3. บนหน้า **Resources**, คุณสามารถเรียกดูข้อมูล ความสัมพันธ์ของหน่วยความจำและตัวประมวลผล.

สำหรับข้อมูลเพิ่มเติมบนแต่ละฟิลด์, ให้ดูคำอธิบายออนไลน์.

สำหรับข้อมูลเพิ่มเติมบนค่ากำหนดของระบบที่มีความสัมพันธ์กัน, ให้ดู Thread affinity (QTHDRSCAFN). นอกจากนี้, คุณสามารถระบุให้รีซอร์สของ thread ปรับเปลี่ยนอย่างอัตโนมัติโดยการใช้ค่ากำหนดของระบบ Automatically adjust thread resources (QTHDRSCADJ). ◀◀

## การสิ้นสุดการทำงาน

ในบางกรณี คุณต้องการจบงานเพราะต้องใช้เวลาในการรันงานนานเกินไปหรืองานเหล่านั้นต้องใช้หน่วยความจำมากเกินไป ซึ่งสิ่งนี้จะมีผลกระทบต่อการทำงานของงานอื่นๆ บนระบบ.

ในการจบงาน, ให้ทำดังนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → *server-name* → **Work Management** → **Active Jobs**.

**หมายเหตุ:** คุณสามารถ **Delete/End** งานจากที่ใดก็ได้ในการจัดการระบบงานที่คุณเห็นงานนั้นๆ.

2. คลิกขวาไปบนงานที่ต้องการ (เช่น, Qdfjjobd) แล้วเลือกDelete/End.

Job Name	User	Number	Type
Qdfjjobd	Timr	091080	Batch

How to end: Controlled

Time limit for controlled end:  seconds

Delete printer output: No

Maximum job log entries: Use job value

Action for related interactive jobs: Do not end

3. ในฟิลด์ **How to end**, เลือก “สิ้นสุดการทำงาน” ในหน้า 49.
4. ในฟิลด์ **Time limit for controlled end**, ให้ใส่จำนวนวินาทีก่อนที่งานจะสลับจากการจบแบบควบคุมไปเป็นการจบแบบทันที. (พารามิเตอร์ตัวนี้จะใช้ได้กับ Delete/End แบบควบคุม.)
5. ในฟิลด์ **Delete printer output**, ให้เลือก **Yes** หรือ **No**.
6. ในฟิลด์ **Maximum job log entries**, ให้เลือก **Use job value** หรือ **No maximum**.
7. ในฟิลด์ **Action for related interactive jobs**, ให้เลือก **Do not end**, **End for group jobs**, หรือ **End all**.
8. คลิก **Delete** เพื่อทำการลบงาน.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับสิ่งที่คุณสามารถทำกับงานได้, ให้อูที่ “การปฏิบัติงาน”.

## การปฏิบัติงาน

การจัดการงานและ thread สามารถทำได้อย่างมีประสิทธิภาพเพิ่มขึ้นด้วยการดำเนินงานที่มีอยู่ในการจัดการระบบงาน. เมื่อคุณ “การค้นหางานบนเซิร์ฟเวอร์ iSeries” ในหน้า 10 ที่คุณต้องการจัดการแล้ว, ฟังก์ชันการปฏิบัติงานต่อไปนี้จะถูกเตรียมไว้ให้โดยการคลิกขวาไปบนงานนั้น:

## Reset Statistics

จะอนุญาตให้คุณรีเซ็ตลิสต์รายการข้อมูลที่คุณกำลังดูอยู่และจะเซตเวลาที่ผ่านไปให้เป็น 00:00:00.

### “เอาต์พุตคิว” ในหน้า 77

จะแสดงพรินเตอร์เอาต์พุต, หากมีอยู่, ในหน้าต่างแยกต่างหาก.

### “บันทึกการใช้งาน” ในหน้า 52

จะแสดงบันทึกการใช้งานของงานที่เลือกไว้, ในหน้าต่างแยกต่างหาก.

### “รายละเอียด: การปฏิบัติงานแอคทีฟ” ในหน้า 50

ประกอบด้วยข้อมูลโดยละเอียดเกี่ยวกับคุณสมบัติต่อไปนี้สำหรับงานแอคทีฟ:

- Call Stack
- รายชื่อไลบรารี
- “อ็อบเจกต์ที่ถูกล็อก” ในหน้า 51
- เปิดไฟล์
  - [» Library Objects«](#)
  - [» File System Objects«](#)
- “thread” ในหน้า 53
- Transaction
- “สถิติการทำงานที่ผ่านมา” ในหน้า 48
- ข้อความ SQL สุดท้าย

## ตอบกลับ

จะให้คุณตอบกลับข้อความนั้นได้, หากคุณมีงานที่กำลังรอข้อความ.

## หยุดพัก

อนุญาตให้คุณหยุดพักงานไว้. การพักงานเอาไว้จะเป็นการหยุดพัก thread ทั้งหมดในงานนั้น. ซึ่งจะทำได้กับงานที่ถูกปล่อยออกมา ซึ่งต้องไม่ป็นงานระบบ. เมื่อคุณพักงานเอาไว้, งานดังกล่าวจะไม่สามารถประมวลผลได้. งานแอคทีฟจะถูกพักไว้เพื่อเป็นการหยุดการประมวลผลชั่วคราว.

## ปล่อย

จะเป็นการปล่อยงานที่ถูกพักไว้. การปล่อยงานออกจะปล่อย thread ทั้งหมดที่ถูกพักไว้ด้วยการปฏิบัติงานแบบหยุดพัก. ดังนั้นแล้วงานจะถูกปล่อยออกมาให้ประมวลผลต่อไปได้.

### “การย้ายงานไปยังคิวงานอื่น” ในหน้า 21

อนุญาตให้คุณย้ายงานที่เลือกไปไว้ยังคิวงานอื่น. คุณสามารถย้ายงานที่อยู่ในคิวงานได้เท่านั้น.

### “สิ้นสุดการทำงาน” ในหน้า 49

อนุญาตให้คุณจบการทำงานของงานที่เลือกไว้. สองวิธีในการจบงาน, คือ การจบแบบควบคุมหรือจบทันที หรืออย่างใดอย่างหนึ่ง.



## มอนิเตอร์

อนุญาตให้คุณสร้างมอนิเตอร์งานสำหรับงานหนึ่งชิ้นหรือมากกว่า.

“คุณสมบัติของงาน” ในหน้า 46

คุณสามารถดูหรือเปลี่ยนคุณสมบัติของงานได้.

## การดู threads รั้นภายใต้งานเฉพาะ

งานแอ็คทีฟแต่ละงานที่รั้นอยู่ในระบบ iSeries มี thread อย่างน้อยหนึ่งตัวที่รั้นอยู่ภายใต้งานนั้น. thread คือหน่วยงานอิสระที่รั้นอยู่ภายในงานที่ใช้รีซอร์สเดียวกันกับงาน. เนื่องจากงาน(job)ขึ้นอยู่กับงาน(work)ที่เสร็จสิ้นโดย thread, จึงเป็นเรื่องสำคัญที่ต้องเรียนรู้วิธีการหา thread ที่รั้นภายในงานเฉพาะ.

หากต้องการดู thread ที่รั้นภายใต้งานเฉพาะ, ให้ทำตามต่อไปนี้:

1. จากที่ iSeries Navigator, ขยาย My Connections → *server-name* → Work Management → Active Jobs.
2. คลิกขวาที่งานที่คุณสนใจ, แล้วเลือก Details > Threads.

สำหรับข้อมูลอย่างละเอียด, โปรดดูที่“thread” ในหน้า 53 หรือดูที่คำอธิบาย iSeries Navigator .

## การดูคุณสมบัติ thread

thread จะช่วยให้งานสามารถทำงานหลายๆ อย่างได้ในเวลาเดียวกัน. หาก thread หยุดการประมวลผล, thread จะหยุดงานไม่ให้รั้น. หน้า Thread Properties จะช่วยให้คุณดู thread ต่างๆ และคุณสมบัติการทำงานต่างๆ ของ thread ที่สามารถช่วยในการศึกษาสาเหตุว่าทำไม thread ไม่รั้น.

หากต้องการดูคุณสมบัติของ thread, ให้ทำตามต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย My Connections → *server-name* → Work Management → Active Jobs or Server Jobs.
2. คลิกขวาที่งานที่คุณต้องการทำงานด้วย, และเลือก Details > Threads.
3. คลิกขวาที่ thread ที่คุณต้องการทำงานด้วย, และเลือก Properties.

สำหรับข้อมูลโดยละเอียด, ให้ดูที่ “thread” ในหน้า 53 หรือดูที่คำอธิบาย iSeries Navigator .

## การลบหรือจบการทำงานของ thread

initial thread, ซึ่งถูกสร้างเมื่องานเริ่มทำงาน, ไม่สามารถถูกลบทิ้งหรือสิ้นสุดการทำงานได้. อย่างไรก็ตาม, บางครั้งก็เป็นเรื่องจำเป็นที่ต้องจบการทำงานของ thread รองเพื่อที่ว่างานนั้นจะยังคงรั้นอยู่. โปรดทราบว่า thread ที่คุณต้องการให้สิ้นสุดการทำงานอาจทำให้งานที่รั้นอยู่ภายในไม่สามารถเสร็จสมบูรณ์ได้โดยปราศจากงานของ thread นั้น.

**สิ่งสำคัญ:** การจบการทำงานของ thread ตอนปลายไม่ควรเป็นส่วนหนึ่งของการจัดการงานประจำวัน. การจบการทำงานของ thread สำคัญกว่าการจบการทำงานของงานเพราะงานใน thread อาจหยุดหรือไม่หยุดทำงาน. เมื่อคุณสิ้นสุดการทำงาน, งานทั้งหมดจะหยุดทำงาน. อย่างไรก็ตาม, เมื่อคุณจบการทำงานของ thread, จะมีส่วนของงานเท่านั้นที่หยุดทำงาน. thread อื่นๆ อาจรันต่อหรือไม่รันต่อ. หาก thread รันต่อไปโดยไม่มี thread ที่คุณจบการทำงาน, thread อาจให้ผลที่ไม่เป็นที่ต้องการ.

หากต้องการลบหรือจบการทำงานของ thread รอง, คุณต้องมีสิทธิการบริการพิเศษ (\*SERVICE) หรือ “Thread Control” ในหน้า 55.

หากต้องการลบหรือจบการทำงานของ thread, ให้ทำดังต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → *server-name* → **Work Management** → **Active Jobs** หรือ **Server Jobs**.
2. คลิกขวาที่งานซึ่งคุณต้องการทำงานด้วย, และเลือก **Details**, และ **Threads**.
3. คลิกขวาที่ thread ที่คุณต้องการจบการทำงาน, และเลือก **Delete/End**.

สำหรับข้อมูลอย่างละเอียด, โปรดดูที่“thread” ในหน้า 53 หรือดูที่คำอธิบาย iSeries Navigator .

## การจัดการคิวงาน

ในวงจรชีวิตของงานแบบแบ็ตซ์, คิวงานจะเป็น entry point เพื่อเข้าสู่ระบบย่อย. คิวงานจะดูแลเกี่ยวกับจำนวนของงานที่ได้เข้าสู่ระบบย่อย ณ เวลาหนึ่งและลำดับที่งานจะได้รับอนุญาตให้เข้ามาสู่ระบบย่อย.

หัวข้อย่อยเหล่านี้จะมีคำสั่งสำหรับภารกิจต่อไปนี้:

- “การดูงานบนคิวงาน”
- “การเปลี่ยนระดับความสำคัญของงานภายในคิวงาน”
- “การย้ายงานไปยังคิวงานอื่น” ในหน้า 21

สำหรับข้อมูลเพิ่มเติม, ให้ดู “คิวงาน” ในหน้า 56.

## การดูงานบนคิวงาน

คิวงานกรองงานบางส่วนที่จะถูกประมวลผลในการจัดการระบบงาน (ยกตัวอย่างเช่น, งานแบบแบ็ตซ์บางงาน). การดูงานในคิวงานได้ทำให้คุณสามารถดูว่างานใดที่กำลังรอที่จะถูกส่งไปยังระบบย่อย.

ในการดูงานในคิวงาน, ให้ทำดังนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → *server-name* → **Work Management** → **Job Queues** → **Active Job Queues** หรือ **All Job Queues**.
2. เลือกคิวงานที่คุณต้องการจะดูงาน (เช่น, Jobqueue1). งานที่อยู่ในคิวงานจะปรากฏให้เห็น.

สำหรับข้อมูลเพิ่มเติม, ให้ดูที่ “คิวงาน” ในหน้า 56.

## การเปลี่ยนระดับความสำคัญของงานภายในคิวงาน

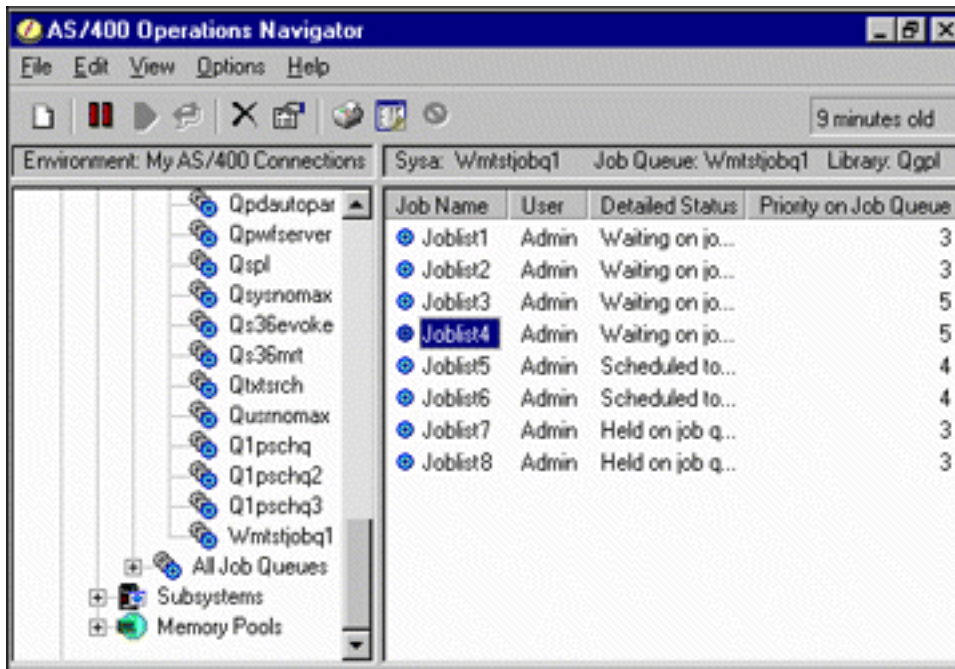
ในบางกรณี ความสำคัญของงานก็มีการเปลี่ยนไปเมื่องานดังกล่าวผ่านไปในช่วงวงจรชีวิต. ซึ่งระดับความสำคัญอาจเพิ่มขึ้นหรือลดลงเมื่อเทียบกับงานอื่น. เนื่องจากการเปลี่ยนแปลงดังกล่าวเกิดขึ้น, จึงจำเป็นที่จะต้องรู้วิธีการเปลี่ยนระดับความสำคัญของงานในคิวงาน. ระดับความสำคัญของงานในคิวงานจะช่วยกำหนดได้ว่าเมื่อไรที่งานสมควรจะวิ่ง ไปยังระบบย่อยเพื่อทำงาน. ค่าตั้งแต่ 1 ถึง 9 (โดยที่ 1 มีความสำคัญที่สุด) จะถูกใช้ในการกำหนดระดับความสำคัญของงานในคิวงาน.

ภายใน iSeries Navigator, คุณสามารถลากงานแล้วนำไปหย่อน หรือใช้ชื้ที่คุณสมบัติเพื่อเพิ่มหรือลดระดับความสำคัญของงานได้.

ในการเปลี่ยนระดับความสำคัญของงานในคิวงานโดยการใช้วิธีลากไปหย่อน, ให้ทำดังนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → *server-name* → **Work Management** → **Active Job Queues** or **All Job Queues**. จะมีรายการของคิวงานปรากฏขึ้นที่ด้านกรอบหน้าต่างขวา.

2. ให้เลือกคิวงานที่ต้องการทำงาน (เช่น, Qbatch). จะมีรายการของงาน ในคิวงานปรากฏขึ้นมา.
3. คลิกที่งานที่คุณต้องการเพื่อทำการย้ายหรือลากไปยังตำแหน่งระดับความสำคัญใหม่ (เช่น คุณต้องการย้าย joblist4 ที่มีระดับความสำคัญ 5 ไปไว้หลัง joblist1 ที่มีระดับความสำคัญ 3).



การใช้ชัตคุณสมบัติเพื่อเปลี่ยนระดับความสำคัญของงานในคิวงาน:

1. ใน iSeries Navigator, ให้ขยาย My Connections → server-name → Work Management → Active Job Queues หรือ All Job Queues. จะมีรายการของคิวงานปรากฏขึ้นที่ด้านกรอบหน้าต่างขวา.
2. ให้เลือกคิวงานที่ต้องการทำงาน (เช่น, Qbatch). จะมีรายการของงาน ในคิวงานปรากฏขึ้นมา.
3. คลิกขวาบนงานที่คุณต้องการเปลี่ยนระดับความสำคัญ แล้วเลือก Properties. ไดอะล็อก Properties จะปรากฏขึ้น.
4. คลิกแท็บ Job Queue.
5. ที่รายการ Priority on job queue, ให้เลือกหมายเลขระดับความสำคัญสูงขึ้น (หรือต่ำลง). ระดับความสำคัญในการรันจะอยู่ระหว่าง 0-9, โดยที่ 0 มีความสำคัญสูงสุด.
6. คลิก OK. ค่าระดับความสำคัญของคิวงานจะถูกเปลี่ยนให้กับงานของคุณ. เช่น การเปลี่ยนงานระดับความสำคัญ 4 เป็น 3 จะเป็นการย้ายงานไปอยู่ใต้รายการงานทั้งหมดที่มี ระดับความสำคัญ 3.
7. กดปุ่ม F5 เพื่อทำการรีเฟรชหน้าต่าง Job Queue.

สำหรับข้อมูลเพิ่มเติม, ให้อูที่หัวข้อ “คิวงาน” ในหน้า 56.

## การย้ายงานไปยังคิวงานอื่น

ในบางครั้ง คุณอาจต้องการย้ายงานจากคิวงานอื่นไปยังอีกอัน หนึ่ง, อาจเป็นเพราะว่าคิวงานนั้นมีความแน่นมากเกินไป และงานไม่สามารถส่งไปยังระบบย่อยได้อย่างรวดเร็ว หรือคุณอาจต้องการสร้างคิวงานพิเศษ สำหรับงานสำคัญๆ เป็นการเฉพาะ. iSeries Navigator ทำการย้ายงานระหว่างคิวงานได้อย่างสะดวกและรวดเร็ว.

งานสามารถย้ายจากคิวงานหนึ่งไปอีกคิวหนึ่งได้ด้วยกัน 2 วิธี, คือการลากและปล่อย หรือ การใช้ไดอะล็อก Move Job.

ในการลากและปล่อยจากคิวงานหนึ่งไปยังอีกอันหนึ่ง, ให้ทำดังนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → *server-name* → **Work Management** → **Job Queues** → **Active Job Queues** หรือ **All Job Queues**.
2. ดับเบิลคลิกคิวงานที่ต้องการ.
3. เลือกงานที่ต้องการย้าย.

**หมายเหตุ:** คุณสามารถเลือกงานได้หลายงานเพื่อย้ายงานเหล่านั้นไปคิวงานอื่นได้โดยการกด Ctrl+Shift และเลือกงานแต่ละงานที่คุณต้องการเคลื่อนย้าย.

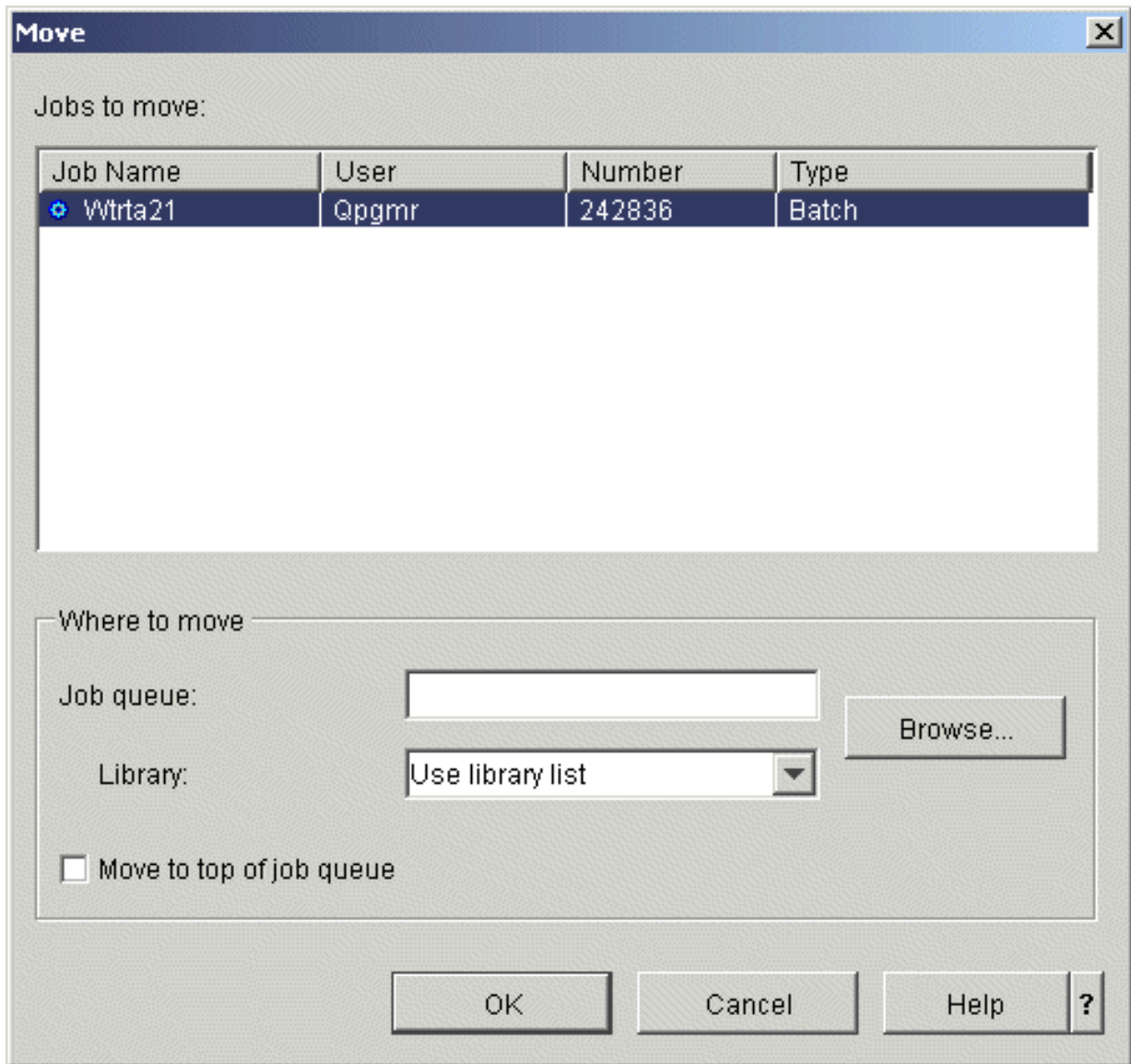
4. ลากงานไปไว้ยังคิวงานที่ต้องการ. เมื่องานหนึ่งงานหรือหลายงานถูกปล่อยลงในคิวงานอันใหม่, งานหรืองานเหล่านั้นจะถูกใส่ไว้ในตำแหน่งที่สัมพันธ์กันเหมือนกับที่เคยอยู่ในคิวงานอันเดิม. ตัวอย่างเช่น, งานที่มีความสำคัญเป็น 3 ถูกย้ายไปอยู่ในคิวงานอันใหม่จะอยู่เป็นงานอันสุดท้ายของพวกงานที่มีความสำคัญเป็น 3 ในคิวงานอันใหม่นั้น.

**หมายเหตุ:** หากคุณลากโดยคลิกเมาส์ขวา, จะมีเมนูปรากฏขึ้นมาพร้อมกับคำสั่ง **Move**, **Move to Top**, และ **Cancel**. คลิกคำสั่งที่คุณต้องการ.

ในการใช้ไอคอนล๊อค**Move...** เพื่อย้ายงานจากคิวงานหนึ่งไปสู่อีกคิวงานหนึ่ง, ให้ทำตามต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → *server-name* → **Work Management** → **Job Queues** → **Active Job Queues** หรือ **All Job Queues**.
2. คลิกคิวงานที่ต้องการ.
3. คลิกขวาที่งานที่ต้องการย้ายไปยังคิวงานอื่น (ตัวอย่างเช่น, Qdftjobd) แล้วเลือก **Move...**

**หมายเหตุ:** คุณสามารถเลือกงานได้หลายงานเพื่อย้ายงานเหล่านั้นจากคิวงานหนึ่งไปยังอีกคิวงานหนึ่ง.



4. ในฟิลด์ **Jobs to move**, ตรวจสอบว่างานที่ต้องการได้ถูกไฮไลต์ไว้. หากคุณต้องการลบไฟล์ที่เลือกไว้ ออก, ก็สามารถทำได้ด้วยการกด Ctrl และคลิกซ้ายเลือกงานที่ต้องการเอาออก.
5. ในฟิลด์ **Where to move Job Queue**, ให้พิมพ์หรือบราวซ์ไปยังคิวงาน ที่คุณต้องการย้ายงานไป (เช่น, Qusmmax).
6. ใน **Library field**, ให้พิมพ์ชื่อไลบรารีคิวงานหรือเลือกจากลิสต์ ที่มีให้ไว้.
7. คลิก **OK**.

เมื่องานหนึ่งงานหรือหลายงานถูกย้ายไปยังคิวงานอันใหม่, งานหรืองานเหล่านั้นจะถูกใส่ไว้ในตำแหน่งที่สัมพันธ์กัน เหมือนกับที่เคยอยู่ในคิวงานอันเดิม. ตัวอย่างเช่น, งานที่มีความสำคัญเป็น 3 ถูกย้ายไปอยู่ในคิวงานอันใหม่จะอยู่เป็นงานอันสุดท้ายของพวกงานที่มีความสำคัญเป็น 3 ในคิวงานอันใหม่นั้น. ถ้างานที่ถูกพักไว้ถูกย้ายไป, งานนั้นก็ยังคงถูกพักไว้เหมือนเดิม และจะวางไว้ในตำแหน่งที่สัมพันธ์เหมือนเดิมในคิวงานใหม่.

เมื่อทำเครื่องหมายในช่อง **Move to Top**, งานจะถูกย้ายไปยังบนสุดของคิวเป้าหมาย, โดยไม่สนใจสถานะปัจจุบันและระดับความสำคัญของงานนั้นเลย. (อย่างไรก็ตาม, หากงานที่อยู่บนสุดของคิวเป้าหมายมีระดับความสำคัญมากกว่าผู้ใช้ได้รับอนุญาต, ข้อความแจ้งข้อผิดพลาดจะแสดงขึ้นมาและจะไม่สามารถย้ายงานได้.) สามารถย้ายงานที่กำลังรอที่จะรัน

ไปไว้ยังบนสุดของอีกคิวงานหนึ่ง. ยกตัวอย่างเช่น, หากงานที่เลือกมีระดับความสำคัญคิวงานระดับที่ 5 และงานแรกบนคิวเป้าหมายมีระดับความสำคัญระดับที่ 3, ระดับความสำคัญของงานที่เลือกไว้จะถูกเปลี่ยนไปเป็นระดับ 3 และถูกวางไว้เหนืองานอื่นบนคิวเป้าหมาย.

งานที่ถูกพักไว้ถูกปล่อยออกมาและถูกย้ายไปไว้บนสุดของคิวงาน. ไม่สามารถย้ายงานที่กำลังรอที่จะรันไปไว้ยังบนสุดของอีกคิวงานหนึ่ง. ข้อความแสดงข้อผิดพลาดจะปรากฏขึ้นโดยระบุว่าไม่สามารถย้ายงานที่เลือกไว้ได้.

สำหรับข้อมูลเพิ่มเติม, ให้ดูที่หัวข้อ “คิวงาน” ในหน้า 56.

## การจัดการระบบย่อย

ระบบย่อยเป็นสถานที่ทำงานของงานบนเซิร์ฟเวอร์ iSeries. ชุดงานของผู้ใช้ได้ทำผ่านทางชิ้นงานต่างๆ ที่รันอยู่ในระบบย่อยจึงเป็นสิ่งสำคัญที่จะต้อง มอนิเตอร์พื้นที่เหล่านี้หากมีการทำงานที่ช้าเกิดขึ้น. ใน iSeries Navigator, คุณสามารถดูงานและคิวงานที่เชื่อมโยงกับระบบย่อย. นอกจากนั้นแล้ว, คุณยังใช้ฟังก์ชันการทำงานเดียวกันกับงานและคิวงานจากพื้นที่ใดๆ เพื่อแสดงงานและคิวงานให้เห็น.

ในการศึกษาเรื่องระบบย่อยเพิ่มเติม, ให้ดูหัวข้อต่อไปนี้:

- “การมอนิเตอร์ระบบย่อย”
- “การดูงานในระบบย่อย” ในหน้า 25
- “การเริ่มใช้งานระบบย่อย” ในหน้า 25
- “หยุดระบบย่อย” ในหน้า 25

## การมอนิเตอร์ระบบย่อย

เนื่องจากระบบย่อยมีความสำคัญกับกิจกรรมประจำวันที่ทำอยู่บนระบบ, ดังนั้น จึงเป็นสิ่งสำคัญที่คุณจะต้องตรวจสอบกิจกรรมในระบบย่อยด้วย. ภายใน subsystem description คุณสามารถระบุจำนวนของงานที่สามารถรันได้พร้อมกันในระบบย่อย โดยการเช็คค่า “คุณสมบัติของระบบย่อย” ในหน้า 72. เมื่อจำนวนงานบนระบบของคุณ เพิ่มขึ้น, คุณอาจต้องการปรับเปลี่ยนค่าจำนวนงานแอคทีฟสูงสุดในระบบย่อย ของคุณ. เลขจำนวนที่คุณใส่เข้ามานั้นควรจะเช็คเอาไว้ในระดับที่รีซอร์สที่มีอยู่จะถูกใช้งานได้อย่างเหมาะสมด้วย. หากเพิ่มจำนวนงานแอคทีฟโดยไม่ได้เพิ่มรีซอร์สที่มีให้พอดีกันแล้ว อาจทำให้ประสิทธิภาพการทำงานของระบบลดลงได้.

ในการตรวจดูค่าจำนวนงานแอคทีฟสูงสุดในระบบย่อยของคุณ, ให้ทำดังนี้:

1. ใน iSeries Navigator, ให้ขยาย My Connections → server-name → Work Management → Subsystems → Active Subsystems.
2. คลิกเมาส์ขวาที่ระบบย่อยที่คุณต้องการมอนิเตอร์.
3. เลือก Properties.

หมายเหตุ: ดูให้แน่ใจว่าคุณได้เช็คอ็อปชันนี้อย่างระมัดระวัง. หากคุณเช็คจำนวนงานแอคทีฟสูงสุดไว้สูงเกินไป, คุณอาจทำให้ระบบโดยรวมทำงานได้ช้าลง. อย่างไรก็ตาม, ถ้าคุณเช็คจำนวนงานแอคทีฟสูงสุดไว้ต่ำเกินไป, งานของคุณ อาจเริ่มเกิดการติดขัดและทำงานได้ช้าลง. สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการปรับแต่งการทำงานของระบบของคุณ, ให้ดูที่ การ

ปรับแต่งการทำงาน (บทที่ 14) ในคู่มือ V4R5 Work Management  (ประมาณ 2720 กิโลไบต์ หรือ 573 หน้า) หรือดูที่ การปรับแต่งการทำงาน.



## การดูงานในระบบย่อย

ระบบย่อยจะประสานงานกับการไหลของงานกับรีซอร์สที่งานใช้ในการรัน. iSeries Navigator จะช่วยให้คุณสามารถดูได้ว่างานใดกำลังแอ็คทีฟอยู่ (แต่อาจไม่จำเป็นต้องรันอยู่) ในระบบย่อย.

ในการดูงานในระบบย่อย, ให้ตามขั้นตอนนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → *server-name* → **Work Management** → **Subsystems** → **Active Subsystems**.
2. เลือกระบบย่อยที่คุณต้องการแสดงงานของระบบนั้น.

สำหรับรายละเอียดเพิ่มเติม, กรุณาดูหัวข้อ “ระบบย่อย” ในหน้า 58.

## การเริ่มใช้งานระบบย่อย

เมื่อระบบย่อยเริ่มทำงาน, ระบบจะจัดสรรรีซอร์สที่กำหนดเอาไว้ใน “Subsystem description” ในหน้า 59 — ให้กับระบบย่อยนั้น เช่น พูลหน่วยความจำ, เวิร์กสเตชัน, และคิวงาน. รีซอร์สเหล่านี้จะเตรียมระบบย่อยให้พร้อมใช้งาน.

สำหรับรายละเอียดเกี่ยวกับลำดับเหตุการณ์ที่จะถูกกระตุ้นให้ทำงานขึ้นเมื่อระบบย่อยถูกเรียกใช้งาน, ให้ดูที่ “สิ่งที่เกิดขึ้นเมื่อระบบย่อยเริ่มทำงาน” ในหน้า 74.

ในการสตาร์ทระบบย่อยขึ้นมา, ให้ทำตามขั้นตอนต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → *server-name* → **Work Management** → **Subsystems**.
2. คลิกขวาไปบน **Subsystems**, แล้วเลือก **Start Subsystem**.
3. ระบุชื่อและไลบรารีที่จะให้ระบบย่อยเริ่มสตาร์ทที่ขึ้น, หรือคลิก **Browse...** เพื่อเลือกจากลิสต์รายการของระบบย่อย.
4. คลิก **OK**.

## หยุดระบบย่อย

คุณสามารถใช้ iSeries Navigator เพื่อหยุดระบบย่อยที่แอ็คทีฟอยู่หนึ่งระบบหรือมากกว่า และระบุสิ่งที่เกิดขึ้นให้กับงานแอ็คทีฟที่กำลังถูกประมวลผลอยู่. จะไม่มีงานใหม่หรือขั้นตอนตามเส้นทางใดๆ ถูกเรียกให้ทำงานเมื่อระบบย่อยถูกสั่งให้หยุดทำงาน.

เมื่อระบบย่อยถูกหยุด, คุณสามารถระบุสิ่งที่เกิดขึ้นให้กับงานแอ็คทีฟที่กำลัง ถูกระบบประมวลผลอยู่ได้. เช่น, คุณสามารถระบุให้งานทุกงานในระบบย่อยหยุดการทำงานทันที (**Immediate**), หรือระบุให้งานได้รับอนุญาตให้ทำต่อไปจนเสร็จสิ้นการประมวลผลก่อนที่ระบบย่อยจะจบลง (**Controlled**).

**สิ่งสำคัญ:** ขอแนะนำให้จบการทำงานของระบบย่อยด้วยอ็อปชัน **Controlled** เท่าที่จะเป็นไปได้. ซึ่งจะช่วยให้งานแอ็คทีฟจบการทำงานด้วยตัวของมันเอง. ใช้อ็อปชันนี้เพื่อป้องกันให้งานเสร็จสิ้นการทำงานก่อนที่ระบบย่อยจะจบการทำงาน. ซึ่งสิ่งนี้จะช่วยให้โปรแกรมที่กำลังรันอยู่ทำการ cleanup (กระบวนการสิ้นสุดงาน) ให้เรียบร้อย. การระบุค่า **Immediate** อาจก่อให้เกิดผลที่ไม่พึงปรารถนาจากข้อมูลที่มีการอัปเดตครั้งๆ กลางๆ เป็นต้น.

ยังมีอ็อปชันนอกจากนี้ในการใช้หยุดระบบย่อยอีก. จะมีการอธิบายถึงอ็อปชันเหล่านี้ไว้อย่างละเอียดในคำอธิบายที่เกี่ยวข้องกับไอคอน **Stop Subsystem** ใน iSeries Navigator.

ในการหยุดระบบย่อย, ให้ทำตามขั้นตอนต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → *server-name* → **Work Management** → **Subsystems** → **Active Subsystems**.
2. คลิกขวาไปบนระบบย่อยหนึ่งหรือหลายอันที่ต้องการให้หยุดทำงาน, แล้วเลือก **Stop....**
3. ระบุอ็อปชันที่ต้องการใช้ในการหยุดการทำงานของระบบย่อย.
4. คลิก **Stop**.

## การจัดการพูลหน่วยความจำ

พูลหน่วยความจำจะจัดสรรหน่วยความจำที่ระบบย่อยจะต้องใช้ในการรันงาน. ถ้าหากระบบย่อย ได้รับหน่วยความจำมากเกินไปและระบบย่อยอื่นได้ไม่เพียงพอ, งานในระบบย่อยจะเริ่มทำงาน อย่างด้อยคุณภาพ. เซิร์ฟเวอร์ iSeries ได้จัดเตรียมตัวปรับแต่งโดยดีฟอลต์ให้เพื่อสนอง ความต้องการของผู้ใช้ที่มีจำนวนมาก. อย่างไรก็ตาม, ถ้าหากความต้องการของคุณเกินกว่าความสามารถของตัวปรับแต่งระบบ, คุณก็จำเป็นต้องเรียนรู้วิธีการจัดการกับพูลหน่วยความจำด้วย. คุณสามารถแอดดเชสค่าปรับแต่งการทำงานได้ใน iSeries Navigator โดยผ่านทาง Properties สำหรับพูลหน่วยความจำที่ใช้งานร่วมกันของหน้า **Tuning**. สำหรับข้อมูลเพิ่มเติม, ให้ดูที่ ประสิทธิภาพการทำงาน. หากคุณต้องการข้อมูลเพิ่มเติมเกี่ยวกับวิธีการปรับแต่งการทำงานบนระบบของท่าน, ให้ดู การปรับแต่งการทำงาน.

หากต้องการจัดการพูลหน่วยความจำ, ให้ดูหัวข้อเหล่านี้:

- “การมอนิเตอร์จำนวนงานในพูลหน่วยความจำ”
- “การมอนิเตอร์จำนวนระบบย่อยที่ใช้พูลหน่วยความจำ” ในหน้า 28
- “ตรวจสอบการใช้พูลหน่วยความจำ” ในหน้า 28
- “การเปลี่ยนขนาดของพูลหน่วยความจำ” ในหน้า 29

## การมอนิเตอร์จำนวนงานในพูลหน่วยความจำ

เนื่องจากพูลหน่วยความจำจะให้หน่วยความจำแก่ระบบย่อยเพื่อรันงาน, จึงเป็นสิ่งสำคัญที่จะต้องตรวจสอบจำนวนงานที่กำลังรันอยู่ในพูลหน่วยความจำ. เพราะหากงานมีจำนวนมากเกินไปในพูลหน่วยความจำหนึ่งๆ จะทำให้มีผลกระทบต่อ การทำงานของระบบทั้งหมด.

ในการมอนิเตอร์จำนวนงานในพูลหน่วยความจำ, ให้ทำสิ่งต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → *server-name* → **Work Management** → **Memory Pools** → **Active Pools** หรือ **Shared Pools**.
2. คลิกขวาที่พูลหน่วยความจำที่ต้องการใช้ (เช่น, Base) แล้วเลือก **Jobs**. โดอะล็อกจะปรากฏขึ้นเพื่อแสดงลิสต์งานภายในพูลหน่วยความจำ.

คุณสามารถดูจำนวนของ thread ในพูลหน่วยความจำโดยการดูคอลลัมน์ Thread Count. การนับ thread จะเป็นข้อมูลเพิ่มเติมตัวหนึ่งในการบอกถึงปริมาณ activity ที่เกิดขึ้นในพูลหน่วยความจำ.



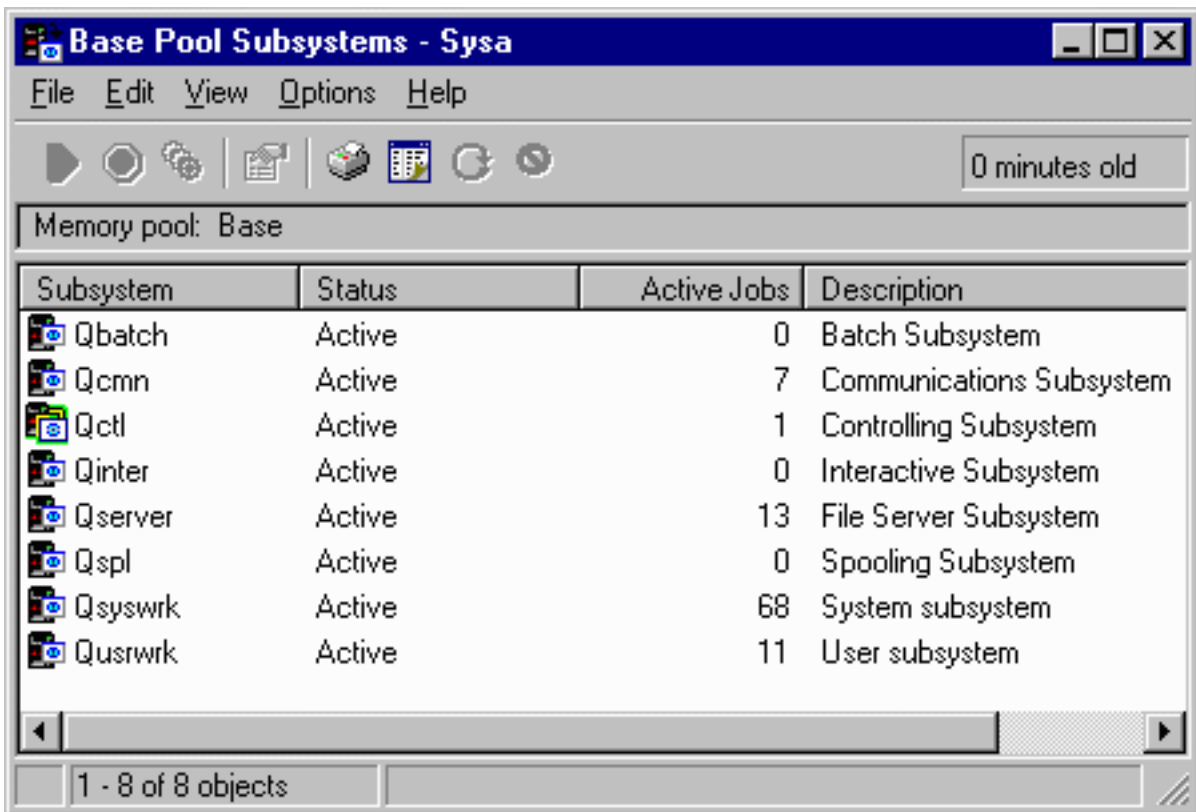


## การมอนิเตอร์จำนวนระบบย่อยที่ใช้พูลหน่วยความจำ

ระบบย่อยจะได้รับการจัดสรรหน่วยความจำจำนวนหนึ่งเพื่อใช้รันงาน. ดังนั้น จึงเป็นสิ่งสำคัญมากที่จะต้องรู้ว่ามียระบบย่อยอยู่ที่ระบบที่ดึงหน่วยความจำออกมาใช้จากพูลเดียวกัน. เมื่อคุณสามารถรู้ได้ว่ามียระบบย่อยอยู่ที่ระบบที่จะส่งงานไปไว้ที่พูลและมีกี่งานที่จะรันในพูล, คุณอาจต้องการเปลี่ยนขนาดและ activity level ของพูลเพื่อลดการแย่งใช้รีซอร์สลง.

ในการมอนิเตอร์จำนวนระบบย่อยที่ใช้พูลหน่วยความจำ, ให้ทำตามต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย My Connections → *server-name* → Work Management → Memory Pools → Active Pools หรือ Shared Pools.
2. คลิกขวาที่พูลหน่วยความจำที่ต้องการ แล้วเลือก Subsystems (ตัวอย่างเช่น, Base).



Subsystem	Status	Active Jobs	Description
Qbatch	Active	0	Batch Subsystem
Qcmn	Active	7	Communications Subsystem
Qctl	Active	1	Controlling Subsystem
Qinter	Active	0	Interactive Subsystem
Qserver	Active	13	File Server Subsystem
Qspl	Active	0	Spooling Subsystem
Qsyswrk	Active	68	System subsystem
Qusrwrk	Active	11	User subsystem

จากหน้าต่างนี้, คุณสามารถระบุจำนวนระบบย่อยที่กำลังใช้หน่วยความจำอันใดอันหนึ่งเฉพาะเพื่อรันงานอยู่.

สำหรับข้อมูลเพิ่มเติม, ให้อูที่ “activity level ของพูลหน่วยความจำ” ในหน้า 75.

## ตรวจสอบการใช้พูลหน่วยความจำ

การตรวจสอบหน่วยความจำที่พูลหน่วยความจำใช้เป็นสิ่งสำคัญอย่างมาก. การตรวจสอบสิ่งดังกล่าวจะช่วยให้คุณปรับแต่งพูลให้รันอย่างมีประสิทธิภาพสูงสุด ซึ่งสิ่งนี้จะรักษาให้วงรอบของงานรันอย่างราบรื่น. ใน iSeries Navigator, คุณสามารถตรวจสอบปริมาณหน่วยความจำที่พูลกำลังใช้งานอยู่ได้.

ในการตรวจสอบการใช้งานหน่วยความจำ, ให้ทำตามนี้:

1. ใน iSeries Navigator, ให้ขยาย My Connections → *server-name* → Work Management → Memory Pools → Active Pools or Shared Pools.
2. คลิกขวาที่พูลหน่วยความจำที่ต้องการทำงานด้วย (ตัวอย่างเช่น, Interactive) แล้วเลือก Properties.
3. คลิกเลือกแท็บ Configuration. ฟิลด์ Current ภายใต้ Size จะแสดงขนาดของหน่วยความจำที่พูลกำลังใช้งานอยู่ให้เห็น.

หมายเหตุ: คุณสามารถดูขนาดของพูลหน่วยความจำปัจจุบัน เมื่อคลิก Active Pools หรือ Shared Pools. Current Size (เป็นหน่วยเมกะไบต์) เป็นคอลัมน์ฟิลด์ที่คุณจะเห็นเมื่อลิสต์ของพูลหน่วยความจำปรากฏขึ้นในกรอบด้านขวาของ iSeries Navigator.

สำหรับข้อมูลเพิ่มเติม, โปรดดูที่ “พูลหน่วยความจำ” ในหน้า 75.

## การเปลี่ยนขนาดของพูลหน่วยความจำ

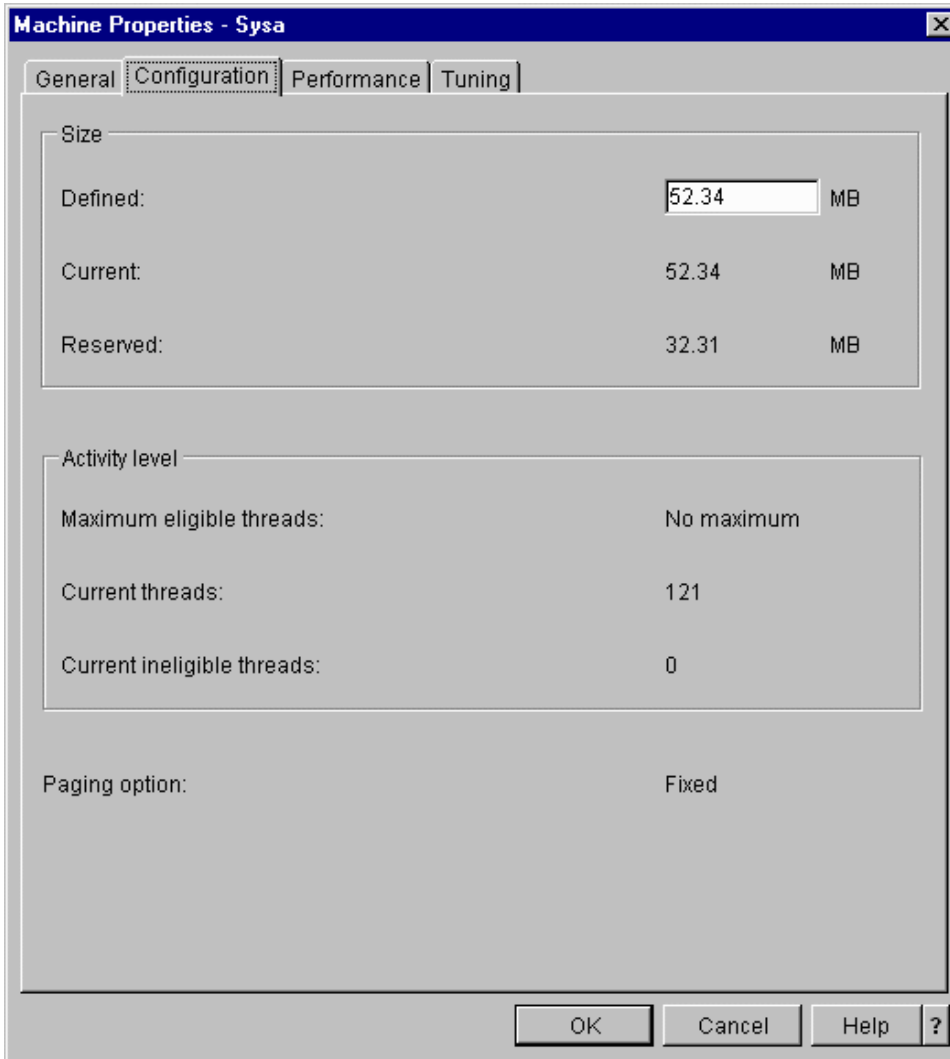
ขนาดของพูลหน่วยความจำจะมีผลโดยตรงกับปริมาณของงานที่ระบบย่อยจะสามารถประมวลผลได้. หน่วยความจำยิ่งมาก, ระบบย่อยก็ยิ่งทำให้งานเสร็จสมบูรณ์ได้มากขึ้นด้วย. ใน iSeries Navigator, คุณสามารถเปลี่ยนขนาดของหน่วยความจำที่กำหนด (หรือที่มีอยู่) ให้กับพูลได้. อย่างไรก็ตาม, การตรวจสอบระบบอย่างระมัดระวังก่อนจะเริ่มเปลี่ยน พารามิเตอร์ของพูลหน่วยความจำก็เป็นสิ่งที่สำคัญอย่างยิ่ง. และต้องทำการตรวจสอบ ระดับดังกล่าวซ้ำๆ ช่วง เพื่อดูว่าอาจต้องมีการปรับเปลี่ยนให้เหมาะสมยิ่งขึ้น อีก.

หมายเหตุ: ตรวจสอบให้แน่ใจว่าคุณได้ปิดตัวปรับแต่งระบบก่อนทำการเปลี่ยน ขนาดของพูลหน่วยความจำด้วยมือ. ตัวปรับแต่งระบบจะปรับขนาดของพูลหน่วยความจำที่แบ่ง ใช้โดยอัตโนมัติให้กับปริมาณงานที่ระบบกำลังทำงานอยู่. หากตัวปรับแต่งระบบไม่ถูก ปิด การเปลี่ยนแปลงที่คุณทำด้วยมืออาจถูกเปลี่ยนคืนด้วยตัวปรับแต่งโดยอัตโนมัติ.

ในการเปลี่ยนขนาดของพูลหน่วยความจำ, ให้ทำสิ่งต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย My Connections → *server-name* → Work Management → Memory Pools → Active Pools หรือ Shared Pools.
2. คลิกขวาที่พูลหน่วยความจำที่ต้องการ (เช่น, Interactive) แล้วเลือก Properties. หน้าต่าง Memory Pool Properties จะปรากฏขึ้นมา.

### 3. คลิกเลือกแท็บ Configuration.



ที่หน้า Configuration ของหน้าต่างคุณสมบัติ, คุณสามารถเปลี่ยนขนาดหน่วยความจำที่กำหนดไว้. หน่วยความจำที่กำหนดไว้เป็นขนาดหน่วยความจำใหญ่ที่สุดที่พูลสามารถใช้ได้. ตัวเลขที่คุณใส่ให้ในที่นี้ควรจะเป็นขนาดหน่วยความจำที่คุณคิดว่าพูลจะต้องการใช้เพื่อสนับสนุนระบบย่อยที่พูลให้บริการอยู่.

**ข้อควรพิจารณาสำหรับ Base pool:** Base pool เป็นพูลหน่วยความจำที่ไม่มีขนาดหน่วยความจำที่กำหนดไว้. แต่มีขนาดหน่วยความจำต่ำสุดที่จำเป็นในการใช้รัน. Base pool ประกอบด้วยสิ่งต่างๆ ที่ไม่ได้ถูกจัดสรรไว้ในที่อื่น. ตัวอย่างเช่น, คุณมีหน่วยความจำอยู่ 1000MB บนระบบ; ซึ่ง 250MB ได้ถูกจัดสรรให้กับ Machine pool และอีก 250MB ได้จัดสรรให้กับ Interactive pool. ดังนั้น ยังมีอีก 500MB ที่ยังไม่ได้ถูกจัดสรรให้กับสิ่งใด. หน่วยความจำที่ ไม่ได้ถูกจัดสรรดังกล่าวจะเก็บอยู่ใน Base pool จนกว่าจะมีความจำเป็นต้องใช้. ให้ใช้ความระมัดระวังในโยกย้ายหน่วยความจำ. เพราะการโยกย้ายหน่วยความจำจากพูลหนึ่งไปยังอีกพูลหนึ่งอาจช่วยแก้ปัญหาให้กับระบบย่อยหนึ่งได้, แต่ก็อาจสร้างปัญหาให้กับระบบย่อยอื่นๆ ซึ่งก็จะทำให้ประสิทธิภาพโดยรวมของระบบลดลงด้วย.


สำหรับข้อมูลเพิ่มเติม, โปรดดู “พูลหน่วยความจำ” ในหน้า 75.

## การจัดการบันทึกการใช้งาน

งานส่วนใหญ่บน iSeries ของคุณจะมีบันทึกการใช้เชื่อมโยงอยู่ด้วย. บันทึกการใช้งานจะให้รายละเอียดหลายสิ่งแก่ผู้ใช้ เช่น เวลาที่งานเริ่ม, เวลาที่งานจบ, คำสั่งใดที่รันอยู่, การแจ้งความล้มเหลว, และข้อความแสดงจุดผิดพลาด. ข้อมูลนี้จะให้แนวทางอันเป็นประโยชน์แก่ผู้ใช้งานว่าวงรอบของงานรันไปเช่นไร.

ค้นหาวิธีการแอ็กเซสบันทึกการใช้งานของงานแอ็คทีฟและการแอ็กเซสพริเตอร์เอาต์พุตของบันทึกการใช้งาน.

- “การเข้าถึงบันทึกการทำงานสำหรับงานแอ็คทีฟ, รวมถึงงานเซิร์ฟเวอร์”
- “การแอ็กเซสพริเตอร์เอาต์พุต” ในหน้า 32

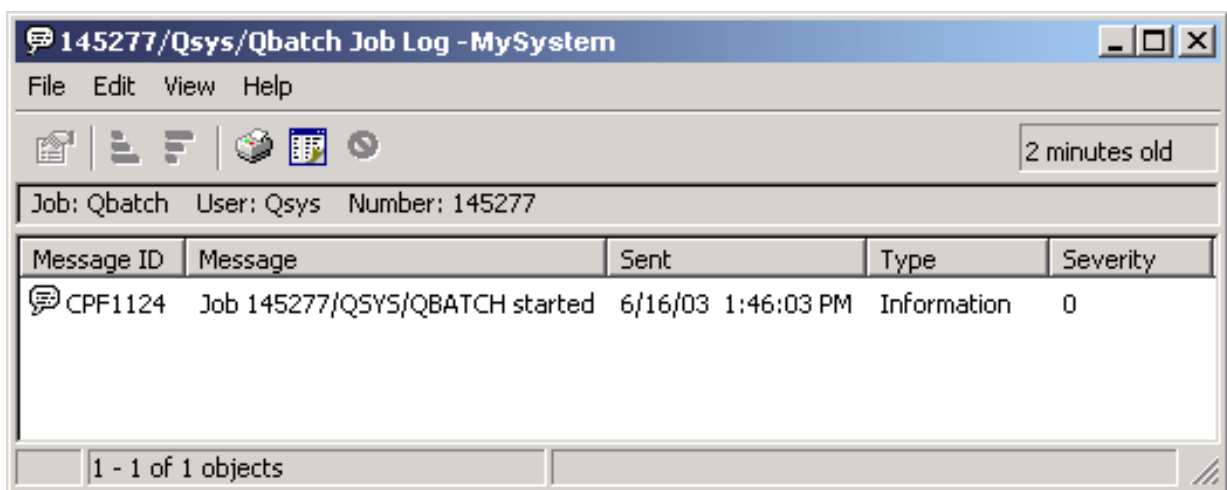
สำหรับข้อมูลเพิ่มเติม, ให้อ่าน Job logs ในบทที่ 5 ของคู่มือ V4R5 Work Management  (ประมาณ 2720 กิโลไบต์ หรือ 573 หน้า).

### การเข้าถึงบันทึกการทำงานสำหรับงานแอ็คทีฟ, รวมถึงงานเซิร์ฟเวอร์


เป็นสิ่งจำเป็นที่จะต้องรู้วิธีที่จะแอ็กเซสเข้าสู่บันทึกการใช้งาน เนื่องจาก บันทึกการใช้งานจะจัดบันทึกข้อมูลเกี่ยวกับงานให้ในขณะทำการรันงาน.

ในการแอ็กเซสเข้าสู่บันทึกการใช้งานเพื่อดู “งานแอ็คทีฟและ inactive” ในหน้า 36 หรือ “งานเซิร์ฟเวอร์” ในหน้า 44, โปรดดำเนินการต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → *server-name* → **Work Management** → **Active Jobs** หรือ **Server Jobs**. **หมายเหตุ:** คุณสามารถดูบันทึกการใช้งานจากที่ใดก็ได้ภายในการจัดการระบบงานที่คุณแอ็กเซสเข้าไปในงาน (เช่น ผ่านทางพื้นที่ระบบย่อยหรือพื้นที่พูลหน่วย ความจำ).
2. คลิกขวาไปบนงาน (เช่น, Qbatch) และเลือก **Job Log**. ใช้ภาพที่อยู่ ด้านล่างเพื่อดูชนิดของข้อมูลที่คุณสามารถหาได้ในบันทึกการทำงาน. สำหรับรายละเอียดเพิ่มเติม, โปรดดูคำอธิบายในไดอะล็อก **Job Log**.



แสดงรายละเอียดเพิ่มเติมของข้อความ,  คลิกเมาส์ขวาที่ข้อความ และเลือก **Properties**.

ซึ่งจะแสดงข้อมูลรายละเอียดเพิ่มเติม.  ไดอะล็อกขึ้นนี้แสดงรายละเอียดเกี่ยวกับข้อความ พร้อมกับคำอธิบายข้อความ. ข้อความอย่างละเอียดจะช่วยบอกข้อมูลเพื่อให้คุณสามารถแก้ไขปัญหาได้.

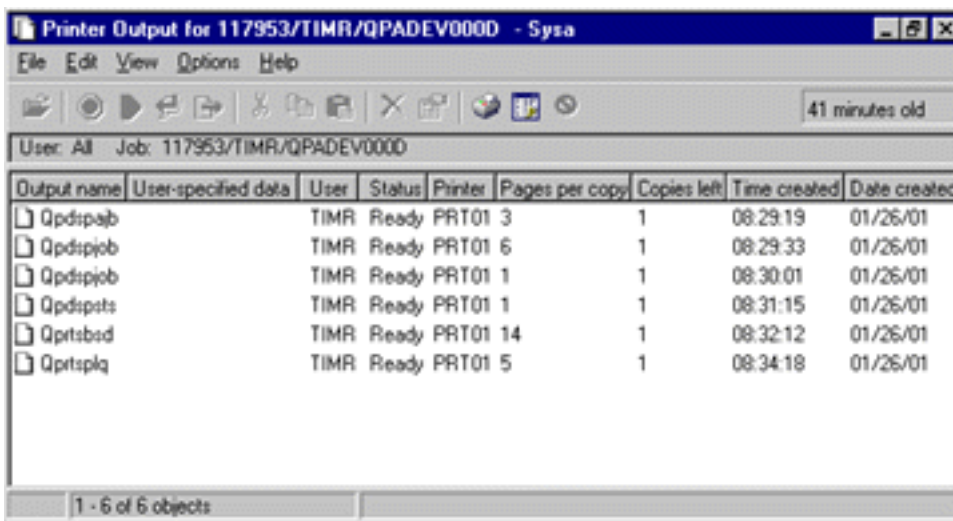
สำหรับข้อมูลเพิ่มเติม, โปรดดูที่ “บันทึกการใช้งาน” ในหน้า 52 หรือดูคำอธิบาย.

## การแอ็กเซสพรินเตอร์เอาต์พุต

เนื่องจากคุณมีตัวเลือกในการ“การดึงพรินเตอร์เอาต์พุตออกมา”ในหน้า 48 จากงานเมื่องานรันเสร็จสิ้นแล้ว (โดยแยกพรินเตอร์เอาต์พุตออกจากงานอย่างสมบูรณ์), คุณสามารถแอ็กเซสพรินเตอร์เอาต์พุตของคุณได้ใน iSeries Navigator ผ่านทาง Basic Operations หรือผ่านทาง Work Management.

ในการแอ็กเซสพรินเตอร์เอาต์พุตของงานผ่าน Basic Operations, ให้ทำดังต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย My Connections → *server-name* → Basic Operations.
2. เลือก Job. งานทั้งหมดของผู้ใช้ปัจจุบันจะปรากฏขึ้น. โปรดดูใน “การค้นหางานบนเซิร์ฟเวอร์ iSeries” ในหน้า 10 สำหรับวิธีต่างๆ ในการค้นหางานบนเซิร์ฟเวอร์ iSeries.
3. คลิกขวางานที่คุณต้องการแสดงพรินเตอร์เอาต์พุต และคลิกที่ Printer Output. ไอคอน Printer Output จะปรากฏขึ้น.



Output name	User-specified data	User	Status	Printer	Pages per copy	Copies left	Time created	Date created
Qpdrspab		TIMR	Ready	PRT01	3	1	08:29:19	01/26/01
Qpdrspjob		TIMR	Ready	PRT01	6	1	08:29:33	01/26/01
Qpdrspjob		TIMR	Ready	PRT01	1	1	08:30:01	01/26/01
Qpdrspst		TIMR	Ready	PRT01	1	1	08:31:15	01/26/01
Qpdrsbad		TIMR	Ready	PRT01	14	1	08:32:12	01/26/01
Qpdrspq		TIMR	Ready	PRT01	5	1	08:34:18	01/26/01

หากต้องการแอ็กเซสพรินเตอร์เอาต์พุตผ่านทางโพลเดอร์ Output Queues, โปรดปฏิบัติตามดังต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย My Connections → *server-name* → Work Management → Output Queues.
2. เลือกเอาต์พุตคิวที่คุณต้องการแสดงผลพรินเตอร์เอาต์พุตด้วย (เช่น, Qprint2). พรินเตอร์เอาต์พุตภายในเอาต์พุตคิวจะปรากฏขึ้น.

## การจัดการเอาต์พุตคิว

พรินเตอร์เอาต์พุตทำงานอยู่บนเอาต์พุตคิว. เอาต์พุตคิวกำหนดลำดับที่พรินเตอร์เอาต์พุตจะถูกประมวลผลโดยอุปกรณ์เครื่องพิมพ์. เมื่อจัดการกับเอาต์พุตคิวของคุณ, คุณสามารถแน่ใจว่าจะได้การประมวลผลพรินเตอร์เอาต์พุตที่ราบรื่น.

ด้วย“แอ็ททริบิวต์ของเอาต์พุตคิว”ในหน้า 79, คุณสามารถดำเนินการภารกิจให้เสร็จสมบูรณ์ได้จากโพลเดอร์ Output Queues:

- ดูเอาต์พุตคิวบนระบบ
- ดูคุณสมบัติของเอาต์พุตคิว
- ยึดเอาต์พุตคิวไว้
- ปลปล่อยเอาต์พุตคิว

- เคลียร์เอาต์พุตคิว
- ดูเอาต์พุตที่กำลังรออยู่บนเอาต์พุตคิว
- ย้ายเอาต์พุตระหว่างและภายในเอาต์พุตคิว
- เปลี่ยนคุณสมบัติของเอาต์พุตคิว

ใช้หัวข้อย่อยเหล่านี้เพื่อดูเอาต์พุตคิวบนระบบของคุณ, เคลียร์เอาต์พุตคิว, และย้ายเอาต์พุตพริเตอร์ระหว่างและภายในเอาต์พุตคิว.

- “การดูเอาต์พุตคิวบนระบบ”
- “ย้ายเอาต์พุตระหว่างและภายในเอาต์พุตคิว”
- “เคลียร์เอาต์พุตคิว” ในหน้า 34

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับภารกิจอื่นๆ ที่คุณสามารถทำเสร็จสมบูรณ์ด้วยเอาต์พุตคิว, ให้ดูที่คำอธิบาย iSeries Navigator แบบออนไลน์. สำหรับข้อมูลเพิ่มเติม, ให้ดูที่ “เอาต์พุตคิว” ในหน้า 77.

## การดูเอาต์พุตคิวบนระบบ

เอาต์พุตคิวจะกำหนดลำดับที่พริเตอร์เอาต์พุตถูกส่งไปยังอุปกรณ์การพิมพ์.

หากต้องการดูเอาต์พุตคิวบนระบบ, ให้ทำตามขั้นตอนต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → *server-name* → **Work Management**.
2. เลือก **Output Queues**.

ใน iSeries Navigator, คุณสามารถปรับรายการเอาต์พุตคิวที่คุณกำลังดูโดยใช้ไดอะล็อก Include.... ไดอะล็อก Include... จะอนุญาตให้คุณใส่เงื่อนไขกับสิ่งที่แสดงอยู่ใน iSeries Navigator. ยกตัวอย่างเช่น, คุณสามารถรัน Include... เพื่อแสดงเอาต์พุตคิวบางรายการเท่านั้น. หากต้องการใช้งานฟังก์ชัน include, ให้ใช้เมนู **View**, และ **Customize this View**.

สำหรับข้อมูลเพิ่มเติม, ให้ดูที่ “เอาต์พุตคิว” ในหน้า 77.

## ย้ายเอาต์พุตระหว่างและภายในเอาต์พุตคิว

ในบางครั้งคุณต้องย้ายเอาต์พุตออกจากคิวหนึ่งไปยังอีกคิวหนึ่งหรืออาจต้องย้ายเอาต์พุตไปยังระดับความสำคัญที่สูงขึ้นเพื่อให้เอาต์พุตนั้นถูกส่งไปยังอุปกรณ์การพิมพ์เร็วขึ้น. อาจเกิดเหตุการณ์นี้ขึ้นหากมีเอาต์พุตมากเกินไปบนเอาต์พุตคิว.

คุณสามารถย้ายเอาต์พุตออกจากเอาต์พุตคิวหนึ่งไปยังอีกเอาต์พุตคิวหนึ่งหรือคุณอาจย้ายเอาต์พุตภายในเอาต์พุตคิวได้.

หากต้องการย้ายเอาต์พุตระหว่างเอาต์พุตคิว, ให้ทำตามขั้นตอนต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → *server-name* → **Work Management** → **Output Queues**.
2. ดับเบิลคลิกเอาต์พุตคิวที่มีเอาต์พุตที่คุณต้องการย้าย.
3. คลิกเอาต์พุตที่คุณต้องการย้าย, และลากเอาต์พุตนั้นไปไว้ยังเอาต์พุตคิวที่ตำแหน่งที่คุณต้องการย้ายเอาต์พุตนั้นไปไว้ในกรอบหน้าต่างด้านซ้ายของ iSeries Navigator.

**หมายเหตุ:** เอาต์พุตถูกย้ายไปสู่คิวเป้าหมายและวางอยู่บนคิวตามระดับความสำคัญ.



หากต้องการย้ายเอาต์พุตภายในเอาต์พุตคิว, ให้ทำตามขั้นตอนต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → *server-name* → **Work Management** → **Output Queues**.
2. ดับเบิลคลิกเอาต์พุตคิวที่มีเอาต์พุตที่คุณต้องการย้าย.
3. คลิกเอาต์พุตที่คุณต้องการย้าย, และลากไปไว้ยังเอาต์พุตในคิวที่คุณต้องการย้ายเอาต์พุตนั้นไปไว้ในภายหลัง.

**หมายเหตุ:** เอาต์พุตถูกย้ายโดยตรงหลังจากเอาต์พุตเป้าหมาย.

สำหรับข้อมูลเพิ่มเติม, ให้ดูที่ “เอาต์พุตคิว” ในหน้า 77.

## เคลียร์เอาต์พุตคิว

เมื่องานสร้างพรินเตอร์เอาต์พุต งานจะถูกส่งไปยังเอาต์พุตคิวเพื่อพิมพ์. ส่วนใหญ่แล้วคุณจะไม่พิมพ์พรินเตอร์เอาต์พุตทั้งหมด. iSeries Navigator จะให้คุณสามารถเคลียร์เอาต์พุตคิวของคุณได้โดยใช้ไอพชั่น **Clear**. การเคลียร์เอาต์พุตคิวจะลบเอาต์พุตทั้งหมดออกจากคิว.

หากต้องการเคลียร์เอาต์พุตคิว, ให้ทำตามขั้นตอนต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → *server-name* → **Work Management** → **Output Queues**.
2. คลิกขวาเอาต์พุตคิวที่คุณต้องการเคลียร์, และเลือก **Clear**.

สำหรับข้อมูลเพิ่มเติม, โปรดดูที่ “เอาต์พุตคิว” ในหน้า 77.

---

## โครงสร้างของระบบ

คุณสามารถแยกการจัดการระบบงานออกเป็นห้าส่วนด้วยกัน คือ งาน, คิวงาน, ระบบย่อย, พูลหน่วยความจำ, และเอาต์พุต. ในแต่ละส่วนจะมีคำศัพท์และหลักการเป็นของตนเอง. โดยตัวมันเองแล้ว แต่ละส่วนจะผลิตข้อมูลออกมาแตกต่างกัน แต่อย่างไรก็ตาม, เมื่อผนวกรวมเข้าด้วยกันแล้ว, ทุกส่วนจะรวมเป็นทูลที่มีศักยภาพในการจัดการตัวงานบนเซิร์ฟเวอร์ iSeries.

หากต้องการศึกษาเพิ่มเติมเกี่ยวกับส่วนการทำงานต่างๆ ภายในการจัดการระบบงาน, ให้ดูหัวข้อต่อไปนี้:

### “งาน” ในหน้า 35

ศึกษาเกี่ยวกับงานประเภทต่างๆ และคุณสมบัติของงาน. และศึกษา การปฏิบัติงานต่างๆ ที่คุณสามารถทำกับงานนั้นได้.

### “คิวงาน” ในหน้า 56

ศึกษาเกี่ยวกับบทบาทของคิวงานในวงจรชีวิตของการจัดการระบบงาน.

### “ระบบย่อย” ในหน้า 58

ศึกษาเกี่ยวกับระบบย่อยประเภทต่างๆ และคุณสมบัติของระบบย่อย.

### “พูลหน่วยความจำ” ในหน้า 75

เพื่อศึกษาเกี่ยวกับพูลหน่วยความจำประเภทต่างๆ และคุณสมบัติของพูลหน่วยความจำ.

### “เอาต์พุตคิว” ในหน้า 77

ศึกษาว่าเกิดอะไรขึ้นกับงานเมื่อเสร็จสิ้นการรันแล้ว.

**หมายเหตุ:** iSeries Navigator เรียกใช้ application programming interfaces (APIs) ในการดึงข้อมูลจากระบบ iSeries. APIs คืออุปกรณ์อินพุตและเอาต์พุตของ iSeries Navigator สำหรับเซิร์ฟเวอร์ iSeries. สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ APIs, ให้ดู Application programming interfaces (APIs) หรือ API Concepts.



## งาน

งาน (Work) ทั้งหมดที่ทำอยู่บนระบบจะกระทำผ่านชิ้นงาน(job). งานแฉีกทีฟประกอบด้วย thread (initial thread) อย่างน้อยหนึ่ง thread และอาจประกอบด้วย thread สำรองเพิ่มเติมด้วย. thread เป็นหน่วยย่อยอิสระของชุดงาน. คุณสมบัติของงานจะถูกแบ่งใช้ใน thread ของงานนั้น, อย่างไรก็ตาม, thread ยังมีชุดคุณสมบัติของตนเองต่างหาก, เช่น call stack. คุณสมบัติของงานประกอบด้วยรายละเอียดเกี่ยวกับวิธีที่งานจะถูกประมวลผล. งานจะทำหน้าที่เป็นเจ้าของคุณสมบัติที่แบ่งใช้กันใน thread ภายในงานนั้น. การจัดการระบบงานจะจัดเตรียมวิธีควบคุมให้งานดำเนิน ไปโดยสำเร็จสมบูรณ์บนระบบผ่านทางคุณสมบัติของงาน.

คุณสมบัติทั่วไปของงานจะกำหนดวิธีที่ระบบรันงานแต่ละงาน. คุณสมบัติบางประการจะถูกจัดกลุ่มเข้าด้วยกันใน “Job description” เพื่อให้จัดการงานต่างๆ ได้ง่ายขึ้น. ระบบจะรู้ว่าจะรับคุณสมบัติอันใดและ เมื่อใด, ขึ้นอยู่กับวิธีที่ใช้กำหนดคุณสมบัติของงาน. ระบบ iSeries จะรันงานประเภทต่างๆ เพื่อสนองความต้องการที่หลากหลาย. ประเภทของงานส่วนใหญ่จะใช้ job description.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับงาน ให้ดูหัวข้อต่อไปนี้:

“งานแฉีกทีฟและ inactive” ในหน้า 36  
ศึกษาว่างานแฉีกทีฟและ inactive คืออะไร.

“ประเภทของงาน” ในหน้า 36  
ศึกษาประเภทต่างๆ ของงานที่รันอยู่บน iSeries.

“คุณสมบัติของงาน” ในหน้า 46  
ศึกษาการทำงานกับคุณสมบัติงาน.

“การปฏิบัติงาน” ในหน้า 17  
ศึกษาวิธีจัดการงานผ่านทาง iSeries Navigator.

“thread” ในหน้า 53  
ศึกษาความแตกต่างระหว่าง thread และ job.

“คิวงาน” ในหน้า 56  
ศึกษาการที่งานจะออกจากการรอในคิวงานไปสู่การทำงานจริง.

**ช่วงชีวิตของงาน**  
ศึกษาถึงสิ่งที่เกิดในระหว่างช่วงชีวิตของงานจากจุดเริ่มต้นไปถึงจุดสิ้นสุด.


**หมายเหตุ:**

สามารถเรียกใช้ APIs, เช่น Open List of Jobs (QGYOLJOB) และ Retrieve Job Information (QUSRJOB), เพื่อรับข้อมูลบนงาน. สำหรับข้อมูลเพิ่มเติมเรื่อง APIs, ให้ดูที่ Application programming interfaces (APIs).

## Job description

job description ใช้ในการสร้างชุดคุณสมบัติของงานที่จะถูกบันทึกไว้และเตรียมไว้ใช้ในลักษณะงานต่างๆ. job description จะถูกใช้เป็นแหล่งข้อมูลให้กับ “คุณสมบัติของงาน” ในหน้า 46 ซึ่งจะช่วยบอกให้ระบบรู้วิธีการรันงานนั้น. คุณสมบัติดังกล่าวจะ

บอกระบบให้รู้ว่าเมื่อไรจะรันงาน, จะไปเอางานมาได้จากที่ไหน, และจะรันงานนั้นอย่างไร. Job descriptions จะถูกใช้งานจากงานประเภท“งานแบบ autostart” ในหน้า 37, “งานแบบแบตช์” ในหน้า 37, “งานโต้ตอบ” ในหน้า 38, และประเภทงาน“งาน Prestart” ในหน้า 39. คุณสามารถใช้ job description เดียวกันนี้สำหรับงานหลายๆ งานได้. job description จะถูกสร้างขึ้นโดยผ่านอินเตอร์เฟซที่ใช้อักขระ.

สำหรับข้อมูลเพิ่มเติม, โปรดดูหัวข้อ **Job Description** ในบทที่ 5 ของ V4R5 Work Management  (ประมาณ 2720 กิโลไบต์ หรือ 573 หน้า).

## งานแอ็คทีฟและ inactive

### งานที่แอ็คทีฟ:

งานที่แอ็คทีฟคืองานที่ถูกเรียกทำงานอยู่แต่ยังไม่เสร็จสิ้นการทำงาน. ต่อไปนี้เป็นคุณลักษณะบางประการของงานที่แอ็คทีฟ:

- มีโค้ดที่ทำงานอยู่
- มี call stack
- มีอีอบเจกต์ที่ถูกล็อก
- มีสถานะของงานที่แอ็คทีฟ เช่น:
  - Running
  - Waiting for (บางสิ่ง)

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับคุณสมบัติของงานที่แอ็คทีฟ, โปรดดูที่ “คุณสมบัติของงาน” ในหน้า 46.

หากต้องการทราบวิธีการจัดการงานแอ็คทีฟ, โปรดดูที่ “การจัดการงานและ thread” ในหน้า 7.

### งานที่ inactive:

งานที่ Inactive คืองานบน“คิวงาน” ในหน้า 56 ที่รอเรียกทำงาน, หรืองานที่เสร็จสิ้นการประมวลผล (ถูกยุติ) แต่กำลังรอไฟล์พรินเตอร์เอาต์พุต (มักเรียกว่าไฟล์สพูล) พิมพ์ออกมา.

## ประเภทของงาน

ระบบ iSeries จะประมวลผลงานหลากหลายประเภท. คุณสามารถเลือกประเภท ของงานอย่างใดอย่างหนึ่งเพื่อศึกษาเพิ่มเติมเกี่ยวกับงานประเภทนั้นๆ .

“งานเซิร์ฟเวอร์” ในหน้า 44 คืองานที่ถูกเซ็ทประเภทเซิร์ฟเวอร์ไว้โดยใช้ Change Job (QWTCHEGJB) API, และจะมีการแบ่งประเภทของ Server ด้วยประเภทงานอย่างใดอย่างหนึ่งต่อไปนี้:

“งานแบบ autostart” ในหน้า 37

งาน autostart จะเริ่มขึ้นโดยอัตโนมัติเมื่อระบบย่อยที่งาน autostart เชื่อมโยงอยู่สตาร์ทขึ้นมา.

“งานแบบแบตช์” ในหน้า 37

งานแบตช์เป็นกลุ่มการประมวลผลของภารกิจที่กำหนดไว้ล่วงหน้าซึ่งได้ ส่งไปยังระบบ.

“งานสื่อสาร” ในหน้า 38

งานสื่อสารเป็นงานแบ็คที่ถูกระบุขึ้นโดย request เริ่มต้นใช้งานโปรแกรมจากระบบรีโมต.

“งานโต้ตอบ” ในหน้า 38

งานแบบโต้ตอบต้องการอินพุตจากผู้ใช้ที่ sign-on เข้ามา และจากระบบ iSeries.

“งาน Prestart” ในหน้า 39

งาน prestart เป็นงานแบ็คที่สตาร์ทก่อนที่จะได้รับ work request เข้ามา.ต่อไปนี้เป็นงาน prestart ทั้งสองประเภท:

- การสื่อสารแบบ prestart - งานคืองานแบ็คประเภทสื่อสารที่สตาร์ทการทำงานก่อนที่ระบบรีโมตจะส่ง request เริ่มต้นใช้งานโปรแกรมเข้ามา.
- งานแบ็คแบบ prestart - งานคืองานแบ็คที่เริ่มต้นทำงานก่อนที่จะได้รับ request งาน.

“งาน reader และ writer” ในหน้า 39

งาน reader เป็นงานสพูลอินพุต, และงาน writer เป็นงานสพูลเอาต์พุต.

“งานระบบย่อย” ในหน้า 39

งานระบบย่อยจะทำการควบคุมระบบย่อยแฉีกทีฟ.

“งานระบบ” ในหน้า 40

งานระบบจะถูกสร้างขึ้นโดยระบบปฏิบัติการเพื่อควบคุมหรือสวอร์ชของระบบและปฏิบัติงานที่เป็นฟังก์ชันของระบบ.

**งานแบบ autostart:** งานแบบ autostart จะทำงานโดยอัตโนมัติเมื่อระบบย่อยที่เชื่อมโยง อยู่ทำงานขึ้น. งานเหล่านี้มักจะทำการเตรียมค่าเริ่มต้นที่สัมพันธ์กับระบบย่อยต่างๆกันให้. งานแบบ autostart ยัง สามารถทำงานที่วนซ้ำหรือเตรียมฟังก์ชันบริการแบบรวมศูนย์ให้กับงาน อื่นๆ ในระบบย่อยเดียวกันได้อีกด้วย.

งานของระบบย่อยจะใช้ข้อมูลจาก autostart job entry ใน subsystem description เมื่อมีการเริ่มทำงานขึ้น.

**หมายเหตุ:** งานแบบ autostart ทั้งหมดจะเริ่มทำงานเมื่อระบบย่อยทำงานเท่านั้น. ค่าที่กำหนดไว้ใน “Subsystem description” ในหน้า 59 จะไม่สามารถหยุดไม่ให้งานแบบ autostart ทำงานได้. แต่ถ้าค่าสูงสุด ในระบบย่อยเกินที่ได้ระบุไว้, งานอื่นๆ จะไม่สามารถทำงานได้. เมื่องานแบบ autostart เสร็จสิ้นลงซึ่งมีเพียงพอให้จำนวนของงาน ต่ำกว่าจำนวนสูงสุดของ activity level, งานอื่นๆ ในระบบย่อยจึง จะทำงานได้.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ autostart job และวิธีการเริ่มทำงาน, ropicดูที่ หัวข้อ Autostart Jobs (บทที่ 9) และ หัวข้อ

Autostart Job Entry (บทที่ 4) ในคู่มือ V4R5 Work Management  (ประมาณ 2720 KB หรือ 573 หน้า).

**งานแบบแบ็ค:** งานแบบแบ็คเป็นกลุ่มของการประมวลผลที่ได้กำหนดไว้ล่วงหน้าและถูกส่งไปยังระบบ. งานแบบแบ็ค จะทำงานอยู่ในส่วนแบ็คกราวนด์ของระบบ เพื่อให้ผู้ใช้งานที่ส่งงานเข้ามาสามารถทำงาน อื่นได้. เมื่องานดังกล่าวทำงานแล้ว จะไม่ต้องการการโต้ตอบในส่วนของผู้ใช้งานอีกเลย. ดังนั้น งานแบบแบ็คจึงเป็นงานที่มีระดับความสำคัญที่ต่ำ. งานแบบแบ็คหลายๆ ชิ้นจึงสามารถทำงานได้ในเวลาเดียวกัน.

ต่อไปนี้เป็นงานแบบแบ็คในประเภทต่างๆ:

### งานแบ็คซ์แบบธรรมดา

คนส่วนใหญ่จะคุ้นเคยกับงานแบบแบ็คซ์ประเภทธรรมดาที่ถูกส่งไปไว้ในคิวงาน. สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับช่วงชีวิตของงานแบบแบ็คซ์ประเภทธรรมดา ให้ดูหัวข้อ ช่วงชีวิตของงาน.

### งานแบ็คซ์แบบ immediate

งานแบ็คซ์แบบ immediate เป็นงานที่ถูกเริ่มขึ้นพร้อมกับแอ็คทริวิตีอื่นๆ ของงานตัวแม่. และงานจะรันอยู่ในระบบย่อยเดียวกันกับงานแม่. เนื่องจากงานชนิดนี้ใช้แอ็คทริวิตีชุดเดียวกันกับงานแม่และไม่ได้วิ่งผ่านคิวงาน, ดังนั้น จึงสามารถทำงานได้เร็วกว่างานที่ส่งผ่านคิวงาน.

### งานแบ็คซ์แบบ MRT

งานแบบแบ็คซ์ MRT คืองานแบบ multiple requester terminal (MRT). งานแบบ MRT เป็นงานของ S/36 Environment ที่ทำงานเหมือนเป็นเซิร์ฟเวอร์ ที่จะอนุญาตให้งาน S/36 Environment ตัวอื่นสามารถผนวกติดตามด้วยเพื่อที่จะรันโปรแกรม MRT ด้วยกัน.


### แบ็คซ์งานพิมพ์

แบ็คซ์งานพิมพ์จะติดตามดูไฟล์พริ้นเตอร์เอาต์พุต(หรือที่เรียกว่าสพูลไฟล์) ที่ถูกสร้างขึ้นโดยงานที่มีโปรแกรมเมอร์ใช้ในปัจจุบันที่แตกต่างไปจากโปรแกรมเมอร์ที่ใช้ในตอนทำงานถูกเรียกทำงาน.

สำหรับข้อมูลเพิ่มเติม, โปรดดูที่หัวข้อ การเริ่มงานแบ็คซ์ในบทที่ 8 ของคู่มือ V4R5 Work Management  (ประมาณ 2720 KB หรือ 573 หน้า).

งานสื่อสาร: งานสื่อสารจะเริ่มต้นทำงานเมื่อได้รับการร้องเริ่มต้นใช้งานโปรแกรมจากระบบรีโมต. แต่เพื่อให้การทำงานมีประสิทธิภาพ, แทนที่จะเริ่มต้นงาน การสื่อสารทุกครั้งที่ได้รับการร้องขอเริ่มต้นใช้งานโปรแกรม, คุณสามารถตั้งค่างาน “งาน Prestart” ในหน้า 39 เพื่อให้จัดการกับการร้องขอเริ่มต้นใช้งานโปรแกรมจากระบบรีโมตโดยเฉพาะ.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการร้องขอเริ่มต้นใช้งานโปรแกรม, โปรดดูบทที่ 3 ของคู่มือ ICF Programming .

สำหรับข้อมูลเพิ่มเติม, โปรดดู Communications Jobs ในบทที่ 10 ของหนังสือ V4R5 Work Management  (ประมาณ 2720 กิโลไบต์ หรือ 573 หน้า).

งานโต้ตอบ: งานโต้ตอบต้องการการสื่อสารสองทางอย่างต่อเนื่องระหว่างผู้ใช้และระบบ iSeries เพื่อทำงานกับภารกิจต่างๆ. งานโต้ตอบจะเริ่มต้นเมื่อ ผู้ใช้ได้ sign on เข้าสู่ระบบ. ระบบจะร้องขอข้อมูลการ sign-on. ถ้าระบบตอบรับการ sign-on request, ระบบจะสร้างงานโต้ตอบขึ้นมา. หลังจากนั้นระบบจะขอให้ผู้ใช้ส่ง request. หลังจากนั้นผู้ใช้ต้องใส่ request, และระบบจะตอบรับโดยการประมวลผล request นั้น. กระบวนการนี้จะทำซ้ำจนกระทั่งผู้ใช้จบงานโต้ตอบโดยการ sign off ระบบด้วยตัวเอง. หากงานโต้ตอบเป็นส่วนหนึ่งของกลุ่มงาน หรือคู่ของงาน, เมื่อนั้นงานโต้ตอบจะมีประเภทของงานอย่างใดอย่างหนึ่งต่อไปนี้:

### โต้ตอบ - แบบกลุ่ม

งานกลุ่มแบบโต้ตอบเป็นส่วนหนึ่งของกลุ่มงานที่เชื่อมโยงไว้กับอุปกรณ์แสดงผลหนึ่งเครื่อง. โต้ตอบ - System request งานโต้ตอบแบบ system request เป็นหนึ่งในคู่ของงานที่เชื่อมโยงเข้าไว้ด้วยกันโดยฟังก์ชัน system request.

**งาน Prestart:** งาน prestart จะสตาร์ทก่อนที่จะได้รับการร้องขอการทำงาน, ไม่ว่าจะเป็น ตอนทีระบบย่อยสตาร์ทขึ้น หรือที่เป็นผลมาจากคำสั่ง Start Prestart Jobs (STRPJ). งาน Prestart จะเริ่มทำงานตั้งแต่ prestart job entry (PJE) ใน subsystem description. prestart job entry จะระบุคุณสมบัติต่างๆ เช่นโปรแกรมใดจะรันในงาน prestart, โพรไฟล์ผู้ใช้ที่อยู่ภายใต้งาน prestart ที่เริ่มทำการรัน, “Job description” ในหน้า 35, คลาสที่ใช้ระบุคุณสมบัติรันไทม์ของงาน, และ “พูลหน่วยความจำ” ในหน้า 75 ที่งาน prestart รันอยู่.


งาน prestart สามารถเริ่มทำงานและ initialize ตัวมันเองได้ก่อนที่จะได้รับการร้องขอการทำงาน. สิ่งนี้จะช่วยลดจำนวนเวลาที่ต้องใช้ในการจัดการ การร้องขอ ต่างๆ. การร้องขอการทำงานทุกอัน ไม่จำเป็นต้องใช้งานใหม่. นอกจากนั้นแล้ว, งาน prestart ยังช่วยให้คุณสามารถ initialize เพียงครั้งเดียวแต่จัดการ การร้องขอ ต่างๆ ได้มากมาย เพื่อให้งานใหม่ไม่จำเป็นต้องใช้การร้องขอทุกครั้งไป. แอปพลิเคชันโคลเอนต์เซิร์ฟเวอร์ส่วนใหญ่จะใช้งาน prestart เพื่อจัดการ การร้องขอ ให้กับผู้ใช้โคลเอนต์. การมีงานที่พร้อมจะทำงานได้จะช่วยให้การทำงานมีประสิทธิภาพมากขึ้นในกรณีนี้ เพราะงาน prestart จะเริ่มการประมวลผล การร้องขอ ให้กับผู้ใช้ได้ทันที.

**หมายเหตุ:** ค่าที่กำหนดไว้ของ “Subsystem description” ในหน้า 59 สามารถป้องกันไม่ให้งาน prestart เริ่มทำงานได้. หากจำนวนงานสูงสุด ในระบบย่อยมีจำนวนมากเกิน, งาน prestart จะไม่สามารถเริ่มทำงานได้. แต่เมื่อมีจำนวนหนึ่งได้เสร็จสิ้นลงจนทำให้จำนวนงานที่รันอยู่ต่ำกว่าจำนวนงานสูงสุดในระบบย่อยแล้ว, งาน prestart ในระบบย่อยจึงจะสามารถทำงานได้.

งาน prestart มีอยู่ 2 ประเภทด้วยกัน. แต่ละประเภทจะจัดการการร้องขอประเภทต่างๆ. ก่อนที่งานจะรอสำหรับการร้องขอแรกนั้น, งานจะถูกแสดงว่าเป็น “Prestart” เท่านั้น, เพราะวาระบบจะยังไม่รู้ว่า การร้องขอประเภทใดที่งานนั้นจะทำการจัดการ. ต่อไปนี้ เป็นงาน prestart ทั้งสองประเภท:

### งานสื่อสาร prestart

งานสื่อสาร prestart เป็นงานแบ็คซ์ประเภทสื่อสารที่จะเริ่มทำงานก่อนทีระบบริโมตจะส่งโปรแกรมเริ่มทำงานที่ร้องขอ.

สำหรับข้อมูลเพิ่มเติม, โปรดดูที่หัวข้อ Prestart Jobs ในบทที่ 11 ของคู่มือ V4R5 Work Management  (ประมาณ 2720 KB หรือ 573 หน้า).

### งานแบ็คซ์ prestart

งานแบ็คซ์ prestart คืองานแบ็คซ์ที่เริ่มทำงานก่อนทีจะได้รับการร้องขอการทำงาน.

#### งาน reader และ writer: reader

งาน reader จะอ่าน กระแสงานแบ็คซ์จากฐานข้อมูลและไฟล์ดิสก์เก็ต, และนำเอางานไปใส่ไว้ในคิวงาน. งาน reader เป็นส่วนหนึ่งของอินพุตสพูล และเป็นโปรแกรมที่ IBM จัดส่งมาให้ตั้งแต่ต้น.

#### writer

งาน writer จะเขียนเร็กคอร์ดจากไฟล์พริ้นเตอร์เอาต์พุต (หรือเรียกว่าสพูลไฟล์) ไปยังพริ้นเตอร์. งาน writer เป็นโปรแกรมที่ IBM จัดส่งมาให้ตั้งแต่ต้น, และจะสตาร์ทขึ้นในส่วนสพูลระบบย่อย ที่ซึ่งงาน writer จะเลือกไฟล์จากเอาต์พุตคิวไปพิมพ์ต่อไป.

**งานระบบย่อย:** งานระบบย่อย (บางครั้งเรียกว่างานมอนิเตอร์ระบบย่อย) ถูกสร้างขึ้น โดยระบบปฏิบัติการเพื่อจัดการรีซอร์สและเริ่มต้นทำงาน, ควบคุมงาน, และจบ งาน. งานระบบย่อยจะช่วยควบคุมระบบย่อยที่แเอคทีฟ. งานระบบย่อยจำนวนมากจึงสามารถ รันบนระบบได้พร้อมๆ กัน.

สำหรับรายละเอียดเพิ่มเติม, กรุณาดูหัวข้อ “ระบบย่อย” ในหน้า 58.

งานระบบ: งานระบบจะถูกสร้างขึ้นโดยระบบปฏิบัติการเพื่อควบคุมรีซอร์สของระบบ และปฏิบัติงานที่เป็นฟังก์ชันของระบบ. งานระบบจะเริ่มรันเมื่อเซิร์ฟเวอร์ iSeries เริ่มทำงาน» หรือเมื่อดิสก์พูลอิสระถูก varied on. « งานเหล่านี้ จะปฏิบัติการหลายอย่าง ตั้งแต่การเริ่มงานระบบปฏิบัติการ, การเริ่มและจบงานระบบย่อย, และการกำหนดตารางการทำงานของงาน.

ต่อไปนี้เป็นงานระบบประเภทต่างๆ และฟังก์ชันของงานระบบ เหล่านี้:

#### งานเริ่มใช้งาน System: Scpf (start control program function)

งานตัวนี้เป็นงานศูนย์กลางเมื่อคุณเริ่มงานระบบ.» Scpf เริ่มชุดงาน Qsysarb, แต่ Qsysarb3 (ดู 40) เริ่มงานระบบอย่างอื่นส่วนใหญ่ (แต่ไม่ใช่ Qlus) « รวมทั้งทำให้ระบบอยู่ในสถานะที่ใช้งานได้. งานตัวนี้จะยังคงแฉีกที่พอย์หลังจากระบบถูกเรียกใช้งาน, เพื่อจัดเตรียมสถานะแวดล้อมสำหรับการรันฟังก์ชันที่มีความสำคัญต่ำและมีแนวโน้มจะรันเป็นเวลานาน. Scpf จะรันอยู่ในระหว่างกระบวนการ power down (Pwrdownsys), และเป็นงานตัวที่ใช้จบกระบวนการ ของเครื่องด้วย.

#### Qwbcleanup (job table cleanup)

งานตัวนี้จะถูกใช้ระหว่างการเริ่มต้นใช้งานระบบเพื่อยืนยันให้แน่ใจว่าโครงสร้างงานได้ถูกเตรียมไว้พร้อมใช้งาน. งานตัวนี้จะเสร็จสิ้น การประมวลผลก่อนที่การเริ่มต้นงานระบบจะสมบูรณ์, และยังสามารถรันต่อไปนี้หลังจากระบบเริ่มทำงานเสร็จแล้ว, ถ้าหากยังมีโครงสร้างงานอีกจำนวน มากที่จะต้องสะสางให้หมด. งานระบบตัวนี้จะจบลงเมื่อเสร็จสิ้นการประมวลผล.



#### Qlpsvr (ข้อตกลงการใช้ซอฟต์แวร์)

งานนี้จะเริ่มทำงานโดยอัตโนมัติระหว่างช่วง IPL หลังจากการตอบรับเงื่อนไขการใช้ซอฟต์แวร์. งานนี้จะสิ้นสุดลงเมื่อมีการยอมรับหรือปฏิเสธข้อตกลงทั้งหมด. «

#### System arbiter: Qsysarb (system arbiter)

system arbiter จะจัดเตรียมสถานะแวดล้อมสำหรับการรันฟังก์ชันที่มีความสำคัญสูง. และยังช่วยจัดการรีซอร์สระบบและช่วยติดตามสถานะของ ระบบให้อีกด้วย. system arbiter จะตอบสนองเหตุการณ์ต่างๆ ที่ทั้ง ระบบที่จะต้องมีการจัดการในทันที และเหตุการณ์ที่สามารถจัดการได้อย่างมีประสิทธิภาพมากขึ้นด้วยงานชิ้นเดี่ยว. Qsysarb,» Qtaparb (tape arbiter), « และ Qcmnarbxx (communications arbiters) มีหน้าที่ตอบสนองต่อการร้องขอบริการทางการสื่อสาร, การล็อกอุปกรณ์, สาย, คอนโทรลเลอร์, device configuration, และการจัดการรีซอร์สโดยทั่วไปของระบบ.

#### Qsysarb2 (system arbiter 2)

งานตัวนี้จะรับผิดชอบการจัดการเทปรีซอร์ส, จัดการ command analyzer space สำหรับการประมวลผลคำสั่ง และการประมวลผลอื่นๆ ที่ระบบสำหรับระบบปฏิบัติการ.

#### Qsysarb3 (system arbiter 3)

งานตัวนี้จะรับผิดชอบการสร้างและการดูแลรักษาโครงสร้างงานบน ระบบ. เมื่อใดก็ตามที่จำเป็นต้องใช้โครงสร้างงานชั่วคราวหรือถาวร ในการเริ่มต้นงาน, การร้องขอดังกล่าวจะถูกประมวลผลโดย Qsysarb3.» Qsysarb3 ยังทำหน้าที่เริ่มต้นและสิ้นสุดงานของระบบอีกหลายอย่าง. «

#### Qsysarb4 (system arbiter 4)

งานตัวนี้จะรับผิดชอบการเริ่มต้นใช้งานและจบระบบย่อย. ซึ่งจะรวม ถึงกระบวนการปิดเครื่อง (Pwrdownsys) ในระยะต้น.

#### Qsysarb5 (system arbiter 5)

งานตัวนี้จะรับผิดชอบการประมวลผลเหตุการณ์ต่างๆ ของเครื่อง. ซึ่งจะ รวมถึงการจัดการเหตุการณ์เพื่อสนับสนุนการใช้พลัง

งานสำรอง, continuous powered mainstore (CPM), system auxiliary storage pools (ASPs) และ storage threshold และ lock table limit. โดยปกติแล้ว, เหตุการณ์ของเครื่องจะถูกจัดการและข้อความ CPF ที่สัมพันธ์กันนั้นจะถูกส่งไปยัง Qsysopr และ Qhst.

#### งานสื่อสาร: Qlus (logical unit services)

Qlus จะจัดการเหตุการณ์ที่ดูแลอุปกรณ์ลอจิคัลยูนิต, ซึ่งรู้จักกันว่า อุปกรณ์สื่อสาร. Qlus ยังรับผิดชอบการจัดการจัดสรรอุปกรณ์ให้กับระบบย่อย สื่อสารให้ถูกต้องอีกด้วย.

#### Qcmnarbxx (communications arbiters)

communications arbiters พร้อมด้วย Qsysarb (system arbiter) และ Qtaparb (tape arbiter) จะรองรับงานด้านอุปกรณ์ทุกชนิด, โดยไม่จำกัดเฉพาะอุปกรณ์สื่อสาร. งานเหล่านี้รวมถึงการเชื่อมต่อระบบสื่อสาร, การตัดการติดต่อ, การล็อกอุปกรณ์, และการประมวลวิธีแก้ไขข้อผิดพลาด.

งาน communication arbiter, ที่ค่าระบบ (QCMNARB) เป็นตัวกำหนดจำนวนของงาน communications arbiter จะที่เกิดขึ้น. communications arbiter อย่างน้อย 3 ตัวจะถูกเรียกใช้งานบนระบบโปรเซสเซอร์เดี่ยว.

#### Qsyscomm1 (system communications)

งานตัวนี้จะจัดการการสื่อสารบางตัวและ activity อินพุต/เอาต์พุต (I/O).

#### Q400filsvr (remote file system communication)

งานตัวนี้จะใช้ทำการสื่อสารแบบ common programming interface (APPN หรือ APPC) สำหรับระบบรีโมตไฟล์.

#### งานฐานข้อมูล: Qdbfstccol (database file statistic collection)

งานตัวนี้จะสะสมสถิติไฟล์ไว้. สถิติเหล่านี้สำคัญต่อการกำหนดค่าแบบสอบถามของฐานข้อมูลเพื่อให้ได้ประสิทธิภาพการทำงานที่ดีที่สุด.

Qdbsrvxr (database cross-reference) และ Qdbx###xr สำหรับกลุ่มดิสก์พูลอิสระ ###



งานตัวนี้จะดูแลรักษา file level system cross-reference file ใน Qsys. ไฟล์เหล่านี้จะประกอบด้วยข้อมูลอ้างอิงข้ามเกี่ยวกับไฟล์ฐานข้อมูลและข้อมูล SQL ข้ามระบบ. ไฟล์เหล่านี้จะเริ่มต้นด้วยคำนำหน้า เป็น Qadb ในไลบรารี Qsys. ไฟล์พื้นฐานที่จะต้องได้รับการดูแล คือ Qdbxref, ซึ่งเป็นไฟล์อ้างอิงข้าม file cross-reference file. ไฟล์ตัวนี้จะประกอบด้วย เร็กคอร์ดของฐานข้อมูลฟิสิกัล, ฐานข้อมูลลอจิคัล, DDM และไฟล์ Alias บนระบบ. Qdbsrvxr จะทำงานขึ้นมาเมื่อมีการสร้าง, ลบ, เรียกคืน, เปลี่ยนชื่อ, หรือสภาวะความเป็นเจ้าของของไฟล์ได้ถูกเปลี่ยนไป.

Qdbsrvxr2 (database cross-reference 2) และ Qdbx###xr2 สำหรับกลุ่มดิสก์พูลอิสระ ###



งานตัวนี้จะดูแลรักษาไฟล์อ้างอิงข้ามแบบสองฟิลด์. Qadbifld ในไลบรารี Qsys จะเป็นไฟล์อ้างอิงข้ามแบบสองฟิลด์. Qadbkfld ในไลบรารี library Qsys จะเป็นไฟล์อ้างอิงข้ามแบบฟิลด์หลัก. Qdbsrvxr2 จะถูกเรียกทำงานเมื่อมีการสร้าง, ปรับเปลี่ยน หรือลบไฟล์.

Qdbsrv01 (เซิร์ฟเวอร์ฐานข้อมูล) และ Qdbs###v01 สำหรับกลุ่มดิสก์พูลอิสระ ###





งานตัวนี้เรียกได้ว่าเป็น database maintenance task dispatcher. จำนวนงานเซิร์ฟเวอร์ฐานข้อมูลบนระบบ คือ หนึ่งบวกด้วยจำนวนสองเท่า ของจำนวนโพรเซสเซอร์, หรือหนึ่งบวกด้วยจำนวนสองเท่าของจำนวน ASP อันใดอันหนึ่งที่มากกว่า. จำนวนเริ่มต้นต่ำสุด คือ ห้า. Qbsrv01 เป็น ระบบหลักที่มอบหมายงานให้กับงานอื่นต่อ. ปกติแล้ว, Qbsrv01 จะแอนด์ที่พื้นที่หลังจากมีการเรียกคืนโลบรารีที่ประกอบด้วยไฟล์ฐานข้อมูล. ฟังก์ชันของงานตัวนี้ คือ:

- การให้สัญญาณกับภารกิจ system-managed access path protection (SMAPP) Licensed Internal Code (LIC), ที่แอนด์เซสพาทตัวใหม่ถูกเรียกคืนมา. และ SMAPP จะระบุได้ว่าแอนด์เซสพาทเหล่านี้ต้องการให้มีการปกป้องหรือไม่.
- การเตรียมลิสต์ของแอนด์เซสพาทที่จำเป็นต้องการสร้างขึ้นมาใหม่ เนื่องจากแอนด์เซสพาทไม่ได้ถูกเรียกคืนเอาไว้.

ในส่วนครั้งแรกของงานเซิร์ฟเวอร์ฐานข้อมูลที่เหลืออยู่นั้น จะประมวลผล request ที่มีความสำคัญสูง, ส่วนที่มีความสำคัญจะประมวลผลที่ครั้งหลัง. ตั้งแต่ Qbsrv02 ถึง Qbsrv05 จะมีระดับความสำคัญสูง, ส่วน Qbsrv06 ถึง Qbsrv09 มีระดับความสำคัญต่ำ.

Qbsrvxx (เซิร์ฟเวอร์ฐานข้อมูล, ระดับความสำคัญสูง) >> และ Qbs###vxx สำหรับกลุ่มดิสก์พูลอิสระ ###



งานเหล่านี้จะทำเจอร์นัลและ commitment control maintenance ให้กับ ระบบและถูกจัดว่าเป็นงานที่มีการรันรวดเร็วและใช้เวลาสั้น.

Qbsrvxx (เซิร์ฟเวอร์ฐานข้อมูล, ระดับความสำคัญต่ำ) >> และ Qbs###vxx สำหรับกลุ่มดิสก์พูลอิสระ ###




งานเหล่านี้ดูแลรักษาแอนด์เซสพาทในไฟล์ข้อมูลของผู้ใช้. โดยทั่วไปแล้ว, งานเหล่านี้จะเป็นงาน inactive, แต่ในบางกรณี, ก็อาจขึ้นมาทำการสร้างแอนด์เซสพาทใหม่. เหตุผลบางประการที่งานเหล่านี้ อาจจะมีแอนด์ที่ขึ้นมา คือ:

- การกู้คืนไฟล์ฐานข้อมูลที่ไม่มีการเซฟแอนด์เซสพาทเอาไว้.
- การกู้คืนลอจิคัลไฟล์ลอจิคัลโดยไม่มีฟิสิคัลไฟล์มารองรับ.
- การยกเลิกคำสั่ง Rgzpfm ในขณะที่ทำการประมวลผล.
- ตรวจจับเกิดผิดปกติเนื่องจากพบความเสียหายในตรรกะ.
- เกิด activity หลังการติดตั้ง iSeries เพื่อให้การอ้างอิงข้ามหรือ activity การอัปเดต DB อื่นๆเสร็จสมบูรณ์.
- เกิดการตรวจสอบข้อจำกัด

### Qqtemp1 และ Qqtemp2 (database parallelism)

งานระบบ database parallelism ทำหน้าที่ประมวลผลฐานข้อมูลแบบอะซิงโครนัสให้กับ DB2<sup>(R)</sup> Multisystem. ถ้าผู้ใช้ทำการสอบถามไฟล์แบบกระจาย, งานหลายงานจะถูกเรียกใช้เพื่อให้ได้ผลลัพธ์ที่เร็วขึ้นโดยการแบ่งงานเป็นหลายงานในลักษณะขนาน.

งานอื่นๆ: Qalert (ข้อความแจ้งเตือน)

งานนี้เป็นส่วนสำคัญของกระบวนการแจ้งเตือน (สำหรับข้อมูลเกี่ยวกับการเตือน, โปรดดูที่คู่มือ Alerts Support ). ซึ่งจะพูดถึง activity ต่างๆ เช่น การประมวลผลข้อความเตือน ที่ได้รับมาจากระบบอื่น, การประมวลผลข้อความเตือนที่เกิดขึ้นที่ระบบโลคัล, และการรักษาขอบเขตของการควบคุม.



### Qdcpobjx (decompress system object)

งานเหล่านี้จะคลายการบีบอัดอ็อบเจ็กต์ระบบปฏิบัติการที่เพิ่งติดตั้งตามความต้องการ. งานเหล่านี้ต้องการใช้หน่วยความจำเพื่อจะทำงาน. ซึ่งหากว่า หน่วยความจำที่มีอยู่บนระบบของคุณต่ำกว่าระดับที่ควรจะเป็น, งานเหล่านี้จะหยุดทำงาน. จำนวนของงานคลายการบีบอัดของอ็อบเจ็กต์ระบบคือจำนวนของ หน่วยประมวลผลบวกด้วยหนึ่ง.

### Qfilesys1 (file system)

งานตัวนี้จะสนับสนุนการประมวลผลส่วนหลังของระบบไฟล์รวม. ซึ่งจะช่วยตรวจสอบความเปลี่ยนแปลงของไฟล์ที่ถูกเขียนไปยังหน่วยความจำ และทำ activity cleanup ไฟล์ระบบทั่วไปให้ด้วย.

### Qjobsd (job schedule)

งานตัวนี้จะควบคุมฟังก์ชันการกำหนดเวลาของระบบ. Qjobsd จะมอนิเตอร์ตัวจับเวลาของ job schedule entry และงานที่มีการ กำหนดเวลา.



### Qli###cl สำหรับกลุ่มดิสก์พูลอิสระ ### (library cleanup)

งานนี้ทำหน้าที่ล้างไลบรารีที่อยู่ในกลุ่มดิสก์พูลอิสระ. <<



### Qli###rp สำหรับกลุ่มดิสก์พูลอิสระ ###: (object cleanup)

งานนี้ทำหน้าที่ล้างอ็อบเจ็กต์ที่ถูกแทนที่ในกลุ่มดิสก์พูลอิสระ. <<

### Qlur (LU 6.2 resynchronization)

Qlur จะจัดการกระบวนการ two-phase commit resynchronization.

### Qpfradj (performance adjustment)

งานตัวนี้จะจัดการการเปลี่ยนแปลงของขนาดและ activity level ของพูล หน่วยความจำ. การร้องขอทั้งหมดที่จะเปลี่ยนแปลงพูลหน่วยความจำจะถูก ประมวลผลโดยงานตัวนี้. นอกจากนั้น, หากค่ากำหนดระบบ Qpfradj ถูกเซตให้เป็น 2 หรือ 3, งานตัวนี้จะเปลี่ยนขนาดและ activity levels ของพูลหน่วยความจำให้อย่างเต็มที่ เพื่อเพิ่มประสิทธิภาพการทำงานของระบบ.

Qsplmaint (ส่วนบำรุงรักษา system spool) >> และ Qspmn##### สำหรับกลุ่มดิสก์พูลอิสระ #####



งานนี้ทำหน้าที่ของฟังก์ชันการทำสพูลของระบบที่ประกอบด้วย:

- >> การทำความสะอาดสพูลไฟล์หลังจาก IPL หรือ varied on ระบบ.
- ย้ายสพูลไฟล์ที่ใช้ไม่ได้ของเอาต์พุตควิผู้ใช้ที่เสียหายในพูลหน่วยความจำสำรองของระบบ หรือในพูลหน่วยควาจำสำรองพื้นฐาน ไปที่เอาต์พุตควิ QSPRCLOUTQ ในไลบรารี QRCL.
- ล้างspool database member ที่มีข้อมูลและแอ็ททริบิวต์ของสพูลไฟล์ที่ถูกลบออกไป.
- ลบ spool database member ที่ไม่มีการใช้ซ้ำภายในเวลาที่กำหนดไว้ในค่า QRCLSPLSTG. <<



Qspff##### สำหรับกลุ่มดิสก์พูลอิสระ ##### (system spool PRTQ updater)

งานนี้จะปฏิบัติหน้าที่ของสพูลไฟล์ให้กับบางกลุ่มของดิสก์พูลอิสระ. <<



Qtaparb (อุปกรณ์เทป)

งานนี้ทำหน้าที่ที่เกี่ยวกับอุปกรณ์เทป ได้แก่ การล็อกอุปกรณ์และการกู้คืนจากข้อผิดพลาด. <<

**งานเซิร์ฟเวอร์:** งานเซิร์ฟเวอร์คืองานที่รันต่อเนื่องในแบ็กกราวนด์บนระบบ iSeries ที่กำลังรอกงาน. งานอาจมาจาก ฟังก์ชันเน็ตเวิร์ก, ฟังก์ชันระบบปฏิบัติการ, ในนามของผู้ใช้, ระบบอื่นภายในเน็ตเวิร์ก, หรือจากเซอริวิระบบทั่วไป, เช่นงานเซิร์ฟเวอร์การรวมกลุ่ม. โดยปกติงานเซิร์ฟเวอร์จะรันในหนึ่งในสามระบบย่อยพื้นฐาน “ระบบย่อยที่มีมาพร้อมกับระบบ” ในหน้า 70 - QSYSWRK, QSERVER, หรือ QUSRWRK. งานเซิร์ฟเวอร์มักจะเกี่ยวโยงกับฟังก์ชันต่างๆ เป็นส่วนใหญ่ อย่างเช่น HTTP, Lotus Notes<sup>(R)</sup>, และ TCP/IP. ระบบ iSeries มีแบบจำลองพื้นฐานสามแบบสำหรับงานเซิร์ฟเวอร์:

**แบบจำลองงานที่ถูก thread** - ในแบบจำลองงานที่ถูก thread งานเซิร์ฟเวอร์คืองานที่มี “thread” ในหน้า 53 จำนวนมาก. thread หนึ่งตัวทำหน้าที่เป็นผู้กระจายงานให้กับ thread อื่นๆ. ยกตัวอย่าง, เมื่อเซิร์ฟเวอร์รับ client request, initial thread จะอ่าน request และส่งผ่านไปยังอีก thread หนึ่งเพื่อดำเนินการตาม request ให้เสร็จสมบูรณ์. ด้วยแบบจำลองนี้, จำนวนงานบนระบบจะลดจำนวนลงเพราะงานถูกจัดการใน thread ต่างๆ มากกว่าต้องการงานจำนวนมาก. ตัวอย่างของงานเซิร์ฟเวอร์ที่ใช้แบบจำลองงานที่ถูก thread คือ Domino<sup>(TM)</sup>, เซิร์ฟเวอร์ HTTP, และ Websphere.

**แบบจำลองงานแบบ prestart** - ในแบบจำลองงานแบบ prestart โดยปกติแล้วจะมีงานหลักที่ทำหน้าที่เป็นผู้รับ request ที่เข้ามาในระบบ. งานนี้จะถูกเรียกว่างาน daemon. งาน daemon จะดำเนินการตาม request แล้วส่งผ่าน request ไปยังงานเซิร์ฟเวอร์ “งาน Prestart” ในหน้า 39 ที่ถูกต้อง. ด้วยแบบจำลองนี้, การใช้งานแบบ prestart สามารถลดจำนวนงานจำเป็นเพราะเมื่อดำเนินการตาม request เสร็จแล้ว งานเซิร์ฟเวอร์แบบ prestart จะรอการ request ครั้งต่อไป. งานเซิร์ฟเวอร์จะถูกนำมาใช้ใหม่. นอกจากนี้, จากมุมมองประสิทธิภาพการทำงาน, งานแบบ prestart จะทำการรันและรอเพื่อประมวลผล request อยู่แล้ว. ตัวอย่างงานเซิร์ฟเวอร์บางตัวอย่างที่ใช้แบบจำลองงาน prestart คือ SQL server, host server, และ simple mail transfer protocol (SMTP).

**หมายเหตุ:** สำหรับงานที่รันโค้ดผู้ใช้, ซึ่งโดยปกติจะเป็นงานที่ไม่ถูกนำกลับมาใช้ใหม่ (เหมือนกับงานเซิร์ฟเวอร์ส่วนใหญ่). นี่เป็นเพราะโค้ดผู้ใช้อาจเปลี่ยนข้อมูลใดๆ ในงาน.

**Multiple Listening Job Model** - ใน multiple listening job model, งานเซิร์ฟเวอร์หลายงานจะถูกเรียกใช้งาน. เมื่อมี request เข้ามา, งานที่ได้รับ request จะดำเนินการกับงาน, ขณะที่งานเซิร์ฟเวอร์ที่พร้อมใช้งานอื่นถัดไปจะรอให้ request ถัดไปเข้ามา. เมื่องานเซิร์ฟเวอร์ดำเนินการตาม request เสร็จสมบูรณ์, งานจะหยุดการเชื่อมต่อและหยุดการทำงาน. งานเซิร์ฟเวอร์ใหม่จะเริ่มทำงานและวงจรจะดำเนินการต่อไป. ด้วยแบบจำลองนี้, คุณไม่ต้องกังวลเรื่อง prestart job entry. อย่างไรก็ตาม, ในบางครั้ง อาจไม่สามารถกำหนดค่าระบบย่อยเฉพาะสำหรับสภาพแวดล้อมของคุณ เพราะแบบจำลองนี้จะรันที่ระบบย่อยดีฟอลต์. ข้อยกเว้นหนึ่งคือ file transfer protocol (FTP). ด้วย file transfer protocol คุณ สามารถกำหนดค่าระบบย่อยที่รันเซิร์ฟเวอร์ file transfer protocol ได้. ไม่มีฟังก์ชันใดสามารถทำให้ file transfer protocol รันในระบบย่อยหนึ่งและงานที่เหลือรันในระบบย่อยอื่น. นอกจากนี้, จากมุมมองประสิทธิภาพการทำงาน, ไม่สามารถหลีกเลี่ยงต้นทุนของการเริ่มต้นทำงานและการสิ้นสุดการทำงานได้เพราะเมื่องานถูกรันและสิ้นสุดการทำงาน งานอื่นจะเริ่มทำงาน. อย่างไรก็ตาม, เนื่องมาจากงานจะสิ้นสุดการทำงานเมื่อการเชื่อมต่อเสร็จสมบูรณ์ และงานต่อไปได้เริ่มขึ้น, งานใหม่จะถูกเรียกขึ้นมาและทำการรันเมื่อได้รับ request ต่อไป, ดังนั้นต้นทุนการเริ่มต้นใช้งานและการสิ้นสุด

สุดท้ายการทำงานไม่ควรมีผลต่อเวลาที่ใช้เชื่อมต่อกับเซิร์ฟเวอร์. ตัวอย่างบางตัวของงานเซิร์ฟเวอร์ที่ใช้ multiple listening job model คือ transfer protocol (FTP) และ line printer daemon (LPD).

“ข้อความ”อนุญาตให้ผู้ใช้ศึกษาสถานะของเซิร์ฟเวอร์และแก้ปัญหาที่อาจเกิดขึ้นได้. ข้อความมีบทบาทสำคัญในการจัดการงานเซิร์ฟเวอร์.

สำหรับข้อมูลโดยละเอียดเกี่ยวกับชื่องานของงานเซิร์ฟเวอร์ที่รันบนระบบ, ให้ดูที่ ตารางงานเซิร์ฟเวอร์. ตารางนี้จะแสดงให้คุณเห็นถึงระบบย่อยและชื่องานเพื่อที่ว่าคุณสามารถ “การเข้าถึงบันทึกการทำงานสำหรับงานแอคทีฟ, รวมถึงงานเซิร์ฟเวอร์” ในหน้า 31. ตารางจะแสดง job description ที่งานเซิร์ฟเวอร์แต่ละตัวใช้ด้วย. ตามค่าดีฟอลต์ งานเซิร์ฟเวอร์ส่วนใหญ่จะไม่สร้างบันทึกการทำงานเมื่องานสิ้นสุดการทำงาน (พารามิเตอร์ LOG ถูกตั้งค่าเป็น 4 0 \*NOLIST), ซึ่งหมายความว่าไม่มีการสร้างบันทึกการทำงาน. หากคุณต้องการให้สร้างบันทึกการใช้งานด้วยข้อความทั้งหมดที่ส่งไปยังบันทึกการใช้งาน, พารามิเตอร์ LOG ต้องระบุ 4 0 \*SECLVL.

หากคุณต้องการสร้างบันทึกการใช้งาน, ให้ปฏิบัติตามดังต่อไปนี้:

- หากคุณต้องการเปลี่ยนพารามิเตอร์บันทึกการใช้งานของงานที่ต้องการเมื่องานนั้นเป็นงานแอคทีฟ ให้ใช้ **Change Job (CHGJOB)** (จากอินเทอร์เฟซที่ใช้อักขระ) และเปลี่ยนพารามิเตอร์ LOG หรือไปยังไดอะล็อก Job log ใน Job Properties ใน iSeries Navigator.
- หากคุณต้องการเปลี่ยนพารามิเตอร์บันทึกการใช้งานให้มีระยะเวลายาวขึ้นหรือของงานหลายประเภท ให้ใช้ **CHGJOB** (จากอินเทอร์เฟซที่ใช้อักขระ) และเปลี่ยนพารามิเตอร์ LOG บน job description. เมื่อทำการเปลี่ยนแปลงแล้ว คุณต้องหยุดทำงานและรีสตาร์ทงานเซิร์ฟเวอร์ใหม่เพื่อใช้งานการเปลี่ยนแปลงนั้น. วิธีนี้จะสร้างบันทึกการใช้งานสำหรับงานทั้งหมดที่ใช้ job description. หากต้องการเปลี่ยนค่าบันทึกการใช้งานกลับไปเป็นตามเดิม คุณต้องทำตามขั้นตอนเหล่านี้อีกครั้งและตั้งค่าพารามิเตอร์บันทึกการใช้งานกลับไปเป็น 4 0 \*NOLIST.

**ข้อความ:** เนื่องจากเซิร์ฟเวอร์งานรันเป็นเวลาที่ไม่จำกัด ดังนั้นจึงจำเป็นที่คุณต้องเข้าใจข้อความที่ถูกส่งไปยัง QSYSOPR message queue และที่ส่งไปยังบันทึกการใช้งานเพื่อให้คุณสามารถแก้ไขปัญหาที่อาจเกิดขึ้นบนเซิร์ฟเวอร์ของคุณได้. ข้อความต่างๆ ประกอบด้วย ชื่องาน, ประเภทข้อความ, วันที่และเวลาที่ส่ง, การดำเนินการที่เกิดขึ้น, และการดำเนินการที่จำเป็นเพื่อแก้ไขปัญหา. คุณสามารถ “การเข้าถึงบันทึกการทำงานสำหรับงานแอคทีฟ, รวมถึงงานเซิร์ฟเวอร์” ในหน้า 31 ของงานเซิร์ฟเวอร์ได้ทาง iSeries Navigator.



**ข้อความแจ้งเตือน** - ข้อความเหล่านี้จะถูกส่งไปยัง QSYSOPR เนื่องจากข้อความเหล่านี้ต้องการการดำเนินการในทันที. ข้อความจะประกอบด้วยปัญหา, สาเหตุ, และการดำเนินการกู้คืนที่จำเป็น. ตัวอย่างเช่น, เซิร์ฟเวอร์ไม่สามารถเริ่มทำงานได้หรือเซิร์ฟเวอร์หยุดทำงานโดยไม่คาดคิด. บางเซิร์ฟเวอร์ส่งข้อความเตือนไปที่ QSYSOPR. ข้อความเหล่านี้มี Alert Option (ALROPT) ระบุไว้ในรายละเอียดข้อความ. คุณสามารถใช้การแจ้งเตือนเพื่อใช้ในการจัดการข้อความแจ้งเตือนจากส่วน

กลาง. สำหรับข้อมูลเพิ่มเติม, ให้ดูที่ การสนับสนุนการแจ้งเตือน .

**ข้อความในบันทึกการใช้งาน** - ข้อความเหล่านี้มีลักษณะเป็นข้อความวินิจฉัย, หมายความว่าข้อความเหล่านี้ไม่ใช่ข้อความที่วิกฤติแต่มีเพื่อเตือนผู้ใช้ว่าต้องมีการดำเนินการบางอย่าง. ข้อความเหล่านี้อาจถูกสร้างขึ้นโดยระบบเช่นเดียวกับที่ถูกสร้างขึ้นโดยผู้ใช้.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับข้อความ, ให้ดูที่คำอธิบาย iSeries Navigator แบบออนไลน์หรือให้ดู Messages.

## คุณสมบัติของงาน

คุณสมบัติของงานประกอบด้วยรายละเอียดของวิธีที่งานจะถูกประมวลผล. ซึ่งจะถูกกำหนดไว้ตั้งแต่เมื่องานได้ถูกสร้างขึ้นมา. คุณสมบัติบางประการ มาจาก “Job description” ในหน้า 35. เมื่องานถูกสร้างขึ้นมา, คุณสามารถดูและจัดการคุณสมบัติของงานได้ผ่านทาง Work Management ใน iSeries Navigator. หน้าคุณสมบัติของงานใน iSeries Navigator จะช่วยให้การทำงานของผู้ควบคุมระบบทำได้ง่ายขึ้นผ่านทางฟังก์ชันที่มีประสิทธิภาพและใช้งานง่ายในการจัดการงานต่างๆ. ผู้ใช้งานทั่วไปสามารถดูคุณสมบัติของงานได้, แต่ผู้ใช้ที่มี “สิทธิในการใช้งานที่เหมาะสม” ในหน้า 48 เท่านั้นที่สามารถทำการเปลี่ยนแปลงคุณสมบัติของงานได้. เช่นเดียวกัน, ผู้ใช้ที่ได้รับสิทธิจะจัดการงานผ่านทาง “การปฏิบัติงาน” ในหน้า 17. คุณสมบัติของ “งานระบบ” ในหน้า 40 จะไม่สามารถเปลี่ยนแปลงได้ใน iSeries Navigator. อย่างไรก็ตาม, ระดับความสำคัญในการรันของ  บาง  งานระบบสามารถเปลี่ยนแปลงโดยการใช้คำสั่ง change system job (CHGSYSJOB) command.

### การจัดการระดับความสำคัญของงาน

การดูหรือการเปลี่ยนแปลงคุณสมบัติของงาน, ให้ทำตามขั้นตอนต่อไปนี้:

1. จาก iSeries Navigator, ขยายที่ My Connections → *server-name* → Work Management → Active Jobs หรือ Server Jobs, ขึ้นอยู่กับชนิดของงานที่คุณต้องการ.
2. “การค้นหงานบนเซิร์ฟเวอร์ iSeries” ในหน้า 10 ที่มีคุณสมบัติที่ต้องการดูหรือเปลี่ยนแปลง.
3. คลิกขวาที่ Job Name.
4. เลือก Properties.

### แผ่นคุณสมบัติงาน

คุณสมบัติของงานทั่วไปจะอนุญาตให้คุณดู รายละเอียดทั่วไปเกี่ยวกับงานได้. โดยมีข้อมูลเกี่ยวกับชื่อของงานและ “ประเภทของงาน” ในหน้า 36, เวลาที่งานเข้าสู่ระบบ, เวลาที่งานเริ่ม, รายละเอียดเกี่ยวกับ “สถานะโดยละเอียด” ในหน้า 49, และข้อมูลอื่น..

คุณสมบัติของประสิทธิภาพของการทำงานจะอนุญาตให้คุณดูข้อมูลของ การทำงานพื้นฐานและทำการแก้ไขใดๆ ที่มีผลต่อประสิทธิภาพการทำงานของงานได้. คุณสามารถดูสถิติการทำงานที่ได้คำนวณเอาไว้ตลอดช่วงชีวิตของงาน, เช่น การใช้ CPU และ disk I/O. คุณยังสามารถแก้ไขค่าต่อไปนี้ที่มีผล ต่อการรันของงานได้:

- ระดับความสำคัญของการรัน
- การแบ่งเวลา
- ช่วงเวลารอดีฟอลด์

คุณยังสามารถดู, รีเฟรช, เซ็ตอัพการรีเฟรชอัตโนมัติ, หรือรีเซ็ตสถิติการทำงานที่ผ่านมาที่ถูกคำนวณไว้สำหรับ “งานแอคทีฟและ inactive” ในหน้า 36. สำหรับข้อมูลเพิ่มเติม, ให้ดู “สถิติการทำงานที่ผ่านมา” ในหน้า 48.

คุณสมบัติของ “คิวงาน” ในหน้า 56 จะมีไว้ให้กับงานที่อยู่ในคิวงานหรือได้เริ่มสตาร์ทมาจากคิวงาน. คุณสามารถเปลี่ยนแปลงรายละเอียดของงานที่อยู่ในคิวงานได้. คุณสามารถจัดการระดับความสำคัญของงานในคิวงาน, ดูวันที่และเวลาที่งานถูกนำมาไว้ในคิวงาน, และเปลี่ยนแปลงเวลาที่งานจะสามารถรันได้อีกด้วย.

คุณสมบัติของ Printer Output จะอนุญาตให้คุณดู และเปลี่ยนแปลงคุณสมบัติที่จะมีผลต่อการพิมพ์เอาต์พุตของงาน. คุณสามารถแสดงผลพรินเตอร์เอาต์พุตสำหรับงานได้โดยการใช้ปุ่มพรินเตอร์เอาต์พุต. คุณสามารถเลือกที่จะ “การดึงพรินเตอร์

เอาต์พุตออกมา” ในหน้า 48 จากงาน, เลือกเครื่องพิมพ์, เลือกเอาต์พุตคิวและไลบรารีของเอาต์พุตคิว, ระบุลำดับที่คุณต้องการให้ข้อมูลพิมพ์ออกมา (ตามความสำคัญ), ระบุส่วนท้ายของหน้ากระดาษ, และระบุว่าจะพิมพ์ขอบและส่วนหัวของข้อมูลหรือไม่.

คุณสมบัติของ Messages จะอนุญาตให้คุณกำหนดวิธีการจัดการ คำถาม และ ข้อความค้น ได้. ถ้าหากงานเป็นแบบ “งานแบบแบดซ์” ในหน้า 37, ระดับความรุนแรงของข้อความที่แสดงสาเหตุของการทำงานต้องจบลง จะปรากฏขึ้นมาให้คุณ.

คุณสมบัติ Job Log จะอนุญาตให้คุณดูและเปลี่ยนแปลงข้อมูลเกี่ยวกับบันทึกการใช้งานได้เช่นเดียวกับการแสดงผลบันทึกการใช้งาน. บันทึกการใช้งานประกอบด้วยข้อมูลโดยละเอียดเกี่ยวกับ request ที่ได้ป้อนไว้ให้กับงาน, เช่น คำสั่งในงาน, คำสั่งจาก CL program, และข้อความต่างๆ. หน้านี้จะอนุญาตให้คุณกำหนดได้ว่าต้องการเก็บข้อความในบันทึกการใช้งานหรือไม่, จะให้งานทำอะไรเมื่อบันทึกการใช้งานเต็ม, ข้อความประเภทใดที่จะเก็บเอาไว้, และจะสร้างบันทึกการใช้ที่พิมพ์ออกมา (พรินเตอร์เอาต์พุต) ให้กับงานที่จบลงอย่างปกติ รวมถึงระดับความละเอียดของข้อความในแต่ละข้อความอีกด้วย. สำหรับรายละเอียดเพิ่มเติม, ให้อูที่ “บันทึกการใช้งาน” ในหน้า 52.

คุณสมบัติของ Security จะอนุญาตให้คุณดูคุณสมบัติของ ความปลอดภัยของงานที่เป็นแอ็คทีฟอยู่ได้. สิ่งนี้รวมถึง job user identity, วิธีที่ใช้เซต job user identity (Set by), ผู้ใช้ปัจจุบันและชื่อของโปรไฟล์กลุ่มที่เกี่ยวข้อง อยู่กับ initial thread เริ่มต้นของงาน (กลุ่ม).

คุณสมบัติ >> Date/Time ให้ข้อมูลเกี่ยวกับวันและเวลาของงาน. คุณสามารถแสดงค่าของตัวแบ่งวันและเวลา. นอกจากนี้, คุณสามารถแสดงข้อมูลเกี่ยวกับเขตเวลา และวัน เวลาโลคัลของงาน. <<

คุณสมบัติ International ช่วยคุณตรวจสอบแก้ไขคุณสมบัติด้าน >> ภาษา ประเทศ/ภูมิภาค ของงาน. รวมทั้งรูปแบบการใช้ทศนิยม. นอกจากนี้, ค่าดังกล่าวยังกำหนด ภาษา, ประเทศ/ภูมิภาค, ตัวควบคุม character identifier, และการเรียงลำดับให้กับงาน. << นอกจากนี้ยังแสดงให้เห็นว่างานนั้นสามารถรองรับชุดอักขระแบบสองไบต์ (DBCS) ได้หรือไม่.

คุณสมบัติของ Threads จะอนุญาตให้คุณดูรายละเอียดที่เกี่ยวข้องกับ thread ของงานที่แอ็คทีฟอยู่หรืออยู่ใน คิวงาน. คุณสามารถแสดงผล thread ของงานโดยการใช้ปุ่ม Threads. หน้าดังกล่าวจะประกอบด้วยรายละเอียดเช่นว่า งานสามารถ รันโดยใช้ multiple user thread ได้หรือไม่, จำนวนของ thread แอ็คทีฟในงาน, และจำนวนของ user และ system thread สูงสุด ที่งานสามารถรันได้ในเวลาหนึ่ง.

คุณสมบัติของ Server จะอนุญาตให้คุณดูข้อมูลเกี่ยวกับงานเซิร์ฟเวอร์. สำหรับงานเซิร์ฟเวอร์แต่ละตัว, คุณสามารถดู ชนิดของเซิร์ฟเวอร์, job user identity, และ client IP address ถ้ามี. client IP address เป็นแอดเดรสของผู้ใช้ที่เซิร์ฟเวอร์ดังกล่าวให้บริการอยู่.

คุณสมบัติ >> Resources อนุญาตให้คุณตรวจสอบพูลหน่วยความจำของงาน, กลุ่มของดิสก์พูล, และข้อมูลของหน่วยความจำและหน่วยประมวลผลข้างเคียง. คุณสามารถแสดงได้ว่างานนั้นถูกจัดกลุ่มโดยมี initial thread หรือระดับที่ใกล้เคียงหรือไม่. นอกจากนี้, คุณยังสามารถตรวจสอบได้ว่างานนั้นถูกจัดกลุ่มเข้ากับงานอื่นหรือไม่. <<

คุณสมบัติ Other อนุญาตให้คุณดูและทำการเปลี่ยนแปลงคุณสมบัติที่เกี่ยวข้องกับ accounting code, การกำหนดสวิตช์, และการให้เลือกว่าจะทำให้การเชื่อมต่อ DDM แอ็คทีฟหรือไม่. คุณสามารถตรวจสอบว่างานนั้นทำงานในสภาพแวดล้อมของ System/36 หรือไม่.

สำหรับข้อมูลเพิ่มเติม, ให้อูที่คำอธิบาย iSeries Navigator.

สิทธิในการใช้งานที่เหมาะสม: หากต้องการทำการเปลี่ยนแปลงคุณสมบัติของงาน คุณต้องมี “Job control special authority (\*JOBCTL)”(\*JOBCTL) หรือโปรไฟล์ผู้ใช้ของคุณต้องตรงกับ job user identity ของงานที่ถูกเปลี่ยนแปลง.

มีคุณสมบัติไม่มีประเภทที่จำเป็นต้องมีสิทธิใช้งานพิเศษ \*JOBCTL ในการทำการเปลี่ยนแปลง. คุณสมบัติเหล่านี้คือ:

- ดีฟอลต์ช่วงเวลารอ
- ระดับความสำคัญของการรัน
- ช่วงเวลาที่ระบบแบ่งเวลาประมวลผล

หมายเหตุ: หากคุณวางแผนที่จะทำการเปลี่ยนแปลงโค้ดบัญชีของงาน, คุณต้องมีสิทธิ \*USE สำหรับคำสั่ง Change Accounting Code (CHGACGCDE) นอกเหนือจากสิทธิพิเศษแบบ \*JOBCTL หรือโปรไฟล์ผู้ใช้ที่ตรงกับ job user identity ของงาน.

สำหรับคุณสมบัติงานใดๆ ที่อ้างถึงอ็อบเจกต์ OS/400, เช่นคิวงาน, เอาต์พุตคิว, และตารางเรียงลำดับ, ผู้ใช้ต้องมีสิทธิใช้งานที่เหมาะสมต่ออ็อบเจกต์นั้น. สำหรับข้อมูลเพิ่มเติมเกี่ยวกับสิทธิการใช้งานใน iSeries, ดูที่ภาคผนวก D ในหัวข้อสิทธิการใช้งาน

งานที่จำเป็นสำหรับอ็อบเจกต์ที่ถูกใช้งานโดยคำสั่งใน คู่มือด้านความปลอดภัยสำหรับ iSeries 

**Job control special authority (\*JOBCTL):** Job control special authority (\*JOBCTL) จะให้คุณระบุ, ปลด, เปลี่ยน, และยกเลิกงานอื่นของผู้ใช้, เปลี่ยนแอตทริบิวต์การรันของงาน, เช่นอุปกรณ์เครื่องพิมพ์สำหรับงาน, หยุดการทำงานของระบบย่อยและดำเนินการ initial program load (IPL). คุณต้องมีสิทธิการบริการพิเศษ \*JOBCTL เพื่อเปลี่ยน ระดับความสำคัญ (RUNPTY) ของงานคุณที่ถูกรัน. การเปลี่ยนแปลงระดับความสำคัญของเอาต์พุตและระดับความสำคัญของงานจะถูกจำกัดโดย priority limit (PTYLMT) ในโปรไฟล์ของผู้ใช้ที่ทำการเปลี่ยนแปลง.

ผู้ใช้ที่มีสิทธิพิเศษ \*JOBCTL สามารถเปลี่ยนระดับความสำคัญของงานและการพิมพ์, สามารถจบการทำงานก่อนที่งานจะเสร็จสิ้น, หรือลบเอาต์พุตก่อนที่จะพิมพ์ออกมา. สิทธิพิเศษ \*JOBCTL สามารถทำให้ผู้ใช้แอสซเข้าสู่พริเตอร์เอาต์พุตที่เป็นความลับ, หากเอาต์พุตคิวถูกระบุว่าเป็น OPRCTL(\*YES). ผู้ใช้ที่ใช้งานสิทธิพิเศษ \*JOBCTL ผิดจะก่อให้เกิดผลเสียต่องานแต่ละชิ้นและประสิทธิภาพการทำงานของระบบโดยรวม.

การดึงพริเตอร์เอาต์พุตออกมา: ในรีลิสก่อน V5R2, พริเตอร์เอาต์พุตจะติดอยู่กับงานจนกระทั่งถูกลบทิ้งไปไม่ว่าจะเป็นเพราะถูกส่งไปยังพริเตอร์หรือถูกลบโดยผู้ใช้อย่างชัดเจน.

คุณมีอ็อปชันในการดึงพริเตอร์เอาต์พุตออกจากงานเมื่อสิ้นสุดการทำงาน. พริเตอร์เอาต์พุตที่ถูกดึงออกจากงานจะไม่ถูกลบทิ้งออกจากระบบ แต่จะอยู่ในเอาต์พุตคิว. วิธีนี้จะทำให้งานออกจากระบบได้, ซึ่งจะทำให้โครงสร้างงานที่จะถูกใช้โดยงานอื่นมีเนื้อที่ว่าง.

หมายเหตุ: หากคุณเลือกที่จะดึงพริเตอร์เอาต์พุตออกจากงาน, คุณไม่สามารถตรวจพริเตอร์เอาต์พุตได้ผ่านการสำรวจงาน. คุณจะต้องดูที่เอาต์พุตคิวที่มีเอาต์พุตอยู่เพื่อดูพริเตอร์เอาต์พุต.

สถิติการทำงานที่ผ่านมา: หน้าสถิติการทำงานที่ผ่านมาจะให้คุณได้เห็นสถิติของการทำงานของงานแอคทีฟ, ซึ่งได้คำนวณไว้ตลอดช่วงของเวลาที่ผ่านมา. สิ่งนี้มีความจำเป็นต่อการมอนิเตอร์ดูงานและการตรวจจับปัญหาที่อาจเกิดขึ้น. สถิติเหล่านี้รวมถึง CPU, disk I/O, page fault rate, average response time, และ interactive transactions.



หมายเหตุ: สถิติการทำงานที่ผ่านมาสำหรับ thread ไม่รวมถึงเวลาการตอบสนองเฉลี่ยและ transaction แบบโต้ตอบ.

คุณสามารถเปลี่ยนอ็อพชันการดูสถิติเหล่านี้ได้โดยการเลือกปุ่มใดปุ่มหนึ่งต่อไปนี้จากหน้า **Elapsed performance statistics**:

- **Refresh Now**  
จะรีเฟรชสถิติการทำงานที่ผ่านมาและขยายช่วงเวลาสถิติ เหล่านี้ถูกคำนวณออกไป.
- **Timed Refresh**  
จะให้คุณตั้งค่าการรีเฟรชสถิติการทำงานที่ผ่านมาโดยอัตโนมัติ. ซึ่งสามารถนำมาใช้มอนิเตอร์ข้อมูลการทำงานของงานได้.
- **Reset Statistics**  
จะล้างสถิติการทำงานที่ผ่านมาและจะรีเซ็ตช่วงเวลาที่มีการ คำนวณสถิติออกด้วย.

สถานะโดยละเอียด: สามารถดูสถานะปัจจุบันของงานในหน้า ทั่วไป ใน “คุณสมบัติของงาน” ในหน้า 46, ภายใต้สถานะโดยละเอียด. ตัวอย่างของสถานะโดยละเอียดมีดังนี้:

#### ถูกตั้งเวลาให้รันเมื่อ

งานจะยังคงรออยู่ในคิวงานจนกว่าจะถึงวันและเวลาที่ได้อัปเดตเวลาเอาไว้ให้รันได้. ณ เวลาในวันที่ได้ตั้งไว้, งานจะพร้อมให้ถูกเลือกขึ้นมาจากคิวงานเพื่อทำงานต่อไป.

สถานะโดยละเอียดจะแสดงค่าสถานะเชื่อมโยง (สถานะ - x), ซึ่งจะบอกรายละเอียดเพิ่มเติมเกี่ยวกับสถานะปัจจุบันของงาน. ตัวอย่างของสถานะโดยละเอียดบวกค่าสถานะเชื่อมโยงคือ: *จบการทำงาน - เกินขอบเขตการใช้ CPU* *จบการทำงาน* อ้างอิงถึงสถานะของงาน (งานได้จบการทำงานแล้ว), และ *เกินขอบเขต CPU* เป็นการอธิบายว่าทำไมงานจึงมีสถานะเช่นนั้น (จบการทำงาน).

สถานะโดยละเอียดอาจมีการแสดงค่าสถานะเชื่อมโยง [สถานะ - x (x)] เพื่อบอกถึงสถานะปัจจุบันของงาน. ตัวอย่างเช่นงานที่กำลังจะจบการทำงานอาจมีสถานะดังต่อไปนี้: *กำลังจบการทำงาน - เกินขอบเขตการใช้ CPU (รอล็อก)* งานที่อยู่ระหว่างกระบวนการกำลังจะจบการทำงาน (Ending) เพราะเกินขอบเขตการใช้ CPU (CPU limit exceeded), และงานกำลังรอการล็อก (Waiting for lock) ในระหว่างกระบวนการจบการทำงาน.

ถ้าหากงานไม่จบลงตามเวลาที่กำหนดไว้, ข้อมูลตัวนี้จะช่วยในการวิเคราะห์ปัญหาที่เกิดขึ้น.

ค่าสถานะอาจมีข้อมูลเพิ่มเติมเกี่ยวกับหน้าคุณสมบัติ. ตัวอย่าง, สถานะที่กำลังรอการล็อก, บนหน้า properties, จะแสดงว่าอ็อบเจ็กต์อะไรเชื่อมโยงกับ request การล็อก.

**สิ้นสุดการทำงาน:** การสิ้นสุดงานสามารถทำได้สองแบบคือ แบบ controlled และแบบ immediate. การเลือกแบบ **Controlled** จะเป็นวิธีที่ดีกว่าเพราะจะเป็นการอนุญาตให้โปรแกรมที่รันอยู่ในงานทำการ cleanup เมื่อจบงานและทำการจบงานอย่างเรียบร้อย. >> ส่วนแบบ **immediate** จะหยุดการทำงานทันทีหลังจากครบกำหนดเวลา. ขอแนะนำให้ทำการจบงานอย่างทันทีเฉพาะในกรณีที่มีอ็อพชันแบบควบคุมไม่สามารถทำงานได้เท่านั้น. <<

งานสามารถตรวจสอบสถานะการจบของงานได้ผ่านทาง Job API เช่น Retrieve Job Information (QUSRJOB1) API. เมื่อมีการเลือกวิธีจบงานแบบ controlled, แอ็พพลิเคชันที่ต้องทำการ cleanup ในการจบงานจะตรวจหาวิธีการจบแบบควบคุม. วิธีหนึ่งที่แอ็พพลิเคชันสามารถทำได้ก็คือ การดู asynchronous signal SIGTERM.

» เมื่องานที่กำลังจะมีโปรซีเดอร์การจัดการสัญญาณสำหรับสัญญาณ asynchronous SIGTERM อยู่, จะมีการสร้างสัญญาณ SIGTERM สำหรับงานขึ้น. เมื่อการควบคุมอยู่ที่โปรซีเดอร์การจัดการสัญญาณสำหรับ SIGTERM, โปรซีเดอร์จะทำวิธีต่างๆเพื่อหลีกเลี่ยงผลที่ไม่ต้องการเช่นป้องกันไม่ให้เกิดการอัปเดตข้อมูลของแอปพลิเคชันเพียงบางส่วน. แต่ถ้าตัวจัดการสัญญาณ SIGTERM ไม่สามารถสิ้นสุดได้ภายในเวลาที่กำหนด, ระบบจะสิ้นสุดงานทันที.

ถ้าการสิ้นสุดงานเป็นแบบ immediate, เวลาสูงสุดสำหรับตัวจัดการสัญญาณจะถูกกำหนดไว้ที่ค่าระบบ Maximum time for immediate end หรือ(QENDJOBLMT). ค่าการจำกัดเวลานี้ใช้สำหรับการสิ้นสุดงานเดียว, หรืองานทั้งหมดในระบบย่อยเดียว, หรืองานทั้งหมดในทุกระบบย่อย. หลังจากการ request ไปสองนาที, ผู้ควบคุมระบบสามารถใช้คำสั่ง End Job (ENDJOB) ด้วย OPTION(\*IMMED) เพื่อยับยั้งค่า QENDJOBLMT และสิ้นสุดแต่ละงานได้ทันที. ควรใช้คำสั่งนี้เฉพาะกรณีที่งานไม่สามารถทำการ cleanup ได้เพราะเกิดการ lock หรือติด wait conditions.

เพื่อให้มีเวลาเพียงพอสำหรับการ cleanup และ end-of-job ในการประมวลผล, คุณอาจต้องปรับค่าระบบ Maximum time for immediate shutdown (QPWRDWNLMT) ในส่วนการ Restart. ถ้าคุณตั้งค่าระบบ Maximum time for immediate end (QENDJOBLMT) สูงกว่าค่า Maximum time for immediate shutdown, จะมีข้อความเตือนออกมา. เมื่อมีการปิดเครื่อง, งานทั้งหมดต้องสิ้นสุดภายในเวลาไม่เกินที่กำหนดไว้ที่ค่า Maximum time for immediate shutdown เพื่อให้การปิดเครื่องเป็นไปอย่างสมบูรณ์. «

สำหรับขั้นตอนโดยละเอียดเกี่ยวกับวิธีการจบงาน, ให้อ่านที่ “การสิ้นสุดการทำงาน” ในหน้า 16.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการจบงาน และตรวจหาวิธีการจบแบบควบคุม, โปรดดูที่ การจบงาน ในบทที่ 5 ของคู่มือ

V4R5 Work Management  (ประมาณ 2720 กิโลไบต์ หรือ 573 หน้า).

รายละเอียด: การปฏิบัติงานแอ็คทีฟ: เมนู Details ในโพลเดอร์ Work Management จะสามารถแอ็กเซสไปยังรีซอร์สต่อไปนี้ที่ถูกใช้โดยงานหรือ thread ขึ้นต้นของงาน:

### Call stack

call stack ของงานจะแสดงขึ้นมา. call stack เป็นโปรแกรมและโปรซีเดอร์ที่กำลังถูกใช้งานอยู่. การดู call stack จะมีช่วยในการค้นหาว่างานกำลังรันโปรแกรมอะไรอยู่และงานกำลังทำอะไรถึงไหนแล้ว.

### รายชื่อไลบรารี

รายชื่อไลบรารีของงานหรือ thread ที่เลือกไว้จะแสดงขึ้นมา. รายชื่อ ไลบรารีจะเป็นลิสต์ของไลบรารีระบบและผู้ใช้ที่ถูกสร้างขึ้นมาเพื่อค้นหาและเป็นลำดับที่จะถูกค้นหา. ไลบรารี คือ สถานที่เก็บอ็อบเจกต์ ซึ่งอ็อบเจกต์ทั้งหมดบนเซิร์ฟเวอร์ iSeries จะต้องการการอ้างอิงที่ประกอบด้วยชื่ออ็อบเจกต์และไลบรารี. ดังนั้น จึงเป็นสิ่งสำคัญที่จะต้องมียรายชื่อ ไลบรารีที่ถูกสร้างไว้อย่างถูกต้อง เพราะอ็อบเจกต์จะถูกค้นหาได้โดยการเข้าไปค้นในไลบรารีเท่านั้น. หากรายชื่อไลบรารีถูกสร้างขึ้นมาโดยไม่สมบูรณ์, งานจะค้นหาอ็อบเจกต์ไม่พบ หรืออาจไปค้นหาอ็อบเจกต์ในไลบรารีที่ไม่ถูกต้อง. IBM ได้จัดเตรียมไลบรารีบางตัวมาให้ (ชื่อไลบรารีที่ขึ้นต้นด้วย Q), แต่คุณต้องสามารถสร้างของคุณขึ้นมาเองด้วย. คุณสามารถทำงานกับคุณสมบัติของไลบรารีได้โดยการเลือกไลบรารีจากไดอะล็อกแล้วคลิกขวา.

### อ็อบเจกต์ที่ถูกล๊อค

แสดงรายการของ “อ็อบเจกต์ที่ถูกล๊อค” ในหน้า 51 และอ็อบเจกต์ที่งานหรือ thread กำลังรอล๊อคอยู่. ซึ่งสิ่งนี้จะช่วยให้คุณเห็นว่าอ็อบเจกต์ใดที่งานกำลังใช้อยู่ พร้อมอ็อบเจกต์ที่งานกำลังพยายามเข้าใช้ด้วย.



## เปิดไฟล์

➤ คุณสามารถตรวจสอบดู library objects หรือ file system objects ของงานนั้น. ซึ่งจะเป็นประโยชน์ในการดีบั๊กและตรวจสอบสถานะของงาน.

### Library Objects

แสดงรายการของ library objects ที่สัมพันธ์กับงานที่กำหนด.

### File System Objects

แสดงรายการ IFS objects ทั้งหมดในระบบไฟล์ใดๆ, รวมถึง QSYS.LIB, ที่สัมพันธ์กับงานที่กำหนด. <<

## thread

รายการของ “thread” ในหน้า 53 ที่รันภายในงาน. thread เริ่มต้น, โดยปกติ, ถูกจัดอันดับไว้บนด้านบนของหน้าต่าง. thread คือชิ้นงานอิสระที่ช่วยให้งานสามารถประมวลผลในแต่ละครั้งได้มากกว่าหนึ่งรายการ.

## Transactions

รายการของ transaction เชื่อมโยงกับงาน. transaction คือ logical unit of work บนระบบ iSeries. transaction มักจะถูกอ้างถึงเมื่อเกี่ยวข้องกับการดำเนินการของฐานข้อมูล. สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับ Transaction, ให้ดูคำอธิบายของ iSeries Navigator หรือไปที่ Transactions.

## สถิติการทำงานที่ผ่านมา

ลิสต์ของ “สถิติการทำงานที่ผ่านมา” ในหน้า 48 ที่ถูกคำนวณตลอดช่วงเวลาที่ผ่านมามีถูกแสดงขึ้น. ข้อมูลนี้จะเป็นประโยชน์สำหรับการมอนิเตอร์งานและยังช่วยในการวิเคราะห์ปัญหาให้อีกด้วย.

## ข้อความ SQL สุดท้าย

ตัวเลือกข้อความ SQL ล่าสุดจะแสดงข้อความ SQL ล่าสุดที่รันในงาน. คำสั่ง SQL นี้จะแสดงอยู่ใน สคริปต์การรัน SQL. จากสคริปต์การรัน SQL, คุณสามารถรันคำสั่งอีกครั้งได้, แก้ไขและรันคำสั่ง, หรือบันทึกคำสั่งไว้กับไฟล์ฐานข้อมูล หรือไฟล์ PC.

**อ็อบเจกต์ที่ถูกล็อก:** งาน และ thread ใช้อ็อบเจกต์เพื่อประมวลผลงาน. เนื่องจากมีงานมากกว่าหนึ่งชิ้นที่ประมวลผลในหนึ่งครั้ง, จึงมีการล็อกอ็อบเจกต์เพื่อรักษา data integrity ไว้. อ็อบเจกต์ที่ถูกล็อก คืออ็อบเจกต์ระบบที่ถูกใช้งานโดยงานและ thread เพื่อประมวลผลงาน. เมื่องานหรือ thread เสร็จสิ้นการรัน, อ็อบเจกต์จะถูกปลดล็อกและพร้อมใช้งานเพื่อประมวลผลงานอื่นๆ. การล็อกอ็อบเจกต์จะอนุญาตให้ผู้ใช้เพียงรายเดียวใช้งานอ็อบเจกต์ในหนึ่งครั้ง, ทั้งนี้ขึ้นอยู่กับประเภท request ล็อกที่ใช้. เช่น, ถ้าผู้ใช้สองคนหรือมากกว่านั้นพยายามที่จะเปลี่ยนอ็อบเจกต์ในเวลาเดียวกัน, การเปลี่ยนแปลงที่เกิดขึ้นกับอ็อบเจกต์โดยผู้ใช้คนที่สองจะถูกล็อกเอาไว้จนกว่าผู้ใช้คนแรกจะเสร็จสิ้นการอัปเดตอ็อบเจกต์เสียก่อน ด้วยการใช้งาน lock holder, ผู้ใช้สามารถดูว่ารายการใดมีการล็อกหรือกำลังรอการล็อกอ็อบเจกต์.

**ขอบเขต** ระบุว่าการล็อกเกี่ยวข้องกับงาน, thread, หรือ lock space. ขอบเขตมักจะกำหนดว่าจะมีการล็อกอยู่นานเท่าไรและประเภทการร้องล็อกและกฎความขัดแย้งใดที่อ็อบเจกต์มี.

ประเภท request ล็อก มีระดับการแอ็กเซสต่างๆ กันที่งาน, thread หรือ lock space อาจมีสำหรับอ็อบเจ็กต์ที่ถูกล็อก. ตัวอย่างเช่น, การล็อกเฉพาะ, ไม่ต้องใช้ประเภทการล็อกแบบอ่านหากอ็อบเจ็กต์ถูกเปลี่ยนหรือลบทั้งบนระบบ. การร้องขอล็อกประเภทนี้จะไม่อนุญาตให้ผู้อื่นอ่านอ็อบเจ็กต์, และอนุญาตให้ผู้อื่นอ่านอ็อบเจ็กต์.

ประเภท request ล็อกอื่นๆ คือ:

- **Exclusive - No read**

อ็อบเจ็กต์จะถูกสงวนไว้เพื่อการใช้งานเฉพาะ. อย่างไรก็ตาม, หากอ็อบเจ็กต์ถูกล็อกโดยการร้องขอล็อกประเภทใดก็ตาม, คุณไม่สามารถได้สิทธิ์เฉพาะในการใช้งานอ็อบเจ็กต์นั้น. สถานะการล็อกนี้เหมาะสำหรับเมื่อผู้ใช้ไม่ต้องการให้ผู้ใช้รายอื่นแอ็กเซสเข้าสู่อ็อบเจ็กต์จนกว่าฟังก์ชันจะทำงานเสร็จสมบูรณ์.

- **Exclusive-Read**

สามารถใช้งานอ็อบเจ็กต์ร่วมกับประเภทการร้องขอล็อกแบบ shared-read. การล็อกนี้เหมาะสำหรับเมื่อผู้ใช้ต้องการป้องกันผู้ใช้รายอื่นไม่ให้ดำเนินการอย่างอื่นนอกเหนือจากการอ่าน.

- **Shared-Update**

สามารถใช้งานอ็อบเจ็กต์ร่วมกับประเภทการร้องขอล็อกแบบ shared-read หรือ shared-update. นั่นคือ, ผู้ใช้รายอื่นสามารถ request สถานะการล็อกแบบ shared-read หรือ shared-update ให้กับอ็อบเจ็กต์เดียวกันได้. สถานะการล็อกนี้เหมาะสำหรับเมื่อผู้ใช้ตั้งใจจะเปลี่ยนแปลงอ็อบเจ็กต์แต่ต้องการให้ผู้ใช้รายอื่นอ่านหรือเปลี่ยนแปลงอ็อบเจ็กต์เดียวกัน.

- **Shared-No update**

สามารถใช้งานอ็อบเจ็กต์ร่วมกันโดยใช้ประเภทการร้องขอล็อกแบบ share - no update, และ shared-read เท่านั้น. สถานะการล็อกนี้เหมาะสำหรับเมื่อผู้ใช้ไม่ตั้งใจจะเปลี่ยนแปลงอ็อบเจ็กต์แต่ต้องการให้แน่ใจว่าไม่มีผู้ใช้รายอื่นเปลี่ยนอ็อบเจ็กต์.

- **Shared-Read**

สามารถใช้งานอ็อบเจ็กต์ร่วมกับประเภทการร้องขอล็อกทั้งหมดนอกเหนือจากแบบ exclusive - no read. นั่นคือ, ผู้ใช้รายอื่นสามารถ request สถานะการล็อกแบบ exclusive-read, shared-update, shared-read, หรือ shared-no update.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับข้อขัดแย้งการล็อก, ให้ดูที่คำอธิบาย iSeries Navigator ออนไลน์.

สถานะการล็อกเป็นตัวบอกสถานะของการร้องขอล็อก. สถานะการล็อกอื่นๆ คือ:

**Held** - request การล็อกถูกทำให้สมบูรณ์และงาน, thread หรือ lock space ได้ครอบครองการล็อก.

**Waiting** - งานหรือ thread กำลังรอเพื่อรับล็อก.

**Requested** - งานหรือ thread ได้ร้องขอการล็อก.

**Lock holders** คืองาน, thread และ lock space ที่กำลังถือครองล็อกอยู่หรือกำลังรอการล็อกบนอ็อบเจ็กต์ที่ถูกล็อกที่ระบุไว้.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับอ็อบเจ็กต์การล็อก, การร้องขอล็อก, lock holders, สถานะการล็อก และขอบเขต, โปรดดูที่คำอธิบายของ iSeries Navigator แบบออนไลน์.

**บันทึกการใช้งาน:** บันทึกการใช้งานจะประกอบด้วยลิสต์แสดงข้อความที่เกี่ยวข้องกับ งานใดงานหนึ่งเฉพาะ. รายละเอียดเพิ่มเติมเกี่ยวกับข้อความต่างๆ เช่น วันและเวลาที่ข้อความถูกส่งออกมา, ก็จะมีการแสดงให้ดูด้วย. เนื่องจากวันและเวลาถูกบันทึกเอาไว้ด้วยในบันทึกการใช้งาน, คุณจึงสามารถระบุได้ว่าเกิดข้อผิดพลาดขึ้นเมื่อใด. >>> [คลิกขวาที่ข้อความแล้วเลือก Properties](#) สำหรับข้อมูลเพิ่มเติม. ส่วนในหน้า **General**, คุณสามารถตรวจสอบว่าใครเป็นคนส่งข้อความ, สาเหตุของข้อความ

และคำอธิบายว่าควรทำอย่างไรต่อไป, หรือ, แก้ข้อผิดพลาดอย่างไร. สำหรับข้อความในบันทึกการใช้งาน, คุณสามารถตรวจสอบที่หน้า **Details** เพื่อดูข้อมูลเกี่ยวกับโปรแกรมที่เป็นตัวส่งข้อความและโปรแกรมที่ข้อความส่งไปหา. ❖ คุณสามารถเปลี่ยนวิธีจัดการบันทึกการใช้งานและประเภทข้อมูลที่เก็บในบันทึกการใช้งานได้ที่หน้า **Job Log** ในส่วนของ “คุณสมบัติของงาน” ในหน้า 46.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการดูบันทึกการใช้งานของงาน, ให้ดูที่ “การเข้าถึงบันทึกการทำงานสำหรับงานแอคทีฟ, รวมถึงงานเซิร์ฟเวอร์” ในหน้า 31.

## thread

thread คือหน่วยงานอิสระภายในงานที่ใช้รีซอร์สงานหลายตัวเพื่อทำให้งานเสร็จสมบูรณ์. ความแตกต่างระหว่างงานและ thread คือ thread ทำงานอยู่ภายในงานโดยช่วยให้งานนั้นเสร็จสิ้นหน้าที่. งานแอคทีฟแต่ละตัวมี thread อย่างน้อยหนึ่งตัว, ซึ่งเรียกว่า initial thread. initial thread จะถูกสร้างขึ้นเป็นส่วนหนึ่งของการเริ่มต้นของงาน. การใช้ thread ภายในงานจะอนุญาตให้รายการต่างๆ เสร็จสิ้นการดำเนินการในเวลาเดียวกัน. ตัวอย่างเช่น, ขณะทำงานกำลังประมวลผล, thread อาจเรียกข้อมูลออกมาดูและคำนวณข้อมูลที่งานต้องการเพื่อให้เสร็จสิ้นการประมวลผล.

สำหรับข้อมูลเพิ่มเติมเรื่อง thread, ให้ดูที่หัวข้อต่อไปนี้:

- “การดำเนินการของ thread”  
การจัดการ thread ผ่าน iSeries Navigator.
- “ชนิด thread” ในหน้า 55  
หัวข้อนี้ครอบคลุมชนิด thread ต่างๆ ที่รันภายในงาน.
- “สถานะ thread” ในหน้า 55  
หัวข้อนี้กล่าวถึงสถานะต่างๆ ของ thread.

**การดำเนินการของ thread:** thread ช่วยให้งานประมวลผลสามารถดำเนินการได้มากกว่าหนึ่งรายการพร้อมกัน. การมอนิเตอร์ thread ที่กำลังรันภายในงานอาจเป็นเรื่องจำเป็นเมื่อคุณพยายามให้งานรันอย่างมีประสิทธิภาพ. เมื่อคุณ “การดู threads รันภายในได้งานเฉพาะ” ในหน้า 19 ที่คุณต้องการจัดการ, การปฏิบัติการดังต่อไปนี้จะพร้อมทำงานโดยการคลิกขวาที่ thread.

## รีเซ็ตสถิติ

อนุญาตให้คุณรีเซ็ตรายการข้อมูลที่คุณกำลังดูอยู่และจะเซตเวลาที่ผ่านไปให้เป็น 00:00:00.

“รายละเอียด: การปฏิบัติงานแอคทีฟ” ในหน้า 50

เนื่องจากฟังก์ชันของ thread คล้ายกับฟังก์ชันของงาน, ดังนั้นฟังก์ชันทั้งสองจึงมีการดำเนินการแบบเดียวกัน. รายละเอียดประกอบด้วยข้อมูลโดยละเอียดเกี่ยวกับการดำเนินการของ thread ต่อไปนี้:

- Call stack
- รายชื่อไลบรารี
- “อ็อบเจกต์ที่ถูกล็อก” ในหน้า 51
- Transaction
- “สถิติการทำงานที่ผ่านมา” ในหน้า 48

## หยุดพัก

อนุญาตให้คุณพัก thread. สามารถพัก thread ได้หลายเวลา. ระบบปฏิบัติการจะคอยตรวจสอบจำนวนครั้งที่ thread ถูกพัก.

## ปล่อย

ปล่อย thread ที่ถูกพักไว้. ต้องปล่อย thread ในแต่ละครั้งที่ถูกพักไว้เพื่อให้ thread รัน.

## ลบ/สิ้นสุด

อนุญาตให้คุณสิ้นสุดการทำงานของ thread หนึ่งตัวหรือมากกว่าที่เลือกไว้. สำหรับข้อมูลเพิ่มเติม, ให้ดูที่ “การลบหรือจบการทำงานของ thread” ในหน้า 19.

## “คุณสมบัติ thread”

แสดงคุณสมบัติอื่นๆ ของ thread.

สำหรับข้อมูลเพิ่มเติมละเอียดเพิ่มเติมเกี่ยวกับการดำเนินการที่คุณสามารถใช้ได้กับ Thread, ให้ดูที่คำอธิบายของ iSeries Navigator.

**คุณสมบัติ thread:** คุณสมบัติ thread ให้ข้อมูลเกี่ยวกับกระบวนการจัดการ thread ของ การจัดการระบบงาน. thread ใช้คุณสมบัติของงานและรีซอร์สหลายอย่างเพื่อประมวลผลงาน (work) ให้กับงาน(job)ที่รันภายใน. เมื่อสร้าง thread แล้ว คุณสามารถ “การดูคุณสมบัติ thread” ในหน้า 19. ผู้ใช้ต้องมี “สิทธิในการใช้งานที่ถูกต้องสำหรับ thread” ในหน้า 55 เพื่อดูรายการ thread หรือเพื่อดูคุณสมบัติของ thread.

*Work with thread properties:* หากต้องการดูหรือเปลี่ยนแปลงคุณสมบัติของ thread, โปรดทำตามขั้นตอนเหล่านี้:

1. ใน iSeries Navigator, ให้แตก **My Connections**.
2. แยกจุดเชื่อมต่อของ iSeries server ออกมา.
3. แยก **Work Management** ออกมา.
4. ดับเบิลคลิก **Active Jobs** หรือ **Server Jobs**, ขึ้นอยู่กับชนิดของงานที่คุณต้องการทำงานด้วย.
5. คลิกขวาที่งานที่คุณต้องการทำงานด้วย, เลือก **Details > Threads**.
6. คลิกขวาที่ thread ที่คุณต้องการทำงานด้วย, เลือก **Properties**.

คุณสมบัติ thread ทั่วไป จะอนุญาตให้คุณดูคุณสมบัติของ thread ได้. คุณสมบัติเหล่านี้รวมถึง thread identifier, “สถานะ thread” ในหน้า 55 โดยละเอียดของ thread, ผู้ใช้ปัจจุบัน, “ชนิด thread” ในหน้า 55 ของการรัน thread, งานที่ thread กำลังรันภายใต้, และ disk pool group ที่ thread กำลังรันเข้า.

คุณสมบัติ การทำงาน อนุญาตให้คุณดูส่วนประกอบพื้นฐานการทำงานและให้คุณเปลี่ยนระดับความสำคัญในการรัน thread. ระดับความสำคัญในการรัน จะระบุความสำคัญของ thread ที่สัมพันธ์กับ thread อื่นๆ ที่รันในระบบ. ช่วงค่าจะเริ่มตั้งแต่ระดับความสำคัญของงานไปจนถึง 99 (หมายความว่าระดับความสำคัญสูงสุดจะต่างกันไป). ระดับความสำคัญในการรันอาจไม่สูงไปกว่าระดับความสำคัญในการรันงานที่ thread กำลังรัน.

คุณยังสามารถดูค่าการทำงานที่ถูกคำนวณไว้ นับตั้งแต่ที่ thread เริ่มทำงาน, ซึ่งรวมถึง CPU และ disk I/O รวม. นอกจากนี้ คุณสามารถดู, รีเฟรช, เซ็ตออฟการรีเฟรชอัตโนมัติ, หรือรีเซ็ต **Elapsed performance statistics** ที่ถูกคำนวณไว้สำหรับ thread. สำหรับรายละเอียดเพิ่มเติม, ให้ดู “สถิติการทำงานที่ผ่านมา” ในหน้า 48.

**สิทธิในการใช้งานที่ถูกต้องสำหรับ thread:** หากต้องการดูและเปลี่ยนคุณสมบัติส่วนใหญ่ของ thread คุณต้องมีสิทธิ์พิเศษ “Job control special authority (\*JOBCTL)” ในหน้า 48, หรือโปรไฟล์ผู้ใช้ของคุณต้องตรงกับ job user identity ของงานที่มี thread อยู่. หากต้องการเปลี่ยนระดับความสำคัญในการรันของ thread, คุณต้องมีสิทธิ์พิเศษ \*JOBCTL. “Thread Control” จะอนุญาตให้คุณดูคุณสมบัติบางตัวของ thread ได้.

หากต้องการพักหรือปล่อย thread, คุณต้องมีสิทธิ์พิเศษ \*JOBCTL หรือสิทธิสำหรับ Thread Control, หรือโปรไฟล์ผู้ใช้ของคุณต้องตรงกับ job user identity ของงานที่มี thread. หากต้องการสิ้นสุด thread, คุณต้องมีสิทธิ์พิเศษ \*SERVICE หรือสิทธิสำหรับ Thread Control.

สำหรับคุณสมบัติของ thread ใดๆ ที่หมายถึงอ็อบเจกต์ OS/400, เช่นไลบรารีในรายชื่อไลบรารี, ผู้ใช้ต้องมีสิทธิ์ที่ถูกต้องสำหรับอ็อบเจกต์นั้น. สำหรับรายละเอียดเพิ่มเติมเรื่องสิทธิในการใช้งานของ iSeries, ให้อ่านที่ภาคผนวก D สิทธิในการใช้งาน

ของคำสั่งที่จำเป็นสำหรับอ็อบเจกต์ที่ใช้งาน ในคู่มือ การอ้างอิงความปลอดภัย iSeries. 

**Thread Control:** สิทธิ Thread Control อนุญาตให้ผู้ใช้ยุติ, พัก, และปล่อย thread ของงานอื่น. สิทธิดังกล่าวจะอนุญาตให้ผู้ใช้เรียกดูข้อมูลเกี่ยวกับ thread ของงานอื่นได้. ผู้ใช้แต่ละรายสามารถรับและเรียก Thread Control ได้โดยใช้ฝ่ายสนับสนุน iSeries Navigator’s Application Administration, หรือโดยการ ใช้ Change Function Usage Information (QSYCHFUI) API, พร้อมด้วย function ID ของ QIBM\_SERVICE\_THREAD. สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการใช้งานแอสพลิคชัน, ให้อ่านที่ การใช้งานแอสพลิคชัน.

**ชนิด thread:** ชนิด thread จะกำหนดวิธีการสร้าง thread บนระบบ.

ชนิดของ thread คือ:


### ผู้ใช้

thread จะถูกสร้างขึ้นโดย customer application. initial thread ในงานจะเป็น thread ผู้ใช้เสมอ. ฟิลด์ Allow multiple threads ต้องถูกตั้งค่าให้เป็น yes เพื่อให้ multiple user thread พร้อมใช้งาน.

### ระบบ

thread จะถูกสร้างขึ้นโดยระบบในนามของผู้ใช้. ฟังก์ชันระบบบางอย่างใช้งาน thread ระบบเพื่อเสร็จสิ้นการประมวลผล. หากแอสพลิคชันของลูกค้าใช้งานฟังก์ชันระบบที่ใช้งาน thread, thread ระบบจะถูกใช้งาน.

**หมายเหตุ:** threads บน iSeries Navigator, ตามค่าดีฟอลต์, คุณจะเห็น **Initial** เป็นชนิดของ thread แรกในรายการ. initial thread คือ

thread แรกที่ถูกสร้างขึ้นภายในงานเมื่อเริ่มทำงาน. ใน iSeries Navigator, initial thread ถูกแทนค่าด้วยไอคอนนี้ . คุณไม่สามารถ “การลบหรือจบการทำงานของ thread” ในหน้า 19 initial thread ได้.

**สถานะ thread:** สามารถดูสถานะของ thread ได้จากหน้า **ทั่วไป** ในไดอะล็อก Thread properties, ภายใต้อสถานะ Detailed. ตัวอย่างของสถานะโดยละเอียด คือ:

### กำลังรอดอนจากคิว

thread ของงานกำลังรอดอนการดำเนินการถอนจากคิวที่เสร็จสมบูรณ์. การถอนจากคิวคือการดำเนินการเพื่อย้ายข้อความจากคิว. ข้อความคือการสื่อสารที่ถูกส่งจากบุคคลหนึ่งหรือโปรแกรมหนึ่งไปยังอีกบุคคลหนึ่งหรือโปรแกรม

หนึ่ง. โดยทั่วไป, ข้อความจะถูกวางคิว (วาง) ไว้บนอ็อบเจ็กต์ระบบคิวโดย thread หนึ่งตัวและถูกถอนออกจากคิว (ลบออก) โดย thread อีกตัวหนึ่ง.

**หมายเหตุ:** เมื่อการรอกการถอนจากคิวแสดงอยู่ในหน้าคุณสมบัติ, ข้อมูลเพิ่มเติมที่ระบุคิวกที่กำลังรออยู่จะปรากฏขึ้นมา. เมื่องานหรือ thread กำลังรอกการถอนออกจากคิวของอ็อบเจ็กต์ OS/400<sup>(R)</sup>, คุณจะเห็นชื่ออ็อบเจ็กต์ความยาว 10 อักขระ, โลบราลีของอ็อบเจ็กต์, และชนิดของอ็อบเจ็กต์. ถ้างานหรือ thread กำลังรอกการดำเนินการถอนออกจากคิวให้เสร็จสิ้นสำหรับอ็อบเจ็กต์ภายใน, คุณจะเห็นชื่ออ็อบเจ็กต์ที่มีความยาว 30 อักขระ. สำหรับอ็อบเจ็กต์ภายใน คุณต้องมีสิทธิ์พิเศษ (\*JOBCTL) เพื่อดูชื่อที่มีความยาว 30 อักขระ.

สถานะโดยละเอียดจะแสดงค่าสถานะเชื่อมโยง (สถานะ - x), ซึ่งจะบอกรายละเอียดเพิ่มเติมเกี่ยวกับสถานะปัจจุบันของ thread. ตัวอย่างของสถานะโดยละเอียดบวกค่าสถานะเชื่อมโยงคือ:

### Held (n)

thread แต่ละอันจะถูกพักไว้. ต่างจากงาน, thread อาจมีการพักไว้หลายครั้งในเวลาเดียวกัน. จำนวน (ตัวอย่างเช่น, Held (3)) ที่ตามหลังสถานะ thread จะเป็นตัวบอกผู้ใช้ถึงจำนวนครั้งที่ thread ถูกพักโดยไม่ถูกปล่อย. ตัวอย่างเช่น, หาก thread มีการค้างไว้สามครั้งและถูกปล่อยหนึ่งครั้ง, thread จะมีการพักสองครั้ง. จะแสดงหมายเลขก็ต่อเมื่อสถานะปรากฏขึ้นบนหน้า Properties และจะไม่ปรากฏเมื่อมีการแสดงผลในรายการ. หากต้องการเรียกการประมวลผล thread ขึ้นมาทำงานใหม่, ให้เลือกการดำเนินการแบบ Release สำหรับ thread.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับสถานะ thread อื่นๆ, ให้ดูที่คำอธิบาย iSeries Navigator.

## คิวงาน

คิวงานประกอบด้วย “รายการเรียงลำดับ” ในหน้า 57 ของงาน ที่รอกการประมวลผลโดยระบบย่อย. คิวงานจะเป็นที่แรกที่ “งานแบบแบตซ์” ในหน้า 37 ที่ส่งไปพักรอ ก่อนที่จะไปทำงาน อยู่ในระบบย่อย. งานจะหยุดรออยู่ที่นั่นจนกว่าจะได้เงื่อนไขให้ทำงาน. งานที่อยู่ในคิวงานจะได้รับการประมวลผลก็ต่อเมื่อมีระบบย่อยที่แอกทีฟรับเอางานที่อยู่ในคิวงานไป. เมื่อระบบย่อยเริ่มทำงาน, ระบบย่อยดังกล่าวจะพยายามจัดสรรคิวงานที่ถูกตั้งไว้ให้รับงานเข้ามา และจะต้องจัดสรรคิวงานเพื่อทำการประมวลผลงานจากคิวงานนั้นให้ประสบความสำเร็จ. ดังนั้น, ในขณะที่ระบบย่อยกำลังประมวลผลงานจากคิวงานต่างๆ อยู่, จะมีระบบย่อยเดียวเท่านั้นที่จะประมวลผลงานเฉพาะจากคิวงานเดียวในหนึ่งครั้ง.


ระบบย่อยจะทำการเลือกงานจากคิวงานตามลำดับความสำคัญ, ภายใต้เงื่อนไขที่ได้ตั้งไว้ให้กับหัวข้อความสำคัญแต่ละอัน. งานแต่ละชิ้นจะมีค่าความสำคัญสำหรับคิวงานที่สามารถจัดการได้เมื่องานนั้นอยู่ในคิวงานผ่านทาง “คุณสมบัติของงาน” ในหน้า 46. ชุดคิวงานพื้นฐานจะมาพร้อมกับระบบของคุณ. นอกจากนั้น, คุณยังสามารถสร้างคิวงานเพิ่มเติมตามความต้องการได้อีกด้วย.

**หมายเหตุ:** สามารถเรียกใช้ APIs, เช่น Open List of Job Queues (QSPOLJBQ) และ Retrieve Job Queue Information (QSPRJQBQ), เพื่อรับข้อมูลบนคิวงาน. สำหรับข้อมูลเพิ่มเติมเรื่อง APIs, โปรดดู Application programming interfaces (APIs).

สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับงานและคิวงาน, กรุณาดูหัวข้อต่อไปนี้:

- “วิธีที่การทำงานจะเข้าสู่ระบบ” ในหน้า 82.  
เรียนรู้วิธีการที่งานจะเข้าสู่คิวงาน.
- “วิธีการทำงานของคิวงาน” ในหน้า 57  
เรียนรู้วิธีที่งานจะออกจากคิวงานเข้าสู่ระบบย่อย.

## • การสร้างคิวงาน

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการสร้างคิวงาน ให้ดูบทที่ 8 ของคู่มือ V4R5 Work Management  (ประมาณ 2720 กิโลไบต์ หรือ 573 หน้า).

## รายการเรียงลำดับ

รายการเรียงลำดับหมายถึงลำดับที่งานจะปรากฏขึ้นบนคิวงาน. ค่าที่ช่วยกำหนดลำดับงานบนคิวงานมีดังต่อไปนี้:

### สภาพพร้อมใช้งาน

หมายถึงสถานะงานบนคิวงาน. ค่าที่เป็นไปได้ตามลำดับคือ กำลังรอ, ถูกกำหนด, และถูกพัก.

### ระดับความสำคัญ

หมายถึงระดับความสำคัญของงานบนคิวงาน. ค่าระดับความสำคัญที่เป็นไปได้คือ 0-9, โดยที่ 0 เป็นระดับความสำคัญสูงสุด.

### วันที่และเวลา

หมายถึงวันที่และเวลาของงาน:

- หากงานถูกกำหนดไว้, วันที่และเวลาหมายถึงเมื่องานถูกกำหนดให้รัน.
- หากงานไม่ได้ถูกกำหนดลำดับไว้, วันที่และเวลาหมายถึงเมื่องานเข้าสู่ระบบ.

หมายเหตุ: มีกรณีที่วันที่และเวลาหยุดทำงานซึ่งถูกตั้งค่าด้วยตนเองให้วางตำแหน่งงานที่ถูกย้ายไปยังคิวงานเฉพาะให้ถูกต้อง.

## วิธีการทำงานของคิวงาน

งานจะถูกส่งเข้าไปในคิวงานเพื่อทำงานใน ระบบย่อยหลังจากที่คิวงานได้รับการจัดสรรจากระบบย่อย แอ็คทีฟ. มีปัจจัยหลายประการที่กำหนดวิธีการเลือกงานจาก คิวงาน. งานที่ยังอยู่ในคิวงานสามารถย้ายจากคิวงานหนึ่งไปยังคิวงานอื่นได้, ทั้งนี้เพื่อให้เกิดประสิทธิภาพในการทำงานที่ดีขึ้น.

ต่อไปนี้เป็นวิธีการเองงานออกจากคิวงาน:

### จำนวนงานแอ็คทีฟมากที่สุดสำหรับระบบย่อย

ตัวเลขนี้แทนจำนวนสูงสุดของงานที่สามารถรันได้ในระบบย่อยหนึ่งๆ. เมื่อจำนวนงานมีมากถึงจำนวนดังกล่าวแล้ว, งานอื่นๆ ก็ไม่สามารถทำงานในระบบย่อยนี้ได้.

### จำนวนงานแอ็คทีฟสูงสุดสำหรับคิวงาน

ตัวเลขนี้แทนจำนวนสูงสุดของงานในคิวงานที่สามารถรันพร้อมกัน ได้ในระบบย่อย. เมื่อจำนวนงานมีมากถึงจำนวนดังกล่าวแล้ว, งานอื่นๆ ก็ไม่สามารถทำงานในคิวงานนี้ได้.

### ระดับความสำคัญในคิวงาน

งานที่กำลังรอที่จะรันจะถูกเลือกขึ้นมารันตามระดับความสำคัญใน คิวงาน. ระบบจะพยายามที่จะรันงานที่มีความสำคัญสูงกว่าก่อน (ระดับความสำคัญในคิวงานมีตั้งแต่ 0-9 โดยที่ 0 จะมีความสำคัญสูงกว่า), แต่ถ้าจำนวนของงานที่รันจากระดับความสำคัญมีมากถึงค่า Maximum Active Jobs, ค่าระดับความสำคัญต่อไปจะถูกประมวลผลต่อไป. (หากงานที่มีความสำคัญอยู่ในคิวงานเดียวกัน, งานที่ส่งเข้า มาก่อนจะทำงานก่อน, แล้วตามมาด้วยงานที่เข้ามาที่สอง, เป็นเช่นนี้ต่อไป.)



สำหรับข้อมูลอย่างละเอียด, โปรดดูที่ “การเปลี่ยนระดับความสำคัญของงานภายในคิวงาน” ในหน้า 20.

## ลำดับ

คุณสามารถระบุลำดับในการเรียงคิวงานใน subsystem description ได้. ตัวเลขลำดับจะบอกถึงการจัดเรียงของคิวงานเพื่อให้ระบบย่อยทำการประมวลผล. ระบบย่อยจะดึงเอางานที่มีค่าลำดับต่ำสุดออกจากคิวงานก่อน. หากมีงานในคิวงาน มากกว่าหนึ่งงานขึ้นไป, หรือมีการไปถึงค่าสูงสุดค่าใดค่าหนึ่งที่เกี่ยวข้องกับคิวงาน, ระบบย่อยจะประมวลผลคิวงานที่มีค่าลำดับสูงสุดเป็นการถัดไป.

สำหรับข้อมูลอย่างละเอียดเกี่ยวกับการย้ายงาน, ให้อ่านที่ “การย้ายงานไปยังคิวงานอื่น” ในหน้า 21.

## ระบบย่อย

ระบบย่อย เป็นที่ซึ่งมีการประมวลผลบนเซิร์ฟเวอร์ iSeries<sup>(TM)</sup>. งาน ทั้งหมด, ยกเว้น “งานระบบ” ในหน้า 40, จะรันอยู่ภายในระบบย่อย.

ในเชิงเทคนิคแล้ว, ระบบย่อยเป็นสภาวะปฏิบัติการหนึ่งๆ ที่ได้กำหนดไว้ล่วงหน้าเพื่อให้ระบบประสานการไหลของงานและจัดการรีซอร์ส. ระบบจะประกอบด้วยระบบย่อยหลายๆ ระบบ, โดยที่ระบบทั้งหมดดำเนินการ เป็นเอกเทศต่อกัน. ระบบย่อยจะจัดการรีซอร์สเอง. และระบบย่อยแต่ละอันจะรันการปฏิบัติการที่ไม่เหมือนกัน. ตัวอย่างเช่น, ระบบย่อยหนึ่งระบบ อาจถูกตั้งค่าให้จัดการกับงานแบบโต้ตอบเท่านั้น, ขณะที่ระบบย่อยอื่นจะจัดการกับงานแบ็ตช์เท่านั้น. สามารถออกแบบระบบย่อยให้จัดการงานได้หลายชนิด. ระบบจะอนุญาตให้คุณกำหนดจำนวนระบบย่อยและชนิดของงานที่ระบบย่อยเหล่านั้นจะจัดการได้อีกด้วย.

ระบบย่อยอาจเป็นแบบแอคทีฟหรือ inactive. ระบบย่อยที่แอคทีฟคือระบบที่ถูกใช้งานอยู่ (โปรดดูที่ “การเริ่มใช้งานระบบย่อย” ในหน้า 25 สำหรับรายละเอียด). ระบบย่อยแบบ inactive คือระบบที่ยังไม่ได้ถูกเรียกใช้งาน, หรือถูกระงับการทำงาน (โปรดดูที่ “หยุดระบบย่อย” ในหน้า 25 สำหรับรายละเอียด).

ระบบย่อยสำหรับการควบคุมคือระบบย่อยแบบ interactive ซึ่งจะเริ่มทำงานโดยอัตโนมัติเมื่อระบบเริ่มทำงาน และยังเป็นระบบย่อยที่ผู้บริหารระบบควบคุมระบบผ่านในระหว่างการเริ่มต้นเปิดระบบอีกด้วย.

งานของระบบย่อย คืองานที่เกิดจากระบบปฏิบัติการเพื่อใช้บริหารรีซอร์สและใช้เริ่มงาน, ควบคุมงาน, และปิดงาน.

**หมายเหตุ:** สามารถเรียกใช้ APIs, เช่น Retrieve Subsystem Information (QWDRSBSD) และ Retrieve System Status (QWCRSSTS), เพื่อรับข้อมูลบนระบบย่อย. สำหรับข้อมูลเพิ่มเติมเรื่อง APIs, โปรดดู Application programming interfaces (APIs).

กรุณาดูหัวข้อต่อไปนี้เป็นสำหรับรายละเอียดเกี่ยวกับระบบย่อย:

“Subsystem description” ในหน้า 59

คุณลักษณะ run-time ของระบบย่อยที่กำหนดไว้ใน subsystem description.

“ระบบย่อยที่มีมาพร้อมกับระบบ” ในหน้า 70

การกำหนดค่าระบบย่อยที่สมบูรณ์ 2 ชุดซึ่งส่งมาจาก IBM<sup>(R)</sup>.

“ระบบย่อยที่ผู้ใช้กำหนดเอง” ในหน้า 71

คุณสามารถสร้าง subsystem description ด้วยตัวคุณเองได้.

“คุณสมบัติของระบบย่อย” ในหน้า 72



มีแอ็ททริบิวต์ของระบบย่อย.





“วงรอบชีวิตของระบบย่อย” ในหน้า 73



สิ่งนี้จะอธิบายวิธีที่งานถูกประมวลผลบนเซิร์ฟเวอร์ iSeries.

## Subsystem description

คุณลักษณะการรันใหม่ของระบบย่อยจะถูกกำหนดไว้ในแอ็ททริบิวต์ชื่อ **subsystem description**. subsystem description ทำหน้าที่เสมือนชุดคำสั่งเครื่องที่บอกระบบย่อยถึงวิธี, สถานที่, ปริมาณที่งานได้เข้ามาสู่ระบบย่อย, และ รีซอร์สใดบ้างที่ระบบย่อยได้ใช้การทำงาน. ระบบย่อยจะถูกสร้างขึ้น เมื่อได้มีการกำหนดหรือสร้าง subsystem description ขึ้น. ระบบย่อยแอ็ททริบิวต์จะใช้ชื่อธรรมชาติของ subsystem description.

สำหรับรายละเอียดเกี่ยวกับข้อมูลที่อยู่ใน subsystem description, กรุณาดูที่ตารางต่อไปนี้:

ข้อมูลที่อยู่ใน subsystem description	รายละเอียด	ข้อมูลเพิ่มเติม (คู่มือการจัดการระบบงาน)
แอ็ททริบิวต์ระบบย่อย	ระบุลักษณะทั่วไปของระบบ: <ul style="list-style-type: none"><li>แอ็ททริบิวต์เชิงปฏิบัติการ เช่น จำนวนงานที่สามารถแอ็ททริบิวต์ในระบบย่อยในเวลาเดียวกัน, และการแสดงผลการ sign-on.</li><li>พูลหน่วยความจำที่ระบบย่อยใช้งาน.</li><li>สิทธิในการใช้งานกับ subsystem description.</li><li>รายละเอียดแบบข้อความของ subsystem description.</li></ul>	การเปลี่ยนแปลงไฟล์แสดงผล sign-on file, บทที่ 4 ของคู่มือการจัดการระบบงาน.  <a href="#">Link to PDF &lt;twb##" /&gt;</a>
Work entry	work entry ใน subsystem description จะกำหนดซอร์สที่งานจะได้รับเข้าสู่การประมวลผลในระบบย่อย. หรือเรียกอีกอย่างว่าเป็นที่ๆ งานจะเข้าสู่ระบบย่อยนั่นเอง.	Work entry, บทที่ 4 ของคู่มือการจัดการระบบงาน.  <a href="#">Link to PDF &lt;twb##" /&gt;</a>
	Autostart job entry	จะกำหนดให้งาน autostart เริ่มทำงานทันทีที่ระบบย่อยสตาร์ทขึ้นมา.  <a href="#">Link to PDF &lt;twb##" /&gt;</a>
	communication entry	จะกำหนดอุปกรณ์สื่อสารที่ระบบอื่นใช้เพื่อส่งงานเข้ามา.  <a href="#">Link to PDF &lt;twb##" /&gt;</a>

ข้อมูลที่อยู่ใน subsystem description	รายละเอียด	ข้อมูลเพิ่มเติม (คู่มือการจัดการระบบงาน)
Job queue entry	จะกำหนดคิวงานที่จะรับงานเข้ามาและระบุว่าจะรับงานได้มากเพียงใด.	งานแบ็คชิ่ง, บทที่ 8 ของคู่มือการจัดการระบบงาน.  Link to PDF <twb### />
Prestart job entry	จะกำหนดข้อมูลที่ใช้เมื่อมีการเรียกใช้งาน prestart ขึ้นมา.	งาน Prestart, บทที่ 11 ของคู่มือการจัดการระบบงาน.  Link to PDF <twb### />
workstation entry	จะกำหนดเวิร์กสเตชันที่จะรับงานเข้ามา.	งานแบบโต้ตอบ, บทที่ 6 ของคู่มือการจัดการระบบงาน.  Link to PDF <twb### border="0" />
routing entry	จะกำหนดพูลหน่วยความจำของระบบย่อยที่จะใช้, โปรแกรมการควบคุมการรัน, และข้อมูลรันไทม์.	routing entry, บทที่ 4 ของคู่มือการจัดการระบบงาน.  Link to PDF <twb### />

อ็อบเจ็กต์ subsystem descriptions จะมีมากับทุกระบบตั้งแต่แรกเริ่ม. ด้านล่างนี้คือการอัปเดต subsystem description ที่มีมากับเซิร์ฟเวอร์ iSeries. สำหรับแต่ละอ็อบเจ็กต์, ตารางนี้จะให้ข้อมูลดังต่อไปนี้:

ชื่ออ็อบเจ็กต์

คำสั่งที่ใช้เพื่ออัปเดตอ็อบเจ็กต์

พารามิเตอร์คำสั่งอื่นที่นอกเหนือจากพารามิเตอร์ดีฟอลต์

ตารางนี้และภาคผนวก C ในคู่มือ การจัดการงาน  ลิงก์ไปที่ PDF <twb### /> จะแสดงค่า subsystem descriptions ส่วนใหญ่ที่ติดมากับ iSeries.

อ็อบเจ็กต์	Addition, Deletion, หรือ Update	พารามิเตอร์ที่ไม่เป็นดีฟอลต์
QBASE	Added a communication entry (ADDCMNE)	SBSD (QSYS/QBASE) DEV (Q1PLOC) DFTUSR (*NONE) MODE (Q1PMOD) MAXACT (0)
QBASE	Added a communication entry (ADDCMNE)	SBSD (QSYS/QBASE) REMLOCNAME (Q1PLOC) DFTUSR (*NONE) MODE (Q1PMOD) MAXACT (0)

อีอบเจ็กต์	Addition, Deletion, หรือ Update	พารามิเตอร์ที่ไม่เป็นดีฟอลต์
QBASE	Added prestart job entry (ADDPJE)	SBSDB (QSYS/QBASE) PGM (QSYS/QZSCSRVR) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(3) JOB (*PGM) JOBD (*USRPRF) MAXUSE (1) WAIT (*YES) POOLID (2) CLS (QGPL/QCASERVR *CALC *NONE *CALC)
QBASE	Added prestart job entry (ADDPJE)	SBSDB (QSYS/QBASE) PGM (QSYS/QNPSEVR) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(3) JOB (*PGM) JOBD (*USRPRF) MAXUSE (200) WAIT (*YES) POOLID (1) CLS (QGPL/QCASERVR *CALC *NONE *CALC)
QBASE	Added prestart job entry (ADDPJE)	SBSDB (QSYS/QBASE) PGM (QSYS/QZRCSRVR) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(3) JOB (*PGM) JOBD (*USRPRF) MAXUSE (1) WAIT (*YES) POOLID (2) CLS (QGPL/QCASERVR *CALC *NONE *CALC)

อีอบเจ็กต์	Addition, Deletion, หรือ Update	พารามิเตอร์ที่ไม่เป็นดีฟอลต์
QCMN	Added a communication entry (ADDCMNE)	SBSD (QSYS/QCMN) REMLOCNAME (Q1PLOC) DFTUSR (*NONE) MODE (Q1PMOD) MAXACT (0)
QCMN	Added a communication entry (ADDCMNE)	SBSD (QSYS/QCMN) DEV (Q1PLOC) DFTUSR (*NONE) MODE (Q1PMOD) MAXACT (0)
QCMN	Added prestart job entry (ADDPJE)	SBSD (QSYS/QCMN) PGM (QSYS/QZRC SRVR) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(3) JOB (*PGM) JOB (*USRPRF) MAXUSE (1) WAIT (*YES) POOLID (1) CLS (QGPL/QC ASERVR *CALC *NONE *CALC)
QCMN	Added prestart job entry (ADDPJE)	SBSD (QSYS/QCMN) PGM (QSYS/QZSC SRVR) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(3) JOB (*PGM) JOB (*USRPRF) MAXUSE (1) WAIT (*YES) POOLID (1) CLS (QGPL/QC ASERVR *CALC *NONE *CALC)

อีอบเจ็กต์	Addition, Deletion, หรือ Update	พารามิเตอร์ที่ไม่เป็นดีฟอลต์
QCMN	Added prestart job entry (ADDPJE)	SBSDB (QSYS/QCMN) PGM (QSYS/QNPSERVER) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(3) JOB (*PGM) JOB (*USRPRF) MAXUSE (200) WAIT (*YES) POOLID (1) CLS (QGPL/QCASERVER *CALC *NONE *CALC)
QSERVER	Added prestart job entry (ADDPJE)	SBSDB (QSYS/QSERVER) PGM (QSYS/QZDAINIT) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(3) JOB (*PGM) JOB (*USRPRF) MAXUSE (1) WAIT (*YES) POOLID (1) CLS (QGPL/QPWSERVER *CALC *NONE *CALC)
QSERVER	Added prestart job entry (ADDPJE)	SBSDB (QSYS/QSERVER) PGM (QSYS/QPWFSEVSO) USER (QUSER) STRJOBS (*NO) INLJOBS(1) THRESHOLD (1) ADLJOBS(2) MAXJOBS (*NOMAX) JOB (*USRPRF) JOB (*PGM) MAXUSE (200) WAIT (*YES) POOLID (1) CLS (QGPL/QPWFSEVSERVER *CALC *NONE *CALC)

อีอบเจ็กต์	Addition, Deletion, หรือ Update	พารามิเตอร์ที่ไม่เป็นดีฟอลต์
QSYSWRK	Added job queue entry (ADDJOBQE)	SBSD (QSYS/QSYSWRK) JOBQ (QSYS/Q1PSCHQ) MAXACT (1) SEQNBR (70)
QSYSWRK	Added job queue entry (ADDJOBQE)	SBSD (QSYS/QSYSWRK) JOBQ (QSYS/Q1PSCHQ2) MAXACT (1) SEQNBR (80)
QSYSWRK	Added job queue entry (ADDJOBQE)	SBSD (QSYS/QSYSWRK) JOBQ (QSYS/Q1PSCHQ3) MAXACT (1) SEQNBR (90)
QSYSWRK	Added an autostart job entry (ADDAJE)	SBSD (QSYS/QSYSWRK) JOB (QGLDPUBA) JOBQ (QSYS/QGLDPUBA)
QSYSWRK	Added an autostart job entry (ADDAJE)	SBSD (QSYS/QSYSWRK) JOB (QGLDPUBE) JOBQ (QSYS/QGLDPUBE)
QSYSWRK	Added autostart job entry (ADDAJE)	SBSD (QSYS/QSYSWRK) JOB (QPM400) JOBQ (QSYS/Q1PJOBQ)
QSYSWRK	Added a communication entry (ADDCMNE)	SBSD (QSYS/QSYSWRK) DEV (Q1PDEV) JOBQ (*USRPRF) DFTUSR (QUSER) MODE (Q1PMOD) MAXACT (*NOMAX)
QSYSWRK	Added a communication entry (ADDCMNE)	SBSD (QSYS/QSYSWRK) DEV (Q1PLOC) JOBQ (*USRPRF) DFTUSR (QPM400) MODE (Q1PMOD) MAXACT (*NOMAX)
QSYSWRK	Added a communication entry (ADDCMNE)	SBSD (QSYS/QSYSWRK) RMTLOCNAME (Q1PLOC) JOBQ (*USRPRF) DFTUSR (QPM400) MODE (Q1PMOD) MAXACT (*NOMAX)

อีอบเจ็กต์	Addition, Deletion, หรือ Update	พารามิเตอร์ที่ไม่เป็นดีฟอลต์
QSYSWRK	Added routing entries (ADDRTGE)	SBSD (QSYS/QSYSWRK) SEQNBR (2150) CMPVAL (TOTNTP) PGM (QSYS/QTOTSNTTP) CLS (QSYS/QSYSCLS10)
QSYSWRK	Added routing entry (ADDRTE)	SBSD (QSYSWRK) SEQNBR (300) CMPVAL (PGMEVOKE 29) PGM (*RTGDTA) CLS (QSYS/QSYSCLS50) MAXACT (*NOMAX) POOLID (1)
QSYSWRK	Added routing entry (ADDRTGE)	SBSD (QSYS/QSYSWRK) SEQNBR (2536) CMPVAL ('QZSCSRVSD') PGM (QSYS/QZSCSRVSD) CLS (QGPL/QCASERVR)
QSYSWRK	Added routing entry (ADDRTGE)	SBSD (QSYS/QSYSWRK) SEQNBR (2537) CMPVAL ('QZHQSRVD') PGM (QSYS/QZHQSRVSD) CLS (QGPL/QCASERVR)
QSYSWRK	Added routing entry (ADDRTGE)	SBSD (QSYS/QSYSWRK) SEQNBR (2538) CMPVAL ('QNPSEVRD') PGM (QSYS/QNPSEVRD) CLS (QGPL/QCASERVR)
QSYSWRK	Added routing entry (ADDRTGE)	SBSD (QSYS/QSYSWRK) SEQNBR (2539) CMPVAL ('QZRCSRVD') PGM (QSYS/QZRCSRVD) CLS (QGPL/QCASERVR)
QSYSWRK	Added routing entry (ADDRTGE)	SBSD (QSYS/QSYSWRK) SEQNBR (2540) CMPVAL ('QZSOSGND') PGM (QSYS/QZSOSGND) CLS (QGPL/QCASERVR)
QSYSWRK	Added routing entry (ADDRTGE)	SBSD (QSYS/QSYSWRK) SEQNBR (2541) CMPVAL ('QZSOSMAPD') PGM (QSYS/QZSOSMAPD) CLS (QGPL/QCASERVR)

อีอบเจ็กต์	Addition, Deletion, หรือ Update	พารามิเตอร์ที่ไม่เป็นดีฟอลต์
QSYSWRK	Added routing entry (ADDRTGE)	SBSD (QSYS/QSYSWRK) SEQNBR (2170) CMPVAL ('QSYEIMMON') PGM (QSYS/QSYEIMMON) CLS (QSYS/QSYSCLS20) MAXACT (*NOMAX) POOLID (1)
QSYSWRK	Added routing entry (ADDRTGE)	SBSD (QSYS/QSYSWRK) SEQNBR (2200) CMPVAL ('QYASPPGM') PGM (QSYS/QYASPPGM) CLS (QSYS/QSYSCLS20) MAXACT (*NOMAX) POOLID (1)
<pre>&gt;&gt; Start of change &lt;twb###" /&gt;QSYSWRK</pre>	Added an autostart job entry (ADDAJE)	SBSD (QSYS/QSYSWRK) JOB (QS9AJE) JOB(QSYS/QS9AJE) << End of change <twb###" />
<pre>&gt;&gt; Start of change &lt;twb###" /&gt;QSYSWRK</pre>	Added an autostart job entry (ADDAJE)	SBSD (QSYS/QSYSWRK) JOB (QCSTSRCD) JOB(QSYS/QCSTSRCD) << End of change <twb###" />
<pre>&gt;&gt; Start of change &lt;twb###" /&gt;QSYSWRK</pre>	Added routing entry (ADDRTGE)	SBSD (QSYS/QSYSWRK) SEQNBR (2220) CMPVAL ('QS9PAL') PGM (QSYS/QCMD) CLS (QSYS/QSYSCLS50) MAXACT (1) << End of change <twb###" />
<pre>&gt;&gt; Start of change &lt;twb###" /&gt;QSYSWRK</pre>	Added routing entry (ADDRTGE)	SBSD (QSYS/QSYSWRK) SEQNBR (2221) CMPVAL ('QS9PRB') PGM (QSYS/QCMD) CLS (QSYS/QSYSCLS50) MAXACT (1) << End of change <twb###" />
<pre>&gt;&gt; Start of change &lt;twb###" /&gt;QSYSWRK</pre>	Added job queue entry (ADDJOBQE)	SBSD (QSYS/QSYSWRK) JOBQ (QSYS/QSJINV) MAXACT (1) SEQNBR (100) << End of change <twb###" />



อีอบเจ็กต์	Addition, Deletion, หรือ Update	พารามิเตอร์ที่ไม่เป็นดีฟอลต์
<p>▶ Start of change &lt;twb##" /&gt;QSYSWRK</p>	<p>Added routing entry (ADDRTGE)</p>	<p>SBSD (QSYS/QSYSWRK) SEQNBR(2230) CMPVAL('SERVICERMDRVR') PGM(QSYS/QSVRMEVJ) CLS(QSYS/QSYSCLS25) MAXACT(*NOMAX) ◀ End of change &lt;twb##" /&gt;</p>
<p>QUSRWRK</p>	<p>Added prestart job entry (ADDPJE)</p>	<p>SBSD (QSYS/QSYSWRK) PGM (QSYS/QZSOSIGN) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(2) MAXJOBS (*NOMAX) JOB (*PGM) JOB (QSYS/QZBSJOB) MAXUSE (200) WAIT (*YES) POOLID (1) CLS (QGPL/QCASERVR *CALC *NONE *CALC)</p>
<p>QUSRWRK</p>	<p>Added prestart job entry (ADDPJE)</p>	<p>SBSD (QSYS/QUSRWRK) PGM (QSYS/QZSCSRVS) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(2) MAXJOBS (*NOMAX) JOB (*PGM) JOB (QSYS/QZBSJOB) MAXUSE (200) WAIT (*YES) POOLID (1) CLS (QGPL/QCASERVR *CALC *NONE *CALC)</p>

อีอบเจ็กต์	Addition, Deletion, หรือ Update	พารามิเตอร์ที่ไม่เป็นดีฟอลต์
QUSRWRK	Added prestart job entry (ADDPJE)	SBSDB (QSYS/QUSRWRK) PGM (QSYS/QNPSERVS) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(2) MAXJOBS (*NOMAX) JOB (*PGM) JOBD (QSYS/QZBSJOBDB) MAXUSE (200) WAIT (*YES) POOLID (1) CLS (QGPL/QCASERVR *CALC *NONE *CALC)
QUSRWRK	Added prestart job entry (ADDPJE)	SBSDB (QSYS/QUSRWRK) PGM (QSYS/QZRCRVS) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(2) MAXJOBS (*NOMAX) JOB (*PGM) JOBD (QSYS/QZBSJOBDB) MAXUSE (1) WAIT (*YES) POOLID (1) CLS (QGPL/QCASERVR *CALC *NONE *CALC)
QUSRWRK	Added prestart job entry (ADDPJE)	SBSDB (QSYS/QUSRWRK) PGM (QSYS/QZDASOINIT) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(2) MAXJOBS (*NOMAX) JOB (*PGM) JOBD (*USRPRF) MAXUSE (200) WAIT (*YES) POOLID (1) CLS (QGPL/QPWFSEVER *CALC *NONE *CALC)

อีอบเจ็กต์	Addition, Deletion, หรือ Update	พารามิเตอร์ที่ไม่เป็นดีฟอลต์
QUSRWRK	Added prestart job entry (ADDPJE)	SBSDB (QSYS/QUSRWRK) PGM (QSYS/QZHQSSRV) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(2) MAXJOBS (*NOMAX) JOB (*PGM) JOBDB (QSYS/QZBSJOBDB) MAXUSE (200) WAIT (*YES) POOLID (1) CLS (QGPL/QCASERVR *CALC *NONE *CALC)
QUSRWRK	Added prestart job entry (ADDPJE)	SBSDB (QSYS/QUSRWRK) PGM (QSYS/QZDASSINIT) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(2) MAXJOBS (*NOMAX) JOB (*PGM) JOBDB (QSYS/*USRPRF) MAXUSE (200) WAIT (*YES) POOLID (1) CLS (QSYS/QPWFSESRVR *CALC *NONE *CALC)
QUSRWRK	Added prestart job entry (ADDPJE)	SBSDB (QSYS/QUSRWRK) PGM(QSYS/QRWTSRVR) USER (QUSER) STRJOBS (*YES) INLJOBS(1) THRESHOLD (1) ADLJOBS(2) MAXJOBS (*NOMAX) JOB (*PGM) JOBDB (*USRPRF) MAXUSE (200) WAIT (*YES) POOLID (1) CLS (QSYS/QSYSCLS20 *CALC *NONE *CALC)

อีอบเจ็กต์	Addition, Deletion, หรือ Update	พารามิเตอร์ที่ไม่เป็นดีฟอลต์
<pre>&gt;&gt; Start of change &lt;twb##" /&gt;QUSRWRK</pre>	Added routing entry (ADDRTGE)	SBSDB (QSYS/QUSRWRK) SEQNBR (2210) CMPVAL (WATCHEVENT) PGM (QSYS/QSCWCMON) CLS (QSYS/QSYSCLS25) MAXACT (*NOMAX) POOLID (1) << End of change <twb##" />
<pre>&gt;&gt; Start of change &lt;twb##" /&gt;QUSRWRK</pre>	Added routing entry (ADDRTGE)	SBSDB (QSYS/QUSRWRK) SEQNBR (2211) CMPVAL (WATCHLICEVENT) PGM (QSYS/QSCLICEV) CLS (QSYS/QSYSCLS25) MAXACT (*NOMAX) POOLID (1) << End of change <twb##" />

<TWB--UTF8--TWB>

### ระบบย่อยที่มีมาพร้อมกับระบบ

subsystem configuration ที่สมบูรณ์สองชุดที่จัดส่งมาโดย IBM สามารถนำมาใช้ได้โดยไม่ต้องทำการเปลี่ยนแปลงใดๆ .

configuration ที่ระบบใช้เมื่อมีการเริ่มต้นใช้งานนั้นจะถูกควบคุมไว้โดยค่า controlling subsystem description system (QCTLSBSD). The default configuration จะประกอบด้วย“Subsystem description” ในหน้า 59 ต่อไปนี้:

#### Qbase (controlling subsystem)

Qcmn  
  
Qserver  
Qspl  
  
Qsyswrk  
  
Qusrwrk

Qbase จะสนับสนุนงานโต้ตอบ, แบตช์ และสื่อสาร. และ Qbase จะมีงาน autostart, ซึ่งจะสตาร์ทระบบย่อย Qusrwrk, Qserver, และ Qspl โดยอัตโนมัติ.  
 นี้คือระบบย่อยที่ใช้ควบคุมงานด้านการสื่อสาร, เว้นแต่งานสื่อสารในโปรโตคอล TCP/IP. งานสื่อสารเหล่านี้มีความสำคัญต่อโปรโตคอลการสื่อสารต่างๆที่มีอยู่ในระบบ OS/400.  
 เป็นระบบย่อยของไฟล์เซิร์ฟเวอร์. ระบบย่อยนี้มีหน้าที่สนับสนุนงาน reader และงาน writer.  
 นี้คือระบบย่อยของงานระบบ. ซึ่งประกอบด้วยงานที่สนับสนุนฟังก์ชันระบบทั้งหลายที่จะสตาร์ทขึ้นโดยอัตโนมัติในการเริ่มเปิดระบบและเมื่อระบบได้ออกจากสภาวะการจำกัดการทำงาน.  
 เป็นระบบย่อยของงานผู้ใช้. ซึ่งจะประกอบด้วยงานที่ถูกเรียกทำงานโดยเซิร์ฟเวอร์เพื่อทำงานแทนให้กับผู้ใช้.

configuration อื่นๆ, ซึ่งจัดส่งมาโดย IBM , จะประกอบด้วย subsystem descriptions ต่อไปนี้:

#### Qctl (controlling subsystem)

Qinter

Qbatch

Qcmn

Qserver

Qspl

Qsyswrk

Qusrwrk

Qctl จะมีงาน autostart, ที่จะสตาร์ทระบบย่อย Qinter, Qbatch, Qcmn, Qusrwrk, Qserver และ Qspl ให้โดยอัตโนมัติ. ระบบย่อยนี้ใช้ควบคุมงานแบบโต้ตอบ, ยกเว้นในส่วนของคนโซล. นี่คืระบบย่อยที่ใช้ควบคุมงานแบ็ตซ์. นี่คืระบบย่อยที่ใช้ควบคุมงานด้านการสื่อสาร, เว้นแต่งานสื่อสารในโปรโตคอล TCP/IP. งานสื่อสารเหล่านี้มีความสำคัญต่อโปรโตคอลการสื่อสารต่างๆที่มีอยู่ในระบบ OS/400. เป็นระบบย่อยของไฟล์เซิร์ฟเวอร์. ระบบย่อยนี้มีหน้าที่สนับสนุนงาน reader และงาน writer. นี่คืระบบย่อยของงานระบบ, ซึ่งประกอบด้วยงานที่สนับสนุนฟังก์ชันระบบทั้งหลายที่จะสตาร์ทขึ้นโดยอัตโนมัติในการเริ่มเปิดระบบและเมื่อระบบได้ออกจากสภาวะการจำกัดการทำงาน. เป็นระบบย่อยของงานผู้ใช้, ซึ่งจะประกอบด้วยงานที่ถูกเรียกทำงานโดยเซิร์ฟเวอร์เพื่อทำงานแทนให้กับผู้ใช้.


Qbase configuration จะช่วยให้รันฟังก์ชันเดียวกันกับที่คุณ รันกับ Qctl configuration และช่วยจัดการให้ง่ายขึ้นอีกด้วย เพราะประกอบด้วยระบบย่อยต่างๆ ในจำนวนที่น้อยกว่า.

Qctl default configuration จะช่วยให้มีการควบคุมส่วนย่อยในการทำงานของระบบ โดยแตกย่อย activity ระบบออกเป็นระบบย่อยต่างๆ ทั้งนี้ ขึ้นอยู่กับประเภทของ activity . ตัวอย่างเช่น, ถ้าคุณต้องการรันงานแบ็ตซ์ในช่วงสุดสัปดาห์หรือข้ามคืน แต่ไม่ต้องการให้ใคร sign on เข้ามาได้ (ยกเว้นที่คอนโซล), คุณสามารถทำได้อย่างง่ายดายด้วย Qctl configuration โดยเพียงแต่ปิดการทำงานของระบบย่อย Qinter เท่านั้น.

หากคุณกำลังคิดที่จะสร้าง subsystem configuration ของคุณขึ้นเอง, ก็สามารถทำได้อย่างง่ายดาย เพียงใช้ Qctl configuration เป็น จุดเริ่มต้น ซึ่งจะง่ายกว่าการใช้ Qbase configuration.

### ระบบย่อยที่ผู้ใช้กำหนดเอง

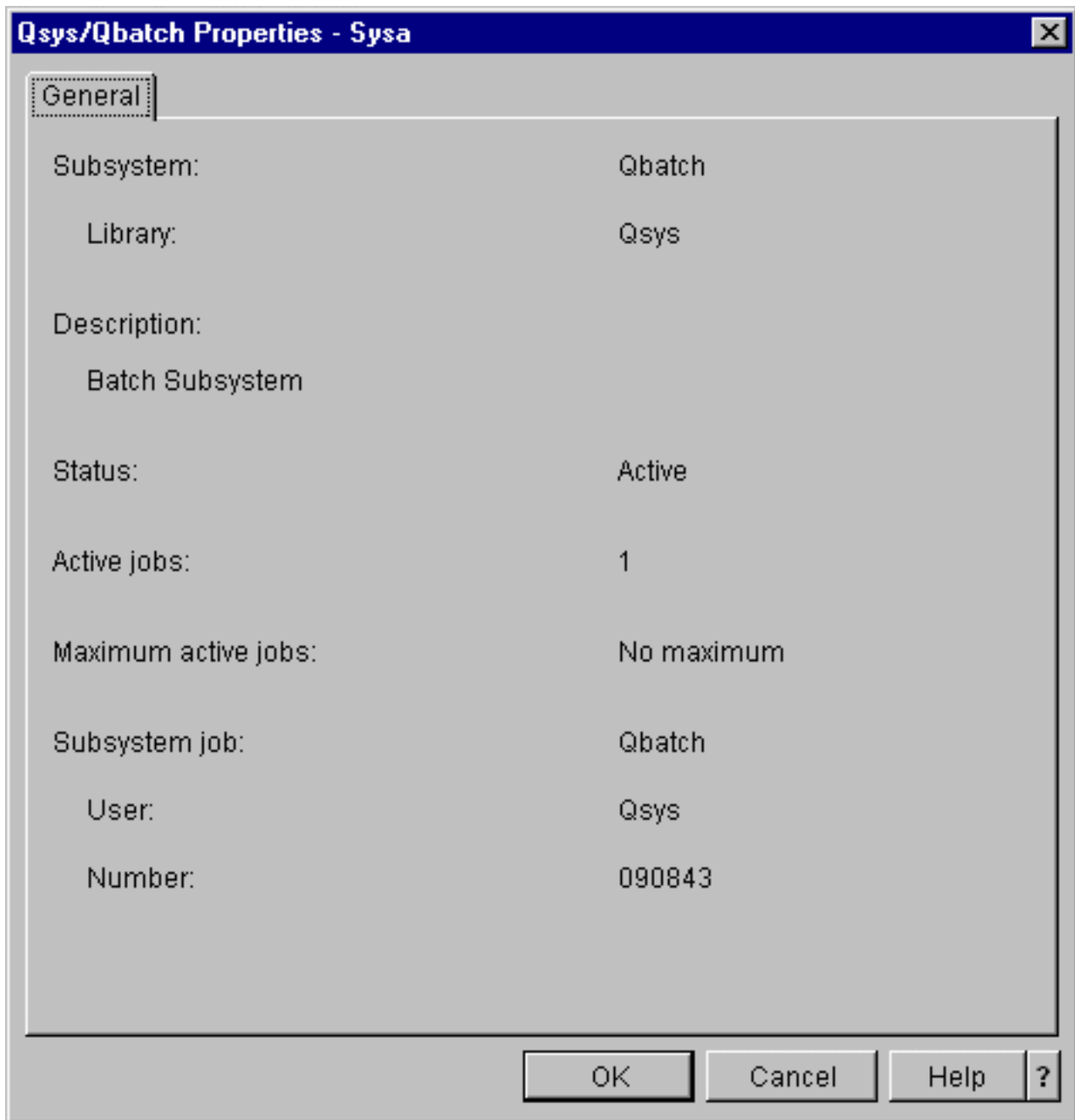
IBM ได้จัดเตรียม “ระบบย่อยที่มีมาพร้อมกัระบบ” ในหน้า 70. คุณสามารถสร้าง subsystem description ของคุณได้เอง. คุณสามารถตัดลอก subsystem description ที่มีอยู่และทำการเปลี่ยนแปลง, หรือคุณสามารถสร้างรายละเอียดใหม่ทั้งหมด.

สำหรับรายละเอียดให้ดูที่ การสร้าง subsystem description ในบทที่ 4 ของคู่มือ V4R5 Work Management  (ประมาณ 2720 KB or 573 หน้า).

## คุณสมบัติของระบบย่อย

ระบบย่อยจะมีแอตทริบิวต์, หรือคุณสมบัติ. คุณสมบัติเหล่านี้จะบอกถึงข้อมูล เกี่ยวกับสถานะปัจจุบันของระบบย่อย, หรือค่าที่ได้ระบุไว้ใน“Subsystem description” ในหน้า 59. เมื่อใช้ iSeries Navigator, จะเห็นคุณสมบัติต่อไปนี้กับระบบย่อยแอ็คทีฟ:

ระบบย่อย	ชื่อของระบบย่อย, พร้อมกับไลบรารีที่เก็บ subsystem description.
Description	description ของระบบย่อย.
สถานะ	สถานะปัจจุบันของระบบย่อย. คำอธิบายจะมีรายละเอียดของสถานะต่างๆ ให้ดู.
งานแอ็คทีฟ	จำนวนของงานที่แอ็คทีฟอยู่ในระบบย่อย, ไม่ว่าจะรันอยู่หรือรอที่จะรัน. จำนวนงานดังกล่าวไม่รวมงานระบบย่อยด้วย.
จำนวนงานแอ็คทีฟสูงสุด	จำนวนงานสูงสุดที่สามารถแอ็คทีฟได้ในระบบย่อย, ไม่ว่าจะรันอยู่หรือรอที่จะรัน.
งานระบบย่อย	ชื่อของงานระบบย่อย, รวมถึงผู้ใช้และจำนวน.



ในการดูคุณสมบัติของระบบย่อย, ให้ทำตามขั้นตอนดังนี้:

1. จาก iSeries Navigator, ให้ขยาย My Connections → *server-name* → Work Management → Subsystems → Active Subsystems.
2. คลิกขวาไปบนระบบย่อยที่คุณต้องการดู, แล้วเลือก Properties.

### วงรอบชีวิตของระบบย่อย

ช่วงชีวิตของระบบย่อยเริ่มต้นเมื่อถูกสตาร์ทขึ้น, และจบลงเมื่อ ระบบย่อยหยุดทำงาน. ในระหว่างนี้, งานจะถูกประมวลผลในระบบย่อย. สำหรับรายละเอียดเพิ่มเติม, ให้ดูหัวข้อต่อไปนี้:

- “การเริ่มใช้งานระบบย่อย” ในหน้า 25

- “สิ่งที่เกิดขึ้นเมื่อระบบย่อยเริ่มทำงาน”
- “หยุดระบบย่อย” ในหน้า 25

**สิ่งที่เกิดขึ้นเมื่อระบบย่อยเริ่มทำงาน:** เมื่อระบบย่อยเริ่มทำงาน, ระบบจะจัดสรรสิ่งต่างๆ ให้ระบบย่อย และ จะเริ่มงาน autostart และ prestart ให้ก่อนที่ระบบย่อยจะพร้อมที่จะทำงาน. “Subsystem description” ในหน้า 59 จะถูกใช้ในการระบุวิธีที่สิ่งต่างๆ จะถูกจัดสรร.

รายการต่อไปนี้จะใช้แทนลำดับของเหตุการณ์ที่เกิดขึ้นเมื่อระบบย่อยเริ่มทำงาน:

1. การร้องขอให้เริ่มระบบย่อยถูกส่งออกมา.

2. พูลหน่วยความจำถูกจัดสรรไว้.

หน่วยความจำจะถูกจัดสรรให้กับพูลที่ได้กำหนดไว้ใน subsystem description. หน่วย ความจำที่ถูกจัดสรรให้กับพูลที่กำหนดไว้แต่ละอันถูกดึงมาจากพูลหน่วยความจำหลัก. ระบบจะไม่จัดสรรหน่วยความจำให้กับพูลหากจำนวนหน่วยความจำที่เหลือให้กับพูลหน่วยความจำหลักมีน้อยกว่าขนาดต่ำสุดที่ได้กำหนดไว้โดยค่ากำหนดระบบ base memory pool minimum size (Qbaspool). หากระบบไม่สามารถจัดสรรหน่วยความจำได้ตามที่มีการร้องขอเข้ามา, ระบบจะ จัดสรรหน่วยความจำให้มากที่สุดเท่าที่จะทำได้และจัดสรรให้เพิ่มเติมเมื่อมี หน่วยความจำเหลือเพิ่มขึ้น.

สำหรับข้อมูลเพิ่มเติมให้ดูที่ การจัดสรรพูล ในบทที่ 4 ของคู่มือ V4R5 Work Management



(ประมาณ 2720 KB หรือ 573 หน้า).

3. จอภาพถูกจัดสรรไว้.

- หากมี workstation entry และอุปกรณ์ถูก vary on และไม่ได้ถูกจัดสรรให้กับระบบย่อยอื่นใด, ระบบย่อยจะสามารถจัดสรรอุปกรณ์นั้นและจะแสดงผล Sign-On display.

- หากอุปกรณ์นั้นถูก vary on และได้ถูกจัดสรรให้กับระบบย่อยอื่น และอยู่ที่ Sign-On display (Sign-On display จะปรากฏก่อนที่ระบบย่อยที่สองจะเริ่มทำงาน ขึ้นมา), ระบบย่อยที่สองจะสามารถจัดสรรอุปกรณ์จากระบบย่อยแรกได้และจะแสดง Sign-On display.

- หากอุปกรณ์ไม่ได้ถูก vary on, ระบบย่อยจะไม่สามารถจัดสรรอุปกรณ์นั้นได้. system arbiter (Qsysarb) และงาน Qcmnarbxx จะถือสิทธิ์ไว้กับอุปกรณ์ที่ vary off เอาไว้ทั้งหมด.

สำหรับข้อมูลเพิ่มเติมให้ดูที่ การจัดสรรอุปกรณ์ Workstation ในบทที่ 4 ของคู่มือ V4R5 Work Management



(ประมาณ 2720 KB หรือ 573 หน้า).

4. อุปกรณ์สื่อสารถูกจัดสรร.

การร้องขอจะถูกส่งไปยังงานระบบ Qlus (LU service) ซึ่งดูแลเรื่องการจัดสรรอุปกรณ์สื่อสารทั้งหมด.

สำหรับข้อมูลเพิ่มเติมให้ดูที่ อุปกรณ์การสื่อสาร และ โหมดการจัดสรร ในคู่มือ V4R5 Work Management



(ประมาณ 2720 KB หรือ 573 หน้า).

5. คิวงานถูกจัดสรร.

ระบบย่อยจะไม่สามารถรับการจัดสรรคิวงานได้หากคิวงานได้ถูกจัดสรรให้กับระบบย่อย ที่แฉีกที่ฟ่อนไปแล้ว.

6. “งาน Prestart” ในหน้า 39 เริ่มต้นทำงาน.



7. “งานแบบ autostart” ในหน้า 37 เริ่มต้นทำงาน.
8. สถานะแวดล้อมพร้อมที่จะทำงาน.

## พูลหน่วยความจำ

พูลหน่วยความจำเป็นส่วนลอจิคัลของหน่วยความจำหลักหรือหน่วยเก็บข้อมูลที่กันไว้สำหรับการประมวลผลงานหรือกลุ่มของงาน. บนเซิร์ฟเวอร์ iSeries<sup>™</sup> server, หน่วยความจำหลักทั้งหมดสามารถแบ่งออกเป็นการจัดสรรแบบลอจิคัลที่เรียกว่า พูลหน่วยความจำ. โดยปกติแล้ว ระบบจะจัดการพูลหน่วยความจำเอง. ระบบจะจัดการการโอนถ่ายข้อมูลและโปรแกรมเข้าไว้ในพูลหน่วยความจำเมื่อจำเป็น.

คุณยังสามารถควบคุมปริมาณของงานที่จะทำในระบบย่อยโดยการควบคุมจำนวนและขนาดของพูลหน่วยความจำ. ขนาดของพูลหน่วยความจำที่ใหญ่ขึ้นในระบบย่อยจะหมายถึงงานที่มากขึ้นที่จะทำในระบบย่อย.

**หมายเหตุ:** แม้ว่าการปรับแต่งและการจัดระบบของคุณจะช่วยให้เกิดประสิทธิภาพในการไหลของงานผ่านเซิร์ฟเวอร์ iSeries, แต่สิ่งนี้ไม่ได้ช่วยแก้ไขเรื่องฮาร์ดแวร์ที่ล้าสมัยที่ไม่เพียงพอได้. ดังนั้นควรพิจารณาเรื่องการอัปเดตฮาร์ดแวร์ หากความต้องการด้านเวิร์กโหลดมีความสำคัญมาก.

พูลหน่วยความจำที่งานของผู้ใช้ได้หน่วยความจำมาจะเป็นพูลอันเดียวกันที่จำกัด activity level ของงานนั้น. งานของระบบ (เช่น Scpf, Qsysarb, และ Qlus) จะได้หน่วยความจำจากพูลพื้นฐาน แต่จะใช้ activity level ของพูลตัวเครื่อง. การมอนิเตอร์ระบบย่อยจะได้รับหน่วยความจำจาก subsystem description pool แต่ไม่ได้จาก activity level. สิ่งนี้จะช่วยให้มอนิเตอร์ระบบย่อยสามารถทำงานได้ตลอดโดยไม่ต้องสนใจต่อการตั้งค่าของ activity level.

**หมายเหตุ:** สามารถเรียกใช้ APIs, เช่น Retrieve System Status (QWCRSSTS), เพื่อรับข้อมูลบนพูลหน่วยความจำ. สำหรับรายละเอียดเพิ่มเติม, โปรดดูที่ Application programming interfaces (APIs)

กรุณาดูที่หัวข้อต่อไปนี้เป็นสำหรับรายละเอียดเพิ่มเติมเกี่ยวกับพูลหน่วยความจำ:

- “activity level ของพูลหน่วยความจำ”
- “ประเภทของพูลหน่วยความจำ” ในหน้า 76

### activity level ของพูลหน่วยความจำ

activity level ของพูลหน่วยความจำจะช่วยให้มีการใช้งานระบบอย่างมีประสิทธิภาพ โดยการจำกัดจำนวน thread ที่จะแอ็คทีฟได้ในเวลาเดียวกัน ในพูลหน่วยความจำ. ระบบ จะจัดการดูแลควบคุมระดับดังกล่าว. บ่อยครั้งในระหว่างการประมวลผล ใน thread, โปรแกรมจะรอหรือสละระบบหรือการตอบกลับจากผู้ใช้งาน เวิร์กสเตชัน. ในระหว่างการรอนั้น, thread จะยกเลิกการใช้งานของ activity level ของพูลหน่วยความจำเพื่อให้ thread อื่นที่พร้อมจะประมวลผลได้สามารถเข้ามาทำงานแทนที่ไปก่อน.

ระดับของ activity (activity level) ของพูลหน่วยความจำ คือ จำนวนของ thread ที่สามารถใช้ CPU ในสถานะแอ็คทีฟได้ในเวลาเดียวกันในพูลหน่วยความจำ. ระบบ จะจัดการดูแลควบคุมระดับดังกล่าว. บ่อยครั้งในระหว่างการประมวลผล ใน thread, โปรแกรมจะรอหรือสละระบบหรือการตอบกลับจากผู้ใช้งาน เวิร์กสเตชัน. ในระหว่างการรอนั้น, thread จะยกเลิกการใช้งานของ activity level ของพูลหน่วยความจำเพื่อให้ thread อื่นที่พร้อมจะประมวลผลได้สามารถเข้ามาทำงานแทนที่ไปก่อน.

เมื่อมี thread สตาร์ทมากขึ้นเกินกว่าจะรันได้ในเวลาเดียวกัน เนื่องจากการควบคุม activity level เอาไว้, thread ที่เกินมาจะต้องรอเพื่อใช้หน่วยประมวลผล (ซึ่งปกติการรอเช่นนี้จะใช้เวลาไม่นาน). activity level ของพูลหน่วยความจำจะให้คุณจำกัดระดับการแย่งใช้ หน่วยความจำหลักในพูลหน่วยความจำทั้งหลายในระบบย่อยของคุณ.

จำนวน thread ที่กำลังรันอยู่ (หรือ active thread) ก็คือ จำนวน thread ที่มีสิทธิในการแย่งใช้ตัวประมวลผล และมีส่วนใน activity level ของพูลหน่วยความจำ. ในลักษณะเช่นนี้เอง, active thread จะไม่รวมถึง thread ที่กำลังรออินพุต, รอข้อความ, รอได้รับ

การจัดสรรอุปกรณ์เพื่อใช้งาน, หรือรอไฟล์เพื่อเปิดใช้. active thread จะไม่รวมถึง thread ที่ไม่มีสิทธิ (คือ thread ที่พร้อมจะรัน แต่ activity level ของพูลหน่วยความจำได้ขึ้นไปถึงจุดสูงสุดแล้ว).

### activity level ทำงานอย่างไร


thread มากกว่าหนึ่งอันจะแก็คที่ฟได้ในเวลาเดียวกันในพูลหน่วยความจำ เนื่องจากการประมวลผล thread นั้นจะมีการอินเตอร์รัปต์เพียงชั่วขณะ เท่านั้น ในขณะที่ข้อมูลที่ต้องการกำลังถูกเรียกออกมาจากหน่วยความจำสำรอง. ในระหว่างช่วง delay นี้, ซึ่งจะสั้นมาก, thread ตัวอื่นก็สามารถรันได้. เมื่อใช้ activity level แล้ว, ตัวเครื่องจะสามารถประมวลผล thread จำนวนมากในพูลหน่วยความจำได้ และในเวลาเดียวกัน, ก็สามารถคงระดับการแย่งใช้งาน เอาไว้ตามการจำกัดที่ได้ระบุให้ได้อีกด้วย.

### activity level สูงสุด

เมื่อถึงระดับสูงสุดของ activity level ของพูลหน่วยความจำแล้ว, thread เพิ่มเติมที่ต้องการใช้พูลหน่วยความจำจะถูกตั้งให้อยู่ในสถานะ ไม่มีสิทธิและต้องรอนกว่าจำนวน active thread ในพูลหน่วยความจำจะลดลง มาต่ำกว่า activity level สูงสุดเสียก่อนหรือรอนกว่า thread จะใช้เวลาของการแบ่งเวลาจนหมด. เมื่อใดก็ตามที่ thread ไม่ได้ใช้พูลหน่วยความจำแล้ว, thread อื่นๆ ที่ยังไม่แก็คที่ฟก็จะมีสิทธิที่จะรันตามระดับความสำคัญของตัวมัน. ตัวอย่างเช่น, หาก thread ที่รันอยู่กำลัง รอกการตอบกลับมาจากเวิร์กสเตชัน, thread นั้นก็จะปล่อย activity level ทิ้งไปทำให้ activity level ลดลงมาไม่ถึงระดับสูงสุด.

### การกำหนด activity level ของพูลหน่วยความจำ

การกำหนดพูลหน่วยความจำและ activity level ให้ถูกต้องนั้นขึ้นอยู่กับ ขนาดของพูลหน่วยความจำ, จำนวนของ CPU, จำนวนแขนของ disk และคุณลักษณะของแก็คที่ฟลิเคชันด้วย. ให้อูที่ การปรับแต่งการทำงาน ในบทที่ 14

ของคู่มือ V4R5 Work Management  (ประมาณ 2720 กิโลไบต์ หรือ 573 หน้า) สำหรับคำอธิบายอย่างละเอียดในการตั้งค่า activity level ที่ถูกต้อง.

ให้อู การควบคุมระดับของ activity ระบบ ในบทที่ 4 ของคู่มือ V4R5 Work Management  สำหรับข้อมูลเพิ่มเติม.

### ประเภทของพูลหน่วยความจำ

พูลหน่วยความจำเป็นส่วนหนึ่งของหน่วยความจำหลักหรือสำรอง. บนเซิร์ฟเวอร์ iSeries, หน่วยความจำหลักทั้งหมดสามารถแบ่งออกเป็นการจัดสรรแบบลोजิคัลที่เรียกว่า พูล หน่วยความจำ. ระบบจะมีพูลหน่วยความจำอยู่ 2 ประเภท, คือ แบบส่วนตัวและแบบใช้ร่วมกัน. พูลหน่วย ความจำสูงสุดทั้งหมด 64 พูลสามารถแก็คที่ฟได้ในเวลาเดียวกัน ไม่ว่าจะเป็นแบบส่วนตัว หรือใช้ร่วมกันก็ตาม.

#### พูลหน่วยความจำแบบส่วนตัว

เป็นพูลที่ถูกกำหนดไว้ด้วยชื่อระบบย่อยใน iSeries Navigator, เพื่อให้ระบบย่อยเดียวสามารถรันงานได้. พูลส่วนตัวจะเป็นพูลของหน่วยความจำหลักที่ไม่สามารถแบ่งใช้กับระบบย่อยอื่นๆ ได้. พูลส่วนตัวประกอบด้วยจำนวนหน่วยความจำที่กำหนดไว้เพื่อให้ใช้กับระบบย่อยหนึ่งเท่านั้น. คุณสามารถมีพูลหน่วยความจำได้มากถึง 62 พูลเพื่อจัดสรรให้ใช้งานกับระบบย่อยที่แก็คที่ฟได้. พูลส่วนตัวไม่จำเป็นต้องมีขนาดใหญ่เพื่อให้เข้ากับโปรแกรมของคุณ.

## พูลหน่วยความจำร่วม

หน่วยความจำร่วมคือพูลที่ระบบย่อยหลายๆระบบสามารถมารันงานร่วมกัน. การใช้พูลหน่วยความจำร่วม เช่นนี้จะช่วยให้ระบบสามารถกระจายงานแบบเดียวกันไปสู่ระบบย่อยหลายๆตัวได้, โดยยังคงให้งานเหล่านี้รันอยู่ในพูลหน่วยความจำอันเดียวกันได้. คุณสามารถกำหนดพูลหน่วยความจำ 63 พูลจากทั้งหมด 64 พูลที่ได้กำหนดไว้บนระบบเพื่อใช้งานตั้งแต่เมื่อมีการสร้าง subsystem description. พูลตัวเครื่องจะถูกสงวนไว้เพื่อการใช้งานของระบบเท่านั้น. พูลแบบใช้ร่วมกัน มีทั้งแบบพิเศษหรือแบบทั่วไป; “พูลหน่วยความจำของเครื่อง”และ “พูลหน่วยความจำพื้นฐาน”จัดว่าเป็นพูลแบบใช้ร่วมกันแบบพิเศษ, และพูลแบบใช้ร่วมกันอื่นๆทั้งหมดถือว่าเป็น “พูลแบบใช้งานร่วมกันทั่วไป”.

**พูลหน่วยความจำพื้นฐาน:** พูลหน่วยความจำพื้นฐาน, ที่ได้กำหนดไว้เป็น Base ใน iSeries Navigator, จะประกอบด้วยหน่วยความจำหลักทั้งหมดที่ยังไม่กำหนด นั่นคือ, หน่วยความจำหลักทั้งหมดที่พูลหน่วยความจำอื่นๆ ไม่ต้องการใช้. พูลพื้นฐานจะประกอบด้วยหน่วยความจำที่สามารถแบ่งใช้งานกับระบบย่อยต่างๆ. พูลหน่วยความจำหลักจะถูกใช้สำหรับงานแบบแบตช์และฟังก์ชันระบบต่างๆไป.

ขนาดต่ำสุดและ activity level สำหรับพูลหน่วยความจำหลักจะถูกควบคุม โดยค่ากำหนดของระบบ. สำหรับรายละเอียดให้ดูค่ากำหนดของระบบต่อไปนี้:

- QBASACTLVL (activity level ของพูลหน่วยความจำหลัก)
- QBASPOOL (ขนาดต่ำสุดของ พูลหน่วยความจำหลัก)

**พูลหน่วยความจำของเครื่อง:** พูลหน่วยความจำเครื่อง, ที่กำหนดไว้เป็น Machine ใน iSeries Navigator, จะถูกใช้สำหรับเครื่องที่มีการแบ่งใช้สูงและสำหรับโปรแกรมระบบ. machine memory pool จะจัดเตรียมหน่วยความจำให้กับงานที่ระบบต้องรันซึ่งเป็นงานที่คุณไม่ต้องไปจัดการอะไร. ขนาดของพูลหน่วยความจำชนิดนี้จะถูกกำหนดไว้ใน machine memory pool size system value (QMCHPOOL). พูลหน่วยความจำชนิดนี้จะไม่มีการใช้งานของผู้ใช้ใดๆ มารัน.

**พูลแบบใช้งานร่วมกันทั่วไป:** พูลแบบใช้งานร่วมกันทั่วไป, ซึ่งได้ระบุไว้เป็น Interactive, Spool, และ Shared 1 – Shared 60 ใน iSeries Navigator, เป็นพูลของหน่วยความจำหลักที่ระบบย่อยต่างๆ สามารถใช้งาน ร่วมกันได้ในเวลาเดียวกัน.

พูลแบบ ใช้งานร่วม	รายละเอียด
Interactive	พูลหน่วยความจำ ที่ใช้กับงานแบบโต้ตอบ
Spool	พูลหน่วยความจำ ที่ใช้สำหรับการพิมพ์
Shared 1 – Shared 60	พูลหน่วยความจำ สำหรับการใช้งานเฉพาะของคุณ

## เอาต์พุตคิว

เอาต์พุตคิวคือส่วนที่ไฟล์ปริ้นเตอร์เอาต์พุต (หรือเรียกว่าสพูลไฟล์) รอการถูกประมวลผลและถูกส่งไปยังปริ้นเตอร์. ปริ้นเตอร์เอาต์พุตจะถูกสร้างขึ้นทั้งโดยระบบหรือโดยผู้ใช้โดยใช้ไฟล์การพิมพ์. ไฟล์การพิมพ์ คล้ายกับเท็มเพลตหรือแนวทางการกำหนดค่าดีฟอลต์สำหรับแอ็ททริบิวต์ของปริ้นเตอร์เอาต์พุต. และเป็นการเริ่มต้นของช่วงชีวิตปริ้นเตอร์เอาต์พุต.

ไฟล์การพิมพ์ประกอบด้วยเอาต์พุตคิว (OUTQ) และแอ็ททริบิวต์อุปกรณ์การพิมพ์ (DEV), ซึ่งระบุวิธีการส่งงานปริ้นเตอร์เอาต์พุต. ค่ากำหนด ดีฟอลต์มักจะเป็น \*JOB, หมายความว่าแอ็ททริบิวต์งานของเอาต์พุตคิวและอุปกรณ์การพิมพ์จะกำหนด

วิธีการสั่งงาน. แอ็ททริบิวต์งานของเอาต์พุตคิวและค่าอุปกรณ์การพิมพ์มาจากข้อมูลที่ได้รับจากงานที่ถูกสร้างขึ้น. ข้อมูลนี้มาจากข้อมูลที่ได้จากโปรไฟล์ผู้ใช้ซึ่งงานกำลังรันภายใต้, จาก job description, รายละเอียดอุปกรณ์เวิร์กสเตชัน, และค่าระบบเครื่องพิมพ์ดีฟอลต์ (QPRTEDEV).

เมื่อเอาต์พุตเครื่องพิมพ์พร้อมที่จะถูกสร้าง, ระบบจะตรวจสอบไฟล์การพิมพ์และแอ็ททริบิวต์งาน (ในลำดับนี้) เพื่อดูว่าเอาต์พุตคิวใดจะประมวลผลพริเตอร์เอาต์พุตและอุปกรณ์การพิมพ์ใดที่ระบบจะใช้งาน. คุณสามารถเปลี่ยนการประมวลผลของเอาต์พุตคิว (OUTQ) และอุปกรณ์การพิมพ์ (DEV) ขณะทำงานถูกส่งหรือขณะรันงานเพื่อข้ามการประมวลผลแบบขยาย. ยกตัวอย่าง, ผู้ใช้สามารถตั้งค่าเอาต์พุตคิวของไฟล์การพิมพ์ให้เป็นคิวเฉพาะได้และตั้งค่าอุปกรณ์การพิมพ์กับเครื่องพิมพ์ที่ต้องการในไฟล์การพิมพ์เมื่อเริ่มต้นทำงานเพื่อใช้งานการเปลี่ยนแปลงในทันที. ในการทำเช่นนี้, พริเตอร์เอาต์พุตไม่ต้องเข้าไปยังแอ็ททริบิวต์งานเพื่อค้นหาเอาต์พุตคิวและ อุปกรณ์การพิมพ์ที่จะใช้งาน. หากไม่พบเอาต์พุตคิวที่ระบุ, พริเตอร์เอาต์พุตจะถูกส่งให้ไปที่ QGPL/QPRINT. สำหรับข้อมูลเพิ่มเติมเกี่ยวกับวิธีสร้างพริเตอร์เอาต์พุต, ให้อูที่บทที่ 1

ของคู่มือ การทำโปรแกรมมิ่งอุปกรณ์การพิมพ์. 

**ไฟล์พริเตอร์เอาต์พุต** คือไฟล์ที่มีข้อมูลซึ่งกำลังรอที่จะพิมพ์หรือถูกประมวลผล. ไฟล์พริเตอร์เอาต์พุตมีแอ็ททริบิวต์ที่สำคัญซึ่งกำหนดตำแหน่งของพริเตอร์เอาต์พุตบนคิวซึ่งสัมพันธ์กับพริเตอร์เอาต์พุตอื่น. ตำแหน่งจะถูกกำหนดโดยระดับความสำคัญ, สถานะ, และแอ็ททริบิวต์กำหนดการ.

#### เอาต์พุตคิว

เอาต์พุตคิว คืออ็อบเจกต์ที่ประกอบด้วยลิสต์ของไฟล์พริเตอร์เอาต์พุตซึ่งจะถูกเขียนสำหรับอุปกรณ์เอาต์พุต. เอาต์พุตคิวมี“แอ็ททริบิวต์ของเอาต์พุตคิว” ในหน้า 79 ที่สำคัญซึ่งกำหนดลำดับที่พริเตอร์เอาต์พุตถูกประมวลผลและลิสต์ในการใช้งานที่ใช้ทำการเปลี่ยนแปลงกับไฟล์พริเตอร์เอาต์พุต.

#### ระดับความสำคัญ

พริเตอร์เอาต์พุตที่กำลังรอการประมวลผลจะถูกย้ายไปเอาต์พุตคิวตามระดับความสำคัญของเอาต์พุตคิว (ตั้งแต่ 1-9 ที่ซึ่ง 1 คือระดับความสำคัญที่สูงที่สุด).

#### สถานะ

“สถานะของพริเตอร์เอาต์พุต” ในหน้า 80 ปัจจุบัน. คุณสามารถดูสถานะนี้จากหน้าที่มีข้อมูลทั่วไปที่บอกถึงคุณสมบัติของเอาต์พุต.

#### กำหนดการ

แอ็ททริบิวต์กำหนดการจะบอกว่าเมื่อใดที่ไฟล์ควรเริ่มการพิมพ์แบบฟิสิกส์ข้อมูลเอาต์พุต.

#### ในทันที

พิมพ์ในทันที, แม้ว่าพริเตอร์เอาต์พุตจะยังไม่ถูกปิด.

#### สิ้นสุดไฟล์ (ดีฟอลต์)

การพิมพ์เริ่มทำงานทันทีที่ไฟล์พริเตอร์เอาต์พุตถูกปิด.

#### สิ้นสุดการทำงาน

การพิมพ์จะเริ่มต้นทำงานเมื่องานสิ้นสุด.

เมื่อไฟล์พริเตอร์เอาต์พุตพร้อมพิมพ์, งานแบบ writer, งานที่ประมวลผลพริเตอร์เอาต์พุตจากเอาต์พุตคิวไปยังอุปกรณ์การพิมพ์, ให้ใช้ข้อมูลจากไฟล์พริเตอร์เอาต์พุตและส่งข้อมูลไปยังเครื่องพิมพ์ที่กำหนด.

## แอ็ตทริบิวต์ของเอาต์พุตคิว

เอาต์พุตคิวจะควบคุมวิธีที่ไฟล์พรินเตอร์เอาต์พุต (มักเรียกว่าสพูลไฟล์) ถูกประมวลผลและควบคุมผู้มีสิทธิ์ปฏิบัติการบนเอาต์พุตคิวและพรินเตอร์เอาต์พุตที่เกี่ยวข้อง.

“ลำดับของไฟล์”แอ็ตทริบิวต์จะกำหนดว่าพรินเตอร์เอาต์พุตจะปล่อยให้เอาต์พุตคิวประมวลผลอย่างไร. มีสองวิธีในการตั้งค่าเอาต์พุตคิว, ตั้งค่าโดยหมายเลขงานหรือด้วยกฎเข้าก่อนออกก่อน, (FIFO).

เนื่องจากข้อมูลส่วนใหญ่ที่คุณพิมพ์บนระบบ iSeries ถูกสร้างไว้เป็นพรินเตอร์เอาต์พุต ดังนั้นความปลอดภัยจึงเป็นสิ่งจำเป็นเพื่อป้องกันผู้ใช้ที่ไม่มีสิทธิ์ในการเข้าไปเรียกใช้สิ่งพิมพ์ที่เป็นความลับหรือสำคัญ. สิทธิการตรวจสอบ, สิทธิในข้อมูล, การควบคุมโอเปอเรเตอร์, การควบคุมสพูล, หรือการเป็นเจ้าของอนุญาตให้คุณสามารถแอ็กเซสและทำการเปลี่ยนแปลงเอาต์พุตคิวหรือไฟล์พรินเตอร์เอาต์พุต. คุณต้องมีสิทธิ์ใช้งานดังต่อไปนี้สิทธิ์ใดสิทธิ์หนึ่งเพื่อดำเนินการใดๆ กับเอาต์พุตคิวหรือพรินเตอร์เอาต์พุต.

**สิทธิในการตรวจสอบ.** คุณต้องเป็นเจ้าของคิวหรือมีสิทธิ์ในข้อมูล.

**แสดงผลข้อมูล.** เมื่อสิทธิ์ใช้งานนี้ถูกกำหนดเป็น \*YES, สิทธิ์นั้นจะให้คุณสามารถดำเนินการต่างๆ เช่น การดู, การย้าย, การส่งเอาต์พุตไปยังระบบอื่น, และการทำสำเนาพรินเตอร์เอาต์พุต.

**การควบคุมโอเปอเรเตอร์.** หากแอ็ตทริบิวต์นี้ถูกกำหนดเป็น \*YES, ผู้ใช้ที่มีสิทธิ์พิเศษ \*JOBCTL จะได้รับสิทธิ์ให้ดำเนินการต่างๆ เช่นการพัก, การปล่อย, และการลบพรินเตอร์เอาต์พุตจากเอาต์พุตคิว. และอนุญาตให้มีการดำเนินการอื่นๆกับ พรินเตอร์เอาต์พุต, เอาต์พุตคิว, และ writers โดยจะมีการลงบันทึกไว้ใน คู่มือด้านความปลอดภัยของ iSeries.



**การควบคุมสพูล.** อนุญาตให้ผู้ใช้ดำเนินการต่างๆ บนพรินเตอร์เอาต์พุต. ผู้ใช้ต้องมีสิทธิ์ใช้งาน \*EXECUTE กับไลบรารีที่มีเอาต์พุตคิวอยู่เพื่อดำเนินการต่างๆ บนเอาต์พุตคิว.

**เจ้าของ.** อนุญาตให้ผู้ใช้ที่เป็นเจ้าของเอาต์พุตคิวเปลี่ยนหรือลบพรินเตอร์เอาต์พุต.

**หมายเหตุ:** สิทธิในการใช้งานซึ่งเป็นค่าดีฟอลต์สำหรับเอาต์พุตคิวคือสิทธิพับลิค \*USE. สิทธิในการใช้งาน Display Data ถูกกำหนดเป็น \*NO (หมายถึงไม่ใช่ทุกคนที่สามารถดูพรินเตอร์เอาต์พุตนี้ได้). สิทธิในการตรวจสอบคือ \*OWNER (ดังนั้นเจ้าของเอาต์พุตคิวจึงสามารถจัดการกับพรินเตอร์เอาต์พุตได้). การควบคุมโอเปอเรเตอร์ถูกกำหนดเป็น \*YES (หมายถึงผู้ใช้ที่มีสิทธิ์ \*JOBCTL สามารถพัก, ปล่อย, และลบพรินเตอร์เอาต์พุตได้).

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับสิทธิ์ใช้งานที่จำเป็นสำหรับการทำงานกับเอาต์พุตคิว, ให้อูที่ ภาคผนวก D ใน คู่มือด้านความ

ปลอดภัย 

**ลำดับของไฟล์:** ลำดับของไฟล์แอ็ตทริบิวต์จะกำหนดลำดับที่ไฟล์พรินเตอร์เอาต์พุต (หรือเรียกว่าสพูลไฟล์) ถูกวางไว้และถูกประมวลผลบนเอาต์พุตคิว. วิธีสองวิธีในการตั้งค่าเอาต์พุตคิวคือการตั้งค่าโดยใช้หมายเลขงาน และกฎเข้าก่อนออกก่อน (FIFO).

## หมายเลขงาน

queue entry สำหรับไฟล์พรินเตอร์เอาต์พุตจะถูกเรียงลำดับตามระดับความสำคัญโดยใช้หมายเลขงานของงานที่สร้างไฟล์พรินเตอร์เอาต์พุตนั้น.

## เข้าก่อนออกก่อน

ไฟล์พรินเตอร์เอาต์พุตใหม่ (หรือเรียกว่าสพูลไฟล์) ที่เข้าสู่คิวถูกวางไว้หลังไฟล์พรินเตอร์เอาต์พุตอื่นๆ ทั้งหมดที่มีระดับความสำคัญระดับเดียวกัน.

หมายเหตุ: คุณสามารถเปลี่ยนได้แต่ลำดับเอาต์พุตคิวของไฟล์แอสริบิวต์เมื่อไม่มีไฟล์พรินเตอร์เอาต์พุตอยู่บนคิว.

## สถานะของพรินเตอร์เอาต์พุต

สถานะของไฟล์พรินเตอร์เอาต์พุต (หรือเรียกว่าสพูลไฟล์) จะกำหนดว่าคุณจะดูสถานะในเอาต์พุตคิวได้ที่ไหน. สถานะต่อไปนี้ไล่มาจากด้านล่างของเอาต์พุตคิวไปจนถึงด้านบน.

### ยังคงสร้างอยู่

กำลังสร้างพรินเตอร์เอาต์พุต.

### พิมพ์และเก็บไว้

ข้อมูลในไฟล์พรินเตอร์เอาต์พุตถูกพิมพ์ออกมาแล้ว, แต่ถูกบันทึกไว้เพื่อใช้งานในภายหลัง.

### พักไว้

ไฟล์พรินเตอร์เอาต์พุตจะถูกพักไว้, เพื่อป้องกันไม่ให้ถูกประมวลผลโดยงาน writer.

### ยังไม่ได้กำหนดให้พิมพ์

การสร้างไฟล์พรินเตอร์เอาต์พุตเสร็จสมบูรณ์, แต่ไฟล์ไม่มีสิทธิ์ถูกส่งพิมพ์. จะเห็นไฟล์พรินเตอร์เอาต์พุตเมื่อแอสริบิวต์ของไฟล์พรินเตอร์เอาต์พุตถูกตั้งค่าเป็น \*JOBEND. นี่หมายความว่างานที่เป็นเจ้าของไฟล์พรินเตอร์เอาต์พุตต้องหยุดทำงานก่อนที่ไฟล์พรินเตอร์เอาต์พุตจะได้รับอนุญาตให้ประมวลผลโดยงานแบบ writer.

### เกินขอบเขตหน้า

ไฟล์ที่ใช้จำนวนหน้าเกินที่กำหนดไว้จะได้รับอนุญาตให้พิมพ์โดยงานแบบ writer. จะดูสถานะนี้ได้ก็ต่อเมื่อเอาต์พุตคิวแอสริบิวต์ต่องานแบบ writer.

### พร้อมใช้งาน

ไฟล์พรินเตอร์เอาต์พุตกำลังรอการประมวลผลโดยงานแบบ writer .

จะดูสถานะต่อไปนี้ได้เมื่อเอาต์พุตคิวแอสริบิวต์ต่องานแบบ writer (ถูกประมวลผลโดยงานแบบ writer) และจะดูได้ที่ด้านบนของเอาต์พุตคิว.

## การแปลงสำหรับพริ้นเตอร์

ไฟล์พริ้นเตอร์เอาต์พุตอยู่ในกระบวนการถูกแปลง (สร้างเรียบร้อยแล้ว) ไปเป็นอุปกรณ์การพิมพ์.

### กำลังพิมพ์

ข้อมูลของไฟล์พริ้นเตอร์เอาต์พุตกำลังถูกส่งไปยังอุปกรณ์การพิมพ์.

### ส่งไปยังเครื่องพิมพ์

ข้อมูลของไฟล์พริ้นเตอร์เอาต์พุตกำลังถูกพิมพ์ออกมา. ระบบปฏิบัติการกำลังรอการยืนยันว่าไฟล์พริ้นเตอร์เอาต์พุตเสร็จสิ้นการพิมพ์.

### กำลังถูกส่งไป

ไฟล์พริ้นเตอร์เอาต์พุตถูกถ่ายโอนจากระบบหนึ่งไปสู่ระบบหนึ่ง.

### ข้อความการรอ

งานแบบ writer มีปัญหา, เช่นกระดาษหมด หรือ กระดาษติด, ซึ่งไม่สามารถทำการพิมพ์ต่อไปได้. เมื่อเกิดปัญหานี้ขึ้น, บางครั้งจำเป็นต้องมีการเข้ามาช่วยเหลือของโอเปอเรเตอร์.

### เสร็จสิ้นการพิมพ์

ไฟล์พริ้นเตอร์เอาต์พุตถูกลบทิ้ง. หมายเหตุ, ไฟล์พริ้นเตอร์เอาต์พุตอาจจะ หรือ อาจจะไม่ถูกส่งพิมพ์.

---

## วิธีการทำให้งานเสร็จสมบูรณ์

หัวข้อนี้จะช่วยอธิบายว่างานคืออะไร, ต้องเช็คอะไรบ้างก่อนให้งานเริ่มต้นทำงาน, งานถูกส่งต่อภายในระบบอย่างไร, และจะเกิดสิ่งใดต่อไปเมื่องานเสร็จสิ้นการทำงานแล้ว.

- “งานคืออะไร”
- “สิ่งที่เกิดขึ้นก่อนงานเข้าสู่ระบบ” ในหน้า 82
- “วิธีที่การทำงานจะเข้าสู่ระบบ” ในหน้า 82
- “การทำงานถูกประมวลผลอย่างไร” ในหน้า 83
- “งานออกจากระบบอย่างไร” ในหน้า 84

สำหรับข้อมูลโดยละเอียดเกี่ยวกับหลักการของ การจัดการระบบงาน, ให้ดูที่ “โครงสร้างของระบบ” ในหน้า 34.

## งานคืออะไร

บนเซิร์ฟเวอร์ iSeries, งานจะถูกรันอยู่ตลอดเวลา, ไม่ว่าคุณจะสร้าง งานขึ้นหรือระบบจะสร้างงานขึ้นเองก็ตาม. งานจะถูกทำเมื่อคุณเปิดระบบ ขึ้น, เมื่อคุณเปิดไฟล์, หรือเมื่อคุณ query ฐานข้อมูล. ไม่ว่าจะเป็นการปฏิบัติงานใดๆ บนเซิร์ฟเวอร์ iSeries ก็จะมีงานที่ กำลังถูกทำอยู่เสมอ.



ชุดงานแต่ละอันบนระบบจะถูกกระทำผ่านทางชิ้นงาน. ชิ้นงานอาจจะเปรียบง่าย ๆ ได้กับแอปพลิเคชันที่รอให้ผู้ใช้เรียกทำงาน หรืออาจเปรียบได้ซับซ้อนกับ system query เพื่อมอนิเตอร์จำนวนผู้ใช้บนระบบทุกๆ ชั่วโมงที่จะต้องรันอยู่อย่างสม่ำเสมอ. งานบางอย่าง, โดยเฉพาะงานแบ็คซ์และงานโต้ตอบ, จะมี “Job description” ในหน้า 35 เกี่ยวข้องอยู่ด้วยเพื่อบอกให้รู้ว่างานจะรันเมื่อใดและที่ใด.

งานประกอบด้วยโปรแกรมต่างๆ ที่จะทำฟังก์ชันบางอย่างเฉพาะ. ไม่มีขีดจำกัดถึงจำนวน ฟังก์ชันที่งานจะสามารถใช้ปฏิบัติงานได้. งานประกอบด้วยคำสั่งตามขั้นตอนที่จะต้องกระทำเพื่อให้งานเสร็จสมบูรณ์. หรือก็คือโปรแกรมที่ทำให้งานรันไปตามลำดับอย่างเจาะจง. ตัวอย่างเช่น, โปรแกรม A จะต้องรันก่อนที่โปรแกรม B จะเริ่มทำงานได้. “thread” ในหน้า 53 ช่วยให้งานนั้นเสร็จสมบูรณ์. งานแอคทีฟจะประกอบด้วย thread อย่างน้อยหนึ่งตัว. เมื่องานมี thread อยู่หลายตัว, ก็จะทำให้งานสามารถทำสิ่งต่างๆ ได้มากขึ้นในเวลาเดียวกัน. ตัวอย่างเช่น, thread อันหนึ่งสามารถออกไปทำการคำนวณ ในขณะที่อีกอันหนึ่งจะรอข้อมูลเข้ามามากขึ้นเพื่อทำการประมวลผลต่อไป.

สำหรับข้อมูลโดยละเอียดเพิ่มเติมเรื่องงานและชนิดของงานบนเซิร์ฟเวอร์ iSeries, ให้อ่านที่ “งาน” ในหน้า 35.

## สิ่งที่เกิดขึ้นก่อนงานเข้าสู่ระบบ

งานทั้งหมดยกเว้นงานระบบ จะรันภายใต้ระบบย่อย. เพื่อจะเริ่มงานในระบบย่อยที่แอคทีฟ, พูลหน่วยความจำและอย่างน้อยที่สุด “วิธีที่การทำงานจะเข้าสู่ระบบ” จะต้องถูกสร้างขึ้นก่อน. คิวงานเป็นตัวอย่างของซอร์สของงาน. เซิร์ฟเวอร์ iSeries มาพร้อมกับชุดดีฟอลต์ของ คิวงาน, ระบบย่อย, และพูลหน่วยความจำที่อนุญาตให้งานเริ่มต้นได้ทันทีที่ระบบถูกเปิดขึ้นมา.

คุณสามารถปรับแต่งระบบย่อย, ค่า configuration ของพูลหน่วยความจำเพื่อให้เซิร์ฟเวอร์ iSeries ของคุณทำงานได้เต็มประสิทธิภาพ. ตัวอย่างเช่น, หากงานแบบแบ็คซ์ มีความสำคัญต่อความสำเร็จของธุรกิจ, คุณอาจต้องการให้จัดสรรหน่วยความจำให้กับงานแบบ แบ็คซ์มากขึ้น. หรือคุณอาจกำหนดจำนวนงานที่จะรันในเวลาหนึ่งในระบบย่อย Qbatch ของคุณ ให้น้อยลงเพื่อให้งานสามารถใช้ปริมาณรีซอร์สในการรันได้มากขึ้น. นอกจากนี้, คุณสามารถสร้างคิวงาน, ระบบย่อย และพูลหน่วยความจำโดยออกแบบให้ทำงานชนิดใดชนิดหนึ่งเป็นการเฉพาะ. ตัวอย่างเช่น, คุณสามารถสร้างคิวงานชื่อ Nightreps เพื่อให้รายงานแบบแบ็คซ์เป็นรายคืนถูกส่งไปยังระบบย่อยที่ชื่อ Nightrep ซึ่งจะจัดสรรหน่วย ความจำไว้เป็นการเฉพาะเพื่อใช้รันงานแบบแบ็คซ์เหล่านี้.

หากต้องการทราบข้อมูลเพิ่มเติมเกี่ยวกับคิวงาน, ระบบย่อย, และพูลหน่วยความจำ, โปรดดูที่ “โครงสร้างของระบบ” ในหน้า 34. สำหรับข้อมูลเพิ่มเติมเกี่ยวกับสิ่งที่ IBM สนับสนุนการจัดการระบบงาน, ให้อ่านที่ ภาคผนวก C. IBM-Supplied Object

Contents ในหนังสือคู่มือ V4R5 Work Management  (ประมาณ 2720 กิโลไบต์ หรือ 573 หน้า).

## วิธีที่การทำงานจะเข้าสู่ระบบ

work entry จะบอกถึงต้นแหล่งที่งานได้เข้ามาสู่ระบบย่อยเพื่อรอให้เรียกทำงาน. ในแต่ละชนิดของงานบนระบบ iSeries ก็จะมี work entry ที่แตกต่างกันตามการใช้งาน.

งานแบบแบ็คซ์ส่วนใหญ่จะใช้คิวงานเพื่อเข้าสู่ระบบย่อย. Entry ของคิวงานจึงเป็นกลไกหนึ่งที่คิวงานจะถูกกำหนดให้เป็นต้นแหล่งของการทำงาน ที่จะส่งเข้าไปสู่ระบบย่อย.

work entry จะถูกเก็บเอาไว้ใน “Subsystem description” ในหน้า 59. ถ้า subsystem description ไม่มี work entry สำหรับชนิดของงานที่กำลังจะถูกรัน, งานดังกล่าวจะไม่สามารถรันในระบบย่อยได้. ระบบย่อยที่จัดส่งมาจาก IBM จะมี work entry ดีฟอลต์



อยู่ใน subsystem description แล้ว. แต่ต้องไม่ลืมว่า work entry ดีฟอลต์บางตัวที่จัดส่งมาพร้อมกับระบบย่อยนั้นได้ถูกจัดสรรให้ทำการรันงานแบบเจาะจงเท่านั้น. ตัวอย่างเช่น, ในระบบย่อย QCMN, work entry เกี่ยวกับการสื่อสารตัวหนึ่งถูกเชื่อมต่อไว้เพื่อรันเซิร์ฟเวอร์ iSeries Access.

สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับวิธีที่งานเข้าสู่ระบบ, ให้ดู work entries ในบทที่ 4 ของคู่มือ V4R5 Work Management



(ประมาณ 2720 กิโลไบต์ หรือ 573 หน้า).

## การทำงานถูกประมวลผลอย่างไร

เมื่อเซิร์ฟเวอร์ iSeries ถูกเรียกใช้งาน, งานมอนิเตอร์ระบบย่อย จะเริ่มทำการรัน. งาน มอนิเตอร์ระบบย่อยจะควบคุมงานภายใน “ระบบย่อย” ในหน้า 58. นอกจากนี้ ยังเป็นตัวช่วยสตาร์ทและจบการทำงานให้, พร้อมกับจัดการรีซอร์สให้กับการทำงานนั้นในระบบย่อยอีกด้วย. การทำงาน (หรืองานหลายๆ งาน) จะเข้าสู่ระบบย่อยผ่านทาง “วิธีที่การทำงานจะเข้าสู่ระบบ” ในหน้า 82 ที่ซึ่งงานจะกลายเป็นแอสคทีฟและพร้อมที่จะรัน. การทำงานจะเสร็จสมบูรณ์ก็ต่อเมื่อระบบย่อยได้จัดสรรหน่วยความจำให้รันเท่านั้น. หน่วยความจำจะถูกจัดสรรให้กับระบบย่อยโดย “พูลหน่วยความจำ” ในหน้า 75.

### subsystem description ช่วยประมวลผลการทำงานได้อย่างไร

เช่นเดียวกับงาน, ระบบย่อยจะมี description, ที่เรียกว่า “Subsystem description” ในหน้า 59. subsystem description จะประกอบด้วยข้อมูลสำคัญที่บอกวิธี, ตำแหน่ง, และปริมาณการทำงานที่สามารถแอสคทีฟในระบบย่อยในเวลาหนึ่ง, และรีซอร์สที่สามารถเรียกใช้เพื่อทำงานได้.

### Routing entry

routing entry จะอยู่ในรายละเอียดระบบย่อยซึ่งจะแจ้งระบบย่อยว่าใช้พูลหน่วยความจำใดรันงานเข้าสู่ระบบ, ใช้โปรแกรมรันงาน, และใช้คลาสอ็อบเจกต์ใดรันงาน. สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ routing entry, โปรดดูที่บทที่ 4 ในคู่มือ



V4R5 Work Management .

### คลาสอ็อบเจกต์

คลาสอ็อบเจกต์กำหนดระดับความสำคัญในการรัน, ดีฟอลต์ช่วงเวลารอ, ช่วงเวลาที่ระบบแบ่งเวลาประมวลผลและแอสคทีฟ บิวต์อื่น ๆ. ระดับความสำคัญในการรัน มีความสำคัญเพราะเป็นตัวกำหนดว่าเมื่อไรที่งานจะได้เวลาของตัวประมวลผลเพื่อรันงาน. ระดับความสำคัญในการรันจะอยู่ระหว่าง 0 ถึง 99, โดยที่ 0 มีความสำคัญสูงสุด. (เฉพาะงานระบบจะได้รับระดับความสำคัญเป็น 0 เพราะเป็นงานที่ใช้รันเซิร์ฟเวอร์ iSeries.)

เมื่องานเข้าสู่ระบบย่อย, ระบบย่อยจะพยายามจับคู่ routing data กับค่าเปรียบเทียบใน routing entry. หาก routing data และค่าเปรียบเทียบใน routing entry จับคู่กันได้, routing entry จะถูกกำหนดไว้ให้กับงานนั้น. แต่ถ้าจับคู่กันไม่ได้, งานนั้นจะหยุดลง.

ปัจจัยอื่นที่มีผลกระทบในกรณีที่งานรันอยู่ในระบบย่อย คือ จำนวนงานที่ได้รับ อนุญาตให้มีสถานะแอสคทีฟอยู่ในระบบย่อยในเวลาหนึ่ง (หรือรู้จักในอีกชื่อว่า “คุณสมบัติของระบบย่อย” ในหน้า 72 ในระบบย่อย). เมื่องานที่แอสคทีฟในระบบย่อยมีมากถึงจำนวนสูงสุดแล้ว, งานอื่น ๆ จะไม่สามารถเข้ามาสู่ระบบย่อยได้จนกว่างานแอสคทีฟจะรันเสร็จสิ้น. “พูลหน่วยความจำ” ในหน้า 75 จะต้องถูกจัดสรรให้กับระบบย่อยเพื่อให้งานรัน. “activity level ของพูลหน่วยความจำ” ในหน้า 75 จะเป็นตัวบอกเซิร์ฟเวอร์ iSeries ว่ามี thread เท่าไรที่สามารถแอสคทีฟภายในพูลหน่วยความจำ. จำไว้ว่า, งานแอสคทีฟจะประกอบด้วย thread อย่างน้อยหนึ่งอัน. เมื่อ activity level ของพูลหน่วยความจำถึงระดับสูงสุด, งานจะต้องรอให้ thread อื่นใช้งานของ activity level ให้เสร็จสิ้นเสียก่อน. งานสามารถแอสคทีฟอยู่ในระบบย่อยโดยไม่ได้รัน.

หมายเหตุ: กรุณาอย่าสับสนระหว่าง “คุณสมบัติของระบบย่อย” ในหน้า 72 ของระบบย่อย กับ “activity level ของพุลหน่วยความจำ” ในหน้า 75.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับงาน, ระบบย่อย, และพุลหน่วยความจำ, ให้อูที่คู่มือ V4R5 Work Management (ประมาณ 2720 กิโลไบต์ หรือ 573 หน้า).




## งานออกจากระบบอย่างไร

เอาต์พุตคิทำงานคล้ายกับคิงานตรงที่มันจะจัดลำดับเอาต์พุตสิ่งที่พิมพ์ให้. ทั้งพริเตอร์ เอาต์พุตและเอาต์พุตคิจะมีแอ็ทริบิวต์ที่จะใช้สำหรับพิมพ์ข้อมูล.

พริเตอร์เอาต์พุตจะมีข้อมูลเอาต์พุตที่รอประมวลผล, เช่นข้อมูลที่รอพิมพ์. พริเตอร์เอาต์พุตจะมีข้อมูลที่สำคัญที่ใช้เพื่อจัดลำดับการพิมพ์. แอ็ทริบิวต์พริเตอร์เอาต์พุตมี “เอาต์พุตคิ” ในหน้า 77 ที่มีพริเตอร์เอาต์พุตอยู่, ระดับความสำคัญ, “สถานะของพริเตอร์เอาต์พุต” ในหน้า 80 และลำดับของพริเตอร์เอาต์พุต.

เอาต์พุตคิมีแอ็ทริบิวต์ของตัวเองที่กำหนด “ลำดับของไฟล์” ในหน้า 79 การประมวลผลไฟล์พริเตอร์เอาต์พุต. นอกจากนั้นเอาต์พุตคิยังมี “แอ็ทริบิวต์ของเอาต์พุตคิ” ในหน้า 79 ที่จำเป็นสำหรับการเปลี่ยนแปลงกับพริเตอร์เอาต์พุตและเอาต์พุตคิ.

เมื่อพริเตอร์เอาต์พุตพร้อมที่จะถูกส่งไปยังพริเตอร์งานแบบ writer จะเป็นคิพริเตอร์เอาต์พุตขึ้นมา. งานแบบ writer จะใช้ข้อมูลจากพริเตอร์เอาต์พุตและเตรียมเอาต์พุตนั้นเพื่อพิมพ์.



สำหรับรายละเอียดเกี่ยวกับวิธีที่เอาต์พุตคิถูกเลือกให้อูที่ การควบคุมการพิมพ์ใน บทที่ 1 ของคู่มือ การโปรแกรมมิงอุปกรณ์เครื่องพิมพ์ .

คุณสามารถสร้างเอาต์พุตคิเจาะจงขึ้นมาเองหรือใช้เอาต์พุตคิที่ส่งมา พร้อมกับระบบก็ได้. สำหรับรายละเอียดเพิ่มเติม, ให้อู การสร้างเอาต์พุตคิ.

---

## การจัดการการแก้ไขปัญหางาน

เมื่องานเกิดประมวลผลไม่เต็มประสิทธิภาพเท่าที่ควร บนเซิร์ฟเวอร์ iSeries, อาจเป็นไปได้ว่างานเกิดติดขัด, หรือเป็นเพียงการทำงานที่ด้อยประสิทธิภาพธรรมดา. ไม่ว่ากรณีใดก็ตาม, ก็มีวิธีที่จะตรวจสอบและแก้ไขเตรียมไว้เพื่อช่วยให้คุณจัดการกับปัญหาได้เสมอ. กรุณาดูหัวข้อต่อไปสำหรับรายละเอียดเพิ่มเติม.

- “งานเกิดหยุดการทำงาน”
- “งานของฉันประสบปัญหาทำงานช้า” ในหน้า 86
-  เวลาระบบไม่ถูกต้องและงานนั้นไม่เสร็จภายในเวลาอันเหมาะสม 

## งานเกิดหยุดการทำงาน

ต่อไปนี้เป็นสาเหตุที่พอจะเป็นไปได้ที่งานหยุดการทำงาน:

งานกำลังรอที่จะได้ล็อกอ็อบเจกต์  
ตรวจสอบได้อย่างไร:

วิธีแก้ไข:

ให้ดูที่สถานะของงานใน iSeries Navigator; ให้ดูที่ “การกำหนดสถานะของงาน” ในหน้า 13. งานที่กำลังรอการล็อกจะมีสถานะเป็น *กำลังรอการล็อก*.  
ให้ดูที่รายชื่อ อ็อบเจกต์ที่ถูกล็อก ของงานที่กำหนดว่างานที่กำลังรอที่จะได้ล็อก “รายละเอียด: การปฏิบัติงานแอ็คทีฟ” ในหน้า 50. แล้วใช้การดำเนินการแบบ Lock Holders กับอ็อบเจกต์เพื่อระบุว่างานใดที่กำลังล็อกอ็อบเจกต์นั้นอยู่. นอกจากนี้ คุณจะต้องกำหนดว่าทำไมงานดังกล่าวจึงต้องถูกล็อกเอาไว้, และจะทำเช่นไรเพื่อปลดล็อกนั้นออกให้ได้. ใน V5R2, ค่าสถานะอาจมีข้อมูลเพิ่มเติมเกี่ยวกับหน้าคุณสมบัติ. ตัวอย่าง, สถานะที่กำลังรอการล็อกบนหน้า Properties จะแสดงว่าอ็อบเจกต์อะไรเชื่อมโยงกับ request การล็อก.

งานถูกพักไว้

ตรวจสอบได้อย่างไร:  
วิธีแก้ไข:

ให้ดูที่สถานะของงานใน iSeries Navigator; ให้ดูที่ “การกำหนดสถานะของงาน” ในหน้า 13. คลิกขวาไปบนงาน แล้วเลือก *ปล่อย*.

ต่อไปนี้เป็นสาเหตุที่พอจะเป็นไปได้ที่งานในคิวงานอาจหยุดการทำงาน ติดขัด:

คิวงานถูกพักไว้

ตรวจสอบได้อย่างไร:  
วิธีแก้ไข:

ให้ดูที่สถานะของคิวงานใน iSeries Navigator;  
1. ย้ายงานไปไว้ในคิวงานที่ไม่ได้ถูกพักไว้; ให้ดูเรื่อง “การย้ายงานไปยังคิวงานอื่น” ในหน้า 21.  
2. ปล่อยคิวงาน. หากต้องการทำเช่นนั้น, ให้คลิกขวาที่งานและเลือก *Release*.

คิวงานไม่ได้ถูกจัดสรรไว้ให้โดยระบบย่อยแอ็คทีฟ

ตรวจสอบได้อย่างไร:  
วิธีแก้ไข:

ให้ดูที่สถานะของคิวงานใน iSeries Navigator.  
1. ย้ายงานไปยังคิวงานที่ได้รับการจัดสรรไว้โดยระบบย่อยแอ็คทีฟ; ให้ดูเรื่อง “การย้ายงานไปยังคิวงานอื่น” ในหน้า 21.  
2. สตาร์ทระบบย่อยที่ไม่มี job queue entry สำหรับคิวงานดังกล่าว; ให้ดูที่ “การเริ่มใช้งานระบบย่อย” ในหน้า 25.  
3. ใส่ job queue entry ของคิวงานดังกล่าวลงไปในระบบย่อยแอ็คทีฟ โดยใช้คำสั่ง Add Job Queue Entry (ADDJOBQE).

ครบจำนวนสูงสุดของระบบย่อย

ตรวจสอบได้อย่างไร:  
วิธีแก้ไข:

ให้ดูที่ค่างานแอ็คทีฟสูงสุดสำหรับระบบย่อยใน iSeries Navigator. หากต้องการทำเช่นนั้น, ให้คลิกขวาที่งานและเลือก *Properties*.  
1. ย้ายงานไปไว้ในคิวงานอื่น; ให้ดูเรื่อง “การย้ายงานไปยังคิวงานอื่น” ในหน้า 21.  
2. เพิ่มค่าสูงสุด. หากต้องการทำเช่นนั้น, ให้ใช้คำสั่ง Change Subsystem Description (CHGSBSD).

ครบจำนวนสูงสุดของคิวงาน  
ตรวจสอบได้อย่างไร:

วิธีแก้ไข:

ให้ดูที่ค่างานแอ็คทีฟสูงสุดสำหรับคิวงานใน iSeries Navigator. ในการทำเช่นนั้น, ให้คลิกขวาที่คิวงาน  
และเลือก *Properties*. แล้วจึงเลือกแท็บ *Activity*.

1. ย้ายงานไปไว้ในคิวงานอื่น; ให้ดูเรื่อง“การย้ายงานไปยังคิวงานอื่น” ในหน้า 21.
2. เพิ่มค่าสูงสุด. ในการเพิ่มค่าสูงสุด, ให้ใช้คำสั่ง Change Job Queue Entry (CHGJOBQE).

ครบค่าสูงสุดของระดับความสำคัญ  
ตรวจสอบได้อย่างไร:

วิธีแก้ไข:

ระบุระดับความสำคัญของคิวงาน ของงานโดยการดูที่คุณสมบัติของงานนั้น. ให้ดูที่งานแอ็คทีฟสูงสุด  
ด้วยค่าระดับความสำคัญงานสำหรับคิวงานใน iSeries Navigator. ในการทำเช่นนั้น, ให้คลิกขวาที่คิว  
งานและเลือก *Properties*. แล้วเลือกแท็บ *Activity* แล้วคลิกปุ่ม *Advanced*.

1. ย้ายงานไปไว้ในคิวงานอื่น; ให้ดูเรื่อง“การย้ายงานไปยังคิวงานอื่น” ในหน้า 21.
2. เปลี่ยนระดับความสำคัญของคิวงานของงานนั้น; ให้ดูเรื่อง“การเปลี่ยนระดับความสำคัญของงาน  
ภายในคิวงาน” ในหน้า 20.
3. เพิ่มค่าสูงสุด. ในการเพิ่มค่าสูงสุด, ให้ใช้คำสั่ง Change Job Queue Entry (CHGJOBQE).

## งานของฉันประสบปัญหาทำงานช้า

ต่อไปนี้เป็นสาเหตุที่เป็นไปได้ว่าทำไมงานจึงประสบปัญหาการทำงานช้า:

หน่วยความจำไม่เพียงพอ  
จะทราบได้อย่างไร:

วิธีแก้ไข:

ให้ดูที่คุณสมบัติของงานเพื่อระบุว่างานกำลังรันอยู่ในพูลหน่วยความจำ อันใด. แล้วดูคุณสมบัติของ  
พูลหน่วยความจำใน iSeries Navigator; โดยดูที่ “ตรวจสอบการใช้พูลหน่วยความจำ” ในหน้า 28.  
อัตราความผิดพลาดสูงในพูลบอกได้ว่ามีหน่วยความจำในพูลที่ไม่ เพียงพอ, หรือในพูลมีงานเป็น  
จำนวนมากที่แย่งใช้หน่วยความจำ กันอยู่.

1. ให้เปิดตัวปรับแต่งระบบ ในกรณีที่คุณไม่ได้ใช้งานอยู่. ค่าของระบบ QPFRADJ จะปรับพูลหน่วย  
ความจำและ activity level โดยอัตโนมัติ.
2. ถ้าเป็นไปได้, ให้ปรับแต่งพูลด้วยตัวคุณเองโดยให้เพิ่มจำนวนหน่วย ความจำในพูล หรือลด  
activity level ของพูลหน่วยความจำลง. คุณอาจต้องการตรวจสอบพูลเครื่องเพื่อตรวจสอบขนาด  
ของหน่วยความจำที่ใช้งานว่าไม่มีผลต่องานทั้งหมดบนระบบ.

Activity level ต่ำเกินไป  
จะทราบได้อย่างไร:

วิธีแก้ไข:

ดูที่คุณสมบัติของงานเพื่อระบุสถานะและดูว่างานกำลังรันอยู่ในพูลหน่วยความจำอันใด. หากงาน  
แสดงสถานะ เป็น *Waiting for activity level*, ก็ให้ดูคุณสมบัติของ พูลหน่วยความจำใน iSeries  
Navigator; ดูที่ “ตรวจสอบการใช้พูลหน่วยความจำ” ในหน้า 28. อัตราการเปลี่ยนไปสู่สถานะการไม่มี  
สิทธิในพูลที่มีสูงนั้นเป็นการ ชี้ให้เห็นว่าในพูลมีจำนวนงานมากเกินไปซึ่งแย่งกันใช้หน่วยความจำกัน  
อยู่.

1. ให้เปิดตัวปรับแต่งระบบ ในกรณีที่คุณไม่ได้ใช้งานอยู่. ค่าของระบบ QPFRADJ จะปรับพูลหน่วย  
ความจำและ activity level โดยอัตโนมัติ.
2. ให้ปรับพูลด้วยตนเองโดยการเพิ่ม activity level ของพูลหน่วยความจำ.

CPU รีซอร์สไม่เพียงพอ  
จะทราบได้อย่างไร:

วิธีแก้ไข:

ให้ดูที่คอลัมน์ "CPU %" ของงาน และงานอื่นๆ ในลิสต์ Active Jobs ของ iSeries Navigator. หากระบบไม่ว่างอยู่, งานของคุณอาจได้รับ CPU รีซอร์สไม่เพียงพอที่จะทำงานให้เสร็จสิ้นได้.

1. หากเป็นไปได้, จบงานหรือพักงานที่ไม่สำคัญในระบบเสีย.
2. หากงานจำนวนหนึ่งคือการทำงานที่เน้นใช้ CPU มาก, ให้เปลี่ยนระดับความสำคัญในการรันงานเหล่านี้ (ค่าระดับความสำคัญที่สูงขึ้นเท่ากับระดับการรันที่ต่ำลงของงาน).

อ็อปชันการเพจพูลหน่วยความจำ  
จะทราบได้อย่างไร:

วิธีแก้ไข:

หากแอพลิเคชันคือการทำงานที่เน้นใช้ disk, หาก CPU ทำงานอย่างไม่เต็มประสิทธิภาพและเหลือหน่วยความจำไม่เพียงพอ, ในกรณีการใช้ expert cache จะมีประโยชน์มาก.

expert cache สามารถเปิดใน iSeries Navigator ได้โดยการเปลี่ยนอ็อปชัน Paging สำหรับพูล หน่วยความจำไปเป็นอ็อปชัน Calculated. อ็อปชัน Paging อยู่ที่แท็บ Configuration ของหน้า Properties ของพูลหน่วยความจำและจะมีอยู่บนพูลที่ใช้งานร่วมกันเท่านั้น (ไม่ใช่พูลที่ใช้งานคนเดียว).

ระดับความสำคัญในการรันของงานต่ำ

จะทราบได้อย่างไร:

วิธีแก้ไข:

ดู "คุณสมบัติของงาน" ในหน้า 46 ของงานเพื่อระบุ ระดับความสำคัญของงานที่สัมพันธ์กับงานอื่นๆ ในระบบ.

ถ้าหากงานมีระดับความสำคัญที่ต่ำ (ค่าสูง) โดยเปรียบเทียบกับงานอื่นๆ และไม่ได้ใช้ CPU มากนัก เพราะ งานที่มีระดับความสำคัญที่สูง (ค่าต่ำ) จะใช้ CPU รีซอร์สส่วนใหญ่อยู่แล้ว, คุณอาจจำเป็นต้องเพิ่มระดับความสำคัญของงานให้มากขึ้น; ให้ดูหัวข้อ "คุณสมบัติของงาน" ในหน้า 46. เช่นเดียวกัน, ในระบบที่มีการใช้งาน CPU สูง และงานมีระดับความสำคัญต่ำ, การเซ็ค่าของการปรับเปลี่ยนระดับความสำคัญของงานแบบ Dynamic ภายในแบนด์ระดับความสำคัญ (QDYNPTYSCD) และการปรับเปลี่ยนระดับความสำคัญของงานแบบ Dynamic ของงานแบบโต้ตอบ (QDYNPTYADJ) อาจมีประโยชน์.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับประสิทธิภาพการทำงาน, ให้ดูที่ การทำงาน. หากคุณต้องการข้อมูลเพิ่มเติมเกี่ยวกับวิธีการปรับแต่งการทำงานบนระบบของท่าน, ให้ดูที่ การปรับแต่งการทำงาน.

---

## ข้อมูลที่เกี่ยวข้องกับการจัดการระบบงาน

» รายการด้านล่างคือคู่มือของ iSeries<sup>(TM)</sup> (ในรูปแบบ PDF) และ experience report ที่เกี่ยวข้องกับการจัดการระบบงาน. คุณสามารถอ่านหรือพิมพ์เอกสาร PDFเหล่านี้.

คู่มือ

- V4R5 Work Management



คู่มือเล่มนี้จะให้ข้อมูลเกี่ยวกับการจัดการเวิร์กโหนดระบบอย่างมีประสิทธิภาพโดยการเปลี่ยนอ็อบเจกต์ของการจัดการระบบงานให้เหมาะสมกับความต้องการ. เอกสารฉบับนี้ยังให้แนวทางการปรับแต่งการทำงาน, รายละเอียดของค่ากำหนดระบบ, รายละเอียดเกี่ยวกับการเก็บรวบรวมข้อมูลการทำงาน, การเก็บข้อมูลการใช้งานของระบบ, การใช้ work entries, และการกำหนดเวลาให้กับงานแบบแบตช์.

- **Job Scheduler for OS/400**



คู่มือนี้ให้ข้อมูลเกี่ยวกับระบบตารางเวลางานใน OS/400 .

### บันทึกจากประสบการณ์

- **configuration ของระบบย่อย**

default subsystem configuration ที่มากับ OS/400 เป็น configuration ของระบบย่อยแบบเบื้องต้นที่สามารถทำงานได้กับระบบขนาดเล็ก. อย่างไรก็ตาม, เมื่อจำนวนผู้ใช้ระบบเพิ่มขึ้น, การแยกออกเป็นหลายๆระบบย่อยจะกลายเป็นสิ่งจำเป็นเพื่อให้การควบคุมงานในระบบเป็นไปได้ง่ายขึ้น. บันทึกจากประสบการณ์นี้จะให้ข้อมูลเพิ่มเติมเกี่ยวกับ configuration ของระบบย่อย.

- **การปรับแต่ง prestart job entries**

ในบันทึกนี้ได้มีการอธิบายวิธีการจัดการ prestart jobs เพื่อปรับปรุงประสิทธิภาพการทำงานโดยรวมของระบบให้ดีขึ้น. งาน prestart คืองานที่เริ่มต้นรัน ก่อนที่งานหลักจะมา. prestart job entry ใน subsystem description กำหนดจำนวนงานที่ระบบต้องสร้างและวิธีการที่ระบบจัดการงานแบบ prestart.

- **ค่าปรับแต่งประสิทธิภาพ(QPFRADJ)**


เซิร์ฟเวอร์ iSeries<sup>(TM)</sup> สามารถจัดการพูลหน่วยความจำร่วมได้เองแบบอัตโนมัติโดยไม่ต้องมีผู้ใช้คอยควบคุม. ความสามารถนี้ควบคุมโดยค่าการปรับแต่งประสิทธิภาพของระบบ, QPFRADJ. เมื่อตั้งค่านีไว้ที่ '2' หรือ '3', ระบบจะคอยตรวจสอบประสิทธิภาพของพูลร่วมที่แอคทีฟทั้งหมดเป็นระยะ โดยจะปรับแต่งหรือลำดับค่าหน่วยความจำและ activity level ใหม่ตามที่จำเป็น. ฟังก์ชันนี้ถูกตั้งเป็นแอคทีฟโดยดีฟอลต์ (ระบบที่ส่งมาจะมีค่าของ QPFRADJ เป็น '2' ซึ่งหมายถึง 'ทำการปรับแต่งตั้งแต่ตอน IPL รวมทั้งให้ปรับแต่งแบบอัตโนมัติ'). บันทึกจากประสบการณ์จะอธิบายว่าค่า Work with Shared Pools (WRKSHRPOOL) ที่ผู้ใช้กำหนดเองจะมีผลกระทบกับ algorithm ของการปรับแต่งประสิทธิภาพได้อย่างไร, รวมถึงแสดงตัวอย่างในการตั้งค่าให้เหมาะสมกับสภาพแวดล้อมของคุณ.

### การบันทึกไฟล์ PDF

การบันทึกไฟล์ PDF ไว้ที่เวิร์กสเตชันของคุณเพื่อใช้ดูหรือพิมพ์:

1. คลิกขวาที่ PDF ในบราวเซอร์ของคุณ (คลิกขวาที่ลิงก์ด้านบน).
2. คลิก Save Target As... ถ้าคุณใช้ Internet Explorer. คลิก Save Link As... ถ้าคุณใช้ Netscape Communicator.
3. เปลี่ยนไปยังไดเรกทอรีที่คุณต้องการจะ เก็บ PDF.
4. คลิก Save.

### การดาวน์โหลดโปรแกรม Adobe Acrobat Reader

คุณต้องใช้โปรแกรม Adobe Acrobat Reader เพื่อใช้อ่านหรือพิมพ์ไฟล์ที่เป็น PDF. โดยคุณสามารถดาวน์โหลดได้ที่เว็บไซต์  
ของ Adobe ([www.adobe.com/products/acrobat/readstep.html](http://www.adobe.com/products/acrobat/readstep.html))  .





---

## ภาคผนวก. ประกาศ

ข้อมูลนี้ถูกพัฒนาขึ้นสำหรับผลิตภัณฑ์และบริการที่เสนอขายในประเทศสหรัฐอเมริกา.

ไอบีเอ็มอาจไม่เสนอขายผลิตภัณฑ์, บริการ, หรือคุณสมบัติต่างๆ ที่กล่าวถึงในเอกสารนี้ในประเทศอื่น. ปรีกษาตัวแทนไอบีเอ็มในท้องที่นั้นๆ ของคุณสำหรับข้อมูลเกี่ยวกับผลิตภัณฑ์และบริการที่เสนอขายในท้องที่ของคุณ. การอ้างถึงผลิตภัณฑ์, โปรแกรม, หรือบริการของไอบีเอ็ม ไม่ได้หมายความว่าต้องเฉพาะ ผลิตภัณฑ์, โปรแกรม, หรือบริการที่เป็นของไอบีเอ็มเท่านั้นที่จะถูกใช้. ผลิตภัณฑ์, โปรแกรม, หรือบริการอื่นที่ทำงานได้เทียบเท่า และไม่ละเมิดลิขสิทธิ์ของไอบีเอ็ม ก็สามารถใช้ได้. อย่างไรก็ตาม, เป็นความรับผิดชอบของผู้ใช้ที่จะประเมิน และตรวจสอบผลิตภัณฑ์, โปรแกรม, หรือบริการที่ไม่ใช่ของไอบีเอ็ม.

ไอบีเอ็มอาจมีสิทธิบัตรหรือเอกสารการขอสิทธิบัตร ที่ครอบคลุมสิ่งที่อธิบายในเอกสารนี้. การตกแต่งเอกสารใหม่ไม่ได้ทำให้คุณได้สิทธิของสิทธิบัตรเหล่านั้น. คุณสามารถสอบถามเกี่ยวกับไลเซนส์, โดยเขียนและส่งไปที่:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-17855  
U.S.A.

สำหรับการสอบถามเกี่ยวกับไลเซนส์โดยพิจารณาข้อมูล double-byte (DBCS), ให้ติดต่อ IBM Intellectual Property Department ในประเทศของคุณ หรือโดยการเขียน และส่งคำถามไปที่:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106-0032, Japan

**ย่อหน้าต่อไปนี้ ไม่ใช้กับประเทศสหราชอาณาจักร หรือประเทศอื่นที่ข้อกำหนดไม่สอดคล้องกับกฎหมายท้องถิ่น:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. บางรัฐไม่อนุญาตการปฏิเสธของการรับประกันอย่างชัดเจน หรือโดยนัยในการทำการซื้อขายบางอย่าง, ดังนั้น ประโยชน์ข้างต้นนี้อาจไม่ได้มีความหมายต่อคุณ.

ข้อมูลนี้ได้รับความไม่ถูกต้องทางเทคนิคหรือความผิดพลาดทางการพิมพ์. การเปลี่ยนแปลงข้อมูลในนี้จะมีเป็นระยะๆ ซึ่งจะสอดคล้องกับการตีพิมพ์ในครั้งใหม่. ไอบีเอ็มอาจทำการปรับปรุง และ/หรือเปลี่ยนแปลงในผลิตภัณฑ์ และ/หรือโปรแกรมที่อธิบายในสิ่งพิมพ์นี้ได้ตลอดเวลาโดยไม่ต้องแจ้งให้ทราบ.

การอ้างถึงเว็บไซต์ที่ไม่ใช่ของไอบีเอ็มนั้นถูกจัดหามาเพื่อความสะดวกเท่านั้น และไม่ได้มีการรับรองเว็บไซต์เหล่านั้น. เนื้อหาในเว็บไซต์เหล่านั้น ไม่ได้เป็นส่วนหนึ่งของเนื้อหาสำหรับผลิตภัณฑ์นี้ของไอบีเอ็มและการใช้เว็บไซต์เหล่านั้นเป็นความเสี่ยงของตัวเอง.

ไอบีเอ็มอาจใช้หรือเผยแพร่ข้อมูลใดๆ ที่คุณให้ไว้ในทางที่ไอบีเอ็มเชื่อว่าเหมาะสม โดยไม่มีข้อผูกมัดใดๆ กับคุณ.

สำหรับผู้ที่มีไลเซนส์ของโปรแกรมนี้ที่ต้องการมีข้อมูลเกี่ยวกับโปรแกรมสำหรับ จุดประสงค์ให้ทำงานได้: (i) การแลกเปลี่ยนข้อมูลระหว่างโปรแกรมที่ถูกสร้างขึ้นอย่าง เป็นอิสระและโปรแกรมอื่น (รวมทั้งโปรแกรมนี้) และ (ii) การใช้ข้อมูลร่วมกันซึ่งมีการ แลกเปลี่ยน ควรติดต่อ:

IBM Corporation  
Software Interoperability Coordinator, Department 49XA  
3605 Highway 52 N  
Rochester, MN 55901  
U.S.A.

ข้อมูลเหล่านี้ อาจมีให้โดยขึ้นอยู่กับเงื่อนไขและสถานการณ์ที่เหมาะสม, ซึ่งรวมถึงบางกรณี, เช่น การจ่ายค่าธรรมเนียม.

โปรแกรมไลเซนส์ที่อธิบายในข้อมูลนี้ และเนื้อหาของไลเซนส์ทั้งหมด ถูกจัดหามาโดยบริษัทไอบีเอ็ม ภายใต้เงื่อนไขของ IBM Customer Agreement, IBM International Program Licensed Agreement, หรือข้อตกลงอื่นๆ ที่เทียบเท่าระหว่างไอบีเอ็ม และคุณ.

ข้อมูลประสิทธิภาพใดๆ ที่มีอยู่ในนี้ถูกกำหนดอยู่ในสภาพแวดล้อมที่ถูกควบคุม. ดังนั้น, ผลที่ได้จากสภาพแวดล้อมของการปฏิบัติการนี้อาจแตกต่างกันเป็นอย่างมาก. บางระบบการวัดอาจทำบนระบบในระดับการพัฒนาและไม่มีการรับรองว่าระบบการวัดเหล่านี้ จะเหมือนกันกับระบบที่มีทั่วไป. นอกเหนือจากนี้, การวัดบางอย่างอาจเป็นการประมาณ ผ่านการคาดการณ์. ซึ่งผลที่แท้จริงอาจแตกต่างกัน. ผู้ใช้เอกสารนี้ควรตรวจสอบข้อมูลที่ใช้ได้ สำหรับสภาพแวดล้อมเฉพาะสำหรับผู้ใช้.

ข้อมูลเกี่ยวกับผลิตภัณฑ์ที่ไม่ใช่ของ IBM ได้รับมาจากซัพพลายเออร์ของผลิตภัณฑ์เหล่านั้น, การประกาศทางสาธารณะหรือแหล่งที่เป็นของสาธารณะอื่นๆ. ไอบีเอ็มไม่ได้ทดสอบผลิตภัณฑ์เหล่านั้น และไม่สามารถยืนยันความถูกต้องของประสิทธิภาพการทำงาน, ความเข้ากันได้ หรือข้อร้องเรียนอื่นๆ ที่เกี่ยวข้อง กับผลิตภัณฑ์ที่ไม่ใช่ของไอบีเอ็ม. คำถามเกี่ยวกับผลิตภัณฑ์ที่ไม่ใช่ของไอบีเอ็ม ควรถามไปที่ซัพพลายเออร์ของ ผลิตภัณฑ์เหล่านั้น.

ข้อความทั้งหมดที่เกี่ยวข้องกับทิศทางในอนาคตหรือความตั้งใจของไอบีเอ็มสามารถ เปลี่ยนแปลงหรือยกเลิกได้โดยไม่ต้องแจ้งให้ทราบ, และแสดงถึงเป้าหมายและวัตถุประสงค์ เท่านั้น.

ราคาทั้งหมดของไอบีเอ็มซึ่งแสดงไว้เป็นราคาค่าปลีกที่แนะนำของไอบีเอ็ม, เป็นราคาไม่คงที่ และสามารถเปลี่ยนแปลงได้โดยไม่ต้องแจ้งให้ทราบ. ราคาของดีลเลอร์อาจแตกต่างกัน.

ข้อมูลนี้ไว้สำหรับวัตถุประสงค์ของการวางแผนเท่านั้น. ข้อมูลนี้อาจมีการเปลี่ยนแปลง ก่อนที่ผลิตภัณฑ์ที่อธิบายนั้นมีวางจำหน่าย.

ข้อมูลนี้มีตัวอย่างของข้อมูลและรายงานที่ใช้ในการปฏิบัติงานประจำวัน. เพื่อแสดงให้เห็นอย่างสมบูรณ์ที่สุดที่เป็นไปได้, ตัวอย่างเหล่านี้ประกอบด้วย ชื่อของแต่ละราย, ชื่อของบริษัท, ตราสินค้าและผลิตภัณฑ์. ชื่อทั้งหมดเหล่านี้ถูกทำขึ้น และคล้ายคลึงกับชื่อและที่อยู่ของหน่วยธุรกิจจริงๆ.

#### COPYRIGHT LICENSE:

ข้อมูลนี้ประกอบด้วยโปรแกรมแอสเพคเคชันตัวอย่างในภาษาต้นฉบับ (source language), ซึ่งแสดงเทคนิคของโปรแกรมบนระบบปฏิบัติการที่หลากหลาย. คุณอาจคัดลอก, ดัดแปลง และ กระจายโปรแกรมตัวอย่างเหล่านี้ในรูปแบบใดๆ โดยไม่ต้องจ่ายค่าตอบแทนแก่ไอบีเอ็ม, สำหรับ จุดประสงค์ของการพัฒนา, การใช้งาน, การตลาด หรือการกระจายแอสเพคเคชันโปรแกรม ตาม แอสเพคเคชันโปรแกรมมิงอินเตอร์เฟซสำหรับแพลตฟอร์มปฏิบัติการที่โปรแกรมตัวอย่างถูกเขียน. ตัวอย่างเหล่านี้ ไม่ได้ผ่านการทดสอบภายใต้ทุกสถานการณ์. ดังนั้น ไอบีเอ็มไม่สามารถรับประกัน หรือกล่าวถึงความเชื่อถือได้, ความ

สามารถในการบริการ, หรือการทำงานของโปรแกรมเหล่านี้. คุณอาจคัดลอก, ตัดแปลง, และกระจายโปรแกรม ตัวอย่างเหล่านี้ ในรูปแบบใดๆ โดยไม่ต้องจ่ายเงินแก่ไอบีเอ็ม สำหรับจุดประสงค์ของการพัฒนา, การใช้งาน, การตลาด หรือการกระจายแอปพลิเคชันโปรแกรมตามแอปพลิเคชันโปรแกรมมิงอินเทอร์เน็ตเฟสของไอบีเอ็ม.

แต่ละสำเนาหรือบางส่วนของโปรแกรมตัวอย่าง หรืองานใดๆ ที่มาจากโปรแกรมเหล่านี้ ต้องมีข้อความแสดงลิขสิทธิ์ ดังนี้:

(C) (ชื่อบริษัทของคุณ) (ปี). บางส่วนของโค้ดนี้ได้มาจาก IBM Corp. ตัวอย่างโปรแกรม. (C) ลิขสิทธิ์ของ IBM Corp. \_ใส่จำนวนปี\_. สงวนลิขสิทธิ์.

ถ้าคุณกำลังดูสำเนาชั่วคราวของข้อมูล ภาพหรือสีที่แสดงอาจไม่ปรากฏ.

---

## เครื่องหมายการค้า

คำต่อไปนี้ เป็นเครื่องหมายการค้าของ International Business Machines Corporation ในประเทศสหรัฐอเมริกา, ประเทศอื่น, หรือทั้งสองกรณี:

Application System/400

AS/400

IBM

iSeries

Operating System/400

OS/400

400

System/36

Lotus Notes

Domino

DB2

WebSphere

Lotus, Freelance, และ WordPro เป็นเครื่องหมายการค้าของ International Business Machines Corporation และ Lotus Development Corporation ใน ประเทศสหรัฐอเมริกา, ประเทศอื่น, หรือทั้งสองกรณี.

C-bus เป็นเครื่องหมายการค้าของ Corollary, Inc. ในสหรัฐ, ประเทศอื่นๆ, หรือทั้งสองกรณี.

ActionMedia, LANDesk, MMX, Pentium, และ ProShare เป็นเครื่องหมายการค้า หรือเครื่องหมายการค้าจดทะเบียนของ Intel Corporation ในประเทศสหรัฐอเมริกา, ประเทศอื่น, หรือทั้งสองกรณี.

Microsoft, Windows, Windows NT, และโลโก้ Windows เป็นเครื่องหมายการค้าของ Microsoft Corporation ในประเทศสหรัฐอเมริกา, ประเทศอื่น, หรือทั้งสองกรณี.

SET และ SET Logo เป็นเครื่องหมายการค้าของ SET Secure Electronic Transaction LLC.

Java และเครื่องหมายการค้าที่เกี่ยวข้องกับ Java ทั้งหมดเป็นเครื่องหมายการค้าของ Sun Microsystems, Inc. ในประเทศสหรัฐอเมริกา, ประเทศอื่น, หรือทั้งสองกรณี.

UNIX เป็นเครื่องหมายการค้าจดทะเบียนของ The Open Group ในประเทศสหรัฐอเมริกาและ ประเทศอื่นๆ.

ชื่ออื่นๆ ของบริษัท, ผลิตภัณฑ์, และการบริการ อาจเป็นเครื่องหมายการค้า หรือเครื่องหมายการบริการของผู้อื่น.

---

## ข้อกำหนดและเงื่อนไขในการดาวน์โหลดและพิมพ์หนังสือนี้

การอนุญาตในการใช้ข้อมูลต่างๆ ที่คุณได้เลือกสำหรับดาวน์โหลดเป็นไปตามเกณฑ์และเงื่อนไขต่างๆ รวมถึงการตกลงยอมรับของคุณดังต่อไปนี้.

**การใช้งานเป็นส่วนส่วนตัว:** คุณอาจสร้างข้อมูลนี้ใหม่ เพื่อใช้เป็นการส่วนตัว, ไม่ใช่เชิงธุรกิจ โดยมีเงื่อนไขว่า เอกสารแสดงความเป็นเจ้าของทั้งหมดได้รับการคุ้มครอง. ไม่อนุญาตให้คุณแจกจ่าย, แสดงหรือผลิตงานจากข้อมูลต่างๆ เหล่านี้, หรือส่วนใด, โดยปราศจากการยินยอมของ IBM.

**การใช้งานในเชิงธุรกิจ:** คุณอาจสร้าง, แจกจ่ายและแสดงข้อมูลนี้ได้เฉพาะภายในองค์กรของคุณ โดยมีเงื่อนไขว่าประกาศความเป็นเจ้าของทั้งหมดได้รับการคุ้มครอง. ไม่อนุญาตให้คุณผลิตงานที่มาจากข้อมูลนี้, หรือทำสำเนา, แจกจ่าย หรือแสดงข้อมูลนี้หรือส่วนใดส่วนหนึ่งภายนอกบริษัทของคุณ, โดยปราศจากการยินยอมของ IBM.

ยกเว้น, ตามที่ประกาศอนุญาตไว้ ณ ที่นี้, ห้ามไม่ให้มีการให้การอนุญาต, ไลเซนส์ หรือสิทธิ์ต่างๆ, ไม่ว่าจะประกาศหรือแฝงไว้, แก่ข้อมูลหรือ, ซอฟต์แวร์ หรือทรัพย์สินทางปัญญาใดๆ ที่มีในนั้น.

IBM ขอสงวนสิทธิ์ตลอดอนการอนุญาตที่ให้คุณ ที่นี้เมื่อไรก็ตามที่การใช้ข้อมูลดังกล่าวเป็นภัยต่อผลประโยชน์ของ IBM หรือเมื่อคำสั่งข้างบนดังกล่าวไม่ได้ถูกปฏิบัติตามอย่างเหมาะสม.

คุณไม่สามารถดาวน์โหลด, เอ็กซ์พอร์ตหรือทำการเอ็กซ์พอร์ตข้อมูลนี้เข้าได้ ยกเว้นการปฏิบัติตามกฎหมายและข้อบังคับที่กำหนดไว้, รวมไปถึงกฎหมายและข้อบังคับในการเอ็กซ์พอร์ตของสหรัฐอเมริกา. IBM ไม่รับประกันเกี่ยวกับเนื้อหาของข้อมูลเหล่านี้. ข้อมูลนี้นำเสนอเนื้อหา "ตามที่เป็น" โดยไม่มีการรับประกันใดๆ, ไม่ว่าจะโดยทางตรงหรือทางอ้อม, รวมถึง และไม่จำกัดอยู่กับ การรับประกันทางนัยในแง่การนำไปจำหน่ายได้และ ด้านความเหมาะสมสำหรับวัตถุประสงค์เฉพาะด้าน.

เนื้อหาทั้งหมดเป็นลิขสิทธิ์ของ IBM Corporation.

ด้วยการดาวน์โหลดและการพิมพ์ข้อมูลจากไซต์นี้, คุณได้ตกลงยอมรับเกณฑ์และเงื่อนไขต่างๆ เหล่านี้.





พิมพีในสหรัฐอเมริกา