



ระบบ IBM - iSeries

การจัดการระบบ

การจัดการระบบงาน

เวอร์ชัน 5 รีลีส 4





ระบบ IBM - iSeries

การจัดการระบบ

การจัดการระบบงาน

เวอร์ชัน 5 รีลีส 4

หมายเหตุ

ก่อนใช้ข้อมูลนี้และผลิตภัณฑ์ที่ข้อมูลนี้สนับสนุน, ต้องแน่ใจว่าได้อ่านข้อมูลใน “หมายเหตุ”, ในหน้า 213.

ฉบับแก้ไขครั้งที่ห้า (กุมภาพันธ์ปี 2006)

ฉบับแก้ไขนี้ใช้ได้กับเวอร์ชัน 5, รีลีส 4, โมดิฟิเคชัน 0 ของ IBM i5/OS System/400 (หมายเลขผลิตภัณฑ์ 5722-SS1) และใช้กับรีลีสและโมดิฟิเคชันต่อจากนี้ทั้งหมด จนกว่าจะมีการระบุเป็นอย่างอื่นในฉบับแก้ไขใหม่. เวอร์ชันนี้ไม่รันบนรุ่น reduced instruction set computer (RISC) ทุกรุ่น และไม่รันบนรุ่น CISC.

© ลิขสิทธิ์ของ International Business Machines Corporation 2004, 2006. สงวนลิขสิทธิ์ทั้งหมด.

สารบัญ

การจัดการระบบงาน	1	การมอนิเตอร์ activity ระบบ	102
มีอะไรใหม่ใน V5R4	1	การจัดการงาน	109
PDF ที่สามารถพิมพ์ได้	3	การจัดการการจัดตารางเวลางาน	132
บทนำการจัดการระบบงาน	3	การจัดการระบบย่อย	163
ระบบของคุณเปรียบเสมือนธุรกิจ	4	การจัดการพูลหน่วยความจำ	180
ช่วงชีวิตของงาน	5	การจัดการคิวงาน	185
วิธีการทำงานให้เสร็จสมบูรณ์	7	การจัดการเอาต์พุตคิว	191
หลักการ	10	การจัดการบันทึกการใช้งาน	194
โครงสร้างระบบของคุณ	10	การจัดการ job accounting	200
ระบบย่อย	13	การอ้างอิง	206
พูลหน่วยความจำ	23	การแก้ไขปัญหา	206
งาน	28	งานของฉันหยุดทำงาน	207
อ็อปชันการจัดตารางเวลางาน	56	งานของฉันประสบปัญหาทำงานช้า.	208
คิวงาน	62	การสืบสวนงานแบบ prestart	209
เอาต์พุตคิว	66	ข้อมูลที่เกี่ยวข้องกับการจัดการระบบงาน	210
บันทึกการใช้งาน	75	ภาคผนวก. หมายเหตุ	213
Job accounting.	83	เครื่องหมายการค้า	215
การจัดการงาน	98	ข้อกำหนดและเงื่อนไข	215
การเปลี่ยนโปรแกรมเริ่มต้น IPL	98		
การเรียกโปรแกรมการกู้คืน IPL พิเศษ	101		

การจัดการระบบงาน

การจัดการระบบงานเป็นองค์ประกอบพื้นฐานที่สำคัญภายในระบบปฏิบัติการของเซิร์ฟเวอร์ iSeries™ .

ฟังก์ชันของการจัดการระบบงานเป็นพื้นฐานของการประมวลผลงาน, การรัน, และการจบงานที่ผ่านเข้ามาในระบบ, บนเซิร์ฟเวอร์ iSeries . ไม่ว่าคุณจะรัน งานแบ็คเอนด์แบบธรรมดาสัปดาห์ละครั้งหรือเรียกใช้แอปพลิเคชันประจำวัน (เช่น Lotus Notes®), การจัดการระบบงานช่วยในการบริหารงานและอ็อบเจกต์ต่างๆ ที่รันบนระบบของคุณ. นอกจากนี้ ยังสนับสนุนคำสั่งและฟังก์ชันภายในที่จำเป็นในการควบคุมการปฏิบัติการของระบบ และจัดสรรรีซอร์สให้กับแอปพลิเคชันเมื่อต้องการอีกด้วย.

เซิร์ฟเวอร์ iSeries ถูก เชื้อต่อและพร้อมใช้งาน. ผู้ใช้งานส่วนใหญ่ไม่จำเป็นต้องเปลี่ยนค่ากำหนดดีฟอลต์แต่ประการใด. อย่างไรก็ตาม, ถ้าคุณต้องการปรับแต่งการจัดการระบบงานให้เหมาะกับบริษัทของคุณ, คุณจะต้องทำความเข้าใจกับคำศัพท์และหลักการต่างๆ ที่เกี่ยวข้อง และวิธีการที่ส่วนประกอบเหล่านี้จะทำงานร่วมกันเพื่อให้เซิร์ฟเวอร์ iSeries มีประสิทธิภาพการทำงานสูงสุด.

ไม่ว่าคุณจะเป็นผู้ที่มีประสบการณ์ในการใช้เซิร์ฟเวอร์ iSeries หรือเพิ่งหัดใช้ก็ตาม, กลุ่มหัวข้อ นี้ช่วยให้คุณเข้าใจภาพการทำงานของการจัดการระบบงานได้อย่างง่ายดาย. หัวข้อนี้มี entry point ต่างๆ , เพื่อให้คุณสามารถเลือกจุดเพื่อเริ่มต้นเรียนรู้เกี่ยวกับการจัดการระบบงาน.

หมายเหตุ: นอกจากนี้, คุณยังสามารถทำงานกับการจัดการระบบงานโดยใช้ iSeries Navigator tasks on the Web. สิ่งนี้ช่วยให้คุณสามารถทำงานกับฟังก์ชันการจัดการระบบงานโดยใช้เว็บเบราว์เซอร์ได้. สำหรับข้อมูลเพิ่มเติม, ให้ดู iSeries Navigator tasks on the Web.

มีอะไรใหม่ใน V5R4

รีลีสนี้มีการเปลี่ยนแปลงที่น่าสนใจไม่เฉพาะในฟังก์ชันการจัดการระบบงาน เท่านั้น, แต่ยังมีเปลี่ยนแปลงที่หัวข้อ Work management ใน Information Center ด้วย.

ขณะนี้ คุณสามารถควบคุมการจับทำบันทึกการใช้งานได้มากขึ้น.

สภาพการคงค้างบันทึกการใช้งานมีใช้กันเป็นเวลาหลายปีแล้ว. เมื่อบันทึกการใช้งาน อยู่ในสภาพคงค้างบันทึกการใช้งาน, จะไม่มีการจับทำบันทึกการใช้งาน. ด้วยรีลีส V5R4 iSeries Navigator หน้าต่าง Job Properties - Job Log ได้มีการพัฒนาขึ้นเพื่อให้คุณสามารถควบคุม วิธีการและสถานการณ์ที่จะมีการสร้างบันทึกการใช้งานสำหรับงานเฉพาะได้.

ถ้าคุณ ต้องการใช้อินเตอร์เฟซแบบอักษร, คำสั่ง Work with Job Logs (WRKJOBLOG) เป็นคำสั่งใหม่ที่คุณสามารถใช้ในการจัดการบันทึกการใช้งานได้. เมื่อรัน คำสั่ง นี้, จอภาพ Work with Job Logs จะแสดงขึ้น. จากจอภาพนี้ คุณสามารถแสดงบันทึกการใช้งาน, ลบบันทึกการใช้งาน, ทำงานกับสพูลไฟล์, และ ทำงานกับงานที่เชื่อมโยงกับบันทึกการใช้งานได้.

ในรายละเอียดของงาน, แอ็ตทริบิวต์ใหม่ LOGOUTPUT ระบุว่าบันทึกการใช้งาน จะถูกจัดทำโดยเซิร์ฟเวอร์บันทึกการใช้งาน, โดยตัวงานเอง, หรือไม่ถูกจัดทำเลย. ถ้ารายละเอียดของงานระบุ *SYSVAL สำหรับค่า LOGOUTPUT, ค่ากำหนดของระบบ Job log output (QLOGOUTPUT) ใหม่จะถูกใช้ในการควบคุมวิธีการจับทำบันทึกการใช้งาน.

ขณะนี้ข้อมูลเพิ่มเติมสำหรับ call stacks มีพร้อมใช้งานแล้ว.

ใน V5R4, call stack ถูกเปลี่ยนบนทั้ง iSeries Navigator และอินเตอร์เฟซแบบอักษร เพื่อเพิ่มการสนับสนุนสำหรับ i5/OS PASE, Java™, และแอ็พพลิเคชัน LIC. ฟิลด์เดิมหลายฟิลด์ถูกเปลี่ยนชื่อ, จัดโครงสร้างใหม่, หรือ โอเวอร์โหลดเพื่อแสดงข้อมูลหลากหลายมากขึ้น.

ขณะนี้คำสั่ง Change Job (CHGJOB) และ End Job (ENDJOB) สามารถ ดำเนินการบนงานปัจจุบันหรืองานที่เสร็จสมบูรณ์แล้ว.

ในรีลีสก่อนหน้านี้, คำสั่งไม่ทำงานสำหรับงานพิเศษ เช่น งานระบบ, งานการมอนิเตอร์ระบบย่อย, spool readers, และ spool writers. การเปลี่ยนแปลงนี้ทำให้ระบบสามารถจัดการเอาต์พุต (สพูล) ไฟล์ได้อย่างเป็นเอกภาพ.

จอภาพ Work with Active Jobs (WRKACTJOB) ได้รับการพัฒนาขึ้น.

ขณะนี้คุณสามารถดูโปรไฟล์ผู้ใช้ปัจจุบัน บนจอภาพ Work with Active Jobs. แอ็ททริบิวต์นี้มีพร้อมใช้งานอยู่แล้ว ใน Active Jobs list ของ iSeries Navigator.

มีการพัฒนาหลายอย่างใน Advanced Job Scheduler.

- Work Flow Manager คือเครื่องมือใหม่ที่ช่วยให้คุณกำหนดหน่วยงาน ที่สามารถประกอบด้วยชุดของขั้นตอนแบบอัตโนมัติและขั้นตอนแบบทำเอง. จากนั้นสามารถ จัดตารางเวลาหรือเริ่มต้นหน่วยงานด้วยตนเองได้.
- คุณสามารถใส่เพิ่มหลายคำสั่งที่ภารกิจที่จัดตารางเวลา.
- คุณสามารถควบคุม notification banner ได้มากขึ้น.
- คุณสามารถใช้ Advanced Job Scheduler เพื่อส่งอีเมล.
- คุณสามารถแจกจ่ายรายงานผ่านทาง Basic Operations container.

หัวข้อ Work management ทั้งหมดถูกจัดโครงสร้างใหม่และอัปเดตสำหรับ IBM® iSeries Information Center Version 5 Release 4 (V5R4).

หัวข้อ Work management ใหม่ใช้แทนคู่มือการจัดการระบบงาน V4R5 ที่พร้อมใช้งานใน IBM iSeries Information Center Version 5 Release 3 (V5R3). คู่มือนี้นำเสนอข้อมูลเกี่ยวกับวิธีการจัดการเวิร์กโหลตระบบอย่างมีประสิทธิภาพ โดยการเปลี่ยนอ็อบเจ็กต์การจัดการระบบงานให้เหมาะสมกับความต้องการของคุณ. คู่มือนี้ยังให้แนวทางการปรับประสิทธิภาพ, รายละเอียดของค่ากำหนดของระบบ, ข้อมูลเกี่ยวกับ การรวบรวมข้อมูลประสิทธิภาพ, การเก็บข้อมูลการใช้งานของระบบ, การใช้ work entries, และการจัดตารางเวลางานแบ็ตซ์.

วิธีการดูว่ามีอะไรใหม่หรือเปลี่ยนไป

- รูปภาพ ➤ ทำเครื่องหมาย ตำแหน่งเริ่มต้นของข้อมูลใหม่หรือข้อมูลที่เปลี่ยนแปลง.
- รูปภาพ ⏪ ทำเครื่องหมาย ตำแหน่งสิ้นสุดของข้อมูลใหม่หรือข้อมูลที่เปลี่ยนแปลง.

PDF ที่สามารถพิมพ์ได้

ใช้ไฟล์นี้เพื่อดูและพิมพ์ PDF ของข้อมูลนี้.

เพื่อดูหรือดาวน์โหลดเวอร์ชัน PDF ของเอกสารนี้, ให้เลือก การจัดการระบบงาน (ประมาณ 2000 KB).

คุณสามารถดูหรือดาวน์โหลดหัวข้อที่เกี่ยวข้องเหล่านี้:


- Performance Tuning (1700 KB) มีหัวข้อดังต่อไปนี้:
 - แผนประสิทธิภาพ
 - การจัดการประสิทธิภาพของ iSeries
 - แอปพลิเคชันสำหรับการจัดการประสิทธิภาพ
- Management Central (810 KB) มีหัวข้อดังต่อไปนี้:
 - การจัดการผู้ใช้และกลุ่ม
 - การรวบรวมข้อมูลที่มีอยู่และประสิทธิภาพ
 - การจัดการค่ากำหนดของระบบ
 - ฟังก์ชันการซิงโครไนส์
 - การจัดแพ็คเกจและการส่งอีอบเจกต์
 - คำสั่งการรัน
 - การจัดตารางเวลาภารกิจหรืองานของคุณ
 - การรักษาความปลอดภัยระบบระดับต้นและการวางแผน

การบันทึกไฟล์ PDF

ถ้าต้องการบันทึก PDF บนเวิร์กสเตชันของคุณ สำหรับการดูและการพิมพ์ให้ทำดังนี้:

1. คลิกขวาที่ PDF ในบราวเซอร์ของคุณ (คลิกขวาที่ลิงก์ข้างต้น).
2. คลิกอ็อปชันที่จะบันทึก PDF แบบโลคัล.
3. นำทางไปที่ไดเรกทอรีซึ่งคุณต้องการบันทึก PDF.
4. คลิก Save.

การดาวน์โหลด Adobe Reader

- | คุณต้องติดตั้ง Adobe Reader บนระบบของคุณเพื่อดูหรือพิมพ์ PDFs เหล่านี้. คุณสามารถดาวน์โหลด ก็อปปีฟรีจาก Adobe
- | Web site (www.adobe.com/products/acrobat/readstep.html)  .

บทนำการจัดการระบบงาน

การจัดการระบบงานสนับสนุนคำสั่งและฟังก์ชันภายในที่จำเป็น สำหรับการควบคุมการดำเนินการระบบและเวิร์กโหลดประจำวันบนระบบ. นอกจากนี้, การจัดการระบบงานยังมีฟังก์ชันที่คุณต้องใช้ในการแจกจ่ายริชอร์สสำหรับแอปพลิเคชันของคุณ เพื่อให้ระบบของคุณสามารถจัดการแอปพลิเคชันได้.

วัตถุประสงค์ของระบบของคุณคือการปฏิบัติงาน. งานเข้าสู่, งานถูกประมวลผล, และงานออกจากระบบ. ถ้าคุณคิดถึงการจัดการระบบงานเป็นสามระยะเหล่านี้, จะทำให้เข้าใจการจัดการระบบงานได้ง่ายขึ้น. การจัดการระบบงานอธิบายว่างานจะเข้าสู่ระบบที่ใด, งานจะถูกประมวลผลที่ใดและด้วยรีซอร์สใด, และเอาต์พุตจากงานจะไปที่ไหน.

คุณยังไม่เคยใช้การจัดการระบบงานใช่หรือไม่? กลุ่มหัวข้อข้างใต้เรื่อง บทนำการจัดการระบบงาน ได้รับการออกแบบมาเพื่อนำเสนอข้อมูลภาพรวมในด้านต่างๆ ของการจัดการระบบงาน. ในวิธีนี้, ช่วยให้คุณสามารถ เข้าใจพื้นฐานในหลักการของการจัดการระบบงาน, ไม่ว่าคุณจะมีแบ็กกราวนด์เกี่ยวกับระบบมากเพียงใดก็ตาม.

ระบบของคุณเปรียบเสมือนธุรกิจ

เพื่อให้เข้าใจหลักการโดยรวมของการจัดการงานได้ง่ายขึ้น, ให้ลองเปรียบเทียบระบบของคุณเป็นเสมือนธุรกิจอย่างหนึ่ง.

ระบบพื้นฐานสามารถเปรียบเป็นธุรกิจขนาดเล็ก, และระบบที่ซับซ้อนสามารถเปรียบได้เป็นห้างสรรพสินค้า. สมมุติว่ามีร้านค้าขนาดเล็กที่ทำธุรกิจ การประกอบเฟอร์นิเจอร์ไม้แกะสลัก. *งานที่เข้าสู่ระบบ*, เช่น ไม้สั่งซื้อ โต๊ะขนาดเล็ก, เก้าอี้, และชั้นหนังสือ. *งานที่ประมวลผล*, ช่างไม้โทรหาลูกค้า เพื่อให้รับรองไม้สั่งซื้อ, และปรึกษาเกี่ยวกับการออกแบบ เช่น สไตล์, ขนาด, และสี. ช่างไม้ออกแบบเฟอร์นิเจอร์แต่ละชิ้น, รวบรวมวัสดุที่จำเป็น, แล้วผลิตเฟอร์นิเจอร์. หลังจากที่เฟอร์นิเจอร์เสร็จสมบูรณ์แล้ว, เฟอร์นิเจอร์จะถูกจัดส่งให้ลูกค้า: *งานออกจากระบบ*.

เนื่องจากระบบที่ซับซ้อนคือระบบที่ประกอบไปด้วยระบบพื้นฐานหลายระบบ, สิ่งที่เปรียบเทียบได้กับระบบที่ซับซ้อนคือห้างสรรพสินค้า, ซึ่งมีธุรกิจขนาดเล็กและขนาดใหญ่หลายธุรกิจในพื้นที่หนึ่ง. อาจเป็นไปได้ที่ช่างไม้มีร้านทำเฟอร์นิเจอร์อยู่ในมุมด้าน ตะวันตกเฉียงเหนือของห้างสรรพสินค้า และคนทำขนมมีร้านเบเกอรี่อยู่ทางทิศตะวันออก. คนทำขนม และช่างไม้มีอินพุตและเอาต์พุตที่แตกต่างกัน, นั่นคือ, ไม้สั่งและ ผลิตภัณฑ์ของพวกเขาจะไม่เหมือนกัน. ยิ่งกว่านั้น, เวลาในการทำงานของแต่ละธุรกิจค่อนข้างแตกต่างกัน, และผู้ใช้ของพวกเขาทราบและ เข้าใจเรื่องนี้ดี.

คำศัพท์ในการจัดการระบบงาน

ระบบที่ซับซ้อน (ห้างสรรพสินค้า) คือที่ที่รวบรวมระบบพื้นฐานหลายระบบ (ร้านค้า). ระบบพื้นฐานเหล่านี้เรียกว่า *ระบบย่อย*.

ชิ้นงานใดๆ ภายในธุรกิจ ถือว่าเป็น *งาน*. ตัวอย่างของชิ้นงานอาจจะเป็น จุดหมายของลูกค้า, การโทรศัพท์, ไม้สั่งซื้อ, หรือการทำความสะอาดตอนกลางคืน. ในระบบ IBM iSeries ก็เช่นกัน. บนระบบ, งานแต่ละงานมีชื่อที่ไม่ซ้ำกัน.

รายละเอียดของงาน อธิบายวิธีการจัดการงาน ที่เข้ามาในระบบย่อย. รายละเอียดของงานมีชั้นของข้อมูล เช่น user IDs, คิวงาน, และข้อมูลการเรอต์. ข้อมูลในรายละเอียดของงานอาจ เปรียบเทียบได้กับรายละเอียดของงานในธุรกิจขนาดเล็ก.

ธุรกิจ มีลักษณะอย่างไร? ทุกร้านค้ามีพิมพ์เขียวหรือแผนผังร้านค้า. แผนผังเหล่านี้ใช้สำหรับการอธิบาย, โดยมีรายละเอียดแตกต่างกันไป, ถึง องค์ประกอบทางกายภาพของธุรกิจ. ธุรกิจอาจมีร้านค้าที่มีสิ่งดังต่อไปนี้: พื้น 2 ชั้น, ประตู 5 บาน, ตู้ไปรษณีย์ 3 ตู้, และโทรศัพท์ 2 เครื่อง. บนระบบ iSeries, *subsystem description* มีข้อมูลทั้งหมดเกี่ยวกับระบบย่อย.

งานมาจากไหน? สำหรับช่างไม้, งานมาจากโทรศัพท์ของลูกค้า, จากการอ้างอิง, และจากบุคคลที่วะชมเฟอร์นิเจอร์. บนระบบ iSeries, งานสามารถมาได้จาก หลายที่. ตัวอย่างเช่น คิวงาน, เวิร์กสเตชัน, การสื่อสาร, งานแบบ autostart, และงานแบบ prestart.

งานหาพื้นที่ว่างได้อย่างไร? ภายใน ห้องสรรพสินค้า, แต่ละธุรกิจ (ระบบย่อย) มีพื้นที่ร้านจำนวนหนึ่ง. บนระบบ iSeries , *พูลหน่วย ความจำ* ช่วยให้คุณสามารถควบคุมแหล่งเก็บข้อมูลหลัก (หรือพื้นที่ร้าน) ที่แต่ละระบบย่อย (ธุรกิจ) สามารถใช้ทำงานของตนได้. ร้านค้ายังมีพื้นที่ร้าน (ระบบย่อย) มาก, ยิ่งสามารถรองรับลูกค้า, หรืองาน, ได้มากขึ้น.

งานเข้ามาได้อย่างไร? ลูกค้าที่ไม่สามารถหาร้านค้าที่ต้องการได้ อาจพบว่าบูธข้อมูล สามารถช่วยให้ข้อมูลทิศทางและทำให้เขาไปยังร้านค้าที่ต้องการได้. เช่นเดียวกับใน ระบบ iSeries. *Routing entries* คล้ายกับไดเรกทอรีร้านค้าหรือบูธข้อมูล. หลังจาก queuing routing entry, entry จะช่วยแนะนำให้งานทราบถึงตำแหน่งที่ถูกต้องของงาน. อย่างไรก็ตาม, ต้องพบ routing entry ก่อน. สามารถพบได้โดยใช้ *ข้อมูล การเรดท์*. ข้อมูลการเรดท์คือสิ่งที่งานใช้ในการค้นหา routing entry ที่ถูกต้อง.

งานถูกจัดการอย่างไร? ช่างไม้ต้องจัดระดับความสำคัญของแต่ละงาน. แก้วที่ต้องจัดส่งลูกค้าตอนปลายสัปดาห์ควรจะทำก่อนชิ้นวางหนังสือที่มีกำหนดส่ง ตอนสิ้นเดือน. บนระบบ iSeries , *คลาส* ให้ข้อมูล เกี่ยวกับวิธีการจัดการงานในขณะที่อยู่ในระบบย่อย. ข้อมูลนี้รวมถึง ระดับความสำคัญในขณะที่รัน, หน่วยเก็บสูงสุด, เวลา CPU สูงสุด, และการแบ่ง เวลา. แต่ละแอตทริบิวต์เหล่านี้ประกอบกันเป็นวิธีการและเวลาที่งานจะถูกประมวลผล.

เช่นเดียวกับที่มีกฎที่ส่งผลต่อร้านค้าทั้งหมดในห้าง, บนระบบ iSeries มีกฎที่ส่งผลต่อระบบย่อยทั้งหมดเช่นกัน. ตัวอย่างของกฎเหล่านี้คือ *ค่ากำหนดของระบบ*. ค่ากำหนดของระบบคือชั้นของข้อมูลที่ ใช้กับระบบทั้งระบบ. ค่ากำหนดของระบบรวมถึง ข้อมูล เช่น, วันที่และ เวลา, ข้อมูล configuration, ข้อมูล signon, การรักษาความปลอดภัยระบบและ, การจัดการหน่วยเก็บ.

ลูกค้าแต่ละรายในห้างสรรพสินค้ามีข้อมูลเฉพาะของตน. บนระบบ iSeries , *โปรไฟล์ผู้ใช้* มีข้อมูลเฉพาะสำหรับผู้ใช้. ในลักษณะคล้ายกับบัตรเครดิตของลูกค้า, โปรไฟล์ผู้ใช้ให้สิทธิเฉพาะของผู้ใช้ และกำหนดแอตทริบิวต์ผู้ใช้สำหรับงานของผู้ใช้นั้น. *แอตทริบิวต์งาน* เหล่านี้แนะนำเสนอข้อมูลที่รวมถึง, แต่ไม่จำกัด เฉพาะ, รายละเอียดของงาน, เอาต์พุตคิวหรืออุปกรณ์การพิมพ์, message queue, โค้ดบัญชีผู้ใช้, และระดับความสำคัญในการจัดตารางเวลา.

ช่วงชีวิตของงาน

เพื่อให้เข้าใจพื้นฐานของการจัดการระบบงานของ iSeries , ให้ลองติดตามงานแบ็คซ์แบบธรรมดา เมื่องานดำเนินการผ่านระบบ.

ช่วงชีวิตของงานแบ็คซ์แบบธรรมดาจะเริ่มต้นขึ้นเมื่อคุณได้ส่งงานดังกล่าวเข้าสู่ระบบ. จากนั้นงานจะถูกส่งเข้าไปในคิวงานที่ซึ่งงานจะไปรอเข้าสู่ระบบย่อยเพื่อการรันต่อไป. หลังจากทำงานย้ายไปที่ระบบย่อย งานจะได้รับการจัดสรรหน่วยความจำเพื่อใช้รัน. จากนั้น printer output file (เรียกอีกอย่างว่าสพูลไฟล์) จะถูกส่งไปที่เอาต์พุตคิว เพื่อรอคำสั่งต่อไปว่าจะทำอะไร (เช่น, การพิมพ์). แม้ว่างานทั้งหมดไม่ได้มีพารามิเตอร์ที่เหมือนกับพารามิเตอร์ทุกอย่าง, แต่คุณสามารถเข้าใจวิธีการที่จะทำให้งานอื่นเสร็จสมบูรณ์บนระบบได้ดีขึ้น โดยการเรียนรู้เพิ่มเติมเกี่ยวกับวงจรชีวิตงานปกตินี้.

ส่งงาน → งานเข้าสู่ คิวงาน → งานเข้าสู่ระบบย่อย → พูล หน่วยความจำจัดสรรหน่วยความจำไปที่ระบบย่อย → งานเสร็จสิ้น และย้ายไปที่เอาต์พุตคิว

ช่วงชีวิตของงาน: การส่งงาน

เมื่องานถูกส่ง, งานจะถูกสร้างขึ้นและเข้าสู่ระบบ. ในเวลานี้, แอตทริบิวต์จะถูกกำหนดให้กับงาน.

รายละเอียดของงานมี *แอตทริบิวต์* ที่งานจะใช้ เพื่อดำเนินการผ่านวงจรชีวิตการจัดการระบบงาน. แอตทริบิวต์เหล่านี้รวมถึง โปรไฟล์ผู้ใช้ที่งานจะเริ่มรันอยู่ภายใต้, ข้อมูลคำขอ (ซึ่งบอกงานว่า จะทำอะไร), และส่วน initial user ของรายชื่อไลบรารี, และอื่นๆ. รายละเอียดของงานยังเก็บข้อมูลที่บอกให้งานรู้ว่าจะต้องเข้าไปอยู่ในคิวงานใด และข้อมูลการเรดท์. ข้อมูลการเรดท์

จะถูกใช้งานโดยระบบย่อยในภายหลังเพื่อค้นหา routing entry ที่ประกอบไปด้วยข้อมูลที่งานจำเป็นต้องใช้ในการสตาร์ทเพื่อจะทำงาน. เอดต์พุตคิวยังถูกกำหนดเอาไว้ใน รายละเอียดของงานด้วย. เอดต์พุตคิวเป็นตัวบอกว่าพรินเตอร์เอดต์พุต (หรือเรียกว่าไฟล์สพูล) จากงานจะออกไปที่จุดใด.

หลังจากที่งานได้รับค่า (การกำหนดค่าเริ่มต้น, การปรับแต่งค่า) สำหรับแอ็ททริบิวต์ งาน, งานจะย้ายไปที่คิวงานที่ซึ่งงานจะรอเพื่อเข้าสู่ระบบย่อย.

ช่วงชีวิตของงาน: งานเข้าสู่คิวงาน

คิวงาน เป็น entry point ของการทำงานสำหรับงานแบ็คซ์ที่จะเข้าสู่ระบบ. โดยอาจเปรียบเทียบเสมือนกับเป็น "ห้องรับรอง" ของระบบย่อย.

มีปัจจัยหลายประการที่มีผลกระทบ เมื่องานถูกดึงออกจากคิวงาน เพื่อส่งต่อไปยังระบบย่อย, เช่น ระดับความสำคัญของงานในคิวงาน, หมายเลขลำดับของคิวงาน, และจำนวนงานแอ็คทีฟสูงสุด. เมื่อปัจจัยเหล่านี้ประกอบเข้าด้วยกัน, งานจะถูกดึงออกจากคิวงานเพื่อเริ่มการทำงานในระบบย่อยได้.

เมื่องานเข้าสู่คิวงานแล้ว, งานจะพร้อมทำงานให้กับ ระบบย่อยที่มีคิวงานที่จัดสรรให้. เพราะ ระบบย่อยอาจมีคิวงานที่ป้อนให้มากกว่าหนึ่งคิว (อย่างไรก็ตาม, คิวงานไม่สามารถป้อนเข้าไปยังระบบย่อยได้มากกว่าหนึ่งระบบ), หมายเลขลำดับในระบบย่อยจะกำหนดว่าเมื่อใดที่ระบบย่อย จะประมวลผลคิวงาน. ระบบย่อยดูหมายเลขลำดับของคิวงาน ก่อนดูระดับความสำคัญของงานในคิวงาน. ระบบย่อย ใช้ระดับความสำคัญบนคิวงานเพื่อกำหนดว่างานสามารถเข้าสู่คิวงานได้เมื่อใดเมื่อ เปรียบเทียบกับงานอื่น. ระดับความสำคัญของงานและจำนวนงานแอ็คทีฟสูงสุดเป็นตัวกำหนดว่า เมื่อใดที่งานจะสามารถเข้าสู่ระบบย่อยได้.

ช่วงชีวิตของงาน: งานเข้าสู่ระบบย่อย

ระบบย่อย คือสถานะแวดล้อมการปฏิบัติการที่ระบบ จัดการรีซอร์สที่งานใช้และควบคุมงานที่รันอยู่ภายใน ระบบย่อย. หลังจากทำงานรันอยู่ในระบบย่อยแล้ว, งานระบบย่อยจะดำเนินการงานตาม คำขอของผู้ใช้ เช่น การพัก, การรีลีส, และการจบงาน. เมื่องาน เข้าสู่ระบบย่อย งานจะกลายเป็นแอ็คทีฟ.

เช่นเดียวกันกับงาน, ระบบย่อยจะมี description ที่เก็บข้อมูลสำคัญเพื่อใช้ในการทำงานให้สำเร็จ. ใน subsystem description คือ routing entry. routing entry อ้างอิงคลาสอ็อบเจกต์, ซึ่งมีแอ็ททริบิวต์ ที่ควบคุมสถานะแวดล้อมรันไทม์. อย่างไรก็ตาม, ก่อนที่งานจะได้รับ routing entry, ข้อมูลการเรดต์จะต้องจับคู่กับค่าเปรียบเทียบที่อยู่ใน routing entry เสียก่อน. หากไม่เกิดการจับคู่ดังกล่าวแล้ว, งานจะไม่สามารถรันได้.

หลังจากสร้างความสัมพันธ์ระหว่างข้อมูลการเรดต์และ routing entry แล้ว, จะมีการกำหนดคลาสอ็อบเจกต์ที่งานจะใช้. แอ็ททริบิวต์บางอย่าง ที่ควบคุมสถานะแวดล้อมรันไทม์รวมถึง ระดับความสำคัญการรัน, การแบ่งเวลา, ช่วงเวลารอสูงสุด, เวลาประมวลผลสูงสุด, หน่วยเก็บชั่วคราวสูงสุด, และจำนวนสูงสุดของ threads.

subsystem description จะกำหนดพูลหน่วยความจำที่จะถูกจัดสรรให้กับระบบย่อย. subsystem description จะประกอบด้วยจำนวนงานแอ็คทีฟสูงสุดซึ่งก็คือจำนวนสูงสุดที่สามารถมีได้ของงานแอ็คทีฟในช่วงเวลาหนึ่งในระบบย่อย.

งานไม่สามารถรันได้, จนกว่างานจะได้ activity level และถูกกำหนดพูลหน่วย ความจำ. subsystem description, ซึ่งเหมือนกับ job description, จะเก็บข้อมูล, เช่น พูลหน่วยความจำที่จะใช้, routing entry, จำนวนงานแอ็คทีฟสูงสุด, และจำนวนงานแอ็คทีฟสูงสุดที่มีอยู่ในระบบย่อยในปัจจุบัน.

ช่วงชีวิตของงาน: ระบบย่อยจะใช้หน่วยความจำจากพูลหน่วยความจำเพื่อรันงาน

หน่วยความจำคือรีซอร์สจากพูลหน่วยความจำที่ระบบย่อยจะใช้ในการรันงาน. จำนวนของหน่วยความจำในพูลหน่วยความจำ, และจำนวนงานอื่นที่แย่งกันใช้หน่วยความจำ, มีผลกระทบต่อประสิทธิภาพในการรันงาน.

พูลหน่วยความจำจัดสรรหน่วยความจำซึ่งใช้ในการรันให้งาน. มีปัจจัยหลายอย่างส่งผลต่อวิธีการรันงานในพูลหน่วยความจำ, เช่น ขนาดของพูล หน่วยความจำ, activity level ในพูลหน่วยความจำ, และการเพจและ faulting. activity level ในพูลหน่วยความจำมีความสัมพันธ์โดยตรงกับจำนวนของ threads ที่ได้รับอนุญาตให้รันในพูลหน่วยความจำในเวลาหนึ่ง. จำไว้ว่า, งานทุกงานจะต้องมี thread แอคทีฟอย่างน้อยหนึ่งอัน, แต่บางงานสามารถมี thread ได้หลายอัน. Thread จะช่วยให้งานสามารถทำงานหลายๆอย่างได้ในเวลาเดียวกัน. ตัวอย่างเช่น, หนึ่ง thread สามารถออกไปและทำการคำนวณ ในขณะที่ thread อื่นจะรอข้อมูลเพิ่มเติมเพื่อประมวลผล.

การเพจเป็นการเคลื่อนที่เข้าออกหน่วยความจำของข้อมูล, ซึ่งเป็นได้ทั้งแบบซิงโครนัสและอะซิงโครนัส. สามารถเขียนเพจไปยังหน่วยเก็บหรือย้ายออกจากหน่วยความจำโดยไม่ต้องถูกเขียนหากไม่มีการปรับเปลี่ยนใดๆ. Faulting จะทำให้เกิดการเพจบนเซิร์ฟเวอร์. Faulting จะเกิดขึ้นเมื่อน้ำที่อ้างอิง, หรือชั้นของข้อมูล, ไม่ได้อยู่ในหน่วยความจำ. ข้อผิดพลาดนี้อาจทำให้โปรแกรมหยุดทำงานเพราะโปรแกรมต้องรอการเพจข้อมูล.

ระบบย่อยจะใช้พูลหน่วยความจำต่างๆ เพื่อสนับสนุนประเภทต่างๆ ของงานที่จะรันภายในระบบย่อยนั้น.

ช่วงชีวิตของงาน : งานเสร็จสิ้นลงและย้ายไปที่เอาต์พุตคิว

พรินเตอร์เอาต์พุตของงาน (เรียกอีกอย่างว่าสพูลไฟล์) ถูกส่งไปที่เอาต์พุตคิวที่ซึ่งเอาต์พุตจะรอเพื่อให้ส่งต่อไปที่พรินเตอร์หรือไฟล์. เอาต์พุตคิวจะเหมือนกับคิวงานตรงที่เอาต์พุตคิวจะ ควบคุมการส่งเอาต์พุตไปให้กับอุปกรณ์การพิมพ์. เอาต์พุตคิวอนุญาตให้ผู้ใช้ควบคุมว่าไฟล์ใดที่จะถูกพิมพ์ก่อนหลัง.

เอาต์พุตคิว คือพื้นที่ที่ printer output files รอที่จะถูก ประมวลผลและส่งไปที่พรินเตอร์. พรินเตอร์เอาต์พุตจะถูกสร้างขึ้นทั้งโดยระบบหรือโดยผู้ใช้โดยใช้ไฟล์พิมพ์อย่างใดอย่างหนึ่ง. ไฟล์พิมพ์คล้ายกับเท็มเพลต หรือต้นแบบที่ซึ่งมีการตั้งค่าตีพอลต์สำหรับแอตทริบิวต์ของพรินเตอร์เอาต์พุตไว้แล้ว. ไฟล์พิมพ์เป็นจุดเริ่มต้นของวงรอบช่วงชีวิตของพรินเตอร์เอาต์พุต.

ไฟล์พิมพ์มีแอตทริบิวต์เอาต์พุตคิว (OUTQ) และอุปกรณ์การพิมพ์ (DEV), ซึ่งบ่งชี้ว่าพรินเตอร์เอาต์พุตจะถูกกำหนดทิศทางอย่างไร. โดยปกติ การตั้งค่า ตีพอลต์คือ *JOB, ซึ่งหมายความว่าแอตทริบิวต์งานของเอาต์พุตคิว และอุปกรณ์การพิมพ์เป็นตัวกำหนดทิศทางของพรินเตอร์เอาต์พุต. แอตทริบิวต์งานของเอาต์พุตคิวและการตั้งค่าอุปกรณ์การพิมพ์คือข้อมูลที่ได้มาเมื่องานถูกสร้างขึ้น. ข้อมูลนี้สร้างขึ้นจากข้อมูลที่ได้มาจาก โปรไฟล์ผู้ใช้ที่ทำงานกำลังรันอยู่ภายใต้, รายละเอียดของงาน, workstation device description, และค่ากำหนดของระบบ Default printer (QPRTDEV).

เมื่อพร้อมที่จะสร้างพรินเตอร์เอาต์พุต, ระบบจะตรวจสอบไฟล์พิมพ์และแอตทริบิวต์งาน (ในลำดับนี้) เพื่อดูว่าเอาต์พุตคิวใดจะประมวลผลพรินเตอร์เอาต์พุตและอุปกรณ์การพิมพ์ใดที่ระบบจะใช้งาน. หากไม่พบ เอาต์พุตคิวที่ระบุ, พรินเตอร์เอาต์พุตจะถูกกำหนดทิศทางไปที่ QGPL/QPRINT.

หลังจากที่พร้อมจะพิมพ์ printer output file, งาน writer, งาน ที่ประมวลผลพรินเตอร์เอาต์พุตจากเอาต์พุตคิวไปที่อุปกรณ์การพิมพ์, จะนำข้อมูลจาก printer output file และส่งไปที่พรินเตอร์ที่กำหนด.

วิธีการทำงานให้เสร็จสมบูรณ์

ใช้ข้อมูลนี้เพื่อเรียนรู้เกี่ยวกับว่างานคืออะไร, ต้องตั้งค่าอะไรก่อน จึงสามารถเริ่มงานได้, งานเดินทางผ่านระบบอย่างไร, และเกิดอะไรขึ้นกับงานหลังจากที่รันแล้ว.

งานคืออะไร

บนเซิร์ฟเวอร์ iSeries, งานจะถูกทำอยู่เสมอ, ไม่ว่า คุณจะเริ่มงานหรือระบบจะเริ่มงานก็ตาม. action ใดๆ บนเซิร์ฟเวอร์ iSeries มีงานบางชนิดที่กำลังปฏิบัติเพื่อให้ action เสร็จสมบูรณ์.

งานจะถูกทำเมื่อคุณเปิดระบบ ขึ้น, เมื่อคุณเปิดไฟล์, หรือเมื่อคุณเคียวรีฐานข้อมูล. ชิ้นงาน (work) แต่ละชิ้นบนระบบจะถูกกระทำผ่านทางงาน (job). งานอาจเป็นสิ่งที่ง่าย ๆ เช่นแอ็พพลิเคชันที่รอให้ผู้ใช้เรียก หรืออาจเป็นสิ่งที่ซับซ้อนเช่นเคียวรีระบบที่รันอย่างต่อเนื่องซึ่งจะมอนิเตอร์จำนวนของผู้ใช้บนระบบทุกชั่วโมง. งานบางงาน, โดยเฉพาะงานแบ็คซ์ และงานแบบโต้ตอบ, มีรายละเอียดของงานที่เชื่อมโยงซึ่งบอกว่า งานจะรันเมื่อไรและรันที่ใด.

งานประกอบด้วยโปรแกรมต่างๆ ที่จะทำฟังก์ชันบางอย่างเฉพาะ. ไม่มีขีดจำกัดถึงจำนวน ฟังก์ชันที่งานจะสามารถใช้ปฏิบัติงานได้. งานประกอบด้วยคำสั่งตามขั้นตอนที่จะต้อง กระทำเพื่อให้งานเสร็จสมบูรณ์. หรือก็คือโปรแกรมที่ทำให้งานรันไปตามลำดับอย่างเจาะจง. (ตัวอย่างเช่น, โปรแกรม A จะต้องรันก่อน โปรแกรม B จึงจะสามารถเริ่มได้.) Threads ช่วยให้งานสามารถทำงานของตน ได้เสร็จสมบูรณ์. งานแอ็คทีฟ จะประกอบด้วย thread อย่างน้อยหนึ่งตัว. เมื่องานมี thread อยู่หลายตัว, ก็จะทำให้งาน สามารถทำสิ่งต่างๆ ได้มากขึ้นในเวลาเดียวกัน. ตัวอย่างเช่น, thread อันหนึ่งสามารถออกไปทำการคำนวณ ในขณะที่อีกอันหนึ่งจะรอข้อมูลเข้ามามากขึ้นเพื่อทำการประมวลผลต่อไป.

สิ่งที่เกิดขึ้นก่อนงานเข้าสู่ระบบ

งานทั้งหมด, ยกเว้นงานระบบ, จะรันภายในระบบย่อย. เพื่อให้งานเริ่มในระบบย่อยที่แอ็คทีฟ, จำเป็นต้องสร้างพูลหน่วยความจำและซอร์สของ work entry point อย่างน้อยหนึ่งซอร์ส. คิวงานคือตัวอย่างของ ซอร์สของงาน.

เซิร์ฟเวอร์ iSeries จัดส่งมาพร้อมกับชุดดีฟอลต์ของคิวงาน, ระบบย่อย, และพูลหน่วยความจำ, ซึ่งช่วยให้งานสามารถเริ่มต้นได้ในทันทีที่เปิด เครื่อง.

คุณสามารถปรับแต่งระบบย่อยและ configuration ของพูลหน่วยความจำเพื่อให้ เซิร์ฟเวอร์ iSeries ของคุณ มีความสามารถและประสิทธิภาพสูงสุด. ตัวอย่างเช่น, หากงานแบ็คซ์มีความสำคัญต่อความสำเร็จของธุรกิจ, คุณอาจต้องการจัดสรรหน่วยความจำให้กับ งานแบ็คซ์มากขึ้น. หรือ, คุณอาจกำหนดจำนวนงานที่จะรันในเวลาหนึ่งในระบบย่อย Qbatch ของคุณ ให้น้อยลง เพื่อให้งานสามารถใช้ปริมาณรีซอร์สในการรันได้มากขึ้น. นอกจากนี้, คุณสามารถสร้างคิวงาน, ระบบย่อย และพูลหน่วยความจำ โดยออกแบบให้ทำงานชนิดใดชนิดหนึ่งเป็นการเฉพาะ. ตัวอย่างเช่น, คุณสามารถสร้างคิวงานชื่อ Nightreps, เพื่อให้รายงานแบบแบ็คซ์เป็นรายคืนถูกส่งไปยังระบบย่อยที่ชื่อ Nightrep ซึ่งจะจัดสรร หน่วยความจำไว้เป็นการเฉพาะเพื่อใช้รันงานแบ็คซ์เหล่านี้.

งานเข้าสู่ระบบได้อย่างไร

work entry จะบอกถึงต้นทางที่งานได้เข้ามาสู่ระบบย่อยเพื่อรอให้เรียกทำงาน. งานแต่ละชนิดบน iSeries มีชนิดของ work entries ที่จะใช้แตกต่างกัน.

ตัวอย่างเช่น, งานแบ็คซ์ส่วนใหญ่จะใช้คิวงานเพื่อเข้าสู่ระบบย่อย. Job queue entries เป็นกลไกหนึ่งที่คิวงานจะถูกกำหนดให้เป็นต้นทางของงาน ที่จะส่งเข้าไปสู่ระบบย่อย.

Work entries ถูกเก็บอยู่ใน subsystem description. ถ้า subsystem description ไม่มี work entry สำหรับชนิดของงานที่กำลังจะถูกรัน, งานดังกล่าวจะไม่สามารถรันในระบบย่อยนั้นได้. ระบบย่อยที่จัดส่งมาจาก IBM มี work entry ดีฟอลต์อยู่ใน subsystem description แล้ว. แต่ต้องไม่ลืมว่า, work entry ดีฟอลต์บางตัวที่จัดส่งมาพร้อมกับระบบย่อยนั้นได้ถูกจัดสรรให้ทำการรันงานแบบเจาะจงเท่านั้น.

งานถูกประมวลผลอย่างไร

เมื่อเซิร์ฟเวอร์ iSeries ถูกเริ่มต้น, งานการมอนิเตอร์ระบบย่อย จะเริ่มการรัน. งานการมอนิเตอร์ระบบย่อยควบคุมงานภายในระบบย่อย. นอกจากนั้น ยังเป็นตัวช่วยสตาร์ทและจบการทำงานให้, พร้อมกับจัดการรีซอร์สให้กับการทำงานนั้นในระบบย่อยอีกด้วย.

งาน (หรือหลายงาน) เข้าสู่ระบบย่อยผ่านทาง work entries ที่ซึ่งงาน จะกลายเป็นแอ็คทีฟและพร้อมที่จะรัน. งานสามารถทำให้เสร็จสมบูรณ์ได้เฉพาะเมื่อ ระบบย่อยได้จัดสรรหน่วยความจำให้รันเท่านั้น. หน่วยความจำถูกจัดสรรให้กับระบบย่อยโดยพูลหน่วยความจำ.

subsystem description ช่วยประมวลผลงานได้อย่างไร

เช่นเดียวกับงาน, ระบบย่อยมี description, ที่เรียกว่า subsystem description. subsystem description มีข้อมูลสำคัญที่บอกวิธี, ตำแหน่ง, และจำนวนงานที่สามารถแอ็คทีฟในระบบย่อยในเวลาเดียวกัน, และรีซอร์สที่สามารถเรียกใช้เพื่อทำงานได้.

Routing entry

routing entry มีอยู่ใน subsystem description และบอกให้ระบบย่อยทราบ ถึงโปรแกรมที่จะรันสำหรับงาน, พูลหน่วยความจำที่จะรันงานในนั้น, และคลาสอ็อบเจกต์ที่จะใช้ในการรันงาน.

คลาสอ็อบเจกต์

คลาสอ็อบเจกต์กำหนดระดับความสำคัญการรัน, เวลารอดีฟอลต์, การแบ่งเวลา, และแอ็คทีวิตีอื่น. ระดับความสำคัญการรันมีความสำคัญ เพราะเป็นตัวกำหนดว่าเมื่อไรที่งานจะได้เวลาของตัวประมวลผลเพื่อรันงาน. ระดับความสำคัญในการรันอยู่ระหว่าง 0 ถึง 99, โดยที่ 0 มีความสำคัญสูงสุด. (เฉพาะงานระบบเท่านั้นที่จะได้รับ ระดับความสำคัญ 0 เนื่องจากเป็นงานที่รันเซิร์ฟเวอร์ iSeries.)

เมื่องานเข้าสู่ระบบย่อย, ระบบย่อยจะพยายามจับคู่ข้อมูลการเรดกับค่าเปรียบเทียบใน routing entry. หากข้อมูลการเรดและค่าเปรียบเทียบใน routing entry จับคู่กันได้, routing entry จะถูกกำหนดไว้ให้กับงานนั้น. แต่ถ้าจับคู่กันไม่ได้ใน routing entry ใดๆ, งานจะสิ้นสุดลง.

ปัจจัยอื่นที่มีผลกระทบในกรณีที่งานรันอยู่ในระบบย่อย คือ จำนวนงานที่ได้รับ อนุญาตให้มีสถานะแอ็คทีฟในระบบย่อยได้พร้อมกัน (หรือรู้จักในอีกชื่อว่างานแอ็คทีฟสูงสุดในระบบย่อย). เมื่องานที่แอ็คทีฟในระบบย่อยมีมากถึงจำนวนสูงสุดแล้ว, งานอื่นๆ จะไม่สามารถเข้ามาสู่ระบบย่อยได้จนกว่างานแอ็คทีฟจะรันเสร็จสิ้น. หน่วยความจำจะต้องถูกจัดสรรให้กับระบบย่อยเพื่อให้งานรัน. Memory pool activity levels บอกให้เซิร์ฟเวอร์ iSeries ทราบถึง จำนวน threads ที่สามารถแอ็คทีฟได้ภายในพูลหน่วยความจำ. จำไว้ว่า, งานแอ็คทีฟจะประกอบด้วย thread อย่างน้อยหนึ่งอัน. เมื่อ activity level ของพูลหน่วยความจำถึงระดับสูงสุด, งานจะต้องรอให้ thread อื่นใช้งาน activity level ให้เสร็จสิ้นเสียก่อน. ดังนั้น, งานจึงสามารถแอ็คทีฟในระบบย่อยและไม่ได้รัน.

หมายเหตุ: อย่าสับสนระหว่างงานแอ็คทีฟสูงสุดในระบบย่อย และ memory pool activity level.

งานออกจากระบบอย่างไร

เอาต์พุตคิวทำงานคล้ายกับคิวงานตรงที่มันจะจัดลำดับเอาต์พุตที่จะพิมพ์ให้. ทั้งพรินเตอร์เอาต์พุตและเอาต์พุตคิวมีแอ็คทีวิตีบิตต์ ที่จะใช้ในการพิมพ์ข้อมูล.

พรินเตอร์เอาต์พุตมีข้อมูลเอาต์พุตที่รอประมวลผล, เช่น ข้อมูลที่รอพิมพ์. พรินเตอร์เอาต์พุตยังมีข้อมูลที่สำคัญที่ใช้เพื่อจัดลำดับการพิมพ์ด้วย. แอ็คทีวิตีบิตต์พรินเตอร์เอาต์พุตมีเอาต์พุตคิว ที่มีพรินเตอร์เอาต์พุตอยู่, ระดับความสำคัญ, สถานะ และตารางเวลาของพรินเตอร์เอาต์พุต.

เอาต์พุตควมมีแอ็ททริบิวต์ของตัวมันเองที่กำหนดลำดับการประมวลผลไฟล์พรินเตอร์เอาต์พุต. นอกจากนั้นเอาต์พุตควมยังมีสิทธิในการใช้งาน ที่จำเป็นสำหรับทำการเปลี่ยนแปลงกับพรินเตอร์เอาต์พุตและเอาต์พุตควม.

เมื่อพรินเตอร์เอาต์พุตพร้อมที่จะถูกส่งไปยังพรินเตอร์งานแบบ writer จะดึงพรินเตอร์เอาต์พุตขึ้นมา. งานแบบ writer จะนำข้อมูลจากพรินเตอร์เอาต์พุตและเตรียมข้อมูลเอาต์พุตนั้นเพื่อพิมพ์.

หลักการ

ไม่ว่าคุณจะใช้การจัดการระบบงานหรือใช้เครื่องมือการจัดการระบบ งานมาเป็นเวลาหลายปีแล้ว, คุณจะพบสิ่งที่มีประโยชน์ในส่วน หลักการ ของกลุ่มหัวข้อ การจัดการระบบงาน.

โครงสร้างระบบของคุณ

คุณเพิ่งได้รับ iSeries ของคุณ. ระบบยังอยู่ในหีบห่อและคุณพร้อมใช้งานแล้ว - ใช่หรือไม่? ระบบย่อยอะไรที่จัดส่งมาพร้อมกับระบบ? มีโปรแกรมเริ่มต้นใดที่คุณต้องการเปลี่ยนหรือไม่? ส่วนการติดต่อกับผู้ใช้ชนิดใด ที่คุณคาดว่าจะทำงานด้วย?

ระบบย่อยที่ส่งมาพร้อมกับระบบ

subsystem configurations สมบูรณ์ที่ได้มาจาก IBM มีอยู่สองชุด และสามารถใช้ได้โดยไม่ต้องเปลี่ยนแปลงสิ่งใดๆ .

configuration ที่ระบบจะใช้เมื่อระบบเริ่มขึ้น ถูกควบคุมโดย ค่ากำหนดของระบบ Controlling subsystem/library (QCTLSBSD). default configuration ประกอบด้วย subsystem descriptions ดังต่อไปนี้:

Qbase (ระบบย่อยการควบคุม)	Qbase จะสนับสนุนงานโต้ตอบ, แบตช์ และ สื่อสาร. และ Qbase จะมีงาน autostart, ซึ่ง จะสตาร์ทระบบย่อย Qusrwrk, Qserver, และ Qspl โดยอัตโนมัติ.
Qserver	เป็นระบบย่อยของไฟล์เซิร์ฟเวอร์.
Qspl	นี่คือระบบย่อยสพูลที่สนับสนุนงานเครื่องอ่านและงาน writer.
Qsyswrk	นี่คือระบบย่อยของงานระบบ. ซึ่งประกอบด้วยงานที่สนับสนุนฟังก์ชันระบบ ทั้งหลายที่จะสตาร์ทขึ้นโดยอัตโนมัติในการเริ่มเปิดระบบและเมื่อระบบได้ออกจากสภาพถูกจำกัดการทำงาน.
Qusrwrk	นี่คือระบบย่อยของงานผู้ใช้. ซึ่งประกอบด้วยงานที่ถูกเรียกทำงานโดยเซิร์ฟเวอร์เพื่อทำงานในนามของผู้ใช้.

configuration อื่น, ซึ่งได้มาจาก IBM, ประกอบด้วย subsystem descriptions ดังต่อไปนี้:

Qctl (ระบบย่อยการควบคุม)	Qctl มีงานแบบ autostart, ที่จะสตาร์ทระบบย่อย Qinter, Qbatch, Qcmn, Qusrwrk, Qserver และ Qspl ให้โดยอัตโนมัติ.
Qinter	นี่คือระบบย่อยที่สนับสนุนงานแบบโต้ตอบ, ยกเว้นในส่วนของคนโซล.
Qbatch	นี่คือระบบย่อยที่สนับสนุนงานแบตช์.
Qcmn	นี่คือระบบย่อยที่สนับสนุนงานสื่อสาร, เว้นแต่งานสื่อสารในโปรโตคอล TCP/IP. งานสื่อสารเหล่านี้เป็นสิ่งจำเป็นสำหรับโปรโตคอลสื่อสารต่างๆ ที่ระบบ i5/OS™ สนับสนุน.
Qserver	เป็นระบบย่อยของไฟล์เซิร์ฟเวอร์.
Qspl	นี่คือระบบย่อยสพูลที่สนับสนุนงานเครื่องอ่านและงาน writer.
Qsyswrk	นี่คือระบบย่อยของงานระบบ. ซึ่งประกอบด้วยงานที่สนับสนุนฟังก์ชันระบบทั้งหลายที่จะสตาร์ทขึ้นโดยอัตโนมัติในการเริ่มเปิดระบบและเมื่อระบบได้ออกจากสภาพถูกจำกัดการทำงาน.
Qusrwrk	นี่คือระบบย่อยของงานผู้ใช้. ซึ่งประกอบด้วยงานที่ถูกเรียกทำงานโดยเซิร์ฟเวอร์เพื่อทำงานในนามของผู้ใช้.

Qbase configuration ช่วยให้สามารถรันฟังก์ชันเดียวกันทั้งหมด ซึ่งคุณสามารถรันด้วย Qctl configuration และสามารถจัดการได้ง่ายกว่าเนื่องจาก configuration นี้ประกอบด้วยระบบย่อยน้อยกว่า.

Qctl default configuration จะช่วยให้มีการควบคุมส่วนย่อยในการทำงานของระบบ โดยแยกย่อย activity ระบบออกเป็นระบบย่อยต่างๆ ทั้งนี้ขึ้นอยู่กับชนิดของ activity. ตัวอย่างเช่น, ถ้าคุณต้องการรันงานแบตช์ในวันหยุด, แต่ไม่ต้องการให้บุคคลใดสามารถ sign on (ยกเว้นที่ คอนโซล), คุณสามารถทำเช่นนั้นได้ง่ายด้วย Qctl configuration โดยการจบ ระบบย่อย Qinter.

หากคุณกำลังพิจารณาการสร้าง subsystem configuration ของคุณเอง, คุณอาจ พบว่าเป็นการง่ายกว่าที่จะใช้ Qctl configuration เป็นจุดเริ่มต้น เมื่อเปรียบเทียบกับ Qbase configuration.

โปรแกรมเริ่มต้น

QSTRUPPGM คือโปรแกรมเริ่มต้น. นี่คือการกำหนดของระบบซึ่ง ระบุชื่อของโปรแกรมที่จะถูกเรียกจากงานแบบ autostart เมื่อระบบย่อยการควบคุม เริ่มต้น. โปรแกรมนี้ปฏิบัติฟังก์ชันการตั้งค่า, เช่น การเริ่ม ระบบย่อยและพรีนเตอร์. ค่ากำหนดของระบบนี้สามารถเปลี่ยนได้โดยเจ้าหน้าที่ดูแลความปลอดภัยหรือโดยบุคคลที่มีสิทธิเจ้าหน้าที่ดูแลความปลอดภัยเท่านั้น. การเปลี่ยนแปลงค่ากำหนดของ ระบบนี้จะมีผลบังคับใช้ในการทำ IPL ครั้งถัดไป.

QSTRUPPGM สามารถมีค่าเหล่านี้:

- QSTRUP QSYS: โปรแกรมที่ระบุถูกรัน อันเป็นผลมาจากการถ่ายโอนการควบคุมไปที่โปรแกรมจากงานแบบ autostart ในระบบย่อยการควบคุม.
- *NONE: งานแบบ autostart สิ้นสุดตามปกติโดยไม่มีการเรียกโปรแกรม.

สิ่งที่เกิดขึ้นในระหว่าง IPL: โปรแกรมเริ่มต้นดีฟอลต์ QSYS/QSTRUP จะทำดังต่อไปนี้:

- เริ่มระบบย่อย QSPL สำหรับงานที่สูญพล.
- รีเซ็ตคิวงาน QS36MRT และ QS36EVOKE ถ้าคิวงานถูกพักไว้ (คิวงานเหล่านี้ ถูกใช้โดยสภาวะแวดล้อม System/36™).
- เริ่มการล้าง Operational Assistant, ถ้าทำได้.
- เริ่ม print writers ทั้งหมดยกเว้นว่าผู้ใช้ได้ระบุไม่ให้ทำบนจอแสดงผล IPL Options.
- เริ่มระบบย่อย QSERVER และ QUSRWRK.
- ถ้าระบบย่อยการควบคุมคือ QCTL, โปรแกรมจะเริ่มระบบย่อย QINTER, QBATCH, และ QCMN.

ชนิดของ start-ups

ในระหว่าง initial program load (IPL), โปรแกรมระบบจะโหลดจาก อุปกรณ์ซอร์สการโหลดที่กำหนดในหน่วยความจำรองของระบบ. ฮาร์ดแวร์ ระบบจะถูกตรวจสอบด้วย. คอนโทรลพาเนลเซิร์ฟเวอร์ iSeries แสดงชุดของโค้ดอ้างอิงของระบบ ที่บ่งชี้สถานะปัจจุบันของเซิร์ฟเวอร์และเตือนคุณ ถึงปัญหาใดๆ . เมื่อ IPL เสร็จสิ้น, อินเทอร์เฟซแบบอักษร จะแสดงบนจอแสดงผล signon, และผู้ใช้สามารถ sign on ด้วย iSeries Navigator.

มีหลายอ็อปชันสำหรับการเริ่มเซิร์ฟเวอร์ iSeries . คุณสามารถ:

- เริ่มระบบโดยไม่เปลี่ยน configuration. สิ่งนี้เรียกว่า *unattended IPL*.
- เปลี่ยน configuration ระบบของคุณในระหว่าง IPL. สิ่งนี้เรียกว่า *attended IPL*.
จอภาพเพิ่มเติมของ attended IPL ขึ้นอยู่กับอ็อปชันที่คุณเลือกบนจอแสดงผลอ็อปชัน IPL. จอเหล่านี้รวมถึงจอแสดงผลที่อนุญาตให้คุณเปลี่ยนค่ากำหนดของระบบและ แอ็ททริบิวต์ระบบอื่นในระหว่าง IPL, reconstruct แอ็คเซสพาธ, ตรวจสอบสถานะ ของข้อจำกัดฟิลิคัลไฟล์, ตั้งค่าและตั้งชื่ออุปกรณ์ใหม่, และระบุ อ็อปชันสำหรับสภาวะแวดล้อมการปฏิบัติการ.
- เปลี่ยนชนิดของ IPL จากคอนโทรลพาเนลระบบของคุณ.
- จัดตารางเวลาให้ระบบปิดและรีสตาร์ท.

ปัญหาทั่วไปในระหว่าง IPL ถูกเรียกว่าเป็น *IPL ผิดปกติ*.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ IPL และการปิดระบบ, ให้ดูหัวข้อ information center , Start and stop the iSeries.

การปิดเครื่อง

การปิดระบบของคุณต้องทำอย่างระมัดระวัง. ถ้าคุณปิดระบบ โดยที่ยังทำภารกิจบางอย่างไม่สมบูรณ์ คุณอาจทำให้ข้อมูลเสียหายหรือ ทำให้ระบบทำงานในลักษณะที่คาดเดาไม่ได้.

หัวข้อศูนย์กลางข้อมูลต่อไปนี้มีข้อมูลเพิ่มเติมเกี่ยวกับการปิดเครื่อง อย่างปลอดภัย.

- วิธีการปิด iSeries อย่างปลอดภัยเมื่อมีการแสดงเซิร์ฟเวอร์ Windows แบบรวม อยู่
- ปิดเครื่องด้วยโลจิคัลพาร์ติชัน
- Power Down System Exit Program
- Exit Program for Tailoring Power Off

iSeries Navigator

iSeries Navigator เป็นอินเทอร์เน็ตเฟสแบบกราฟิกส์ที่เปี่ยมประสิทธิภาพสำหรับโคลเอ็นต์ Windows®. ด้วย iSeries Navigator, คุณสามารถจัดการและดูแลระบบ iSeries ของคุณจากเดสก์ทอป Windows ของคุณ. คุณสามารถใช้ Navigator เพื่อปฏิบัติการกิจส่วนใหญ่ที่เชื่อมโยงกับการจัดการระบบงาน.

อินเทอร์เน็ตเฟสนี้ได้รับการออกแบบมาเพื่อให้คุณสามารถทำงานได้ดีขึ้น. ด้วยเหตุนี้, ขอแนะนำให้คุณใช้ Navigator, ซึ่งมีคำอธิบายทางออนไลน์ที่ช่วยแนะนำคุณ. ในขณะที่อินเทอร์เน็ตเฟสนี้ยังคงอยู่ระหว่างการพัฒนา, คุณอาจยังคงต้องใช้ฮิวเลเตอร์แบบดั้งเดิม เช่น PC5250 เพื่อทำการกิจบางอย่างของคุณ. ถ้าหัวข้ออธิบายถึงภารกิจเช่นนั้น, คุณจะรับคำแนะนำให้ใช้อินเทอร์เน็ตเฟสแบบอักษร ภายในขั้นตอนที่แนะนำหัวข้อ.

ระบบย่อย

ระบบย่อยคือที่ทำงานถูกประมวลผลบนเซิร์ฟเวอร์ iSeries. ระบบย่อยเป็น, สภาวะแวดล้อมการดำเนินการหนึ่งที่ได้กำหนดไว้ก่อนเพื่อให้ระบบประสานเวิร์กโฟลว์และการใช้รีซอร์สให้เหมาะสมกัน. ระบบอาจประกอบด้วยระบบย่อยหลายระบบ, โดยที่ระบบทั้งหมดดำเนินการ เป็นอิสระจากกัน. ระบบย่อยจะจัดการรีซอร์สเอง.

งานทั้งหมด, ยกเว้นงานระบบ, จะรันภายใต้ระบบย่อย. และระบบย่อยแต่ละระบบจะรันการปฏิบัติการที่ไม่เหมือนกัน. ตัวอย่างเช่น, ระบบย่อยหนึ่งระบบ อาจถูกตั้งค่าให้จัดการกับงานแบบโต้ตอบเท่านั้น, ขณะที่ระบบย่อยอื่นจะจัดการกับงานแบ็คซ์เท่านั้น. สามารถออกแบบระบบย่อย ให้จัดการงานได้หลายชนิด. ระบบจะอนุญาตให้คุณกำหนดจำนวนระบบย่อยและชนิดของงานที่ระบบย่อยเหล่านั้นจะจัดการได้อีกด้วย.

ลักษณะรันใหม่ของระบบย่อยถูกกำหนดในอ็อบเจกต์ที่เรียกว่า subsystem description. ตัวอย่างเช่น, ถ้าคุณต้องการเปลี่ยนปริมาณของงาน (จำนวนของงาน) ที่มาจากคิวงานอย่างถาวรเข้าในระบบย่อย คุณ เพียงแต่เปลี่ยน job queue entry ใน subsystem description เท่านั้น.

ระบบย่อยการควบคุม

ระบบย่อยการควบคุมคือ ระบบย่อยแบบโต้ตอบที่เริ่มทำงานโดยอัตโนมัติเมื่อระบบเริ่มขึ้น, และเป็นระบบย่อยซึ่งผู้ควบคุมระบบใช้ในการควบคุมระบบผ่านทางคอนโซลของระบบ. ระบบย่อยการควบคุมถูกระบุในค่ากำหนดของระบบ Controlling subsystem/library (QCTLSBSD).

IBM จัดเตรียม controlling subsystem descriptions ที่สมบูรณ์สองรายการ: QBASE (ระบบย่อย การควบคุมดีฟอลต์) และ QCTL. ในแต่ละครั้ง สามารถมีระบบย่อยการควบคุมที่แอคทีฟ บนระบบได้เพียงหนึ่งระบบย่อยเท่านั้น.

เมื่อระบบอยู่ในสภาพที่ถูกจำกัด, activity ส่วนใหญ่บนระบบจะจบลง, และมีเวิร์กสเตชันที่แอคทีฟเพียงเวิร์กสเตชันเดียวเท่านั้น. ระบบต้องอยู่ใน สภาพนี้สำหรับการรันคำสั่ง เช่น Save System (SAVSYS) หรือ Reclaim Storage (RCLSTG). โปรแกรมการวินิจฉัยปัญหา เกี่ยวกับอุปกรณ์บางโปรแกรมอาจต้องการให้ระบบอยู่ในสภาพที่ถูกจำกัดด้วย. เพื่อสิ้นสุดสภาพนี้, คุณต้องเริ่มระบบย่อยการควบคุมอีกครั้งหนึ่ง.

หมายเหตุ: นอกจากนี้ยังมีข้อจำกัดเกี่ยวกับงานแบ็คซ์ ซึ่งสามารถมีงานแบ็คซ์ที่แอคทีฟ ได้เพียงหนึ่งงานเท่านั้น.

เมื่อระบบย่อยทั้งหมด, รวมถึงระบบย่อยการควบคุมได้สิ้นสุดลงแล้ว, สภาพที่ถูกจำกัดจะถูกสร้างขึ้น. คุณสามารถจบระบบย่อยแต่ละระบบ หรือคุณสามารถใช้ ENDSBS SBS(*ALL) OPTION(*IMMED).

สำคัญ: ระบบไม่สามารถเข้าถึงสภาพที่ถูกจำกัด จนกว่าจะมีงานที่เหลื่ออยู่ในระบบย่อยการควบคุมเพียงงานเดียวเท่านั้น. แต่ในบางครั้งอาจเป็นไปได้ที่แม้ว่าจะมีงานเหลื่ออยู่เพียงงานเดียว, แต่ระบบ ไม่ได้เข้าไปในสภาพที่ถูกจำกัด. ในกรณีนี้ คุณต้องตรวจสอบว่า ไม่มีงานค่าขอระบบที่ถูกหยุดชั่วคราว, งานกลุ่มที่หยุดชั่วคราว, หรืองานที่ตัดการเชื่อมต่ออยู่บนจอแสดงผลแอ็คทีฟที่เหลื่ออยู่. ใช้คำสั่ง Work with Active Jobs (WRKACTJOB) และกด F14=รวม เพื่อแสดงงาน ที่ถูกหยุดชั่วคราวหรืองานที่ตัดการเชื่อมต่อ. ถ้ามีงานเหล่านี้อยู่, คุณต้องจบงาน เพื่อให้ระบบเข้าถึงสภาพที่ถูกจำกัด. ฟังก์ชัน ENDSYS และ ENDSBS จะส่งข้อความ CPI091C ไปให้ผู้ออกคำสั่ง เมื่อมีการ ตรวจพบสภาพนี้.

เพราะเหตุใดจึงพิจารณาการใช้หลายระบบย่อย

เมื่อจำนวนของผู้ใช้บนระบบเพิ่มขึ้น, ระบบย่อยเพียงระบบเดียว มักจะไม่เพียงพอสำหรับงานต่างๆ. โดยการแบ่งผู้ใช้ของคุณลงในหลายระบบย่อย คุณจะได้รับประโยชน์หลายอย่าง.

การจัดการงานได้ดีขึ้น

คุณสามารถควบคุมงานที่กำลังรันในแต่ละระบบย่อยได้ดีขึ้น. ตัวอย่าง เช่น, สำหรับงานเซิร์ฟเวอร์, คุณอาจต้องการแยกงานเซิร์ฟเวอร์ฐานข้อมูล ทั้งหมดไปไว้ที่ระบบย่อยหนึ่ง, งานเซิร์ฟเวอร์คำสั่งรีโมตไปไว้ที่อีกระบบย่อยหนึ่ง, งานเซิร์ฟเวอร์ DDM ไปไว้ที่อีกระบบย่อยหนึ่ง และอื่นๆ. นอกจากนี้, การใช้หลายระบบย่อยยังทำให้คุณสามารถแบ่งกลุ่มของงานด้วยพูลหน่วยความจำ ของงานเองได้ด้วย. ในวิธีนี้, งานกลุ่มหนึ่งจะไม่ส่งผลกระทบต่องานอื่น.

ผลกระทบ downtime ที่ลดลงสำหรับผู้ใช้

ตัวอย่างเช่น, ถ้าทุกตอนบ่ายวันศุกร์ คุณต้องนำระบบเข้าสู่สภาพถูกจำกัดสำหรับการสำรองข้อมูล, คุณสามารถนำผู้ใช้ออฟไลน์ได้ทีละน้อย โดยการจบระบบย่อยที่ละหนึ่งระบบ.

การกำหนดสเกลและสภาพพร้อมใช้งานที่พัฒนาขึ้น

โดยการใส่ระบบย่อยหนึ่งระบบทำงานให้ผู้ใช้จำนวนน้อยลง, ระบบย่อยจึงยุ่ง ลดลงและสามารถตอบสนองต่อคำขอ งานที่ระบบย่อยจัดการได้มากขึ้น.

error tolerance ที่พัฒนาขึ้นในระบบย่อยแบบโต้ตอบ

โดยการกระจายงานในหลายระบบย่อย, ถ้าเกิดความล้มเหลวของเน็ตเวิร์ก, ระบบย่อยหลายระบบสามารถจัดการกระบวนการกู้คืนอุปกรณ์ได้.

เวลาเริ่มทำงานของระบบย่อยแบบโต้ตอบที่พัฒนาขึ้น

คุณสามารถลดเวลาการเริ่มทำงานของระบบย่อยได้โดยการแบ่งงาน ในหลายระบบย่อย.

อ็อปชันเพิ่มเติมสำหรับการปรับประสิทธิภาพ

โดยการใส่หลายระบบย่อย คุณจึงสามารถตั้งค่าระบบย่อยโดยมี routing entries จำนวนน้อยได้.

Subsystem description

Subsystem description คืออ็อบเจ็กต์ระบบที่มีข้อมูล ซึ่งกำหนดลักษณะของสถานะแวดล้อมการปฏิบัติการที่ควบคุมโดยระบบ. identifier ที่ระบบรับรู้สำหรับชนิดอ็อบเจ็กต์คือ *SBSD. subsystem description กำหนดว่างานจะเข้าสู่ระบบย่อยอย่างไร, ที่ไหน, และงานมีจำนวนเท่าใด, และระบบย่อยจะใช้รีซอร์สใดในการปฏิบัติงาน. ระบบย่อยแอ็คทีฟจะใช้ชื่อธรรมดาของ subsystem description.

เหมือนกับชุดของพิมพ์เขียวที่ละเอียด, แต่ละ subsystem description จะไม่ซ้ำ กัน, และมีลักษณะเฉพาะตัวที่อธิบายระบบย่อย. description ได้แก่ งานสามารถเข้าสู่ระบบย่อยได้เท่าใด, ระบบย่อยสามารถจัดการงาน ได้จำนวนเท่าใด, จะใช้แหล่งเก็บข้อมูลหลัก (หน่วยความจำ) จำนวนเท่าใด, และ งานในระบบย่อยสามารถรันได้รวดเร็วเพียงใด.

คุณสามารถใช้ subsystem description ที่ได้มาพร้อมกับระบบของคุณ (โดยทำ หรือไม่ทำการเปลี่ยนแปลง), หรือคุณสามารถสร้าง subsystem descriptions ของคุณเองได้.

Subsystem description attributes:

Subsystem description attributes คือแอตทริบิวต์ระบบโดยรวมทั่วไป. เมื่อคุณสร้างระบบย่อย, ขั้นตอนแรกคือการกำหนดแอตทริบิวต์ระบบย่อย.

แอตทริบิวต์ระบบย่อยประกอบด้วย:

- ชื่อของ subsystem description และไลบรารีที่ถูกจัดเก็บอยู่
- definition พูลหน่วยความจำทั้งหมดที่ระบบย่อยนี้ใช้
definition ระบบย่อยหนึ่ง สามารถมี definition พูลหน่วยความจำที่ระบุได้สูงสุด 10 definition. สิ่งประกอบ อยู่ใน definition ระบบย่อยมีดังนี้:
 - Pool definition identifier: นี่คือ identifier ภายใน subsystem description, ของ definition พูลของหน่วยความจำ.
 - ขนาด: นี่คือขนาดของพูลของหน่วยความจำที่ระบุในหน่วย KB (1K=1024 ไบต์) และเป็นจำนวนของแหล่งเก็บข้อมูลหลักที่พูลสามารถใช้ได้.
 - Activity level: นี่คือจำนวนสูงสุดของ threads ที่สามารถรันพร้อมกัน ในพูล.
- จำนวนสูงสุดของงานที่สามารถแอ็คทีฟในระบบย่อยใน เวลาเดียวกัน
- text description ของ subsystem description
- ชื่อและไลบรารีของไฟล์แสดงผล signon ที่ใช้ในการแสดงจอแสดงผล signon ที่เวิร์กสเตชันที่จัดสรรให้กับระบบย่อย
- ชื่อไลบรารีระบบย่อยที่คุณสามารถใช้ได้ ถ้าคุณต้องการระบุไลบรารี ที่ควรป้อนก่อนหน้าไลบรารีอื่นในส่วนรายชื่อไลบรารีของระบบ (พารามิเตอร์นี้ช่วยให้คุณสามารถใช้ไลบรารีภาษาที่สองได้.)

นอกจากนี้ subsystem description ยังมีข้อมูลเกี่ยวกับระดับสิทธิในการใช้งาน ระบบย่อยด้วย. ข้อมูลนี้ถูกเก็บรักษาโดย การรักษาความปลอดภัย และไม่ถูกจัดเก็บ พร้อมกับแอตทริบิวต์อื่นของ subsystem description. คุณสามารถดูสิทธิในการใช้งาน subsystem description ได้โดยใช้คำสั่ง Display Object Authority (DSPOBJAUT).

Work entry:

Work entries ระบุซอร์สที่งานสามารถป้อนระบบย่อยได้. ระบุชนิดของ work entries ที่จะใช้สำหรับงานชนิดต่างๆ. Work entries เป็นส่วนหนึ่งของ subsystem description.

ข้อมูลต่อไปนี้อธิบายถึงชนิดต่างๆ ของ work entries และวิธีการจัดการ. work entries มีอยู่ห้าชนิดดังนี้; autostart job entries, communication entries, job queue entries, prestart job entries, และ workstation entries.

Autostart job entries:

Autostart job entries ระบุให้งานแบบ autostart เริ่มขึ้นในทันที ที่ระบบย่อยเริ่มต้นขึ้น. เมื่อระบบย่อยเริ่มทำงาน, ระบบจะจัดสรรสิ่งต่างๆ ให้ระบบย่อย และเริ่มงานแบบ autostart และ prestart ก่อนที่ระบบย่อยจะพร้อมที่จะ ทำงาน.

งานแบบ autostart ที่เชื่อมโยงกับระบบย่อยจะถูกเริ่มต้นโดยอัตโนมัติ ในทุกครั้งที่ระบบย่อยถูกเริ่มขึ้น. งานแบบ autostart ในระบบย่อยการควบคุม สามารถใช้ในการเริ่มระบบย่อยอื่น (เช่นระบบย่อยการควบคุม ที่ได้มาจาก IBM). งานแบบ autostart คืองานแบ็คซ์ที่ทำงานซ้ำ.

ตัวอย่างเช่น: เพื่อเรียกโปรแกรมการกู้คืนพิเศษถ้า IPL ตรวจพบว่า ระบบก่อนหน้านี้จบลงอย่างผิดปกติ, คุณสามารถใส่เพิ่ม autostart job entry ที่ subsystem description ของระบบย่อยการควบคุม. โปรแกรมนี้ตรวจสอบค่ากำหนดของระบบ Previous system ending status (QABNORMSW). สำหรับการจบระบบตามปกติ, ค่าของ QABNORMSW คือ '0', และสำหรับการจบระบบอย่างผิดปกติ, ค่าของ QABNORMSW คือ '1'.

Communications entries:

communications work entry ระบุให้ระบบย่อยทราบถึงซอร์สสำหรับงานสื่อสาร ที่ระบบจะประมวลผล. การประมวลผลงาน เริ่มขึ้นเมื่อระบบย่อยได้รับ คำขอการเริ่มโปรแกรมการสื่อสารจากระบบรีโมตและพบ routing entry ที่เหมาะสมสำหรับคำขอ.

แต่เพื่อให้การทำงานมีประสิทธิภาพ, แทนที่จะเริ่มต้นงาน การสื่อสารทุกครั้งที่ได้รับคำขอการเริ่มโปรแกรม, คุณสามารถ ตั้งค่างานแบบ prestart เพื่อให้จัดการกับคำขอการเริ่มโปรแกรมจาก ระบบรีโมตได้. สำหรับงานแบ็คการสื่อสารที่จะรันบนระบบ iSeries, subsystem description ที่มี work entry สำหรับงานสื่อสารต้องมีอยู่บนระบบ.

Job queue entries:

Job queue entries ใน subsystem description ระบุ คิวงานซึ่งระบบย่อยจะได้รับงานมา. เมื่อระบบย่อยถูกเริ่มขึ้น, ระบบย่อยจะพยายามจัดสรรคิวงานแต่ละคิวที่กำหนดใน subsystem job queue entries.

ตัวอย่างเช่น, job queue entry ใน subsystem description QSYS/QBASE ระบุว่างานสามารถเริ่มโดยใช้คิวงาน QGPL/QBATCH. งานสามารถ ถูกวางบนคิวงานได้แม้ว่าระบบย่อยยังไม่ได้ถูกเริ่มต้นขึ้น. เมื่อ ระบบย่อย QBASE เริ่มขึ้น, ระบบย่อย จะประมวลผลงานบน คิว. subsystem description สามารถระบุจำนวนสูงสุดของงาน (แบบแบ็ค หรือแบบโต้ตอบ) ที่สามารถประมวลผลได้พร้อมกัน. จำนวนของงาน ที่สามารถแฉีกทีฟจากคิวงานใดๆ ถูกระบุอยู่ใน job queue entry.

Prestart job entries:

คุณกำหนดงานแบบ prestart โดยใช้ prestart job entry. prestart job entry ไม่มีผลต่อการจัดสรรอุปกรณ์หรือการกำหนดคำขอการเริ่มโปรแกรม.

แอ็ททริบิวต์งานของงานแบบ prestart จะไม่ถูกเปลี่ยนแปลงโดยระบบย่อย เมื่อ คำขอการเริ่มโปรแกรมแนบกับงานแบบ prestart. อย่างไรก็ตาม, โดยทั่วไป งานเซิร์ฟเวอร์จะเปลี่ยนแอ็ททริบิวต์งานเป็นแอ็ททริบิวต์ของโปรไฟล์ผู้ใช้ที่สลับค่า.

คำสั่ง Change Prestart Job (CHGPJ) อนุญาตให้งานแบบ prestart เปลี่ยนแอ็ททริบิวต์งานบางอย่างเป็นแอ็ททริบิวต์ของ รายละเอียดของงาน (ที่ระบุในรายละเอียดของงานที่เชื่อมโยงกับโปรไฟล์ผู้ใช้ของคำขอการเริ่มโปรแกรม หรือในรายละเอียดของงานที่ระบุใน prestart job entry).

งานแบบ prestart สำหรับเซิร์ฟเวอร์:

ในต้นแบบงานแบบ prestart จะมีงานการฟังหลักหนึ่งงาน, โดยทั่วไป เรียกว่างาน daemon หรืองาน listener, และหลายงานเซิร์ฟเวอร์ที่ประมวลผล คำขอไคลเอ็นต์. งาน daemon จะรอฟังคำขอการเชื่อมต่ออยู่บนพอร์ต. เมื่อได้รับการเชื่อมต่อใหม่, daemon จะทำงานทั่วไป, แล้วให้ socket descriptor ที่งานเซิร์ฟเวอร์แบบ prestart ที่กำลังรออยู่.

งานแบบ prestart สามารถถูกใช้ใหม่ได้. เมื่องานทำงานสำหรับไคลเอ็นต์หนึ่งเสร็จสมบูรณ์แล้ว, สภาวะแวดล้อมจะถูกรีเซ็ต และงานจะถูกทำให้พร้อมที่จะจัดการ คำขอจากไคลเอ็นต์อื่น.

สำหรับงานเซิร์ฟเวอร์ที่รันโค้ดผู้ใช้ (เช่น, เซิร์ฟเวอร์คำสั่งรีโมต), โดยปกติงานจะไม่ถูกนำมาใช้ใหม่. ที่เป็นเช่นนี้เนื่องจากโค้ดผู้ใช้อาจเปลี่ยน บางสิ่งในงานและไม่มีวิธีที่แน่นอนในการรีเซ็ตสถานะแวดล้อม สำหรับโคลเอ็นต์ใหม่. ถ้าเซิร์ฟเวอร์ของคุณนำงานมาใช้ใหม่, สามารถใช้ Change Job (QWTCGJJB) API เพื่อเปลี่ยนแอตทริบิวต์ของงานกลับไปยังสภาพที่รู้จัก หลังจากที่คุณโคลเอ็นต์ได้เสร็จสมบูรณ์แล้ว.

เซิร์ฟเวอร์ที่ใช้ต้นแบบงานแบบ prestart ได้แก่ โฮสต์เซิร์ฟเวอร์, เซิร์ฟเวอร์ SMTP, เซิร์ฟเวอร์ PPP, เซิร์ฟเวอร์ DDM/DRDA, เซิร์ฟเวอร์ SQL, และอื่นๆ.

Workstation entries:

งานแบบโต้ตอบคืองานที่เริ่มขึ้นเมื่อผู้ใช้ signs on ที่จอภาพ และสิ้นสุดลงเมื่อผู้ใช้ signs off. สำหรับงานที่จะรัน, ระบบย่อยจะค้นหารายละเอียดของงาน, ซึ่งอาจจะอยู่ใน workstation entry หรือโปรไฟล์ผู้ใช้.

workstation entry ช่วยให้ระบบย่อยทราบถึงเวิร์กสเตชันที่เป็นไปได้. ถ้าเวิร์กสเตชันพร้อมใช้งานได้, ระบบย่อยจะส่งจอภาพ signon ไปที่จอแสดงผล.

หมายเหตุ: subsystem description สำหรับระบบย่อยการควบคุมต้องมี workstation entry สำหรับคอนโซล, และ entry นั้นต้องเป็นชนิด *SIGNON. (*SIGNON คือค่าสำหรับพารามิเตอร์ AT, ที่ระบุบน คำสั่ง Add Work Station Entry (ADDWSE).) คำ *SIGNON บ่งชี้ว่าจอแสดงผล signon จะถูกแสดงขึ้นที่เวิร์กสเตชันเมื่อ ระบบย่อยถูกเริ่มขึ้น. ข้อกำหนดนี้ทำให้มั่นใจได้ว่า ระบบย่อยมี อุปกรณ์แบบโต้ตอบสำหรับ entry ของคำสั่งระดับระบบและระบบย่อย. คำสั่ง End System (ENDSYS) จะจบโปรแกรมที่ได้ไลเซนส์ของ iSeries เป็น single session (หรือจอแสดงผล signon) ที่คอนโซลในระบบย่อย การควบคุม. subsystem description ที่ไม่มี workstation entry สำหรับคอนโซลไม่สามารถถูกเริ่มเป็นระบบย่อยการควบคุม.

routing entry:

routing entry ระบุพลระบบย่อยของแหล่งเก็บข้อมูลหลักที่จะใช้, โปรแกรมการควบคุมที่จะรัน (โดยปกติคือ โปรแกรมที่ระบบจัดให้ QCMD), และข้อมูลรันไทม์เพิ่มเติม (จัดเก็บอยู่ในคลาสสอ็อบเจกต์). Routing entries ถูกจัดเก็บใน subsystem description.

routing entry สามารถถือเสมือนเป็น entry เต็มในไดเรกทอรี shopping mall. ลูกค้าที่ไม่สามารถหาร้านค้าที่ต้องการสามารถใช้ไดเรกทอรี เพื่อช่วยให้ทราบทิศทางที่ถูกต้องได้. ในลักษณะเดียวกันบนระบบ iSeries. Routing entries ช่วยให้งานทราบถึงตำแหน่งที่ถูกต้อง. Routing entries ใน subsystem description ระบุโปรแกรมที่จะถูกเรียกเพื่อควบคุมขั้นตอนการเราต์สำหรับงานที่รันอยู่ในระบบย่อย, พูลหน่วยความจำที่งานจะใช้, และคลาส ซึ่งจะเรียกแอตทริบิวต์รันไทม์มา. ข้อมูลการเราต์ระบุ routing entry สำหรับงานที่จะใช้. ในขณะเดียวกัน, routing entries และข้อมูลการเราต์นำเสนอ ข้อมูลเกี่ยวกับการเริ่มงานในระบบย่อย.

Routing entries ประกอบด้วยส่วนเหล่านี้; subsystem description, คลาส, ข้อมูลการเปรียบเทียบ, ขั้นตอนการเราต์สูงสุดที่แอ็คทีฟ, memory pool ID, โปรแกรม ที่จะเรียก, thread resources affinity, resources affinity group, และหมายเลข ลำดับ.

คลาส:

แอ็ททริบิวต์รันไทม์ของงานมีอยู่ในคลาสอ็อบเจกต์ที่ถูกระบุในพารามิเตอร์ (CLS) ใน routing entry. ถ้างานประกอบด้วยขั้นตอนการเรอต์หลายขั้นตอน, คลาสที่ใช้โดยแต่ละขั้นตอนการเรอต์ต่อมา จะถูกระบุใน routing entry ที่ใช้ในการเริ่มขั้นตอนการเรอต์. ถ้าคลาสไม่มีอยู่เมื่อใส่เพิ่ม routing entry, จะต้องระบุ library qualifier เนื่องจากชื่อคลาสที่ถูกต้องถูกเก็บอยู่ใน subsystem description.

แอ็ททริบิวต์รันไทม์ที่มีอยู่ใน routing entry class คือ:

ระดับความสำคัญการรัน (RUNPTY)

ระดับความสำคัญการรันคือช่วงของค่าตั้งแต่ 1 (ระดับความสำคัญสูงสุด) ถึง 99 (ระดับความสำคัญต่ำสุด) ที่แสดงถึงระดับความสำคัญซึ่งงานแย่งชิงหน่วยการประมวลผลเมื่อเปรียบเทียบกับงานอื่นที่แอ็คทีฟในเวลาเดียวกัน.

สำหรับงานแบบ multi-threaded, ระดับความสำคัญการรันคือระดับความสำคัญการรันสูงสุดที่ใช้ได้สำหรับ thread ใดๆ ภายในงาน. แต่ละ threads ภายในงานอาจมีระดับความสำคัญต่ำกว่า.

การแบ่งเวลา (TIMESLICE)

การแบ่งเวลาเป็นตัวกำหนดจำนวนเวลา ที่ต้องการโดย thread ในงานในการทำให้การประมวลผลจำนวนหนึ่งเสร็จสมบูรณ์. เมื่อสิ้นสุดการแบ่งเวลา, thread อาจถูกกำหนดให้มีสถานะ inactive เพื่อให้ threads อื่นสามารถแอ็คทีฟในพูลของหน่วยความจำได้.

ช่วงเวลารอดีฟอลต์ (DFTWAIT)

ค่านีระบุเวลาสูงสุดดีฟอลต์ (ในวินาที) ที่ thread ใน งานจะคอยคำสั่งระบบ, เช่นคำสั่ง LOCK machine interface (MI), เพื่อให้ได้รีซอร์ส. จะใช้ช่วงเวลารอดีฟอลต์นี้ถ้าไม่มีการระบุช่วงเวลารอค่าอื่นสำหรับสถานการณ์ที่กำหนดให้. โดยทั่วไป, ช่วงเวลานี้เป็นเวลา ที่ผู้ใช้ระบบเต็มใจที่จะรอรระบบก่อนที่จะจับค่าขอ. ถ้ารอนานเกินช่วงเวลารอสำหรับคำสั่งใดๆ, ข้อความแสดงความผิดพลาดอาจถูกแสดงขึ้นหรืออาจถูกจัดการโดยอัตโนมัติโดยคำสั่ง Monitor Message (MONMSG).

เวลา CPU สูงสุด (CPUTIME)

ค่านีระบุเวลาสูงสุดของหน่วยการประมวลผล (ในมิลลิวินาที) ที่งานสามารถใช้ได้. ถ้างานประกอบด้วยขั้นตอนการเรอต์หลายขั้นตอน, แต่ละขั้นตอน การเรอต์สามารถใช้เวลาของหน่วยการประมวลผลนี้ได้. ถ้าใช้เกินเวลาสูงสุด, งานจะถูกจบ.

หน่วยเก็บชั่วคราวสูงสุด (MAXTMPSTG)

ค่านีระบุจำนวนสูงสุดของหน่วยเก็บชั่วคราว (auxiliary) ที่งานสามารถใช้ได้. ถ้างานประกอบด้วยขั้นตอนการเรอต์หลายขั้นตอน, ค่านีคือ หน่วยเก็บชั่วคราวสูงสุดที่ขั้นตอนการเรอต์สามารถใช้ได้. หน่วยเก็บชั่วคราวนี้ใช้สำหรับหน่วยเก็บที่ต้องการโดยตัวโปรแกรมเองและโดยอ็อบเจกต์ระบบภายใน ที่สร้างโดย ทางอ้อมซึ่งใช้ในการสนับสนุนงาน. ค่านีไม่รวมหน่วยเก็บ ในโลบารี QTEMP. ถ้าใช้เกินหน่วยเก็บชั่วคราวสูงสุด, งานจะถูกจบ. พารามิเตอร์นี้ไม่ได้ใช้กับการใช้หน่วยเก็บถาวร, ซึ่งถูก ควบคุมผ่านทางโปรไฟล์ผู้ใช้.

thread สูงสุด (MAXTHD)

ค่านีระบุจำนวนสูงสุดของ threads ที่งานซึ่งใช้คลาสนี้ สามารถรันได้ทุกเมื่อ. ถ้าหลาย thread ถูกเริ่มต้นพร้อมกัน, ค่านีอาจ มากเกินได้. ถ้าใช้เกินค่าสูงสุดนี้, thread ที่มากเกินไป จะได้รับอนุญาตให้รันจนเสร็จสมบูรณ์ตามปกติ. การเริ่มต้น threads เพิ่มเติมจะถูกห้าม จนกว่าจำนวนสูงสุดของ threads ในงานจะลดลงต่ำกว่าค่าสูงสุดนี้.

คำอธิบาย (TEXT)

ค่านีระบุข้อความที่อธิบายอ็อบเจกต์อย่างคร่าวๆ. นี่คือแอ็ททริบิวต์ของคลาสอ็อบเจกต์เมื่อคลาสอ็อบเจกต์ถูกสร้างขึ้น, แต่ไม่ใช่ แอ็ททริบิวต์รันไทม์สำหรับงาน.

สิทธิ (AUT)

ค่านี้ระบุสิทธิที่คุณให้แก่ผู้ใช้ที่ไม่มีสิทธิเฉพาะ สำหรับอ็อบเจกต์, ผู้ใช้ที่ไม่ได้อยู่บน authorization list, และผู้ใช้ที่มีโปรไฟล์กลุ่มหรือโปรไฟล์กลุ่มเสริมที่ไม่มีสิทธิเฉพาะ สำหรับอ็อบเจกต์. นี่คือนิวแอตทริบิวต์ของคลาสอ็อบเจกต์เมื่อคลาสอ็อบเจกต์ถูกสร้างขึ้น, แต่ไม่ใช่แอตทริบิวต์รันไทม์สำหรับงาน.

ข้อมูลการเปรียบเทียบ:

พารามิเตอร์ค่าการเปรียบเทียบ (CMPVAL) ของ routing entry ระบุข้อมูลที่จะถูกเปรียบเทียบกับข้อมูลการเรดท์ เพื่อกำหนด routing entry ที่จะใช้. (routing entry ยังระบุตำแหน่งเริ่มต้นสำหรับการเปรียบเทียบด้วย.) ข้อมูลการเรดท์จะถูกเปรียบเทียบกับค่าการเปรียบเทียบของแต่ละ routing entry ตามหมายเลขลำดับจนกว่าจะพบข้อมูลที่ตรงกัน. หมายเลขลำดับที่อยู่ใน routing entry กำหนดลำดับในการสแกน routing entries และสามารถใช้เป็น identifier ของ routing entry.

เมื่อพบ routing entry ที่มีค่าเปรียบเทียบตรงกับข้อมูลการเรดท์, ขั้นตอนการเรดท์จะถูกเริ่มขึ้น และโปรแกรมที่ระบุใน routing entry จะถูกเรียก. แอตทริบิวต์รันไทม์ในคลาสที่เชื่อมโยงกับ routing entry จะถูกใช้สำหรับขั้นตอนการเรดท์, และขั้นตอนการเรดท์จะรันใน พูลของหน่วยความจำที่ระบุใน routing entry.

คุณสามารถระบุค่าเปรียบเทียบของ *ANY บน routing entry ที่มีหมายเลขสูงสุด. *ANY หมายความว่าบังคับให้จับคู่โดยไม่คำนึงว่าข้อมูลการเรดท์จะเป็นอะไรก็ตาม. เพียงหนึ่ง routing entry เท่านั้นที่สามารถมีค่าเปรียบเทียบเป็น *ANY, และต้องเป็น entry หลังสุด (หมายเลขลำดับสูงสุด) ใน subsystem description.

ขั้นตอนการเรดท์แอคทีฟสูงสุด:

พารามิเตอร์ maximum active routing steps (MAXACT) ของ routing entry ระบุจำนวนสูงสุดของขั้นตอนการเรดท์ (งาน) ที่สามารถแอคทีฟพร้อมกันผ่านทาง routing entry นี้.

ในงานหนึ่ง, จะมีขั้นตอนการเรดท์ที่แอคทีฟเพียงหนึ่งขั้นตอนเท่านั้นในแต่ละเวลา. เมื่อระบบย่อย แอคทีฟและใช้งานถึงจำนวนสูงสุดของขั้นตอนการเรดท์, ความพยายามต่อมาในการเริ่มขั้นตอนการเรดท์ผ่านทาง routing entry นี้จะล้มเหลว. งานที่พยายาม เริ่มขั้นตอนการเรดท์จะถูกจบ, และระบบย่อยจะส่งข้อความ ไปที่ไฟล์บันทึกของงาน.

โดยปกติ ไม่มีเหตุผลที่ต้องควบคุมจำนวนของขั้นตอนการเรดท์, ดังนั้น ค่าที่แนะนำคือ *NOMAX.

Memory pool ID:

พารามิเตอร์ memory pool ID (POOLID) ของ routing entry ระบุ pool identifier ของพูลของหน่วยความจำซึ่งโปรแกรมจะรันในนั้น. pool identifier ที่ระบุที่นี่เกี่ยวข้องกับพูลของหน่วยความจำ ใน subsystem description.

โปรแกรมที่จะเรียก

พารามิเตอร์โปรแกรมที่จะเรียก (PGM) ของ routing entry ระบุชื่อและไลบรารีของโปรแกรม ที่เรียกเป็นโปรแกรมแรกที่จะรันในขั้นตอนการเรดท์. ไม่สามารถส่งผ่านพารามิเตอร์ใด ไปที่โปรแกรมที่ระบุ. ชื่อโปรแกรมอาจถูกระบุโดยตรงใน routing entry, หรือดึงข้อมูลมาจากข้อมูลการเรดท์ อย่างไม่อย่างหนึ่ง.

ถ้าชื่อ โปรแกรมถูกระบุใน routing entry, การเลือก routing entry นั้นส่งผลให้ routing entry program ถูกเรียก (ไม่คำนึงถึงชื่อโปรแกรม ที่ส่งผ่านในฟังก์ชัน EVOKE). ถ้าจะเรียกโปรแกรมที่ระบุในฟังก์ชัน EVOKE, ต้องระบุ *RTGDTA ในพารามิเตอร์นี้. ถ้าโปรแกรมไม่มีอยู่เมื่อใส่เพิ่มหรือเปลี่ยน routing entry, ต้องระบุ library qualifier เนื่องจากชื่อโปรแกรมที่ถูกต้อง ถูกจัดเก็บใน subsystem description.

หมายเลขลำดับ

พารามิเตอร์หมายเลขลำดับ (SEQNBR) ของ routing entry บอกให้ระบบย่อยทราบถึงลำดับในการค้นหาข้อมูลการเรดที่ตรงกันใน routing entries. routing entries จะถูกค้นหาตามหมายเลขลำดับ. เมื่อคุณใส่เพิ่ม routing entries ที่ subsystem description, คุณควรจัดลำดับ entry เพื่อให้ entry ที่จะถูกเปรียบเทียบบ่อยที่สุด เป็น entry แรก. การทำเช่นนี้จะช่วยลดเวลาการค้นหา.

Sequence Number	Comparison Value
10	'ABC'
20	'AB'
30	'A'
40	'E'
50	'D'

ในตัวอย่างข้างต้น, routing entries จะถูกค้นหาตามหมายเลข ลำดับ. ถ้าข้อมูลการเรดคือ 'A', การค้นหาจะจบด้วย routing entry 30. ถ้าข้อมูลการเรดคือ 'AB', การค้นหาจะจบด้วย routing entry 20. ถ้าข้อมูลการเรดคือ 'ABC', การค้นหาจะจบด้วย routing entry 10. เนื่องจาก ข้อมูลการเรดสามารถยาวกว่าค่าเปรียบเทียบของ routing entry, การเปรียบเทียบ (ซึ่งทำในลำดับจากซ้ายไปขวา) จะหยุดเมื่อมาถึง ตอนจบของค่าเปรียบเทียบ. ดังนั้น, ถ้าข้อมูลการเรดคือ 'ABCD', การค้นหาจะจบด้วย routing entry 10.

เมื่อคุณกำหนด routing entries, entry ต้องถูกจัดลำดับจาก entry ที่เฉพาะที่สุดไปถึง entry ทั่วไปที่สุด. ตัวอย่างต่อไปนี้ แสดงวิธีที่ถูกต้องและไม่ถูกต้องในการกำหนด routing entries:

ถูกต้อง		ไม่ถูกต้อง	
Sequence Number	Comparison Value	Sequence Number	Comparison Value
10	'ABC'	10	'ABC'
20	'AB'	20	'ABCD'
30	'A'		
40	'E'		
9999	*ANY		

ในตัวอย่างวิธีที่ไม่ถูกต้อง, ไม่สามารถจับคู่ routing entry 20 เนื่องจากข้อมูลการเรดใดๆ ที่ตรงกับค่าเปรียบเทียบสำหรับ routing entry 20 จับคู่กับ routing entry 10 ก่อน. เมื่อเปลี่ยนหรือใส่เพิ่ม routing entry ที่ subsystem description ด้วยค่าเปรียบเทียบที่ทำให้เกิดสถานการณ์นี้, ระบบจะส่งข้อความวินิจฉัยที่ระบุสถานการณ์.

โปรแกรม ที่มีชื่อใน routing entry จะถูกควบคุมเมื่อขั้นตอนการเรดสำหรับงานเริ่มขึ้น. พารามิเตอร์ในการควบคุมสภาวะแวดล้อมรันไทม์ (ระดับความสำคัญ, การแบ่งเวลา, และอื่นๆ) ของขั้นตอนการเรดสำหรับงานจะถูกนำมาจากคลาสที่ได้ระบุใน routing entry.

วิธีการเริ่มระบบย่อย

เมื่อระบบย่อยเริ่มต้น, ระบบจะจัดสรรสิ่งต่างๆ ให้ระบบย่อยและเริ่ม งานแบบ autostart และงานแบบ prestart ก่อนที่ระบบย่อยจะพร้อมสำหรับงาน.

subsystem description ถูกใช้ในการกำหนดวิธีการจัดการไอเท็มต่างๆ . รายการต่อไปนี้แสดงลำดับของเหตุการณ์ที่เกิดขึ้นเมื่อระบบย่อยเริ่มทำงาน:

1. มีการออกคำขอเพื่อเริ่มระบบย่อย. มีการออกคำสั่ง Start Subsystem (STRSBS) . ระบุตำแหน่งข้อมูล startup หลักใน subsystem description.
2. พูลหน่วยความจำถูกจัดสรร. หน่วยความจำจะถูกจัดสรรให้กับพูลที่ได้กำหนดไว้ใน subsystem description. หน่วย ความจำที่ถูกจัดสรรให้กับพูลที่กำหนดไว้แต่ละอันถูกดึงมาจากพูลหน่วยความจำ หลัก. ระบบจะไม่จัดสรรหน่วยความจำไปที่ พูล ถ้าจำนวนของหน่วยความจำที่พร้อมใช้งานที่พูลของหน่วยความจำ Base น้อยกว่า ขนาดต่ำสุดที่ระบุโดยค่ากำหนด ของระบบ Base memory pool minimum size QBASPOOL. หากระบบไม่สามารถจัดสรรหน่วยความจำได้ตามที่มีการร้องขอเข้ามา, ระบบจะ จัดสรรหน่วยความจำให้มากที่สุดเท่าที่จะทำได้และจัดสรรให้เพิ่มเติมเมื่อมี หน่วยความจำเหลือเพิ่มขึ้น.
3. งานแบบ prestart เริ่มต้น. ข้อมูลนี้มาจาก prestart job entries.
4. งานแบบ autostart เริ่มต้น. ข้อมูลนี้มาจาก autostart jobs entries.
5. จอภาพถูกจัดสรร (จอแสดงผล signon ปรากฏขึ้น). ถ้ามี workstation entries และอุปกรณ์ถูก varied on และยังไม่ได้ถูก จัดสรรโดยระบบ ย่อยอื่นใด, ระบบย่อยสามารถจะจัดสรรอุปกรณ์และแสดงจอแสดงผล signon . ถ้าอุปกรณ์ถูก varied on และได้ถูกจัดสรรโดยระบบย่อยอื่นแล้ว และอยู่ที่จอแสดงผล signon (จอแสดงผล signon ถูกแสดงขึ้นก่อนที่ระบบย่อย ที่สองจะเริ่มต้น), ระบบย่อยที่สองสามารถจัดสรรอุปกรณ์ จากระบบย่อยแรกและแสดงจอแสดงผล signon. ถ้าอุปกรณ์ ไม่ได้ถูก varied on, ระบบย่อยจะไม่สามารถจัดสรรอุปกรณ์. งาน system arbiter (QSYSARB) และงาน QCMNARB พักล็อกบนอุปกรณ์ที่ varied-off ทั้งหมด. Workstation entries ให้ข้อมูลว่าอุปกรณ์ใดจะถูกตรวจสอบการจัดสรร.

หมายเหตุ: สำหรับอุปกรณ์แสดงผล เสมือน, signon prompt ถูกแสดงขึ้นเมื่ออุปกรณ์ถูก varied on ทั้งหมด. สิ่งนี้เกิดขึ้นเมื่อผู้ใช้เชื่อมต่อกับ iSeries โดยใช้ device description นั้น (สมมุติว่าคำขอการเชื่อมต่อไม่มีข้อมูล ที่จะใช้ในการข้ามกระบวนการ signon prompt). อุปกรณ์อาจถูกนำมาจาก พูลของ device descriptions ที่สร้างขึ้นก่อนหน้าและถูก varied on เป็นส่วนหนึ่งของกระบวนการเชื่อมต่อนั้น, หรืออุปกรณ์สามารถถูกสร้างและ varied on ได้. เมื่อระบบย่อยเริ่มขึ้น, ระบบย่อยจะคงค้างล็อกสำหรับ device descriptions ที่สร้างขึ้นก่อนหน้าซึ่งระบบย่อยต้องการ.

6. คิวงานถูกจัดสรร. ระบบย่อยจะไม่สามารถจัดสรรคิวงานได้หากคิวงานได้ถูกจัดสรรให้กับระบบย่อย ที่แฉีกที่อื่นไปแล้ว. ข้อมูลนี้มาจาก job queue entries.
7. อุปกรณ์สื่อสารถูกจัดสรร. คำขอร้องส่งไปที่งานระบบ QLU (LU services), ซึ่งจัดการการจัดสรรอุปกรณ์สำหรับ อุปกรณ์สื่อสารทั้งหมด. ข้อมูลนี้มาจาก communication entries.
8. สภาวะแวดล้อมพร้อมสำหรับงาน.

วิธีการจัดสรรอุปกรณ์เวิร์กสเตชัน:

ระบบย่อยพยายามจัดสรรอุปกรณ์เวิร์กสเตชันทั้งหมดใน subsystem description สำหรับ AT(*SIGNON) workstation entries.

สถานการณ์ต่อไปนี้อาจเกิดขึ้นในระหว่างเวลาที่ระบบย่อยเริ่มต้น:

- ถ้าอุปกรณ์ไม่ได้ถูก varied on, ระบบย่อยจะไม่สามารถจัดสรรอุปกรณ์. งาน system arbiter (QSYSARB) และ QCMNARBxx พักล๊อคบนอุปกรณ์ที่ถูก varied-off ทั้งหมด.
- ถ้าอุปกรณ์ถูก varied on และยังไม่ได้ถูกจัดสรรโดยระบบย่อยอื่น, ระบบย่อยสามารถจัดสรรอุปกรณ์และแสดงจอแสดงผล signon.
- ถ้าอุปกรณ์ถูก varied on และได้ถูกจัดสรรโดยระบบย่อยอื่นแล้ว และอยู่ที่จอแสดงผล signon (จอแสดงผล signon ถูกแสดงขึ้นก่อนที่ระบบย่อย ที่สองจะเริ่มต้น), ระบบย่อยที่สองสามารถจะจัดสรรอุปกรณ์จากระบบย่อยแรกและแสดงจอแสดงผล signon.

ถ้ามีมากกว่าหนึ่งระบบย่อยพยายามจัดสรรเวิร์กสเตชันเดียวกัน (ตั้งที่ระบุใน workstation entries) และเวิร์กสเตชันถูก varied off, จะไม่สามารถคาดการณ์ ระบบย่อยที่เรียกเวิร์กสเตชันเมื่อเวิร์กสเตชันถูก varied on. ในลักษณะคล้ายกัน, ถ้า workstation entry ระบุชนิดเวิร์กสเตชันแทนชื่อเวิร์กสเตชัน, ระบบย่อยอาจจะเรียกเวิร์กสเตชันชนิดนั้นทั้งหมด, บางส่วน, หรือไม่เรียกเวิร์กสเตชันชนิดนั้น. (สิ่งนี้ใช้กับ workstation entries ที่มีชื่อทั่วไป ด้วย.) เพื่อหลีกเลี่ยงสถานการณ์ดังกล่าว, คุณสามารถตั้งค่า workstation entries สำหรับระบบย่อย เพื่อไม่ให้หลายระบบย่อยใช้เวิร์กสเตชันเดียวกัน.

หลังจากผู้ใช้ได้ signed on

เมื่อผู้ใช้ signs on ที่เวิร์กสเตชัน, งานจะรันในระบบย่อยที่ถูกแสดงบนจอแสดงผล signon บนเวิร์กสเตชัน (ระบบย่อยถูกระบุในจอแสดงผล signon ที่ได้มาจาก IBM). สถานการณ์ต่อไปนี้อาจเกิดขึ้นหลังจากที่ผู้ใช้ได้ signed on:

- ถ้าระบบย่อยที่สองเริ่มขึ้นและพยายามที่จะจัดสรรเวิร์กสเตชัน ซึ่งผู้ใช้ signed on บนนั้น, ระบบย่อยที่สองจะไม่สามารถจัดสรรเวิร์กสเตชัน. งานของ ผู้ใช้ยังคงรันต่อไปในระบบย่อยแรก.
- ถ้าผู้ใช้เลือกอ็อปชัน 1 (Display signon for alternative job) บนเมนู System Request หรือออกคำสั่ง Transfer to Secondary Job (TFRSECJOB), งานใหม่จะรันในระบบย่อยเดียวกันกับงานแรก.
- เมื่อผู้ใช้ signs off, เวิร์กสเตชันจะยังคงถูกจัดสรรที่ระบบย่อยที่ใช้เมื่อ ผู้ใช้ signed on, ยกเว้นว่าผู้ใช้ได้ถ่ายโอนเข้าในระบบย่อยโดยใช้ คำสั่ง Transfer Job (TFRJOB), และระบุ AT(*ENTER) สำหรับ workstation entry สำหรับเวิร์กสเตชันนี้. จอแสดงผล signon ถูกแสดงขึ้น, และงานต่อมาใดๆ จากเวิร์กสเตชันนั้นยังคงรันต่อไปในระบบย่อยนั้น, (ยกเว้นว่ามีการเริ่มต้นระบบย่อยอื่นที่จัดสรรเวิร์กสเตชัน ในขณะที่ มันอยู่ที่จอแสดงผล signon).
- ถ้าผู้ใช้ signs off และระบบย่อยซึ่งงานของผู้ใช้เคยรันได้จบลง, อุปกรณ์จะถูกจัดสรรคืน. จากนั้นระบบย่อยที่สองสามารถจัดสรรอุปกรณ์ และแสดงจอแสดงผล signon.

สถานการณ์จำลอง: การจัดสรรเวิร์กสเตชัน:

ข้างล่างนี้เป็นตัวอย่างของวิธีการจัดสรรสองเวิร์กสเตชันไปที่ระบบย่อย ที่แตกต่างกันสองระบบ.

ในสถานการณ์จำลองนี้, ระบบย่อย A และระบบย่อย B มีเวิร์กสเตชัน DSP01 และ DSP02 ใน subsystem descriptions ของตน (workstation entries ระบุ AT(*SIGNON)).

Device Name	จัดสรรไปที่
DSP01	ระบบย่อย A
DSP02	ระบบย่อย A

สมมุติว่าทั้งสองเวิร์กสเตชันถูก varied on เมื่อระบบย่อย A เริ่มต้น.

ระบบย่อย A จะจัดสรรทั้งสองเวิร์กสเตชันและแสดงจอแสดงผล signon บน ทั้งสองเวิร์กสเตชัน. แม้ว่าระบบย่อย A มีจอแสดงผล signon ที่แสดงบนเวิร์กสเตชัน, เวิร์กสเตชันสามารถถูกจัดสรรโดยระบบย่อยหรืองานอื่น; จากนั้น เวิร์กสเตชัน จะไม่พร้อมใช้งานที่ระบบย่อย A อีกต่อไป.

Device Name	จัดสรรไปที่
DSP01	USER1
DSP02	ระบบย่อย A

เมื่อผู้ใช้ (USER1) signs on ที่เวิร์กสเตชัน DSP01, อุปกรณ์จะถูกจัดสรรที่งานของ USER1, ซึ่งกำลังรันในระบบย่อย A. เวิร์กสเตชัน DSP02 ยังคงอยู่ที่ จอแสดงผล signon. ดังนั้นจึงสามารถถูกจัดสรรโดยระบบย่อยหรืองานอื่นได้. จากนั้น เวิร์กสเตชันจะไม่พร้อมใช้งานที่ระบบย่อย A อีกต่อไป.

Device Name	จัดสรรไปที่
DSP01	USER1
DSP02	ระบบย่อย B

ระบบย่อย B เริ่มต้น. เนื่องจาก USER1 ได้ signed on ที่เวิร์กสเตชัน DSP01, ระบบย่อย B จึงไม่สามารถจัดสรรอุปกรณ์. ระบบย่อย B ร้องขอการจัดสรรของ อุปกรณ์เมื่ออุปกรณ์พร้อมใช้งาน. DSP02 จะถูกจัดสรรที่ระบบย่อย B เนื่องจาก ไม่มีใคร signed on ที่อุปกรณ์ในระบบย่อย A. งานใดๆ ที่เริ่มบน DSP02 จะ รันในระบบย่อย B.

Device Name	จัดสรรไปที่
DSP01	ระบบย่อย A
DSP02	ระบบย่อย B

USER1 signs off. เนื่องจากงานผู้ใช้เคยรันอยู่ในระบบย่อย A, ที่ระบบย่อยแสดงจอแสดงผล signon เพื่อให้ผู้ใช้คนอื่นสามารถ sign on เวิร์กสเตชันและรันในระบบย่อย A. ถ้าระบบย่อย A ถูกจบ, เวิร์กสเตชัน DSP01 จะถูกจัดสรรโดยระบบย่อย B (เนื่องจากมีคำขอที่จะจัดสรรอุปกรณ์ค้างอยู่.)

ชื่อของระบบย่อยที่มีเวิร์กสเตชันที่จัดสรรอยู่ในปัจจุบันจะปรากฏขึ้น ในมุมมองด้านขวาบนของจอแสดงผล signon ที่ได้มาจาก IBM.

พูลหน่วยความจำ

พูลหน่วยความจำเป็นส่วนโลจิคัลของหน่วยความจำหลักหรือหน่วยเก็บข้อมูลที่สงวนไว้สำหรับการประมวลผลงานหรือกลุ่มของงาน. บนเซิร์ฟเวอร์ iSeries , แหล่งเก็บข้อมูลหลักทั้งหมดสามารถแบ่งออกเป็นการจัดสรรแบบโลจิคัลที่เรียกว่าพูลหน่วยความจำ. โดยค่าเริ่มต้น, ระบบจะจัดการถ่ายโอนข้อมูลและโปรแกรมเข้าในพูลหน่วยความจำ.

พูลหน่วยความจำที่งานของผู้ใช้ได้หน่วยความจำมาจะเป็นพูลอันเดียวกันกับที่จำกัด activity level ของพูลนั้น. (activity level ของพูลหน่วยความจำ คือจำนวนของ threads ที่สามารถแอ็คทีฟพร้อมกันในพูลหน่วยความจำ.) Exceptions ของสิ่งนี้คืองานระบบ (เช่น Scpf, Qsysarb, และ Qlus) ที่ได้รับ หน่วยความจำมาจาก Base pool แต่ใช้ machine pool activity level. นอกจากนี้, การมอนิเตอร์ระบบย่อยได้รับหน่วยความจำมาจาก subsystem description pool แรก, แต่ใช้ machine pool activity level. สิ่งนี้จะช่วยให้มอนิเตอร์ระบบย่อยสามารถทำงานได้ตลอดโดยไม่ต้องสนใจต่อการตั้งค่าของ activity level.

เพราะเหตุใดจึงใช้พูลหน่วยความจำ

คุณสามารถควบคุมปริมาณของงานที่จะทำในระบบย่อยโดยการควบคุมจำนวนและขนาดของพูลหน่วยความจำ. ขนาดของพูลในระบบย่อยยิ่งใหญ่มากขึ้น, จำนวนงานที่สามารถทำในระบบย่อยนั้น ยิ่งมากขึ้น.

การใช้พูลหน่วยความจำแบบแบ่งใช้ช่วยให้ระบบสามารถกระจายงานสำหรับผู้ใช้งานแบบโต้ตอบไปยังระบบย่อยหลายระบบ ในขณะที่ยังคงอนุญาตให้งานรันในพูลหน่วยความจำเดียวกันได้.

พูลหลายพูลในระบบย่อย ช่วยคุณควบคุมการแย่งชิงรีซอร์สของระบบของงาน. ประโยชน์ของการมีพูลหลายพูลในระบบย่อยคือ คุณสามารถแบ่งปริมาณของงานที่ทำและเวลาตอบสนองสำหรับงานเหล่านี้ได้. ตัวอย่างเช่น, ในระหว่างวัน คุณอาจต้องการให้งานแบบโต้ตอบรันโดยมีเวลาตอบสนองที่ดี. สำหรับประสิทธิภาพที่ดีขึ้น คุณสามารถทำให้พูลแบบโต้ตอบใหญ่ขึ้น. ในตอนกลางคืน คุณอาจ รันงานแบตช์หลายงาน, คุณจึงทำให้พูลแบบแบตช์ใหญ่ขึ้น.

หมายเหตุ: แม้ว่า การปรับและการจัดการระบบของคุณสามารถช่วยเพิ่มประสิทธิภาพความต่อเนื่อง ของงาน ผ่านเซิร์ฟเวอร์ iSeries ของคุณ, แต่การปรับไม่สามารถช่วยแก้ปัญหาการขาดแคลนฮาร์ดแวร์รีซอร์ส. ดังนั้นควรพิจารณาเรื่องการอัปเดตฮาร์ดแวร์ หากความต้องการด้านเวิร์กโหลดมีความสำคัญมาก.

วิธีการจัดการข้อมูลในพูลหน่วยความจำ

ถ้าข้อมูลมีอยู่แล้วใน แหล่งเก็บข้อมูลหลัก, ข้อมูลสามารถถูกอ้างอิงแยกต่างหากจากพูลหน่วยความจำที่เก็บอยู่ได้. อย่างไรก็ตาม, ถ้าข้อมูลที่ต้องการไม่มีอยู่ในพูลหน่วยความจำใดๆ, ข้อมูล จะถูกนำกลับมาอยู่ในพูลหน่วยความจำเดียวกันกับงานที่อ้างอิงถึงข้อมูล (สิ่งนี้ เรียกว่า page fault). เมื่อข้อมูลถูกถ่ายโอนเข้าไปในพูลหน่วยความจำ, ข้อมูลอื่นจะถูก สับเปลี่ยน และ, ถ้าเปลี่ยนแปลง, ถูกบันทึกอย่างอัตโนมัติในหน่วยความจำรอง (สิ่งนี้เรียกว่าการเพจ). ขนาดของพูลหน่วยความจำควรใหญ่พอที่จะเก็บ การถ่ายโอนข้อมูล (การเพจ) ที่ระดับที่เหมาะสม เนื่องจากอัตราามีผลต่อ ประสิทธิภาพ.

ชนิดของพูลหน่วยความจำ

บนเซิร์ฟเวอร์ iSeries, แหล่งเก็บข้อมูลหลักทั้งหมดสามารถแบ่งออกเป็น การจัดสรรแบบโลจิคัลที่เรียกว่า พูลหน่วยความจำ. หน่วยความจำทั้งหมด ในระบบสามารถแบ่งเป็นแบบส่วนตัวหรือแบบแบ่งใช้ ใช้อย่างใดอย่างหนึ่ง. มีพูลหน่วยความจำส่วนตัว, พูลหน่วยความจำแบบแบ่งใช้, และพูลหน่วยความจำแบบแบ่งใช้พิเศษ. พูลหน่วย ความจำสูงสุดทั้งหมด 64 พูล, สามารถเอ็ดคที่พีได้ในเวลาเดียวกัน, ไม่ว่าจะ เป็นแบบ ส่วนตัวหรือแบบแบ่งใช้ก็ตาม.

พูลหน่วยความจำส่วนตัว

พูลหน่วยความจำส่วนตัว (เรียกอีก อย่างว่าพูลหน่วยความจำที่ผู้ใช้กำหนด) มีจำนวนของแหล่งเก็บข้อมูลหลัก ที่ระบบย่อยเดี่ยวสามารถใช้ในการรันงานต่างๆ. พูลเหล่านี้ไม่สามารถแบ่งใช้โดยหลายระบบย่อย. พูลชนิดนี้ถูกระบุใน iSeries Navigator ตามชื่อของระบบย่อย. คุณสามารถมีพูลหน่วยความจำส่วนตัวได้มากถึง 62 พูลสำหรับใช้ในระบบย่อย ที่แอ็คทีฟ.

พูลหน่วยความจำแบบแบ่งใช้

พูลแบบแบ่งใช้ แบ่งเป็น แบบพิเศษหรือทั่วไป ใช้อย่างใดอย่างหนึ่ง; Machine pool และ Base pool ถือเป็นพูล แบบแบ่งใช้พิเศษ, และพูลแบบแบ่งใช้อื่นทั้งหมดถือเป็นพูลแบบแบ่งใช้ทั่วไป. คุณสามารถระบุพูลหน่วยความจำแบบแบ่งใช้ได้ 63 จากทั้งหมด 64 พูลที่ถูกกำหนด บนระบบ สำหรับใช้เมื่อสร้าง subsystem descriptions (machine pool ถูกสงวนไว้ สำหรับการใช้งานของระบบ).

Special Shared Pools (*MACHINE และ *BASE)

*MACHINE

พูลหน่วยความจำ Machine ใช้สำหรับโปรแกรม highly-shared Machine และ ระบบปฏิบัติการ. พูลนี้ถูกระบุเป็น Machine ใน iSeries Navigator. พูลหน่วยความจำ Machine นำเสนอหน่วยความจำสำหรับภารกิจที่ระบบต้องรันโดยไม่ต้องจัดการของคุณ. ขนาดของพูลหน่วยความจำระบุอยู่ในค่ากำหนดของระบบ Machine memory pool size (QMCHPOOL). ไม่มีงานของผู้ใช้ใดๆ รันในพูลหน่วยความจำนี้. (บนจอแสดงผล Work with System Status (WRKSYSSTS), พูลหน่วยความจำ Machine จะปรากฏเป็น system pool identifier 1.)

*BASE พูลหน่วยความจำ Base, ที่ถูกระบุเป็น Base ใน iSeries Navigator, มีแหล่งเก็บข้อมูลหลักที่ไม่ได้จัดสรร ทั้งหมดบนระบบ, (แหล่งเก็บข้อมูลหลักทั้งหมดที่พูลหน่วยความจำอื่น ไม่ต้องการ). Base pool มีหน่วยความจำที่สามารถถูกแบ่งใช้ระหว่างหลายระบบย่อยได้. พูลหน่วยความจำ Base ใช้สำหรับงานแบ็คซ์และฟังก์ชันระบบปลีกย่อย. ค่ากำหนดของระบบ Base memory pool minimum size (QBASPOOL) ระบุขนาดต่ำสุดของพูลหน่วยความจำ Base. activity level สำหรับพูลหน่วยความจำนี้ถูกระบุใน ค่ากำหนดของระบบ Base memory pool maximum eligible threads (QBASACTLVL). (บนจอแสดงผล Work with System Status (WRKSYSSTS), พูลหน่วยความจำ Base จะปรากฏเป็น system pool identifier 2.)

General Shared Pools

พูลแบบแบ่งใช้ทั่วไปคือ พูลของแหล่งเก็บข้อมูลหลักที่หลายระบบย่อยสามารถใช้ได้พร้อมกัน. บน อินเทอร์เน็ตแบบอักษร, พูลถูกระบุดังนี้:

- *INTERACT คือพูลของหน่วยความจำแบบโต้ตอบที่ใช้สำหรับงานแบบโต้ตอบ.
- *SPOOL คือพูลของหน่วยความจำที่ใช้สำหรับ spool writers.
- *SHRPOOL1 ถึง *SHRPOOL60 คือพูลของหน่วยความจำที่คุณสามารถใช้สำหรับการใช้งานของคุณเอง.

ใน iSeries Navigator, พูลแบบแบ่งใช้ทั่วไปถูกระบุเป็น Interactive, Spool, และ Shared 1 – Shared 60.

แบบแผนการกำหนดหมายเลขพูล

พูลมีแบบแผนการกำหนดหมายเลขอยู่สองชุด: ชุดหนึ่งใช้ภายใน ระบบย่อยและอีกชุดหนึ่งใช้ในระบบทั้งหมด. ระบบย่อยจะใช้ชุดของหมายเลขที่อ้างอิงถึงพูลที่ระบบย่อยใช้. ดังนั้น, เมื่อคุณสร้างหรือเปลี่ยน subsystem description คุณสามารถกำหนดหนึ่งหรือหลายพูลและกำหนดหมายเลขพูลเป็น 1, 2, 3, และต่อไป. นี่คือการกำหนดของพูลระบบย่อย, และไม่สอดคล้องกับหมายเลขพูล ที่แสดงบนจอแสดงผล Work with System Status (WRKSYSSTS).

มีการใช้ชุดของหมายเลขที่แตกต่างอื่นในการเก็บรักษาแฟร็กของพูลทั้งหมดบนระบบ. จอแสดงผล Work with Subsystems (WRKSBS) เกี่ยวข้องกับ subsystem pool identifiers และหัวข้อคอลัมน์ที่ system pool identifiers.

Work with Subsystems

System: XXXXXXXX

พิมพ์ตัวเลข, แล้วกด Enter.

4=จบระบบย่อย 5=แสดง subsystem description

8=ทำงานกับงานระบบย่อย

ผลรวม	-----Subsystem Pools-----									
Opt Subsystem Storage (M)	1	2	3	4	5	6	7	8	9	10
_ NYSBS	.48	2	4	5						
_ PASBS	.97	2	6	5						
_ QINTER	11.71	2	3							

ล่างสุด

พารามิเตอร์หรือคำสั่ง

==>

F3=ออก F5=รีเฟรช F11=แสดงข้อมูลระบบ F12=ยกเลิก

F14=ทำงานกับสถานะของระบบ

วิธีการกำหนดหมายเลขพูล - ตัวอย่าง

ตัวอย่างต่อไปนี้ สาธิตวิธีการกำหนดหมายเลขพูล.

ระบบย่อย			
CRTSBSDQINTER		CRTSBSDNYSBS	CRTSBSDPASBS
Pools (1 *BASE)		Pools (1 *BASE)	Pools (1 *BASE)
(2 1200 25)		(2 500 3)	(2 1000 3)
		(3 *SHRPOOL2)	(3 *SHRPOOL2)
(พูลระบบ 2, 3)		(พูลระบบ 2, 4, 5)	(พูลระบบ 2, 5, 6)

หลังจากที่ QINTER เริ่มต้น, พูลต่อไปนี้จะถูกจัดสรร:

System Pool Number	Description	QINTER
1	*Machine pool	
2	*BASE pool	1
3	QINTER private pool	2

หลังจากที่ NYSBS เริ่มต้น พูลต่อไปนี้จะถูกจัดสรร:

System Pool Number	Description	QINTER	NYSBS
1	*MACHINE pool		

System Pool Number	Description	QINTER	NYSBS
2	*BASE pool	1	1
3	QINTER private pool	2	
4	NYSBS private pool		2
5	*SHRPOOL2 shared pool		3

หลังจากที่ PASBS เริ่มต้น พูลต่อไปนี้จะถูกจัดสรร:

System Pool Number	Description	QINTER	NYSBS	PASBS
1	*MACHINE pool			
2	*BASE pool	1	1	1
3	QINTER private pool	2		
4	NYSBS private pool		2	
5	SHRPOOL2 shared pool		3	3
6	PASBS private pool			2

การจัดสรรพูลหน่วยความจำ

เมื่อคุณเริ่มระบบย่อย, ระบบจะพยายามจัดสรรพูลของหน่วยความจำ ที่ผู้ใช้กำหนด ซึ่งถูกกำหนดใน subsystem description ของ ระบบย่อยที่เริ่มต้น.

ถ้าระบบไม่สามารถจัดสรรหน่วยความจำทั้งหมดที่ร้องขอ, ระบบจะจัดสรร หน่วยความจำมากที่สุดเท่าที่พร้อมใช้งาน แล้วจัดสรรหน่วยความจำส่วนที่เหลือ เมื่อหน่วยความจำพร้อมใช้งานได้. ตัวอย่างเช่น, ให้พิจารณาตารางต่อไปนี้. ถ้ามีหน่วยความจำพร้อม ใช้งานอยู่ 700KB, และถ้า *SHRPOOL2 ถูกกำหนด 500KB, ผลคือหน่วยความจำ 300KB จะถูกจัดสรรให้พูลของหน่วยความจำแรกและ 400KB ถูกจัดสรรให้พูลของหน่วยความจำที่สอง.

Pool ID Specified in SBS	1	2
Storage Requested	300K	*SHRPOOL2
System Pool ID	3	4
Storage Allocated	300K	400K
Activity Level	1	
Pool Type	Private	Shared

พูลของหน่วยความจำที่คุณกำหนดจะลดขนาดของ พูลหน่วยความจำ Base เมื่อพูลถูกจัดสรร. ระบบจะจัดสรรหน่วยความจำมากที่สุดเท่าที่พร้อม ใช้งานได้ในพูลหน่วยความจำ Base ไปที่พูลส่วนตัว. ค่ากำหนดของระบบ Base memory pool minimum size (QBASPOOL) กำหนดขนาดต่ำสุดของ Base pool.

activity level ของพูลหน่วยความจำ

activity level ของพูลหน่วยความจำ คือ จำนวนของ thread ที่สามารถใช้ CPU ในสถานะแอดทีฟได้ในเวลาเดียวกันในพูลหน่วยความจำ. สิ่งนี้ช่วยให้สามารถใช้รีซอร์สของระบบได้อย่างมีประสิทธิภาพ. ระบบจัดการการควบคุมของ activity level.

บ่อยครั้งในระหว่างการประมวลผล ใน thread, โปรแกรมจะรอรีซอร์สระบบหรือการตอบกลับจากผู้ใช้งาน เวิร์กสเตชัน. ในระหว่างการรอนั้น, thread จะยกเลิกการใช้งานของ activity level ของพูลหน่วยความจำเพื่อให้ thread อื่นที่พร้อมจะประมวลผลได้สามารถเข้ามาทำงานแทนที่ไปก่อน.

เมื่อมี threads ถูกเริ่มมากกว่า thread ที่สามารถรันได้พร้อมกัน threads ที่มากเกินไปต้องรอที่จะใช้หน่วยการประมวลผล (โดยปกติการรอนี้ใช้เวลาไม่นานนัก). activity level ของพูลหน่วยความจำจะให้คุณจำกัดระดับการแย่งใช้หน่วยความจำหลักในพูลหน่วยความจำทั้งหลายในระบบย่อยของคุณ.

จำนวน thread ที่กำลังรันอยู่ (หรือ active thread) คือ จำนวน thread ที่มีสิทธิ์ในการแย่งใช้โปรเซสเซอร์ และมีส่วนใน activity level ของพูลหน่วยความจำ. ในลักษณะเช่นนี้เอง, active thread จะไม่รวมถึง thread ที่กำลังรออินพุต, รอข้อความ, รอได้รับการจัดสรรอุปกรณ์เพื่อใช้งาน, หรือรอไฟล์เพื่อเปิดใช้. active thread จะไม่รวมถึง thread ที่ไม่มีสิทธิ์ (คือ thread ที่พร้อมจะรันแต่ activity level ของพูลหน่วยความจำได้ขึ้นไปถึงจุดสูงสุดแล้ว).

activity level ทำงานอย่างไร

thread มากกว่าหนึ่งอันจะแอดทีฟได้ในเวลาเดียวกันในพูลหน่วยความจำ เนื่องจากการประมวลผล thread อาจถูกอินเทอร์รัプトเพียงชั่วขณะเท่านั้น ในขณะที่ข้อมูลที่ต้องการกำลังถูกเรียกออกมาจากหน่วยความจำสำรอง. ในระหว่างช่วงหน่วงนี้, ซึ่งโดยปกติจะใช้เวลาไม่นาน, thread อื่นสามารถรันได้. ด้วยการใช้ activity level, เครื่องสามารถประมวลผล threads ในพูลหน่วยความจำได้เป็นจำนวนมากในขณะที่เดียวกับที่ยังสามารถรักษาระดับของ contention ตามข้อจำกัด ที่คุณระบุ.

activity level สูงสุด

หลังจากใช้งานถึง activity level สูงสุดสำหรับพูลหน่วยความจำแล้ว, threads เพิ่มเติมที่ต้องการพูลหน่วยความจำจะถูกวางในสภาพที่ไม่มีสิทธิ์เพื่อรอให้จำนวนของ active threads ในพูลหน่วยความจำลดลง ต่ำกว่า activity level สูงสุด หรือรอให้ thread ใช้เวลาของการแบ่งเวลาจนหมด. เมื่อใดก็ตามที่ thread ไม่ได้ใช้พูลหน่วยความจำแล้ว, thread อื่นๆที่ยังไม่แอดทีฟก็จะมีสิทธิ์ที่จะรันตามระดับความสำคัญของตัวมัน. ตัวอย่างเช่น, หาก thread ที่รันอยู่กำลังรอการตอบกลับมาจากเวิร์กสเตชัน, thread นั้นก็จะปล่อย activity level ที่ทิ้งไปทำให้ activity level ลดลงมาถึงระดับสูงสุด.

การกำหนด activity level ของพูลหน่วยความจำ

การกำหนดพูลหน่วยความจำและ activity level ให้ถูกต้องนั้น โดยทั่วไปแล้ว ขึ้นอยู่กับขนาดของพูลหน่วยความจำ, จำนวนของ CPU, จำนวนของ disk unit arms, และลักษณะของแอปพลิเคชัน.

งาน

งาน (Work) ทั้งหมดที่ทำอยู่บนระบบจะกระทำผ่านชิ้นงาน (job). งานแต่ละงานมีชื่อที่ไม่ซ้ำกันภายในระบบ. งานทั้งหมด, ยกเว้นงานระบบ, จะรันภายใต้ระบบย่อย. งานสามารถป้อนระบบย่อยจาก work entries ใดๆ, เช่น job queue entry, workstation entry, communications entry, autostart job entry, หรือ prestart job entry.

งานแอดทีฟแต่ละงานประกอบด้วย thread (initial thread) อย่างน้อยหนึ่ง thread และอาจประกอบด้วย thread สำรองเพิ่มเติมด้วย. thread เป็นหน่วยย่อยอิสระของชุดงาน. แอ็ททริบิวต์ของงานถูกแบ่งใช้ระหว่าง thread ต่างๆ ของงาน, อย่างไรก็ตาม, thread ยังมีชุดแอ็ททริบิวต์ของตนเองต่างหาก, เช่น call stack. แอ็ททริบิวต์ของงานประกอบด้วยข้อมูลเกี่ยวกับวิธีที่งานจะถูก

ประมวลผล. งานจะทำหน้าที่เป็นเจ้าของแอตทริบิวต์ที่แบ่งใช้กันระหว่าง thread ภายในงานนั้น. การจัดการระบบงานจะจัดเตรียมวิธีควบคุมใ้งานดำเนิน ไปโดยสำเร็จสมบูรณ์บนระบบผ่านทางแอตทริบิวต์ของงาน.

สิทธิในการใช้งานที่เหมาะสม

เพื่อทำการเปลี่ยนแปลงแอตทริบิวต์ของงานส่วนใหญ่ คุณต้องมีสิทธิพิเศษ job control (*JOBCTL) หรือโปรไฟล์ผู้ใช้ของคุณ ต้องตรงกับ job user identity ของงานที่จะเปลี่ยน อย่างใดอย่างหนึ่ง.

มีสองสามแอตทริบิวต์ที่จำเป็นต้องใช้สิทธิพิเศษ *JOBCTL ในการเปลี่ยนแปลง. แอตทริบิวต์เหล่านี้คือ:

- ช่วงเวลารอดดีฟอลต์
- ระดับความสำคัญของการรัน
- การแบ่งเวลา

หมายเหตุ: ถ้าคุณวางแผนที่จะเปลี่ยน accounting code ของคุณ, คุณต้องมีสิทธิ *USE ที่คำสั่ง Change Accounting Code (CHGACGCDE) เพิ่มเติมจากสิทธิพิเศษ *JOBCTL หรือโปรไฟล์ผู้ใช้ที่ตรงกับ job user identity ของงาน.

สำหรับแอตทริบิวต์งานใดๆ ที่อ้างอิงถึงอ็อบเจกต์ iSeries , เช่น คิวงาน, เอาต์พุตคิว, และตารางลำดับ, คุณต้องมีสิทธิที่ถูกต้องที่อ็อบเจกต์. สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับสิทธิ iSeries , ให้อ่าน Appendix D Authorities Required for Objects Used by

Commands ใน iSeries Security Reference  .

ลักษณะงาน

การจัดการระบบงานจัดเตรียมวิธีการควบคุมใ้งานดำเนิน ไปโดยสำเร็จสมบูรณ์บนระบบผ่านทางแอตทริบิวต์ของงาน. อย่างไรก็ตาม, ก่อนที่คุณจะสามารถควบคุมลักษณะหลากหลายของงานได้, คุณต้องเข้าใจลักษณะต่างๆ ของงาน.

ข้อมูลต่อไปนี้อธิบายลักษณะของงาน:

ไวยากรณ์ชื่องาน:

เพื่อให้การควบคุมและการระบุงานบนระบบง่ายขึ้น, แต่ละงาน จึงมีชื่องานที่ถูกต้องไม่ซ้ำกัน. ชื่องานที่ถูกต้องประกอบด้วยสามส่วน: ชื่องาน (หรือชื่องานธรรมดา), ชื่อผู้ใช้, และหมายเลขงาน.

- สำหรับงานแบบโต้ตอบ, ชื่องานคือชื่อเดียวกันกับชื่อของ เวิร์กสเตชันหรืออิมูเลเตอร์เซสชันที่คุณได้ signed on. สำหรับงานแบ็ตซ์ คุณสามารถระบุชื่องานของคุณเองได้. ชื่องานสามารถยาวได้ถึง 10 อักขระ.
- ชื่อผู้ใช้คือชื่อของโปรไฟล์ผู้ใช้ซึ่งงานถูกเริ่มอยู่ภายใต้. สำหรับงานแบบโต้ตอบ, ชื่อผู้ใช้คือโปรไฟล์ผู้ใช้ที่ใช้ในการ signon ที่ ระบบ. นี่คือนามผู้ใช้ที่คุณป้อนในฟิลด์ผู้ใช้นบนจอแสดงผล signon. ถ้าคุณใช้ Telnet และข้ามการ signon, ชื่อนี้คือ ชื่อผู้ใช้ที่คุณใช้ sign on โดยอัตโนมัติที่ระบบ. สำหรับงานแบ็ตซ์ คุณสามารถระบุโปรไฟล์ผู้ใช้ซึ่งงานแบ็ตซ์จะถูกรันอยู่ภายใต้ได้. ชื่อผู้ใช้สามารถยาวได้ถึง 10 อักขระ.
- หมายเลขงานคือหมายเลขเฉพาะที่กำหนดโดยระบบ เพื่อให้คุณสามารถระบุถึง งานได้, แม้ว่ามีงานที่มีชื่องานและชื่อผู้ใช้เหมือนกันอยู่มากกว่าหนึ่งงาน. หมายเลขงานคือตัวเลข 6 หลักเสมอ.

ไวยากรณ์

ไวยากรณ์สำหรับชื่องานที่ถูกต้องเหมือนกับไวยากรณ์ ของชื่อที่ถูกต้องสำหรับอ็อบเจกต์. ตัวอย่างเช่น, ถ้าชื่องานคือ DSP01, ผู้ใช้คือ QPGMR, และหมายเลขงานคือ 000578, ชื่องานที่ถูกต้องที่ป้อนบนคำสั่ง Work with Job (WRKJOB) คือ:

WRKJOB JOB(000578/QPGMR/DSP01)

อีกอย่างหนึ่งที่คล้ายกับชื่ออ็อบเจกต์คือ คุณไม่ต้องระบุ qualifier ทั้งหมด. เช่น ลองพิจารณาดังนี้:

WRKJOB JOB(QPGMR/DSP01)

หรือ

WRKJOB JOB(DSP01)

สิ่งนี้ให้ผลเหมือนกับการป้อนชื่องานที่ถูกต้องทั้งหมด. ถ้างานหลายงาน บนระบบมีส่วนที่ตรงกับชื่องานที่คุณป้อน, จอแสดงผล Select Job จะปรากฏขึ้น. จอแสดงผลนี้อนุญาตให้คุณเลือกงานที่คุณ ต้องการจากรายการของชื่องานที่เข้าได้.

แอ็ททริบิวต์งาน:

แอ็ททริบิวต์งานกำหนดวิธีการที่ระบบใช้รันงานแต่ละงาน. แอ็ททริบิวต์งาน บางแอ็ททริบิวต์ถูกเซ็ทจากโปรไฟล์ผู้ใช้. แอ็ททริบิวต์งานอื่นมาจากค่ากำหนดของระบบ, จากโลแคล, จากคำสั่ง Submit Job (SBMJOB), จากรายละเอียดของงาน, และจากคำสั่ง Change Job (CHGJOB) (ซึ่งคุณสามารถเปลี่ยนค่าสำหรับแอ็ททริบิวต์ในขณะที่งานกำลังรันอยู่).

แอ็ททริบิวต์การควบคุมงานช่วยให้คุณเลือกที่จะควบคุมงานที่ ระดับงาน, ระดับผู้ใช้, หรือระดับระบบ. ตัวอย่างเช่น, คุณสามารถให้ระบบของคุณ ตั้งค่าค่ากำหนดของระบบสำหรับแอ็ททริบิวต์งาน (ซึ่งคือค่าดีฟอลต์ของระบบ). จากนั้นถ้าคุณ ต้องการเปลี่ยนค่าสำหรับ งานใหม่ทั้งหมดบนระบบ, คุณสามารถจะเปลี่ยนค่ากำหนดของระบบได้.

โดยการระบุค่าในรายละเอียดของงาน, คุณสามารถเปลี่ยนชนิดของงานทั้งหมด ที่ใช้รายละเอียดของงานนั้นได้. ตัวอย่างเช่น, ถ้างานแบ็ตช์ทั้งหมดของคุณ ใช้รายละเอียดของงานเดียวกัน, การเปลี่ยนรายละเอียดของงานสำหรับงานแบ็ตช์ สามารถส่งผลกระทบต่องานแบ็ตช์ทั้งหมดของคุณและทำให้ไม่มีผลต่องานอื่นทั้งหมดได้.

รายละเอียดของงาน:

รายละเอียดของงานอนุญาตให้คุณสร้างชุดของแอ็ททริบิวต์งาน ที่จะถูกบันทึกและพร้อมใช้งานสำหรับการใช้ในลักษณะต่างๆ. รายละเอียดของงานสามารถใช้เป็น ซอร์สสำหรับแอ็ททริบิวต์งานบางอย่างที่บอกให้ระบบทราบถึงวิธีการ รันงาน. แอ็ททริบิวต์บอกให้ระบบทราบว่า จะเริ่มงานเมื่อไร, เรียกงาน มาจากที่ไหน, และจะรันงานอย่างไร. คุณสามารถคิดได้ว่ารายละเอียดของงาน เป็นเท็มเพลตที่งานหลายงานสามารถใช้ได้, ดังนั้นจึงลดจำนวนของพารามิเตอร์ เฉพาะที่คุณต้องเซ็ทสำหรับงานแต่ละงาน.

รายละเอียดของงานถูกใช้โดยงานแบบ autostart, แบบแบ็ตช์, แบบโต้ตอบ, และ งานแบบ prestart. คุณสามารถใช้รายละเอียดของงาน เดียวกันสำหรับงานหลายๆงานได้. เมื่อคุณ กำหนดงาน, คุณสามารถใช้รายละเอียดของงานในหนึ่งจากสองวิธีดังนี้:

- ใช้รายละเอียดของงานที่ระบุโดยไม่แทนที่ค่าเดิมของแอ็ททริบิวต์. ตัวอย่างเช่น:

SBMJOB JOB(OEDAILY) JOBDBATCH

- ใช้รายละเอียดของงานที่ระบุแต่แทนที่ค่าเดิมบางแอ็ททริบิวต์ (โดยใช้คำสั่ง BCHJOB หรือ SBJOB). ตัวอย่างเช่น, ถ้าต้องการแทนที่ค่าเดิมของไฟล์บันทึกข้อความในรายละเอียดของงาน QBATCH, ให้ระบุ:

SBMJOB JOB(OEDAILY) JOBDBATCH
LOG(2 20 *SECLVL)

หมายเหตุ: คุณไม่สามารถแทนที่ค่าเดิมแอ็ททริบิวต์รายละเอียดของงานใดๆ สำหรับงานแบบ autostart, งานเวิร์กสแตชัน, หรืองานการสื่อสาร.

รายละเอียดของงานและการรักษาความปลอดภัย:

งานทุกงานในระบบใช้รายละเอียดของงานในระหว่างการเริ่มงาน. สิ่งนี้ควบคุมแอตทริบิวต์ต่างๆ ของงาน. พารามิเตอร์ USER ควบคุมชื่อของโปรไฟล์ผู้ใช้ที่กำหนดที่งาน. รายละเอียดของงานที่มีชื่อโปรไฟล์ผู้ใช้ (USER) ที่ระบุควรถูกอนุญาตเฉพาะสำหรับผู้ใช้ที่ระบุเท่านั้น. ถ้าไม่, ที่ระดับความปลอดภัย 30 และต่ำกว่า, ผู้ใช้อื่นจะสามารถส่งงานเพื่อรันภายใต้โปรไฟล์ผู้ใช้นั้นได้.

ตัวอย่างเช่น, ลองพิจารณา

```
CRTJOB JOB(XX) USER(JONES) . . . AUT(*USE)
```

ตัวอย่างนี้ มีความเสี่ยงในการรักษาความปลอดภัยเนื่องจากผู้ใช้ใดๆ สามารถส่งงานโดยใช้รายละเอียดของงาน XX ได้, และได้รับอนุญาตในสิ่งที่ JONES ได้รับอนุญาตให้ทำ. ถ้ามีการใช้รายละเอียดของงานชนิดนี้บน workstation entry, รายละเอียดของงาน จะอนุญาตให้ทุกคนสามารถ sign on เป็นผู้ใช้นั้นโดยเพียงแค่กดปุ่ม Enter. เพื่อหลีกเลี่ยง exposure การรักษาความปลอดภัย, อย่านุญาตให้รายละเอียดของงานชนิดนี้เป็น *PUBLIC.

หมายเหตุ: ที่ระดับความปลอดภัย 40 และ 50, คำสั่ง Submit Job (SBMJOB) กำหนดว่าผู้ส่งงานต้องได้รับอนุญาต (*USE) ในการใช้โปรไฟล์ผู้ใช้ ที่ระบุชื่อในรายละเอียดของงาน. สิ่งนี้สมมติว่า SBJJOB ระบุผู้ใช้ (*JOB). ถึงกระนั้นก็ตาม, ให้หลีกเลี่ยงการระบุผู้ใช้ในรายละเอียดของงาน ยกเว้น ว่ามีความจำเป็นเนื่องจากเหตุผลเฉพาะบางอย่าง (เช่น งานแบบ autostart) และ คุณควบคุมการเข้าใช้งานอย่างแน่นหนา.

USER Parameter and Interactive Jobs

รายละเอียดของงาน ที่จะใช้ถูกกำหนดบนคำสั่ง Add Work Station Entry (ADDWSE). ค่าดีฟอลต์คือการใช้รายละเอียดของงานในโปรไฟล์ผู้ใช้. ถ้า USER(*RQD) ถูกระบุในรายละเอียดของงาน, ผู้ใช้ต้องป้อนชื่อผู้ใช้. ถ้ามีการระบุ USER(XXXX) (โดยที่ XXXX คือชื่อโปรไฟล์ผู้ใช้เฉพาะ), ผู้ใช้จะได้รับอนุญาตให้กดปุ่ม Enter บนจอแสดงผล signon และดำเนินการ ภายใต้ชื่อโปรไฟล์ผู้ใช้ XXXX, ยกเว้นว่าระดับความปลอดภัยจะเป็น 40 หรือสูงกว่า.

USER Parameter and Batch Jobs

รายละเอียดของงาน ที่ใช้สำหรับงานแบตช์ถูกระบุบนคำสั่ง Submit Job (SBMJOB) หรือ Batch Job (BCHJOB).

ถ้ามีการป้อน input stream ที่มีคำสั่ง BCHJOB, ผู้ใช้ที่ป้อนหนึ่งในคำสั่ง Start Reader commands (STRDBRDR, STRDKTRDR) หรือหนึ่งในคำสั่ง Submit Job (SBMDBJOB, SBMDKTJOB, และอื่นๆ.) ต้องมีสิทธิการดำเนินการอีอบเจกต์ (*OBJOPR) ที่รายละเอียดของงาน ที่ถูกระบุ. เมื่อใช้ input stream, งานจะดำเนินการภายใต้โปรไฟล์ผู้ใช้ของรายละเอียดของงานเสมอ ไม่ใช่ของผู้ใช้ที่วางงาน บนคิวงาน. ถ้ามีการระบุ USER(*RQD) ในรายละเอียดของงาน, จะไม่สามารถใช้รายละเอียดของงานบนคำสั่ง BCHJOB .

ถ้าใช้ คำสั่ง SBJJOB, คำสั่งจะใช้ค่าดีฟอลต์เพื่อให้งานแบตช์ดำเนินการภายใต้ชื่อโปรไฟล์ผู้ใช้ของผู้ส่งงาน. อย่างไรก็ตาม, ถ้ามีการระบุ USER(*JOB) บนคำสั่ง SBJJOB, งาน จะดำเนินการภายใต้ชื่อที่ระบุในพารามิเตอร์ USER ของรายละเอียดของงาน.

บ่อยครั้งที่ ต้องใช้ชื่อที่ระบุในรายละเอียดของงานเพื่ออนุญาตให้ผู้ใช้ส่งงาน สำหรับโปรไฟล์ผู้ใช้ที่ระบุ. ตัวอย่างเช่น, รายละเอียดของงาน QBATCH ถูกจัดส่งมา พร้อมด้วย USER(QPGR) เพื่อให้อนุญาตสิ่งนี้. เพื่อหลีกเลี่ยงความเสี่ยง การรักษาความปลอดภัย, อย่านุญาตให้รายละเอียดของงานชนิดนี้เป็น *PUBLIC.

Call stacks:

call stack คือรายการที่จัดลำดับของโปรแกรมหรือโปรซีเดอร์ทั้งหมดที่รันอยู่ในปัจจุบันสำหรับงาน. โปรแกรมและโปรซีเดอร์สามารถ เริ่มต้นได้โดยตรงด้วยคำสั่ง CALL, หรือโดยทางอ้อมจากเหตุการณ์อื่น.

call stack พร้อมใช้งานทั้งระดับงานและระดับ thread. บนอินเทอร์เฟซแบบอักษระ, call stack คือรายการ last-in-first-out (LIFO) ของ call stack entries, หนึ่ง entry สำหรับแต่ละโปรซีเดอร์หรือโปรแกรมที่เรียก. ใน iSeries Navigator, โดยค่าดีฟอลต์, entry หลังสุดใน stack จะปรากฏอยู่ บนสุดของรายการ. อย่างไรก็ตาม, สามารถเปลี่ยนลำดับได้โดยใช้ปุ่ม **Sort ascending** หรือ **Sort descending**.

ข้อมูลที่มีอยู่ในจอแสดงผล Call Stack ได้แก่ ข้อมูลการเรียกใช้สำหรับ original program model (OPM), integrated language environment (ILE), i5/OS Portable Application Solutions Environment (PASE), และ Java applications. นอกจากนี้, ถ้าคุณรันอยู่ภายใต้โปรไฟล์ผู้ใช้ที่มีสิทธิพิเศษ *SERVICE, คุณสามารถดู entries เพิ่มเติมสำหรับ licensed internal code (LIC) และ i5/OS PASE Kernel ได้.

คลาสอ็อบเจกต์:

คลาสอ็อบเจกต์มีแอตทริบิวต์การรันที่ควบคุมสถานะแวดล้อมรันไทม์ ของงาน. คลาสอ็อบเจกต์ที่ได้มาจาก IBM, หรือคลาส, ใช้ได้กับทั้ง แอ็พพลิเคชันแบบโต้ตอบและแอ็พพลิเคชันแบบแบ็ตช์ตามปกติ. คลาสที่ให้มาพร้อมกับระบบ (ตามชื่อ) มีดังนี้:

- QGPL/QBATCH: สำหรับใช้โดยงานแบ็ตช์
- QSYS/QCTL: สำหรับใช้โดยระบบย่อยการควบคุม
- QGPL/QINTER: สำหรับใช้โดยงานแบบโต้ตอบ
- QGPL/QPGMR: สำหรับใช้โดยระบบย่อยการทำโปรแกรมมิง
- QGPL/QSPL: สำหรับใช้โดยเครื่องเขียนพริ้นเตอร์ระบบย่อยการสพูล
- QGPL/QSPL2: สำหรับการใช้สพูลทั่วไปในพูลของระบบ Base

แอตทริบิวต์รันไทม์

ข้อมูลต่อไปนี้เป็นรายการของแอตทริบิวต์รันไทม์, หรือพารามิเตอร์, บางส่วนที่พบในคลาสอ็อบเจกต์ ซึ่งมีความสำคัญต่อการจัดการระบบงาน.

ระดับความสำคัญการรัน (RUNPTY)

ตัวเลขที่ระบุถึงระดับความสำคัญที่ได้กำหนดให้แก่งานทั้งหมดที่รันอยู่โดยใช้คลาส. ระดับความสำคัญถูกใช้ในการกำหนดงาน, จาก งานทั้งหมดที่แย่งชิงรีซอร์สของระบบ, ที่จะรันถัดไป. ค่าอาจเป็นได้ตั้งแต่ 1 ถึง 99, โดยที่ 1 คือระดับความสำคัญสูงสุด (งานทั้งหมดที่มีระดับความสำคัญ 1 จะรันก่อน). ค่านี้คือระดับความสำคัญการรันสูงสุดที่ใช้ได้สำหรับ thread ใดๆ ภายในงาน. แต่ละ threads ภายในงานอาจมีระดับความสำคัญที่ต่ำกว่าได้. การเปลี่ยนระดับความสำคัญการรันของงานจะส่งผลกระทบต่อระดับความสำคัญการรันของ threads ทั้งหมดภายในงาน. ตัวอย่างเช่น, ถ้างานรันอยู่ที่ระดับความสำคัญ 10, thread A ภายในงานรันอยู่ที่ระดับความสำคัญ 10 และ thread B ภายในงานรันอยู่ที่ระดับความสำคัญ 15. ถ้าระดับความสำคัญของงานเปลี่ยนเป็น 20, ระดับความสำคัญของ thread A จะถูกปรับเปลี่ยนเป็น 20 และระดับความสำคัญของ thread B จะถูกปรับเปลี่ยนเป็น 25.

การแบ่งเวลา (TIMESLICE)

จำนวนสูงสุดของเวลาของโปรเซสเซอร์ (ในหน่วยมิลลิวินาที) ที่ให้แก่แต่ละ thread ในงานที่ใช้คลาสนี้ ก่อนที่ threads อื่นในงานหรืองานอื่นจะได้รับ โอกาสให้รัน. การแบ่งเวลาเป็นตัวกำหนดจำนวนเวลา ที่ต้องการโดย thread ในงานใน

การทำให้การประมวลผลที่สำคัญเสร็จสมบูรณ์. เมื่อสิ้นสุดการแบ่งเวลา, thread อาจถูกกำหนดให้มีสถานะ inactive เพื่อให้ threads อื่นสามารถแฉีกทีในพูลของหน่วยความจำได้.

ช่วงเวลารอดีฟอลต์ (DFTWAIT)

จำนวนดีฟอลต์ของเวลาที่ระบบจะคอยการสมบูรณ์ของ คำสั่งที่ดำเนินการรอ. ช่วงเวลารอนี้ใช้กับเวลาที่ คำสั่งกำลังรอการดำเนินการของระบบ, ไม่ได้ใช้กับเวลาที่คำสั่ง กำลังรอการตอบสนองจากผู้ใช้. โดยทั่วไป, ช่วงเวลานี้เป็นเวลาที่คุณเต็มใจที่จะรอรอบก่อนที่จะจบคำขอ. ถ้าผ่านพ้น ช่วงเวลารอแล้ว, ข้อความแสดงความผิดพลาดจะถูกส่งผ่านไปทำงาน. จะใช้ช่วงเวลารอดีฟอลต์ นี้ถ้าไม่มีการระบุช่วงเวลารอค่าอื่นสำหรับสถานการณ์ที่กำหนดให้.

ช่วงเวลารอที่ใช้สำหรับการจัดสรรรีซอร์สของไฟล์ถูกระบุอยู่ใน file description และสามารถแทนที่ค่าเดิมด้วยคำสั่งแทนที่ค่าเดิม. คำสั่งนี้ระบุว่า ให้ใช้ช่วงเวลารอที่ระบุในคลาสอ็อบเจกต์. ถ้ารีซอร์สของไฟล์ไม่มีพร้อมใช้งานเมื่อเปิดไฟล์, ระบบจะรอจนกว่าช่วงเวลารอจะ สิ้นสุดลง.

หมายเหตุ: คลาสแอ็ททริบิวต์ใช้กับขั้นตอนการเรอต์แต่ละชั้น ของงาน. งานส่วนใหญ่มีขั้นตอนการเรอต์เพียงชั้นเดียว, แต่ถ้างานถูกรีเรอต์ (เนื่องจากบางสิ่งเช่นคำสั่ง Reroute Job (RRTJOB) หรือ Transfer Job (TFRJOB)) คลาสแอ็ททริบิวต์จะถูก รีเซ็ต.

เวลา CPU สูงสุด (CPUTIME)

จำนวนสูงสุดของเวลาของโพรเซสเซอร์ที่อนุญาตสำหรับการประมวลผล ขั้นตอนการเรอต์ของงานให้เสร็จสมบูรณ์. ถ้าขั้นตอนการเรอต์ของงานไม่เสร็จสิ้นภายในเวลานี้, ขั้นตอนจะจบลง, และมีการบันทึกข้อความที่บันทึกการใช้งาน.

หน่วยเก็บชั่วคราวสูงสุด (MAXTMPSTG)

จำนวนสูงสุดของหน่วยเก็บชั่วคราวที่ขั้นตอนการเรอต์ของงาน สามารถใช้ได้. หน่วยเก็บชั่วคราวนี้ใช้สำหรับโปรแกรมที่รันอยู่ในงาน, สำหรับอ็อบเจกต์ระบบที่ใช้เพื่อสนับสนุนงาน, และสำหรับอ็อบเจกต์ชั่วคราว ที่งานสร้างขึ้น.

thread สูงสุด (MAXTHD)

จำนวนสูงสุดของ threads ซึ่งงานในคลาสนี้สามารถรันได้ในทุกเมื่อ. ถ้าหลาย thread ถูกเริ่มต้นพร้อมกัน, ค่านี้อาจมากเกินได้. thread ที่มากเกินจะได้รับอนุญาตให้รันจนเสร็จสมบูรณ์ตามปกติ. แต่การเริ่มต้น threads เพิ่มเติมจะถูกห้าม จนกว่าจำนวนสูงสุดของ threads ในงานจะลดลงต่ำกว่าค่าสูงสุดนี้.

หมายเหตุ: รีซอร์สที่ใช้โดย threads และรีซอร์สที่พร้อมใช้งานบนระบบอาจไม่ตรงกัน. ด้วยเหตุนี้, การเริ่มต้น threads เพิ่มเติมอาจถูกห้ามก่อนที่จะใช้ thread ถึงจำนวน สูงสุดนี้.

Job user identity:

job user identity (JUID) คือชื่อของโปรไฟล์ผู้ใช้ซึ่งทำให้งานนี้เป็นที่รู้จักของงานอื่น. ชื่อนี้ใช้สำหรับการตรวจสอบการให้สิทธิเมื่องานอื่นพยายามดำเนินการกับงานนี้.

บางตัวอย่างของฟังก์ชันที่ดำเนินการกับงานอื่น ได้แก่ คำสั่ง Start Service Job (STRSRVJOB), Retrieve Job Information (QUSRJOBI) API, Change Job (QWTCGJB) API, คำสั่งการ ควบคุมงานทั้งหมด, และฟังก์ชันที่ส่งสัญญาณจากงานหนึ่งไปยังงานอื่น.

ในสถานการณ์ที่งานสลับค่าโปรไฟล์ผู้ใช้, โปรไฟล์ผู้ใช้ปัจจุบันระบุ โปรไฟล์ซึ่ง initial thread กำลังรันอยู่ภายใต้ แทนที่จะเป็น JUID.

ไม่ใช่ JUID ในการตรวจสอบการให้สิทธิจากภายในงาน . การให้สิทธิเพื่อปฏิบัติฟังก์ชันจะใช้โปรไฟล์ผู้ใช้ปัจจุบันของ thread ซึ่งฟังก์ชันถูกเรียกเสมอ.

เมื่องานอยู่บนคิวงานหรือเอาต์พุตคิว, JUID จะเหมือนกับชื่อผู้ใช้ ของงานเสมอ และไม่สามารถเปลี่ยนได้.

เมื่องานเริ่มขึ้น, และที่จุดเริ่มต้นของขั้นตอนการเรอต์ที่ตามมาใดๆ, JUID คือชื่อที่เหมือนกับชื่อของโปรไฟล์ผู้ใช้ปัจจุบันของงาน. ในขณะที่งานแอ็คทีฟ, สามารถเปลี่ยน JUID ในวิธีดังต่อไปนี้.

- JUID สามารถถูกเซตโดยตรงโดยแอฟพลิเคชันโดยใช้ Set Job User Identify (QWTSJUID) application program interface (API) หรือฟังก์ชัน QwtSetJuid(). JUID ถูกเซตด้วยชื่อของโปรไฟล์ผู้ใช้ที่ thread ซึ่งเรียก API หรือฟังก์ชันกำลังรันอยู่ ภายใต้โปรไฟล์นั้น.
- JUID สามารถถูกลบโดยตรงโดยแอฟพลิเคชันโดยใช้ QWTSJUID API หรือฟังก์ชัน QwtClearJuid(). งานต้องกำลังรันเป็นงานแบบ single threaded ในขณะนั้น. เมื่อลบ, JUID จะถูกเซตโดยทางอ้อมโดยระบบเป็น ชื่อของโปรไฟล์ผู้ใช้ที่ single thread ของงานกำลังรันภายใต้ ณ จุดนั้น.
- ถ้างานกำลังรันเป็นงานแบบ single threaded, และ JUID ไม่ได้ถูกเซต โดยตรงโดยแอฟพลิเคชัน, ผลคือ ในแต่ละครั้งที่งานใช้ Set Profile (QWTSETP) API เพื่อรันภายใต้โปรไฟล์ผู้ใช้อื่น JUID จะถูกเซตโดยทางอ้อม โดยระบบเป็นชื่อของโปรไฟล์ผู้ใช้ที่ถูกเซตโดย QWTSETP.
- เมื่องานแบบ single threaded เริ่ม secondary thread และ JUID ยังไม่ได้ถูก เซตโดยตรงโดยแอฟพลิเคชัน, ผลคือ ระบบจะเซต JUID โดยทางอ้อมด้วยชื่อของโปรไฟล์ผู้ใช้ที่ single thread ของงาน กำลังรันอยู่ภายใต้ ณ จุดที่งานเริ่ม secondary thread.
เมื่องาน กลับไปเป็น single thread, ระบบจะเซต JUID โดยทางอ้อมเป็นชื่อของ โปรไฟล์ผู้ใช้ที่ single thread ของงานกำลังรันอยู่ภายใต้ ณ จุดนั้น.

ตัวอย่าง Job user identity:

หัวข้อนี้มีตัวอย่างที่จะช่วยให้คุณเข้าใจได้ดีขึ้นเกี่ยวกับวิธีการกำหนด job user identity (JUID) ในสถานการณ์ต่างๆ .

- งานรันอยู่ภายใต้โปรไฟล์ผู้ใช้ที่มีชื่อว่า USERA. JUID คือ USERA. ถ้างาน ใช้ QWTSETP API เพื่อสลับไปที่ USERB, JUID จะเปลี่ยนเป็น USERB.

ใน สถานการณ์นี้, ค่า Set By สำหรับ JUID คือ *DEFAULT. เนื่องจากงาน กำลังรันแบบ single-threaded, ดังนั้น job user identity คือโปรไฟล์ผู้ใช้ปัจจุบัน ซึ่ง initial thread ของงานกำลังรันอยู่ภายใต้ (ยกเว้นว่า, job user identity จะถูกเซตโดยตรงโดยแอฟพลิเคชัน). สำหรับงานคิวงาน และงานที่เสร็จสมบูรณ์, job user identity คือชื่อผู้ใช้จากชื่อ งาน ที่ถูกต้อง.

- งาน single-threaded รันภายใต้โปรไฟล์ผู้ใช้ USERX. JUID คือ USERX. ถ้างานเริ่ม secondary threads, JUID จะยังคงเป็น USERX. ถ้าจากนั้น threads ทั้งหมดสลับค่าเป็น USERY, JUID จะยังคงเป็น USERX.

ในสถานการณ์นี้, ค่า Set By สำหรับ JUID คือ *SYSTEM. เนื่องจากงานนี้เป็นงานแอ็คทีฟซึ่ง กำลังรันในปัจจุบันเป็นงาน multi-threaded, ดังนั้น job user identity จึงถูกเซต โดยทางอ้อมโดยระบบ. job user identity จะถูกเซตเป็นชื่อของโปรไฟล์ผู้ใช้ซึ่งงานจะรันอยู่ภายใต้ ณ นั้น เมื่องานกลายเป็น multi-threaded. เมื่องานกลับ ไปรัน single-threaded, job user identity จะถูกรีเซตเป็นค่า *DEFAULT.

- ถ้าเซิร์ฟเวอร์ที่รันภายใต้โปรไฟล์ผู้ใช้ที่มีชื่อว่า SERVER เรียก QWTSJUID API, JUID จะถูกเซตเป็น SERVER. จากนั้น ถ้าเซิร์ฟเวอร์เรียก Set Profile (QWTSETP) API เพื่อเซตโปรไฟล์ผู้ใช้ปัจจุบันเป็น CLIENT ในขณะที่ประมวลผลงานใน

นามของไคลเอ็นต์นั้น, JUID ยังคงเป็น SERVER. ในลักษณะเดียวกัน, ถ้าเซิร์ฟเวอร์ เริ่ม secondary threads ที่แต่ละ thread เรียก QWTSETP เพื่อรันภายใต้ โปรไฟล์ผู้ใช้หลากหลายโปรไฟล์, JUID จะยังคงเป็น SERVER.

ในสถานการณ์นี้, ค่า Set By สำหรับ JUID คือ *APPLICATION. job user identity ถูกเซตโดยตรงโดย แอปพลิเคชัน โดยใช้ API. ค่านี้ใช้กับงานทั้งแบบ single-threaded และ multi-threaded .

thread:

คำว่า thread ย่อมาจาก "thread of control". thread คือพาทที่โปรแกรมใช้ในขณะทำงาน, ขั้นตอนที่ถูกปฏิบัติ, และลำดับ การปฏิบัติตามขั้นตอน. thread รันโค้ดจากตำแหน่งเริ่มต้น ในลำดับที่จัดลำดับ, ที่กำหนดไว้ล่วงหน้าสำหรับชุดของอินพุตที่กำหนดให้.

การใช้ thread ภายในงานช่วยให้สามารถทำสิ่งต่างๆ ได้ในเวลาเดียวกัน. ตัวอย่างเช่น, ขณะทำงานกำลังประมวลผล, thread อาจเรียกข้อมูลออกมาและคำนวณข้อมูลที่งานต้องการเพื่อให้การประมวลผลเสร็จสิ้น.

งานแอคทีฟแต่ละตัวมี thread อย่างน้อยหนึ่งตัว, ซึ่งเรียกว่า initial thread. initial thread จะถูกสร้างขึ้นเป็นส่วนหนึ่งของการเริ่มต้นของงาน. ใน threads บน iSeries Navigator, โดยค่าดีฟอลต์, คุณจะเห็น Initial เป็นชนิดของ thread แรกในรายการ. initial thread คือ thread แรกที่ถูกสร้างขึ้นภายในงานเมื่อเริ่มทำงาน.

ชนิด thread

ชนิด thread กำหนดวิธีการสร้าง thread บนระบบ.

ผู้ใช้ thread สามารถถูกสร้างขึ้นโดยแอปพลิเคชันลูกค้า. initial thread ในงานจะเป็น thread ผู้ใช้เสมอ. ฟิลด์ Allow multiple threads ต้องถูกเซตเป็น yes เพื่อให้สามารถใช้ threads ผู้ใช้หลายตัวได้.

ระบบ thread จะถูกสร้างขึ้นโดยระบบในนามของผู้ใช้. ฟังก์ชันระบบบางอย่างใช้ thread ระบบเพื่อเสร็จสิ้นการประมวลผล. หากแอปพลิเคชันของลูกค้าใช้ฟังก์ชันระบบที่ใช้ thread, thread ระบบจะถูกใช้งาน.

สิทธิในการใช้งาน thread ที่ถูกต้อง:

ต้องมีระดับสิทธิในการใช้งานบางอย่างก่อน คุณจึงจะสามารถทำงานกับ threads ได้.

เพื่อดูและเปลี่ยนแอตทริบิวต์ส่วนใหญ่ของ thread คุณต้องมีสิทธิพิเศษ *JOBCTL, หรือโปรไฟล์ผู้ใช้ของคุณต้องตรงกับ job user identity ของงานที่มี thread. หากต้องการเปลี่ยนระดับความสำคัญการรันของ thread, คุณต้องมีสิทธิพิเศษ *JOBCTL. สิทธิ Thread Control จะช่วยให้คุณดูบางแอตทริบิวต์ของ thread ได้.

หากต้องการพักหรือรีเซ็ต thread, คุณต้องมีสิทธิพิเศษ *JOBCTL หรือสิทธิ Thread Control, หรือโปรไฟล์ผู้ใช้ของคุณต้องตรงกับ job user identity ของงานที่มี thread. หากต้องการสิ้นสุด thread, คุณต้องมีสิทธิพิเศษ *SERVICE หรือสิทธิ Thread Control.

สำหรับแอตทริบิวต์ thread ใดๆ ที่อ้างอิงอ็อบเจกต์ iSeries, เช่น ไลบรารีใน รายชื่อไลบรารี, ผู้ใช้ต้องมีสิทธิที่ถูกต้องที่อ็อบเจกต์.

สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับสิทธิ iSeries, ให้อ่าน Appendix D Authorities Required for Objects Used by Commands ใน

iSeries Security Reference  .

หมายเหตุ: สิทธิ Thread Control ช่วยให้คุณสามารถดึงข้อมูลเกี่ยวกับ threads ของงานอื่นได้. Thread Control สามารถให้และเรียกคืนสำหรับผู้ใช้แต่ละรายได้โดยใช้ iSeries Navigator's Application Administration support, หรือโดยการใช้ Change Function Usage Information (QSYCHFUI) API, ด้วย function ID ของ QIBM_SERVICE_THREAD. สำหรับ ข้อมูลรายละเอียดเพิ่มเติมเกี่ยวกับการจัดการแอปพลิเคชัน, ให้ดู Information Center หัวข้อ Application Administration.

สถานะ thread:

สถานะปัจจุบันของ thread สามารถดูได้จากหน้า General ในหน้าต่าง Thread Properties, ภายใต้สถานะ Detailed.

ตัวอย่างของสถานะโดยละเอียดคือ:

กำลังรอดอนจากคิว

thread ของงานกำลังรอดอนการดำเนินการดอนจากคิวที่เสร็จสมบูรณ์. การดอนจากคิวคือการดำเนินการเพื่อย้ายข้อความจากคิว. ข้อความคือการสื่อสารที่ถูกส่งจากบุคคลหนึ่งหรือโปรแกรมหนึ่งไปยังอีกบุคคลหนึ่งหรือโปรแกรมหนึ่ง. โดยทั่วไป, ข้อความจะถูกวางคิว (วาง) ไว้บนอ็อบเจกต์ระบบคิวโดย thread หนึ่งตัวและถูกดอนออกจากคิว (ลบออก) โดย thread อีกตัวหนึ่ง.

หมายเหตุ: เมื่อการรอดอนจากคิวแสดงอยู่ในหน้าคุณสมบัติ, ข้อมูลเพิ่มเติมที่ระบุคิวที่กำลังรอดอนอยู่จะปรากฏขึ้นมา. เมื่องานหรือ thread กำลังรอให้การดำเนินการดอนจากคิว เสร็จสมบูรณ์สำหรับอ็อบเจกต์ i5/OS, คุณจะเห็นชื่ออ็อบเจกต์ 10 อักขระ, โลกบารี่ของอ็อบเจกต์, และชนิดอ็อบเจกต์. ถ้างานหรือ thread กำลังรอดอนการดำเนินการดอนออกจากคิวให้เสร็จสิ้นสำหรับอ็อบเจกต์ภายใน, คุณจะเห็นชื่ออ็อบเจกต์ที่ยาว 30 อักขระ. สำหรับอ็อบเจกต์ภายใน คุณต้องมีสิทธิพิเศษ job control (*JOBCTL) เพื่อดูชื่อที่ยาว 30 อักขระ.

สถานะโดยละเอียดสามารถแสดงค่าสถานะที่เชื่อมโยง, ซึ่งให้รายละเอียดเพิ่มเติม เกี่ยวกับสถานะปัจจุบันของ thread. ตัวอย่างของสถานะโดยละเอียดบวกค่าสถานะเชื่อมโยงคือ:

ถูกพัก (n)

thread แต่ละอันจะถูกพักไว้. ต่างจากงาน, thread อาจมีการพักไว้หลายครั้งในเวลาเดียวกัน. จำนวน (ตัวอย่างเช่น, Held (3)) ที่ตามหลังสถานะ thread จะเป็นตัวบอกผู้ใช้ถึงจำนวนครั้งที่ thread ถูกพักโดยไม่ถูกรีลีส. ตัวอย่างเช่น, หาก thread ถูกพักไว้สามครั้งและถูกรีลีสหนึ่งครั้ง, thread จะมีการพักสองครั้ง. จำนวนจะแสดง เฉพาะเมื่อสถานะปรากฏบนหน้าคุณสมบัติ และจะไม่ปรากฏขึ้นเมื่อ แสดงอยู่ในรายการ. ถ้าต้องการคืนสภาพการประมวล thread, ให้เลือก Release action สำหรับ thread.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับสถานะ thread อื่น, ให้ดูคำอธิบายทางออนไลน์ iSeries Navigator .

อ็อบเจกต์ที่ถูกล็อก:

งานและ threads ใช้อ็อบเจกต์ในการประมวลผลงาน.

เนื่องจากมีงานมากกว่าหนึ่งชิ้นที่ประมวลผลในหนึ่งครั้ง, จึงมีการล็อกอ็อบเจกต์เพื่อรักษา data integrity ไว้. **อ็อบเจกต์ที่ถูกล็อก** คืออ็อบเจกต์ระบบที่ถูกใช้โดยงานและ thread เพื่อประมวลผลงาน. หลัง จากที่งานหรือ thread รันเสร็จสิ้นแล้ว, อ็อบเจกต์จะถูกปลดล็อกและพร้อมที่จะใช้ ประมวลผลงานเพิ่มเติม. การล็อกอ็อบเจกต์จะอนุญาตให้ผู้ใช้เพียงรายเดียวใช้งานอ็อบเจกต์ในหนึ่งครั้ง, ทั้งนี้ขึ้นอยู่กับประเภทค่าของการล็อกที่ใช้. ตัวอย่างเช่น, ถ้าผู้ใช้สองราย หรือมากกว่าพยายามเปลี่ยนอ็อบ

เจ็ทในเวลาเดียวกัน, การเปลี่ยนแปลงที่อ็อบเจ็กต์โดยผู้ใช้ที่สองจะถูกบล็อก จนกว่าผู้ใช้แรกจะอัปเดตอ็อบเจ็กต์เสร็จสิ้น. ด้วยการใช้ lock holders, ผู้ใช้สามารถดูได้ว่าสิ่งใดมีการล็อก อยู่ในปัจจุบันหรือกำลังรอการล็อกอ็อบเจ็กต์อยู่ในปัจจุบัน.

ขอบเขต ระบุว่าการล็อกเกี่ยวข้องกับงาน, thread, หรือ lock space. ขอบเขตมักจะกำหนดว่าจะมีการล็อกอยู่นานเท่าไรและชนิดคำขอล็อกและกฎความขัดแย้งใดที่อ็อบเจ็กต์มี.

ชนิดคำขอล็อก คือระดับการเข้าถึงอ็อบเจ็กต์ที่จะถูกล็อกของงาน, thread หรือ lock space. ตัวอย่างเช่น, lock exclusive, จะไม่มีการใช้ read lock type ถ้าอ็อบเจ็กต์กำลังถูกเปลี่ยนหรือ ลบบนระบบ. คำขอล็อกชนิดนี้ไม่อนุญาตให้บุคคลใดใช้อ็อบเจ็กต์, และไม่อนุญาตให้บุคคลใดอ่านอ็อบเจ็กต์.

ชนิดคำขอล็อกอื่นๆ คือ:

Exclusive – No read

อ็อบเจ็กต์จะถูกสงวนไว้เพื่อการใช้งานเฉพาะ. อย่างไรก็ตาม, หากอ็อบเจ็กต์ถูกล็อกโดยคำขอล็อกชนิดใดก็ตาม, คุณไม่สามารถได้สิทธิเฉพาะในการใช้งานอ็อบเจ็กต์นั้น. สถานะล็อก นี้มีความเหมาะสม เมื่อผู้ใช้ไม่ต้องการให้ผู้อื่น เข้าใช้งานอ็อบเจ็กต์ จนกว่าฟังก์ชันที่กำลังปฏิบัติการจะเสร็จสมบูรณ์.

Exclusive-Read

สามารถใช้งานอ็อบเจ็กต์ร่วมกับชนิดคำขอล็อกแบบ shared-read. การล็อกนี้เหมาะสมเมื่อผู้ใช้ต้องการป้องกันไม่ให้ผู้ใช้รายอื่นปฏิบัติการดำเนินการใดๆ นอกเหนือจากการอ่าน.

Shared-Update

สามารถใช้งานอ็อบเจ็กต์ร่วมกับชนิดคำขอล็อกแบบ shared-read หรือ shared-update. นั่นคือ, ผู้ใช้รายอื่นสามารถร้องขอสถานะการล็อกแบบ shared-read หรือ shared-update ให้กับอ็อบเจ็กต์เดียวกันได้. สถานะล็อกนี้เหมาะสมเมื่อผู้ใช้ตั้งใจที่จะเปลี่ยนอ็อบเจ็กต์ แต่ต้องการอนุญาตให้ผู้ใช้รายอื่นสามารถอ่านหรือเปลี่ยนอ็อบเจ็กต์เดียวกันได้.

Shared-No update

สามารถใช้งานอ็อบเจ็กต์ร่วมกับชนิดคำขอล็อกแบบ share - no update, และ shared-read เท่านั้น. สถานะล็อกนี้เหมาะสม เมื่อผู้ใช้ไม่ต้องการ เปลี่ยนอ็อบเจ็กต์ แต่ต้องการทำให้มั่นใจว่าไม่มีผู้ใช้รายอื่นเปลี่ยนอ็อบเจ็กต์.

Shared-Read

สามารถใช้งานอ็อบเจ็กต์ร่วมกับชนิดคำขอล็อกทั้งหมดนอกเหนือจากแบบ exclusive - no read. นั่นคือ, ผู้ใช้อื่นสามารถร้องขอชนิดล็อกแบบ exclusive-read, shared-update, shared-read, หรือ shared-no update.

*สถานะการล็อก*เป็นตัวบอกสถานะของคำขอล็อก. สถานะการล็อกอื่นๆ คือ:

พักไว้: คำขอการล็อกถูกดำเนินการเสร็จสิ้นแล้ว และงาน, thread หรือ lock space มีการล็อกอยู่.

กำลังรอ: งานหรือ thread กำลังรอเพื่อรับล็อก.

ร้องขอแล้ว: งานหรือ thread ได้ร้องขอการล็อกแล้ว.

Lock holders คืองาน, thread และ lock space ที่กำลังถือครองล็อกอยู่หรือกำลังรอการล็อกบนอ็อบเจ็กต์ที่ถูกล็อกที่ระบุไว้.

ชนิดของงาน

เซิร์ฟเวอร์ iSeries ประมวลผลงานได้หลายชนิด.

งานแบบ autostart:

งานแบบ autostart คืองานแบ็คกราวด์ที่ทำงานซ้ำ, งาน การกำหนดค่าเริ่มต้นหนึ่งครั้ง que เชื่อมโยงกับระบบย่อย, initializes ฟังก์ชันสำหรับแอปพลิเคชัน, หรือนำเสนอฟังก์ชันการให้บริการศูนย์กลาง สำหรับงานอื่นในระบบย่อยเดียวกัน. งานแบบ autostart ในระบบย่อยการควบคุม สามารถใช้ในการเริ่มระบบย่อยอื่น (เช่นระบบย่อยการควบคุม ที่ได้มาจาก IBM). งานแบบ autostart ที่เชื่อมโยงกับระบบย่อยจะถูกเริ่มต้นโดยอัตโนมัติ ในทุกครั้งที่ระบบย่อยถูกเริ่มขึ้น.

เนื่องจากงานแบบ autostart ทั้งหมดจะถูกเริ่มเมื่อระบบย่อยเริ่มต้นขึ้น, ค่า ที่ระบุสำหรับจำนวนสูงสุดของงานในระบบย่อยไม่ได้ป้องกันการเริ่มต้นของ งานแบบ autostart. แต่ถ้าจำนวนงานในระบบย่อยมากเกินไปเกินจำนวนงานสูงสุด, งานอื่นๆ จะไม่สามารถเริ่มต้นได้. เมื่องานแบบ autostart เสร็จสิ้นลงและมีจำนวนเพียงพอให้จำนวนของงาน ต่ำกว่าจำนวนสูงสุดของ activity level, งานอื่นๆ ในระบบย่อยจึง จะสามารถเริ่มได้.

รายละเอียดของงานที่ใช้สำหรับงานแบบ autostart ถูกระบุโดยใช้คำสั่ง Add Autostart Job Entry (ADDAJE). เมื่อระบบย่อยเริ่มขึ้น, งานจะดำเนินการภายใต้ชื่อโปรไฟล์ผู้ใช้ในรายละเอียดของงาน ที่ระบุ. คุณไม่สามารถระบุรายละเอียดของงานที่มี USER(*RQD). เนื่องจากงานแบบ autostart ดำเนินการภายใต้โปรไฟล์ผู้ใช้ ที่ถูกระบุโดยรายละเอียดของงาน, คุณจึงต้องควบคุมบุคคลที่สามารถ จะเปลี่ยนรายละเอียดของงานได้.

ถ้ามีการระบุงานแบบ autostart มากกว่าหนึ่งงานสำหรับระบบย่อยหนึ่ง, งานแบบ autostart ทั้งหมดจะถูกเริ่มในทันทีแทนที่จะเริ่มทีละงาน. ถ้าจำนวนของงานมาก เกินจำนวนสูงสุดของงานของระบบย่อย, จะไม่มีงานอื่นใดสามารถเริ่มขึ้นได้ใน ระบบย่อยจนกว่าจะมีงานแบบ autostart เสร็จสมบูรณ์มากเพียงพอ เพื่อให้จำนวน ของงานที่รันอยู่ต่ำกว่า activity level สูงสุด.

งานแบ็คกราวด์:

งานแบ็คกราวด์คือกลุ่มของการประมวลผลที่กำหนดไว้ล่วงหน้า ซึ่งถูกส่ง ไปที่ระบบเพื่อปฏิบัติการ โดยมีการโต้ตอบระหว่างผู้ใช้ และระบบน้อยมากหรือ ไม่มีการโต้ตอบเลย. งานที่ไม่ต้องการการโต้ตอบกับผู้ใช้ในการรันสามารถถูก ประมวลผลเป็นงานแบ็คกราวด์. โดยปกติ งานแบ็คกราวด์เป็นงานที่มีระดับความสำคัญต่ำและอาจ ต้องการสภาวะแวดล้อมระบบพิเศษในการรัน.

งานแบ็คกราวด์จะทำงานอยู่ในส่วนแบ็คกราวด์ของระบบ, เพื่อให้ผู้ใช้งานที่ส่งงานเข้ามาสามารถทำงานอื่นได้. งานแบ็คกราวด์สามารถแอ็คทีฟพร้อมกันได้หลายงาน.

รายการต่อไปนี้จะแสดงชนิดต่างๆ ของงานแบ็คกราวด์:

งานแบ็คกราวด์แบบธรรมดา

งานแบ็คกราวด์แบบธรรมดาคืองานที่ถูกส่งไปที่คิวงาน. งานชนิดนี้ จะรออยู่ในแถวพร้อมกับงานแบ็คกราวด์ชนิดอื่น และถูกประมวลผลตามระดับความสำคัญ และหมายเลขลำดับ.

งานแบ็คกราวด์แบบ immediate

งานแบ็คกราวด์แบบ immediate เป็นงานที่ถูกเริ่มขึ้นพร้อมกับแอ็คทีวิตีต่างๆ ของงานตัวแม่. งานจะรันอยู่ในระบบย่อยเดียวกันกับงานแม่. (งานนี้สำเร็จได้โดยใช้ spawn() API.) เนื่องจากงาน จะถือปฎิบัติแอ็คทีวิตีมาจากงานแม่และไม่ผ่านคิวงาน, งานนี้จึงสามารถเริ่มได้รวดเร็วกว่างานที่ส่งไปที่คิวงาน.

งานแบ็คกราวด์แบบ MRT

งานแบบแบ็คกราวด์ MRT คืองานแบบ multiple requester terminal (MRT). งานแบบ MRT เป็นงานของ S/36 Environment ที่ทำงานเหมือนเป็นเซิร์ฟเวอร์, ที่จะอนุญาตให้งาน S/36 Environment ตัวอื่นสามารถผนวกติดมาด้วยเพื่อที่จะรันโปรแกรม MRT ด้วยกัน.

งานแบ็คซ์แบบพิมพ์

งานแบ็คซ์แบบพิมพ์จะติดตามดูไฟล์พรินเตอร์เอาต์พุต (หรือเรียกอีกอย่างว่าสพูลไฟล์) ที่ถูกสร้างขึ้นโดยงานที่มีโปรไฟล์ผู้ใช้งานปัจจุบันแตกต่างไปจากโปรไฟล์ผู้ใช้ในตอนทำงานเริ่มต้น.

งานแบ็คซ์สามารถเริ่มได้เมื่อผู้ใช้:

- ทำให้งานอยู่ในคิวงาน
- ออกคำขอการเริ่มโปรแกรมการสื่อสาร
- เริ่มระบบย่อยพร้อมด้วยงานแบบ prestart
- ใช้ spawn() API

วิธีการเริ่มงานแบ็คซ์:

เมื่อผู้ใช้ส่งงานแบ็คซ์, งานจะรวบรวมข้อมูลจากอ็อบเจกต์ระบบต่างๆ ก่อนที่งานจะถูกวางบนคิวงาน.

1. ผู้ใช้ส่งงาน.
2. งานจะค้นหาแอ็ททริบิวต์งาน. ถ้าไม่พบแอ็ททริบิวต์งานบนคำสั่ง Submit Job (SBMJOB), งานจะค้นหาใน รายละเอียดของงาน (ที่ระบุบนคำสั่ง SBJOB), โปรไฟล์ผู้ใช้ของผู้ใช้ปัจจุบัน, และงานที่แอ็คทีฟในปัจจุบัน (งานที่ออกคำสั่ง SBJOB).

หมายเหตุ: ในลักษณะคล้ายกันกับการเริ่มงานแบบโต้ตอบ, คุณสามารถระบุในรายละเอียดของงานให้ใช้โปรไฟล์ผู้ใช้. โปรไฟล์ผู้ใช้สามารถระบุให้ใช้ค่ากำหนดของระบบในการค้นหาแอ็ททริบิวต์งาน.

3. หลังจากที่มีแอ็ททริบิวต์ครบทั้งหมดแล้ว, งานจะอยู่บนคิวงาน.
4. เมื่อระบบย่อยพร้อมที่จะจัดการงาน, ระบบย่อยจะค้นหางานใน คิวงาน (คิวงานที่ระบบย่อยได้จัดสรร).
5. จากนั้น, เช่นเดียวกับขั้นตอนของงานแบบโต้ตอบ, ระบบย่อยจะตรวจสอบ รายละเอียดของงานสำหรับข้อมูลการเราต์.
6. ระบบย่อยใช้ข้อมูลการเราต์ในการค้นหา routing entry. routing entry นำเสนอข้อมูลพูลที่งานจะใช้, โปรแกรมการเราต์ที่จะใช้, และคลาสที่งานจะเรียกแอ็ททริบิวต์รันไทม์ของงานมา.
7. หลังจากได้ข้อมูลนี้แล้ว, โปรแกรมการเราต์จะรัน. ถ้าคุณใช้ QCMD, QCMD จะใช้คำสั่ง SBJOB. คำสั่งจะรันคำสั่งที่ระบุบนพารามิเตอร์ CMD หรือ RQSDTA.

การ Spawn งานแบ็คซ์:

Spawn คือฟังก์ชันที่สร้างกระบวนการงานใหม่ (child process) ที่ได้รับแอ็ททริบิวต์หลายอย่างของกระบวนการเรียก (parent process). โปรแกรมใหม่จะถูกระบุและเริ่มการรันใน child process. เมื่อคุณ spawn งานแบ็คซ์ นั่นคือคุณกำลังใช้ parent job เพื่อส่งผ่าน อากิวเมนต์และตัวแปรสถานะแวดล้อมไปที่ child job. spawn() API ใช้งานแบ็คซ์แบบ immediate, งานแบบ prestart, หรืองานแบ็คซ์แบบ prestart.

งานสื่อสาร:

งานสื่อสารคืองานแบ็คซ์ที่ถูกเริ่มขึ้นโดยคำขอการเริ่มโปรแกรมจากระบบรีโมต. การประมวลผลงานเกี่ยวข้องกับคำขอการสื่อสาร และข้อกำหนดคุณลักษณะที่เหมาะสม.

สำหรับงานแบ็คการสื่อสารที่จะรันบนระบบ i5/OS , subsystem description ที่มี work entry สำหรับงานสื่อสารต้องมีอยู่บนระบบ. communications work entry ระบุให้ระบบย่อยทราบถึงซอร์สสำหรับงานสื่อสารที่ระบบจะประมวลผล. การประมวลผลงานเริ่มขึ้นเมื่อระบบย่อยได้รับ คำขอการเริ่มโปรแกรมการสื่อสารจากระบบรีโมตและพบ routing entry ที่เหมาะสมสำหรับคำขอ.

ข้อมูลการเรอต์สำหรับงานสื่อสาร

การเรอต์งาน ของงานสื่อสารถูกกำหนดโดยคำขอการเริ่มโปรแกรมที่ได้รับ จากระบบรีโมต. เมื่อคำขอการเริ่มโปรแกรมถูกประมวลผล บนระบบเป้าหมาย, data stream ความยาวคงที่ที่ถูกใช้เป็นข้อมูลการเรอต์ จะถูกสร้างขึ้น. ตำแหน่ง 25 ของข้อมูลการเรอต์จะมี PGMEVOKE สำหรับ คำขอการสื่อสารเสมอ. Subsystem routing entries ที่ระบุค่าเปรียบเทียบ เป็น PGMEVOKE ในตำแหน่ง 29 โดยปกติจะมี *RTGDTA เป็นชื่อโปรแกรม. นี่หมายความว่าชื่อโปรแกรมที่ได้รับระบุในข้อมูลการเรอต์ (จากคำขอการเริ่ม โปรแกรมของระบบรีโมต) คือโปรแกรมที่จะรัน.

หากต้องใช้สภาวะแวดล้อม การประมวลผลพิเศษสำหรับงานสื่อสารบางงาน, คุณสามารถใส่เพิ่ม routing entry เพิ่มเติมที่ subsystem description, โดยระบุ ค่าเปรียบเทียบที่มีตำแหน่งเริ่มต้นเป็น 37. ค่าเปรียบเทียบนี้ควรมีชื่อโปรแกรมสำหรับคำขอการเริ่มโปรแกรม. routing entry ต้องมี หมายเลขลำดับต่ำกว่า routing entry ที่ใช้ PGMEVOKE เป็นค่าเปรียบเทียบ . วิธีนี้ช่วยให้งานสื่อสารบางงานสามารถรันด้วยข้อกำหนดคุณลักษณะ คลาสหรือพูลที่แตกต่างได้.

การรักษาความปลอดภัย

การรักษาความปลอดภัยบนระบบ i5/OS ควบคุมบุคคลที่สามารถจะใช้อุปกรณ์สื่อสาร และบุคคลที่สามารถเข้าใช้งานคำสั่งด้วย device descriptions ที่เกี่ยวข้อง. คุณควรพิจารณามาตรการการรักษาความปลอดภัยเพิ่มเติมเมื่อ บันทึกลงและรันแอพลิเคชันโปรแกรมบนทั้งระบบรีโมตและระบบเป้าหมาย.

รายละเอียดของงานสำหรับงานสื่อสาร

รายละเอียดของงาน ที่ใช้สำหรับงานสื่อสารถูกระบุบนคำสั่ง Add Communications Entry (ADDCMNE). ผู้ใช้ที่ระบุบนรายละเอียดของงานนี้จะถูกข้ามไป. ระบบเรียกชื่อผู้ใช้สำหรับงานสื่อสารมาจาก คำขอการเริ่มโปรแกรม. หากคำขอการเริ่มโปรแกรมไม่ได้ระบุชื่อผู้ใช้, ระบบจะใช้ค่าผู้ใช้ที่ฟอลต์จาก communications entry. เพื่อให้มั่นใจว่าระบบมีการรักษาความปลอดภัยที่แน่นหนามากขึ้น, ให้รวมข้อมูลผู้ใช้ในคำขอการเริ่ม โปรแกรม แทนที่จะระบุผู้ใช้ที่ฟอลต์ใน communications work entry.

ชนิดของงานสื่อสาร:

หัวข้อนี้อธิบายชนิดทั่วไปของงานสื่อสาร.

Qlus (logical unit services)

Qlus จะจัดการเหตุการณ์ที่ดูแลอุปกรณ์โลจิคัลยูนิต, ซึ่งรู้จักกันว่า อุปกรณ์สื่อสาร. Qlus ยังรับผิดชอบการจัดสรรอุปกรณ์ให้กับระบบย่อย สื่อสารที่ถูกต้องอีกด้วย.

Qcmnarbxx (communications arbiters)

communications arbiters พร้อมกับ Qsysarb (system arbiter) และ Qtaparb (tape arbiter) ทำหน้าที่ประมวลผลงานสำหรับอุปกรณ์ทุกชนิด, ไม่ใช่เฉพาะอุปกรณ์ การสื่อสารเท่านั้น. งานเหล่านี้รวมถึงการเชื่อมต่อระบบสื่อสาร, การตัดการเชื่อมต่อ, การล็อกอุปกรณ์, และการประมวลวิธีแก้ไขข้อผิดพลาด.

คำกำหนดของระบบ communication arbiter jobs, at restart (QCMNARB) เป็นตัวกำหนดจำนวนของงาน communications arbiter ที่จะถูกเริ่มต้น. communications arbiter อย่างน้อย 3 ตัวจะถูกเรียกใช้งานบนระบบ โพรเซสเซอร์เดี่ยว.

Qsyscomm1 (system communications)

งานตัวนี้จะจัดการการสื่อสารบางตัวและ activity อินพุต/เอาต์พุต (I/O).

Q400filsvr (remote file system communication)

งานตัวนี้จะทำการสื่อสารแบบ common programming interface ทั่วไป (APPN หรือ APPC) สำหรับระบบรีโมตไฟล์.

งานโต้ตอบ:

งานแบบโต้ตอบคืองานที่เริ่มขึ้นเมื่อผู้ใช้ signs on ที่จอภาพ และสิ้นสุดลงเมื่อผู้ใช้ signs off. สำหรับงานที่จะรัน, ระบบย่อยจะค้นหารายละเอียดของงาน, ซึ่งสามารถระบุใน workstation entry หรือโปรไฟล์ผู้ใช้.

งานแบบโต้ตอบต้องการการสื่อสารสองทางอย่างต่อเนื่องระหว่างผู้ใช้ และ iSeries เพื่อ ปฏิบัติภารกิจ. งานโต้ตอบจะเริ่มต้นเมื่อผู้ใช้ sign on เข้าสู่ระบบ. ระบบจะร้องขอข้อมูลการ signon. ถ้าระบบยอมรับคำขอ signon, ระบบจะสร้างงานแบบโต้ตอบ. หลังจากนั้นระบบจะขอให้ผู้ใช้ส่งคำขอ. หลังจากนั้นผู้ใช้ต้องใส่ คำขอ, และระบบจะตอบรับโดยการประมวลผลคำขอนั้น. รูปแบบนี้จะทำซ้ำจนกว่าผู้ใช้จะจบงาน แบบโต้ตอบโดยการ signing off ระบบ, หรืองานจบเนื่องจาก application exception หรือการแก้ไขข้อผิดพลาดอุปกรณ์.

หากงานโต้ตอบเป็นส่วนหนึ่งของกลุ่มงาน หรือคู่ของงาน, งานโต้ตอบจะมีประเภทของงานอย่างใดอย่างหนึ่งต่อไปนี้:

Interactive – Group

งาน Interactive – Group เป็นส่วนของกลุ่มของงานที่เกี่ยวข้องกับ อุปกรณ์แสดงผลเดี่ยว.

Interactive – System request

งาน Interactive – System request เป็นหนึ่งในคู่ของงานซึ่งเกี่ยวข้อง ซึ่งกันและกันโดยฟังก์ชันคำขอระบบ.

คุณทราบหรือไม่? คุณสามารถ sign on ที่ระบบได้สองวิธี. คุณสามารถเข้าสู่ ระบบด้วยตนเองโดยใช้ user id และรหัสผ่าน. และคุณยังสามารถสร้างโปรแกรม เพื่อส่ง user id และรหัสผ่านไปที่เซิร์ฟเวอร์โดยอัตโนมัติ, โดยข้ามจอภาพ signon.

วิธีการเริ่มงานแบบโต้ตอบ:

เมื่อผู้ใช้ signs on ที่ระบบ, ระบบย่อยจะรวบรวมข้อมูลจาก อ็อบเจกต์ระบบต่างๆ ก่อนที่งานแบบโต้ตอบจะพร้อมใช้งาน.

1. ระบบย่อยจะค้นหารายละเอียดของงานใน workstation entry เพื่อ เรียกใช้แอ็ททริบิวต์สำหรับงานแบบโต้ตอบ. ถ้า workstation entry ระบุ *USRPRF สำหรับรายละเอียดของงาน, งานจะใช้ข้อมูล จากโปรไฟล์ผู้ใช้.

หมายเหตุ: ความยืดหยุ่นนี้ช่วยให้คุณระบุได้ว่า แอ็ททริบิวต์ของงาน จะอยู่ที่เวิร์กสแตชันหรืออยู่ที่ผู้ใช้แต่ละราย.

2. หลังจากที่ระบบย่อยทราบถึงรายละเอียดของงานที่จะใช้แล้ว, ระบบย่อยอาจจะไม่ ค้นหาแอ็ททริบิวต์งานทั้งหมดในรายละเอียดของงาน. บางแอ็ททริบิวต์อาจอยู่ใน โปรไฟล์ผู้ใช้. ถ้าโปรไฟล์ผู้ใช้ไม่มีข้อมูล, ระบบย่อย จะดูที่คำกำหนดของระบบ.

หมายเหตุ: โปรไฟล์ผู้ใช้มีแอ็ททริบิวต์งาน ที่ช่วยให้คุณสามารถปรับแต่งบางสิ่งเฉพาะสำหรับผู้ใช้รายนั้น.

3. หลังจากที่ระบบย่อยรวบรวมแอ็ดทริบิวต์งานทั้งหมดแล้ว, ระบบย่อยจะกำหนดว่า สามารถจะเริ่มงานแบบโต้ตอบใหม่ได้หรือไม่ หรือควรจะแสดงข้อความแสดงความผิดพลาดบนจอภาพ signon หรือไม่. ระบบย่อยจะตรวจสอบว่ามีการใช้งานถึงจำนวนสูงสุดของงานที่อนุญาตโดยระบบย่อยหรือโดย workstation entry แล้วหรือไม่. จากนั้น ระบบย่อยจะตรวจสอบว่ามีการป้อนชื่อโปรไฟล์ผู้ใช้ที่ถูกต้อง, ชื่อโปรไฟล์ผู้ใช้เป็น โปรไฟล์ผู้ใช้ที่ใช้ได้, และตรวจสอบว่ารหัสผ่านที่ใช้ (ถ้าต้องการ) ถูกต้องหรือไม่. ถัดไป, จะตรวจสอบว่าผู้ใช้มีสิทธิที่ถูกต้อง ที่รายละเอียดของงาน, subsystem description, workstation device description, และเอาต์พุตคิวและไลบรารีหรือไม่. ท้ายสุด, ระบบย่อยจะตรวจสอบว่า ผู้ใช้ได้ใช้งานถึงขีดจำกัดสำหรับ signons ที่อนุญาตสำหรับโปรไฟล์ใช้นั้นหรือไม่. ถ้าพบข้อผิดพลาดการตรวจสอบใดๆ, จอภาพ signon จะแสดงขึ้น พร้อมกับข้อความที่เหมาะสม. มิฉะนั้น, กระบวนการของการเริ่มงานแบบโต้ตอบ จะดำเนินต่อไป.
4. หลังจากที่ระบบย่อยตรวจสอบว่างานแบบโต้ตอบสามารถเริ่มได้แล้ว, ระบบย่อย จะตรวจสอบรายละเอียดของงานสำหรับข้อมูลการเรอต์. ระบบย่อยใช้ข้อมูลการเรอต์ในการค้นหา routing entry ใน subsystem description. routing entry นำเสนอข้อมูลพูลที่งานจะใช้, โปรแกรมการเรอต์ ที่จะใช้, และคลาสที่งานจะเรียกแอ็ดทริบิวต์รันใหม่ของงานมา.
5. เมื่อได้ข้อมูลเหล่านี้ทั้งหมดแล้ว, โปรแกรมการเรอต์จะรัน. IBM จัดเตรียม โปรแกรมการเรอต์ที่มีชื่อว่า QCMD, ซึ่งคุณสามารถใช้สำหรับงานได้ทุกชนิด. QCMD ทราบว่างานเป็นงานแบบโต้ตอบหรือไม่ และตรวจสอบโปรไฟล์ผู้ใช้สำหรับ initial program ที่จะรัน. ถ้า initial program รันเสร็จสิ้นแล้ว, QCMD จะแสดง เมนูเริ่มต้น.

การตัดการเชื่อมต่องานแบบโต้ตอบ:

เมื่อใช้คำสั่ง Disconnect Job (DSCJOB), งานจะถูกตัดการเชื่อมต่อและจอแสดงผล signon แสดงขึ้นอีกครั้งหนึ่ง. ถ้าต้องการเชื่อมต่อกับงานอีกครั้ง, ให้ sign on ที่อุปกรณ์เดิมซึ่งคุณตัดการเชื่อมต่อ. งานแบบโต้ตอบอื่นอาจถูกเริ่มบนอุปกรณ์ภายใต้ชื่อผู้ใช้ที่แตกต่างกัน.

- อีอ็อปชันบนเมนู System Request ช่วยให้คุณสามารถตัดการเชื่อมต่องานแบบโต้ตอบ, ซึ่งทำให้จอแสดงผล signon ปรากฏขึ้น. อีอ็อปชันจะเรียกคำสั่ง DSCJOB .
- เมื่อเชื่อมต่อกับงานอีกครั้ง, ค่าที่ระบุบนจอแสดงผล signon สำหรับโปรแกรม, เมนู, และไลบรารีปัจจุบันจะถูกข้ามไป.
- งานซึ่งมีฟังก์ชัน PC organizer หรือ PC text assist แอ็ดทริบิวต์ที่อยู่ไม่สามารถตัด การเชื่อมต่อได้.
- งาน TCP/IP TELNET สามารถตัดการเชื่อมต่อได้เฉพาะถ้าเซสชันกำลังใช้ device description ที่ผู้ใช้ระบุชื่อเท่านั้น. คุณ สามารถสร้าง device description ที่ผู้ใช้ ระบุชื่อโดยใช้วิธีใดวิธีหนึ่งดังนี้:
 - การใช้ Network Stations พร้อมกับพารามิเตอร์ DISPLAY NAME
 - การใช้ iSeries Access PC 5250 Client Access support พร้อมกับฟังก์ชัน workstation ID
 - การใช้ TCP/IP TELNET Device Initialization exit point ในการระบุ ชื่อเวิร์กสเตชัน
 - Telnet client (STRTCPTELN) พร้อมกับพารามิเตอร์อุปกรณ์รีโมต

หมายเหตุ: ชื่ออุปกรณ์ที่ระบุระบุ, เช่น QPADEV*, ไม่อนุญาตให้ตัดการเชื่อมต่องาน เนื่องจากเป็นไปไม่ได้ที่ผู้ใช้ราย เดียวกันจะ sign on ที่อุปกรณ์เดียวกัน.

- งานทั้งหมดจะถูกตัดการเชื่อมต่อสำหรับงานกลุ่ม. เมื่องานถูกเชื่อมต่ออีกครั้ง, คุณจะกลับไปไปที่ที่มีการตัดการเชื่อมต่อ. ถ้างานกลุ่มหลังสุดที่ แอ็ดทริบิวต์ลงก่อนที่คุณจะเชื่อมต่ออีกครั้ง, คุณจะกลับไปทำงานกลุ่มถัดไป.
- ถ้าไม่สามารถตัดการเชื่อมต่องานเนื่องจากเหตุผลใดๆ, งานจะถูกจบแทน.
- งานที่ตัดการเชื่อมต่อทั้งหมดในระบบย่อยจะจบลง เมื่อระบบย่อยสิ้นสุดลง. ถ้า ระบบย่อยกำลังสิ้นสุด, จะไม่สามารถใช้คำสั่ง DSCJOB ในงานใดๆ ในระบบย่อย.
- สามารถใช้ค่ากำหนดของระบบ Disconnect Job Interval (QDSCJOBITV) เพื่อระบุช่วงเวลาที่สามารถตัดการเชื่อมต่องานได้. ถ้าถึงช่วงเวลานั้น, งานที่ตัดการเชื่อมต่อจะจบลง.

- งานที่ตัดการเชื่อมต่อที่ไม่เกินกว่าค่า QDSCJOBITV จะจบลง เมื่อระบบย่อยจบลง หรือเมื่อเกิดการ IPL.

ข้อผิดพลาด I/O สำหรับอุปกรณ์ที่ร้องของาน:

อุปกรณ์ที่ร้องขอคือ เวิร์กสเตชันซึ่งผู้ใช้สามารถล็อกออนที่โตเมนและใช้เน็ตเวิร์กทีชอร์ส. แอ็ททริบิวต์งาน Device Recovery Action (DEVRCYACN) ระบุ action ที่จะใช้เมื่อเกิดข้อผิดพลาด I/O สำหรับอุปกรณ์ที่ร้องขอของงาน.

แอ็ททริบิวต์ DEVRCYACN มีอ็อปชันดังต่อไปนี้:

***SYSVAL**

นี่เป็นค่าดีฟอลต์. ค่านี้จะชี้ไปที่ Action ที่จะใช้เมื่อเกิดข้อผิดพลาดอุปกรณ์ บนค่ากำหนดของระบบ workstation (QDEVRCYACN). ค่ากำหนดของระบบจะสนับสนุน ค่าทั้งหมดที่แอ็ททริบิวต์งานสนับสนุน (ยกเว้น *SYSVAL).

***MSG** ส่งสัญญาณข้อความแสดงความผิดพลาด I/O และอนุญาตให้แอ็พพลิเคชัน โปรแกรมทำการแก้ไขข้อผิดพลาด. นี่ไม่ใช่การตั้งค่าที่แนะนำ.

***DSCMSG**

ตัดการเชื่อมต่องาน. นี่เป็นค่าดีฟอลต์ที่จัดส่งมาให้. ในทันทีที่เชื่อมต่ออีกครั้ง, ข้อความแสดงความผิดพลาดใหม่จะส่งสัญญาณให้แอ็พพลิเคชันโปรแกรมของผู้ใช้ทราบว่าอุปกรณ์หายไปแล้วและถูกกู้คืนนับตั้งแต่ I/O, และต้องแสดงเนื้อหาของจอแสดงผลอีกครั้ง.

***DSCENDRQS**

ตัดการเชื่อมต่องาน. ในทันทีที่เชื่อมต่ออีกครั้ง, จะมีการปฏิบัติฟังก์ชันคำขอการจบ เพื่อให้การควบคุมของงานกลับไปเป็นระดับคำขอหลังสุด.

***ENDJOB**

จบงาน. อาจมีการจัดทำบันทึกการใช้งานสำหรับงาน. ข้อความจะถูกส่งไปที่บันทึกการใช้งานและไปที่ไฟล์บันทึก QHST โดยระบุว่างานจบลงเนื่องจากข้อผิดพลาดอุปกรณ์.

***ENDJOBNO LIST**

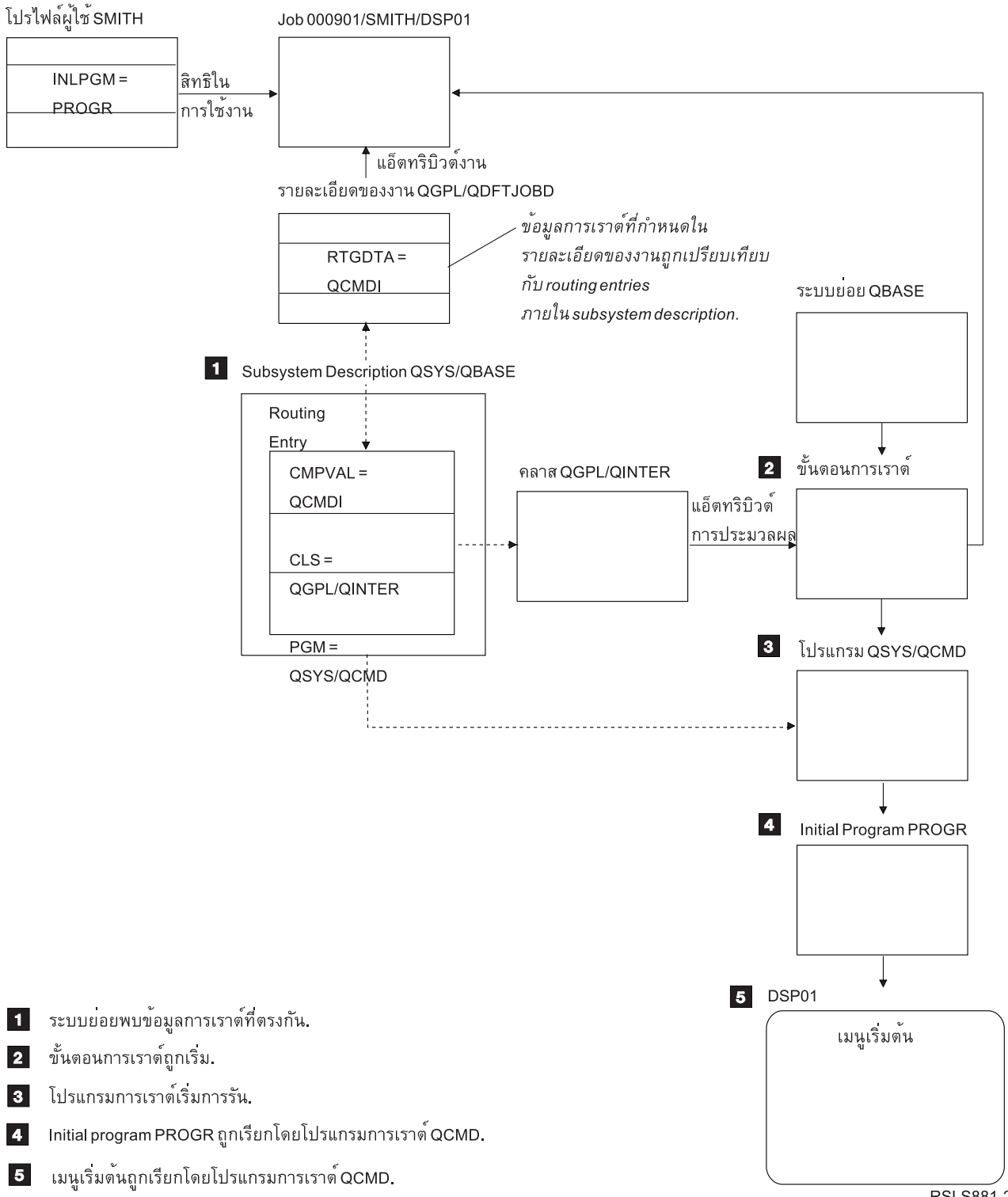
จบงาน. ไม่มีการจัดทำบันทึกการใช้งาน. ข้อความจะถูกส่งไปที่ไฟล์บันทึก QHST เพื่อระบุว่างานจบลงเนื่องจากข้อผิดพลาดอุปกรณ์.

หมายเหตุ: ถ้ามีการระบุ *DSCENDRQS, *ENDJOB, หรือ *ENDJOBNO LIST สำหรับ DEVRCYACN, การกู้คืนจะเกิดขึ้นเมื่อเกิดข้อผิดพลาดบนอุปกรณ์. ถ้ามีการระบุค่าอื่น, การกู้คืนจะเกิดขึ้นเมื่ออุปกรณ์เกิดข้อผิดพลาด I/O ถัดไป.

งานแบบโต้ตอบและขั้นตอนการเราต์:

ก่อนการเรียกเมนูเริ่มต้น ข้อมูลการเราต์จะถูกเปรียบเทียบกับ routing entries ใน subsystem description. เมื่อจับคู่ได้, โปรแกรมที่ระบุใน routing entry จะถูกเรียกและขั้นตอนการเราต์ถูก เริ่มต้น.

ข้อมูลต่อไปนี้แสดง activity ก่อนหน้าที่จะนำไปสู่การเริ่มขั้นตอนการเราต์ และการแสดงเมนูเริ่มต้นสำหรับโปรไฟล์ผู้ใช้ที่ระบุ initial program.



- 1 ระบบย่อยพบข้อมูลการเรดที่ตรงกัน.
- 2 ขั้นตอนการเรดถูกเริ่ม.
- 3 โปรแกรมการเรดเริ่มการรัน.
- 4 Initial program PROGR ถูกเรียกโดยโปรแกรมการเรด QCMD.
- 5 เมนูเริ่มต้นถูกเรียกโดยโปรแกรมการเรด QCMD.

รูปที่ 1. Subsystem Activity

Interactive Job Approaches

คุณสามารถจัดการงานแบบโต้ตอบได้หลายวิธี. วิธีเหล่านี้ขึ้นอยู่กับว่าคุณควบคุมขั้นตอนการเรดอย่างไร. ในอันดับแรกคุณควรกำหนดสิ่งดังต่อไปนี้:

- โปรแกรมใดจะควบคุมขั้นตอนการเรด: QSYS/QCMD หรือโปรแกรมผู้ใช้?
- การเรดจะขึ้นอยู่กับผู้ใช้หรือขึ้นอยู่กับเวิร์กสเตชัน?

โปรแกรมที่ควบคุมขั้นตอนการเรด:

ถ้าต้องการกำหนดแนวทางที่ดีที่สุดสำหรับงานหนึ่ง, อันดับแรก คุณต้อง กำหนดโปรแกรมที่ควรใช้ในการควบคุมขั้นตอนการเรด.

การใช้ QSYS/CMD สำหรับงานแบบโต้ตอบ - ประโยชน์

ตัวประมวลผลคำสั่ง ที่ได้มาจาก IBM QSYS/QCMD มีความยืดหยุ่นสูงสุดในแง่ของการทำให้ฟังก์ชัน พร้อมใช้งานสำหรับผู้ใช้เวิร์กสเตชัน. การใช้ QCMD เพื่อควบคุม ขั้นตอนการเรดมีประโยชน์ดังต่อไปนี้:

- attention program จะถูกเรียกทำงานถ้าโปรแกรมถูกระบุในโปรไฟล์ผู้ใช้.
- initial program ที่ระบุในโปรไฟล์ผู้ใช้จะถูกเรียก.
- เมนูเริ่มต้นที่ระบุในโปรไฟล์ผู้ใช้จะถูกเรียก.
- ผู้ใช้จะดูวางในสภาวะแวดล้อม System/36 เมื่อมีการระบุ ในโปรไฟล์ผู้ใช้.

ยิ่งไปกว่านั้น, คำดีฟอลต์ที่ใช้ QCMD จะนำคุณไปที่ Main Menu ที่ซึ่ง คุณสามารถป้อนคำสั่งได้โดยตรง, รวมถึงคำสั่ง CALL, ซึ่งถูกใช้ในการเรียก ฟังก์ชันที่ผู้ใช้บันทึก. มีการนำเสนอเมนูอธิบายทางออนไลน์ เพื่อให้เข้าใช้งานฟังก์ชันระบบได้ง่าย. และมีการนำเสนอเมนูการเลือกคำสั่ง, การเข้าใช้งานการค้นหาที่ค่อนข้างรวดเร็ว, และฟังก์ชัน command entry (เรียกโดย CALL QCMD). ฟังก์ชัน command entry มีวัตถุประสงค์หลักเพื่อให้ใช้โดย โปรแกรมเมอร์และผู้ควบคุมเครื่องที่ต้องใช้ฟังก์ชันหลายอย่างผ่านทางการใช้คำสั่ง โดยตรง.

การเรียกโปรแกรมผู้ใช้โดยตรงสำหรับงานแบบโต้ตอบ - ประโยชน์

โปรแกรม ของคุณสามารถถูกเรียกโดยตรงเพื่อควบคุมขั้นตอนการเรดสำหรับงานแบบโต้ตอบ . สามารถออกแบบโปรแกรมเหล่านี้เพื่อให้สามารถเข้าใช้งานฟังก์ชันในรูปแบบ พิเศษตามที่ผู้ใช้เวิร์กสเตชันของคุณต้องการได้มากกว่าการเข้าใช้งานที่โปรแกรมที่ได้มาจาก IBM. ยิ่งไปกว่านั้น, เนื่องจากโปรแกรมของคุณถูกปรับแต่งสำหรับฟังก์ชันเฉพาะ, โดยปกติแล้ว โปรแกรมจึงควรจะต้องการรีซอร์สของระบบในการสนับสนุนการรัน น้อยกว่าที่โปรแกรมที่ได้มาจาก IBM ต้องการ. และคุณยังอาจต้องการนำเสนอฟังก์ชัน เช่น initial program และเมนูเริ่มต้นด้วย.

การเรดตามเวิร์กสเตชันและตามผู้ใช้:

หลังจากที่คุณกำหนดโปรแกรมที่จะใช้ในการควบคุมขั้นตอนการเรดแล้ว, คุณต้องกำหนดว่าจะกำหนดการเรดตามเวิร์กสเตชันซึ่ง งานถูกเริ่มต้น, หรือตามผู้ใช้ (โปรไฟล์ผู้ใช้) ที่ signed on.

การเรดตามเวิร์กสเตชันสามารถทำได้โดยใช้ข้อมูลการเรดที่ระบุในรายละเอียดของงานที่เชื่อมโยงกับ workstation entry หรือ โปรไฟล์สำหรับอุปกรณ์. การเรดตามผู้ใช้สามารถทำได้โดยการใช้ initial program ที่ระบุในโปรไฟล์ผู้ใช้หรือรายละเอียดของงานในโปรไฟล์ผู้ใช้ ที่แม้กับ routing entry ไม่ใช่ QCMD.

การใช้ initial program

Initial programs อาจโต้ตอบกับ เวิร์กสเตชันเพื่อเรียกค่าอินพุตจากผู้ใช้เวิร์กสเตชัน. เมื่อเรียก initial program, โปรแกรมไม่สามารถได้รับค่าพารามิเตอร์. สามารถใช้ initial program ในหนึ่งจากสองวิธีดังนี้:

- เพื่อสร้างสภาวะแวดล้อมแรกเริ่มสำหรับให้ผู้ใช้ป้อนคำสั่ง. ตัวอย่างเช่น , สามารถเปลี่ยนรายชื่อไลบรารีหรือสามารถแทนที่ค่าเดิมของ ไฟล์พื้มพ์และไฟล์ข้อความ. เมื่อ initial program ทำฟังก์ชันของโปรแกรมเสร็จสมบูรณ์ และกลับไป QSYS/QCMD, เมนูเริ่มต้นจะถูกแสดงขึ้น.
- เป็นโปรแกรมควบคุมสำหรับงาน. ถ้า initial program ไม่ได้กลับไป QSYS/QCMD, โปรแกรมจะกลายเป็นโปรแกรมควบคุมสำหรับขั้นตอนการเรอต์. เมนูเริ่มต้นจะไม่แสดงขึ้น. ผู้ใช้สามารถร้องขอได้เฉพาะฟังก์ชันที่ พร้อมใช้งานผ่าน initial program เท่านั้น.
ตัวอย่างเช่น, สามารถแสดงเมนู พร้อมด้วยแอ็พพลิเคชันอ็อพชันเฉพาะ. ผู้ใช้สามารถปฏิบัติได้เฉพาะฟังก์ชันบน เมนูเท่านั้น. ตัวอย่างหนึ่งของอ็อพชันนั้นคือ sign off. ถ้ารันคำสั่ง SIGNOFF, งานจะจบและไม่มีการแสดง Main Menu. ถ้าคุณใช้วิธีนี้, ให้พิจารณาการใช้อ็อพชันโปรไฟล์ผู้ใช้ INLMNU เพื่อให้มั่นใจว่าไม่มีการแสดงเมนูใด.

initial program สามารถถูกบันทึกได้เพื่อให้เมื่อส่งคืน, โปรแกรม จะหรือจะไม่กลับไป QSYS/QCMD อย่างไรก็ดีอย่างหนึ่ง. ถ้า initial program กลับไปที่ QSYS/QCMD, เมนูเริ่มต้นจะถูกแสดงขึ้น.

เมื่องานจบลงพร้อมกัน:

ในบางครั้ง, งานหลายงานอาจจบลงในเวลาเดียวกัน. ตัวอย่างเช่น, เมื่อเกิดข้อผิดพลาด เน็ตเวิร์ก และแอ็พพลิเคชันถูกเซ็ทเป็น *ENDJOB หรือ *ENDJOBNO LIST. นอกเหนือจาก การจบของงานแล้ว, อาจเกิดการแก้ไขอุปกรณ์ดังต่อไปนี้.

- ระดับความสำคัญของงานถูกลดลง. สิ่งนี้จะเกิดขึ้นเมื่องานไม่ได้มีระดับความสำคัญเหมือน กับงานแบบโต้ตอบที่แอ็คทีฟอื่นอีกต่อไป.
- การแบ่งเวลาของงานถูกเซ็ทเป็น 100 มิลลิวินาที. สิ่งนี้เกิดขึ้นเพื่อให้งานที่มีระดับความสำคัญ สูงกว่ามีโอกาสที่ดีกว่าในการใช้รีซอร์สในการประมวลผล.

บันทึกการใช้งานสำหรับงานที่มีแอ็พพลิเคชันถูกเซ็ทเป็น *ENDJOB หรือ *ENDJOBNO LIST อยู่ใน บันทึกการใช้งานที่คงค้าง. เพื่อสร้างพรินเตอร์เอาต์พุตจากบันทึกการใช้งานที่อยู่ใน บันทึกการใช้งานคงค้าง, ให้ใช้คำสั่ง Display Job Log (DSPJOBLOG).

เมื่องานจบ คุณสามารถควบคุมวิธีการบันทึกบันทึกการใช้งานไปที่สพูลไฟล์ได้. สิ่งนี้สามารถทำโดยตัวงานเองในขณะที่กำลังจบ, ทำโดยงานเซิร์ฟเวอร์ แบ็กกราวนด์, หรือไม่ทำเลย. ค่าที่คุณระบุส่งผลกระทบต่อเวลา การแก้ไขโดยรวม ถ้างานหลายงานจบลงพร้อมกัน. สำหรับข้อมูลเพิ่มเติม, ให้ดูหลักการ บันทึกการใช้งานที่คงค้าง ที่เกี่ยวข้อง.

งานแบบ prestart:

งานแบบ prestart คืองานแบ็คชั้ที่เริ่มต้นก่อนที่จะได้รับค่าของงาน. งานแบบ prestart ถูก เริ่มต้นก่อนงานชนิดอื่นๆ ในระบบย่อย. งานแบบ prestart แตกต่างจากงานอื่นๆ เนื่องจาก การใช้ prestart job entries (ส่วนของ subsystem description) ในการกำหนด โปรแกรม, คลาส, และพูลของหน่วยความจำที่จะใช้เมื่องานถูกเริ่มต้น.

ภายใน prestart job entry, คุณต้องระบุแอ็พพลิเคชันที่ระบบย่อย จะใช้ในการสร้างและจัดการพูลของงานแบบ prestart. ใช้งานแบบ prestart เพื่อลดจำนวนเวลา ที่ต้องใช้ในการจัดการคำขอของงาน. งานแบบ prestart มีอยู่ 2 ชนิด. แต่ละชนิดจะจัดการคำขอต่างชนิดกัน. ก่อนที่งานจะรอสำหรับคำขอแรกของงานนั้น, งานจะถูกแสดงว่าเป็น "Prestart" เท่านั้น, เพราะวาระบบจะยังไม่รู้ว่า คำขอชนิดใดที่งานนั้นจะจัดการ.

Prestart communications

งานคืองานแบ็คชั้การสื่อสารที่เริ่มต้นก่อนที่ระบบรีโมต จะส่งคำขอการเริ่มโปรแกรม.

Prestart batch

งานคืองานแบตช์ที่เริ่มต้นก่อนที่จะได้รับคำขอร้องงาน.

งานแบบ prestart จะเริ่มขึ้นก่อนที่จะได้รับคำขอร้องการทำงาน, ไม่ว่าจะ เป็น ตอนที่ระบบย่อยเริ่มขึ้น หรือที่เป็นผลมาจากคำสั่ง Start Prestart Jobs (STRPJ) อย่างใดอย่างหนึ่ง. งานแบบ prestart จะเริ่มทำงานจาก prestart job entry (PJE) ใน subsystem description. prestart job entry ระบุแอตทริบิวต์ต่างๆ เช่น โปรแกรมที่จะรันใน งานแบบ prestart, โปรไฟล์ผู้ใช้ที่งานแบบ prestart จะเริ่มต้นรันภายใต้มัน, รายละเอียดของงาน, คลาสที่ใช้ในการระบุแอตทริบิวต์รันใหม่ของงาน, และพูลหน่วยความจำซึ่งงานแบบ prestart จะรันในนั้น.

งานแบบ prestart สามารถเริ่มทำงานและ initialize ตัวมันเองได้ก่อนที่จะได้รับคำขอร้องการทำงาน. สิ่งนี้จะช่วยลดจำนวนเวลาที่ต้องใช้ในการจัดการคำขอต่างๆ. งานแบบ prestart ช่วยให้สามารถ initialize และจัดการกับหลายคำขอได้พร้อมกัน เพื่อให้ไม่ต้องใช้งานใหม่สำหรับทุกคำขอ. หลาย client server applications ใช้งานแบบ prestart ในการจัดการคำขอของผู้ใช้ไคลเอ็นต์. การมีงานที่พร้อมจะทำงานได้จะช่วยให้การทำงานมีประสิทธิภาพมากขึ้นในกรณีนี้ เพราะงานแบบ prestart จะเริ่มการประมวลผลคำขอให้กับผู้ใช้ได้ทันที.

หมายเหตุ: ค่าที่ระบุสำหรับจำนวนสูงสุดของงานในระบบย่อย อาจทำให้งานแบบ prestart ไม่สามารถเริ่มขึ้นได้. หากจำนวนงานสูงสุด ในระบบย่อยมีจำนวนมากเกิน, งานแบบ prestart จะไม่สามารถเริ่มทำงานได้. แต่เมื่อมีจำนวนหนึ่งได้เสร็จสิ้นลงจนทำให้จำนวนงานที่รันอยู่ ต่ำกว่าจำนวนงานสูงสุดในระบบย่อยแล้ว, งานแบบ prestart ในระบบย่อย จึงจะสามารถทำงานได้.

Program Start Requests

Program Start Request (PSR) เป็นวิธีทางสถาปัตยกรรมสำหรับ SNA clients ในการเชื่อมต่อกับ เซิร์ฟเวอร์ SNA. เมื่อมีการตั้ง คำงานแบบ prestart เพื่อจัดการกับ PSRs, สถานะภายนอกของงานจะเป็น PSRW (Program Start Request Wait).

นอกจากนี้ยังสามารถใช้งานแบบ prestart สำหรับเซิร์ฟเวอร์ TCP/IP ที่ได้มาจาก IBM, และโฮสต์เซิร์ฟเวอร์ส่วนใหญ่. งานแบบ prestart เหล่านี้รับงานผ่านทางอินเตอร์เฟซภายในและไม่ได้ใช้ PSRs. อย่างไรก็ตาม, งานแบบ prestart ที่กำลังรอกงานอยู่, แม้ว่าจะไม่ได้ใช้ PSRs, ยังคงแสดงสถานะเป็น PSRW.

ชื่องานแบบ prestart:

ชื่อสามส่วนที่ถูกต้องทั้งหมดของงานแบบ prestart ไม่เคยเปลี่ยนแปลง หลังจากที่งานแบบ prestart เริ่มขึ้น. ส่วนชื่อผู้ใช้ของชื่อ งานสามส่วนที่ถูกต้องทั้งหมดคือ โปรไฟล์ผู้ใช้ซึ่งงานแบบ prestart ถูกเริ่มต้นภายใต้มันเสมอ .

ถ้าสพูลไฟล์ถูกเปิดก่อนที่งานแบบ prestart จะจัดการคำขอร้องงานใดๆ , สพูลไฟล์จะถูกเชื่อมโยงกับโปรไฟล์ผู้ใช้ของ prestart job entry. มิฉะนั้น, สพูลไฟล์จะถูกเชื่อมโยงกับโปรไฟล์ผู้ใช้ปัจจุบันของงาน.

ถ้าโปรไฟล์ prestart job entry และโปรไฟล์ผู้ใช้ปัจจุบันแตกต่างกัน, สพูลไฟล์จะถูกสพูลภายใต้งานที่มีชื่องานเป็น QPRTJOB และมี ชื่อผู้ใช้เป็นโปรไฟล์ผู้ใช้ปัจจุบัน. (และใช้กับ prestart job entries สำหรับงานเซิร์ฟเวอร์ด้วย.)

พารามิเตอร์คลาส (CLS) บน prestart job entry นำเสนอวิธีการควบคุมลักษณะประสิทธิภาพของสองคลาสของงานแบบ prestart ต่อ prestart job entry.

วิธีการทำงานของงานแบบ prestart:

งานแบบ prestart คืองานที่ถูกเริ่มก่อนที่งานจะมาถึง. การเริ่มก่อนนี้ช่วยให้ระบบสามารถจัดการคำขอสำหรับงานได้โดยไม่ต้องเสียเวลากับ การเริ่มต้นงานใหม่.

งานแบบ prestart เป็นงานแบ็คชชนิดหนึ่ง. นี่หมายความว่างานมีชนิดของงานเป็น 'B' และชนิดย่อยของงานเป็น 'J'. ชนิดงานที่พัฒนา กำหนดงานเป็นงานแบบ prestart (1610), งานแบ็คชแบบ prestart (1620), หรืองานสื่อสารแบบ prestart (1630). ชนิดงานที่พัฒนาอธิบายวิธีการที่งานแบบ prestart ยอมรับคำขอของงาน. ถ้าโปรแกรมที่รันอยู่ในงานแบบ prestart ใช้อินเตอร์เฟซการสื่อสารสำหรับการยอมรับงาน, งานจะเป็นงานสื่อสารแบบ prestart. ถ้าโปรแกรมที่รันอยู่ในงานแบบ prestart ยอมรับงานผ่านอินเตอร์เฟซงานแบ็คช, งานจะเป็นงานแบ็คชแบบ prestart. ถ้าโปรแกรมยังไม่ถึงจุดของการยอมรับงาน, งานจะเป็นเพียงงานแบบ prestart. งานแบ็คชแบบ Prestart มักถูกอ้างอิงเป็นงานเซิร์ฟเวอร์ เนื่องจากงานนำเสนอการให้บริการ สำหรับคำขอของงาน.

คำขอของงานการสื่อสารถูกจัดการโดยระบบย่อยที่มีการจัดสรร อุปกรณ์การสื่อสารที่จำเป็น. โดยปกติคำขอของงานแบ็คช ถูกจัดการโดยระบบย่อยระดับต้นระบบใดระบบหนึ่งที่ถูกจัดส่งมาพร้อมกับระบบ: QSYSWRK, QUSRWRK, หรือ QSERVER.

งานแบบ prestart ถูกเริ่มตามข้อมูลที่มีอยู่ใน prestart job entries. พารามิเตอร์ Start jobs (STRJOBS) สามารถระบุให้งานแบบ prestart ถูกเริ่มขึ้นเมื่อระบบย่อยถูกเริ่มต้นหรือเมื่อ ป้อนคำสั่ง Start Prestart Jobs (STRPJ). พารามิเตอร์ Initial number of jobs (INLJOBS) กำหนด จำนวนของงานแบบ prestart ที่เริ่มต้นในครั้งแรกสำหรับโปรแกรม.

เมื่อคำขอของงานมาถึง, อาจต้องการงานแบบ prestart มากขึ้น. พารามิเตอร์ Threshold (THRESHOLD) บอกให้ทราบว่าจะเริ่มงานมากขึ้นเมื่อใด. เมื่อจำนวนของงานแบบ prestart ที่พร้อมใช้สำหรับการจัดการคำขอลดลงต่ำกว่า ค่าที่ระบุโดยพารามิเตอร์ THRESHOLD, งานเพิ่มเติม จะถูกเริ่มขึ้น. พารามิเตอร์ Additional number of jobs (ADLJOBS) บอกให้ทราบว่าจะเริ่มงานเพิ่มขึ้นจำนวนเท่าใด.

คุณต้องตรวจให้แน่ใจว่ามีงานเพียงพอที่จะจัดการกับคำขอของงาน โดยไม่มีความล่าช้าที่ไม่จำเป็น. คุณสามารถใช้ข้อมูลจากคำสั่ง Display Active Prestart Jobs (DSPACTPJ) เพื่อดูว่ามีคำขอของงาน ใดๆ, ที่อ้างอิงเป็น "คำขอการเริ่มโปรแกรม" ในข้อมูล DSPACTPJ, ต้องรอให้งานพร้อมใช้งานหรือไม่. เพิ่มค่าของพารามิเตอร์ THRESHOLD จนกว่า DSPACTPJ จะแสดงว่าไม่มีคำขอของงานใดต้องรอ.

งานแบบ prestart บางงานจัดการคำขอของงาน แล้วจึงพร้อมที่จะจัดการ คำขอของงานอื่น. พารามิเตอร์ Maximum number of uses (MAXUSE) ช่วยให้คุณสามารถระบุจำนวนของคำขอของงานที่งานแบบ prestart เหล่านี้ จัดการได้. งานแบบ prestart บางงานจัดการคำขอของงานเพียงคำขอเดียว แล้วจึงจบ, โดยข้ามค่า MAXUSE. งานแบบ prestart จะจัดการคำขอของงานหลายคำขอ หรือจัดการคำขอของงานเพียงคำขอเดียวขึ้นอยู่กับโปรแกรม ที่รันอยู่ในงานแบบ prestart.

เมื่องานแบบ prestart จบลงหลังจากที่จัดการคำขอของงานอย่างน้อยหนึ่งคำขอ, ระบบย่อยจะเปรียบเทียบจำนวนของงานที่ยังคงรันอยู่กับจำนวน ที่ระบุในพารามิเตอร์ INLJOBS. ถ้าจำนวนของงาน ที่เหลืออยู่น้อยกว่า INLJOBS, ระบบย่อยจะเริ่ม งานอื่น.

ถ้างานแบบ prestart จบลงโดยไม่ได้จัดการคำขอของงานอย่างน้อยหนึ่งคำขอ และ งานไม่จบลงโดยคำสั่ง End Job (ENDJOB), ถือว่า โปรแกรมงานแบบ prestart มีข้อผิดพลาด. ระบบย่อยจบ prestart job entry ในลักษณะที่ควบคุม. สิ่งนี้ช่วยให้งานที่ใช้ในคำขอของงานสามารถทำคำขอนั้นให้เสร็จสิ้นได้, แต่ป้องกันไม่ให้ระบบย่อยเริ่ม งานเพิ่มเติม.

ระบบย่อยจะตรวจสอบจำนวนของงานแบบ prestart เป็นครั้งคราวเพื่อกำหนดว่า มีงานแบบ prestart ที่พร้อมใช้มากเกินไปหรือไม่. งานแบบ prestart จะพร้อมใช้งาน เมื่องานกำลังรอสำหรับคำขอของงาน. ถ้าจำนวนของงานแบบ prestart ที่พร้อมใช้ มากกว่าพารามิเตอร์ THRESHOLD และ จำนวนทั้งหมดของงานแบบ prestart มากกว่าพารามิเตอร์ INLJOBS, ระบบย่อยจะจบงานแบบ prestart ที่พร้อมใช้งานบางงาน. สำหรับงานแบบ prestart ที่จัดการคำขอของงานมากกว่าหนึ่งคำขอ, คุณควรใช้ข้อมูล

DSPACTPJ เพื่อกำหนดโหนดสูงสุด, ใส่เพิ่มจำนวนนี้ที่ค่าของพารามิเตอร์ THRESHOLD , และเซตพารามิเตอร์ INLJOBS เป็นผลรวม. การทำเช่นนี้ป้องกันไม่ไห้ระบบย่อย เข้าใจว่างานที่พร้อมใช้เป็นงานส่วนเกิน ยกเว้นว่าเวิร์กโหนดจะมากกว่า โหนดสูงสุดที่คาดไว้.

Prestart job entries:

คุณกำหนดงานแบบ prestart โดยใช้ prestart job entry. prestart job entry ไม่มีผลต่อการจัดสรรอุปกรณ์หรือการกำหนดค่าขอ การเริ่มโปรแกรม.

แอตทริบิวต์งานของงานแบบ prestart จะไม่ถูกเปลี่ยนแปลงโดยระบบย่อย เมื่อ คำขอการเริ่มโปรแกรมแนบกับงานแบบ prestart. อย่างไรก็ตาม, โดยทั่วไป งานเซิร์ฟเวอร์ จะเปลี่ยนแอตทริบิวต์งานเป็นแอตทริบิวต์ของโปรไฟล์ผู้ใช้ที่สลับค่า.

คำสั่ง Change Prestart Job (CHGPGJ) อนุญาตให้ งานแบบ prestart เปลี่ยนแอตทริบิวต์งานบางอย่างเป็นแอตทริบิวต์ของ รายละเอียดของงาน (ที่ระบุในรายละเอียดของงานที่เชื่อมโยงกับโปรไฟล์ผู้ใช้ของคำขอการเริ่มโปรแกรม หรือในรายละเอียดของ งานที่ระบุใน prestart job entry).

การรักษาความปลอดภัยและงานแบบ prestart:

เมื่องานแบบ prestart เริ่มต้น, งานจะรันภายใต้โปรไฟล์ผู้ใช้ของ งานแบบ prestart. เมื่อคำขอการเริ่มโปรแกรมแนบกับงาน แบบ prestart, โปรไฟล์ผู้ใช้ของงานแบบ prestart จะถูกแทนที่โดยโปรไฟล์ผู้ใช้ของคำขอการเริ่มโปรแกรม. เมื่องานแบบ prestart จัดการกับคำขอการเริ่มโปรแกรมเสร็จสิ้นแล้ว, โปรไฟล์ผู้ใช้ของ คำขอการเริ่มโปรแกรมจะถูกแทนที่ด้วยโปรไฟล์ผู้ใช้ ของงานแบบ prestart. ถ้า มีโปรไฟล์กลุ่มที่เชื่อมโยงกับโปรไฟล์ผู้ใช้, โปรไฟล์กลุ่ม จะถูกเปลี่ยนด้วย.

การแลกเปลี่ยนโปรไฟล์ผู้ใช้ใช้สำหรับการตรวจสอบสิทธิเท่านั้น. ไม่มีการแลกเปลี่ยน แอตทริบิวต์อื่นที่เชื่อมโยงกับโปรไฟล์ผู้ ใช้. สำหรับไลบรารีต่างๆ บนรายชื่อไลบรารีซึ่งโปรไฟล์ผู้ใช้ของ prestart job entry ได้รับสิทธิในการใช้งานแบบ prestart ยังคง ได้รับสิทธิในไลบรารีนั้นต่อไปเมื่อโปรไฟล์ผู้ใช้ของคำขอ การเริ่มโปรแกรมแทนที่โปรไฟล์ผู้ใช้ของ prestart job entry. อย่างไรก็ตาม, รายชื่อไลบรารี อาจถูกเปลี่ยนแปลงโดยคำสั่ง Change Library List (CHGLIBL).

การให้สิทธิอ็อบเจกต์งานแบบ prestart

เมื่องานแบบ prestart เริ่มขึ้น, จะมีการตรวจสอบสิทธิในโปรไฟล์ผู้ใช้ของ prestart job entry บนทุกอ็อบเจกต์ที่จำเป็นสำหรับการ เริ่มงาน. ก่อนที่คำขอการเริ่ม โปรแกรมจะได้รับอนุญาตให้แนบงานแบบ prestart, จะมีการตรวจสอบเฉพาะโปรไฟล์ผู้ใช้ /รหัสผ่านของคำขอการเริ่มโปรแกรมและสิทธิในการใช้งานอุปกรณ์สื่อสารและ ไลบรารี/โปรแกรมเท่านั้น.

เพื่อหลีกเลี่ยงกรณีที่โปรไฟล์ผู้ใช้ของคำขอการเริ่ม โปรแกรมไม่ได้รับสิทธิในอ็อบเจกต์ที่โปรไฟล์ผู้ใช้ของ prestart job entry ได้รับสิทธิ, คุณต้องตรวจสอบให้แน่ใจว่าโปรไฟล์ผู้ใช้ของ คำขอการเริ่มโปรแกรมได้รับสิทธิในอ็อบเจกต์เท่ากับสิทธิของโปร ไฟล์ผู้ใช้ของ prestart job entry เป็นอย่างน้อย. เพื่อให้เป็นเช่นนั้น, สามารถสร้างโปรแกรมงานแบบ prestart โดยผู้ใช้ prestart job entry ที่มี USRPRF(*OWNER) ระบุอยู่บน คำสั่ง CRTxxxPGM (โดยที่ xxx คือภาษาโปรแกรม). สิทธิเจ้าของโปรแกรม จะถูก ถ่ายโอนโดยอัตโนมัติไปที่โปรแกรมใดๆ ที่ถูกเรียกโดย โปรแกรมงานแบบ prestart. มิฉะนั้น, คุณอาจเลือกที่จะตรวจสอบการ ให้สิทธิอ็อบเจกต์โดยตรง (CHKOBJ) ก่อนการอ้างอิงถึงอ็อบเจกต์ใดๆ .

ไฟล์และอ็อบเจกต์ ที่โปรไฟล์ผู้ใช้ของงานแบบ prestart ไม่ได้รับสิทธิควรจะถูกปิดและ จัดสรรคืน ก่อนทำการจบ transaction บนอุปกรณ์requestor . ถ้าไฟล์ฐานข้อมูลถูกเปิดทิ้งไว้ในงานแบบ prestart, เพื่อให้มั่นใจถึง ความปลอดภัยของฐานข้อมูล, โปรแกรมงานแบบ prestart ต้องตรวจสอบสิทธิในการ ใช้ไฟล์เปิดของโปรไฟล์ผู้ใช้ของคำขอการเริ่มโปรแกรม.

คำแนะนำเกี่ยวกับประสิทธิภาพสำหรับงานแบบ prestart:

งานแบบ prestart ควรทำงานให้มากที่สุดเท่าที่จะเป็นไปได้ ก่อนที่งานจะ พยายามจัดหาอุปกรณ์โปรแกรม ICF หรือยอมรับ CPI Communications conversation. จำนวนงานที่ทำก่อนยิ่งมาก (การจัดสรรอ็อบเจกต์, การเปิดไฟล์ฐานข้อมูล, และอื่นๆ), สิ่งที่ต้องทำเมื่อได้รับคำขอการเริ่มโปรแกรมจะยิ่งน้อย, ดังนั้นจึงตอบสนอง transaction ได้เร็วขึ้น. ข้อมูลต่อไปนี้เป็นข้อควรพิจารณาเกี่ยวกับ ประสิทธิภาพเพิ่มเติมเมื่อใช้งานแบบ prestart:

เตือนความจำ: ถ้า active prestart job entry อยู่ในระบบย่อย, ระบบย่อยจะตรวจสอบเป็นครั้งคราวเพื่อให้ทราบจำนวนของงานแบบ prestart ในพูลที่พร้อมให้บริการคำขอการเริ่มโปรแกรม เพื่อกำหนดว่ามีงานแบบ prestart ที่พร้อมใช้งานมากเกินไปหรือไม่. งานแบบ prestart ที่พร้อมใช้งานมากเกินจะถูกจบลงโดย ระบบย่อยที่ละน้อย. อย่างไรก็ตาม, ระบบย่อยจะปล่อยให้ม้งานอย่างน้อยที่สุดตามจำนวน ของงานแบบ prestart ที่ระบุในแอตทริบิวต์ INLJOBS ในพูล.

- คุณควรจัดสรรคืนรีซอร์สเฉพาะที่ transaction ที่คุณ ต้องการดำเนินการเท่านั้น. รีซอร์สใดๆ ที่ใช้ทั่วไปสำหรับ transaction อื่นที่ดำเนินการ โดยโปรแกรมงานแบบ prestart ควรยังคงได้รับการจัดสรร ในขณะที่งาน กำลังรอคำขอถัดไป. คุณควรเปิดไฟล์ไว้และให้อ็อบเจกต์ถูกจัดสรร เพื่อประหยัดเวลาเมื่อได้รับคำขอถัดไป.

หมายเหตุ: โดยทั่วไป ไฟล์ฐานข้อมูลที่ปล่อยให้เปิดไว้ใน งานแบบ prestart ต้องคำนึงถึงข้อควรพิจารณาเดียวกันกับของไฟล์ฐานข้อมูลที่ถูกแบ่งใช้ในงานเดียวกัน.

- เนื่องจากบางไลบรารี QTEMP ถูกใช้สำหรับตลอดช่วงชีวิตของงานแบบ prestart , ดังนั้นจึงควรลบอ็อบเจกต์ที่ไม่ต้องการอีกต่อไปออก.
- เนื่องจาก Local Data Area (LDA) เดียวกันถูกใช้สำหรับตลอดช่วงชีวิตของงาน แบบ prestart, ดังนั้นจึงสามารถเก็บและส่งผ่านข้อมูลไปยัง transaction ถัดไปได้.
- เนื่องจากงานแบบ prestart แต่ละงานสามารถจัดการคำขอการเริ่มโปรแกรมได้ หลายคำขอ, และมีบันทึกการใช้งานเพียงบันทึกเดียว, ดังนั้นคุณจึงอาจต้องการให้ แอ็พพลิเคชันของคุณส่งข้อความไปที่บันทึกการใช้งานเพื่อระบุ activity ของงานแบบ prestart. สิ่งนี้ยังมีประโยชน์เนื่องจากบันทึกการใช้งานของงานแบบ prestart จะถูกลบระหว่างการใช้.
- แอ็ตทริบิวต์งานของงานแบบ prestart จะไม่ถูกเปลี่ยนแปลงโดยระบบย่อย เมื่อ คำขอการเริ่มโปรแกรมแนบกับงานแบบ prestart. คำสั่ง change Prestart Job (CHGPI) อนุญาตให้ งานแบบ prestart เปลี่ยนแอ็ตทริบิวต์งานบางอย่างเป็นแอ็ตทริบิวต์ของ รายละเอียดของงาน (ที่ระบุในรายละเอียดของงานที่เชื่อมโยงกับโปรไฟล์ผู้ใช้ ของคำขอการเริ่มโปรแกรม หรือในรายละเอียดของงานที่ระบุใน prestart job entry).
- ถ้าระบบของคุณใช้ job accounting, โปรแกรมงานแบบ prestart ควรจะรันคำสั่ง CHGPI ด้วยคำคำขอการเริ่มโปรแกรมสำหรับพารามิเตอร์ accounting code (CHGPI ACGCDE(*PGMSTRRQS)) โดยทันที หลังจากที่คำขอการเริ่มโปรแกรมแนบกับงานแบบ prestart. action นี้เปลี่ยน accounting code เป็นค่าที่ระบุในโปรไฟล์ผู้ใช้ที่เชื่อมโยงกับคำขอการเริ่มโปรแกรม. ในทันทีก่อนที่โปรแกรมจะ เสร็จสิ้นการจัดการคำขอการเริ่มโปรแกรม, โปรแกรมควรจะรันคำสั่ง CHGPI ด้วย คำ Prestart Job Entry สำหรับพารามิเตอร์ accounting code (CHGPI ACGCDE(*PJE)). สิ่งนี้จะเปลี่ยน accounting code กลับเป็นค่าที่ระบุในรายละเอียดของงาน ของ prestart job entry.
- พารามิเตอร์คลาส (CLS) บน prestart job entry นำเสนอวิธีการควบคุม ลักษณะประสิทธิภาพของสองคลาสของงานแบบ prestart ต่อ prestart job entry. ตัวอย่างเช่น, คุณสามารถใช้ระดับความสำคัญการดำเนินการที่ ต่ำกว่าสำหรับงานที่ได้รับเมื่อระบบอยู่อยู่แล้ว.

สพูลไฟล์และ prestart job entry:

ถ้าสพูลไฟล์ถูกเปิดก่อนที่งานแบบ prestart จะจัดการคำขอการเริ่มโปรแกรมใดๆ , สพูลไฟล์จะถูกเชื่อมโยงกับโปรไฟล์ผู้ใช้ของ prestart job entry ; มิฉะนั้นสพูลไฟล์จะถูกเชื่อมโยงกับโปรไฟล์ผู้ใช้ของคำขอการเริ่มโปรแกรม ปัจจุบัน.

ถ้าโปรไฟล์ prestart job entry และโปรไฟล์ผู้ใช้ของคำขอการเริ่มโปรแกรม ปัจจุบันแตกต่างกัน, สพูลไฟล์จะถูกสพูลภายใต้งานที่มีส่วนแรกของชื่องานสามส่วน เป็น QPRTJOB และส่วนที่สองเป็น ชื่อของโปรไฟล์ผู้ใช้.

งาน reader และงาน writer:

งานแบบ reader คืองานอินพุตที่สพูล, และงานแบบ writer คืองานเอาต์พุตที่สพูล.

Reader งานแบบ reader จะอ่าน streams งานแบ็ตซ์จากไฟล์ฐานข้อมูล, และวางงาน บนคิวงาน. งาน reader เป็นส่วนหนึ่ง ของการสพูลอินพุต และเป็นโปรแกรมที่ได้มาจาก IBM.

Writer งาน writer จะเขียนเร็กคอร์ดจาก printer output files (หรือเรียกว่าสพูลไฟล์)ไปยังพรินเตอร์. งานแบบ writer คือโปรแกรมที่ได้มาจาก IBM, ที่เริ่มต้นใน ระบบย่อยการสพูลที่ซึ่งงานจะเลือกไฟล์จากเอาต์พุตคิวที่จะพิมพ์.

งานเซิร์ฟเวอร์:

งานเซิร์ฟเวอร์คืองานที่รันอย่างต่อเนื่องในแบ็กกราวนด์บนระบบ iSeries .

งานอาจมาจากฟังก์ชันเน็ตเวิร์ก, ฟังก์ชันระบบปฏิบัติการ, ในนามของผู้ใช้, ระบบอื่นภายในเน็ตเวิร์ก, หรือจากเซอวิสเซระบบทั่วไป, เช่นงานเซิร์ฟเวอร์การรวมกลุ่มเพื่อทำงานเสมือนระบบเดียว. โดยปกติ งานเซิร์ฟเวอร์จะรันในหนึ่งในสามระบบย่อยระดับต้น ที่ให้มาพร้อมกับระบบ - QSYSWRK, QSERVER, หรือ QUSRWRK. งานเซิร์ฟเวอร์มีความเชื่อมโยงมากที่สุดกับฟังก์ชันเช่น HTTP, Lotus Notes , และ TCP/IP. ระบบ iSeries มีต้นแบบจำลองพื้นฐานสามแบบสำหรับงาน เซิร์ฟเวอร์:

Threaded Job Model

ในแบบจำลองงาน threaded งานเซิร์ฟเวอร์คืองานที่มีหลาย threads. thread หนึ่งตัวทำหน้าที่เป็นผู้กระจายงานให้กับ thread อื่นๆ . ยกตัวอย่าง, เมื่อเซิร์ฟเวอร์รับคำขอของไคลเอ็นต์, initial thread จะอ่านคำขอและส่งผ่านไปยังอีก thread หนึ่งเพื่อดำเนินการตามคำขอให้เสร็จสมบูรณ์. ด้วยแบบจำลองนี้, จำนวนงานบนระบบจะลดจำนวนลงอย่างมากเพราะงานถูกจัดการใน thread ต่างๆ แทนที่จะต้องใช้งานหลายงาน. บางตัวอย่างของงานเซิร์ฟเวอร์ ที่ใช้แบบจำลองงาน threaded คือ Domino®, เซิร์ฟเวอร์ HTTP, และ WebSphere®.

Prestart Job Model

ในแบบจำลองงานแบบ prestart โดยปกติจะมีงานหลักที่ทำหน้าที่เป็น listener สำหรับคำขอที่เข้ามาในระบบ. โดยปกติ งานนี้เรียกว่างาน daemon. งาน daemon จะจัดการคำขอแรกเริ่ม แล้วส่งผ่านคำขอ ไปที่งานเซิร์ฟเวอร์แบบ prestart ที่เหมาะสม. ด้วยแบบจำลองงานนี้, การใช้งานแบบ prestart สามารถช่วยลดจำนวนของงานที่ต้องใช้ เนื่องจากหลังจากที่ คำขอเสร็จสมบูรณ์แล้ว งานเซิร์ฟเวอร์แบบ prestart จะคอยรับคำขอถัดไป. งานเซิร์ฟเวอร์จะถูกนำมาใช้ใหม่. นอกจากนี้, จากมุมมองประสิทธิภาพการทำงาน, งานแบบ prestart จะทำการรันและรอเพื่อประมวลผลคำขออยู่แล้ว. บางตัวอย่างของงานเซิร์ฟเวอร์ ที่ใช้แบบจำลองงานแบบ prestart คือ เซิร์ฟเวอร์ SQL, โฮสต์เซิร์ฟเวอร์, และ Simple Mail Transfer Protocol (SMTP).

หมายเหตุ: สำหรับงานที่รันโค้ดผู้ใช้, โดยปกติงานจะไม่ถูกนำกลับมาใช้ใหม่ (เหมือนกับงานเซิร์ฟเวอร์ส่วนใหญ่). ที่เป็นเช่นนี้เนื่องจากโค้ดผู้ใช้ อาจเปลี่ยนแปลงใดๆ ในงาน (เช่น เซิร์ฟเวอร์คำสั่งรีโมต).

Multiple Listening Job Model

ในแบบจำลองงาน multiple listening, มีการเริ่มงานเซิร์ฟเวอร์หลายงาน. เมื่อคำขอเข้ามาในระบบ, งานที่ได้รับคำขอ

จะจัดการค่าของงาน, ในขณะที่งานเซิร์ฟเวอร์ที่พร้อมใช้งานถัดไปจะคอยค่าขอใหม่ที่จะเข้ามา. เมื่องานเซิร์ฟเวอร์ดำเนินการตามค่าขอเสร็จสมบูรณ์, งานจะหยุดการเชื่อมต่อและจบลง. งานเซิร์ฟเวอร์ใหม่จะเริ่มต้นและดำเนินการในลักษณะวัฏจักรนี้ต่อไป.

ด้วยแบบจำลองนี้, คุณไม่ต้องกังวลกับ prestart job entries. อย่างไรก็ตาม, ในบางครั้งอาจไม่สามารถตั้งค่าระบบย่อยเฉพาะสำหรับสภาพแวดล้อมของคุณ เพราะแบบจำลองนี้จะรันที่ระบบย่อยดีฟอลต์. exception หนึ่งคือ File Transfer Protocol (FTP). ด้วย file transfer protocol คุณสามารถตั้งค่าระบบย่อยที่รันเซิร์ฟเวอร์ file transfer protocol ได้. ไม่สามารถมีงาน FTP บางงานรันในระบบย่อยหนึ่งและงานที่เหลือรันในระบบย่อยอื่น. นอกจากนี้, จากมุมมองประสิทธิภาพการทำงาน, ไม่สามารถหลีกเลี่ยงต้นทุนของการเริ่มต้นทำงานและการสิ้นสุดการทำงานได้เพราะเมื่องานถูกรันและสิ้นสุดการทำงาน งานอื่นจะเริ่มต้นงาน. อย่างไรก็ตาม, เนื่องมาจากงานจะสิ้นสุดการทำงานเมื่อการเชื่อมต่อเสร็จสมบูรณ์ และงานถัดไปได้เริ่มขึ้น, โดยทั่วไปงานใหม่จะถูกเรียกขึ้นมาและทำการรันเมื่อได้รับค่าขอถัดไป, ดังนั้นต้นทุนการเริ่มต้นใช้งานและการสิ้นสุดการทำงานไม่ควรมีผลต่อเวลาที่ใช้เชื่อมต่อกับเซิร์ฟเวอร์.

บางตัวอย่างของงานเซิร์ฟเวอร์ที่ใช้แบบจำลองงาน multiple listening คือ FTP และ line printer daemon (LPD).

สำหรับข้อมูลรายละเอียดเกี่ยวกับชื่อของงานเซิร์ฟเวอร์ที่รันบนระบบ, ให้ดูตารางงานเซิร์ฟเวอร์. ตารางนี้แสดงระบบย่อยและชื่องาน เพื่อให้คุณสามารถค้นหางานแอ็คทีฟและบันทึกการใช้งานของงานได้. ตารางจะแสดง job description ที่งานเซิร์ฟเวอร์แต่ละตัวใช้ด้วย. ตามค่าดีฟอลต์ งานเซิร์ฟเวอร์ส่วนใหญ่จะไม่สร้างบันทึกการทำงานเมื่องานสิ้นสุดการทำงาน (พารามิเตอร์ LOG ถูกตั้งค่าเป็น 4 0 *NOLIST), ซึ่งหมายความว่าไม่มีการสร้างบันทึกการทำงาน. ถ้าคุณต้องการให้มีการสร้างบันทึกการทำงานที่มีข้อความทั้งหมดที่ส่งไปที่ บันทึกการใช้งาน, พารามิเตอร์ LOG ต้องระบุ 4 0 *SECLVL.

งานระบบ:

งานระบบจะถูกสร้างขึ้นโดยระบบปฏิบัติการ เพื่อควบคุมรีซอร์สของระบบและปฏิบัติฟังก์ชันของระบบ. งานระบบจะรันเมื่อเซิร์ฟเวอร์ iSeries เริ่มขึ้นหรือเมื่อติสก์พูลอิสระ varied on. งานเหล่านี้ปฏิบัติการกิจหลายอย่าง ตั้งแต่การเริ่มต้นระบบปฏิบัติการ, การเริ่มต้นและการจบ ระบบย่อย, ไปจนถึงการจัดตารางเวลางาน.

งานการเริ่มทำงาน:

งานการเริ่มทำงาน คืองานระบบที่รันที่ IPL. งานนี้จัดการภารกิจการจัดเตรียมสภาวะแวดล้อมระบบปฏิบัติการและ ทำให้พร้อมใช้งาน. ข้างล่างนี้เป็นรายการของงานการเริ่ม ระบบต่างๆ.

Scpf (start control program function)

งานตัวนี้เป็นงานศูนย์กลางเมื่อคุณเริ่มงานระบบ. Scpf เริ่มชุด Qsysarb , แต่ Qsysarb3 เริ่มงานระบบอื่นส่วนใหญ่ (ไม่ใช่ Qlus) และนำ ระบบไปที่สภาพสามารถใช้ได้. งานตัวนี้จะยังคงแอ็คทีฟอยู่หลังจากระบบถูกเรียกใช้งาน, เพื่อจัดเตรียมสภาวะแวดล้อมสำหรับการรันฟังก์ชันที่มีความสำคัญต่ำและมีแนวโน้มจะรันเป็นเวลานาน. Scpf จะรันอยู่ในระหว่างกระบวนการ ปิดเครื่อง (Pwrdownsys), และเป็นงานตัวที่ใช้จบกระบวนการ ของเครื่องด้วย.

Qwcbtclnup (job table cleanup)

งานตัวนี้จะถูกใช้ระหว่างการเริ่มต้นระบบเพื่อยืนยันให้แน่ใจว่าโครงสร้างงานพร้อมใช้งานได้. โดยปกติงานจะประมวลผลเสร็จสมบูรณ์ก่อน การจบของการเริ่มระบบ, แต่งานสามารถรันต่อไปได้หลังจากที่ระบบเริ่มต้น, ถ้ามีโครงสร้างงานที่จะล้างอยู่มาก. งานระบบตัวนี้จะจบลงเมื่อเสร็จสิ้นการประมวลผล.

Qlpsvr (software agreements acceptance)

งานนี้จะเริ่มทำงานโดยอัตโนมัติระหว่างช่วง IPL หลังจากการตอบรับข้อตกลงการใช้ซอฟต์แวร์ทางออนไลน์. งานจะจบเมื่อข้อตกลงทั้งหมดถูกยอมรับหรือปฏิเสธ อย่างไรก็ตามอย่างหนึ่ง.

System arbiter:

system arbiters (QSYSARB และ QSYSARB2 ถึง QSYSARB5), ที่เริ่มต้น โดยงานระบบ SCPF, นำเสนอสถานะแวดล้อม สำหรับการรันฟังก์ชันที่มี ระดับความสำคัญสูง. system arbiter อนุญาตให้ระบบย่อยเริ่มและจบ และเก็บแท็กรักของ สภาพของ ระบบ (เช่น, สภาพที่ถูกจำกัด).

system arbiters, ที่ระบุโดยชื่องาน QSYSARB และ QSYSARB2 ถึง QSYSARB5, คืองานศูนย์กลางและมีระดับความสำคัญ สูงสุดภายในระบบปฏิบัติการ. แต่ละ system arbiter ตอบสนองต่อเหตุการณ์ต่างๆ ที่ทั้งระบบที่ต้องถูกจัดการโดยทันที และ เหตุการณ์ที่สามารถจัดการโดยงานเดี่ยวได้อย่างมีประสิทธิภาพมากกว่าหลายงาน.

system arbiter (QSYSARB) ยังรับผิดชอบสำหรับการเริ่มงาน Logical Unit Services (QLUS) ในระหว่าง IPL ด้วย. system arbiter ยังคง แอ็คทีฟจนกว่าระบบจะจบลง.

ข้างล่างนี้เป็นรายการของ system arbiters.

Qsysarb (system arbiter)

system arbiter จะจัดเตรียมสถานะแวดล้อมสำหรับการรันฟังก์ชันที่มี ความสำคัญสูง. และยังช่วยจัดการรีซอร์สระบบ และช่วยติดตามสภาพของ ระบบให้อีกด้วย. system arbiter จะตอบสนองเหตุการณ์ต่างๆ ที่ทั้ง ระบบที่จะต้องมีการ จัดการในทันที และเหตุการณ์ที่สามารถจัดการได้ อย่างมีประสิทธิภาพมากขึ้นด้วยงานชิ้นเดียว. Qsysarb, Qtaparb (tape arbiter), และ Qcmnarbxx (communications arbiters) รับผิดชอบสำหรับการประมวลผลคำขอการสื่อสาร, การลือกอุปกรณ์, โลไน้, คอนโทรลเลอร์, และ configuration อุปกรณ์, และการจัดการรีซอร์สทั่วทั้งระบบอื่น.

Qsysarb2 (system arbiter 2)

งานตัวนี้จะรับผิดชอบการจัดการเทปรีซอร์ส, จัดการ command analyzer space สำหรับการประมวลผลคำสั่ง และการ ประมวลผล อื่นๆ ที่ระบบสำหรับระบบปฏิบัติการ.

Qsysarb3 (system arbiter 3)

งานตัวนี้จะรับผิดชอบการสร้างและการดูแลรักษาโครงสร้างงานบน ระบบ. เมื่อใดก็ตามที่จำเป็นต้องใช้โครงสร้าง งานชั่วคราวหรือถาวร ในการเริ่มต้นงาน, คำขอดังกล่าวจะถูกประมวลผลโดย Qsysarb3. Qsysarb3 ยังเริ่ม และจบ งานระบบหลายอย่างด้วย.

Qsysarb4 (system arbiter 4)

งานตัวนี้จะรับผิดชอบการเริ่มต้นและจบระบบย่อย. ซึ่งจะรวม ถึงกระบวนการปิดเครื่อง (Pwrdownsys) ในระยะต้น.

Qsysarb5 (system arbiter 5)

งานตัวนี้จะรับผิดชอบการประมวลผลเหตุการณ์ต่างๆ ของเครื่อง. สิ่งนี้รวมถึง การจัดการ เหตุการณ์เพื่อสนับสนุน auxiliary power, พูลหน่วยความจำรองของระบบ (ASPs) และ storage threshold, และขีดจำกัดตารางการลือก. โดย ปกติแล้ว, เหตุการณ์ ของเครื่องจะถูกจัดการและข้อความ CPF ที่สัมพันธ์กันนั้นจะถูกส่งไปยัง Qsysopr และ Qhst.

งานสื่อสารของระบบ:

หัวข้อนี้มีรายการของงานสื่อสารของ ระบบ.

Qlus (logical unit services)

Qlus จัดการเหตุการณ์สำหรับอุปกรณ์โลจิคัลยูนิต, หรือที่เรียกกันว่า อุปกรณ์สื่อสาร. Qlus ยังรับผิดชอบการจัดสรร อุปกรณ์ให้กับระบบย่อย การสื่อสารที่ต้องอีกด้วย.

Qcmnarbxx (communication arbiters)

communications arbiters ที่มี Qsysarb (system arbiter) และ Qtaparb (tape arbiter) ประมวลผลงานสำหรับอุปกรณ์ทุกชนิด, ไม่เฉพาะแต่อุปกรณ์ การสื่อสาร. งานเหล่านี้รวมถึงการเชื่อมต่อระบบสื่อสาร, การตัดการเชื่อมต่อ, การลืออกอุปกรณ์, และการประมวลวิธีแก้ไขข้อผิดพลาด. เมื่อรีสตาร์ท, ค่ากำหนดของระบบ communication arbiter jobs (QCMNARB) จะกำหนดจำนวนของงาน communications arbiter ที่จะถูกเริ่มต้น. communications arbiter อย่างน้อย 3 ตัวจะถูกเรียกใช้งานบนระบบโปรเซสเซอร์เดียว.

Qsyscomm1 (การสื่อสารระบบ)

งานตัวนี้จะจัดการการสื่อสารบางตัวและ activity อินพุต/เอาต์พุต (I/O).

Q400filsvr (การสื่อสารระบบรีโมตไฟล์)

งานนี้ปฏิบัติการสื่อสารผ่านอินเทอร์เน็ตเพสการทำโปรแกรมมิ่งทั่วไป (APPN หรือ APPC) สำหรับระบบรีโมตไฟล์.

งานฐานข้อมูล:

ข้างล่างนี้เป็นรายการของงานฐานข้อมูล.

Qdbfstccol (การรวบรวมสถิติไฟล์ฐานข้อมูล)

งานนี้จะรวบรวมข้อมูลสถิติเกี่ยวกับไฟล์ฐานข้อมูล. สถิติเหล่านี้มีความสำคัญอย่างยิ่ง ต่อการแก้ไขการเคียวรีฐานข้อมูลให้มีประสิทธิภาพสูงสุด.

Qdbsrvxr (การอ้างอิงข้ามฐานข้อมูล) และ Qdbx###xr สำหรับกลุ่มดิสก์พูลอิสระ

งานตัวนี้จะดูแลรักษาไฟล์การอ้างอิงข้ามระบบระดับไฟล์ใน Qsys. ไฟล์เหล่านี้จะประกอบด้วยข้อมูลการอ้างอิงข้ามเกี่ยวกับไฟล์ ฐานข้อมูลและข้อมูล SQL ข้ามระบบ. ไฟล์เหล่านี้จะเริ่มต้นด้วยคำเสริมหน้า เป็น Qadb ในไลบรารี Qsys. ไฟล์หลักที่จะต้องได้รับการดูแล คือ Qadbxref, ซึ่งเป็นไฟล์อ้างอิงข้าม. ไฟล์นี้ประกอบด้วย เร็กคอร์ดของฐานข้อมูลฟิสิกัล, ฐานข้อมูลโลจิคัล, DDM และไฟล์ Alias บนระบบ. Qdbsrvxr จะทำงานขึ้นมาเมื่อมีการสร้าง, เปลี่ยน, ลบ, เรียกคืน, เปลี่ยนชื่อ, หรือสภาวะความเป็นเจ้าของของไฟล์ได้ถูกเปลี่ยนไป.

Qdbsrvxr2 (การอ้างอิงข้ามฐานข้อมูล 2) และ Qdbx###xr2 สำหรับ กลุ่มดิสก์พูลอิสระ

งานตัวนี้จะดูแลรักษาไฟล์อ้างอิงข้ามแบบสองฟิลด์. Qadbifld ในไลบรารี Qsys คือไฟล์อ้างอิงข้ามฟิลด์. Qadbkfld ในไลบรารี Qsys คือไฟล์อ้างอิงข้ามคีย์ฟิลด์. Qdbsrvxr2 จะถูกเรียกทำงานเมื่อมีการสร้าง, เปลี่ยน หรือลบไฟล์.

Qdbsrv01 (เซิร์ฟเวอร์ฐานข้อมูล) และ Qdb###v01 สำหรับกลุ่มดิสก์พูลอิสระ

งานตัวนี้เรียกได้ว่าเป็น database maintenance task dispatcher. จำนวนงานเซิร์ฟเวอร์ฐานข้อมูลบนระบบ คือ หนึ่งบวกด้วยจำนวนสองเท่า ของจำนวนโปรเซสเซอร์, หรือหนึ่งบวกด้วยจำนวนสองเท่าของจำนวน ASP อันใดอันหนึ่งที่มากกว่า. จำนวนเริ่มต้นต่ำสุดคือห้า. Qdbsrv01 เป็น ระบบหลักที่มอบหมายงานให้กับงานอื่นต่อ. ปกติแล้ว, Qdbsrv01 จะแอ็คทีฟทันทีหลังจากมีการเรียกคืนไลบรารีที่ประกอบด้วยไฟล์ฐานข้อมูล. ฟังก์ชันของงานตัวนี้คือ:

- การให้สัญญาณกับภารกิจ system-managed access path protection (SMAPP) Licensed Internal Code (LIC), ที่แอ็กเซสพาทตัวใหม่ถูกเรียกคืนมา. จากนั้น SMAPP จะระบุได้ว่าแอ็กเซสพาทเหล่านี้ต้องการให้มีการ ปกป้องหรือไม่.
- การเตรียมลิสต์ของแอ็กเซสพาทที่จำเป็นต้องสร้างขึ้นใหม่ เนื่องจากแอ็กเซสพาทไม่ได้ถูกเรียกคืน.

ในส่วนครั้งแรกของงานเซิร์ฟเวอร์ฐานข้อมูลที่เหลืออยู่นั้น จะประมวลผลคำขอที่มีระดับความสำคัญสูง, และคำขอที่มีระดับความสำคัญต่ำจะ ประมวลผลที่ครั้งหลัง. (ตัวอย่าง: Qdbsrv02 ถึง Qdbsrv05 มีระดับความสำคัญสูง, Qdbsrv06 ถึง Qdbsrv09 มีระดับความสำคัญต่ำ.)

Qdbsrvxx (เซิร์ฟเวอร์ฐานข้อมูล, ระดับความสำคัญสูง) และ Qdbs###vxx สำหรับกลุ่มดิสก์พูล อีสระ ###

งานเหล่านี้จะทำเจอร์นัลและ commitment control maintenance ให้กับ ระบบและถูกจัดว่าเป็นงานที่มีการรันรวดเร็ว หรือใช้เวลาน้อย.

Qdbsrvxx (เซิร์ฟเวอร์ฐานข้อมูล, ระดับความสำคัญต่ำ) และ Qdbs###vxx สำหรับกลุ่มดิสก์พูล อีสระ ###

งานเหล่านี้ดูแลรักษาแอ็คเซสพาร์ไนไฟล์ข้อมูลของผู้ใช้. โดยปกติ, งานเหล่านี้จะ inactive, แต่ในบางกรณี, งานอาจปฏิบัติงานการสร้างแอ็คเซสพาร์ไนไฟล์ใหม่. เหตุผลบางประการที่งานเหล่านี้อาจจะแอ็คทีฟคือ:

- การคืนสภาพไฟล์ฐานข้อมูลที่ไม่ได้ถูกบันทึกพร้อมกับแอ็คเซสพาร์ไน
- การคืนสภาพโลจิคัลไฟล์โดยไม่มีฟิสิคัลไฟล์ที่ไฟล์ตั้งอยู่
- การยกเลิกคำสั่ง Rgzpfm ในระหว่างที่ดำเนินการอยู่
- ความไม่ถูกต้องของดรรชนีเนื่องจากพบความเสียหายในดรรชนี
- activity การติดตั้ง Post-iServer เพื่อให้การอ้างอิงข้ามสมบรูณ์ หรือ DB upgrade activity อื่น
- การตรวจสอบข้อจำกัด

Qqqtemp1 และ Qqqtemp2 (database parallelism)

งานระบบ database parallelism ปฏิบัติการอะซิงโครนัสการประมวลผลฐานข้อมูล สำหรับ DB2® Multisystem. ถ้าผู้ใช้เดี่ยวไฟล์แบบกระจาย, งานหลายงานจะถูกเรียกใช้เพื่อให้ได้ผลลัพธ์ที่เร็วขึ้นโดยการแบ่งงานเป็นหลายงานในลักษณะขนาน .

งานระบบอื่น:

บทความนี้มีรายการของงานระบบชนิดอื่น.

Qalert (alert manager)

งานนี้ปฏิบัติการกิจที่จำเป็นในการประมวลผลข้อความเตือน (ถึงปัญหาที่เกิดขึ้นบนเน็ตเวิร์ก). การกิจนี้รวมถึง activity ต่างๆ เช่น การประมวลผลข้อความเตือน ที่ได้รับมาจากระบบอื่น, การประมวลผลข้อความเตือนที่สร้างขึ้นที่ระบบโลคัล, และการรักษาขอบเขตของการควบคุม.

Qdcobjx (คลายอ็อบเจ็ทระบบ)

งานนี้คลายอ็อบเจ็ทระบบปฏิบัติการที่ติดตั้งใหม่เมื่อจำเป็น. งานเหล่านี้ต้องการใช้หน่วยความจำเพื่อจะทำงาน. ถ้าหน่วยความจำที่พร้อมใช้งาน บนระบบของคุณลดลงน้อยกว่าขีดจำกัด, งานเหล่านี้จะสิ้นสุด. จำนวนของงานคลายการบีบอัดของอ็อบเจ็ทระบบคือ จำนวนของ โพรเซสเซอร์บวกด้วยหนึ่ง.

Qfilesys1 (ระบบไฟล์)

งานตัวนี้จะสนับสนุนการประมวลผลส่วนหลังของระบบไฟล์รวม. งานนี้ทำให้มั่นใจว่าการเปลี่ยนแปลงที่ไฟล์จะถูกบันทึกลงในหน่วยความจำ และปฏิบัติ activities การล้างระบบไฟล์ทั่วไปด้วย.

Qjobsd (ตารางเวลางาน)

งานตัวนี้จะควบคุมฟังก์ชันการจัดตารางเวลางานของระบบ. Qjobsd จะมอนิเตอร์ตัวจับเวลาของ job schedule entry และงานที่จัดตาราง เวลา.

Qli###cl สำหรับกลุ่มดิสก์พูลอีสระ ### (การล้างไลบรารี)

งานนี้ล้างไลบรารีบนดิสก์พูลอีสระ.

Qli###rp สำหรับกลุ่มดิสก์พูลอีสระ ### (การล้างอ็อบเจ็ท)

งานนี้ล้างอ็อบเจ็ทที่แทนที่บนไลบรารีดิสก์พูลอีสระ.

Qlur (LU 6.2 resynchronization)

Qlur จะจัดการกระบวนการ two-phase commit resynchronization.

Qpfradj (การปรับประสิทธิภาพ)

งานตัวนี้จะจัดการการเปลี่ยนแปลงของขนาดและ activity level ของพูล หน่วยความจำ. คำขอทั้งหมดที่จะเปลี่ยนแปลงพูลหน่วยความจำจะถูกประมวลผลโดยงานตัวนี้. นอกจากนี้, ถ้าคำกำหนดของระบบ Automatically adjust memory pools and activity levels (Qpfradj) ถูกเซตเป็น 2 หรือ 3, งานนี้จะเปลี่ยนขนาดและ activity levels ของพูลของหน่วยความจำอย่างไดนามิก เพื่อพัฒนาประสิทธิภาพระบบ.

Qsplmaint (การรักษาพูลระบบ) และ Qspmn##### สำหรับ กลุ่มดิสก์พูลอิสระ

งานนี้ทำหน้าที่ของฟังก์ชันการสพูลของระบบที่ประกอบด้วย:

- การล้างสพูลไฟล์หลังจาก IPL หรือกลุ่มดิสก์พูลอิสระถูก varied on
- ย้าย stranded spool files ของเอาต์พุตคิวผู้ใช้ที่เสียหายในพูลหน่วยความจำรอง ระบบย่อยหรือในพูลหน่วยความจำรองผู้ใช้ระดับต้น ไปใน เอาต์พุตคิว QSPRCLOUTQ ในไลบรารี QRCL
- ลบเมมเบอร์ฐานข้อมูลสพูลซึ่งมีข้อมูลและแอตทริบิวต์ของสพูลไฟล์ที่ลบออก
- ลบเมมเบอร์ฐานข้อมูลสพูลที่ไม่ได้ถูกใช้ซ้ำภายใน เวลาที่ระบุในคำกำหนดของระบบ Automatically clean up unused printer output storage (QRCLSPLSTG)

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

งานนี้ปฏิบัติการดำเนินการสพูลไฟล์สำหรับกลุ่มดิสก์พูล อิสระเฉพาะ.

Qtaparb (อุปกรณ์เทป)

งานนี้ประมวลผลงานที่เกี่ยวข้องกับอุปกรณ์เทป รวมถึง การล๊อคอุปกรณ์ และการประมวลผลการแก้ไขข้อผิดพลาด.

| Qnwharbxx

งานระบบเหล่านี้จัดการเหตุการณ์ที่เกี่ยวข้องกับอุปกรณ์ Network Server Host Adapter (NWSH). จะมีการเริ่มงานเหล่านี้อย่างน้อยหนึ่งงานเสมอในระหว่าง IPL ปัจจุบัน.

| Qwcpjobs

งานนี้จัดการการล้างแบ็กกราว์นของโครงสร้างงานถาวร.

| Qwctjobs

งานนี้จัดการการล้างแบ็กกราว์นของโครงสร้างงานชั่วคราว.

อีพชั่นการจัดตารางเวลางาน

ฟังก์ชันการจัดตารางเวลางานช่วยในการจัดตารางเวลางานแบ็ตซ์ iSeries ตามเวลา ที่ต้องการได้. คุณสามารถจัดตารางเวลาที่งานจะถูกรีลีสจากคิวงาน ณ เวลา ที่ต้องการ, หรือคุณสามารถใช้ job schedule entry เพื่อส่งงานของคุณไปที่คิวงาน โดยอัตโนมัติ ณ เวลาที่คุณระบุ. การจัดตารางเวลางานช่วยให้คุณควบคุม วันที่และเวลาที่งานแบ็ตซ์จะถูกส่งหรือกลายเป็นมีสิทธิที่จะเริ่ม จากคิวงาน. ความยืดหยุ่นนี้เป็นสิ่งที่มีประโยชน์เมื่อคุณปรับสมดุลเวิร์กโหลด บนระบบของคุณ.

ตัวอย่างเช่น, คุณสามารถใช้การจัดตารางเวลางานเพื่อมอบหมายงานที่ซ้ำ อย่าง เช่นการส่งจดหมายแจ้งเชิญประชุมซ้ำๆ, ค่าจ้าง, หรือรายงานประจำสัปดาห์และ ประจำเดือน จากตารางเวลางานของคุณไปที่ตารางเวลางานของระบบได้. การจัดตารางเวลางานแบ็ตซ์สามารถทำได้สี่วิธี.

Management Central scheduler

iSeries Navigator นำเสนอ integrated scheduler, Management Central scheduler, เพื่อจัดตั้ง ตารางเวลาที่คุณต้องการใช้งานของคุณประมวลผล. คุณมีอ็อปชันการเลือกที่จะปฏิบัติ ภารกิจในทันทีหรือในภายหลัง. คุณสามารถใช้ Management Central scheduler เพื่อจัดตารางเวลาภารกิจส่วนใหญ่ใน Management Central.

หน้าต่าง Management Central Scheduler พร้อมใช้งานได้ตลอดเวลา ที่คุณเห็นปุ่ม **Schedule** บนหน้าต่าง iSeries Navigator.

หมายเหตุ: ถ้าคุณสามารถติดตั้ง Advanced Job Scheduler บนเซิร์ฟเวอร์ Management Central, ปุ่ม **Schedule** จะเริ่ม Advanced Job Scheduler แทน Management Central scheduler.

Advanced Job Scheduler

ไลเซนส์โปรแกรม Advanced Job Scheduler (5722-JS1) คือตัวจัดตารางเวลาที่ยอดเยี่ยมที่ช่วยให้งานแบบ unattended ถูกประมวลผลได้ตลอด 24 ชั่วโมงต่อวัน, 7 วันต่อสัปดาห์. เครื่องมือการจัดตารางเวลานี้แนะนำคุณลักษณะปฏิบัติเพิ่มเติม และแนะนำการควบคุมเหตุการณ์ ที่จัดตารางเวลาในระดับที่สูงขึ้นกว่าตัวจัดตารางเวลา Management Central. นอกจากนี้ คุณยังสามารถดูประวัติความสมบูรณ์ของงานและบริหารการแจ้งเตือนสถานะของงานได้อีกด้วย.

ไม่จำเป็นต้องติดตั้งไลเซนส์โปรแกรม Advanced Job Scheduler บนระบบจุดปลายทุกระบบในเน็ตเวิร์ก Management Central ของคุณ. เมื่อคุณติดตั้ง Advanced Job Scheduler ในระบบศูนย์กลาง, งานหรือภารกิจที่คุณกำหนดบนระบบจุดปลายจะรวบรวมข้อมูลงานที่ต้องการจากระบบศูนย์กลาง. อย่างไรก็ตาม, คุณต้องตั้งค่าข้อมูล job definition ทั้งหมดบนระบบ ศูนย์กลาง.

ถ้ามีการติดตั้ง Advanced Job Scheduler แบบโลคัลบนหลายๆ ระบบในเน็ตเวิร์กของคุณ, คุณสามารถจัดตารางเวลางานภายนอกเน็ตเวิร์ก Management Central ได้. ภายใต้ **My Connections** ใน iSeries Navigator, คุณมีสิทธิเข้าใช้งาน Advanced Job Scheduler บนระบบโลคัลนั้นเมื่อคุณขยาย **Work Management**.

Advanced Job Scheduler for Wireless:

Advanced Job Scheduler for Wireless คือซอฟต์แวร์แอพลิเคชันที่ช่วยให้คุณสามารถเข้าใช้งาน Advanced Job Scheduler บนอุปกรณ์เชื่อมต่อ อินเทอร์เน็ตหลายประเภท, เช่น โทรศัพท์ที่พร้อมเชื่อมต่อกับอินเทอร์เน็ต, PDA Web browser หรือ PC Web browser.

คุณลักษณะพิเศษไร้สายของ Advanced Job Scheduler มีอยู่บนระบบ iSeries ของคุณ, ซึ่งมีการติดตั้ง Advanced Job Scheduler, และอนุญาตให้คุณเข้าถึงงานของคุณ และ activity, ตลอดจนสามารถส่งข้อความให้ผู้รับบนระบบของคุณ, และหยุด และเริ่มการมอนิเตอร์ Advanced Job Scheduler. Advanced Job Scheduler for Wireless อนุญาตให้ผู้ใช้แต่ละรายปรับการตั้งค่าตามความต้องการและความชอบใน การบราวซ์. ตัวอย่าง, ผู้ใช้คนหนึ่งสามารถแสดง activity, แสดงงานต่างๆ, และปรับงานที่แสดงตามความต้องการ.

Advanced Job Scheduler for Wireless อนุญาตให้คุณเข้าถึงงานของคุณ เมื่อ คุณไม่สามารถเข้าใช้งานเทอร์มินัลหรืออิมูเลเตอร์ iSeries ตามปกติ. เชื่อมต่อกับอินเทอร์เน็ตด้วยอุปกรณ์เคลื่อนที่ของคุณและป้อน URL สำหรับ Advanced Job Scheduler for Wireless servlet. การทำเช่นนี้จะเรียกใช้เมนูที่คุณสามารถเข้าใช้งาน Advanced Job Scheduler แบบเรียลไทม์ได้.

Advanced Job Scheduler for Wireless ทำงานบนอุปกรณ์สองประเภท. อุปกรณ์ Wireless Markup Language (WML) คือ โทรศัพท์มือถือที่พร้อมเชื่อมต่อกับอินเทอร์เน็ต. อุปกรณ์ Hypertext Markup Language (HTML) คือ PDA หรือ PC เว็บเบราว์เซอร์. ตลอดหัวข้อนี้, อุปกรณ์ที่แตกต่างอื่นจะถูกอ้างอิงถึงเป็น WML และ HTML.

Job schedule entries

ถ้าระบบของคุณไม่มี Management Central Scheduler หรือ Advanced Job Scheduler, คุณยังคงสามารถจัดตารางเวลางานได้โดยใช้ job schedule entry, ซึ่งเข้าใช้งานได้จากอินเตอร์เฟซแบบอักษร. ด้วยการใช้อินเตอร์เฟซนี้ คุณจึงสามารถจัดตารางเวลางานให้วันซ้ำหรือวันเพียงครั้งเดียวได้.

เนื่องจาก job schedule entries คือ entries ในอ็อบเจ็กต์ถาวร, entry ชนิดนี้ จึงไม่ได้อยู่บนคิวงานเหมือนกับงานที่จัดตารางเวลา, ดังนั้นจึงไม่สูญหาย เมื่อคิวงานถูกลบ. คุณยังสามารถบันทึกและคืนสภาพอ็อบเจ็กต์ ตารางเวลางานได้ด้วย. นี่เป็นวิธีการสำรองข้อมูลตารางเวลางาน ของคุณ.

เมื่อคุณต้องการให้งานประมวลผลในช่วงเวลาสม่ำเสมอ, คุณสามารถสร้าง job schedule entry สำหรับงานได้. job schedule entry มีข้อมูลทั้งหมดที่จำเป็น สำหรับการส่งงานและข้อมูลการจัดตารางเวลางาน. แต่ละ entry ในอ็อบเจ็กต์ถูกระบุอย่างไม่ซ้ำกันโดยชื่องานที่คุณให้และหมายเลข entry 6 ตำแหน่งที่ระบบกำหนดให้. ไม่มีสอง entries ใดมีชุดของ ชื่องานและหมายเลข entry เหมือนกัน.

นอกจากนี้ job schedule entry ยังมีข้อมูลที่ระบบใช้ในการจัดการ entry ในบางสถานการณ์ด้วย. ข้อมูลที่กำหนดงาน คล้ายกับพารามิเตอร์ที่ระบุบนคำสั่ง Submit Job (SBMJOB), ได้แก่ ชื่องาน, รายละเอียดของงาน, คิวงาน, โปรไฟล์ผู้ใช้, และ message queue. local data area (LDA) ของงานที่ส่งจาก job schedule entry จะว่างเปล่าเมื่องานเริ่มต้น.

job schedule entries ทั้งหมดมีอยู่ในอ็อบเจ็กต์ตารางเวลางาน. อ็อบเจ็กต์ตารางเวลางาน, QDFTJOBSCD อยู่ในไลบรารี QUSRSYS และมีชนิด อ็อบเจ็กต์เป็น *JOBSCD. คุณไม่สามารถสร้าง, ลบ, เปลี่ยนชื่อ, หรือทำซ้ำอ็อบเจ็กต์ตารางเวลางาน. คุณไม่สามารถย้ายอ็อบเจ็กต์ไปที่ไลบรารีอื่น. อ็อบเจ็กต์ตารางเวลางานถูกจัดส่งมา พร้อมกับสิทธิพิเศษ *CHANGE. นี่เป็นสิทธิพิเศษที่จำเป็นสำหรับการ ใส่เพิ่ม, เปลี่ยน, พัก, รีเซ็ต, หรือลบ job schedule entries.

หมายเหตุ: คุณยังสามารถ จัดตารางเวลางานที่เกิดซ้ำได้ด้วย โดยใช้ Management Central Scheduler หรือ Advanced Job Scheduler.

ตัวอย่าง: job schedule entry:

หัวข้อนี้แนะนำตัวอย่างสำหรับการใช้คำสั่ง Add Job Schedule Entry (ADDJOBSCDE).

จัดตารางเวลางานรายเดือน: ตัวอย่างนี้แสดงวิธีการส่งงาน เพื่อรันโปรแกรม INVENTORY เวลา 11:30 p.m. ในวันสุดท้ายของทุกเดือน ยกเว้น New Year's Eve.

```
ADDJOBSCDE JOB(MONTHEND)
CMD(CALL INVENTORY)
SCDDATE(*MONTHEND)
SCDTIME('23:30:00')
FRQ(*MONTHLY)
OMITDATE('12/31/05')
```

จัดตารางเวลางานรายวัน: ตัวอย่างนี้แสดงวิธีการส่งงานเพื่อรันโปรแกรม DAILYCLEAN ทุกวัน เวลา 6:00 p.m. งานจะรันภายใต้โปรไฟล์ผู้ใช้ SOMEPMGR. งานนี้จะไม่ถูกส่ง ถ้าระบบปิดหรืออยู่ในสภาพถูกจำกัดในเวลานั้น.

```
ADDJOBSCDE JOB(*JOB)
CMD(CALL DAILYCLEAN)
SCDDAY(*ALL)
SCDTIME('18:00:00')
```


SCDDATE(*NONE)
USER(SOMEPMGR)
FRQ(*WEEKLY)
RCYACN(*NOSBM)

การจัดตารางเวลางานรายสัปดาห์: ตัวอย่างนี้แสดงวิธีการส่งงานเพื่อรันโปรแกรม PGM1 ทุกสัปดาห์โดยเริ่มต้นวันที่ 12/17/05 ที่เวลาปัจจุบัน. เนื่องจาก 12/17/05 เป็นวันเสาร์, ดังนั้นงานจึงจะถูกส่งทุกวันเสาร์, และรันภายใต้โปรไฟล์ผู้ใช้

```
PGMR1. ADDJOBSCDE JOB(*JOB)  
CMD(CALL PGM1)  
SCDDATE('12/17/05')  
FRQ(*WEEKLY)  
USER(PGMR1)
```

การจัดตารางเวลางานทุกวันจันทร์และ วันพุธที่สาม: ตัวอย่างนี้แสดงวิธีการส่งงานเพื่อรันโปรแกรม PGM2 ในวันจันทร์ที่สามและวันพุธที่สามของเดือน ในเวลา 11:30 p.m. งานนี้จะถูกส่งในวันจันทร์หรือวันพุธที่สามถัดไปในเวลา 11:30 p.m., ขึ้นอยู่กับว่าวันเหล่านั้นของเดือนนี้ได้ผ่านไปหรือไม่. ถ้าเมื่อวานเป็น วันจันทร์ที่สาม, วันนี้คือวันอังคารที่สาม, และพรุ่งนี้คือวันพุธที่สาม, งานจะถูกส่งวันพรุ่งนี้, แล้วไม่ส่งอีกจนกว่าจะถึงเดือนถัดไป.

```
ADDJOBSCDE JOB(*JOB)  
CMD(CALL PGM2)  
SCDDAY(*MON *WED) FRQ(*MONTHLY)  
SCDDATE(*NONE)  
RELDAYMON(3) SCDTIME('23:30:00')
```

การจัดตารางเวลางาน ทุกวันจันทร์ที่หนึ่งและสาม: ตัวอย่างนี้แสดงวิธีการส่งงานเพื่อรันโปรแกรม PAYROLL ในวันจันทร์ที่หนึ่งและวันจันทร์ที่สามของทุกเดือน ในเวลา 9:00 a.m. งานรันภายใต้โปรไฟล์ผู้ใช้ PAYROLLMGR.

```
ADDJOBSCDE JOB(PAYROLL)  
CMD(CALL PAYROLL)  
SCDDAY(*MON) FRQ(*MONTHLY)  
SCDDATE(*NONE)  
RELDAYMON(1 3) SCDTIME('09:00:00')  
USER(PAYROLLMGR)
```

การจัดตารางเวลางานทุกวันทำงาน: ตัวอย่างนี้แสดงวิธีการส่งงานเพื่อรัน PGM4 ทุกวันทำงาน ในเวลา 7:00 p.m.

```
ADDJOBSCDE JOB(*JOB)  
CMD(CALL PGM4)  
SCDDAY(*MON *TUE *WED *THU *FRI)  
SCDDATE(*NONE)  
SCDTIME('19:00:00') FRQ(*WEEKLY)
```

การบันทึก job schedule entry: ตัวอย่างนี้แสดงวิธีการส่งงานหนึ่งครั้งและบันทึก entry.

```
ADDJOBSCDE JOB(*JOB)  
CMD(CALL SAVED)  
FRQ(*ONCE)  
SAVE(*YES)
```

คำสั่งการส่งงาน

คำสั่งอินเตอร์เฟซแบบอักษรนี้ควบคุมเวลาที่งาน จะถูกรีสในคิวงาน. คำสั่งนี้เป็นวิธีที่ง่ายในการจัดตารางเวลางานที่ต้องรันเพียงครั้งเดียวเท่านั้น. คำสั่งนี้อุญาตให้คุณใช้แอตทริบิวต์งานที่กำหนดสำหรับ งานปัจจุบันของคุณได้หลายแอตทริบิวต์.

เมื่อคุณจัดตารางเวลางานเพื่อที่จะรันเพียงครั้งเดียว (คำสั่งแบบอักษร SBMJOB), งานจะถูกรีลีสจากคิวงานที่เวลาที่จัดตารางเวลาไว้. ข้อมูลต่อไปนี้เป็นข้อสรุปสำหรับงานระบบที่จะเกิดขึ้นเมื่อคุณใช้ SBMJOB จัดตารางเวลางานแบ็ดซ์.

1. คุณจัดตารางเวลางานโดยใช้ iSeries Navigator interface (**Basic Operations** → **Jobs** → **คลิกขวาที่งาน** → **Properties** → **Job Queue tab**) หรือคำสั่งที่ใช้อินเตอร์เฟซแบบอักษร (SBMJOB ที่มีการระบุพารามิเตอร์ SCDATE และ SCDTIME อย่างใดอย่างหนึ่ง).
2. งานยังคงอยู่บนคิวงานในสถานะถูกจัดตารางเวลา (สถานะ SCD) จนกว่าจะถึงวันที่และเวลาที่ระบุโดยพารามิเตอร์.
3. ณ เวลาที่จัดตารางเวลาไว้, งานจะถูกรีลีสจากคิวงาน. สถานะของงาน จะเปลี่ยนจากถูกจัดตารางเวลา (SCD) เป็นรีลีส (RLS), ยกเว้นว่างานจะถูกพัก (SCDHLD), ซึ่งสถานะจะเปลี่ยนจากถูกจัดตารางเวลาเป็นถูกพัก (HLD).
4. งานถูกประมวลผลเช่นเดียวกับงานอื่นๆ บนคิวงาน.
5. งานจะเริ่มต้นถ้ามีสภาพปกติ (เช่น คิวงานถูกจัดสรรไปที่ระบบย่อยที่แอคทีฟ และงานสูงสุดยังไม่แอคทีฟ).

หมายเหตุ: วิธีการนี้จะวางงานบนคิวงานในทันที, ดังนั้นถ้าคิวงาน ถูกลบออกก่อนวันที่และเวลาที่จัดตารางเวลา, คุณจะเสียงานของคุณ.

ข้อควรพิจารณาเกี่ยวกับตารางเวลางาน

เมื่อทำการเลือกผลิตภัณฑ์ตารางเวลางาน, คุณต้องพิจารณาความหลากหลายของคุณลักษณะพิเศษต่างๆ. ต่อไปนี้เป็นรายการของคุณลักษณะพิเศษที่ต้องพิจารณาเมื่อมีการตัดสินใจว่าจะใช้ตารางเวลางานใด:

- **การจัดตารางเวลาแบบอัตโนมัติ**
 - ความยืดหยุ่นในการจัดตารางเวลางาน
 - การประมวลผลงานที่ไม่ตั้งใจ (หรือ ตั้งใจ) ตลอด 24 ชั่วโมงต่อวัน, 7 วันต่อสัปดาห์, พร้อมด้วยการประยุกต์ทั้งหมดให้เข้ากับตารางเวลางานที่คุณตั้งไว้
 - ส่วนขยายตามธรรมชาติของระบบปฏิบัติการ iSeries
 - การควบคุมโดยสมบูรณ์ของงานที่ส่งเข้ามาว่า อย่างไร, เมื่อไร, และที่ไหน
 - งานส่วนขยายขึ้นอยู่กับ อ็อบเจกต์ (การเกิดขึ้นของไฟล์ หรือเรียกคอร์ตภายในฟิลิคัลไฟล์), activity หรือ inactivity ของงานอื่น, หรือ สถานะของไลน์, คอนโทรลเลอร์, หรือ ระบบย่อย
 - เสรีจลิน์ฟังก์ชันการจัดการปฏิทิน, รวมถึงปฏิทินปีบัญชีและปฏิทินวันหยุด
 - การทำงานหลายอย่างต่อวัน
- **พารามิเตอร์ระบบ และพารามิเตอร์ที่ผู้ใช้กำหนด**
 - วันที่ปัจจุบัน®, วันที่ส่งงาน, วันที่ก่อนหน้านี้, และเวลาปัจจุบันสามารถส่งผ่านเข้าไปใน แอ็พพลิเคชันโปรแกรมได้
 - ค่าพารามิเตอร์ที่ผู้ใช้กำหนดสามารถสร้าง, เปลี่ยน, และส่งผ่านเข้าไปในแอ็พพลิเคชันโปรแกรมได้
- **การพยากรณ์เวิร์กโหลด/ประวัติ**
 - พยากรณ์งานที่จัดตารางเวลาทั้งหมดให้รัน สัปดาห์หน้า, เดือนหน้า, หรือวันถัดไป
 - Optimize ข้อกำหนดในการผลิต
 - การแทรกข้อมูลในอดีต และการบันทึก Advanced Job Scheduler activity ทั้งหมด
- **การจัดการเน็ตเวิร์ก**
 - สามารถตั้งค่างานบนเซิร์ฟเวอร์ iSeries ใดๆ ในเน็ตเวิร์กให้รันบนเซิร์ฟเวอร์ iSeries อื่น บนเน็ตเวิร์กได้
 - ให้ข้อมูลงานในอดีตที่เสร็จสิ้นของงานบนระบบที่มีการส่งงาน
 - เราสามารถส่งงานกลุ่มและงานส่วนเสริมผ่านทางเน็ตเวิร์กได้

- **การแจกจ่ายและการจัดการรายงาน**
 - การเรอต์, การมอนิเตอร์, และการควบคุมรายงานเอาต์พุตทั้งหมดที่สร้างโดย Advanced Job Scheduler หรือระบบปฏิบัติการ iSeries
 - การแจกจ่ายสพูลไฟล์ไปยังเอาต์พุตคิวหลายอัน หรือไปยังระบบรีโมตพร้อมด้วยตัวเลือกแบนเนอร์เพจ
 - สามารถทำสำเนาสพูลเอาต์พุตหรือส่งให้แก่ผู้ใช้ใดๆ บนเน็ตเวิร์ก iSeries ได้
- **การรักษาความปลอดภัย**
 - สามารถใช้การรักษาความปลอดภัย iSeries ที่มีอยู่ใน Advanced Job Scheduler ได้
 - ให้ระบุบุคคลในองค์กรของคุณที่มีสิทธิ์ในการตั้งค่าหรือเปลี่ยนแปลงข้อมูลเกี่ยวกับงานที่จัดตารางเวลา
 - เราสามารถระบุสิทธิ์ในการใช้งานสำหรับฟังก์ชันแต่ละฟังก์ชันของ Advanced Job Scheduler หรือสำหรับงานที่ระบุเฉพาะ อย่างไม่อย่างหนึ่งได้
- **Graphical user interface**
 - ความสามารถในการเลือกและคลิก เมื่อจัดตารางเวลางาน
 - การจัดการงาน
 - การคงไว้ของส่วนเสริม
 - การแท็กรีก activity ของตัวจัดตารางเวลาและข้อมูลไฟล์บันทึก
- **คุณลักษณะพิเศษของคีย์อื่นๆ**
 - หลายคำสั่งต่องาน
 - Definition สำหรับงาน LDA (Local Data Area)
 - มอนิเตอร์คอนโซลเพื่อรันงานในสภาพที่ถูกจำกัด
 - การตรวจสอบรันไทม์สูงสุดสำหรับแต่ละงาน
 - อินเตอร์เฟสโดยตรงกับระบบการเพจแบบข้อความของบริษัทที่ให้บริการอื่นๆ
 - บทบัญญัติสำหรับเอกสารคู่มือออนไลน์แบบเต็มของแต่ละงาน
 - ปุ่มสัมผัสเคอร์เซอร์ที่ขยายข้อความคำอธิบายบนจอแสดงผลทั้งหมด

การจัดตารางเวลางานและสภาพพร้อมใช้งานระบบ

ถ้าระบบปิดเครื่องหรืออยู่ในสภาพถูกจำกัดเมื่อถึงเวลาที่จัดตารางเวลา, จะไม่สามารถส่งงานจาก job schedule entries และไม่สามารถเปลี่ยนสถานะของงานที่จัดตารางเวลาแล้ว. อย่างไรก็ตาม, คุณสามารถควบคุมวิธีการที่ระบบใช้จัดการกับสถานการณ์นี้หลังจากที่ระบบทำ IPL หรือหลังจากที่ระบบออกมาจากสภาพที่ถูกจำกัด.

job schedule entries และงานที่จัดตารางเวลาจะถูกประมวลผลในลำดับของการจัดการตามปกติถ้างานเหล่านั้นสามารถดำเนินการได้. งานจากซอร์สอื่น สามารถเข้าสู่ระบบในขณะที่ missed job schedule entries และงานที่จัดตารางเวลากำลังถูกประมวลผล.

- **Job Schedule Entries:** คุณสามารถควบคุมวิธีการจัดการกับแต่ละ entry โดยใช้ค่าที่คุณระบุสำหรับการกู้คืนของ entry. คุณสามารถระบุให้งานยังคงถูกส่งโดยใช้ entry, ให้งานถูกส่งและพักบน คิวงาน, หรือให้งานไม่ถูกส่ง. ถ้าคุณร้องขอให้งานถูกส่ง, เพียงหนึ่งงานเท่านั้นที่จะถูกส่งจากแต่ละ entry, ไม่ว่าจำนวนการส่ง ที่ขาดไปจะเป็นเท่าใดก็ตามในขณะที่ระบบไม่พร้อมใช้งาน.
- **Scheduled Job:** ระบบจะตรวจสอบเพื่อกำหนดว่าเวลาที่จัดตารางเวลาใด ได้ผ่านไปแล้ว ในขณะที่ระบบไม่พร้อมใช้งาน. ถ้าพบงานที่จัดตารางเวลาที่มีเวลาผ่านพ้นไปแล้ว, สถานะของงานจะถูกอัปเดต.

คิวงาน

คิวงานประกอบด้วยรายการลำดับของงาน ที่รอการประมวลผลโดยระบบย่อย. คิวงานจะเป็นที่แรก ที่งานแบบแบดซ์ที่ส่งไปพักรอ ก่อนที่จะไปทำงาน อยู่ในระบบย่อย. งานจะถูกพักอยู่ที่นั่นจนกว่าจะมีสภาพตรงตามเงื่อนไขการทำงานต่างๆ.

งานที่อยู่ในคิวงานจะได้รับการประมวลผล, ก็ต่อเมื่อมีระบบย่อยที่แก็คที่รับเองงานที่อยู่ในคิวงานไป. เมื่อระบบย่อยเริ่มทำงาน, ระบบย่อยดังกล่าวจะพยายามจัดสรรคิวงานที่ถูกตั้งไว้ให้รับงานเข้ามา, และจะต้องจัดสรรคิวงานเพื่อทำการประมวลผลงานจากคิวงานนั้นให้ประสบความสำเร็จ. ดังนั้น, ในขณะที่ระบบย่อยหนึ่งสามารถประมวลผลงานจากหลายคิวงานได้, แต่จะมีระบบย่อยเพียงระบบเดียวเท่านั้นที่สามารถประมวลผลงานจากคิวงานหนึ่งในแต่ละครั้ง.

ระบบย่อยจะเลือกงานจากคิวงานตามระดับความสำคัญ, ภายใต้ข้อจำกัดที่ได้ตั้งค่าไว้สำหรับแต่ละระดับความสำคัญ. งานแต่ละชั้นมีค่าระดับความสำคัญคิวงานที่สามารถจัดการได้เมื่องานนั้นอยู่บนคิวงานผ่านทางคุณสมบัติของงาน. ชุดคิวงานพื้นฐานจะมาพร้อมกับระบบของคุณ. นอกจากนี้, คุณยังสามารถสร้างคิวงานเพิ่มเติมตามความต้องการได้อีกด้วย.

หมายเหตุ: สามารถเรียกใช้ APIs, เช่น Open List of Job Queues (QSPOLJBQ) และ Retrieve Job Queue Information (QSPRJQB), เพื่อรับข้อมูล เกี่ยวกับคิวงาน.

รายการที่จัดลำดับ

รายการที่จัดลำดับแสดงถึงลำดับที่งานจะปรากฏขึ้นบนคิวงาน. สภาพพร้อมใช้งาน, ระดับความสำคัญ, และค่าวันที่และเวลาช่วยกำหนด ลำดับของงานบนคิวงาน.

ไม่ใช่หมายเลขงานในการกำหนดลำดับของงานในคิวงาน, เช่นเดียวกับการไม่ใช่ผลกระทบเมื่องานถูกรัน.

สภาพพร้อมใช้งาน

หมายถึงสถานะของงานบนคิวงาน. ค่าที่เป็นไปได้ตามลำดับคือ กำลังรอ, ที่จัดตารางเวลา, และถูกพัก.

ระดับความสำคัญ

หมายถึงระดับความสำคัญของงานบนคิวงาน. ค่าระดับความสำคัญที่เป็นไปได้คือ 0-9, โดยที่ 0 เป็นระดับความสำคัญสูงสุด. ในกรณีที่งานเป็น งานที่จัดตารางเวลา, ระดับความสำคัญจะไม่มีส่วนในการจัดลำดับของ งานบนคิวงาน. ตัวอย่างเช่น, ถ้างานสองงานถูกจัดตารางเวลาให้รันเวลา 12:00:00, งานจะถูกจัดลำดับตามตำแหน่งของงานในตารางเวลางาน.

วันที่และเวลา

หมายถึงวันที่และเวลาของงาน:

- หากงานถูกจัดตารางเวลาไว้, วันที่และเวลาหมายถึงเวลาที่งานถูกจัดตารางเวลาให้รัน.
- หากงานไม่ได้ถูกจัดตารางเวลา, วันที่และเวลาหมายถึงเวลาที่งานเข้าสู่ระบบ.

หมายเหตุ: มีโอกาสที่วันที่และเวลาการจบงานจะถูกใช้เป็นที่และเวลาที่เซตด้วยตนเอง เพื่อจัดตำแหน่งงานที่ย้ายไปที่คิวงานอย่างถูกต้อง.

วิธีการทำงานของคิวงาน

คิวงานถูกจัดสรรโดยระบบย่อยผ่านทาง job queue entry. สามารถวางงานบนคิวงานได้แม้ว่ายังไม่ได้เริ่มระบบย่อย. เมื่อระบบย่อยเริ่มขึ้น, ระบบย่อยจะประมวลผลงานบนคิว.

subsystem description ระบุจำนวนสูงสุดของงาน (แบบแบดซ์หรือ แบบโต้ตอบ) ที่สามารถแก็คที่ไฟได้พร้อมกัน. จำนวนของงานที่สามารถ แก็คที่ไฟจากคิวงานใดๆ ถูกระบุอยู่ใน job queue entry.

งานบางงานบนคิวงานอาจไม่จำเป็นต้องพร้อมสำหรับการประมวลผลเมื่อ ระบบย่อยเริ่มขึ้น. งานที่จัดตารางเวลาสามารถวางบน คิวงานได้. งานสามารถถูกพักบนคิวจนกว่าผู้ควบคุมระบบจะรีลีสงาน. ถ้าระบบย่อยถูกจบลงก่อนที่งานทั้งหมดจะถูกประมวลผล, งานจะยังคงอยู่ บนคิวจนกว่าระบบย่อยจะเริ่มขึ้นอีกครั้ง, จนกว่าผู้ควบคุมระบบจะย้าย ไปที่คิวงานอื่น, จนกว่าผู้ควบคุมระบบจะลบบอก, หรือจนกว่าระบบย่อยอื่นจะจัดสรรคิวงานเดียวกัน.

subsystem description มากกว่าหนึ่ง description สามารถอ้างอิงคิวงานเดียวกันได้, แต่ในแต่ละครั้งจะมีระบบย่อยที่แอคทีฟเพียงระบบเดียวเท่านั้นที่สามารถใช้ คิวงานเป็นซอร์สของงานแบ็ตช์ได้. ดังนั้น, ถ้าระบบย่อยจบลงและงานยังคงอยู่บนคิวงาน, ระบบย่อยอื่นที่อ้างอิงไปที่คิวงานนั้นสามารถถูกเริ่มให้ประมวลผลงาน ได้. ถ้าระบบย่อยอื่นได้ถูกเริ่มต้นแล้วและกำลังรอคิวงานเดียวกัน, ระบบย่อยจะจัดสรรคิวงานให้โดยอัตโนมัติเมื่อคิวงาน พร้อมใช้งาน.

วิธีการนำงานมาจากคิวงาน

มีปัจจัยหลายอย่างที่กำหนดวิธีการเลือกงานจากคิวงาน และเริ่มต้น.

งานแอคทีฟสูงสุดสำหรับระบบย่อย

ตัวเลขนี้แสดงจำนวนสูงสุดของงานที่สามารถรันได้ในระบบย่อยหนึ่ง. เมื่อใช้งานถึงจำนวนสูงสุดนี้, จะไม่สามารถเริ่มงานเพิ่มเติมในระบบย่อย.

งานแอคทีฟสูงสุดสำหรับคิวงาน

ตัวเลขนี้แสดงจำนวนสูงสุดของงานจากคิวงานที่สามารถรันพร้อมกันได้ในระบบย่อย. หลังจากใช้งานถึงจำนวนสูงสุดนี้แล้ว, จะ ไม่สามารถเริ่มงานเพิ่มเติมจากคิวงานนั้น.

ระดับความสำคัญบนคิวงาน

งานที่กำลังรอที่จะรันจะถูกเลือกขึ้นมารันตามระดับความสำคัญใน คิวงาน. ระบบจะพยายามที่จะรันงานที่มีระดับความสำคัญสูงกว่าก่อน (ระดับความสำคัญในคิวงานมีตั้งแต่ 0-9 โดยที่ 0 คือระดับความสำคัญสูงสุด), แต่ถ้าจำนวนของงานที่รันจากรดับความสำคัญมีมากถึงค่า Maximum Active Jobs, ระดับความสำคัญถัดไปจะถูกประมวลผลต่อไป. (ถ้างานที่มีระดับความสำคัญเดียวกัน เข้าสู่คิวงาน, งานแรกที่ส่งจะรันก่อน, จากนั้นเป็นงานที่สอง, และต่อไป.)

ลำดับ คุณสามารถระบุลำดับใน job queue entry ของ subsystem description ได้. หมายเลขลำดับกำหนดลำดับซึ่งระบบย่อยจะประมวลผลคิวงาน. ระบบย่อยจะดึงเอางานที่มีหมายเลขลำดับต่ำสุดออกจากคิวงานก่อน. หากไม่มีงานอื่นบนคิวงาน, หรือมีการไปถึงค่าสูงสุดค่าใดค่าหนึ่งที่เกี่ยวข้องกับคิวงาน, ระบบย่อยจะประมวลผลคิวงานที่มีหมายเลขลำดับสูงสุดถัดไป.

Job queue entry

job queue entry ระบุคิวงานซึ่งงานจะถูกเลือกสำหรับการรันในระบบย่อย. พารามิเตอร์ใน job queue entry ที่ควบคุมวิธีการจัดการคิวงานมีอยู่ห้าพารามิเตอร์.

Subsystem Description (SBSD)

นี่คือชื่อและโลบรารีของ subsystem description ซึ่ง job queue entry จะถูกใส่เพิ่มไป.

คิวงาน (JOBQ)

ระบุชื่อและโลบรารีของคิวงานที่เป็นต้นทางของงานแบ็ตช์ ซึ่งถูกเริ่มโดยระบบย่อย.

งานแอคทีฟสูงสุด (MAXACT)

ระบุจำนวนสูงสุดของงานที่สามารถแอคทีฟได้พร้อมกัน จากคิวงานนี้.

หมายเลขลำดับ (SEQNBR)

ระบุหมายเลขลำดับสำหรับคิวงานนี้, ซึ่งถูกใช้โดยระบบย่อย เพื่อกำหนดลำดับการประมวลผลคิวงาน.

ระดับความสำคัญแอ็คทีฟสูงสุด 1 (ถึง 9) (MAXPTYx)

ระบุจำนวนของงานที่สามารถเริ่มสำหรับระดับความสำคัญงาน ที่ระบุ.

วิธีการจัดสรรคิวงานที่ระบบย่อย

สามารถเชื่อมโยงคิวงานกับหลายระบบย่อยได้ แต่คิวงานสามารถถูกจัดสรรให้กับระบบย่อยเพียงหนึ่งระบบเท่านั้นในแต่ละครั้ง. เมื่อระบบย่อยเริ่มต้น, การมอนิเตอร์ระบบย่อยจะพยายามจัดสรรคิวงานแต่ละคิวที่กำหนดใน subsystem job queue entries.

ถ้าคิวงานได้ถูกจัดสรรโดยระบบย่อยอื่นแล้ว, ระบบย่อยแรกต้องจบ และจัดสรรคืนคิวงานก่อน ระบบย่อยที่สองจึงจะสามารถจัดสรรคิวงาน ได้. หลังจากที่เริ่มต้น, ระบบย่อยที่สองนี้จะจัดสรรคิวงานที่ได้กำหนด ที่ระบบย่อยเมื่อคิวงานพร้อมใช้งาน.

หากคิวงานไม่มีอยู่เมื่อเริ่มระบบย่อย, คิวงานจะถูกจัดสรร ที่ระบบย่อยเมื่อเกิดอย่างใดอย่างหนึ่งดังนี้:

- คิวงานถูกสร้างขึ้น.
- คิวงานถูกเปลี่ยนชื่อเป็นชื่อที่กำหนดที่ระบบย่อย.
- คิวงานถูกย้ายไปยังไลบรารีอื่นและชื่อที่ถูกต้องใหม่ ตรงกับชื่อใน subsystem description.
- ไลบรารีที่มีคิวงานถูกเปลี่ยนชื่อ และชื่อที่ถูกต้องใหม่ ตรงกับชื่อใน subsystem description.

หลายคิวงาน

ในหลายกรณี, การใช้ QBATCCH เพียงหนึ่งคิวงานพร้อมกับค่าดีฟอลต์งานแอ็คทีฟหนึ่งงานจะเพียงพอสำหรับความต้องการของคุณ. หากไม่เพียงพอ, คุณอาจต้องการมีหลายคิวงานเพื่อให้บางคิวงานแอ็คทีฟในระหว่างชั่วโมงทำงานปกติ, บางคิวงานสำหรับวัตถุประสงค์พิเศษ, และบางคิวงาน แอ็คทีฟหลังชั่วโมงทำงานปกติ.

ตัวอย่างเช่น, คุณสามารถกำหนดคิวงานที่แตกต่างสำหรับ:

งานที่รันนานเพื่อให้คุณสามารถควบคุมจำนวนงานที่แอ็คทีฟได้พร้อมกัน
คุณยังอาจต้องการให้งานเหล่านี้ใช้ระดับความสำคัญที่ต่ำกว่างานแบ็คชิ่งอื่น.

งานข้ามคืนที่ไม่สะดวกที่จะรันในระหว่างชั่วโมงการทำงานปกติ

ตัวอย่างเช่น, การรันคำสั่ง Reorganize Physical File Member (RGZPFM) บนไฟล์ฐานข้อมูลขนาดใหญ่ต้องใช้การล็อกเฉพาะบนไฟล์. นี่หมายความว่า ผู้ใช้คนอื่นจะไม่สามารถเข้าใช้งานไฟล์ในขณะที่ทำการดำเนินการนี้. นอกจากนี้, การดำเนินการนี้อาจใช้เวลานาน. เป็นการดีกว่าที่จะวางงานนี้ บนคิวงานสำหรับงานซึ่งรันในระหว่างนอกชั่วโมงทำงาน.

งานที่มีระดับความสำคัญสูง

คุณอาจต้องการมีคิวงานที่จะส่งงานที่มีระดับความสำคัญสูงทั้งหมดไปที่นั่น. จากนั้น คุณสามารถแน่ใจได้ว่างานจะเสร็จสมบูรณ์อย่างรวดเร็วและไม่ถูกหน่วง โดยงานที่มีระดับความสำคัญต่ำ.

งานที่ถูกส่งไปยังข้อกำหนดตรีซอร์สเฉพาะ เช่น ดิสเก็ต หรือเทป

คิวงานประเภทนี้ต้องการพารามิเตอร์ MAXACT เป็น 1 ใน job queue entry ของ subsystem description เพื่อให้ในแต่ละครั้งมีงานที่ใช้รีซอร์สเพียงหนึ่งงานเท่านั้น.

ตัวอย่างเช่น, ถ้าใช้เทปสำหรับงานหลายงาน, งานทั้งหมดที่ใช้เทปจะถูกวางบนคิวงานเดียว. จากนั้น ในแต่ละครั้งจะมีการ เลือกงานหนึ่งงานจากคิวงาน. สิ่งนี้ทำให้มั่นใจได้ว่า ไม่มีงานสองงานแย่งชิง อุปกรณ์เดียวกันในเวลาเดียวกัน. ถ้าเกิดสภาพเช่นนี้, งานใดงานหนึ่งจะจบลง ด้วยข้อผิดพลาดการจัดสรร.

หมายเหตุ: เทปเอาต์พุตไม่สามารถสพูลได้.

งานโปรแกรมเมอร์

คุณอาจต้องการมีคิวงานหนึ่งคิวที่จัดการกับงานโปรแกรมเมอร์หรือชนิดของงานที่สามารถพักได้ในขณะที่กำลังรันงานการผลิต.

การรันแบบเป็นลำดับของชุดงาน

คุณอาจมีแอ็พพลิเคชันซึ่งงานหนึ่งงานขึ้นอยู่กับความสมบูรณ์ของงานอื่น. ถ้าคุณวางแผนเหล่านี้บนคิวงานที่เลือกและรันงานหนึ่งงานในแต่ละครั้ง, ลักษณะเช่นนี้ทำให้มั่นใจได้ว่าการรันงานเหล่านี้จะเป็นไปตามลำดับ.

ถ้างานต้องการการควบคุมเฉพาะของไฟล์, คุณอาจต้องการวางแผนบนคิวงานเมื่อคิวเป็นเพียงคิวเดียวที่แอ็คทีฟบนเซิร์ฟเวอร์, เช่นในระหว่างช่วง กลางคืนหรือวันหยุด.

ถ้าคุณใช้คิวงานหลายคิว, คุณจะพบว่า การควบคุมคิวงานหลากหลายคิว เป็นสิ่งที่ต้องพิจารณาเป็นอันดับแรก. โดยปกติ คุณจะต้องควบคุม:

- จำนวนคิวงานที่มีอยู่
- จำนวนคิวงานที่แอ็คทีฟในระบบย่อยในเวลาเดียวกัน
- จำนวนงานแอ็คทีฟที่สามารถเลือกจากคิวงานในเวลาหนึ่ง
- จำนวนงานที่สามารถแอ็คทีฟในระบบย่อยในเวลาหนึ่ง

วิธีการนำงานมาจากหลายคิวงาน

ระบบย่อยประมวลผลงานจากคิวงานตามหมายเลขลำดับ. เนื่องจากระบบย่อยสามารถมีมากกว่าหนึ่ง job queue entry ดังนั้นจึงสามารถจัดสรร มากกว่าหนึ่งคิวงานได้.

จำนวนสูงสุดของงานจากคิวถูกระบุโดยพารามิเตอร์ Maximum active jobs MAXACT บนคำสั่ง Add Job Queue Entry (ADDJOBQE) หรือคำสั่ง Change Job Queue Entry (CHGJOBQE). นอกจากนี้คุณยังสามารถควบคุมจำนวนงานของแต่ละระดับความสำคัญที่สามารถ แอ็คทีฟได้โดยใช้พารามิเตอร์ Maximum active priority MAXACTx. ตัวอย่างเช่น, ถ้า MAXACT=10, MAXACT5=2, และมีงานบนคิวงานที่มีระดับความสำคัญ 5 อยู่สามงาน, จะมีเพียงสองงานเท่านั้นที่สามารถแอ็คทีฟได้ในขณะนี้.

ระบบย่อยจะประมวลผลงานจากคิวงานที่มีหมายเลขลำดับน้อยที่สุดก่อน. เมื่องานทั้งหมดที่อยู่บนคิวงานได้ถูกประมวลผลแล้ว, หรือเมื่อใช้งานถึงจำนวนสูงสุดของงานจากคิว, ระบบย่อยจะประมวลผล งานจากคิวที่มีหมายเลขลำดับสูงกว่าถัดไป.

ลำดับดำเนินต่อไปจนกว่าระบบย่อยได้ประมวลผล job queue entries ที่พร้อมใช้งานทั้งหมด หรือจนกว่าระบบย่อยทำงานถึงขีดจำกัดของงานที่สามารถรันหรือรอในระบบย่อย. จำนวนของงานที่สามารถรันหรือรอได้ถูกกำหนด โดยพารามิเตอร์ Maximum active jobs (MAXACT) ใน subsystem description. ในบางกรณี ลำดับอาจถูกอินเทอร์รัปต์ เมื่องานจบหรือถูกถ่ายโอน. การสร้าง, การพัก, และการรีลีสคิวงาน สามารถเปลี่ยนลำดับของคิวงานที่ประมวลผลได้ด้วย.

การรักษาความปลอดภัยคิวงาน

คุณสามารถรักษาระดับของการรักษาความปลอดภัยที่คิวงานของคุณ โดยอนุญาตเฉพาะบางคน (โปรไฟล์ผู้ใช้) ให้สามารถเข้าใช้คิวงานนั้น. โดยทั่วไป, มีสามวิธี ที่ผู้ใช้สามารถได้สิทธิในการควบคุมคิวงาน (เช่น, พักหรือรีลีสคิวงาน).

- ผู้ใช้ได้รับการกำหนดสิทธิควบคุมการสพูล (SPCAUT(*SPLCTL)) ในโปรไฟล์ผู้ใช้.
- ผู้ใช้ได้รับการกำหนดสิทธิในการควบคุมงาน (SPCAUT(*JOBCTL)) ในโปรไฟล์ผู้ใช้ และคิวงานสามารถถูกควบคุมโดยผู้ควบคุมเครื่อง (OPRCTL(*YES)).

- ผู้ใช้มีสิทธิ์อับเจกต์ที่ต้องการที่คิวงาน. สิทธิ์อับเจกต์ ที่ต้องการถูกระบุโดยพารามิเตอร์ **AUTCHK** บนคำสั่ง **CRTJOBQ**. ค่าของ *OWNER บ่งชี้ว่า เฉพาะเจ้าของของคิวงานเท่านั้นที่ได้รับสิทธิ์ผ่านทางสิทธิ์อับเจกต์สำหรับ คิวงาน. ค่าของ *DTAAUT บ่งชี้ว่าผู้ใช้ที่มีสิทธิ์ *CHANGE สำหรับคิวงานจะได้รับสิทธิ์ในการควบคุมคิวงาน.

หมายเหตุ: สิทธิเฉพาะที่ต้องการ สำหรับ *DTAAUT คือสิทธิ์ *READ, *ADD, และ *DLT ข้อมูล.

วิธีการให้สิทธิ์สามวิธีเหล่านี้ใช้เฉพาะกับคิวงาน, ไม่ได้ใช้กับ งานบนคิวงาน. กฎสิทธิ์ปกติสำหรับการควบคุมงาน ใช้กับงานที่อยู่บนคิวงานและงานที่กำลังรันอยู่ในปัจจุบัน.

เอาต์พุตคิว

เอาต์พุตคิวคือพื้นที่ที่ printer output files (หรือเรียกว่าสพูลไฟล์) รอการถูกประมวลผลและถูกส่งไปยังพรินเตอร์. พรินเตอร์เอาต์พุตจะถูกสร้างขึ้นทั้งโดยระบบหรือโดยผู้ใช้ที่ใช้ไฟล์พิมพ์ อย่างไม่อย่างหนึ่ง.

ไฟล์พิมพ์คล้ายกับแฟ้มเพลต หรือต้นแบบ ที่มีการเซตค่าดีฟอลต์สำหรับแอตทริบิวต์ของพรินเตอร์เอาต์พุต ไว้ให้แล้ว. และเป็นจุดเริ่มต้นของช่วงชีวิตพรินเตอร์เอาต์พุต.

ไฟล์พิมพ์มีแอตทริบิวต์เอาต์พุตคิว (OUTQ) และอุปกรณ์การพิมพ์ (DEV), ซึ่งบ่งชี้ว่าพรินเตอร์เอาต์พุตจะถูกกำหนดทิศทางอย่างไร. โดยปกติ การตั้งค่า ดีฟอลต์คือ *JOB, ซึ่งหมายความว่าแอตทริบิวต์งานของเอาต์พุตคิว และอุปกรณ์การพิมพ์เป็นตัวกำหนดทิศทางของพรินเตอร์เอาต์พุต. แอตทริบิวต์งานของเอาต์พุตคิวและค่าอุปกรณ์การพิมพ์มาจากข้อมูลที่ได้รับจากงานที่ถูกสร้างขึ้น. ข้อมูลนี้มาจากข้อมูลที่ได้อาจมาจาก โปรไฟล์ผู้ใช้ที่งานกำลังรันอยู่ภายใต้, รายละเอียดของงาน, workstation device description, และค่ากำหนดของระบบ Printer device description (QPRTDEV).

เมื่อพรินเตอร์เอาต์พุตพร้อมที่จะถูกสร้าง, ระบบจะตรวจสอบไฟล์พิมพ์และแอตทริบิวต์งาน (ในลำดับนี้) เพื่อดูว่าเอาต์พุตคิวใดจะประมวลผลพรินเตอร์เอาต์พุตและอุปกรณ์การพิมพ์ใดที่ระบบจะใช้งาน. คุณสามารถเปลี่ยนพารามิเตอร์ของเอาต์พุตคิว (OUTQ) และอุปกรณ์การพิมพ์ (DEV) ขณะที่งานถูกส่งหรือขณะรันงานเพื่อข้ามการประมวลผลแบบขยาย. ยกตัวอย่าง, ผู้ใช้สามารถตั้งค่าเอาต์พุตคิวของไฟล์พิมพ์ให้เป็นคิวเฉพาะได้ และตั้งค่าอุปกรณ์การพิมพ์เป็นเครื่องพิมพ์ที่ต้องการในไฟล์พิมพ์เมื่อเริ่มต้นทำงานเพื่อให้การเปลี่ยนแปลงมีผลในทันที. ในการทำเช่นนั้น, พรินเตอร์เอาต์พุตไม่ต้องเข้าไปยังแอตทริบิวต์งานเพื่อค้นหาเอาต์พุตคิวและ อุปกรณ์การพิมพ์ที่จะใช้งาน. หากไม่พบเอาต์พุตคิวที่ระบุ, พรินเตอร์เอาต์พุตจะถูกส่งให้ไปที่ QGPL/QPRINT. สำหรับข้อมูลเพิ่มเติมเกี่ยวกับวิธีการสร้างพรินเตอร์เอาต์พุต, ให้อ่านที่ 1 ของคู่มือ Printer Device Programming.

Printer output files คือไฟล์ที่มีข้อมูลซึ่งกำลังรอที่จะพิมพ์หรือถูกประมวลผล. printer output file มีแอตทริบิวต์ที่สำคัญซึ่งกำหนดตำแหน่งของพรินเตอร์เอาต์พุตบนคิวซึ่งสัมพันธ์กับพรินเตอร์เอาต์พุตอื่น. ตำแหน่งจะถูกกำหนดโดยแอตทริบิวต์ระดับความสำคัญ, สถานะ, และตารางเวลา.

เอาต์พุตคิว

เอาต์พุตคิว คืออับเจกต์ที่ประกอบด้วยรายการของ printer output files ซึ่งจะถูกบันทึกไปที่อุปกรณ์เอาต์พุต. เอาต์พุตคิวมีแอตทริบิวต์ที่สำคัญซึ่งกำหนดลำดับที่พรินเตอร์เอาต์พุตจะถูกประมวลผลและสิทธิ์ที่ต้องใช้ในการทำการเปลี่ยนแปลงที่ printer output file.

ระดับความสำคัญ

พรินเตอร์เอาต์พุตที่กำลังรอการประมวลผลจะถูกย้ายไปที่เอาต์พุตคิวตามระดับความสำคัญของเอาต์พุต (ตั้งแต่ 1-9 โดย 1 คือระดับความสำคัญสูงสุด).

สถานะ สถานะปัจจุบันของพรินเตอร์เอาต์พุต. คุณสามารถดูสถานะนี้จากหน้า General ของหน้าต่าง Output properties.

ตารางเวลา

แอ็ตทริบิวต์ตารางเวลาจะบอกว่าเมื่อใดที่ไฟล์ควรเริ่มการพิมพ์แบบฟิสิกส์ข้อมูลเอาต์พุต.

พิมพ์ทันที

พิมพ์ในทันที, แม้ว่า printer output file จะยังไม่ถูกปิด.

การจบไฟล์ (ดีฟอลต์)

การพิมพ์เริ่มในทันทีที่ printer output file ถูกปิด.

การจบงาน

การพิมพ์จะเริ่มต้นเมื่องานจบลง.

หลังจากที่ printer output file พร้อมที่จะพิมพ์, งาน writer, งาน ที่ประมวลผลพรินเตอร์เอาต์พุตจากเอาต์พุตคิวไปที่อุปกรณ์การพิมพ์, จะนำข้อมูลมาจาก printer output file และส่งไปที่พรินเตอร์ที่กำหนด .

แอ็ตทริบิวต์ของเอาต์พุตคิว

เอาต์พุตคิวควบคุมวิธีการประมวลผลไฟล์พรินเตอร์เอาต์พุต (เรียกอีกอย่างว่าสพูลไฟล์) และผู้มีสิทธิปฏิบัติการบนเอาต์พุตคิวและพรินเตอร์เอาต์พุตที่เกี่ยวข้อง.

เนื่องจากข้อมูลส่วนใหญ่ที่คุณพิมพ์บนระบบ iSeries จะถูกสร้างเป็นพรินเตอร์เอาต์พุต, การรักษาความปลอดภัยจึงเป็นสิ่งจำเป็น เพื่อป้องกันไม่ให้ผู้ใช้ที่ไม่ได้รับอนุญาตเข้าถึงข้อมูลที่เป็นความลับหรือสำคัญ. สิทธิการตรวจสอบ, การให้สิทธิในข้อมูล, การควบคุมของผู้ควบคุมเครื่อง, การควบคุมสพูล, หรือการเป็นเจ้าของ ช่วยให้คุณสามารถเข้าใช้งานและทำการเปลี่ยนแปลงเอาต์พุตคิวหรือ printer output file. คุณต้องใช้สิทธิอย่างใดอย่างหนึ่งต่อไปนี้เพื่อปฏิบัติ action ใดๆ บนเอาต์พุตคิว หรือพรินเตอร์เอาต์พุต:

สิทธิการตรวจสอบ

คุณต้องเป็นเจ้าของคิวหรือมีสิทธิในข้อมูล.

การแสดงผลข้อมูล

เมื่อสิทธินี้ถูกเซตเป็น *YES, สิทธิจะทำให้คุณสามารถดำเนินการต่างๆ เช่น การดู, การย้าย, การส่งเอาต์พุตไปยังระบบอื่น, และการทำสำเนาพรินเตอร์เอาต์พุต.

การควบคุมของผู้ควบคุมเครื่อง

หากแอ็ตทริบิวต์นี้ถูกเซตเป็น *YES, ผู้ใช้ที่มีสิทธิพิเศษ *JOBCTL จะได้รับสิทธิให้ดำเนินการต่างๆ เช่น การพัก, การรีลีส, และการลบพรินเตอร์เอาต์พุตจากเอาต์พุตคิว. และอนุญาตให้มีการดำเนินการอื่นๆ กับพรินเตอร์เอาต์พุต, เอาต์พุตคิว, และ writers โดยมีการบันทึกไว้ในคู่มือ iSeries Security Reference.

การควบคุมสพูล

อนุญาตให้ผู้ใช้ดำเนินการต่างๆ บนพรินเตอร์เอาต์พุต. ผู้ใช้ต้องมีสิทธิใช้งาน *EXECUTE กับไลบรารีที่มีเอาต์พุตคิวอยู่เพื่อดำเนินการต่างๆ บนเอาต์พุตคิว.

เจ้าของ

อนุญาตให้ผู้ใช้ที่เป็นเจ้าของเอาต์พุตคิวเปลี่ยนหรือลบพรินเตอร์เอาต์พุต.

หมายเหตุ: สิทธิในการใช้งานดีฟอลต์สำหรับเอาต์พุตคิวคือสิทธิพัลลิก *USE. สิทธิในการใช้งาน Display Data ถูกกำหนดเป็น *NO (หมายถึงไม่ใช่ทุกคนที่สามารถดูพรินเตอร์เอาต์พุตนี้ได้). สิทธิในการตรวจสอบคือ *OWNER (ดังนั้นเจ้าของเอาต์พุตคิวจึงสามารถจัดการกับพรินเตอร์เอาต์พุตได้). การควบคุมของผู้ควบคุมเครื่องถูกกำหนดเป็น *YES (หมายถึงผู้ใช้ที่มีสิทธิ *JOBCTL สามารถพัก, รีลีส, และลบพรินเตอร์เอาต์พุตได้).

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับสิทธิ์ที่จำเป็นสำหรับการทำงานกับเอาต์พุตคิว, ให้ดูภาคผนวก D ใน Security Reference Manual.

ลำดับของไฟล์

แอ็ททริบิวต์ Order of files on the queue (SEQ) กำหนดว่าพริ้นเตอร์เอาต์พุตจะถูกเอาต์พุตคิวประมวลผลอย่างไร.

แอ็ททริบิวต์นี้มีสองค่า:

- *FIFO: คิวคือ เข้าก่อนออกก่อนภายในระดับความสำคัญของแต่ละไฟล์. นั่นคือ, สพูลไฟล์ใหม่จะถูกวางหลังจาก entries อื่นทั้งหมดบนคิว ของระดับความสำคัญเดียวกัน.
- *JOBNBR : queue entries สำหรับสพูลไฟล์จะถูกเรียงลำดับตามลำดับระดับความสำคัญ โดยใช้หมายเลขงาน (โดยแท้จริงแล้ว, คือวันที่และเวลาการใช้งานที่ถูกป้อน เข้าสู่ระบบ) ของงานที่สร้างสพูลไฟล์.

หมายเหตุ: คุณสามารถเปลี่ยนได้เฉพาะลำดับเอาต์พุตคิวของไฟล์แอ็ททริบิวต์ เมื่อไม่มี printer output files อยู่บนคิว.

สพูลไฟล์

การสพูลคือฟังก์ชันระบบที่บันทึกข้อมูลสำหรับการประมวลผลหรือ การพิมพ์ในภายหลัง. ข้อมูลนี้ถูกจัดเก็บในสพูลไฟล์. สพูลไฟล์ทำงานใน ลักษณะคล้ายกับเทปไฟล์หรือไฟล์อุปกรณ์อื่น. สพูลไฟล์ช่วยให้คุณจัดการข้อมูลของคุณซึ่งถูกกำหนดเป้าหมายสำหรับอุปกรณ์ที่ต่อพ่วงภายนอก เช่น พริ้นเตอร์.

ฟังก์ชันการสพูลช่วยผู้ใช้เซิร์ฟเวอร์ให้สามารถจัดการการดำเนินการ อินพุตและเอาต์พุตได้อย่างมีประสิทธิภาพมากขึ้น. เซิร์ฟเวอร์สนับสนุนการสพูลสองชนิด, การสพูลเอาต์พุตและการสพูลอินพุต. การสพูลเอาต์พุตสามารถใช้สำหรับอุปกรณ์การพิมพ์. การสพูลอินพุตใช้กับอินพุตไฟล์ฐานข้อมูล.

การสพูลเอาต์พุต:

สามารถใช้การสพูลเอาต์พุตสำหรับทั้งพริ้นเตอร์และอุปกรณ์ดิสเก็ต. การสพูลเอาต์พุตจะส่งเอาต์พุตงานไปที่หน่วยเก็บของดิสก์ แทนที่จะส่งตรงไปที่พริ้นเตอร์หรืออุปกรณ์ดิสเก็ตเอาต์พุต. การสพูลเอาต์พุตช่วยให้งานที่สร้างเอาต์พุต สามารถประมวลผลต่อไปได้โดยไม่ต้องพิจารณาถึงความเร็ว หรือสภาพพร้อมใช้งานของอุปกรณ์เอาต์พุต.

นอกจากนี้, การสพูลเอาต์พุตยังช่วยให้เซิร์ฟเวอร์สามารถสร้างเอาต์พุตบนหลาย อุปกรณ์เอาต์พุต, เช่น พริ้นเตอร์และอุปกรณ์ดิสเก็ต, ได้อย่างมีประสิทธิภาพ. ฟังก์ชันทำสิ่งนี้โดยการส่งเอาต์พุตของงานที่มีปลายทางเป็นพริ้นเตอร์ไปที่หน่วยเก็บของดิสก์. กระบวนการนี้ช่วยแก้ไขปัญหาคอขวดงานที่อาจเกิดขึ้นได้จาก สภาพพร้อมใช้งานหรือความเร็วของอุปกรณ์เอาต์พุต.

องค์ประกอบหลักของการสพูลเอาต์พุตมีดังนี้:

- **Device description:** description ของอุปกรณ์การพิมพ์.
- **สพูลไฟล์:** ไฟล์ที่มี spooled output records ซึ่งจะ ถูกประมวลผลบนอุปกรณ์เอาต์พุต.
- **เอาต์พุตคิว:** รายการที่จัดลำดับของสพูลไฟล์.
- **Writer:** โปรแกรมที่ส่งไฟล์จากเอาต์พุตคิวไปที่อุปกรณ์.
- **แอ็พพลิเคชันโปรแกรม:** โปรแกรมภาษาชั้นสูงที่สร้างสพูลไฟล์ โดยใช้ไฟล์อุปกรณ์ที่มีแอ็ททริบิวต์การสพูลถูกระบุเป็น SPOOL(*YES).
- **ไฟล์อุปกรณ์:** description ของรูปแบบของเอาต์พุต, และรายการของ แอ็ททริบิวต์ที่อธิบายว่า เซิร์ฟเวอร์ควรจะประมวลผลสพูลไฟล์อย่างไร.

ฟังก์ชันการสพูลเอาต์พุตถูกดำเนินการโดยเซิร์ฟเวอร์โดยไม่ต้อง การดำเนินการพิเศษใดๆ โดยโปรแกรมที่สร้างเอาต์พุต. เมื่อไฟล์อุปกรณ์ถูกเปิด โดยโปรแกรม, ระบบปฏิบัติการจะกำหนดว่าเอาต์พุตจะถูกสพูลหรือไม่. เมื่อไฟล์พริเตอร์ที่ระบุการสพูลถูกเปิด, สพูลไฟล์ที่มี เอาต์พุตของโปรแกรมจะถูกวางบนเอาต์พุตคิวที่เหมาะสม ในเซิร์ฟเวอร์.

สพูลไฟล์สามารถพร้อมใช้งานสำหรับการพิมพ์เมื่อไฟล์พริเตอร์ ถูกเปิด, เมื่อไฟล์พริเตอร์ถูกปิด, หรือเมื่อจบงาน. เครื่องเขียนพริเตอร์ ถูกเริ่มต้นในระบบย่อยการสพูลเพื่อส่งเร็กคอร์ดไปที่พริเตอร์. สพูลไฟล์จะถูกเลือกจากเอาต์พุตคิว.

Spooling device descriptions

Device descriptions ต้องถูกสร้างขึ้นสำหรับแต่ละพริเตอร์และอุปกรณ์ดิสเก็ต เพื่อกำหนดอุปกรณ์นั้น ไปที่เซิร์ฟเวอร์.

Printer device descriptions ถูกสร้างขึ้นโดยใช้คำสั่ง Create Device Description for Printer (CRTDEVPRN) ; diskette device descriptions ถูกสร้างขึ้นโดยใช้คำสั่ง Create Device Description for Diskette (CRTDEVDKT) .

การเปลี่ยนทิศทางไฟล์ของสพูลไฟล์

การเปลี่ยนทิศทางไฟล์ จะเกิดขึ้นเมื่อสพูลไฟล์ถูกส่งไปที่อุปกรณ์เอาต์พุตที่ไม่ใช่อุปกรณ์เอาต์พุตที่ตั้งใจไว้ตั้งแต่แรก. การเปลี่ยนทิศทางไฟล์อาจเกี่ยวข้องกับอุปกรณ์ที่ประมวลผลสื่อบันทึกต่างๆ (เช่น พริเตอร์เอาต์พุตที่ส่งไปที่อุปกรณ์ดิสเก็ต) หรืออุปกรณ์ที่ประมวลผลสื่อบันทึกชนิดเดียวกันแต่มีชนิดของอุปกรณ์แตกต่างกัน (เช่น เอาต์พุต 5219 Printer ที่ส่งไปที่ 4224 Printer).

ขึ้นอยู่กับอุปกรณ์เอาต์พุตใหม่สำหรับสพูลไฟล์, ไฟล์อาจสามารถถูกประมวลผล เหมือนกับที่เคยใช้บนอุปกรณ์ที่ระบุในครั้งแรก. อย่างไรก็ตาม, อุปกรณ์ที่แตกต่างกันมักทำให้เอาต์พุตถูกจัดรูปแบบแตกต่างกัน. ในกรณีนี้, เซิร์ฟเวอร์จะส่งข้อความสอบถามไปที่ message queue ของ writer เพื่อแจ้งให้คุณทราบถึงสถานการณ์และคุณสามารถระบุได้ว่า คุณต้องการพิมพ์ต่อไปหรือไม่.

เอาต์พุตคิวและสพูลไฟล์:

การประมวลผลงานแบ็คซ์และงานแบบโต้ตอบอาจทำให้เกิด spooled output records ที่จะถูกประมวลผลบนอุปกรณ์เอาต์พุต, เช่น พริเตอร์หรือ ดิสเก็ตไดรฟ์. เอาต์พุตเร็กคอร์ดเหล่านี้ถูกจัดเก็บในสพูลไฟล์จนกว่าจะสามารถประมวลผลได้. อาจมีได้หลายสพูลไฟล์สำหรับหนึ่งงาน .

เมื่อสพูลไฟล์ถูกสร้างขึ้น, ไฟล์จะถูกวางบนเอาต์พุตคิว. แต่ละเอาต์พุตคิวมีรายการที่จัดลำดับของสพูลไฟล์. งานอาจมีสพูลไฟล์ บนหนึ่งหรือหลายเอาต์พุตคิว. สพูลไฟล์ทั้งหมดบนเอาต์พุตคิวหนึ่ง ควรมีชุดของแอ็ททริบิวต์เอาต์พุตที่ใช้ร่วมกัน, เช่น อุปกรณ์, form type, และจำนวนบรรทัดต่อนิว. การใช้แอ็ททริบิวต์ร่วมกันบนเอาต์พุตคิว ช่วยลดจำนวนของ intervention ที่ต้องการและเพิ่มปริมาณงานของอุปกรณ์.

ข้อมูลต่อไปนี้เป็นรายการของบางพารามิเตอร์บนคำสั่ง Create Output Queue (CRTOUTQ) และสิ่งที่พารามิเตอร์ระบุ:

- **MAXPAGES:** ระบุขนาดสพูลไฟล์สูงสุดในหน้า ที่อนุญาตให้พิมพ์ได้ระหว่างเวลาเริ่มต้นและเวลาสิ้นสุด ของวัน.
- **AUTOSTRWTR:** ระบุจำนวนของ writers ที่จะถูก เริ่มโดยอัตโนมัติที่เอาต์พุตคิวนี้.
- **DSPDTA:** ผู้ใช้ที่ไม่มีสิทธิพิเศษใดๆ แต่มีสิทธิ *USE ที่เอาต์พุตคิวสามารถแสดง, ก๊อปปี้, หรือส่ง เนื้อหาของสพูลไฟล์ที่ไม่ใช่ของตนเองได้. ถ้ามีการระบุ *OWNER สำหรับ DSPDTA, นั้นหมายความว่าเฉพาะเจ้าของของไฟล์หรือผู้ใช้ที่มีสิทธิพิเศษ *SPLCTL เท่านั้นที่สามารถแสดง, ก๊อปปี้, หรือส่งไฟล์ได้.
- **JOBSEP:** จำนวน, ถ้ามี, หน้าตัวแบ่งงานที่จะ พิมพ์ระหว่างเอาต์พุตของแต่ละงาน เมื่อพิมพ์เอาต์พุต.

- **DTAQ:** คิวข้อมูลที่เชื่อมโยงกับเอาต์พุตคิวนี้. ถ้าระบุ, entry จะถูกส่งไปที่คิวข้อมูลในทุกเมื่อที่สพูลไฟล์ไปที่ Ready Status บนคิว.
- **OPRCTL:** การมีหรือไม่มีสิทธิในการควบคุมงานของผู้ใช้เป็นตัว กำหนดความสามารถในการควบคุมเอาต์พุตคิว (เช่น, ผู้ใช้สามารถพักเอาต์พุตคิวได้หรือไม่).
- **SEQ:** ควบคุมลำดับซึ่งสพูลไฟล์จะถูก เรียงลำดับบนเอาต์พุตคิว.
- **AUTCHK:** ระบุชนิดของสิทธิที่เอาต์พุตคิว ที่จะเปิดทางให้ผู้ใช้สามารถควบคุมสพูลไฟล์บนเอาต์พุตคิว (เช่น, เปิดทางให้ผู้ใช้พักสพูลไฟล์บนเอาต์พุตคิว).
- **AUT:** สิทธิพับลิก. ระบุว่าผู้ใช้สามารถควบคุมอะไร บนตัวเอาต์พุตคิวเอง.
- **TEXT:** คำอธิบายข้อความ. ข้อความที่ยาวได้ถึง 50 อักขระ ซึ่งอธิบายเอาต์พุตคิว.

Default server output queues:

เซิร์ฟเวอร์ถูกจัดส่งมาพร้อมกับค่าดีฟอลต์ของคำสั่งที่จะใช้ ดีฟอลต์เอาต์พุตคิวสำหรับเซิร์ฟเวอร์พริ้นเตอร์เป็นดีฟอลต์เอาต์พุตคิวสำหรับ เอาต์พุตที่สพูลทั้งหมด. เซิร์ฟเวอร์พริ้นเตอร์ถูกกำหนดโดยค่าเซิร์ฟเวอร์ QPRTDEV.

เมื่อสพูลไฟล์ถูกสร้างขึ้นโดยการเปิดไฟล์อุปกรณ์และไม่พบ เอาต์พุตคิวที่ระบุสำหรับไฟล์, เซิร์ฟเวอร์จะพยายามวาง สพูลไฟล์บนเอาต์พุตคิว QPRINT ในไลบรารี QGPL. ถ้ามีบาง เหตุผลที่ทำให้ไม่สามารถวางสพูลไฟล์บนเอาต์พุตคิว QPRINT, จะมีการส่ง ข้อความแสดงความผิดพลาดและเอาต์พุตจะไม่ถูกสพูล.

เอาต์พุตคิวที่จัดส่งมาพร้อมกับเซิร์ฟเวอร์มีดังต่อไปนี้:

- **QDKT:** Default diskette output queue
- **QPRINT:** Default printer output queue
- **QPRINTS:** Printer output queue สำหรับแบบฟอร์มพิเศษ
- **QPRINT2:** Printer output queue สำหรับกระดาษ 2 ส่วน

Spooling writers:

writer คือโปรแกรม i5/OS ที่จะนำสพูลไฟล์มาจากเอาต์พุตคิว และจัดทำสพูลไฟล์บนอุปกรณ์เอาต์พุต. สพูลไฟล์ที่ถูกวางบนเอาต์พุตคิวแล้ว จะยังคงถูกจัดเก็บในเซิร์ฟเวอร์ จนกว่า writer จะถูกเริ่มที่เอาต์พุตคิว.

writer นำสพูลไฟล์มาครั้งละหนึ่งไฟล์จากเอาต์พุตคิว, ตาม ระดับความสำคัญของสพูลไฟล์. writer จะประมวลผลสพูลไฟล์ เฉพาะถ้า entry ของสพูลไฟล์บนเอาต์พุตคิวระบุว่า สพูลไฟล์มีสถานะพร้อมใช้งาน (RDY). คุณสามารถแสดงสถานะ ของสพูลไฟล์ได้โดยใช้คำสั่ง Work with Output Queue (WRKOUTQ).

ถ้าสพูลไฟล์มีสถานะพร้อมใช้งาน, writer จะนำ entry มาจาก เอาต์พุตคิวและพิมพ์งานที่ระบุ หรือตัวแบ่งไฟล์ หรือทั้งสอง อย่าง, ตามด้วยข้อมูลเอาต์พุตในไฟล์. ถ้าสพูลไฟล์ไม่ได้มีสถานะพร้อมใช้งาน, writer จะปล่อย entry ไว้บนเอาต์พุตคิวและไป ที่ entry ถัดไป. ในกรณีส่วนใหญ่ writer จะประมวลผลสพูลไฟล์ต่อไป (มีงานและตัวแบ่งไฟล์ก่อน) จนกว่าไฟล์ทั้งหมดที่มีสถานะพร้อมใช้งาน ได้ถูกนำออกจากเอาต์พุตคิว.

พารามิเตอร์ **AUTOEND** บนคำสั่งเริ่ม writer เป็นตัวกำหนดว่า writer จะยังคงรอสพูลไฟล์ใหม่ให้พร้อมถูกบันทึก, จะจบลง หลังการประมวลผลหนึ่งไฟล์, หรือจะจบลงหลังจากที่สพูลไฟล์ทั้งหมด ที่มีสถานะพร้อมใช้งานได้ถูกนำออกจากเอาต์พุตคิว แล้ว.

ข้อสรุปของคำสั่ง spooling writer:

มีหลายคำสั่งที่คุณสามารถใช้ในการควบคุม spooling writers.

- Start Diskette Writer (STRDKTWTR): เริ่ม spooling writer ที่อุปกรณ์ดิสก์เก็ตที่ระบุ เพื่อประมวลผลสพูลไฟล์บนอุปกรณ์นั้น.
- Start Printer Writer (STRPRTWTR): เริ่ม spooling writer ที่อุปกรณ์การพิมพ์ที่ระบุ เพื่อประมวลผลสพูลไฟล์บนอุปกรณ์นั้น.
- Start Remote Writer (STRRMTWTR): เริ่ม spooling writer ที่จะส่งสพูลไฟล์จากเอาต์พุตคิวไปที่รีโมตเซิร์ฟเวอร์.
- Change Writer (CHGWTR): อนุญาตให้คุณเปลี่ยนแอตทริบิวต์ writer บางอย่าง, เช่น form type, จำนวนของหน้าตัวแบ่งไฟล์, หรือแอตทริบิวต์ เอาต์พุตคิว.
- Hold Writer (HLDWTR): หยุด writer ที่ตอนท้ายของ เร็กคอร์ด, ที่ตอนท้ายของสพูลไฟล์, หรือที่ตอนท้ายของหน้า.
- Release Writer (RLSWTR): รีลีส writer ที่ถูกพักไว้ก่อนหน้านี้สำหรับการประมวลผลเพิ่มเติม.
- End Writer (ENDWTR): จบ spooling writer และทำให้ อุปกรณ์เอาต์พุตที่เชื่อมโยงพร้อมใช้งานที่เซิร์ฟเวอร์.

หมายเหตุ: คุณสามารถกำหนดฟังก์ชันบางอย่างเพื่อให้การสนับสนุนการสพูลเพิ่มเติม. ตัวอย่างซอร์สและเอกสารคู่มือสำหรับคำสั่ง, ไฟล์, และโปรแกรมสำหรับ ฟังก์ชันเหล่านี้มีอยู่ในไลบรารี QUSRTOOL, ซึ่งเป็นส่วนที่เลือกติดตั้งได้ ของ i5/OS.

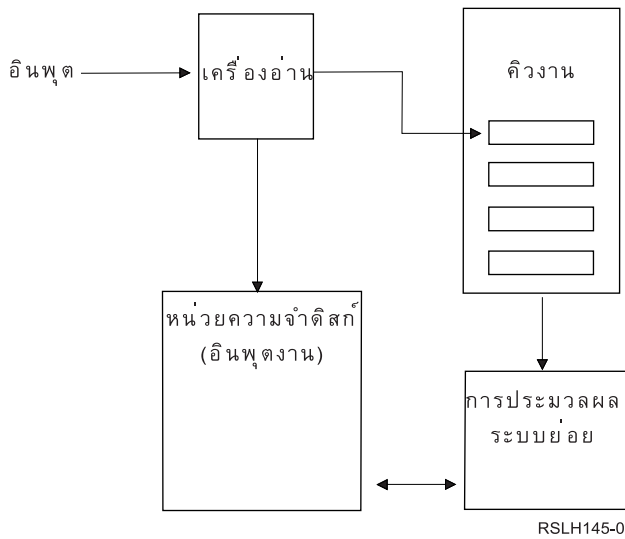
Input spooling:

Input spooling นำข้อมูลมาจากอุปกรณ์อินพุต, จัดเตรียม งานสำหรับการจัดตารางเวลา, และวาง entry ในคิวงาน. ด้วยการใช้ input spooling, โดยปกติสามารถช่วยให้รันไทม์ของงานลดลง, เพิ่มจำนวนของงานที่สามารถ รันในลำดับถัดไปได้, และทำให้ปริมาณงานของอุปกรณ์มากขึ้น.

องค์ประกอบหลักของ input spooling คือ:

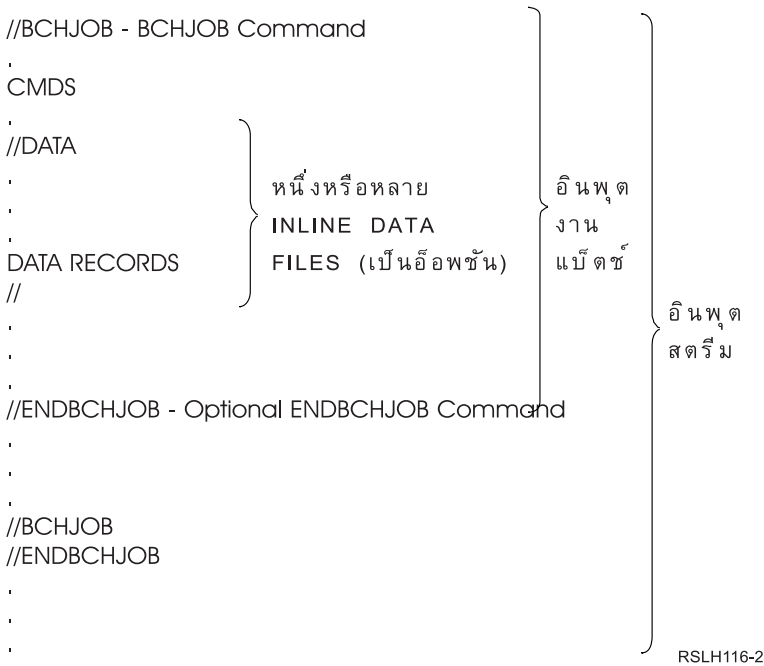
- **คิวงาน:** รายการที่จัดลำดับของงานแบ็ตช์ที่ส่งไปที่เซิร์ฟเวอร์ สำหรับการรันและเป็นที่ที่งานแบ็ตช์ถูกเลือกมาเพื่อรัน.
- **เครื่องอ่าน:** ฟังก์ชันที่นำงานมาจากอุปกรณ์อินพุตหรือไฟล์ฐานข้อมูล และวางงานบนคิวงาน.

เมื่อเครื่องอ่านอ่านงานแบ็ตช์จากอินพุตซอร์ส, คำสั่งใน input stream จะถูกจัดเก็บในเซิร์ฟเวอร์เป็นคำขอสำหรับงาน, ข้อมูลอินไลน์ ถูกสพูลเป็นไฟล์ข้อมูลอินไลน์, และ entry สำหรับงานถูกวางบน คิวงาน. ข้อมูลงานยังคงถูกจัดเก็บอยู่ในเซิร์ฟเวอร์ที่เครื่องอ่านวางไว้จนกว่า job entry จะถูกเลือกจากคิวงาน สำหรับการประมวลผลโดยระบบย่อย.



รูปที่ 2. Relationship of Input Spooling Elements

คุณสามารถใช้ฟังก์ชันเครื่องอ่านเพื่ออ่าน input stream จากดิสเก็ต หรือไฟล์ฐานข้อมูล.



รูปที่ 3. Typical Organization of an Input Stream

คิวงานที่งานถูกวางถูกระบุอยู่บนพารามิเตอร์ **JOBQ** บนคำสั่ง Batch Job **BCHJOB** , บนคำสั่ง Start Database Reader **STRDBRDR** , หรือในรายละเอียดของงาน. ถ้าพารามิเตอร์ **JOBQ** บนคำสั่ง **BCHJOB** เป็น:

- *RDR: คิวงานจะถูกเลือกจากพารามิเตอร์ **JOBQ** บนคำสั่ง Start database reader (**STRDBRDR**) .
- *JOBQ: คิวงานจะถูกเลือกจากพารามิเตอร์ **JOBQ** ในรายละเอียดของงาน.
- คิวงานเฉพาะ: จะใช้คิวที่ระบุ.

สำหรับงานที่มี input streams เล็ก, คุณอาจพัฒนาประสิทธิภาพของเซิร์ฟเวอร์ได้โดยการไม่ใช้ input spooling. คำสั่ง Submit Job (SBMJOB) จะอ่าน input stream และวางแผนบนคิวงานในระบบย่อยที่เหมาะสม, โดยการข้ามระบบย่อยการสพูลและการดำเนินการของเครื่องอ่าน.

ถ้างานของคุณต้องการ input stream ที่จะอ่านขนาดใหญ่, คุณควรใช้ input spooling (คำสั่ง Start Diskette Reader STRDKTRDR หรือ Start Database Reader STRDBRDR) เพื่อให้งานสามารถเป็น input independent เมื่องานถูกประมวลผลจริง.

ข้อสรุปของคำสั่งการอินพุตงาน:

สามารถใช้คำสั่งต่อไปนี้เมื่อส่งงานไปที่เซิร์ฟเวอร์. สามารถใช้คำสั่งการเริ่มเครื่องอ่านสำหรับการสพูลอินพุตงาน; คำสั่งการส่งงานจะไม่ใช้การสพูล. สำหรับคำอธิบายโดยละเอียดของคำสั่งเหล่านี้, ให้อ่านหัวข้อ CL ใน iSeries Information Center.

- Batch Job (BCHJOB): ทำเครื่องหมายจุดเริ่มต้นของงานใน batch input stream และกำหนดลักษณะการดำเนินการของงาน.
- Data (DATA): ทำเครื่องหมายจุดเริ่มต้นของไฟล์ข้อมูลออนไลน์.
- End Batch Job (ENDBCHJOB): ทำเครื่องหมายจุดสิ้นสุดของงานใน batch input stream.
- End Input (ENDINP): ทำเครื่องหมายจุดสิ้นสุดของ batch input stream.
- Submit Database Jobs (SBMDBJOB): อ่าน input stream จากไฟล์ฐานข้อมูลและวางแผนใน input stream บนคิวงานที่เหมาะสม.
- Submit Diskette Jobs (SBMDKTJOB): อ่าน input stream จากดิสเก็ตและวางแผนใน input stream บนคิวงานที่เหมาะสม.
- Start Database Reader (STRDBRDR): เริ่มต้นเครื่องอ่านเพื่ออ่าน input stream จากไฟล์ฐานข้อมูลและวางแผนใน input stream บนคิวงานที่เหมาะสม.
- Start Diskette Reader (STRDKTRDR): เริ่มต้นเครื่องอ่านเพื่ออ่าน input stream จากดิสเก็ตและวางแผนใน input stream บนคิวงานที่เหมาะสม.

การใช้ไฟล์ข้อมูลออนไลน์:

ไฟล์ข้อมูลออนไลน์คือไฟล์ข้อมูลทั้งหมดอยู่เป็นส่วนหนึ่งของงานแบ็ตช์ เมื่องานถูกอ่านโดยเครื่องอ่านหรือคำสั่งการส่งงาน. คุณใช้คำสั่ง SBMDBJOB หรือ STRDBRDR เพื่อจัดคิว CL batch stream (stream ของคำสั่ง CL ที่จะดำเนินการหรือรัน). CL batch stream นั้นสามารถรวมข้อมูลที่จะวางเข้าไปในไฟล์ "ชั่วคราว" (ไฟล์ออนไลน์). เมื่องานสิ้นสุดลง, ไฟล์ออนไลน์จะถูกลบออก.

ไฟล์ข้อมูลออนไลน์ถูกค้นในงานโดยคำสั่ง //DATA ที่ตอนต้นของไฟล์ และถูกค้นโดย end-of-data delimiter ที่ตอนท้ายของไฟล์.

end-of-data delimiter อาจเป็นสตริงอักขระที่ผู้ใช้กำหนด หรือค่าดีฟอลต์ของ // . // ต้องปรากฏในตำแหน่ง 1 และ 2. ถ้าข้อมูลของคุณ มี // ในตำแหน่ง 1 และ 2, คุณควรจะใช้เซตของอักขระที่ไม่ซ้ำกัน เช่น: // *** END OF DATA เพื่อระบุคำสั่งนี้เป็น end-of-data delimiter เฉพาะ, พารามิเตอร์ ENDCHAR บนคำสั่ง //DATA ควรเป็นโค้ดเช่น:

```
ENDCHAR( '// *** END OF DATA ' )
```

หมายเหตุ: ไฟล์ข้อมูลออนไลน์สามารถเข้าถึงได้เฉพาะในระหว่างขั้นตอนการเรดแรกของงานแบ็ตช์เท่านั้น. ถ้างานแบ็ตช์มีคำสั่ง Transfer Job (TFRJOB), Reroute Job (RRTJOB), หรือ Transfer Batch Job (TFRBCHJOB), ไฟล์ข้อมูลออนไลน์จะไม่สามารถเข้าถึงได้ในขั้นตอนการเรดที่ใหม่.

ไฟล์ข้อมูล อินไลน์สามารถเป็นแบบระบุชื่อหรือไม่ระบุชื่อ อย่างไม่อย่างหนึ่ง. สำหรับไฟล์ข้อมูลอินไลน์แบบไม่ระบุชื่อ, อาจมีการระบุ QINLINE เป็นชื่อไฟล์ในคำสั่ง//DATA หรือไม่มีการระบุชื่อ อย่างไม่อย่างหนึ่ง. สำหรับไฟล์ข้อมูลอินไลน์ที่ระบุชื่อ, จะมีการระบุชื่อไฟล์.

ไฟล์ ข้อมูลอินไลน์แบบระบุชื่อมีลักษณะดังต่อไปนี้:

- มีชื่อที่ไม่ซ้ำกันในงาน. ไฟล์ข้อมูลอินไลน์ต้องมีชื่อที่ไม่ซ้ำกัน.
- สามารถใช้ได้มากกว่าหนึ่งครั้งในงานหนึ่ง.
- ในแต่ละครั้งที่เปิดไฟล์, ไฟล์จะถูกจัดตำแหน่งอยู่ที่เร็กคอร์ดแรก.

ในการใช้ไฟล์ข้อมูลอินไลน์แบบระบุชื่อ, คุณต้องระบุชื่อไฟล์ในโปรแกรม หรือใช้คำสั่งแทนที่เพื่อเปลี่ยนชื่อไฟล์ที่ระบุ ในโปรแกรมเป็นชื่อของไฟล์ข้อมูลอินไลน์ อย่างไม่อย่างหนึ่ง. ไฟล์ต้องถูกเปิด สำหรับการอินพุตเท่านั้น.

ไฟล์ข้อมูลอินไลน์แบบไม่ระบุชื่อมีลักษณะดังต่อไปนี้:

- ชื่อของไฟล์คือ QINLINE. (ในงานแบ็คซ์, ไฟล์ข้อมูลอินไลน์แบบไม่ระบุชื่อทั้งหมดจะ ถูกตั้งชื่อเหมือนกัน.)
- สามารถใช้ได้ครั้งเดียวเท่านั้นในงานหนึ่ง.
- เมื่อมีไฟล์ข้อมูลอินไลน์แบบไม่ระบุชื่ออยู่มากกว่าหนึ่งไฟล์ในงาน, ไฟล์ต้องอยู่ใน input stream ในลำดับเดียวกับลำดับเมื่อไฟล์ถูกเปิด.

ถ้าต้องการใช้ไฟล์ข้อมูลอินไลน์แบบไม่ระบุชื่อ, ให้ทำอย่างใดอย่างหนึ่งดังนี้:

- ระบุ QINLINE ในโปรแกรม.
- ใช้คำสั่งการแทนที่ไฟล์เพื่อเปลี่ยนชื่อไฟล์ที่ระบุในโปรแกรม เป็น QINLINE.

ถ้าภาษาชั้นสูงของคุณต้องการชื่อไฟล์ที่ไม่ซ้ำกันในหนึ่งโปรแกรม, คุณสามารถใช้ QINLINE เป็นชื่อไฟล์ได้เพียงครั้งเดียวเท่านั้น. ถ้าคุณต้องการใช้ไฟล์ข้อมูลอินไลน์แบบไม่ระบุชื่อมากกว่าหนึ่งไฟล์, คุณสามารถใช้คำสั่งการแทนที่ไฟล์ ในโปรแกรมเพื่อระบุ QINLINE สำหรับไฟล์ข้อมูลอินไลน์แบบไม่ระบุชื่อเพิ่มเติม.

หมายเหตุ: ถ้าคุณรันคำสั่งแบบมีเงื่อนไขและประมวลผลไฟล์ข้อมูลอินไลน์แบบไม่ระบุชื่อ มากกว่าหนึ่งไฟล์, จะไม่สามารถคาดการณ์ผลลัพธ์ได้ถ้าใช้ไฟล์ข้อมูลอินไลน์แบบไม่ระบุชื่อที่ไม่ถูกต้อง.

ข้อควรพิจารณาในการเปิดไฟล์ข้อมูลอินไลน์:

ข้อควรพิจารณาสำหรับการเปิดไฟล์ข้อมูลอินไลน์มีดังต่อไปนี้:

- ความยาวเร็กคอร์ดระบุความยาวของอินพุตเร็กคอร์ด. (ความยาวเร็กคอร์ด เป็นสิ่งที่เลือกได้.) เมื่อความยาวเร็กคอร์ดเกินกว่าความยาวของข้อมูล, จะมีการส่ง ข้อความไปที่โปรแกรมของคุณ. ข้อมูลจะถูกเสริมด้วยพื้นที่เปล่า. เมื่อความยาวเร็กคอร์ด น้อยกว่าความยาวของข้อมูล, เร็กคอร์ดจะถูกตัดปลาย.
- เมื่อมีการระบุไฟล์ในโปรแกรม, เซิร์ฟเวอร์จะค้นหาไฟล์ตามไฟล์ข้อมูล อินไลน์ที่ระบุชื่อ ก่อนที่จะค้นหาไฟล์ในไลบรารี. ดังนั้น, ถ้าไฟล์ข้อมูลอินไลน์ที่ระบุชื่อมีชื่อเหมือนกับไฟล์ที่ไม่ใช่ไฟล์ข้อมูลอินไลน์, ไฟล์ข้อมูลอินไลน์จะถูกใช้เสมอ, แม้ว่าชื่อไฟล์ จะถูกต้องตามชื่อไลบรารี.
- ไฟล์ข้อมูลอินไลน์ที่ระบุชื่อสามารถถูกแบ่งใช้ระหว่างโปรแกรมต่างๆ ในงานเดียวกันได้โดยการระบุ SHARE(*YES) บนคำสั่งการสร้างไฟล์หรือการแทนที่ไฟล์. ตัวอย่างเช่น, ถ้าคำสั่งการแทนที่ไฟล์ระบุว่าเป็นไฟล์ที่มีชื่อว่า INPUT และ SHARE

(*YES) อยู่ในงานแบ็คอัพพร้อมกับไฟล์ข้อมูลออนไลน์ที่มีชื่อว่า INPUT, โปรแกรมใดๆ ที่รันในงานที่ระบุชื่อไฟล์ INPUT จะแบ่งใช้ไฟล์ข้อมูลออนไลน์ที่มีชื่อเหมือนกัน. ไฟล์ข้อมูลออนไลน์ที่ไม่ได้ระบุชื่อไม่สามารถถูกแบ่งใช้ระหว่างโปรแกรมต่างๆ ในงานเดียวกัน.

- เมื่อคุณใช้ไฟล์ข้อมูลออนไลน์, คุณควรตรวจสอบให้แน่ใจว่าได้ระบุชนิดของไฟล์ ที่ถูกต้องบนคำสั่ง //DATA. ตัวอย่างเช่น, ถ้าไฟล์จะถูกใช้เป็น ไฟล์ต้นฉบับ, ชนิดของไฟล์บนคำสั่ง //DATA ต้องเป็นต้นฉบับ.
- ไฟล์ข้อมูลออนไลน์ต้องถูกเปิดสำหรับอินพุตเท่านั้น.

บันทึกการใช้งาน

บันทึกการใช้งานมีข้อมูลที่เกี่ยวข้องกับคำขอที่ป้อนสำหรับงาน. บันทึกการใช้งานมีสองรูปแบบ, รูปแบบการคงค้างและรูปแบบสพูล.

ในรูปแบบการคงค้าง, บันทึกการใช้งานสำหรับงานที่เสร็จสมบูรณ์สามารถเปลี่ยน เมื่องานอื่น (ระบบย่อย, ผู้ควบคุมระบบ, และอื่นๆ) โต้ตอบกับงานที่เสร็จสมบูรณ์. ในรูปแบบสพูล, บันทึกการใช้งานคือ snapshot (ชั่วขณะหนึ่ง) และไม่เปลี่ยน (เช่น สพูลไฟล์ที่ถูกสร้างขึ้นโดยคำสั่ง Display Job Log (DSPJOBLOG), หรือสร้างหลังจากที่งานทำ activity ของงานเสร็จสิ้นแล้ว).

แต่ละงานมีบันทึกการใช้งานที่เชื่อมโยงซึ่งสามารถมีข้อมูลดังต่อไปนี้สำหรับงาน :

- คำสั่งในงาน
- คำสั่งในโปรแกรม CL (ถ้าโปรแกรม CL ถูกสร้างขึ้นด้วยอ็อปชัน LOG(*YES) หรือด้วยอ็อปชัน LOG(*JOB) และคำสั่ง Change Job (CHGJOB) ถูกรันด้วยอ็อปชัน LOGCLPGM(*YES))
- ข้อความทั้งหมด (ข้อความและข้อความคำอธิบายสำหรับข้อความ) ถูกส่งไปให้ผู้ร้องขอ และไม่ถูกลบออกจาก message queue ของโปรแกรม

เมื่อสิ้นสุดงาน, สามารถบันทึกบันทึกการใช้งานไปที่สพูลไฟล์ QPJOBLOG เพื่อให้พิมพ์บันทึกได้. อย่างไรก็ตาม, การจัดทำบันทึกการใช้งานไม่จำเป็น ต้องหมายถึงการพิมพ์บันทึกการใช้งานหรือการสร้างสพูลไฟล์. (ตัวอย่างเช่น, สามารถใช้ Control Job Log QMHCTLJL API เพื่อระบุให้บันทึกการใช้งานถูกบันทึกเป็น outfile เมื่อจบงาน.)

- | คุณสามารถลดจำนวนของบันทึกการใช้งานที่จัดทำและลด contention สำหรับบรีซอร์ส (เช่นเอาต์พุตคิว) ได้. สิ่งนี้จะช่วยลดการ
- | ไซร้ซอร์ส ที่เกิดจากการจัดทำบันทึกการใช้งาน.

วิธีการสร้างบันทึกการใช้งาน

รีลีส V5R4 นำบันทึกการใช้งานเข้ามาสู่โลก "แห่งความต้องการ". บันทึกการใช้งานจะ พร้อมให้ใช้งานได้เมื่อต้องการ, และไม่มีการสร้างบันทึกการใช้งาน ที่ไม่ต้องการ.

พารามิเตอร์ LOG มีองค์ประกอบอยู่สามอย่างดังนี้: ระดับข้อความ (หรือการบันทึก), ความรุนแรงของข้อความ, และระดับของเนื้อความ. แต่ละองค์ประกอบ เหล่านี้มีค่าเฉพาะซึ่งเมื่อรวมกันแล้วกลายเป็นตัวกำหนดจำนวน และชนิดของข้อมูลทำงาน จะส่งไปที่บันทึกการใช้งาน.

ตัวอย่างเช่น, ค่า *NOLIST ขององค์ประกอบ Text จะทำให้ไม่มีการสร้าง บันทึกการใช้งานถ้างานจบลงอย่างปกติ. (บันทึกการใช้งานจะไม่มีการคงค้าง.) ถ้างานจบลงอย่างผิดปกติ (ถ้าได้การจบงานเป็น 20 หรือสูงกว่า), จะมีการสร้าง บันทึกการใช้งาน. ข้อความที่ปรากฏในบันทึกการใช้งานมีทั้งเนื้อความ และคำอธิบายข้อความ.

คุณสามารถควบคุมสิ่งที่จะสร้างบันทึกการใช้งานได้. โดยใช้พารามิเตอร์ LOGOUTPUT. เมื่องานเสร็จสมบูรณ์, หนึ่งในสาม actions ที่ส่งผลกระทบต่อวิธีการสร้างบันทึกการใช้งาน จะเกิดขึ้น. ข้างล่างนี้เป็นค่าของพารามิเตอร์ LOGOUTPUT :

- เซิร์ฟเวอร์บันทึกการใช้งานจัดทำบันทึกการใช้งาน: (*JOBLOGSVR)
- ตัวงานเองจัดทำบันทึกการใช้งาน: ถ้างานไม่สามารถสร้างบันทึกการใช้งาน ของตัวเองได้, บันทึกการใช้งานจะถูกสร้างขึ้นโดยเซิร์ฟเวอร์บันทึกการใช้งาน. (*JOBEND)
- ไม่มีการสร้างบันทึกการใช้งาน: บันทึกการใช้งานยังคงค้างอยู่จนกว่าจะถูก ลบออก. (*PND)

หมายเหตุ: ค่าเหล่านี้ไม่มีผลต่อบันทึกการใช้งานที่ถูกสร้างขึ้นเมื่อ message queue เต็ม และ job message queue full action ระบุ *PRTWRAP. ข้อความใน job message queue จะถูกบันทึกไปที่สพูลไฟล์, ที่ซึ่งสามารถพิมพ์บันทึกการใช้งานจากที่นั่นได้, ยกเว้นว่าจะใช้ Control Job Log Output (QMCTLJL) API ในงานเพื่อระบุว่าจะข้อความในบันทึกการใช้งานจะถูกบันทึกไปที่ไฟล์ฐานข้อมูล.

อะไรควบคุมพารามิเตอร์บันทึกการใช้งาน?

เมื่องานเริ่มต้น, งานจะได้ค่า LOGOUTPUT ของงานมาจากรายละเอียดของงาน. ถ้ารายละเอียดของงานระบุ *SYSVAL (ดีฟอลต์สำหรับ CRTJOB), งานจะใช้ค่าเอาต์พุตบันทึกการใช้งานที่ระบุอยู่ในค่ากำหนดของระบบ Job log output (QLOGOUTPUT). (ในขณะที่ค่าที่จัดส่งสำหรับค่ากำหนดของระบบ Job log output (QLOGOUTPUT) คือ *JOBEND, แต่ค่าที่แนะนำคือ *JOBLOGSVR.) หลังจากที่งานได้สร้างแอ็ททริบิวต์งาน LOGOUTPUT แล้ว, การเปลี่ยนแปลงใดๆ ที่รายละเอียดของงานหรือค่ากำหนดของระบบจะไม่มีผลต่องาน ที่แอ็ททริบิวต์. การเปลี่ยนแปลงที่ค่ากำหนดของระบบหรือรายละเอียดของงานมีผลต่อ งานที่ป้อนเข้าสู่ระบบหลังจากการเปลี่ยนแปลง.

คุณสามารถใช้คำสั่ง Change Job (CHGJOB) หรือ API (QWTCHGJB) ในการเปลี่ยนแอ็ททริบิวต์งาน LOGOUTPUT หลังจากที่แอ็ททริบิวต์ได้ถูกเซ็ทในงานแล้ว. การเปลี่ยนแปลงที่งานจะมีผล ในทันที.

ไม่ว่าคุณจะเลือกวิธีการใด, อีพซันสำหรับการจัดการบันทึกการใช้งานจะเหมือนกัน. คุณสามารถเซ็ทงานไม่ให้สร้างบันทึกการใช้งาน (*PND), ให้งานสร้างบันทึกการใช้งาน (*JOBEND), หรือให้เซิร์ฟเวอร์บันทึกการใช้งานสร้างบันทึกการใช้งาน (*JOBLOGSVR).

การคงค้างบันทึกการใช้งาน

สภาพการคงค้างบันทึกการใช้งานมีใช้กันเป็นเวลาหลายปีแล้ว. เมื่อ แอ็ททริบิวต์บันทึกการใช้งานของงานเป็น *PND, จะไม่มีการสร้างบันทึกการใช้งาน. ด้วยการอัปเดตที่พัฒนา V5R4 กับทั้ง iSeries Navigator และอินเตอร์เฟซแบบอักษรขณะนี้คุณจึงสามารถควบคุมวิธีและสถานการณ์ที่จะมีการสร้าง บันทึกการใช้งานสำหรับงานเฉพาะหนึ่งได้.

คุณลักษณะพิเศษใหม่นี้มีประโยชน์เมื่อคุณใช้ระบบในสภาพที่ถูกจำกัด. เมื่อระบบอยู่ในสภาพที่ถูกจำกัด, ระบบย่อยจะจบลงและงานนับพันอาจจบลงพร้อมกัน. สิ่งนี้อาจทำให้เกิดภาวะที่ใหญ่หลวง บนเอาต์พุตรีซอร์ส. โดยการป้องกันการสร้างบันทึกการใช้งานเหล่านี้, คุณสามารถลดผลกระทบบนรีซอร์สเหล่านี้ได้เป็นอย่างมาก.

อีกตัวอย่างหนึ่งที่คุณสามารถใช้คุณลักษณะพิเศษใหม่นี้คือ ในระหว่างที่การสื่อสาร ล้มเหลว. บางทีมีงานคล้ายกันจำนวนมากที่สร้างข้อความแสดงความผิดพลาด บันทึกการใช้งานเหมือนกัน. คุณสามารถเซ็ทบันทึกการใช้งานไม่ให้สร้างสพูลไฟล์สำหรับ งานทั้งหมดได้. จากนั้นถ้าการสื่อสารเกิดล้มเหลว, คุณสามารถใช้ คำสั่ง Work with Job Log (WRKJOBLOG) เพื่อกำหนด ไฟล์บันทึกที่จะพิมพ์ได้. คุณยังสามารถใช้จอภาพ Work with Job Logs (WRKJOBLOG) เพื่อจัดการ บันทึกการใช้งานได้ด้วย.

งานอาจอยู่ในสภาพการคงค้างบันทึกการใช้งาน เนื่องจากการทำงานของคำสั่ง Power Down System (PWRDWN SYS). ส่วนการติดต่อกับผู้ใช้ iSeries Navigator แสดง สถานะ "Completed - Job log pending" สำหรับงานเหล่านี้. นี่คือการเซ็ทย่อย ของสถานะอินเตอร์เฟซแบบอักษร *OUTQ.

การใช้ประโยชน์จากการพัฒนาเหล่านี้ช่วยให้คุณลดจำนวน ของบันทึกการใช้งานที่สร้างขึ้น และจึงช่วยลด contention สำหรับรีซอร์สได้. สิ่งนี้ส่งผลให้ประสิทธิภาพของระบบดีขึ้น.

เซิร์ฟเวอร์บันทึกการใช้งาน

โดยปกติ เซิร์ฟเวอร์บันทึกการใช้งานจะพิมพ์บันทึกการใช้งานของงานไปที่ สพูลไฟล์. คุณสามารถเรดบันทึกการใช้งานไปที่พรีนเตอร์หรือไปที่outfile, (ถ้ามี การระบุให้ทำโดยใช้คำสั่ง QMHCTLJL, Control job log API), อย่างไรก็ตาม ไม่แนะนำให้ใช้วิธีการนี้สำหรับการจัดทำบันทึกการใช้งาน.

คุณสามารถดูข้อมูลเกี่ยวกับเซิร์ฟเวอร์บันทึกการใช้งานผ่านทาง iSeries Navigator จากจอแสดงผล **Work Management** → **Server Jobs**, หรือจอแสดงผล **Work Management** → **Active Jobs**. (เพื่อให้ง่ายขึ้นในการระบุงานที่รันอยู่บนเซิร์ฟเวอร์บันทึกการใช้งาน, ให้ตรวจสอบ ให้แน่ใจว่าคุณได้รวมคอลัมน์ Server ในจอแสดงผลของคุณ.)

จำนวนสูงสุดของเซิร์ฟเวอร์บันทึกการใช้งานที่สามารถแอดที่พีได้ในเวลาหนึ่งคือ 30. คุณเริ่มเซิร์ฟเวอร์บันทึกการใช้งานเพิ่มเติมและจัดการเซิร์ฟเวอร์ในวิธีการเดียวกันกับเซิร์ฟเวอร์อื่นในระบบของคุณ. สิ่งนี้ทำได้โดยใช้คำสั่งอินเตอร์เฟซแบบอักขระ STRLOGSVR.

วิธีการเริ่มเซิร์ฟเวอร์บันทึกการใช้งาน

โดยค่าดีฟอลต์, เซิร์ฟเวอร์ บันทึกการใช้งานจะเริ่มโดยอัตโนมัติเมื่อระบบย่อย QSYSWRK เริ่มต้น. เซิร์ฟเวอร์ จะสิ้นสุดเมื่อระบบย่อย QSYSWRK สิ้นสุดลง.

คำสั่ง Start Job Log Server (STRLOGSVR) เริ่มเซิร์ฟเวอร์บันทึกการใช้งาน. เซิร์ฟเวอร์บันทึกการใช้งานจะบันทึกบันทึกการใช้งานสำหรับงานที่อยู่ในบันทึกการใช้งานที่มีสถานะคาง และไม่มีแอตทริบิวต์ของ *PND. เซิร์ฟเวอร์บันทึกการใช้งานบันทึก บันทึกการใช้งานของงานไปที่สพูลไฟล์, ไปที่พรีนเตอร์, หรือไปที่ outfile, อย่างไม่อย่างหนึ่ง (ถ้ามีการระบุให้ทำโดยใช้ QMHCTLJL, Control job log API).

ลักษณะบันทึกการใช้งาน

iSeries Navigator นำเสนออินเตอร์เฟซที่ใช้งาน, และอ่านได้ง่าย ซึ่งคุณสามารถ ดูบันทึกการใช้งานและข้อความของบันทึกการใช้งานได้. คุณยังสามารถดูบันทึกการใช้งานโดยใช้อินเตอร์เฟซแบบอักขระได้ด้วย.

คุณสามารถควบคุมคอลัมน์ที่จะปรากฏในรายการบันทึกการใช้งานได้ โดยใช้ หน้าต่าง Job Log - Columns. (**Work Management** → **Active Jobs** → **คลิกขวาที่งานและเลือก Job Log** → **View menu** → **Customize this view** → **Columns**) คอลัมน์ที่คุณสามารถเลือกให้แสดงในรายการบันทึกการใช้งานมีดังนี้:

Message ID	From Program
Message	Request Level
Sent	Severity
Thread	To Program
Type	

อินเทอร์เน็ตเฟสแบบอักขระ

เมื่อคุณใช้คำสั่ง Display Job Log (DSPJOBLOG), คุณจะเห็นจอแสดงผล Job Log. จอแสดงผลนี้แสดงชื่อโปรแกรมพร้อมกับสัญลักษณ์พิเศษ, ดังต่อไปนี้:

- >> คำสั่งการรันหรือคำสั่งถัดไปที่จะ รัน. ตัวอย่างเช่น, ถ้าโปรแกรม CL หรือโปรแกรมภาษาชั้นสูงถูกเรียก, การเรียกไปที่โปรแกรมจะถูกแสดงขึ้น.
- > คำสั่งประมวลผลเสร็จสมบูรณ์แล้ว.
- .. คำสั่งยังไม่ได้ประมวลผล.
- ? ตอบข้อความ. สัญลักษณ์นี้ทำเครื่องหมายทั้งข้อความ ที่ต้องการคำตอบและข้อความที่ได้ตอบไปแล้ว.

หัวข้อบันทึกการใช้งาน:

หัวข้อบันทึกการใช้งานตั้งอยู่ที่ส่วนบนสุดของแต่ละหน้าของบันทึกการใช้งาน ที่พิมพ์. หัวข้อเหล่านี้ระบุงานซึ่งบันทึกการใช้งานใช้และลักษณะ ของแต่ละ entry. ข้อมูลต่อไปนี้เป็นรายการของ entries ที่อาจใช้ได้ในหัวข้อบันทึกการใช้งาน.

- ชื่อที่ถูกต้องทั้งหมดของงาน (ชื่องาน, ชื่อผู้ใช้, และหมายเลขงาน)
- ชื่อของรายละเอียดของงานที่ใช้ในการเริ่มงาน
- วันที่และเวลาที่งานเริ่มต้น
- message identifier
- ชนิดข้อความ
- ความรุนแรงของข้อความ
- วันที่และเวลาที่ส่งแต่ละข้อความ
- ข้อความ. ถ้าระดับการบันทึกจะรวม second-level text, second-level text จะปรากฏบนบรรทัดที่ตามมาข้างใต้ข้อความ
- โปรแกรมซึ่งข้อความหรือคำขอลูกส่งมา
- หมายเลขคำสั่งอินเทอร์เน็ตเฟสเครื่องหรือออฟเซตที่โปรแกรม ซึ่งข้อความถูกส่งไป

หมายเหตุ: หมายเลขคำสั่งอินเทอร์เน็ตเฟสเครื่อง ปรากฏขึ้นสำหรับข้อความ escape, การแจ้ง, และข้อความวินิจฉัยเท่านั้น. สำหรับข้อความชนิด อื่นทั้งหมด, หมายเลขคำสั่งอินเทอร์เน็ตเฟสเครื่องจะถูกเซตเป็นศูนย์.

- ถ้างานใช้ APPC, หัวข้อจะมีบรรทัดที่แสดงหน่วยของ work identifier สำหรับ APPC.

ข้อความ:

ข้อความประกอบด้วย ชื่องาน, ชนิดข้อความ, วันที่และเวลาที่ส่ง, การดำเนินการที่เกิดขึ้น, และการดำเนินการที่จำเป็นเพื่อแก้ไขปัญหา. สิ่งนี้มีประโยชน์เมื่อคุณพยายามแก้ไขปัญหาใดๆ ที่อาจเกิดขึ้นบนเซิร์ฟเวอร์ของคุณ. คุณสามารถเข้าใช้งานบันทึกการใช้งานสำหรับงานเซิร์ฟเวอร์ผ่านทาง iSeries Navigator. ข้อความแบ่งเป็นสองประเภท, ข้อความที่แจ้งเตือนได้และข้อความที่บันทึกใน บันทึกการใช้งาน.

ข้อความแจ้งเตือน - ข้อความเหล่านี้จะถูกส่งไปยัง QSYSOPR เนื่องจากข้อความเหล่านี้ต้องการการดำเนินการในทันที. ข้อความจะประกอบด้วยปัญหา, สาเหตุ, และการดำเนินการกู้คืนที่จำเป็น. ตัวอย่างเช่น, เซิร์ฟเวอร์ไม่สามารถเริ่มทำงานได้หรือ

ซีิร์ฟเวอร์หยุดทำงานโดยไม่คาดคิด. บางซีิร์ฟเวอร์ส่งข้อความเตือนไปที่ QSYSOPR. ข้อความเหล่านี้มี Alert Option (ALROPT) ระบุไว้ในรายละเอียดข้อความ. คุณสามารถใช้ตัวแจ้งเตือนเพื่อให้มีการจัดการศูนย์กลางของข้อความที่แจ้งเตือนได้.

ข้อความในบันทึกการใช้งาน - ข้อความเหล่านี้มีลักษณะเป็นข้อความวินิจฉัย, หมายความว่าข้อความเหล่านี้ไม่ใช่ข้อความที่สำคัญมากแต่มีไว้เพื่อเตือนผู้ใช้งานว่าต้องมีการดำเนินการบางอย่าง. ข้อความเหล่านี้อาจถูกสร้างขึ้นโดยระบบและโดยผู้ใช้.

ระดับการบันทึกข้อความ

ระดับการบันทึกข้อความกำหนดว่า ข้อความใดและข้อความชนิดใดที่ควรถูกบันทึกสำหรับ งาน. ข้อมูลต่อไปนี้เป็นการอธิบายว่าแต่ละระดับแสดงถึงอะไร.

ระดับ 1	ข้อความทั้งหมดที่ส่งไปที่ message queue ภายนอกของงาน พร้อมกับค่าความรุนแรงมากกว่าหรือเท่ากับค่าความรุนแรงข้อความ. (ใน iSeries Navigator, สามารถดูค่าความรุนแรงข้อความ (0-99) บนหน้าต่าง Job Properties - Job Log. นี้เป็นค่าที่คุณสามารถควบคุมได้.)
ระดับ 2	ข้อความทั้งหมดที่คุณสมบัติตรงกับระดับ 1 และข้อความค่าขอใดๆ ซึ่งส่งผลให้เกิดข้อความระดับสูงมากกว่าหรือเท่ากับค่าความรุนแรงของข้อความ. หมายเหตุ: ข้อความระดับสูงคือข้อความที่จะ ถูกส่งไปที่ program message queue ของโปรแกรมที่ได้รับข้อความค่าขอ. (เช่น, QCMD เป็นโปรแกรมการประมวลผลค่าขอที่ได้มาจาก IBM ที่จะได้รับข้อความค่าขอ.)
ระดับ 3	ข้อความทั้งหมดที่คุณสมบัติตรงกับระดับ 1 หรือระดับ 2 และข้อความค่าขอทั้งหมด. นอกจากนี้, คำสั่งใดๆ จากโปรแกรม CL จะถูกรวมด้วย ถ้ามีการเลือก Log commands from CL programs box (หน้าต่าง Job Properties - Job Log). หมายเหตุ: Log commands from CL programs box เท่าเทียมกับแอตทริบิวต์การล็อกของโปรแกรม CL.
ระดับ 4	ข้อความค่าขอทั้งหมดและข้อความทั้งหมดที่มีค่าความรุนแรง มากกว่าหรือเท่ากับค่าความรุนแรงการล็อกข้อความ, รวมถึงข้อความ trace. นอกจากนี้, คำสั่งใดๆ จากโปรแกรม CL จะถูกรวมด้วย ถ้ามีการเลือก Log commands from CL programs box (หน้าต่าง Job Properties - Job log). หมายเหตุ: Log commands from CL programs box เท่าเทียมกับแอตทริบิวต์การล็อกของโปรแกรม CL.

ไฟล์บันทึกงานแบบโต้ตอบ

รายละเอียดของงานทั้งหมดที่ได้มาจาก IBM QCTL, QINTER, และ QPGMR ล้วนแต่มี ระดับไฟล์บันทึกเป็น LOG(4 0 *NOLIST); ดังนั้นข้อความคำอธิบายทั้งหมดจึง ถูกบันทึกไปที่บันทึกการใช้งาน. อย่างไรก็ตาม, บันทึกการใช้งานจะไม่ถูกพิมพ์ถ้างานจบลง อย่างปกติ ยกเว้นว่าคุณได้ระบุ *LIST บนคำสั่ง SIGNOFF.

ถ้าผู้ใช้จอภาพใช้เมนูที่ได้มาจาก IBM หรือจอแสดงผล command entry, ข้อความแสดงความผิดพลาดทั้งหมดจะถูกแสดงขึ้น. ถ้าผู้ใช้จอภาพใช้ initial program ที่ผู้ใช้บันทึก, ข้อความที่ไม่ได้มอนิเตอร์ใดๆ จะทำให้ initial program จบลงและมีการจัดทำบันทึกการใช้งาน. อย่างไรก็ตาม, ถ้า initial program มอนิเตอร์ข้อความ, โปรแกรมจะได้รับการควบคุมเมื่อได้รับข้อความ. ในกรณี นี้, สิ่งสำคัญคือการตรวจสอบให้แน่ใจว่ามีการจัดทำบันทึกการใช้งาน เพื่อให้ คุณสามารถกำหนดข้อผิดพลาดเฉพาะที่เกิดขึ้นได้.

ตัวอย่างเช่น, สมมติว่า initial program แสดงเมนูที่มีอ็อปชัน signoff, ซึ่งมีค่าดีฟอลต์เป็น *NOLIST. initial program จะมอนิเตอร์ exception ทั้งหมดและรวมคำสั่ง Change Variable (CHGVAR) ที่จะเปลี่ยนอ็อปชัน signoff เป็น *LIST ถ้า exception เกิดขึ้น:

```

PGM
DCLF MENU
DCL &SIGNOFFDPT TYPE(*CHAR) LEN(7)
VALUE(*NOLIST)
.
.
.
MONMSG MSG(CPF0000) EXEC(GOTO ERROR)
PROMPT: SNDRCVF RCDfmt(PROMPT)
CHGVAR &IN41 '0'
.
.
.
IF (&OPTION *EQ '90') SIGNOFF
LOG(&SIGNOFFOPT);
.
.
.
GOTO PROMPT
ERROR: CHGVAR&SIGNOFFOPT '*LIST'
CHGVAR &IN41 '1'
GOTO PROMPT
ENDPGM

```

ถ้า exception เกิดขึ้น, คำสั่ง CHGVAR จะเปลี่ยนอ็อปชัน บนคำสั่ง SIGNOFF เป็น *LIST และเซ็ตบนตัวบ่งชี้. สามารถใช้ตัวบ่งชี้นี้ เพื่อกำหนดเงื่อนไขค่าคงที่ที่จะแสดงข้อความที่อธิบายว่า ข้อผิดพลาดที่ไม่คาดคิดเกิดขึ้นได้อย่างไรและบอกให้ผู้ใช้จ้อภาพทราบว่าจะทำอะไร.

บันทึกประวัติ (ที่ผ่านมา) QHST

ไฟล์บันทึกประวัติ (QHST) ประกอบด้วย message queue และ ฟิสิคัลไฟล์ที่เรียกว่าล็อกเวอร์ชัน. ระบบจะบันทึกข้อความที่ส่งไปที่ log message queue ลงใน log-version physical file ปัจจุบัน.

บันทึกประวัติ (QHST) มี trace ระดับสูงของ activity ระบบ เช่น ข้อความระบบ, ระบบย่อย, ข้อมูลงาน, สถานะอุปกรณ์, และข้อความผู้ควบคุม ระบบ. message queue คือ QHST.

Log-Version

แต่ละล็อกเวอร์ชันคือฟิสิคัลไฟล์ ที่ถูกตั้งชื่อในวิธีต่อไปนี้:

Qxxxxyydddn

โดยที่:

xxx คือ description 3 อักขระของชนิดของไฟล์บันทึก (HST)

yyddd คือวันที่แบบ Julian ซึ่งล็อกเวอร์ชันถูกสร้างขึ้น

n คือหมายเลขลำดับ ภายในวันที่ Julian (0 ถึง 9 หรือ A ถึง Z)

เมื่อ ล็อกเวอร์ชันเต็ม, เวอร์ชันใหม่ของไฟล์บันทึกจะถูกสร้างขึ้นโดยอัตโนมัติ.

หมายเหตุ: จำนวนของเร็กคอร์ดในล็อกเวอร์ชันของบันทึกประวัติถูกระบุในค่ากำหนดของระบบ Maximum records in history log (QHSTLOGSIZ). ค่ากำหนดของระบบนี้ ยังสนับสนุนอ็อปชัน *DAILY ซึ่งสร้างเวอร์ชันใหม่ในแต่ละวันด้วย.

การจัดรูปแบบบันทึกประวัติ (ที่ผ่านมา):

ไฟล์ฐานข้อมูลใช้ในการจัดเก็บข้อความที่ส่งไปที่ไฟล์บันทึกของระบบ. เนื่องจากเร็กคอร์ดทั้งหมดในไฟล์ข้อความมีความยาวเท่ากันและข้อความที่ส่งไปที่ไฟล์บันทึกมีความยาวแตกต่างกัน, ข้อความจึงสามารถแตกออกมากกว่าหนึ่งเร็กคอร์ดได้. แต่ละเร็กคอร์ดสำหรับข้อความมีฟิลด์อยู่สามฟิลด์:

- วันที่และเวลาของระบบ (ฟิลด์แบบอักขระที่มีความยาว 8). นี่เป็นฟิลด์ ภายใน. วันที่และเวลาที่แปลงจะอยู่ในข้อความด้วย.
- หมายเลขเร็กคอร์ด (ฟิลด์ 2-ไบต์). ตัวอย่างเช่น, ฟิลด์มี hex 0001 สำหรับเร็กคอร์ดแรก, hex 002 สำหรับเร็กคอร์ดที่สอง, และต่อไป.
- ข้อมูล (ฟิลด์แบบอักขระที่มีความยาว 132).

รูปแบบสำหรับฟิลด์ที่สาม (ข้อมูล):

ตารางที่ 1. รูปแบบสำหรับฟิลด์ที่สามของเร็กคอร์ดแรก

เนื้อหา	ชนิด	ความยาว	ตำแหน่งในเร็กคอร์ด
ชื่องาน	อักขระ	26	11-36
วันที่และเวลาที่แปลง	อักขระ	13	37-49
Message ID	อักขระ	7	50-56
ชื่อไฟล์ข้อความ	อักขระ	10	57-66
ชื่อไลบรารี	อักขระ	10	67-76
ชนิดข้อความ	อักขระ	2	77-78
โค้ดความรุนแรง	อักขระ	2	79-80
ชื่อโปรแกรมการส่ง	อักขระ	12	81-92
ชื่อโปรแกรมการรับ	อักขระ	10	97-106
หมายเลขคำสั่งโปรแกรมการรับ	อักขระ	4	107-110
ความยาวเนื้อความ	ไบนารี	2	111-112
ความยาวข้อมูลแบบข้อความ	ไบนารี	2	113-114
สำรองไว้	อักขระ	28	115-142

ตารางที่ 2. รูปแบบของฟิลด์ที่สาม (ข้อมูล) ของเรีกคอร์ดที่เหลือ

เนื้อหา	ชนิด	ความยาว
ข้อความ	อักขระ	ตัวแปร (ความยาวนี้ถูกระบุในเรีกคอร์ดแรก (ตำแหน่ง 111 และ 112) และไม่สามารถเกินกว่า 132.)
ข้อมูลแบบข้อความ	อักขระ	ตัวแปร (ความยาวนี้ถูกระบุในเรีกคอร์ดแรก (ตำแหน่ง 113 และ 114).)

ข้อความจะไม่แยกออกเมื่อเริ่มเวอร์ชันใหม่ของไฟล์บันทึก. เรีกคอร์ดแรกและเรีกคอร์ดหลังสุดของข้อความจะอยู่ในเวอร์ชัน QHST เดียวกันเสมอ.

การประมวลผลไฟล์ QHST

ถ้าคุณใช้โปรแกรมภาษาชั้นสูง ในการประมวลผลไฟล์ QHST, ให้จำไว้ว่าข้อมูลแบบข้อความจะเริ่มต้นที่ ตำแหน่งตัวแปร สำหรับการชี้ข้อความเดียวกันในแต่ละครั้ง. เหตุผลที่เป็นเช่นนี้คือ ข้อความมีตัวแปรที่สามารถแทนที่ได้ ดังนั้นความยาวที่แท้จริงของข้อความจึง แตกต่างกันไป.

อย่างไรก็ดี, สำหรับข้อความ CPF1124 (การเริ่มงาน) และข้อความ CPF1165 (การสิ้นสุดงาน) ข้อมูลแบบข้อความจะเริ่มต้นในตำแหน่ง 11 ของ เรีกคอร์ดที่สามเสมอ.

ข้อมูลประสิทธิภาพและ QHST:

ข้อมูลประสิทธิภาพไม่ได้แสดงขึ้นเป็นข้อความบนข้อความ CPF1164. เนื่องจากข้อความอยู่ในไฟล์บันทึก QHST, ผู้ใช้จึงสามารถเขียนแอฟพลิเคชันโปรแกรม เพื่อดึงข้อมูลนี้ออกมาได้.

ข้อมูลประสิทธิภาพถูกส่งผ่านเป็นค่าข้อความการแทนที่ความยาวผันแปรได้. นี่หมายความว่าข้อมูลอยู่ในโครงสร้างภายใน entry แรก ที่เป็นความยาวของข้อมูล. ขนาดของฟิลด์ความยาวไม่ได้รวมอยู่ใน ความยาว.

Time and Date: ฟิลด์ข้อมูลแรกในโครงสร้างคือเวลาและวันที่ ที่งานเข้าสู่ระบบ และเวลาที่ขั้นตอนการเรียดแรกสำหรับงาน ถูกเริ่มขึ้น. เวลาแสดงอยู่ในรูปแบบ 'hh:mm:ss'. ตัวแบ่งเวลา ในตัวอย่างนี้คือเครื่องหมายโคลอน. ตัวแบ่งนี้ถูกกำหนดโดยค่าที่ระบุในค่ากำหนดของระบบ Date and time (QTIMSEP). วันที่อยู่ใน รูปแบบที่กำหนดในค่ากำหนดของระบบ Date and time (QDATFMT) และ ตัวแบ่งอยู่ในค่ากำหนดของระบบ Date and time (QDATSEP). เวลาและวันที่ ที่งานเข้าสู่ระบบเป็น เวลาและวันที่ก่อนหน้าวันที่และเวลาการเริ่มงานในโครงสร้าง. เวลาและวันที่ที่งานเข้าสู่ระบบคือ เวลาที่ระบบรับรู้ ว่างานจะถูก เริ่มต้น (โครงสร้างงานถูกเชื่อมต่ออยู่ข้างงาน). สำหรับงานแบบโต้ตอบ, เวลา job entry คือเวลาที่ระบบรับรู้รหัสผ่าน. สำหรับงานแบ็ตช์, นี่คือนเวลาที่คำสั่ง Batch Job (BCHJOB) หรือ Submit Job (SBMJOB) จะถูกประมวลผล. สำหรับงานการมอนิเตอร์, เครื่องอ่านหรือ writer, นี่คือนเวลาที่คำสั่งการเริ่มต้นที่สอดคล้องกันจะถูกประมวลผล, และสำหรับงานแบบ autostart นี่คือนเวลาในระหว่างการเริ่มต้นของระบบย่อย.

Total Response Time and Number of Transactions: เวลาและวันที่ ต่อไปนี้คือเวลาตอบสนองทั้งหมดและจำนวนของ transactions. เวลาตอบสนอง ทั้งหมดแสดงในหน่วยวินาทีและมีค่าสะสมของช่วงเวลาทั้งหมดของงานที่กำลังประมวลผล นับตั้งแต่การกดปุ่ม Enter ที่ไวก์สเดชัน จนถึงเวลาในการแสดงจอแสดงผลถัดไป. ข้อมูลนี้คล้ายกับข้อมูล ที่แสดงบนจอแสดงผล Work with Active Job (WRKACTJOB). ฟิลด์นี้ใช้สำหรับงานแบบโต้ตอบเท่านั้น.

ในกรณีที่ระบบล้มเหลวหรือมีการจบงานอย่างผิดปกติ อาจเป็นไปได้ที่ เวลาทั้งหมดจะไม่รวมเวลาของ transaction หลังสุด. ในกรณีนี้ได้การจบงาน จะเป็น 40 หรือมากกว่า. นอกจากนี้ การนับ transaction มีความสำคัญสำหรับงานแบบ โต้ตอบมากกว่า งานคอนโซล และคือจำนวน ของช่วงเวลาตอบสนองที่นับโดยระบบในระหว่างงาน.

Number of Synchronous Auxiliary I/O Operations:จำนวนของการดำเนินการ auxiliary I/O แบบซิงโครนัสตามจำนวนของ transactions. สำหรับงานที่มีหลาย threads, ค่านี้จะรวมเฉพาะการดำเนินการ auxiliary I/O แบบซิงโครนัสจาก initial thread เท่านั้น. ฟิลด์นี้เหมือนกับฟิลด์ AUXIO ที่ปรากฏ บนจอแสดงผล WRKACTJOB ยกเว้นความแตกต่างดังต่อไปนี้:

- จอแสดงผล WRKACTJOB แสดงค่าสำหรับ initial thread ของขั้นตอนการเราต์ปัจจุบัน.
- ข้อความ QHST มีผลรวมสะสมสำหรับ initial thread ของแต่ละขั้นตอนการเราต์ในงาน.

ถ้างานจบด้วยโค้ดการจบ 70, ค่านี้อาจไม่มี การนับสำหรับขั้นตอนการเราต์สุดท้าย. นอกจากนี้, ถ้างานมีอยู่ใน IPL (โดยใช้คำสั่ง Transfer Batch Job (TFRBCHJOB)) งานจะจบลงก่อนที่จะแอ็คทีฟหลังจาก IPL, ค่าคือ 0.

สพุลไฟล์

สพุลไฟล์เก็บพักข้อมูลออกจนกว่าข้อมูลจะสามารถพิมพ์ได้. สพุลไฟล์ รวบรวมข้อมูลจากอุปกรณ์ จนกว่าโปรแกรมหรืออุปกรณ์จะสามารถประมวลผล ข้อมูลได้. โปรแกรมจะใช้สพุลไฟล์เมื่ออ่านหรือบันทึก ไปที่อุปกรณ์จริง. นี่คือการสพุลอินพุต และเอาต์พุต.

ระบบทำการสพุลอินพุตสำหรับฐานข้อมูลและดิสเก็ตไฟล์. โปรแกรม ที่ได้มาจาก IBM, เรียกว่าเครื่องอ่าน, ถูกเริ่มในระบบย่อยการสพุล, จะอ่าน streams งานแบ็ตซ์จากอุปกรณ์, และวางงานบนคิว งาน.

ทำการสพุลเอาต์พุตสำหรับพริ้นเตอร์. โปรแกรมที่ได้มาจาก IBM, เรียกว่า เครื่องเขียนพริ้นเตอร์, ถูกเริ่มในระบบย่อยการสพุล, จะเลือกสพุลไฟล์ จากเอาต์พุตคิว, และบันทึกเร็กคอร์ดของเอาต์พุตไฟล์ที่สพุลไปที่ พริ้นเตอร์.

เมื่อสิ้นสุดงาน, สามารถบันทึกบันทึกการใช้งานไปที่สพุลไฟล์ QPJOBLOG เพื่อให้บันทึกสามารถพิมพ์ได้.

Job accounting

ฟังก์ชัน job accounting รวบรวมข้อมูลเพื่อให้คุณสามารถกำหนดว่า ใครกำลังใช้ระบบของคุณและเขากำลังใช้รีซอร์สระบบอะไรอยู่. และยังช่วย คุณในการประเมินการใช้โดยรวมของระบบของคุณ. Job accounting เป็น สิ่งที่เลือกได้. คุณต้องทำขั้นตอนพิเศษเพื่อตั้งค่า job accounting. คุณสามารถร้องขอให้ ระบบรวบรวมข้อมูลการจัดทำบัญชีรีซอร์สของงาน, ข้อมูลการจัดทำบัญชี พริ้นเตอร์ไฟล์, หรือทั้งสองอย่าง. คุณยังสามารถกำหนด accounting codes ที่โปรแกรมผู้ใช้ หรืองานเฉพาะได้.

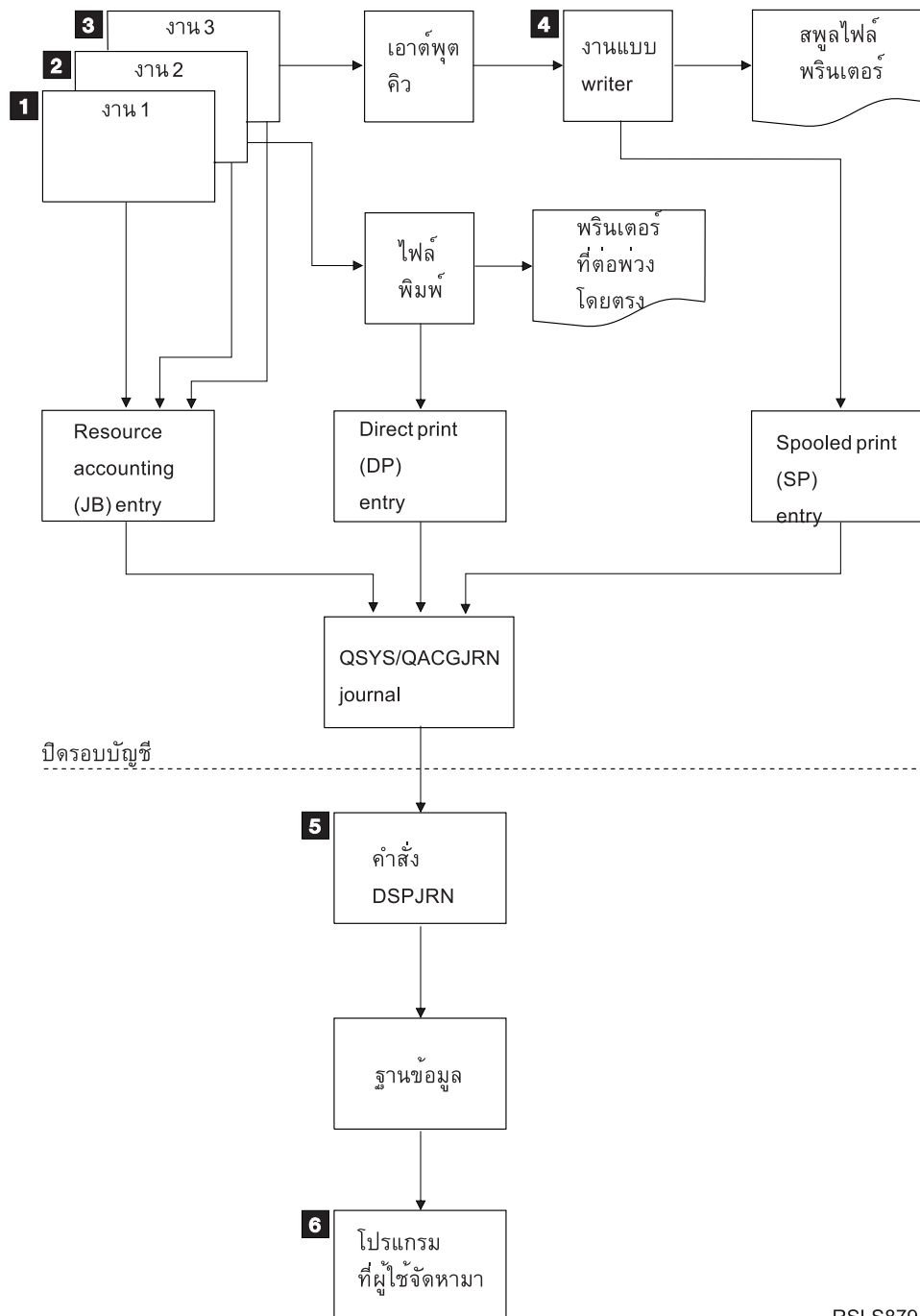
ข้อมูล job accounting ปกติแสดงรายละเอียดของงานที่กำลังรันในระบบของคุณ และรีซอร์สที่งานใช้ เช่น การใช้ของหน่วยการประมวลผล, พริ้นเตอร์, จอภาพ, ฐานข้อมูล และฟังก์ชันการสื่อสาร.

สถิติ job accounting ถูกเก็บไว้โดยใช้ journal entries ที่ทำใน system accounting journal QSYS/QACGJRN. คุณควรทราบวิธีปฏิบัติการดำเนินการ journal management, เช่น การบันทึก journal receiver, การเปลี่ยน journal receivers, และการลบ journal receivers เก่า.

เมื่อคุณต้องการวิเคราะห์ข้อมูล job accounting, ข้อมูลต้องถูกแยก จากเจอร์นัล QACGJRN โดยใช้คำสั่ง Display Journal (DSPJRN). ด้วยคำสั่งนี้ คุณสามารถบันทึก entries เข้าในไฟล์ฐานข้อมูลได้. คุณต้องบันทึกแอ็พพลิเคชันโปรแกรมหรือใช้ยูทิลิตี้ เช่น เคียวรียูทิลิตี้ เพื่อวิเคราะห์ข้อมูล.

วิธีการทำงานของ job accounting

สำหรับภาพรวมนี้ของวิธีการทำงานของ job accounting, ให้สมมติว่า มีงานที่เข้าสู่ระบบสามงาน.



RSLS879-2

รูปที่ 4. Job Accounting Overview

1. เมื่อ Job1 เสร็จสมบูรณ์แล้ว, ระบบจะสรุปรีซอร์สที่ใช้และบันทึก JB journal entry ไปที่เจอร์นัล QACGJRN. ถ้ามีการเปลี่ยน accounting code ในระหว่าง งาน, JB journal entry จะถูกบันทึกในแต่ละครั้งที่ accounting code ถูกเปลี่ยนแปลง

และที่ตอนจบของงาน. Job1 ไม่ทำให้เกิดพริ้นเตอร์เอาต์พุตใดๆ , และไม่มีการสร้างบันทึกการใช้งาน. ด้วยเหตุนี้, จึงไม่มีการสร้าง direct print (DP) หรือ spooled print (SP™) journal entries สำหรับ Job1.

2. Job2 คือการพิมพ์ไฟล์โดยตรงไปที่พริ้นเตอร์. เมื่อไฟล์เสร็จสมบูรณ์ จะมีการบันทึก DP journal entry ซึ่งสรุปข้อมูลที่ได้ออกพิมพ์. เมื่อ Job2 เสร็จสมบูรณ์, ระบบจะสรุปรีจิสเตอร์ที่ใช้และบันทึก JB journal entry. Job2 ไม่ทำให้เกิด spooled printer output ใดๆ และไม่มีการสร้างบันทึกการใช้งาน. ดังนั้น, จึงไม่มีการจัดทำ SP journal entry สำหรับ Job2.
3. Job3 คือการพิมพ์ไปที่ไฟล์ที่ถูกสพูล. SP journal entry ไม่ถูกบันทึก ยกเว้นว่า print writer จะพิมพ์ไฟล์. เมื่อ Job3 เสร็จสมบูรณ์, ระบบจะสรุปรีจิสเตอร์ที่ใช้และบันทึก JB journal entry. ถ้าบันทึกการใช้งานถูกสร้างขึ้นเมื่องานเสร็จสมบูรณ์, จะถือว่าบันทึกการใช้งานเป็นสพูลไฟล์ปกติ และมีการสร้าง SP journal entry ถ้าไฟล์ถูกพิมพ์.
4. print writer จะถูกเริ่มขึ้นและพิมพ์ไฟล์ที่สร้างขึ้นโดยงานหนึ่งหรือ หลายงาน. เมื่อ writer พิมพ์ไฟล์เสร็จแล้ว, writer จะจัดทำ SP journal entry. ไม่มีการทำ SP journal entry ถ้าไฟล์ถูกยกเลิก ก่อนที่การพิมพ์จะเริ่มขึ้น.
5. เมื่อสิ้นสุด accounting period, สามารถใช้คำสั่ง Display Journal (DSPJRN) เพื่อบันทึก journal entries ทั้งหมดลงในไฟล์ฐานข้อมูลได้.
6. สามารถใช้โปรแกรมที่ผู้ใช้เขียนหรือเคียวรี่ยูทิลิตี้เพื่อวิเคราะห์ accounting data. รายงาน เช่น รีจิสเตอร์ที่ใช้ จะคอมไพล์ข้อมูลตาม accounting code เฉพาะ, ผู้ใช้, หรือชนิดของงาน.

ลักษณะการดำเนินการ Job Accounting:

ระบบ iSeries พยายามจัดสรรแหล่งเก็บข้อมูลหลักให้มีประสิทธิภาพมากที่สุดเท่าที่จะเป็นไปได้. งานอาจไม่ได้ใช้จำนวนของรีจิสเตอร์ที่เท่ากันในแต่ละครั้งที่รัน.

ตัวอย่างเช่น, ถ้ามีงานแฉีกที่พหลายงานบนระบบของคุณ, งานจะใช้เวลา ในการสร้างความต้องการรีจิสเตอร์ที่ต้องใช้สำหรับการรันขึ้นใหม่มากกว่าถ้าใช้สภาวะแวดล้อมระบบเฉพาะงาน. ระบบจะใช้งานและระดับความสำคัญการรัน ที่ได้กำหนดที่งานต่างๆ เพื่อช่วยในการจัดการแหล่งเก็บข้อมูลหลัก. ดังนั้น, งานระดับความสำคัญสูงจึงสามารถใช้รีจิสเตอร์ของระบบน้อยกว่างานระดับความสำคัญต่ำได้.

เนื่องจากลักษณะการดำเนินการของระบบเหล่านี้, คุณอาจต้องการใช้ การตีความของคุณเองหรือ algorithm กับข้อมูล job accounting ที่รวบรวมได้. ถ้าคุณคิดค่าธรรมเนียมสำหรับการใช้ระบบของคุณ คุณอาจต้องการคิดค่าธรรมเนียม มากขึ้น สำหรับงานที่มีระดับความสำคัญสูง, งานที่ทำในระหว่างเวลาระบบที่มีการใช้งานมากที่สุด, หรือการใช้รีจิสเตอร์ที่สำคัญมาก.

การประมวลผล Accounting Journal:

accounting journal QSYS/QACGJRN ถูกประมวลผลเช่นเดียวกับเจอร์นัลอื่น. และยังสามารถบันทึกไฟล์ในเจอร์นัลนี้ได้ด้วย แม้ว่าจะแนะนำให้ผู้ใช้เจอร์นัล สำหรับข้อมูลการจัดทำบัญชีเพียงอย่างเดียวเพื่อให้การใช้งานง่าย.

คุณสามารถใช้คำสั่ง Send Journal Entry (SNDJRNE) เพื่อส่ง entries อื่นไปที่เจอร์นัลนี้. ในขณะที่มีข้อควรพิจารณาในการดำเนินการเพิ่มเติม โดยใช้หลายเจอร์นัล, แต่การ ไม่อนุญาตให้มี file entries ใดๆ ในเจอร์นัล QACGJRN ก็มีข้อดี. โดยทั่วไปคือสามารถควบคุมเจอร์นัล QACGJRN แยกต่างหากเพื่อให้ job accounting entries ทั้งหมดสำหรับ รอบเวลาบัญชีหนึ่งมีจำนวน journal receivers น้อยที่สุด และ journal receiver ใหม่ถูกเริ่มต้นที่ตอนต้นของรอบเวลาบัญชี. System entries ยังปรากฏอยู่ในเจอร์นัล QACGJRN ด้วย. มี entries ที่มี journal code เป็น J, ซึ่งเกี่ยวข้องกับ IPL และการดำเนินการทั่วไปที่ปฏิบัติ บน journal receivers (เช่น, การบันทึกของ receiver).

Job accounting entries

Job accounting entries ถูกวางใน journal receiver โดยเริ่มต้นด้วยงานถัดไปที่เข้าสู่ระบบ หลังจากที่คุณสั่ง Change System Value (CHGSYSVAL) มีผลบังคับใช้. accounting level ของงานจะถูกกำหนดเมื่องานเข้าสู่ระบบ. ถ้าค่ากำหนดของระบบ Journal accounting information (QACGLVL) เปลี่ยนแปลง หลังจากที่ยานเริ่มขึ้น, ค่ากำหนดของระบบจะไม่มีผลกระทบต่อชนิดของการจัดทำบัญชีที่ปฏิบัติการสำหรับงานนั้น. direct print (DP) และ spooled print (SP) entries จะเกิดขึ้น ถ้างานที่สร้างไฟล์กำลังดำเนินการภายใต้การจัดทำบัญชี และค่ากำหนดของระบบถูกเซตเป็น *PRINT. ถ้าสพูลไฟล์ถูกพิมพ์หลังจากที่ได้เซต accounting level เป็น *PRINT หรือถ้างานที่สร้างไฟล์ เริ่มขึ้นก่อนที่ accounting level จะถูกเปลี่ยน, จะไม่มีการทำเจอร์นัล สำหรับสพูลไฟล์เหล่านั้น.

ใช้ job accounting เมื่อใด

คุณควรจะใช้ฟังก์ชัน job accounting ในข้อความ QHST หลังจากที่คุณหมายเลขข้อความ CPF1124 และ CPF1164 พร้อมใช้งานเสมอในไฟล์บันทึก QHST หรือไม่? หรือ, คุณควรจะใช้ job accounting หรือไม่? ใช้ข้อมูลต่อไปนี้เพื่อช่วยในการกำหนดวิธีการที่เหมาะสมที่สุดสำหรับองค์กรของคุณ.

ข้อมูลเพิ่มเติมที่ได้มาจาก job accounting

Job accounting มีข้อมูลทั้งหมดที่ได้มาจาก CPF1164 บวก:

- Accounting code
- จำนวนของไฟล์พิมพ์, บรรทัด, และหน้าที่โปรแกรมสร้างขึ้น
- จำนวนของการดำเนินการอ่าน, บันทึก, และอัปเดตฐานข้อมูล
- จำนวนของการดำเนินการอ่านและบันทึกการสื่อสาร
- บรรทัดและหน้าที่พิมพ์จริง
- เวลาที่งานแอ็คทีฟและหยุดชั่วคราว
- จำนวนแท้จริงของไบต์ของข้อมูลการควบคุมและข้อมูลการพิมพ์ที่ส่งไปที่ 프린เตอร์

ฟังก์ชัน job accounting จะมีประสิทธิภาพเพิ่มมากขึ้นสำหรับการรวบรวม สถิติ job accounting ถ้า:

- ข้อมูลรีจิสเตอร์ที่เกี่ยวข้องกับการใช้งานข้อมูล, 프린เตอร์, และการสื่อสาร มีความสำคัญ.
- Accounting codes ถูกกำหนดที่ผู้ใช้หรืองาน.
- ข้อมูลสำหรับเอาต์พุตที่พิมพ์มีความสำคัญ.
- Job accounting ต้องถูกทำบนพื้นฐาน accounting segment ในงานแทนที่จะทำบนพื้นฐานงานที่สมบูรณ์.
- ต้องใช้ข้อมูลเวลาแอ็คทีฟและเวลาที่หยุดชั่วคราว.

ข้อความ QHST จะมีประสิทธิภาพมากขึ้นสำหรับการรวบรวมสถิติ job accounting ถ้า:

- คุณไม่ต้องการจัดการอ็อบเจกต์เพิ่มเติมที่มีอยู่ในการทำเจอร์นัล.
- คุณไม่ต้องการข้อมูลรีจิสเตอร์ใดๆ นอกเหนือจากข้อมูลที่ให้ในข้อความ CPF1124 และ CPF1164, ซึ่งจะถูส่งโดยอัตโนมัติไปที่ไฟล์บันทึก QHST.
- คุณไม่ต้องการข้อมูล print accounting.

หมายเหตุ: สถิติบางอย่างที่บันทึกในข้อความ CPF1164 และ JB journal entries จะไม่ตรงกันทุกประการ. ที่เป็นเช่นนี้เนื่องจากปัจจัยหลักสองอย่าง: (1) มีการบันทึกสถิติ CPF1164 ก่อนหน้าสถิติ JB journal เพียงเล็กน้อย และ (2) ในแต่ละครั้งที่มีการเปลี่ยน accounting code, rounding จะเกิดขึ้นสำหรับฟิลด์บางฟิลด์, ในขณะที่ rounding จะเกิดขึ้นเพียงครั้งเดียวสำหรับข้อความ CPF1164.

การรักษาความปลอดภัยและ job accounting

เฉพาะเจ้าหน้าที่ดูแลความปลอดภัย (หรือโปรแกรมที่ได้รับสิทธิ์นี้) หรือผู้ใช้ที่มีสิทธิ์ *ALLOBJ และ *SECADM เท่านั้นที่สามารถเปลี่ยนค่ากำหนดของระบบ Journal accounting information (QACGLVL) ได้.

การเปลี่ยนแปลงจะมีผลบังคับใช้เมื่องานใหม่เข้าสู่ระบบ. ข้อจำกัดนี้ช่วยให้มั่นใจว่า job accounting มีผลบังคับใช้และเจ้าหน้าที่ดูแลความปลอดภัยปฏิบัติ IPL ระบบ, accounting entry ถูกบันทึกสำหรับงานของเจ้าหน้าที่ดูแลความปลอดภัย.

สิทธิ์ในการกำหนด job accounting codes

คุณสามารถ กำหนด job accounting codes ได้เฉพาะถ้าคุณมีสิทธิ์ในการใช้คำสั่ง Create User Profile (CRTUSRPRF), Change User Profile (CHGUSRPRF) หรือคำสั่ง Change Accounting Code (CHGACGCDE). คำสั่งนี้จำกัด การใช้ accounting codes และนำเสนอพื้นฐานสำหรับการตรวจสอบความถูกต้อง ของการเปลี่ยนแปลงใดๆ.

เฉพาะผู้ใช้ที่มีสิทธิ์พิเศษ *SECADM เท่านั้นที่สามารถใช้ คำสั่ง CRTUSRPRF และ CHGUSRPRF ได้. อย่างไรก็ตาม, เจ้าหน้าที่ดูแลความปลอดภัยสามารถมอบสิทธิ์ของตนได้โดยการสร้าง โปรแกรม CL, ซึ่งอนุญาตให้ผู้อื่นใช้โปรไฟล์ของเจ้าหน้าที่ดูแลความปลอดภัย และเปลี่ยนพารามิเตอร์ ACGCDE ในโปรไฟล์ผู้ใช้. จากนั้น ผู้ใช้แต่ละรายอาจมีสิทธิ์ในการใช้หนึ่งหรือหลายโปรแกรม CL.

พารามิเตอร์ ACGCDE ยังมีอยู่ใน อ็อบเจกต์รายละเอียดของงานด้วย, แต่คุณต้องมีสิทธิ์ในการใช้คำสั่ง CHGACGCDE เพื่อป้อนค่าอื่นที่ไม่ใช่ค่าดีฟอลต์ของ *USRPRF. CHGACGCDE ถูกจัดสงมาโดยมีสิทธิ์ PUBLIC เป็น *USE.

สิทธิ์ในการใช้งานคำสั่ง CHGACGCDE

หากคุณอนุญาตให้ผู้ใช้ใช้คำสั่ง Change Accounting Code (CHGACGCDE), ผู้ใช้นั้นสามารถ:

- สร้างหรือเปลี่ยนพารามิเตอร์ ACGCDE ในรายละเอียดของงาน. (ต้องใช้สิทธิ์ในการสร้างหรือเปลี่ยนรายละเอียดของงานด้วย.)
- เปลี่ยน accounting code ในงานปัจจุบันของตน.
- เปลี่ยน accounting code ของงานของคนอื่น ถ้าผู้ใช้มีสิทธิ์พิเศษ *JOBCTL ด้วย.

คุณสามารถนำเสนอการรักษาความปลอดภัยเพิ่มเติมได้โดยการใช้คำสั่ง CHGACGCDE ในโปรแกรม CL, ซึ่งใช้สิทธิ์ของเจ้าของโปรแกรม. สิทธิ์นี้อนุญาตให้ผู้ใช้ที่กำลังรันฟังก์ชันภายนอกสามารถใช้ฟังก์ชันที่ส่งผลต่อความปลอดภัยได้โดยไม่ต้องได้รับอนุญาตโดยตรงที่คำสั่ง CHGACGCDE.

accounting journal และ receivers ของเจอร์นัลถูกดำเนินการเช่นเดียวกับอ็อบเจกต์เจอร์นัลอื่นจากมุมมอง การรักษาความปลอดภัย. คุณต้องตัดสินใจว่าควรมีการให้สิทธิ์อะไร สำหรับ accounting journal และ journal receiver.

Journal entries สำหรับ job accounting

ระบบนำเสนอ journal entries ต่างๆ สำหรับข้อมูลชนิดต่างๆ ที่สามารถรวบรวมได้:

- Job resource accounting: job (JB) journal entry มีข้อมูลสรุปของ รีซอร์สที่ใช้สำหรับงานหรือสำหรับ accounting codes ต่างๆ ที่ใช้ในงาน.
- Printer file accounting:
 - Direct print (DP) journal entry: มีข้อมูลเกี่ยวกับ printer files ที่สร้างบน อุปกรณ์การพิมพ์ (nospooled).
 - Spooled print (SP) journal entry: มีข้อมูลเกี่ยวกับ printer files ที่สร้าง โดย print writer (spooled).

ข้อมูลฟิลด์ Job accounting journal entry:

หัวข้อนี้มีรายการของฟิลด์ที่อยู่ใน JB journal entry. สามารถดูข้อมูลเพิ่มเติมเกี่ยวกับฟิลด์ต่างๆ ได้ในไฟล์อ้างอิงของฟิลด์ QSYS/QAJBACG4 และ QSYS/QAJBACG44.

ตารางที่ 3.

ชื่อฟิลด์ (อักขระ 14)	Description	ฟิลด์แอตทริบิวต์	ความคิดเห็น
JAJOB	ชื่องาน	อักขระ (10)	
JAUSER	ผู้ใช้งาน	อักขระ (10)	
JANBR	หมายเลขงาน	Zoned (6,0)	
JACDE	Accounting code	อักขระ (15)	
JACPU	เวลาหน่วยการประมวลผลที่ใช้ (ในมิลลิวินาที)	Packed decimal (11,0)	เวลาหน่วยการประมวลผลไม่รวมการใช้หน่วยการประมวลผลและสถิติพรีนเตอร์สำหรับการสร้างบันทึกการใช้งาน.
JARTGS	จำนวนของขั้นตอนการเรอต์	Packed decimal (5,0)	
JAEDTE	งานที่เข้าสู่ระบบ - วันที่ Job entry (รูปแบบ mmdyy)	อักขระ (6)	
JAETIM	งานที่เข้าสู่ระบบ - เวลา Job entry (รูปแบบ hhmmss)	อักขระ (6)	
JASDTE	วันที่และเวลาการเริ่มงาน - วันที่เริ่มงาน (รูปแบบ mmdyy)	อักขระ (6)	สำหรับวันที่และเวลาที่งานเสร็จสมบูรณ์จาก journal entries, ให้ใช้ฟิลด์ JODATE และ JOTIME ที่เป็นส่วนหนึ่งของข้อมูล journal entry prefix มาตรฐาน. (ดูหนังสือคู่มือ Backup and Recovery สำหรับข้อมูลเพิ่มเติม เกี่ยวกับฟิลด์เหล่านี้.) หลังจากการจบระบบที่ไม่ปกติ, ฟิลด์เหล่านี้จะแสดง วันที่และเวลาปัจจุบัน ไม่ใช่ (ด้วยข้อความ CPF1164) เวลาจริงของ การจบระบบ.

ตารางที่ 3. (ต่อ)

ชื่อฟิลด์ (อักขระ 14)	Description	ฟิลด์แอดทริบิวต์	ความคิดเห็น
JASTIM	วันที่และเวลาการเริ่มงาน - เวลาเริ่มงาน (รูปแบบ hhhmmss)	อักขระ (6)	สำหรับวันที่และเวลาที่งานเสร็จสมบูรณ์จาก journal entries, ให้ใช้ฟิลด์ JODATE และ JOTIME ที่เป็นส่วนหนึ่งของข้อมูล journal entry prefix มาตรฐาน. (ดูหนังสือคู่มือ Backup and Recovery สำหรับข้อมูลเพิ่มเติม เกี่ยวกับฟิลด์เหล่านี้.) หลังจากการจบระบบที่ไม่ปกติ, ฟิลด์เหล่านี้จะแสดง วันที่และเวลา ปัจจุบัน ไม่ใช่ (ด้วยข้อความ CPF1164) เวลาจริงของ การจบระบบ.
JATRNT	เวลา transaction ทั้งหมด (ในวินาที)	Packed decimal (11,0)	เวลา transaction ทั้งหมดถูกเซตเป็น -1 เมื่อ: <ul style="list-style-type: none"> • เวลาถูกเซตย้อนหลัง. • เกิดโอเวอร์โฟลวในไฟล์เมื่อทำการคำนวณ. • ระบบปิดในขณะที่งานแอดคทีฟ.
JATRNS	จำนวนของ transactions	Packed decimal (11,0)	ไม่นับ transaction หลังสุด (SIGNOFF).
JAAUX	การดำเนินการ Synchronous auxiliary I/O และการดำเนินการฐานข้อมูล (รวมความบกพร่องการแบ่งหน้าสำหรับเหตุผลต่างๆ)	Packed decimal (11,0)	
JATYPE	ชนิดงาน	อักขระ (1)	ชนิดของงานที่บันทึกมีดังต่อไปนี้: <p>A งานแบบ autostart</p> <p>B งานแบ็คซ์ (รวมการสื่อสารและ MRT)</p> <p>I งานแบบโต้ตอบ</p> <p>M การมอนิเตอร์ระบบย่อย</p> <p>R Spooling reader</p> <p>W Spooling writer</p> <p>หมายเหตุ: ข้อมูลเหล่านี้เหมือนกับข้อมูลที่ใช้ใน ข้อความ CPF1164, ยกเว้นว่าข้อความ CPF1164 รวมข้อมูลงานระบบบางอย่างที่ไม่รวมอยู่ใน journal entries.</p>

ตารางที่ 3. (ต่อ)

ชื่อฟิลด์ (อักขระ 14)	Description	ฟิลด์แอดทริบิวต์	ความคิดเห็น
JACCDE	โค้ดความสมบูรณ์	Packed decimal (3,0)	<p>โค้ดความสมบูรณ์, ซึ่งคล้ายกับโค้ดที่ใช้สำหรับข้อความ CPF1164, มีดังนี้:</p> <p>000 การเสร็จสมบูรณ์ตามปกติ</p> <p>010 การเสร็จสมบูรณ์ตามปกติในระหว่างการจับที่ควบคุมหรือการจบบรรยากาศที่ควบคุม</p> <p>020 งานจบอย่างผิดปกติร้ายแรง</p> <p>030 งานจบอย่างผิดปกติ</p> <p>040 งานจบลงก่อนที่จะแอ็คทีฟ</p> <p>050 งานจบลงในขณะที่แอ็คทีฟ</p> <p>060 ระบบย่อยจบอย่างผิดปกติในขณะที่งานแอ็คทีฟ</p> <p>070 ระบบจบอย่างผิดปกติในขณะที่งานแอ็คทีฟ</p> <p>080 งานเสร็จสมบูรณ์ภายในขีดจำกัดเวลา</p> <p>090 งานถูกบังคับให้เสร็จสมบูรณ์หลังจากขีดจำกัดเวลาสิ้นสุดลง</p> <p>099 Accounting entry ที่เกิดจากคำสั่ง CHGACGCDE</p>
JALINE	จำนวนของบรรทัดพิมพ์	Packed decimal (11,0)	<p>จำนวนของบรรทัดพิมพ์ไม่สะท้อนถึงสิ่งที่จะถูกพิมพ์จริง. เนื่องจากสพูลไฟล์สามารถถูกยกเลิกและพิมพ์ที่ละลายก็อปปีได้. ข้อมูลใน JB journal entry สะท้อนให้เห็นเฉพาะสิ่งที่ถูกบันทึกโดย โปรแกรม. และไม่รวมบรรทัดใดๆ ที่บันทึกสำหรับบันทึกการใช้งาน. สามารถดูคำอธิบายเกี่ยวกับ DP และ SP printer file accounting data ได้ในภายหลังในบทนี้.</p>
JAPAGE	จำนวนของหน้าที่พิมพ์	Packed decimal (11,0)	
JAPRTF	จำนวนของไฟล์พิมพ์	Packed decimal (11,0)	
JADBPT	จำนวนของการดำเนินการบันทึกฐานข้อมูล	Packed decimal (11,0)	<p>จำนวนที่บันทึกสำหรับการดำเนินการ I/O ฐานข้อมูลไม่รวมการดำเนินการ I/O ที่เครื่องอ่านและ writers, หรือการดำเนินการ I/O ที่เกิดจาก คำสั่ง CL CPYSPLF, DSPSPLF, หรือ WRKSPLF. ถ้าใช้ SEQONLY(*YES), จำนวนเหล่านี้จะแสดงแต่ละบล็อกของเร็คคอร์ดที่อ่าน, ไม่ใช่จำนวนของแต่ละเร็คคอร์ดที่อ่าน.</p>
JADBGT	จำนวนของการดำเนินการอ่านฐานข้อมูล	Packed decimal (11,0)	<p>จำนวนที่บันทึกสำหรับการดำเนินการ I/O ฐานข้อมูลไม่รวมการดำเนินการ I/O ที่เครื่องอ่านและ writers, หรือการดำเนินการ I/O ที่เกิดจาก คำสั่ง CL CPYSPLF, DSPSPLF, หรือ WRKSPLF. ถ้าใช้ SEQONLY(*YES), จำนวนเหล่านี้จะแสดงแต่ละบล็อกของเร็คคอร์ดที่อ่าน, ไม่ใช่จำนวนของแต่ละเร็คคอร์ดที่อ่าน.</p>

ตารางที่ 3. (ต่อ)

ชื่อฟิลด์ (อักขระ 14)	Description	ฟิลด์แอตทริบิวต์	ความคิดเห็น
JADBUP	จำนวนของการดำเนินการอัปเดตฐานข้อมูล, ลบ FEOD, รีลีส, commit, และการทำ rollback	Packed decimal (11,0)	จำนวนที่บันทึกสำหรับการดำเนินการ I/O ฐานข้อมูลไม่รวมการดำเนินการ I/O ที่เครื่องอ่านและ writers, หรือการดำเนินการ I/O ที่เกิดจาก คำสั่ง CL CPYSPLF, DSPSPLF, หรือ WRKSPLF. ถ้าใช้ SEQONLY(*YES), จำนวนเหล่านี้จะแสดงแต่ละบล็อกของเรกคอร์ดที่อ่าน, ไม่ใช่จำนวนของแต่ละเรกคอร์ดที่อ่าน.
JACMPT	จำนวนของการดำเนินการบันทึกการสื่อสาร	Packed decimal (11,0)	จำนวนที่บันทึกสำหรับการดำเนินการ I/O การสื่อสาร ไม่รวม remote workstation activity. เมื่อ I/O ใช้สำหรับอุปกรณ์สื่อสาร, จำนวนจะรวมเฉพาะ activity ที่เกี่ยวข้องกับไฟล์ ICF.
JACMGT	จำนวนของการดำเนินการอ่านการสื่อสาร	Packed decimal (11,0)	จำนวนที่บันทึกสำหรับการดำเนินการ I/O การสื่อสาร ไม่รวม remote workstation activity. เมื่อ I/O ใช้สำหรับอุปกรณ์สื่อสาร, จำนวนจะรวมเฉพาะ activity ที่เกี่ยวข้องกับไฟล์ ICF.
JAACT	งานเวลาแอคทีฟ (ในมิลลิวินาที)	Packed decimal (11,0)	
JASPN	งานเวลาถูกหยุดชั่วคราว (ในมิลลิวินาที)	Packed decimal (11,0)	
JAEDTL	งาน timestamp เข้าสู่ระบบ (mmdyyyyhhmmss)	อักขระ (14)	
JAESTL	งาน timestamp เริ่มต้น (mmdyyyyhhmmss)	อักขระ (14)	
JAAIO	การดำเนินการ Asynchronous I/O สำหรับฐานข้อมูลและไม่ใช่ฐานข้อมูล	Packed decimal (11,0)	
JAXCPU	เวลา CPU เพิ่มเติมที่ใช้	Packed decimal (29,0)	
JAXSIO	การดำเนินการ synchronous auxiliary I/O เพิ่มเติม	Packed decimal (29,0)	
JAXAIO	การดำเนินการ asynchronous auxiliary I/O เพิ่มเติม	Packed decimal (29,0)	
JAXDBP	จำนวนเพิ่มเติมของ database puts	Packed decimal (29,0)	
JAXDBG	จำนวนเพิ่มเติมของ database gets	Packed decimal (29,0)	

ตารางที่ 3. (ต่อ)

ชื่อฟิลด์ (อักขระ)			
14)	Description	ฟิลด์แอตทริบิวต์	ความคิดเห็น
JAXDBU	จำนวนเพิ่มเติมของการอัปเดตและการลบฐานข้อมูล	Packed decimal (29,0)	
JAXLIN	จำนวนเพิ่มเติมของบรรทัดที่พิมพ์	Packed decimal (29,0)	
JAXPAG	จำนวนเพิ่มเติมของหน้าที่พิมพ์	Packed decimal (29,0)	
JAXPRT	จำนวนของไฟล์พิมพ์	Packed decimal (29,0)	

Direct Print (DP) และ spooled print (SP) printer file accounting data:

accounting code ที่ใช้สำหรับ DP หรือ SP journal entry คือ accounting code ของงานในเวลาไฟล์ถูกปิด. ในบางครั้ง DP หรือ SP entry ถูกสร้างขึ้น ก่อนที่ไฟล์ถูกปิด (เช่นเมื่อ writer ซึ่งกำลังสร้างไฟล์ SCHEDULE(*IMMED) ล้นสุดลง). เมื่อเกิดกรณีเช่นนี้ จะใช้ accounting code ปัจจุบันของงาน .

DP หรือ SP journal entry ถูกสร้างขึ้นสำหรับแต่ละไฟล์ที่พิมพ์. ถ้าบันทึกการใช้งานถูกสพูล แล้วพิมพ์, SP entry จะถูกสร้างขึ้นสำหรับบันทึกการใช้งาน. และ, SP entry ถูกบันทึกสำหรับดิสเก็ตสพูลไฟล์ที่ redirected ไปที่พริ้นเตอร์โดย print writer.

ข้อมูล DP accounting journal:

ไฟล์ QSYS/QAPTACG5 ประกอบด้วยฟิลด์ที่จะถูกใช้ใน DP journal entry. ตารางในหัวข้อนี้แสดงรายการฟิลด์สามฟิลด์และแอตทริบิวต์ของฟิลด์.

ตารางที่ 4.

Field Name	Description	Field Attributes
JAJOB	ชื่องาน	อักขระ (10)
JAUSER	ผู้ใช้งาน	อักขระ (10)
JANBR	หมายเลขงาน	Zoned (6,0)
JACDE	Accounting code	อักขระ (15)
JADFN	ชื่อไฟล์อุปกรณ์	อักขระ (10)
JADFNL	ไลบรารีที่ไฟล์อุปกรณ์ถูกจัดเก็บอยู่	อักขระ (10)
JADEVN	ชื่ออุปกรณ์	อักขระ (10)
JADEVT	ชนิดของอุปกรณ์	อักขระ (4)
JADEVM	รุ่นอุปกรณ์	อักขระ (4)
JATPAG	จำนวนทั้งหมดของหน้าพิมพ์ที่จัดทำ	Packed decimal (11,0)

ตารางที่ 4. (ต่อ)

Field Name	Description	Field Attributes
JATLIN	จำนวนทั้งหมดของบรรทัดพิมพ์ที่จัดทำ	Packed decimal (11,0)
JASPFN	พื้นที่เปล่าเสมอ	อักขระ (10)
JASPNB	พื้นที่เปล่าเสมอ	อักขระ (4)
JAOPTY	พื้นที่เปล่าเสมอ	อักขระ (1)
JAFMTP	พื้นที่เปล่าเสมอ	อักขระ (10)
JABYTE	ศูนย์เสมอ	Packed decimal (15,0)
JAUSRD	ข้อมูลผู้ใช้	อักขระ (10)
JALSPN	พื้นที่เปล่าเสมอ	อักขระ (6)
JASPSY	พื้นที่เปล่าเสมอ	อักขระ (8)
JASPDT	พื้นที่เปล่าเสมอ	อักขระ (7)
JASPTM	พื้นที่เปล่าเสมอ	อักขระ (6)
JADFASP	พื้นที่เปล่าเสมอ	อักขระ (10)

ข้อมูล SP accounting journal:

หัวข้อนี้มีตารางที่แสดงรายการของฟิลด์ (พบในไฟล์ QSYS/QAPTACG5) ใน SP journal entry.

หมายเหตุ: ข้อมูล SP accounting journal คล้ายกับข้อมูลที่ให้ในข้อมูล DP accounting journal ยกเว้นว่ามีการรวมชื่อสพูลไฟล์, หมายเลขสพูลไฟล์, ระดับความสำคัญของเอาต์พุต, form type, และจำนวนทั้งหมดของไบต์ของข้อมูลการควบคุมและข้อมูลการพิมพ์ ที่ส่งไปที่พริ้นเตอร์. SP journal entry จะไม่ถูกบันทึก ถ้าสพูลไฟล์ ถูกลบออกก่อนที่ writer จะเริ่มบันทึกไฟล์ไปที่อุปกรณ์.

ตารางที่ 5.

ชื่อฟิลด์	Description	ฟิลด์แอตทริบิวต์
JAJOB	ชื่องาน	อักขระ (10)
JAUSER	ผู้ใช้งาน	อักขระ (10)
JANBR	หมายเลขงาน	Zoned (6,0)
JACDE	Accounting code	อักขระ (15)
JADFN	ชื่อไฟล์อุปกรณ์	อักขระ (10)
JADFNL	ไลบรารีที่ไฟล์อุปกรณ์ถูกจัดเก็บอยู่	อักขระ (10)
JADDEVN	ชื่ออุปกรณ์	อักขระ (10)
JAD EVT	ชนิดของอุปกรณ์	อักขระ (4)

ตารางที่ 5. (ต่อ)

ชื่อฟิลด์	Description	ฟิลด์แอตทริบิวต์
JADEVM	รุ่นอุปกรณ์	อักขระ (4)
JATPAG	จำนวนทั้งหมดของหน้าพิมพ์ที่จัดทำ	Packed decimal (11,0)
JATLIN	จำนวนทั้งหมดของบรรทัดพิมพ์ที่จัดทำ	Packed decimal (11,0)
JASPFN	ชื่อสพูลไฟล์	อักขระ (10)
JASPNB	หมายเลขสพูลไฟล์	อักขระ (4)
JAOPTY	ระดับความสำคัญเอาต์พุต	อักขระ (1)
JAFMTP	Form type	อักขระ (10)
JABYTE	จำนวนทั้งหมดของไบต์ที่ส่งไปที่พริเตอร์	Packed decimal (15,0)
JAUSRD	ข้อมูลผู้ใช้	อักขระ (10)
JALSPN	หมายเลขสพูลไฟล์	อักขระ (6)
JASPSY	ชื่อระบบของงานสพูลไฟล์	อักขระ (8)
JASPDT	วันที่สร้างสพูลไฟล์ (รูปแบบ cyymmdd)	อักขระ (7)
JASPTM	เวลาที่สร้างสพูลไฟล์ (รูปแบบ hhmmss)	อักขระ (6)
JADFASP	ชื่อ ASP สำหรับไลบรารีไฟล์อุปกรณ์	อักขระ (10)

หมายเหตุ:

- ระบบพยายามบันทึกจำนวนที่แท้จริงของหน้า, บรรทัด, และไบต์ที่พิมพ์, แต่ถ้า writer ถูกยกเลิก *IMMED หรือถูกคืนจากข้อผิดพลาดอุปกรณ์ (เช่น เมื่อจบฟอร์ม), ระบบจะไม่สามารถกำหนดจำนวนที่แน่นอนของหน้า, บรรทัด, และไบต์ที่พิมพ์.
- หน้าและบรรทัดพิเศษที่เกิดขึ้นจากการจัดตำแหน่งจะไม่ถูกรวม ในการนับหน้า, บรรทัด, และไบต์.
- ถ้าสพูลไฟล์เข้าสู่สถานะ WTR (แต่ถูกเซตเป็น MSGW) หรือถ้า ไฟล์ถูกลบออกในขณะที่อยู่ในสถานะ MSGW, SP journal entry จะปรากฏขึ้นใน DP accounting journal เพื่อบ่งชี้ว่ามี 0 หน้าและ 0 บรรทัดที่พิมพ์.
- ในขณะที่ใช้พริเตอร์ที่ตั้งค่า AFP(*YES), ถ้าคุณลบหรือพักไฟล์ในทันที หลังจากที่ไฟล์มีหน้าที่พิมพ์, SP entry สำหรับไฟล์นั้นอาจบ่งชี้ 0 หน้า และ 0 บรรทัดที่พิมพ์แม้ว่าได้มีการพิมพ์บางหน้าไปแล้ว.
- การนับหน้า, บรรทัด, และไบต์สำหรับงานและตัวแบ่งไฟล์ถูกรวมอยู่ในการนับ สำหรับไฟล์ที่เชื่อมโยงด้วย.
- เมื่อไฟล์ IPDS™ มีกราฟิกส์หรือบาร์โค้ด และถูกส่งไปที่พริเตอร์ IPDS ที่ไม่สนับสนุนกราฟิกส์หรือ บาร์โค้ด, การนับหน้า, บรรทัด, และไบต์จะรวมกราฟิกส์และบาร์โค้ด ที่ไม่ได้พิมพ์ด้วย.
- ถ้า printer configuration เป็น AFP(*YES), ฟิลด์สำหรับจำนวนทั้งหมดของ บรรทัดพิมพ์ที่จัดทำคือศูนย์. ฟิลด์จำนวนทั้งหมดของหน้าจัดทำคือ แก้ว.

เกี่ยวกับ accounting code

initial accounting code (ยาวได้ถึง 15 อักขระ) ของงาน ถูกกำหนดโดยค่าของพารามิเตอร์ ACGCDE (accounting code) ใน รายละเอียดของงานและโปรไฟล์ผู้ใช้สำหรับงาน.

เมื่องานเริ่มต้น, รายละเอียดของงานจะถูกกำหนดที่งาน. อีอบเจ็กต์ รายละเอียดของงานมีค่าสำหรับพารามิเตอร์ ACGCDE . ถ้าใช้ค่าดีฟอลต์ของ *USRPRF, จะใช้ accounting code ในโปรไฟล์ผู้ใช้ของงาน.

หมายเหตุ: เมื่องานถูกเริ่มโดยใช้คำสั่ง Submit Job (SBMJOB), accounting code ของงานจะเหมือนกับโค้ดของงานของผู้ส่ง.

คุณสามารถ เปลี่ยน accounting code หลังจากทำงานได้เข้าสู่ระบบแล้วได้โดยใช้คำสั่ง Change Accounting Code (CHGACGCDE).

คำสั่ง CRTUSRPRF และ CHGUSRPRF สนับสนุนพารามิเตอร์ ACGCDE . ค่าดีฟอลต์คือ *BLANK. ถ้างานทั้งหมดสำหรับผู้ใช้หนึ่งจะถูกบันทึกภายใต้หนึ่ง accounting code, จะต้องเปลี่ยนเฉพาะโปรไฟล์ผู้ใช้นั้น. คุณสามารถเปลี่ยน accounting codes สำหรับรายละเอียดของงานเฉพาะได้โดยการระบุ accounting code สำหรับพารามิเตอร์ ACGCDE บนคำสั่ง CRTJOB และ CHGJOB . คำสั่ง CHGACGCDE ยังอนุญาตให้ใช้ accounting codes อื่นในงานเดี่ยวด้วย.

คำสั่ง Retrieve Job Attributes (RTVJOBA) และ API's ที่ดึงข้อมูลแอตทริบิวต์งานช่วยให้คุณเข้าถึง accounting code ปัจจุบันในโปรแกรม CL.

Resource accounting

ข้อมูล Job resource accounting ถูกสรุปอยู่ใน job (JB) journal entry เมื่องานเสร็จสมบูรณ์. ยิ่งไปกว่านั้น, ระบบจะสร้าง JB journal entry ที่สรุปริชอร์สที่ใช้ในแต่ละครั้งที่ใช้คำสั่ง Change Accounting Code (CHGACGCDE) . JB journal entry รวมถึง:

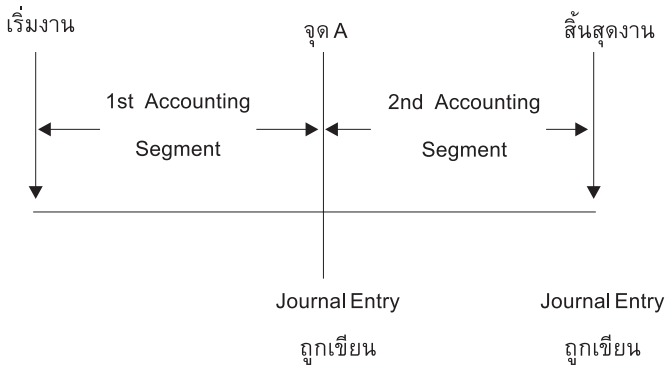
- ชื่องานที่ถูกต้องทั้งหมด
- Accounting code สำหรับ accounting segment ที่เพ็งจบ
- เวลาของหน่วยการประมวลผล
- จำนวนของขั้นตอนการเรด
- วันที่และเวลาที่งานเข้าสู่ระบบ
- วันที่และเวลาที่งานเริ่มต้น
- เวลา transaction ทั้งหมด (รวมเวลาการให้บริการ, เวลาที่ไม่ถูกต้อง, และเวลา แอ็คทีฟ)
- จำนวนของ transactions สำหรับงานแบบโต้ตอบทั้งหมด
- การดำเนินการ Auxiliary I/O
- ชนิดงาน
- โค้ดความสมบูรณ์ของงาน
- จำนวนของบรรทัด, หน้า, และไฟล์ของพริเตอร์ที่สร้าง ถ้าสพูลหรือพิมพ์โดยตรง
- จำนวนของการอ่าน, บันทึก, อัปเดต, และลบไฟล์ฐานข้อมูล
- จำนวนของการดำเนินการอ่านและบันทึกไฟล์ ICF

หมายเหตุ: นอกจากนี้ยังสามารถเข้าใช้งานข้อมูล job accounting บางอย่างได้โดยใช้ข้อความ CPF1124 และ CPF1164 ที่อยู่ในไฟล์บันทึก QHST.

ข้อมูล Resource accounting

เมื่อวิเคราะห์ journal entries, สิ่งสำคัญคือการทำความเข้าใจว่า journal entries จะถูกบันทึกอย่างไรและเมื่อไร. JB journal entry จะถูกบันทึกที่ job accounting journal สำหรับงานในทุกครั้งที่ job accounting code ถูกเปลี่ยนแปลงและเมื่องานจบลง. ดังนั้น, งานหนึ่งอาจมีหลาย journal entries ได้.

แต่ละ resource accounting journal entry มีข้อมูลเกี่ยวกับรีซอร์สที่ใช้ ในขณะที่ accounting code ก่อนหน้านี้มีผลบังคับใช้อยู่. ให้พิจารณาตัวอย่างต่อไปนี้:



RZAKS550-0

รูปที่ 5. ตัวอย่างข้อมูล Resource accounting

ที่จุด A, มีการออกคำสั่ง CHGACGCDE . accounting code ถูกเปลี่ยนแปลงและมีการส่ง JB journal entry ไปที่เจอร์นัล. JB journal entry มีข้อมูลสำหรับ accounting segment แรก. เมื่องาน จบลง, JB entry ที่สองจะถูกทำขึ้นสำหรับงานที่มีข้อมูลของ accounting segment ที่สอง.

ถ้าไม่มีการเปลี่ยนแปลง job accounting code ในระหว่างที่มีงานอยู่, JB entry เดียวจะสรุปรีซอร์สทั้งหมดที่งานใช้. ถ้ามีการเปลี่ยนแปลง job accounting code ในระหว่างที่มีงานอยู่, คุณต้อง ใส่เพิ่มฟิลด์ในหลาย JB entries เพื่อกำหนดรีซอร์สทั้งหมด ที่งานใช้. การสร้างบันทึกการใช้งานไม่นับการใช้ หน่วยการประมวลผลสำหรับงาน หรือเอาต์พุตที่พิมพ์ของงานใน JB accounting entries. อย่างไรก็ตาม, ถ้าคุณกำลังใช้ print file accounting, บันทึกการใช้งานที่พิมพ์ จะถูกรวมอยู่ใน printer file journal entries.

งานแบบ prestart และ job accounting

ถ้าระบบของคุณใช้ job accounting, โปรแกรมงานแบบ prestart ควรจะ รันคำสั่ง Change Prestart Job (CHGPJ) ด้วยคำขอ การเริ่ม โปรแกรมสำหรับพารามิเตอร์ accounting code (CHGPJ ACGCDE(*PGMSTRRQS)) โดยทันที หลังจากที่คุณดำเนินการ เริ่มโปรแกรมแนบกับงานแบบ prestart.

action นี้เปลี่ยน accounting code เป็นค่าที่ระบุในโปรแกรมผู้ใช้ที่เชื่อมโยงกับคำขอการเริ่มโปรแกรม. ในทันทีก่อนที่โปรแกรม จะเสร็จสิ้นการจัดการคำขอการเริ่มโปรแกรม, โปรแกรมควรจะ รันคำสั่ง Change Prestart Job (CHGPJ) โดยมี Prestart Job Entry เป็นค่าสำหรับพารามิเตอร์ accounting code (CHGPJ ACGCDE(*PJE)). สิ่งนี้จะเปลี่ยน accounting code กลับเป็นค่าที่ ระบุในรายละเอียดของงาน ของ prestart job entry.

การประมวลผลงานระบบสำหรับ job accounting

งานระบบที่คุณควบคุม (เช่น, เครื่องอ่านและ writer) ถูกกำหนด accounting code เป็น *SYS. งานระบบอื่นที่คุณไม่ได้ควบคุม (เช่น, QSYSARB, QLUS, SCPF) จะไม่ได้รับ journal entry.

หมายเหตุ: คุณไม่สามารถใช้คำสั่ง Change Accounting Code (CHGACGCDE) เพื่อเปลี่ยน accounting code ของการมอ นิเตอร์ระบบย่อยหรือเครื่องอ่าน หรือ writer. อย่างไรก็ตาม, คุณสามารถ, เปลี่ยน accounting code ของเครื่องอ่าน หรือ writer ได้ โดยการเปลี่ยนรายละเอียดของงานที่ได้มาจาก IBM และ โปรไฟล์ผู้ใช้ จากนั้นเริ่มเครื่องอ่านหรือ writer อีกครั้งหนึ่ง.

การประมวลผลแบบแบ็ตช์และ job accounting

งานแบ็ตช์ใดๆ ที่ถูกส่งโดยใช้คำสั่ง Submit Job (SBMJOB) จะใช้ accounting code เหมือนกันกับงานที่ส่งงานแบ็ตช์โดยอัตโนมัติ. เมื่อใช้คำสั่ง SBJOB, accounting codes ไม่สามารถถูกแทนที่ค่าเดิมไม่ว่า job description entry จะลง ได้ตัวอย่าง ไรก็ตาม.

ถ้าคุณต้องการให้งานแบ็ตช์ดำเนินการภายใต้ accounting code อื่นที่ไม่ใช่โค้ด ของงานที่ส่ง, ควรใช้คำสั่ง Change Accounting Code (CHGACGCDE) ในเวลาใดเวลาหนึ่งดังนี้:

- ก่อนและหลังจากการใช้คำสั่ง SBJOB
- ทันทีโดยงานแบ็ตช์.

งานแบ็ตช์ที่ส่งโดยใช้เครื่องอ่านหรือคำสั่ง Submit Database Job (SBMDBJOB) จะใช้ accounting code ที่ระบุในรายละเอียด ของงานสำหรับงานแบ็ตช์. ถ้ารายละเอียดของงานระบุ ACGCDE(*USRPRF), accounting code จะถูกนำมาจากโปรไฟล์ผู้ใช้ ที่ใช้สำหรับงาน.

การประมวลผลแบบโต้ตอบและ job accounting

ถ้างานแบบโต้ตอบมีอ็อปชันสำหรับผู้ใช้จำนวนคงที่ และ แต่ละอ็อปชันมี accounting code ที่กำหนดให้, คุณอาจต้องการ กำหนดโค้ดใหม่ โดยอัตโนมัติเมื่อผู้ใช้ร้องขอการทำงานบนฟังก์ชันใหม่.

วิธีการปกติสำหรับเมนูอ็อปชันคือ การร้องขอพื้นที่ฟังก์ชันใหม่. จากนั้นจะใช้คำสั่ง Change Accounting Code (CHGACGCDE) ภายในโปรแกรม CL และค่างานที่ใช้สำหรับ accounting code ก่อนหน้านี้จะถูกสรุปใน JB accounting journal entry.

ถ้าผู้ใช้มีงานที่ได้รับมอบหมายหลายอย่าง ซึ่งเฉพาะผู้ใช้เท่านั้นที่ทราบ accounting codes ที่จะใช้, คุณสามารถ:

- ให้สิทธิในการป้อนคำสั่ง CHGACGCDE แก่ผู้ใช้.
- เขียนโปรแกรมเพื่อพร้อมตัวผู้ใช้สำหรับ accounting code.

หมายเหตุ: สำหรับงาน source pass-through, ข้อมูล job accounting ไม่รวมงาน target pass-through job. สำหรับงาน target pass-through, ข้อมูล job accounting ไม่รวมงานแบ็ตช์การสื่อสารที่เชื่อมโยง.

Printer file accounting

journal entries สำหรับ printer file accounting มีอยู่สองชนิด; DP for nonspoiled printer files และ SP for spoiled printer files. journal entry ทั้งสองชนิด แบ่งใช้รูปแบบ journal entry เดียวกัน แม้ว่าข้อมูลบางอย่างจะพร้อมใช้งาน เฉพาะใน SP entry เท่านั้น. DP และ SP journal entries มีข้อมูลเช่น :

- ชื่องานที่ถูกต้องทั้งหมด

- Accounting code
- ชื่อไฟล์อุปกรณ์และไลบรารี
- ชื่ออุปกรณ์
- ชนิดของอุปกรณ์และรุ่น
- จำนวนทั้งหมดของหน้าและบรรทัดที่พิมพ์. ถ้ามีหลายก็อปปี, ข้อมูลเป็นผลรวมของก็อปปีทั้งหมด
- ชื่อสพูลไฟล์ (เฉพาะใน SP entry)
- หมายเลขสพูลไฟล์ (เฉพาะใน SP entry)
- ระดับความสำคัญเอาต์พุต (เฉพาะใน SP entry)
- From type (เฉพาะใน SP entry)
- Form type (เฉพาะใน SP entry)
- จำนวนทั้งหมดของไบต์ของข้อมูลการควบคุมและข้อมูลการพิมพ์ที่ส่งไปที่ อุปกรณ์การพิมพ์. ถ้ามีหลายก็อปปี, นี้เป็นผลรวมของก็อปปีทั้งหมด.(นี้ใช้เฉพาะกับ SP entry เท่านั้น.)

DP และ SP journal entries เกิดขึ้นเมื่อไฟล์ถูกพิมพ์. ถ้าไม่เคยพิมพ์สพูลไฟล์, จะไม่มี SP journal entry ปรากฏขึ้น.

การจัดการงาน

หน้าที่สำคัญของผู้ควบคุมระบบหรือผู้ดูแลระบบ, คือการดูแลให้เซิร์ฟเวอร์ทำงานโดยราบรื่น. นั่นหมายถึงการมอนิเตอร์, จัดการ, และตรวจตราดูว่างาน, คิวงาน, ระบบย่อย, พูลหน่วยความจำ, บันทึกการใช้งาน, และเอาต์พุตคิวทำงานได้อย่างถูกต้อง.

หัวข้อในส่วนนี้นำเสนอข้อมูลเกี่ยวกับชนิดของภารกิจการจัดการระบบงาน รายวัน และภารกิจอื่นที่คุณจะต้องทำบนเซิร์ฟเวอร์ iSeries ของคุณ. ในหัวข้อย่อยแต่ละอันจะบอกถึงความสำคัญของภารกิจแต่ละอย่าง, รวมถึงวิธีการทำงานให้ภารกิจสำเร็จอีกด้วย.

การเปลี่ยนโปรแกรมเริ่มต้น IPL

สร้างโปรแกรมเริ่มต้นที่จะเปลี่ยนรีซอร์สของระบบ, และรีซอร์สและแอ็ททริบิวต์ที่กำหนดที่ระบบ, ซึ่งจะถูกรวมในระหว่าง IPL. โดยปกติ, ระบบย่อย, writers, และ Operational Assistant จะถูก เรียกทำงานโดยโปรแกรมนี้.

งานแบบ autostart ในระบบย่อยการควบคุมจะถ่ายโอนการควบคุม ไปที่โปรแกรมที่ถูกระบุในค่ากำหนดของระบบ start-up program to set up system (QSTRUPPGM). คุณสามารถปรับแต่งโปรแกรมนี้ได้.

คุณสามารถสร้าง โปรแกรมของคุณเองและเปลี่ยนค่ากำหนดของระบบ Start-up program to set up system (QSTRUPPGM) เป็นชื่อโปรแกรมนั้น. หรือ, คุณสามารถใช้โปรแกรมที่จัดส่งให้ที่ชื่อ QSTRUP ใน QSYS เป็นพื้นฐานในการสร้างโปรแกรมของคุณเอง. ถ้าต้องการทำเช่นนั้น:

1. ให้ตั้งข้อมูลซอร์สของโปรแกรมที่จัดส่งให้โดยใช้คำสั่ง RTVCLSRC (เช่น, RTVCLSRC PGM(QSYS/QSTRUP) SRCFILE(YOURLIB/YOURFILE)).
2. เปลี่ยนโปรแกรม.
3. สร้างโปรแกรมโดยใช้คำสั่ง CRTCLPGM, และวางโปรแกรมเข้าใน ไลบรารีของคุณเอง.
4. ทดสอบโปรแกรมเพื่อให้แน่ใจว่าโปรแกรมใช้งานได้.

5. เปลี่ยนค่ากำหนดของระบบ Start-up program to set up system (QSTRUPPGM) เป็นชื่อโปรแกรมและไลบรารีที่คุณระบุบนคำสั่ง CRTCLPGM.

Code example disclaimer

IBM ให้ลิขสิทธิ์ไลเซนส์แบบไม่เฉพาะแก่คุณเพื่อใช้ตัวอย่างโค้ดการทำ โปรแกรมมิง ซึ่งคุณสามารถสร้างฟังก์ชันคล้ายกันที่ปรับตามความต้องการของคุณ จากตัวอย่างนั้น.

| SUBJECT TO ANY STATUTORY WARRANTIES WHICH CANNOT BE EXCLUDED, IBM, ITS PROGRAM
 | DEVELOPERS AND SUPPLIERS MAKE NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR IMPLIED,
 | INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY,
 | FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, REGARDING THE PROGRAM OR
 | TECHNICAL SUPPORT, IF ANY.

| UNDER NO CIRCUMSTANCES IS IBM, ITS PROGRAM DEVELOPERS OR SUPPLIERS LIABLE FOR ANY OF
 | THE FOLLOWING, EVEN IF INFORMED OF THEIR POSSIBILITY:

- | 1. LOSS OF, OR DAMAGE TO, DATA;
- | 2. DIRECT, SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES, OR FOR ANY ECONOMIC
 | CONSEQUENTIAL DAMAGES; OR
- | 3. LOST PROFITS, BUSINESS, REVENUE, GOODWILL, OR ANTICIPATED SAVINGS.

| SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF DIRECT, INCIDENTAL, OR
 | CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT
 | APPLY TO YOU.

ซอร์สสำหรับ CL Start-up Program

อ็อบเจกต์	คำสั่ง	CL Program Source
-----------	--------	-------------------

QSTRUP	CRTCLPGM	<pre> PGM DCL VAR(&STRWTRS) TYPE(*CHAR) LEN(1) DCL VAR(&CTLSBSD) TYPE(*CHAR) LEN(20) DCL VAR(&CPYR) TYPE(*CHAR) LEN(90) VALUE('+ 5722-SS1 (C) COPYRIGHT IBM CORP 1980, 2000. + LICENSED MATERIAL - PROGRAM PROPERTY OF IBM') QSYS/STRSBS SBSD(QSERVER) MONMSG MSGID(CPF0000) QSYS/STRSBS SBSD(QUSRWRK) MONMSG MSGID(CPF0000) QSYS/RLSJOBQ JOBQ(QGPL/QS36MRT) MONMSG MSGID(CPF0000) QSYS/RLSJOBQ JOBQ(QGPL/QS36EVOKE) MONMSG MSGID(CPF0000) QSYS/STRCLNUP MONMSG MSGID(CPF0000) QSYS/RTVSYSVAL SYSVAL(QCTLSBSD) RTNVAR(&CTLSBSD) IF ((&CTLSBSD *NE 'QCTL QSYS ') + *AND (&CTLSBSD *NE 'QCTL QGPL ')) GOTO DONE QSYS/STRSBS SBSD(QINTER) MONMSG MSGID(CPF0000) QSYS/STRSBS SBSD(QBATCH) MONMSG MSGID(CPF0000) QSYS/STRSBS SBSD(QCMN) MONMSG MSGID(CPF0000) DONE: QSYS/STRSBS SBSD(QSPL) MONMSG MSGID(CPF0000) QSYS/RTVSYSVAL SYSVAL(QSTRPRTWTR) RTNVAR(&STRWTRS) IF (&STRWTRS = '0') GOTO NOWTRS CALL PGM(QSYS/QWCSWTRS) MONMSG MSGID(CPF0000) NOWTRS: RETURN CHGVAR VAR(&CPYR) VALUE(&CPYR) ENDPGM </pre>
--------	----------	---

ค่ากำหนดของระบบ Start-up program to set up system (QSTRUPPGM)

Start-up program to set up system (QSTRUPPGM) คือโปรแกรม เริ่มต้น. ค่านี้ระบุชื่อของโปรแกรมที่จะถูกเรียกจากงานแบบ autostart เมื่อระบบย่อยการควบคุมเริ่มต้น. โปรแกรมนี้ปฏิบัติฟังก์ชันการตั้งค่า, เช่น การเริ่ม ระบบย่อยและพรินเตอร์. ค่ากำหนดของระบบนี้สามารถเปลี่ยนได้โดยเจ้าหน้าที่ดูแลความ ปลอดภัยหรือโดยบุคคลที่มีสิทธิเจ้าหน้าที่ดูแลความปลอดภัยเท่านั้น. การเปลี่ยนแปลงค่ากำหนดของ ระบบนี้จะมีผลบังคับใช้ในการทำ IPL ครั้งถัดไป. QSTRUPPGM สามารถมีค่าดังนี้:

- 'QSTRUP QSYS': โปรแกรมที่ระบุถูกรัน อันเป็นผลมาจากการถ่ายโอน การควบคุมไปที่โปรแกรมจากงานแบบ autostart ในระบบย่อยการควบคุม.
- '*NONE': งานแบบ autostart สิ้นสุดตามปกติโดยไม่มีการเรียกโปรแกรม.

โปรแกรมเริ่มต้นดีฟอลต์ QSYS/QSTRUP จะทำดังต่อไปนี้:

- เริ่มระบบย่อย QSPL สำหรับงานที่สพูล
- เริ่มระบบย่อย QSERVER สำหรับงานไฟล์เซิร์ฟเวอร์
- เริ่มระบบย่อย QUSRWRK สำหรับงานผู้ใช้
- รีลีสคิวงาน QS36MRT และ QS36EVOKE ถ้าคิวงานถูกพักไว้ (คิวงานเหล่านี้ ถูกใช้โดยสภาวะแวดล้อม System/36)
- เริ่มการล้าง Operational Assistant, ถ้าทำได้
- เริ่มเครื่องเขียนพรินเตอร์ทั้งหมดยกเว้นว่าผู้ใช้ได้ระบุไม่ให้ทำบนจอแสดงผล IPL Options
- ถ้าระบบย่อยการควบคุมคือ QCTL, โปรแกรมจะเริ่มระบบย่อย QINTER, QBATCH, และ QCMN

ชนิด	ความยาว	Shipped Value CL
อักขระ	20	'QSTRUP QSYS'

ดูค่ากำหนดของระบบ Start-up program to set up system (QSTRUPPGM) สำหรับข้อมูลรายละเอียดเพิ่มเติม.

การเรียกโปรแกรมการกู้คืน IPL พิเศษ

เพื่อเรียกโปรแกรมการกู้คืนพิเศษถ้า IPL ตรวจสอบว่า ระบบก่อนหน้านี้อันสิ้นสุดลงอย่างผิดปกติ, คุณสามารถใส่เพิ่ม autostart job entry ที่ subsystem description ของระบบย่อยการควบคุม.

โปรแกรมนี้ตรวจสอบค่ากำหนดของระบบ Previous system ending status (QABNORMSW). สำหรับการสิ้นสุดระบบตามปกติ, ค่าของ QABNORMSW คือ '0', และสำหรับการสิ้นสุดระบบอย่างผิดปกติ, ค่าของ QABNORMSW คือ '1'. ทางเลือกคือ ทิ้งข้อความและเริ่มระบบย่อยอื่นเมื่อฟังก์ชันการกู้คืนของคุณ เสร็จสมบูรณ์.

```

1.00 /* SPCRECOV - Autostart โปรแกรมเพื่อเรียกโปรแกรมการกู้คืนพิเศษ */
2.00          PGM
3.00          DCL      &QABNORMSW *CHAR LEN(1)
4.00          RTVSYSVAL SYSVAL(QABNORMSW) RTNVAR(&QABNORMSW)
5.00          IF      (&QABNORMSW *EQ '1') DO /* Recover */
6.00          SNDPGMMSG MSG('โปรแกรมการกู้คืนทำงานอยู่-ไม่ +
7.00                      เริ่มระบบย่อยจนกว่ามีการแจ้ง') +
8.00                      TOMSGQ(QSYSOPR)
9.00          CALL    RECOVERY
10.00         SNDPGMMSG MSG('การกู้คืนเสร็จสมบูรณ์-งานอาจเริ่มได้') +
11.00         TOMSGQ(QSYSOPR)
12.00         ENDDO /* Recover */
13.00         ENDPGM

```

การมอนิเตอร์ activity ระบบ

การมอนิเตอร์ activity ระบบเป็นภารกิจประจำวันอันหนึ่งที่มีความสำคัญของผู้ดูแล. การมอนิเตอร์การไหลของงานผ่านระบบเป็นข้อมูลขั้นหนึ่งที่คุณควรได้รับการมอนิเตอร์อยู่เป็นประจำทุกวัน. คุณสามารถมอนิเตอร์ได้หลายวิธี, เช่นการใช้ iSeries Navigator และ iSeries Navigator Management Central.

ถ้าใช้แบบจำลองที่เครื่องบนของจอแสดงผล Work with System Status (WRKSYSSTS) ในอินเทอร์เฟซแบบอักษร, หน้าต่าง System Status นำเสนอวิธีที่ง่ายและรวดเร็วในการตรวจสอบสถานะของระบบ. Management Central ช่วยให้คุณสามารถ มอนิเตอร์ฟังก์ชันได้ละเอียดมากขึ้นโดยการใช้การมอนิเตอร์ระบบ.

คุณสามารถเข้าใช้งานหน้าต่าง System Status จากโพลเดอร์ System หรือโพลเดอร์ Work Management .

ถ้าต้องการเรียก System Status จากโพลเดอร์ System :

1. ใน iSeries Navigator, ให้ขยาย **My Connections**.
2. คลิกขวาการเชื่อมต่อซึ่งคุณต้องการทำงานและคลิก **System Status**.

ถ้าต้องการเรียกสถานะระบบจากโพลเดอร์ Work Management:

1. ใน iSeries Navigator, ให้ขยาย **Work Management**.
2. คลิกขวา **Work Management** และคลิก **System Status**.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับภารกิจต่างๆ ที่คุณสามารถทำให้เสร็จสมบูรณ์โดยใช้ สถานะระบบ, ให้ดูคำอธิบาย iSeries Navigator.

การตรวจสอบการใช้พูลหน่วยความจำ

การตรวจสอบจำนวนของหน่วยความจำที่พูลหน่วยความจำของคุณใช้เป็นครั้งคราวนับเป็นสิ่งสำคัญ. การตรวจดูสิ่งดังกล่าว, จะช่วยให้คุณปรับแต่งพูลให้รันอย่างมีประสิทธิภาพ สูงสุด, ซึ่งสิ่งนี้, จะรักษาให้วงรอบของงานรันอย่างราบรื่น. ใน iSeries Navigator, คุณสามารถมอนิเตอร์จำนวนของหน่วยความจำที่พูลของคุณกำลังใช้อยู่ได้โดยง่าย.

ถ้าต้องการตรวจสอบการใช้งานหน่วยความจำ, ให้ทำดังนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → **server** → **Work Management** → **Memory Pools** → **Active Pools** หรือ **Shared Pools**.
2. คลิกขวาหน่วยความจำที่คุณต้องการทำงานด้วย (ตัวอย่างเช่น, Interactive) และคลิก **Properties**.

3. คลิกแท็บ Configuration. ฟิลด์ **Current** , ซึ่งตั้งอยู่ภายในกลุ่ม Size, แสดงจำนวนของหน่วยความจำที่พูล มีอยู่ในปัจจุบัน.

หมายเหตุ: คุณสามารถดูขนาดปัจจุบันของพูลหน่วยความจำเมื่อคลิก **Active Pools** หรือ **Shared Pools**. **Current Size** (ในเมกะไบต์) คือคอลัมน์ดีฟอลต์ที่คุณจะเห็น เมื่อรายการของพูลของหน่วยความจำแสดงขึ้นในระนาบด้านขวาของ iSeries Navigator.

การควบคุมระดับของ activity ระบบ

คุณสามารถควบคุมจำนวน activity บนระบบได้ โดยการควบคุมจำนวนงานที่สามารถแอ็คทีฟได้พร้อมกันในระบบย่อย หรือ โดยการควบคุมการใช้ หน่วยการประมวลผลของงานซึ่งถูกเริ่มต้นแล้ว.

ตารางที่ 6. วิธีการควบคุมระดับของ activity ระบบ

ฉันสามารถควบคุมอะไรได้บ้าง?	ฉันสามารถใช้อะไรในการควบคุม?	วิธีการใช้อินเตอร์เฟซแบบอักษร	วิธีการใช้อินเตอร์เฟซ iSeries Navigator
จำนวนของงานแอคทีฟ	Subsystem Description	<p>คำสั่ง: CHGSBSDMAXJOBS</p> <p>ใช้ พารามิเตอร์นี้ในการระบุจำนวนงานที่สามารถแอคทีฟได้พร้อมกันใน ระบบย่อย.</p> <p>สำหรับระบบย่อยที่แอคทีฟ, ผลรวมของงานทั้งหมดที่แอคทีฟในเวลาเดียวกันซึ่งถูกเริ่มผ่านทาง work entries ในระบบย่อย ไม่สามารถเกินกว่าค่าพารามิเตอร์ MAXJOBS.</p> <p>จำนวนนี้ไม่รวมงานแบบ autostart , ซึ่งอาจทำให้จำนวนงานเกินกว่าขีดจำกัดชั่วคราวเมื่อเริ่มต้น ระบบย่อย.</p>	<p>ใช้หน้าต่าง Run Command.</p> <p>คลิกขวาที่ ระบบจุดปลาย → Run Command</p> <p>พิมพ์ คำสั่ง CHGSBSD แล้วคลิก Prompt.</p>
	Job Queue Entry	<p>คำสั่ง: CHGJOBQEMAXACT</p> <p>ใช้ พารามิเตอร์นี้เพื่อระบุจำนวนงานแบ็คคิงจากคิวงานที่สามารถแอคทีฟได้พร้อมกันใน ระบบย่อย.</p> <p>MAXACT เป็น 1 สำหรับคิวงานจะทำให้งานถูกเลือกเป็นลำดับตามระดับความสำคัญของงานจากคิวงาน. พารามิเตอร์ MAXPTYn ใช้ในการระบุจำนวนงานที่สามารถแอคทีฟสำหรับระดับความสำคัญของงาน ที่ระบุ.</p>	<p>ใช้หน้าต่าง Run Command.</p> <p>คลิกขวาที่ ระบบจุดปลาย → Run Command</p> <p>พิมพ์ คำสั่ง CHGJOBQE แล้วคลิก Prompt.</p>
	Workstation Entry	<p>คำสั่ง: CHGWSEMAXACT</p> <p>ใช้ พารามิเตอร์นี้ถ้ามีการระบุพารามิเตอร์ WRKSTNTYPE. พารามิเตอร์นี้ระบุจำนวนงานแบบโต้ตอบที่สามารถแอคทีฟได้พร้อมกันในระบบย่อย สำหรับ entry นั้น.</p>	<p>ใช้หน้าต่าง Run Command.</p> <p>คลิกขวาที่ ระบบจุดปลาย → Run Command</p> <p>พิมพ์ คำสั่ง CHGWSE แล้วคลิก Prompt.</p>
	Communications Entry	<p>คำสั่ง: CHGCMNEMAXACT</p> <p>ใช้ พารามิเตอร์นี้เพื่อระบุจำนวนของงานแบ็คคิงการสื่อสารที่สามารถแอคทีฟได้พร้อมกันสำหรับ entry นั้น.</p>	<p>ใช้หน้าต่าง Run Command.</p> <p>คลิกขวาที่ ระบบจุดปลาย → Run Command</p> <p>พิมพ์ คำสั่ง CHGCMNE แล้วคลิก Prompt.</p>
	Routing Entry	<p>คำสั่ง: CHGRTGEMAXACT</p> <p>ใช้ คำสั่งนี้เพื่อระบุจำนวนงานที่สามารถแอคทีฟได้พร้อมกันโดยใช้ routing entry ที่ให้.</p>	<p>ใช้หน้าต่าง Run Command.</p> <p>คลิกขวาที่ ระบบจุดปลาย → Run Command</p> <p>พิมพ์ คำสั่ง CHGRTGE แล้วคลิก Prompt.</p>
	Prestart job entry	<p>คำสั่ง: CHGPJEMAXJOBS</p> <p>ใช้ คำสั่งนี้เพื่อระบุจำนวนของงานแบบ prestart ที่สามารถแอคทีฟได้พร้อมกัน สำหรับ entry นั้น.</p>	<p>ใช้หน้าต่าง Run Command.</p> <p>คลิกขวาที่ ระบบจุดปลาย → Run Command</p> <p>พิมพ์ คำสั่ง CHGPJE แล้วคลิก Prompt.</p>

ตารางที่ 6. วิธีการควบคุมระดับของ activity ระบบ (ต่อ)

ฉันสามารถควบคุมอะไรได้บ้าง?	ฉันสามารถใช้อะไรในการควบคุม?	วิธีการใช้อินเตอร์เฟซแบบอักษร	วิธีการใช้อินเตอร์เฟซ iSeries Navigator
จำนวนของงานแอคทีฟ (ต่อเนื่อง)	ระบบ	ค่ากำหนดของระบบ Maximum eligible threads (QMAXACTLVL) ใช้เพื่อระบุจำนวนของ threads ที่สามารถแชร์แหล่งเก็บข้อมูลหลักและโปรเซสเซอร์รีซอร์สได้ในเวลาเดียวกัน. งานแอคทีฟทั้งหมด (รวมถึงงานระบบ) ในพูลของหน่วยความจำ ทั้งหมดถูกควบคุมโดย QMAXACTLVL.	My Connections → server → Configuration and Service → System Values → Performance category → แท็บ Memory Pools → Maximum eligible threads
การใช้หน่วยการประมวลผลและแหล่งเก็บข้อมูลหลัก	Base storage pools	ค่ากำหนดของระบบ Base memory pool maximum eligible threads (QBASACTLVL) ใช้เพื่อระบุจำนวนของ threads ที่สามารถแชร์ Base storage pool ได้ในเวลาเดียวกัน และใช้เพื่อจำกัด main storage contention.	My Connections → server → Configuration and Service → System Values → Performance category → แท็บ Memory Pools → Base Memory pool: Maximum eligible threads
	Shared pools	คำสั่ง: WRKSHRPOOL ใช้คำสั่งนี้ เพื่อระบุ activity level สำหรับ shared pools	My Connections → server → Work Management → Memory Pools → Shared Pools → คลิกขวาที่ shared pool → Properties → แท็บ Configuration และเปลี่ยนฟิลด์ Maximum eligible threads
	พูลของหน่วยความจำส่วนตัว	คำสั่ง: CHGSBSDPOOLS ใช้ คำสั่งนี้เพื่อระบุ activity level สำหรับพูลแหล่งเก็บข้อมูลหลักที่ผู้ใช้กำหนด.	ใช้หน้าต่าง Run Command. คลิกขวาที่ ระบบจุดปลาย → Run Command พิมพ์ คำสั่ง CHGSBSD แล้วคลิก Prompt.

ตัวอย่าง: การควบคุม activity:

ตัวอย่างเหล่านี้แสดงความสัมพันธ์ของการควบคุม activity บางอย่าง. สมมติว่า activity level ของระบบคือ 100 และงานเป็น single-threaded.

ตัวอย่างพูลหน่วยความจำหลัก

ระบบย่อยสองระบบ, SBSA และ SBSB, ใช้พูลหน่วยความจำ Base ในการรันงาน. ในปัจจุบัน SBSA มีงานที่รันอยู่ใน พูลหน่วยความจำนี้อยู่สองงาน และ SBSB มีหนึ่งงาน. job queue entry ใน subsystem description สำหรับ SBSB ระบุจำนวนของงานที่สามารถเริ่มได้. activity level ของพูลหน่วยความจำ Base คือ 3. ดังนั้น, จะมีเพียงสามงานในพูล หน่วยความจำ Base เท่านั้นที่สามารถแย่งชิงหน่วยการประมวลผลในแต่ละครั้ง. อย่างไรก็ตาม, งาน ทั้งหมดจะถูกเริ่มขึ้น.

ตัวอย่างงานสีงานในระบบย่อย

มีงานแบบ autostart หนึ่งงาน, งานเวิร์กสเตชันสองงาน, และงานแบ็คชหนึ่งงาน (ทั้งหมดเป็นสีงาน) อยู่ในระบบย่อย SBSC. MAXACT สำหรับ SBSC ถูกระบุเป็น 4. ไม่ว่าค่าที่ระบุสำหรับ MAXACT ของ work entries จะเป็นอะไรก็ตาม, ไม่มีงานอื่นใดสามารถเริ่มขึ้นได้จนกว่างาน อย่างน้อยหนึ่งงานจะรันจนเสร็จสมบูรณ์.

ตัวอย่างระบบย่อยแบบแบ็ตซ์ MAXACT(1)

ระบบย่อย SBSE คือระบบย่อยแบบแบ็ตซ์ซึ่งมีการระบุ 1 สำหรับ MAXACT. แม้ว่า job queue entry ไม่ได้ระบุ MAXACT, ซีดจำกัดคือหนึ่งงาน เนื่องจากมีการระบุ 1 สำหรับ MAXACT สำหรับระบบย่อย. ดังนั้น, งานในคิวงานจะถูกประมวลผลตามระดับความสำคัญที่ละหนึ่งงาน.

การกำหนดสถานะของงาน

การมอนิเตอร์งานจะช่วยให้คุณรู้ว่างานดังกล่าวกำลังทำอะไรอยู่. สถานะของงานเป็นข้อมูลชิ้นสำคัญที่คุณสามารถใช้ดูว่างานชิ้นหนึ่งๆ กำลังทำงานอะไรอยู่.

ถ้าต้องการตรวจสอบสถานะของงานแอ็คทีฟหรืองานเซิร์ฟเวอร์, ให้ทำดังนี้:

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

หมายเหตุ: คุณสามารถดูสถานะของงานจากที่ใดก็ได้ ภายในโพลเดอร์ Work Management ที่คุณเข้าถึงงาน.

2. ดูที่คอลัมน์ Detailed Status เพื่อกำหนดสถานะของงาน (ตัวอย่างเช่น, Waiting for event, Waiting for time interval, หรือ Waiting for dequeue).

คำแนะนำ: ถ้าคุณไม่เห็นคอลัมน์ Detailed Status, คุณสามารถใส่เพิ่มคอลัมน์ที่จอแสดงผล โดยการคลิกขวาบน **Active Jobs** (หรือ **Server Jobs**) และเลือก **Customize this view** → **Columns**.

การมอนิเตอร์ระบบย่อย

เนื่องจากระบบย่อยมีความสำคัญกับ activity ประจำวันที่ทำบนระบบของคุณ, ดังนั้นจึงเป็นสิ่งสำคัญที่คุณต้องมอนิเตอร์ activity ในระบบย่อยของคุณ.

ภายใน subsystem description คุณสามารถระบุจำนวนงานที่สามารถรันได้พร้อมกัน ในระบบย่อยโดยการเซตค่าจำนวนงานแอ็คทีฟสูงสุด. เมื่อจำนวนงานบนระบบของคุณเพิ่มขึ้น คุณอาจต้องการปรับเปลี่ยนค่าจำนวนงานแอ็คทีฟสูงสุดในระบบย่อยของคุณ. จำนวนที่คุณป้อนที่นี่ควรถูกเซตเพื่อให้ใช้ประโยชน์รีซอร์สพร้อมใช้งานได้สูงสุด. การเพิ่มจำนวนของงานแอ็คทีฟโดยไม่ได้เพิ่มรีซอร์สพร้อมใช้งาน อาจทำให้ประสิทธิภาพการทำงานของระบบของคุณลดลง.

ถ้าต้องการตรวจสอบจำนวนงาน แอ็คทีฟสูงสุดในระบบของคุณ, คุณสามารถใช้ iSeries Navigator หรืออินเตอร์เฟซแบบอักษร อย่างไม่อย่างหนึ่ง.

iSeries Navigator:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → **server** → **Work Management** → **Subsystems** → **Active Subsystems**.
2. คลิกขวาระบบย่อยที่คุณต้องการมอนิเตอร์.
3. เลือก **Properties**.

หมายเหตุ: ให้แน่ใจว่า คุณเซตอ็อปชันนี้อย่างระมัดระวังมาก. ถ้าคุณเซตค่างานแอ็คทีฟสูงสุดสูงเกินไป, อาจทำให้ระบบของคุณทำงานช้า. อย่างไรก็ตาม, ถ้าคุณเซตงานแอ็คทีฟสูงสุดของคุณ ต่ำเกินไป, งานของคุณอาจติดขัดและมีประสิทธิภาพต่ำ.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Display Subsystem Description (DSPSBSD)

เลือก อีอ็อปชัน 1: แอ็คทีวิตีวีตการดำเนินการ, เพื่อดูจำนวนงานสูงสุดใน ระบบย่อย.

การกำหนดจำนวนของระบบย่อยที่ใช้พูลหน่วยความจำ

ระบบย่อยจะได้รับการจัดสรรหน่วยความจำจำนวนหนึ่งเพื่อใช้รันงาน. เป็นสิ่งสำคัญที่ต้องทราบจำนวนของระบบย่อยที่กำลังกำลังดึงหน่วยความจำออกมาจาก พูลหน่วยความจำเดียวกัน. หลังจากที่คุณทราบจำนวนของระบบย่อยที่กำลังส่งงานไปที่พูลและจำนวนของงานที่กำลังรันในพูล, คุณอาจต้องการลด contention รีซอร์สโดยการปรับเปลี่ยนขนาดและ activity level ของพูล.

iSeries Navigator:

ถ้าต้องการใช้ iSeries Navigator เพื่อมอนิเตอร์จำนวนของระบบย่อยที่กำลังใช้พูลหน่วยความจำ, ให้ทำดังต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → **server** → **Work Management** → **Memory Pools** → **Active Pools** หรือ **Shared Pools**.
2. คลิกขวาพูลหน่วยความจำที่คุณต้องการทำงานด้วยและคลิก **Subsystems**.
จากหน้าต่างนี้ คุณสามารถกำหนดจำนวนของระบบย่อยที่กำลังใช้แต่ละหน่วยความจำเพื่อรันงานของตน.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work with Subsystems (WRKSBS)

คำสั่งนี้แสดงรายการของระบบย่อยทั้งหมดและพูลของระบบย่อย.

การดูสถิติประสิทธิภาพงาน

ประสิทธิภาพของงานเป็นสิ่งสำคัญสำหรับทุกคนที่ใช้เซิร์ฟเวอร์ iSeries เนื่องจากการรันที่ไม่ดีของงานหนึ่งสามารถส่งผลกระทบต่องานอื่นในระบบ. การดูงานที่อาจเกิดปัญหาจะช่วยให้คุณป้องกันปัญหาประสิทธิภาพการทำงานก่อนที่จะเกิดขึ้นได้.

หน้าต่าง Elapsed Performance Statistics ช่วยให้คุณสามารถมอนิเตอร์การใช้งาน CPU ของงาน, disk I/O (ฮาร์ดไดรฟ์อินพุต/เอาต์พุต), อัตรา page fault, เวลาเฉลี่ยของการตอบกลับ, และจำนวนของ interactive transaction. คุณสามารถเลือกอีอ็อปชันในหน้าต่างนี้เพื่อรีเฟรชสถิติเหล่านี้ด้วยตนเองหรือตามตารางเวลา.

ถ้าต้องการแสดงสถิติประสิทธิภาพที่ผ่านมา, ให้ทำตามนี้:

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

หมายเหตุ: คุณสามารถดูประสิทธิภาพของงานได้จากที่ใดๆ ภายในการจัดการระบบงานที่คุณเห็นงาน. สามารถแสดงหน้าต่าง Elapsed Performance Statistics จากแท็บ Performance ของหน้าต่าง Job property.

2. คลิกขวางานซึ่งคุณต้องการแสดงสถิติประสิทธิภาพ, และคลิก **Details** → **Elapsed Performance Statistics**.

คุณสามารถรีเฟรช, รีเซ็ท, และจัดตารางเวลาการรีเฟรชสถิติประสิทธิภาพโดยอัตโนมัติ.

หมายเหตุ: คุณสามารถดูสถิติประสิทธิภาพที่ผ่านมาของงานได้มากกว่าหนึ่งงาน ในเวลาเดียวกัน โดยเปิดหน้าต่างหลายหน้าต่างพร้อมกัน. สิ่งนี้จะช่วยให้คุณมอนิเตอร์งานที่อาจมีปัญหามากกว่าหนึ่งงานได้พร้อมกัน. หน้าต่างแต่ละบานแสดงข้อมูลของงานเพียงหนึ่งงานเท่านั้น.

สถิติประสิทธิภาพที่ผ่านมาคือวิธีการหนึ่งในการดูประสิทธิภาพของงานเมื่องานนั้นเคลื่อนผ่านระบบ. อีกวิธีหนึ่งในการดูงานบนระบบคือดูผ่านทางโพลเดอร์ Management Central. คุณสามารถมอนิเตอร์งานใน Management Central รวมถึงการมอนิเตอร์ ประสิทธิภาพและข้อความของระบบ.

การดูสถานะระบบโดยรวม

iSeries Navigator ใช้ข้อมูลทั้งหมดที่เกี่ยวข้องกับสถานะระบบไว้ในที่เดียว. การทำเช่นนี้ช่วยให้คุณสามารถ มอนิเตอร์วิธีการปฏิบัติการของระบบของคุณ, ระบุพื้นที่ที่อาจเกิดปัญหา, ได้ง่ายขึ้น และสามารถกำหนด action ที่คุณต้องทำเพื่อปรับปรุงประสิทธิภาพ ได้อย่างรวดเร็ว.

หน้าต่าง System Status แบ่งสถานะระบบโดยรวมออกเป็นหกพื้นที่ ดังนี้:

ทั่วไป นี่คือเปอร์เซ็นต์การใช้งาน CPU ที่ผ่านไป, จำนวนของงานแอคทีฟ, เปอร์เซนต์ แอดเดรสที่ใช้, เปอร์เซนต์การใช้งานดิสก์พูลระบบ, งานทั้งหมดบนระบบ, เปอร์เซนต์ของแอดเดรสถาวรและแอดเดรสชั่วคราวที่ใช้, พื้นที่ว่างดิสก์ทั้งหมด, และความจุของดิสก์พูลระบบ.

งาน นี่คือจำนวนทั้งหมดของงาน, จำนวนของงานแอคทีฟ, จำนวนสูงสุดของงาน, และจำนวนของ thread ที่แอคทีฟ.

โพรเซสเซอร์

นี่คือเปอร์เซ็นต์การใช้งาน CPU ที่ผ่านไป. (ขึ้นอยู่กับ configuration ฮาร์ดแวร์ของคุณ, คุณยังอาจดูข้อมูลเพิ่มเติมเกี่ยวกับชนิดของ โพรเซสเซอร์, จำนวนของโพรเซสเซอร์, กำลังไฟฟ้า, โพรเซสเซอร์เสมือน, ประสิทธิภาพแบบโต้ตอบ, การใช้พูลโพรเซสเซอร์ที่แบ่งใช้ร่วมกันที่ผ่านไป, และ การใช้ความจุ uncapped CPU ที่ผ่านไป.)

หน่วยความจำ

นี่คือหน่วยความจำทั้งหมด (แหล่งเก็บข้อมูลหลัก) บนระบบของคุณ และปุ่ม ที่ช่วยให้คุณเข้าถึงรายการของพูลหน่วยความจำแอคทีฟบนระบบ.

Disk Space

นี่คือพื้นที่ว่างดิสก์ทั้งหมด, ความจุและการใช้งานดิสก์พูลระบบ, ข้อมูลเกี่ยวกับหน่วยเก็บชั่วคราวที่ใช้, และปุ่มที่ช่วยให้คุณเข้าถึง สถานะดิสก์มากขึ้น, รายการของดิสก์พูลบนระบบ, และข้อมูลค่ากำหนดของระบบ หน่วยเก็บ.

แอดเดรส

นี่คือข้อมูลเกี่ยวกับแอดเดรสถาวรและแอดเดรสชั่วคราวที่ใช้, แอดเดรสถาวรและแอดเดรสชั่วคราวขนาดใหญ่ (256 MB) ที่ใช้, และแอดเดรสถาวรและแอดเดรสชั่วคราวขนาดใหญ่มาก (4 GB) ที่ใช้.

ถ้าต้องการดูสถานะระบบทั่วไป, ให้ทำดังต่อไปนี้:

1. จาก iSeries Navigator, ให้ขยาย **My Connections**.
2. คลิกขวาที่เซิร์ฟเวอร์และคลิก **System Status**.

หน้าต่าง System Status จะปรากฏขึ้น. สำหรับข้อมูลเพิ่มเติมเกี่ยวกับหน้าต่างนี้, ให้ดูคำอธิบายทางออนไลน์ iSeries Navigator.

การตรวจสอบสถานะของดิสก์:

คุณสามารถตรวจสอบประสิทธิภาพของดิสก์ยูนิทบนระบบของคุณ, หรือดูข้อมูลสถานะของดิสก์ยูนิทได้ตลอดเวลา.

ถ้าต้องการดูหน้าต่าง Disk Status, ให้ทำดังต่อไปนี้:

1. ใน iSeries Navigator ให้ขยาย **My Connections** → **คลิกขวาที่ เซิร์ฟเวอร์** → **Disk Space** → **System Status** แท็บ .

2. บนหน้าต่าง Disk Space, ให้คลิก **Disk Status**. หน้าต่าง Disk Status จะเปิด.

คุณสามารถใช้ออปชั่น **Customize this View** → **Columns** ของหน้าต่าง Disk Status เพื่อดูข้อมูลดังต่อไปนี้:

- Amount Read (KB)
- Amount Written (KB)
- Percentage Busy
- Compression
- Disk Pool
- I/O Requests
- Percentage Used
- Protection Status
- Protection Type
- Read Requests
- Request Size (KB)
- Size (MB)
- Type
- Write Requests

การจัดการงาน

ตามที่คุณดูแลการจัดการระบบงานทราบ, การจัดการงานเป็นสิ่งที่มีมากกว่า การพักผ่อนไว้และการย้ายงานจากคิวงานหนึ่งไปอีกคิวงานหนึ่ง. หัวข้อนี้อธิบายภารกิจการจัดการงานที่ใช้ทั่วไปและ ภารกิจที่เกี่ยวข้องเพิ่มเติม ซึ่งสามารถช่วยพัฒนาประสิทธิภาพของระบบของคุณได้.

ภารกิจงานทั่วไป

ข้อมูลนี้อธิบายภารกิจทั่วไปที่คุณสามารถปฏิบัติกับงาน. และแสดงคำสั่งสำหรับทั้ง iSeries Navigator (ในที่ที่พร้อมใช้งาน) และ อินเทอร์เน็ตแบบอักษระ.

การเริ่มงาน:

งานแบบโต้ตอบจะเริ่มเมื่อผู้ใช้ signs on ที่เวิร์กสเตชัน. คุณเริ่มงานแบบ prestart และงานแบ็ตซ์โดยการใช้อeries Navigator หรืออินเทอร์เน็ตแบบอักษระ, ขึ้นอยู่กับสถานการณ์.

การเริ่มงานแบ็ตซ์ที่กำลังรออยู่ในคิวงาน:

ในบางโอกาส คุณอาจต้องการบังคับให้งานเริ่มในทันที. ในขณะที่การย้ายงานไปที่คิวงานที่ไม่ยุ่งเป็นวิธีการที่มีประสิทธิภาพสูงสุด, แต่มีวิธีการอื่นที่คุณสามารถใช้ได้.

เพื่อเริ่มงานแบ็ตซ์, ในอันดับแรก ให้ตรวจสอบสถานะของคิวงานที่งานตั้งอยู่และพิจารณาว่าการย้ายงานไปยังคิวอื่นเป็นวิธีที่เหมาะสมที่สุดสำหรับสถานการณ์ของคุณหรือไม่. (My Connections → server → Work Management → Job Queues → Active Job Queues หรือ All Job Queues)

ถ้าการย้ายงานไปที่คิวงานอื่นเป็นสิ่งที่ไม่เหมาะสม, คุณสามารถพักงานที่รันอยู่ แล้วเปลี่ยนระดับความสำคัญของงานที่คุณต้องการเริ่ม. อย่างไรก็ตาม, ควรระวังเมื่อใช้วิธีการนี้ เนื่องจากงานที่ถูกพัก ยังคงรวมอยู่ในการนับงานแอ็คทีฟสูงสุด.

ถ้าต้องการเปลี่ยนระดับความสำคัญของงานและระยะเวลาที่งานควรรัน, ให้ทำดังต่อไปนี้:

1. คลิกขวาที่งานและคลิก **Properties**.
2. บนหน้าต่าง Job Properties, ให้คลิกแท็บ **Job Queue**.
3. เปลี่ยน **Priority on job queue** เป็นระดับความสำคัญ ที่สูงกว่า (0 คือสูงสุด).
4. เช็ท **When to make job available to run** เป็น Now หรือระบุวันที่และเวลา ใดอย่างหนึ่ง.
5. คลิก **OK**.

การเริ่มงานแบบ prestart:

โดยปกติงานแบบ prestart จะเริ่มในเวลาเดียวกับที่ระบบย่อยเริ่มขึ้น. คุณเริ่มงานแบบ prestart ด้วยตนเอง เมื่องานแบบ prestart ทั้งหมดได้ถูกจบ โดยระบบเนื่องจากข้อผิดพลาดหรืองานไม่ถูกเริ่มขึ้นในระหว่างที่ระบบย่อย เริ่มขึ้นเนื่องจาก STRJOBS (*NO) บน prestart job entry. ถ้าต้องการเริ่ม งานแบบ prestart, ให้ใช้อินเตอร์เฟซแบบอักษร.

คำสั่ง: Start Prestart Jobs (STRPJ)

คำสั่ง STRPJ ไม่ควรถูกใช้จนกว่าการเริ่มต้นของระบบย่อยที่เกี่ยวข้องจะเสร็จสมบูรณ์. เพื่อให้มั่นใจว่างานแบบ prestart ที่จำเป็นได้เริ่มเรียบร้อยแล้ว, ให้ใช้ไค้ด delay loop แล้วลองใหม่ถ้าคำสั่ง STRPJ ล้มเหลว.

จำนวนของ งานแบบ prestart ที่สามารถแอ็คทีฟได้ในเวลาเดียวกันถูกจำกัดโดยแอ็คทีฟบิตต์ MAXJOBS บน prestart job entry และโดยแอ็คทีฟบิตต์ MAXJOBS สำหรับระบบย่อย. แอ็คทีฟบิตต์ MAXACT บน communications entry ควบคุม จำนวนของ คำขอการเริ่มโปรแกรมที่สามารถให้บริการได้ผ่านทาง communications entry ในเวลาเดียวกัน.

หมายเหตุ: ถ้าคุณระบุ *NO บนแอ็คทีฟบิตต์ STRJOBS, จะไม่มีงานแบบ prestart ใดเริ่มขึ้นสำหรับ prestart job entry เมื่อระบบย่อยเริ่มต้น. การรันคำสั่ง STRPJ ไม่ได้ทำให้ค่าของพารามิเตอร์ STRJOBS เปลี่ยนแปลง.

ตัวอย่าง: ตัวอย่างนี้จะเริ่มงานแบบ prestart สำหรับ prestart job entry PJPGM ในระบบย่อย SBS1. ระบบย่อย SBS1 ต้องแอ็คทีฟที่ฟในขณะที่ย่อ คำสั่งนี้. จำนวนของงานที่เริ่มต้นคือจำนวนที่ระบุในค่า INLJOBS ของ prestart job entry PJPGM. ระบบย่อย จะเริ่มโปรแกรม PJPGM ในไลบรารี PJLIB.

```
STRPJ SBS(SBS1) PGM(PJLIB/PJPGM)
```

การจบงาน:

คุณสามารถใช้ iSeries Navigator หรืออินเตอร์เฟซแบบอักษรเพื่อจบงาน. งานสามารถแอ็คทีฟหรืออยู่บนคิวงาน. คุณสามารถจบงาน โดยทันทีหรือโดยการระบุช่วงเวลาเพื่อให้จบการประมวลผลงาน.

iSeries Navigator:

ถ้าต้องการใช้ iSeries Navigator, ให้ทำดังต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **Work Management** → **Active Jobs**.
2. ระบุตำแหน่งงานที่คุณต้องการจบ.
3. คลิกขวาที่งานและคลิก **Delete/End**.

4. กรอกข้อมูลในหน้าต่าง Confirm Delete/End ให้เสร็จสมบูรณ์และคลิก Delete.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: End Job (ENDJOB)

ถ้าคุณไม่ทราบชื่อ ของงานที่คุณต้องการจบ, คุณสามารถใช้คำสั่งใดคำสั่งหนึ่งดังต่อไปนี้เพื่อ ค้นหาชื่องานได้:

- Work with Active Jobs (WRKACTJOB)
- Work with User Jobs (WRKUSRJOB)
- Work with Submitted Jobs (WRKSBMJOB)
- Work with Subsystem Jobs (WRKSBSJOB)
- End Subsystem (ENDSBS) คำสั่งนี้จะจบงานทั้งหมด ในระบบย่อย.
- End System (ENDSYS) คำสั่งนี้จะจบ activity ส่วนใหญ่บนระบบ และปล่อยให้ระบบอยู่ในสภาพซึ่งมีเฉพาะคอนโซลเท่านั้น ที่แอ็คทีฟในระบบย่อยการควบคุม.
- Power Down System (PWRDWN SYS) คำสั่งนี้จะเตรียม ระบบสำหรับการสิ้นสุด แล้วเริ่มลำดับการปิดเครื่อง.

งานสามารถจบลงโดยทันทีหรือจบในลักษณะที่ควบคุม อย่างใดอย่างหนึ่ง. ขอ แนะนำอย่างยิ่งให้คุณพยายามจบงานในลักษณะที่ควบคุมเสมอ .

การจบงาน: แบบควบคุม:

การจบงานในลักษณะที่ควบคุมช่วยให้โปรแกรมที่กำลังรันอยู่ในงานสามารถปฏิบัติการล้าง end-of-job ได้. สามารถระบุเวลา หนึ่งวง เพื่ออนุญาตให้งานจบลงในลักษณะที่ควบคุม. ถ้าเวลาหนึ่งวงหมดลงก่อนที่ งานจะจบ, งานจะถูกจบในทันที.

แอฟพลิเคชันใดๆ ที่จำเป็นต้องปฏิบัติการล้าง end-of-job ควรตรวจสอบ ว่างานกำลังจบลงในลักษณะที่ควบคุมเมื่อไร. แอฟพลิเคชันสามารถตรวจสอบสิ่งนี้ได้ สามวิธี:

การดึงข้อมูล End Status แบบซิงโครนัส

ที่บางจุด, แอฟพลิเคชันสามารถตรวจสอบ End Status แบบซิงโครนัสของงานซึ่งแอฟพลิเคชันกำลังรันอยู่ได้. คุณสามารถเรียกข้อมูลสถานะการจบ ของงานโดยใช้คำสั่ง Retrieve Job Attributes (RTVJOBA) CL. นอกจากนี้, คุณสามารถใช้หนึ่งจากหลาย APIs ที่เรียกข้อมูลสถานะการจบของงาน. คุณสามารถหาข้อมูลเพิ่มเติมเกี่ยวกับ APIs เหล่านี้ในรายงานจากประสบการณ์, *แอ็คทีวิตีวต์งานการจัดการระบบงาน*

ตรวจสอบคำสั่งคืนหลักและรองอย่างซิงโครนัส หลังจากการดำเนินการ I/O

สำหรับทั้ง display I/O และ ICF communications I/O, คำสั่งคืนหลักของ 02, หรือคำสั่งคืนหลักของ 03 ที่มีคำสั่งคืนรอง 09 บ่งชี้ว่า งานกำลังจบในลักษณะที่ควบคุม.

การจัดการสัญญาณอะซิงโครนัส SIGTERM

บางแอฟพลิเคชันใช้โปรแกรมการจัดการสัญญาณเพื่อให้การล้างข้อมูล ของแอฟพลิเคชันมีประสิทธิภาพมากขึ้น เมื่องานถูกจบ. ระบบจะสร้างสัญญาณอะซิงโครนัส SIGTERM สำหรับงานที่จะจบ, เมื่องานกำลังจบแบบควบคุม และเป็นไปตามเงื่อนไขทั้งหมดดังต่อไปนี้:

- งานถูกเปิดทางสำหรับสัญญาณ
- งานคือโปรแกรมการจัดการสัญญาณที่ถูกสร้างขึ้นสำหรับสัญญาณ SIGTERM
- งานกำลังรันอยู่ในระยะปัญหาในปัจจุบัน

ถ้าไม่เป็นไปตามเงื่อนไขใดๆ ข้างต้น, สัญญา SIGTERM จะไม่ถูกสร้างขึ้น สำหรับงานที่กำลังจบ.

เมื่องานที่กำลังจบในลักษณะแบบควบคุม มีขั้นตอนการจัดการสัญญาสำหรับสัญญาอะซิงโครนัส SIGTERM, สัญญา SIGTERM จะถูกสร้างขึ้นสำหรับงานนั้น. เมื่อสามารถควบคุมขั้นตอนการจัดการ สัญญาสำหรับสัญญา SIGTERM, ขั้นตอนสามารถใช้ actions ที่เหมาะสมเพื่อให้แอพลิเคชันสามารถจบในลักษณะแบบควบคุม.

การจบงาน: ทันที:

ใช้อ็อพชันจบแบบทันทีเฉพาะถ้าการจบแบบควบคุมไม่ประสบความสำเร็จเท่านั้น. เมื่องานจบโดยทันที, คุณอาจได้ผลลัพธ์ที่ไม่ต้องการ เช่น ข้อมูลแอพลิเคชันที่ถูกอัปเดตเป็นบางส่วน.

ก่อนการจบงาน, คุณควรตรวจสอบว่าไม่มี logical unit of work อยู่ในสถานะที่น่าสงสัย เนื่องจากการดำเนินการ two-phase commit ที่อยู่ในระหว่างดำเนินการ. ถ้ามี, ค่าของ Action ifENDJOB commitment option อาจส่งผลกระทบต่อ การประมวลผล ENDJOB. อ็อพชันนี้เป็นส่วนหนึ่งของ Change Commitment Options (QTNCHGCO) API. ตัวอย่างเช่น, ถ้า Action ifENDJOB commitment option เป็นค่าดีฟอลต์ของ WAIT, งานนี้จะถูกพักไว้และจะทำให้การจบการประมวลผลงาน ไม่สมบูรณ์จนกว่าการดำเนินการ commitment control จะเสร็จสมบูรณ์. สิ่งนี้ช่วยให้มั่นใจถึง integrity ฐานข้อมูลบนระบบที่เกี่ยวข้องทั้งหมด.

เมื่อคุณใช้อ็อพชันจบแบบทันที, ระบบจะประมวลผล minimal end-of-job, ซึ่งสามารถรวมถึง:

- การปิดไฟล์ฐานข้อมูล
- การสวอปบันทึกการใช้งานที่เอาต์พุตคิว
- การล้างอ็อบเจกต์ภายในในระบบปฏิบัติการ
- การแสดงจอแสดงผล end-of-job (สำหรับงานแบบโต้ตอบ)
- การทำกระบวนการ commitment control ให้เสร็จสมบูรณ์

การค้นหางาน:

นับเป็นสิ่งสำคัญที่ต้องเข้าใจถึงวิธีการค้นหางานบนเซิร์ฟเวอร์ iSeries ของคุณ. ไม่ว่าด้วยเหตุผลใดก็ตาม, บางเวลาคุณอาจต้องการข้อมูลจากงานเฉพาะใดๆ .

ใน iSeries Navigator, คุณสามารถใช้ Find บนงานทั้งหมดของคุณ หรือสามารถลดจำนวนการค้นหาโดยการ ใช้ฟังก์ชัน Include ตามด้วย Find. ฟังก์ชัน Include ช่วยให้คุณสามารถใส่ ข้อจำกัดของสิ่งที่จะแสดงใน iSeries Navigator. ตัวอย่างเช่น, แทนที่จะใช้ Find เพื่อค้นหางานหลายร้อยงาน, คุณสามารถรับ Include เพื่อให้แสดงเฉพาะงานบางชนิดได้. หรือ, คุณสามารถแสดงเฉพาะงานที่มี job user ID ที่ระบุเท่านั้น.

เพื่อประสิทธิภาพการทำงานสูงสุด, ถ้าคุณ มีงานอยู่เป็นจำนวนมากบนระบบ, ขอแนะนำให้คุณใช้ฟังก์ชัน Include เพื่อลดจำนวนของงานที่จะค้นหา. ถ้าคุณมีงานอยู่เป็นจำนวนมากบนระบบ, การค้นหาจาก งานทั้งหมดอาจทำให้ประสิทธิภาพการทำงานลดลง.

หมายเหตุ: คุณสามารถใช้คำสั่ง Find และฟังก์ชัน Include ได้ในทุกส่วนของการจัดการระบบงานที่คุณค้นหางาน. คุณยังสามารถใช้เครื่องมือต่างๆ ในการค้นหาคิวงาน, ระบบย่อย, และพูลหน่วยความจำในลักษณะเดียวกันอีกด้วย. ให้จำไว้ว่าคุณต้องคลิกพื้นที่ที่คุณต้องการค้นหา ก่อน จึงจะสามารถใช้เครื่องมือเหล่านี้ได้.

iSeries Navigator:

ถ้าต้องการค้นหางานโดยใช้ไอพชั่น Find (Ctrl+F), ให้ทำดังต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → **server** → **Work Management** → **Active Jobs**.
2. บนเมนู Edit, ให้คลิก **Find** (Ctrl+F).
3. ในฟิลด์ **Search for:**, ให้พิมพ์ job ID ที่คุณต้องการค้นหา (เช่น Qqqtemp1). งานของคุณจะถูกค้นหาในคอลัมน์งานทั้งหมด.
4. คลิก **Find**. iSeries Navigator จะไฮไลต์งานของคุณ ในทันทีที่พบ.

เตือนความจำ: จะคำนึงถึงขนาดตัวพิมพ์ ของชื่องาน เฉพาะเมื่อใส่ชื่องานในเครื่องหมายคำพูดเท่านั้น (เช่น, "MyJob"). แต่หากชื่องานไม่ได้อยู่ในเครื่องหมายคำพูด, คำสั่งจะไม่คำนึงขนาดตัวพิมพ์เล็กใหญ่.

การจำกัดข้อมูลที่จะแสดง:

ถ้าต้องการจำกัดข้อมูลที่จะแสดง, ให้ใช้ฟังก์ชัน Include.

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → **server** → **Work Management** → **Active Jobs** หรือ **Server Jobs**.
2. จากเมนู View, ให้คลิก **Customize this View**, แล้ว **Include**. หน้าต่าง Include จะปรากฏขึ้น.
3. ในหน้าต่าง Include, ให้เลือกไอพชั่นที่คุณต้องการในการค้นหา งานของคุณ.
4. คลิก **OK**.

อินเตอร์เฟซแบบอักษร:

ถ้าต้องการค้นหางานบนระบบ, ให้ใช้คำสั่ง Work with Active Job (WRKACTJOB), Work with User Job (WRKUSRJOB), หรือ Work with Submitted Job (WRKSBJOB) อย่างเป็นอย่างหนึ่ง.

การดูงานบนคิวงาน:

คิวงานกรองงานบางส่วนที่จะถูกประมวลผลในการจัดการระบบงาน (ยกตัวอย่างเช่น, งานแบบแบ็คซ์บางงาน). การดูงานในคิวงานได้ทำให้คุณสามารถดูว่างานใดที่กำลังรอที่จะถูกส่งไปยังระบบย่อย.

iSeries Navigator:

ถ้าต้องการดูงานในคิวงาน, ให้ทำดังนี้:

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

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work with Job Queue (WRKJOBQ)

คำสั่งนี้จะ แสดงรายการของคิวงานทั้งหมดที่พร้อมใช้งานบนระบบ. หลังจากที่คุณระบุตำแหน่งคิวงานที่มีงานของคุณแล้ว คุณสามารถเลือกไอพชั่น 5=ทำงานกับ และแสดงงานทั้งหมดใน คิวงาน.

คุณยังสามารถใช้คำสั่ง Work with Subsystems Job เพื่อแสดง รายการคิวงานและงานของคิวงานได้ด้วย.

คำสั่ง: Work with Subsystem Job (WRKSBSJOB) SBS(*JOBQ)

การดูงานในระบบย่อย:

ระบบย่อยจะประสานการไหลของงานกับรีจิสเตอร์ที่งานใช้ในการรัน. iSeries Navigator ช่วยให้คุณสามารถดูได้ว่างานใดที่แอ็คทีฟอยู่ในปัจจุบัน (แต่ไม่จำเป็นต้องกำลังรัน อยู่) ในระบบย่อย.

iSeries Navigator:

ถ้าต้องการดูงานในระบบย่อย, ให้ทำตามขั้นตอนเหล่านี้:

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

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work with Active Jobs (WRKACTJOB SBS(ชื่อระบบย่อย))

คำสั่ง: Work with Subsystem Descriptions (WRKSBSD)

ใช้คำสั่ง Work with Subsystem Descriptions เพื่อแสดงรายการของ ระบบย่อย. หลังจากที่ค้นพบระบบย่อยที่มีงานของคุณ, ให้ใช้อ็อปชัน 8=ทำงาน กับงานระบบย่อย เพื่อแสดงข้อมูลงาน.

หมายเหตุ: ระบบย่อย ต้องแอ็คทีฟเพื่อแสดงข้อมูลงาน.

การดูแอ็คทีฟริบิตงาน:

แอ็คทีฟริบิตงานมีข้อมูลเกี่ยวกับวิธีการประมวลผลงาน. แอ็คทีฟริบิตงานจะถูกระบุไว้ตั้งแต่ตอนที่งานได้ถูกสร้างขึ้น. บางแอ็คทีฟริบิต มาจากรายละเอียดของงาน. หลังจากทำงานถูกสร้างขึ้น, สามารถดูและจัดการแอ็คทีฟริบิตงาน ผ่านทางการจัดการระบบงานใน iSeries Navigator. หน้าคุณสมบัติงานใน iSeries Navigator ทำให้งานของผู้ควบคุมระบบงานขึ้น โดยการนำเสนอฟังก์ชันที่มีประสิทธิภาพและใช้งานได้ง่ายสำหรับการจัดการงาน.

iSeries Navigator:

ถ้าต้องการดูแอ็คทีฟริบิตงาน, ให้ทำตามต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → **server** → **Work Management** → **Active Jobs** หรือ **Server Jobs**, ขึ้นอยู่กับชนิดของงานที่คุณต้องการทำงานด้วย. .
2. ค้นหางานที่มีคุณสมบัติที่คุณต้องการดูหรือเปลี่ยนแปลง.
3. คลิกขวาที่ **Job Name** และคลิก **Properties**.

แอ็คทีฟริบิตงานสามารถดูได้โดยผู้ใช้ใดๆ, แต่สามารถเปลี่ยนได้โดยผู้ใช้ที่มีสิทธิ์ที่ถูกต้องเท่านั้น. ในลักษณะคล้ายกัน, ผู้ใช้ที่ได้รับสิทธิ์สามารถจัดการ งานผ่านการปฏิบัติงาน. แอ็คทีฟริบิตสำหรับงานระบบไม่สามารถเปลี่ยนแปลงได้ใน iSeries Navigator. อย่างไรก็ตาม, ระดับความสำคัญการรันของงานระบบบางงานสามารถเปลี่ยนได้ใน อินเตอร์เฟซแบบอักษรโดยใช้คำสั่ง Change System Job (CHGSYSJOB).

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work with Job (WRKJOB) เมื่อ งานแอคทีฟ คุณสามารถดูข้อมูลดังนี้: แอ็คทีวิตีการรันงาน, ข้อมูล call stack, ข้อมูลการล็อกงาน, ข้อมูลรายชื่อไลบรารี, ข้อมูล บันทึกการใช้งาน, ข้อมูลไฟล์เปิด, ข้อมูล file override, สถานะ commitment control, สถานะการสื่อสาร, ข้อมูล activation group, ข้อมูล mutex , และข้อมูล thread

คำสั่ง: Display Job (DSPJOB)

คำสั่งนี้ แสดงข้อมูลเกี่ยวกับงานดังต่อไปนี้: แอ็คทีวิตีสถานะงาน, job definition attributes, แอ็คทีวิตีการรันงาน, ข้อมูลสพูลไฟล์, ข้อมูลบันทึกการใช้งาน, ข้อมูล call stack, ข้อมูลการล็อกงาน, ข้อมูล รายชื่อไลบรารี, ข้อมูลไฟล์เปิด, ข้อมูล file override, สถานะ commitment control, สถานะการสื่อสาร, ข้อมูล activation group, ข้อมูล mutex , ข้อมูล thread, ไลบรารีที่ล๊อคบันทึก, และข้อมูลแอ็คทีวิตี.

การดู call stacks:

คุณสามารถดูข้อมูลเกี่ยวกับงานหรือ call stack ของ thread ได้โดยใช้ iSeries™ Navigator หรืออินเทอร์เฟซแบบอักษร อย่างไม่ใดอย่างหนึ่ง.

iSeries Navigator:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → **server** → **Work Management** → **Active Jobs** หรือ **Server Jobs**, ขึ้นอยู่กับชนิดของงาน ที่คุณต้องการทำงานด้วย.
2. คลิกขวาที่ชื่องาน แล้วคลิก **Details** → **Call Stack**.

ถ้าคุณต้องการดู call stack สำหรับ thread, ให้ทำตามขั้นตอนเหล่านี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → **server** → **Work Management** → **Active Jobs** หรือ **Server Jobs**, ขึ้นอยู่กับชนิดของงาน ที่คุณต้องการทำงานด้วย.
2. คลิกขวาที่ชื่องาน, แล้วคลิก **Details** → **Threads**.
3. จากรายการของ threads, คลิกขวา thread เฉพาะ, แล้วคลิก **Details** → **Call Stack**.

ถ้าคุณกำลังรันภายใต้โปรไฟล์ผู้ใช้ที่มีสิทธิ์พิเศษ *SERVICE และต้องการดู entries เพิ่มเติมสำหรับ LIC และ i5/OS PASE Kernel, จากหน้าต่าง Call Stack, ให้ใช้อ็อปชัน Include จากหน้าต่าง Customize this view. (**เมนู View** → **Customize this view** → **Include**)

อินเทอร์เฟซแบบอักษร:

คำสั่ง: Work with Jobs (WRKJOB) หรือ Display Jobs (DSPJOB)

เลือกอ็อปชัน 11: แสดงผล call stack, ถ้า แอ็คทีฟ.

ถ้าคุณต้องการดู call stack สำหรับ thread, หลังจากการออกคำสั่ง WRKJOB หรือ DSPJOB, ให้เลือก อ็อปชัน 20: ทำงานกับ threads, ถ้าแอ็คทีฟ. จากนั้น, เลือกอ็อปชัน 10: แสดงอ็อปชัน call stack สำหรับ thread ที่เลือก.

การวางงานบนคิวงาน:

งานถูกวางบนคิวงานโดยการย้ายงานที่มีอยู่จากคิวงานหนึ่งไปยังคิวงานอื่น, หรือโดยการส่งงานใหม่ อย่างไม่ใดอย่างหนึ่ง. ใช้ iSeries Navigator เพื่อย้ายงานระหว่างคิวต่างๆ. ใช้อินเทอร์เฟซแบบอักษรเพื่อส่งงานใหม่.

iSeries Navigator:

ในการใช้อินเตอร์เฟซ iSeries Navigator, งานต้องมีอยู่แล้วใน คิวงานอื่น. จากนั้นคุณสามารถย้ายงานจากคิวหนึ่งไปยังคิวงานอื่นได้. (ถ้าต้องการวางแผนใหม่บนคิวงาน, ให้ใช้อินเตอร์เฟซบรรทัดรับคำสั่ง.)

1. ใน iSeries Navigator, ให้ขยาย **Work Management** → **Job Queues** → **All Job Queues**.
2. คลิกขวาบนที่คิวดูต้องการย้าย. หน้าต่าง Move ที่คุณสามารถระบุคิวปลายทางจะเปิดขึ้น.

อินเตอร์เฟซแบบอักษร:

ข้อมูลต่อไปนี้เป็นรายการของวิธีการใช้อินเตอร์เฟซแบบอักษร สำหรับการวางแผนใหม่บนคิวงานใหม่.

- **Submit Job (SBMJOB):** อนุญาตให้งานที่กำลังรันอยู่ ส่งงานอื่นไปที่คิวงานเพื่อรันในภายหลังเป็นงานแบ็คช. สามารถวางได้เพียงหนึ่งองค์ประกอบของข้อมูลค่าขอเท่านั้นบน message queue ของงานใหม่. ข้อมูลค่าขอสามารถเป็นคำสั่ง CL ถ้า routing entry ที่ใช้สำหรับงานระบุโปรแกรมการประมวลผลคำสั่ง CL (เช่น โปรแกรม QCMD ที่ได้มาจาก IBM).
- **Add Job Schedule Entry (ADDJOBSCDE):** ระบบจะ ส่งงานโดยอัตโนมัติไปที่คิวงานในเวลาและวันที่ที่ระบุใน job schedule entry.
- **Submit Database Jobs (SBMDBJOB):** ส่งงานไปที่คิวงานเพื่อให้งานสามารถรันเป็นงานแบ็คชได้. input stream ถูกอ่านจาก ไฟล์ฐานข้อมูลฟิสิกัลหรือจากไฟล์ฐานข้อมูลโลจิคัล อย่างใดอย่างหนึ่ง ที่มีรูปแบบเร็กคอร์ดเดียว. คำสั่งนี้อนุญาตให้คุณระบุชื่อของไฟล์ฐานข้อมูลนี้และ เมมเบอร์ของไฟล์, ชื่อของคิวงานที่จะใช้, และเลือกตัดสินใจว่างานที่จะส่งสามารถถูกแสดงโดยคำสั่ง Work with Submitted Jobs (WRKSBMJOB) หรือไม่.
- **Start Database Reader (STRDBRDR):** อ่าน batch input stream จากฐานข้อมูลและวางหนึ่งหรือหลายงานบนคิวงาน.
- **Transfer Job (TFRJOB):** ย้ายงานปัจจุบันไปยัง คิวงานอื่นในระบบย่อยแอ็คทีฟ.
- **Transfer Batch Job (TFRBCHJOB):** ย้ายงานปัจจุบัน ไปยังคิวงานอื่น.

การย้ายงานไปที่คิวงานอื่น:

มีหลายเหตุผลที่คุณอาจต้องการย้ายงานไปที่คิวอื่น. ตัวอย่างเช่น, ในบางครั้งงานอาจถูก backlogged ในคิวเนื่องจากเป็นงานที่ใช้เวลานาน. บางครั้งรันใหม่ที่จัดตารางเวลาของงานขัดแย้งกับงานใหม่ที่มีระดับความสำคัญสูงกว่า. วิธีหนึ่งในการจัดการสถานการณ์นี้คือ การย้ายงานที่รอไปยังคิวอื่นที่ไม่ยุ่ง.

คุณสามารถใช้ iSeries Navigator หรืออินเตอร์เฟซแบบอักษร อย่างใดอย่างหนึ่ง เพื่อย้ายงานจากคิวหนึ่งไปยังคิวอื่น.

iSeries Navigator:

ถ้าต้องการใช้ iSeries Navigator, ให้ทำตามต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **Work Management** → **Job Queues** → **All Job Queues**.
2. ระบุตำแหน่งและเปิดคิวที่มีงานอยู่ในปัจจุบัน.
3. คลิกขวาบนที่คิวดูต้องการย้าย. หน้าต่าง Move ที่คุณสามารถระบุคิวเป้าหมายจะเปิดขึ้น.

หมายเหตุ: ถ้าคุณต้องการย้าย งานมากกว่าหนึ่งงานจากคิวนี้, ให้กดปุ่ม CTRL ค้างไว้ในขณะที่คุณคลิก แต่ละงาน. จากนั้นคลิกขวา แล้วคลิก Move.

- งานที่กำลังรอเพื่อรันจะถูกย้ายไปที่ตำแหน่งสัมพันธ์เดียวกันบนคิวเป้าหมาย (เช่น, งานที่มีระดับความสำคัญคิวงาน 3 จะถูกย้ายไปอยู่ หลังจากงานระดับความสำคัญ 3 อื่นที่กำลังรอเพื่อรันบนคิวเป้าหมาย).

- งานที่ถูกพักยังคงถูกพักอยู่และถูกวางในตำแหน่งสัมพันธ์เดียวกัน บนคิวเป้าหมาย (เช่น, งานที่พักที่มีระดับความสำคัญ คิวงาน 3 จะถูกย้ายไปอยู่หลังจากงานที่พักที่มีระดับความสำคัญ 3 อื่นบนคิวเป้าหมาย).
- งานที่ถูกจัดตารางเวลาเพื่อรันจะถูกย้ายไปที่คิวเป้าหมาย และ เวลาที่จัดตารางเวลาของงานยังคงไม่เปลี่ยนแปลง.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Change Job (CHGJOB)

ตัวอย่าง: ตัวอย่างต่อไปนี้จะย้ายงาน JOBA ไปที่คิวงาน JOBQB.

CHGJOB JOB(JOBA) JOBQ(LIBA/JOBQB)

การเปลี่ยนระดับความสำคัญของงานภายในคิวงาน:

งานทั้งหมดในคิวงานรอในแถวสำหรับการประมวลผล. เมื่องานแต่ละงาน ในคิวเสร็จสมบูรณ์, งานถัดไปในแถวจะเริ่มขึ้น. ลำดับการประมวลผล ของงานในคิวขึ้นอยู่กับระดับความสำคัญของงาน, และจำนวนสูงสุด ของงานที่สามารถรันได้พร้อมกัน บนระบบย่อย.

ในบางกรณี ความสำคัญของงานก็มีการเปลี่ยนไปเมื่องานดังกล่าวผ่านไปในช่วงวงรอบชีวิต. ระดับความสำคัญอาจเพิ่มขึ้นหรือลดลงเมื่อเทียบกับงานอื่น. เนื่องจากการเปลี่ยนแปลงดังกล่าวเกิดขึ้น, คุณจึงต้องทราบวิธีการเปลี่ยนระดับความสำคัญของงานในคิวงาน.

ระดับความสำคัญของงานในคิวงานจะช่วยกำหนดได้ว่าเมื่อไรที่งานสมควรจะวิ่ง ไปยังระบบย่อยเพื่อทำงาน. ช่วงตั้งแต่ศูนย์ถึงเก้า (ศูนย์คือสำคัญมากที่สุด) กำหนดระดับความสำคัญของงานบนคิวงาน.

iSeries Navigator:

คุณสามารถใช้ iSeries Navigator เพื่อเปลี่ยนระดับความสำคัญของงาน ในคิวงานได้.

1. ขยาย My Connections → server → Work Management → Job Queues → Active Job Queues หรือ All Job Queues → คิวงาน ซึ่ง งานของคุณตั้งอยู่.
2. คลิกขวาที่งานและคลิก Properties.
3. บนหน้าต่าง Job - Properties, ให้คลิกแท็บ Job Queue .
4. ที่รายการ Priority on job queue, ให้เลือกหมายเลขระดับความสำคัญสูงขึ้น (หรือต่ำลง). ระดับความสำคัญในการรันจะอยู่ระหว่าง 0-9, โดยที่ 0 มีความสำคัญสูงสุด.
5. คลิก OK. ค่าระดับความสำคัญของคิวงานจะถูกเปลี่ยนให้กับงานของคุณ. เช่น การเปลี่ยนงานระดับความสำคัญ 4 เป็น 3 จะเป็นการย้ายงานไปอยู่ใต้รายการงานทั้งหมดที่มีระดับความสำคัญ 3.
6. กด F5 เพื่อรีเฟรชหน้าต่าง Job Queue.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Change Job (CHGJOB)

พารามิเตอร์: JOBPTY

ตัวอย่าง: คำสั่งนี้จะเปลี่ยนระดับความสำคัญการจัดตารางเวลาสำหรับงาน PAYROLL เป็น 4. เนื่องจากมีการระบุเฉพาะชื่อ ธรรมชาติของงาน, จึงมีได้เพียงหนึ่งงานที่มีชื่อว่า PAYROLL เท่านั้นในระบบ. ถ้ามีมากกว่าหนึ่ง, คำดีฟอลต์ของ DUPJOB OPT(*SELECT) จะทำให้พานการเลือกถูกแสดงขึ้น ในงานแบบโต้ตอบ.

```
CHGJOB JOB(PAYROLL) JOBPTY(4)
```

คำแนะนำสำหรับการตั้งค่าระดับความสำคัญงาน:

โดยปกติ ระดับความสำคัญสำหรับงานที่รันในสภาวะแวดล้อมแบ็ตช์ควร ต่ำกว่าระดับความสำคัญสำหรับงานในสภาวะแวดล้อมแบบโต้ตอบ. นอกจากนี้, การแบ่งเวลา ควรน้อยพอที่โปรแกรมการวนซ้ำจะไม่ครอบครองเวลาของโปรเซสเซอร์ และ activity level.

คุณอาจต้องการให้ระดับความสำคัญสำหรับงานของผู้ควบคุมระบบสูงกว่า ระดับความสำคัญของงานอื่น เพื่อให้ผู้ควบคุมระบบสามารถตอบสนองต่อความต้องการ การระบบได้อย่างมีประสิทธิภาพ.

ถ้าคุณใช้ QCTL เป็นระบบย่อยการควบคุม, ผู้ควบคุมเครื่องจะรันที่ ระดับความสำคัญสูงกว่าโดยอัตโนมัติหลังจากการ signing on ที่คอนโซล. ที่เป็นเช่นนี้เพราะ QCTL เรดงานคอนโซลโดยใช้คลาส QCTL, ซึ่งระบุระดับความสำคัญที่สูงกว่า.

อีกวิธีหนึ่งที่คุณสามารถตั้งค่าระบบของคุณเพื่อให้ผู้ควบคุมเครื่อง สามารถรันที่ระดับความสำคัญสูงกว่าได้คือขั้นตอนต่อไปนี้:

1. ใส่เพิ่ม routing entry ที่ระบบย่อยพร้อมด้วยข้อมูลการเรดเฉพาะ และ ระบุคลาส QSYS/QCTL.
2. สร้างรายละเอียดของงานใหม่สำหรับผู้ควบคุมเครื่อง, โดยระบุข้อมูลการเรดเฉพาะที่เหมือนกับข้อมูลการเรดที่คุณใช้ใน routing entry.
3. เปลี่ยนโปรไฟล์ผู้ใช้ของผู้ควบคุมเครื่องเพื่อระบุรายละเอียดของงานใหม่.
4. ขณะนี้เมื่อผู้ควบคุมเครื่อง signs on ที่ระบบย่อยนั้น, งานจะเรดโดยใช้คลาส QCTL, ซึ่งระบุระดับความสำคัญที่สูงกว่า คลาสที่ใช้โดยงานแบบโต้ตอบปกติ.

ระดับความสำคัญการรันงานคือระดับความสำคัญสูงสุดซึ่ง thread ใดๆ ในงานสามารถรันได้. แต่ละ thread อาจมีระดับความสำคัญ thread ของตนเองซึ่ง ต่ำกว่าระดับความสำคัญงาน. คำสั่ง Change Job (CHGJOB) จะเปลี่ยน เฉพาะระดับความสำคัญงานเท่านั้น. สามารถใช้คำสั่ง Change Job (QWTCHGJB) API เพื่อเปลี่ยนระดับความสำคัญงานหรือระดับความสำคัญ thread อย่างไม่อย่างหนึ่ง.

การส่งงานครั้งเดียว:

เมื่อคุณต้องรันงานเพียงครั้งเดียว, ไม่ว่าจะรันในทันทีหรือรันในวันที่และ เวลาที่จัดตารางไว้, ให้ใช้คำสั่ง Submit Job (SBMJOB). วิธีนี้จะ วางงานบนคิวงานในทันที.

ถ้าต้องการส่งงานแบ็ตช์เพียงครั้งเดียว, ให้ใช้อินเตอร์เฟสแบบอักขระ.

คำสั่ง: Submit Job (SBMJOB)

คำสั่ง SBJOB จะส่งงานไปที่คิวงานแบ็ตช์โดยระบุรายละเอียดของงานและโดย ระบุคำสั่ง CL หรือข้อมูลคำขอ, หรือระบุข้อมูลการเรดที่จะรัน โปรแกรม. ถ้าคุณต้องการรันคำสั่ง CL เดียวในงานแบ็ตช์, ให้ใช้พารามิเตอร์ CMD บน SBJOB, ซึ่งจะ ตรวจสอบไวยากรณ์และอนุญาตการพร้อมท์.

ตัวอย่าง: ในตัวอย่างต่อไปนี้, คำสั่ง SBMJOB จะส่งงานที่มีชื่อว่า WSYS, โดยใช้รายละเอียดของงาน QBATCH, ไปที่คิวงาน QBATCH. พารามิเตอร์ CMD ให้คำสั่ง CL ที่จะรันในงาน .

```
SBMJOB JOB(QBATCH) JOB(WSYS) JOBQ(QBATCH) CMD(WRKSYSSTS)
```

การดูข้อมูล job affinity:

แต่ละงานบน iSeries มีข้อมูล affinity (ความสัมพันธ์) ของหน่วยความจำและโปรเซสเซอร์.

ข้อมูล affinity อธิบายว่า threads จะมีความสัมพันธ์กับ กลุ่มของโปรเซสเซอร์และหน่วยความจำเดียวกันกับ initial thread เมื่อ thread ถูกเริ่มต้นหรือไม่. และยังระบุถึงระดับของระบบที่พยายามจะรักษาไว้ซึ่งความสัมพันธ์ระหว่าง threads และ เซ็ตย่อยของรีซอร์สระบบที่ได้กำหนดไว้แล้ว. นอกจากนี้, ข้อมูล affinity ยังระบุว่า งานจะถูกแบ่งกลุ่มพร้อมกับงานอื่น เพื่อให้งานมีความสัมพันธ์กับเซ็ตย่อยเดียวกันกับเซ็ตย่อยของรีซอร์สของระบบหรือไม่.

โดยการรวมกลุ่ม threads เข้าด้วยกันที่จะแบ่งใช้งานชุดของข้อมูลทั่วไปในแหล่งเก็บข้อมูลหลัก, การแคชของระบบของคุณ และอัตราการเข้าถึงข้อมูลหน่วยความจำจะดีขึ้น.

iSeries Navigator:

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

หมายเหตุ: คุณสามารถดูข้อมูล affinity ของงานจากตำแหน่งใดๆ ก็ได้ภายในการจัดการระบบงาน ที่ซึ่งคุณสามารถเรียกดูงานได้.

2. คลิกขวางานที่คุณต้องการดู, และคลิก Properties.
3. บนหน้า Resources, คุณสามารถดูข้อมูล ความสัมพันธ์ของหน่วยความจำ และโปรเซสเซอร์ได้.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work with Job (WRKJOB)

เลือก อ็อปชัน 3: แสดงแอตทริบิวต์ job run, ถ้าแอคทีฟ

การจัดการรายละเอียดของงาน

เนื่องจากรายละเอียดของงานมีชุดเฉพาะของแอตทริบิวต์ที่เกี่ยวข้องกับงาน, รายละเอียดของงานเดียวกันจึงสามารถถูกใช้โดยงานหลายงานได้. ดังนั้น, ถ้าคุณใช้รายละเอียดของงาน, คุณไม่ต้องระบุพารามิเตอร์เดียวกันซ้ำสำหรับแต่ละงาน. คุณสามารถสร้างรายละเอียดของงานเพื่ออธิบายงานแบ็คซ์ หรืองานแบบโต้ตอบ. คุณยังสามารถสร้าง descriptions เฉพาะสำหรับผู้ใช้แต่ละรายของระบบได้ด้วย. รายละเอียดของงานถูกสร้างและจัดการโดยใช้อินเตอร์เฟซแบบอักษร.

การสร้างรายละเอียดของงาน:

ใช้อินเตอร์เฟซแบบอักษรและคำสั่ง Work With Job Description (WRKJOBDD) หรือคำสั่ง Create Job Description (CRTJOBDD) เพื่อสร้างรายละเอียดของงาน.

คำสั่ง: Create Job Description (CRTJOBDD)

ตัวอย่าง: ในตัวอย่างนี้, รายละเอียดของงานถูกสร้างโดยมีชื่อว่า INT4 ในไลบรารีปัจจุบันของผู้ใช้. รายละเอียดของงานนี้ใช้สำหรับงานแบบโต้ตอบ และถูกใช้โดย Department 127. เมื่อคุณ sign on, คุณต้องพิมพ์รหัสผ่าน. อักษร QCMDI ถูกใช้เป็น

ข้อมูลการเรดท์ซึ่งจะนำไปเปรียบเทียบกับตาราง การเรดท์ของระบบย่อยที่งานรันอยู่. ข้อความสอบถามทั้งหมดจะถูกเปรียบเทียบกับ entries ในรายการคำตอบของระบบ เพื่อกำหนดว่าจะมีออกคำตอบ โดยอัตโนมัติหรือไม่.

```
CRTJOBDB  JOBDB(INT4)  USER(*RQD)  RTGDTA(QCMDI)
           INQMSGRPY(*SYSRPLY)
           TEXT('Interactive #4 JOBDB for Department 127')
```

คำสั่งนี้จะสร้างรายละเอียดของงานที่มีชื่อว่า BATCH3 ใน โลกวารีย์ปัจจุบันของผู้ใช้. งานที่ใช้ description นี้จะถูกวางบน คิวงาน NIGHTQ. ระดับความสำคัญสำหรับงานที่ใช้ description นี้และเอาต์พุต ที่สพูลคือ 4. QCMDDB คือข้อมูลการเรดท์ที่จะนำไปเปรียบเทียบกับ entries ในตารางการเรดท์ของระบบย่อยที่งานจะรัน. จะใช้ไคด์บัญชีผู้ใช้ NIGHTQ012345 เมื่อบันทึกสถิติเกี่ยวกับบัญชีผู้ใช้สำหรับงานที่ใช้รายละเอียดของงานนี้.

```
CRTJOBDB  JOBDB(BATCH3)  USER(*RQD)  JOBQ(NIGHTQ)  JOBPTY(4)
           OUTPTY(4)  ACGCDE(NIGHTQ012345)  RTGDTA(QCMDDB)
           TEXT('Batch #3 JOBDB for high priority night work')
```

หมายเหตุ: โดยปกติ ค่าในรายละเอียดของงานจะถูกใช้เป็น ค่าดีฟอลต์ของพารามิเตอร์ที่สอดคล้องกันในคำสั่ง Batch Job (BCHJOB) และ Submit Job (SBMJOB) ถ้าไม่มีการระบุพารามิเตอร์. ค่าในรายละเอียดของงาน สามารถถูกแทนที่ค่าเดิมโดยค่าที่ระบุบนคำสั่ง BCHJOB และ SBMJOB

การเปลี่ยนรายละเอียดของงาน:

ใช้อินเตอร์เฟซแบบอักษรและคำสั่ง Work With Job Description (WRKJOBDB) หรือคำสั่ง Change Job Description (CHGJOBDB) เพื่อเปลี่ยนรายละเอียดของงาน.

คำสั่ง: Change Job Description (CHGJOBDB)

งานทั้งหมดซึ่งใช้รายละเอียดของงานที่ถูกเริ่มหลังจากการเปลี่ยน รายละเอียดของงานจะได้รับผลกระทบ. ถ้าคุณเปลี่ยนพารามิเตอร์งานเป็นค่าที่แตกต่างจาก ค่าที่ระบุในรายละเอียดของงาน , พารามิเตอร์นั้นจะไม่ได้รับผลกระทบ.

การใช้รายละเอียดของงาน:

วิธีทั่วไปในการใช้รายละเอียดของงานคือโดยการระบุรายละเอียดของงาน ในคำสั่ง Submit Job (SBMJOB). พารามิเตอร์ job description (**JOBDB**) คือที่ที่คุณระบุรายละเอียดของงานที่คุณต้องการให้งานนี้ใช้. เมื่อคุณกำหนดงานแบ็ตช์, คุณสามารถใช้รายละเอียดของงานในหนึ่งจากสอง วิธีดังนี้:

- ใช้รายละเอียดของงานที่ระบุโดยไม่ได้แทนที่ค่าเดิมของแอตทริบิวต์. ตัวอย่างเช่น:

```
SBMJOB JOB(OEDAILY) JOBDB(QBATCH)
```

- ใช้รายละเอียดของงานที่ระบุแต่แทนที่ค่าเดิมของบางแอตทริบิวต์ (โดยใช้คำสั่ง BCHJOB หรือ SBMJOB). ตัวอย่างเช่น, ถ้าต้องการแทนที่ค่าเดิมของไฟล์บันทึกข้อความในรายละเอียดของงาน QBATCH, ให้คุณระบุ:

```
SBMJOB JOB(OEDAILY) JOBDB(QBATCH) LOG(2 20 *SECLVL)
```

ข้างล่างนี้เป็นคำสั่งเพิ่มเติมที่สนับสนุนพารามิเตอร์รายละเอียดของงาน :

- Batch Job (BCHJOB): คำสั่งนี้บังคับจุดเริ่มต้นของงานแบ็ตช์ใน batch input stream. คำสั่งนี้ยังสามารถระบุค่าที่แตกต่างสำหรับ แอตทริบิวต์สำหรับงานแทนแอตทริบิวต์ที่ระบุในรายละเอียดของงาน หรือโปรไฟล์ผู้ใช้สำหรับงานนี้. ค่าที่มีอยู่ในรายละเอียดของงาน หรือในโปรไฟล์ผู้ใช้ที่มีชื่อในรายละเอียดของงานจะถูกใช้สำหรับพารามิเตอร์ ส่วนใหญ่ที่ไม่ได้ระบุในคำสั่ง BCHJOB .

- Add Prestart Job Entry (ADDPJE): คำสั่ง Add Prestart Job Entry (ADDPJE) ใส่เพิ่ม prestart job entry ที่ subsystem description ที่ระบุ.entry ระบุงานแบบ prestart ที่อาจถูกเริ่มขึ้น เมื่อระบบย่อยเริ่มต้นหรือเมื่อมีการป้อนคำสั่ง Start Prestart Jobs (STRPJ).
- Add Autostart Job Entry (ADDAJE): คำสั่ง Add Autostart Job Entry (ADDAJE) ใส่เพิ่ม autostart job entry ที่ subsystem description ที่ระบุ.entry ระบุชื่องานและ รายละเอียดของงานที่จะใช้เพื่อเริ่มงานโดยอัตโนมัติ.
- Add Work Station Entry (ADDWSE): คำสั่ง Add Work Station Entry (ADDWSE) ใส่เพิ่ม workstation entry ที่ subsystem description ที่ระบุ. แต่ละ entry อธิบายถึงหนึ่งหรือ หลาย workstations ที่ถูกควบคุมโดยระบบย่อย. workstations ที่ระบุ ใน workstation entries สามารถ sign on หรือเข้าสู่ระบบย่อยและ ใช้งานได้.

หมายเหตุ: คุณไม่สามารถแทนที่ค่าเดิมแอ็ดทริบิวต์รายละเอียดของงานสำหรับงานแบบ autostart, งานเวิร์กสเตชัน, หรือ งานสื่อสาร.

การควบคุมซอร์สของแอ็ดทริบิวต์งาน:

แอ็ดทริบิวต์ที่ระบบย่อยจะกำหนดให้กับงานมาจากซอร์สห้าซอร์ส; รายละเอียดของงาน, โปรไฟล์ผู้ใช้ของผู้ใช้, ค่ากำหนดของระบบ, งาน ที่ออกคำสั่ง Submit Job (SBMJOB), และเวิร์กสเตชัน (งานแบบโต้ตอบเท่านั้น). คุณควบคุมซอร์สที่ระบบย่อยจะดึงข้อมูลแอ็ดทริบิวต์งานเฉพาะ มาได้ โดยการระบุซอร์สในรายละเอียดของงาน. ถ้าต้องการ แก้ไขรายละเอียดของงาน, ให้ใช้อินเตอร์เฟสแบบอักษร.

คำสั่ง: Change Job Description (CHGJOB)

เพื่อควบคุม แอ็ดทริบิวต์งานและบอกให้ระบบย่อยทราบว่า จะเรียกแอ็ดทริบิวต์งานจากอ็อบเจกต์ ระบบอื่นใดที่ไหนและเมื่อใด, ให้ใช้อย่างใดอย่างหนึ่งดังต่อไปนี้:

- *JOB: บอกให้งานเรียกแอ็ดทริบิวต์ของงานมาจากรายละเอียดของงาน.
- *USRPRF: บอกให้งานเรียกแอ็ดทริบิวต์ของงานมาจากโปรไฟล์ผู้ใช้ของผู้ใช้.
- *SYSVAL: บอกให้งานเรียกแอ็ดทริบิวต์ของงานมาจากค่ากำหนดของระบบ.
- *CURRENT: บอกให้งานเรียกแอ็ดทริบิวต์ของงานมาจากงานที่ออกคำสั่ง Submit Job (SBMJOB).
- *WRKSTN: บอกให้งานเรียกแอ็ดทริบิวต์ของงานมาจากเวิร์กสเตชันที่มีงาน (งานแบบโต้ตอบเท่านั้น).

การลบรายละเอียดของงาน:

ใช้อินเตอร์เฟสแบบอักษรและคำสั่ง Work With Job Description (WRKJOB) หรือคำสั่ง Delete Job Description (DLTJOB) ในการลบรายละเอียดของงาน.

คำสั่ง: Delete Job Description (DLTJOB)

หมายเหตุ: งานที่อยู่ระหว่างการประมวลผลแล้วจะไม่ได้รับผลกระทบโดยคำสั่งนี้.

การจัดการงานแบ็คซ์

งานที่ไม่ต้องการข้อมูลผู้ใช้ในการรันสามารถถูกประมวลผล เป็นงานแบ็คซ์. โดยปกติ งานแบ็คซ์เป็นงานที่มีระดับความสำคัญต่ำและอาจ ต้องการสภาวะแวดล้อมระบบพิเศษในการรัน.

การส่งงานแบ็คซ์:

เนื่องจากโดยปกติงานแบ็ตช์เป็นงานที่มีระดับความสำคัญต่ำที่ต้องการ สภาวะแวดล้อมระบบพิเศษในการรัน (เช่น การรันในเวลากลางคืน) งานแบ็ตช์จึงถูกวางในคิวงานแบ็ตช์. ในคิวงาน งานแบ็ตช์จะได้รับตารางเวลารันใหม่ และระดับความสำคัญ. ถ้าต้องการส่งงานไปที่คิวงานแบ็ตช์, ให้คุณ ใช้อินเตอร์เฟซแบบอักษรและหนึ่งในสองคำสั่งต่อไปนี้.

คำสั่ง: Submit Job (SBMJOB)

คำสั่ง: Submit Database Job (SBMDBJOB)

ความแตกต่าง ระหว่างสองคำสั่งนี้คือซอร์สของงาน:

- คำสั่ง SBJOB จะส่งงานไปที่คิวงานแบ็ตช์โดยการระบุรายละเอียดของงานและโดยการระบุคำสั่ง CL หรือข้อมูลคำขอ, หรือระบุข้อมูลการเรดที่จจะรันโปรแกรม. ถ้าคุณต้องการรันคำสั่ง CL เดียวในงานแบ็ตช์หนึ่ง, ให้ใช้พารามิเตอร์ CMD บน SBJOB, ซึ่งจะทำให้การ ตรวจสอบไวยากรณ์และอนุญาตการพร้อมตัว.
- คำสั่ง SBMDBJOB สามารถใช้เพื่อส่งงานจากไฟล์ฐานข้อมูล ไปที่คิวงานแบ็ตช์. สำหรับงานเหล่านี้, รายละเอียดของงาน จะมาจากข้อความ BCHJOB ใน input stream.

ตัวอย่าง: ในตัวอย่างต่อไปนี้, คำสั่ง SBJOB จะส่งงานที่มีชื่อว่า WSYS, โดยใช้รายละเอียดของงาน QBATCH, ไปที่คิวงาน QBATCH. พารามิเตอร์ CMD ให้คำสั่ง CL ที่จะรันในงาน.

```
SBMJOB JOB(QBATCH) JOB(WSYS) JOB(QBATCH) CMD(WRKSYSSTS)
```

หมายเหตุ: หากคุณได้รับข้อความแจ้งว่างานไม่ถูกส่ง, คุณสามารถแสดง สหุผลไฟล์บันทึกการใช้งานเพื่อค้นหาข้อผิดพลาด. ใช้คำสั่ง WRKJOB. ระบุงานที่ไม่ถูกจัดตารางเวลา, เลือกอ็อปชัน 4 สำหรับสหุผลไฟล์. แสดงสหุผลไฟล์บันทึกการใช้งานเพื่อค้นหาข้อผิดพลาด.

การใช้ไฟล์ข้อมูลออนไลน์:

ไฟล์ข้อมูลออนไลน์คือไฟล์ข้อมูลที่อยู่เป็นส่วนหนึ่งของงานแบ็ตช์ เมื่องานถูกอ่านโดยเครื่องอ่านหรือคำสั่งการส่งงาน. คุณ ใช้ คำสั่ง SBMDBJOB หรือ STRDBRDR เพื่อจัดคิว CL batch stream (stream ของคำสั่ง CL ที่จะดำเนินการหรือรัน). CL batch stream นั้นสามารถรวมข้อมูลที่จะวางเข้าในไฟล์ "ชั่วคราว" (ไฟล์ออนไลน์). เมื่องานสิ้นสุดลง, ไฟล์ออนไลน์จะถูกลบออก.

ไฟล์ข้อมูลออนไลน์ถูกค้นในงานโดยคำสั่ง //DATA ที่ตอนต้นของไฟล์ และถูกค้นโดย end-of-data delimiter ที่ตอนท้ายของไฟล์.

end-of-data delimiter อาจเป็นสตริงอักขระที่ผู้ใช้กำหนด หรือค่าตีฟอลต์ของ //. // ต้องปรากฏในตำแหน่ง 1 และ 2. ถ้าข้อมูลของคุณ มี // ในตำแหน่ง 1 และ 2, คุณควรจะใช้เซตของอักขระที่ไม่ซ้ำกัน เช่น: // *** END OF DATA เพื่อระบุว่าสิ่งนี้เป็น end-of-data delimiter เฉพาะ, พารามิเตอร์ ENDCHAR บนคำสั่ง //DATA ควรเป็นโค้ดเช่น:

```
ENDCHAR('// *** END OF DATA')
```

หมายเหตุ: ไฟล์ ข้อมูลออนไลน์สามารถเข้าถึงได้เฉพาะในระหว่างขั้นตอนการเรดแรกของงานแบ็ตช์เท่านั้น. ถ้างานแบ็ตช์มี คำสั่ง Transfer Job (TFRJOB), Reroute Job (RRTJOB), หรือ Transfer Batch Job (TFRBCHJOB), ไฟล์ข้อมูลออนไลน์จะไม่สามารถเข้าถึงได้ในขั้นตอนการเรดที่ใหม่.

ไฟล์ข้อมูล ออนไลน์สามารถเป็นแบบระบุชื่อหรือแบบไม่ระบุชื่อ อย่างใดอย่างหนึ่ง. สำหรับไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อ, อาจมีการระบุ QINLINE เป็นชื่อไฟล์ในคำสั่ง //DATA หรือไม่มีการระบุชื่อ อย่างใดอย่างหนึ่ง. สำหรับไฟล์ข้อมูลออนไลน์ที่ระบุชื่อ, จะมีการระบุชื่อไฟล์.

ไฟล์ข้อมูลออนไลน์แบบระบุชื่อมีลักษณะดังต่อไปนี้:

- มีชื่อที่ไม่ซ้ำกันในงาน. ไฟล์ข้อมูลออนไลน์ต้องมีชื่อที่ไม่ซ้ำกัน.
- สามารถใช้ได้มากกว่าหนึ่งครั้งในงานหนึ่ง.
- ในแต่ละครั้งที่เปิดไฟล์, ไฟล์จะถูกจัดตำแหน่งอยู่ที่เร็กคอร์ดแรก.

ในการใช้ไฟล์ข้อมูลออนไลน์แบบระบุชื่อ, คุณต้องระบุชื่อไฟล์ในโปรแกรม หรือใช้คำสั่งแทนที่เพื่อเปลี่ยนชื่อไฟล์ที่ระบุในโปรแกรมเป็นชื่อของไฟล์ข้อมูลออนไลน์ อย่างใดอย่างหนึ่ง. ไฟล์ต้องถูกเปิดสำหรับการอินพุตเท่านั้น.

ไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อมีลักษณะดังต่อไปนี้:

- ชื่อของไฟล์คือ QINLINE. (ในงานแบ็ตซ์, ไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อทั้งหมดจะ ถูกตั้งชื่อเหมือนกัน.)
- สามารถใช้ได้ครั้งเดียวเท่านั้นในงานหนึ่ง.
- เมื่อมีไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่ออยู่มากกว่าหนึ่งไฟล์ในงาน, ไฟล์ต้องอยู่ใน input stream ในลำดับเดียวกับลำดับเมื่อไฟล์ถูกเปิด.

ถ้าต้องการใช้ไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อ, ให้ทำอย่างใดอย่างหนึ่งดังนี้:

- ระบุ QINLINE ในโปรแกรม.
- ใช้คำสั่งการแทนที่ไฟล์เพื่อเปลี่ยนชื่อไฟล์ที่ระบุในโปรแกรม เป็น QINLINE.

ถ้าภาษาชั้นสูงของคุณต้องการชื่อไฟล์ที่ไม่ซ้ำกันในหนึ่งโปรแกรม, คุณสามารถใช้ QINLINE เป็นชื่อไฟล์ได้เพียงครั้งเดียวเท่านั้น. ถ้าคุณต้องการใช้ไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อมากกว่าหนึ่งไฟล์, คุณสามารถใช้คำสั่งการแทนที่ไฟล์ในโปรแกรมเพื่อระบุ QINLINE สำหรับไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อเพิ่มเติม.

หมายเหตุ: ถ้าคุณรันคำสั่งแบบมีเงื่อนไขและประมวลผลไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อ มากกว่าหนึ่งไฟล์, จะไม่สามารถคาดการณ์ผลลัพธ์ได้ถ้าใช้ไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อที่ไม่ถูกต้อง.

ข้อควรพิจารณาในการเปิดไฟล์ข้อมูลออนไลน์:

ข้อควรพิจารณาสำหรับการเปิดไฟล์ข้อมูลออนไลน์มีดังต่อไปนี้:

- ความยาวเร็กคอร์ดระบุความยาวของอินพุตเร็กคอร์ด. (ความยาวเร็กคอร์ด เป็นสิ่งที่เลือกได้.) เมื่อความยาวเร็กคอร์ดเกินกว่าความยาวของข้อมูล, จะมีการส่ง ข้อความไปที่โปรแกรมของคุณ. ข้อมูลจะถูกเสริมด้วยพื้นที่เปล่า. เมื่อความยาวเร็กคอร์ด น้อยกว่าความยาวของข้อมูล, เร็กคอร์ดจะถูกตัดปลาย.
- เมื่อมีการระบุไฟล์ในโปรแกรม, เซิร์ฟเวอร์จะค้นหาไฟล์ตามไฟล์ข้อมูล ออนไลน์ที่ระบุชื่อ ก่อนที่จะค้นหาไฟล์ในไลบรารี. ดังนั้น, ถ้าไฟล์ข้อมูลออนไลน์ที่ระบุชื่อเหมือนกับไฟล์ที่ไม่ใช่ไฟล์ข้อมูลออนไลน์, ไฟล์ข้อมูลออนไลน์จะถูกใช้เสมอ, แม้ว่าชื่อไฟล์ จะถูกต้องตามชื่อไลบรารี.
- ไฟล์ข้อมูลออนไลน์ที่ระบุชื่อสามารถถูกแบ่งใช้ระหว่างโปรแกรมต่างๆ ในงานเดียวกันได้โดยการระบุ SHARE(*YES) บนคำสั่งการสร้างไฟล์หรือการแทนที่ไฟล์. ตัวอย่างเช่น, ถ้าคำสั่งการแทนที่ไฟล์ระบุว่าไฟล์ที่มีชื่อว่า INPUT และ SHARE(*YES) อยู่ในงานแบ็ตซ์พร้อมกับไฟล์ข้อมูลออนไลน์ที่มีชื่อว่า INPUT, โปรแกรมใดๆ ที่รันในงานที่ระบุชื่อไฟล์ INPUT จะแบ่งใช้ไฟล์ข้อมูลออนไลน์ที่มีชื่อเหมือนกัน. ไฟล์ข้อมูลออนไลน์ที่ไม่ได้ระบุชื่อไม่ สามารถถูกแบ่งใช้ระหว่างโปรแกรมต่างๆ ในงานเดียวกัน.
- เมื่อคุณใช้ไฟล์ข้อมูลออนไลน์, คุณควรตรวจสอบให้แน่ใจว่าได้ระบุชนิดของไฟล์ ที่ถูกต้องบนคำสั่ง //DATA. ตัวอย่างเช่น, ถ้าไฟล์จะถูกใช้เป็น ไฟล์ต้นฉบับ, ชนิดของไฟล์บนคำสั่ง //DATA ต้องเป็นต้นฉบับ.

- ไฟล์ข้อมูลออนไลน์ต้องถูกเปิดสำหรับอินพุตเท่านั้น.

การเริ่มงานแบ็คชิ่งที่กำลังรออยู่ในคิวงาน:

ในบางโอกาส คุณอาจต้องการบังคับให้งานเริ่มในทันที. ในขณะที่การย้ายงานไปที่คิวงานที่ไม่ยุ่งเป็นวิธีการที่มีประสิทธิภาพสูงสุด, แต่มีวิธีการอื่นที่คุณสามารถใช้ได้.

เพื่อเริ่มงานแบ็คชิ่ง, ในอันดับแรกให้ตรวจสอบสถานะของคิวงานที่งานตั้งอยู่และพิจารณาว่าการย้ายงานไปยังคิวอื่นเป็นวิธีที่เหมาะสมที่สุดสำหรับสถานการณ์ของคุณหรือไม่. (My Connections → server → Work Management → Job Queues → Active Job Queues หรือ All Job Queues)

ถ้าการย้ายงานไปที่คิวงานอื่นเป็นสิ่งที่ไม่เหมาะสม, คุณสามารถพักงานที่รันอยู่แล้วเปลี่ยนระดับความสำคัญของงานที่คุณต้องการเริ่ม. อย่างไรก็ตาม, ควรระวังเมื่อใช้วิธีการนี้ เนื่องจากงานที่ถูกพัก ยังคงรวมอยู่ในการนับงานแอ็คทีฟสูงสุด.

ถ้าต้องการเปลี่ยนระดับความสำคัญของงานและระยะเวลาที่งานควรรัน, ให้ทำตามต่อไปนี้:

1. คลิกขวาที่งานและคลิก **Properties**.
2. บนหน้าต่าง Job Properties, ให้คลิกแท็บ **Job Queue**.
3. เปลี่ยน **Priority on job queue** เป็นระดับความสำคัญ ที่สูงกว่า (0 คือสูงสุด).
4. เช็ท **When to make job available to run** เป็น Now หรือระบุวันที่และเวลา อย่างใดอย่างหนึ่ง.
5. คลิก **OK**.

การจัดการงานแบบโต้ตอบ

งานแบบโต้ตอบเริ่มต้นเมื่อคุณ sign on ที่ระบบหรือ, ถ่ายโอน ไปที่งานรองหรืองานกลุ่ม. งานแบบโต้ตอบจบลงเมื่อคุณ sign off. การทำงานจากจอภาพ, ช่วยให้คุณสามารถโต้ตอบกับระบบโดยการออก คำสั่ง, การใช้ฟังก์ชันคีย์, และการรันโปรแกรมและแอ็พพลิเคชัน. ข้อมูล ต่อไปนี้อธิบายวิธีการต่างๆ สำหรับการจัดการและการควบคุมงานแบบโต้ตอบ.

การควบคุมงาน inactive และเวิร์กสเตชัน:

คุณสามารถควบคุมจำนวนเวลาที่เวิร์กสเตชันยังคง inactive ก่อนที่ระบบย่อยจะส่งข้อความ (เรียกว่าโทมเอด) ได้โดยการระบุช่วงเวลา ในค่ากำหนดของระบบ Time-out interval for inactive jobs (QINACTITV). การควบคุมงาน inactive เป็นการรักษาความปลอดภัยเพื่อไม่ให้ผู้ใช้ปล่อยให้ จอแสดงผลที่ signed on inactive.

ระบบกำหนดว่าเวิร์กสเตชัน inactive ได้อย่างไร

ระบบย่อย จะกำหนดว่าเวิร์กสเตชันนั้น inactive ถ้าเงื่อนไขต่อไปนี้เป็นจริง:

- งานไม่ได้ประมวลผล transactions เพิ่มเติมใดๆ ในระหว่าง ช่วงของตัวจับเวลา.

หมายเหตุ: transaction คือการโต้ตอบของผู้ควบคุมเครื่องใดๆ, เช่น การเลื่อนจอภาพ, การกดปุ่ม enter, การกดฟังก์ชันคีย์, และอื่นๆ. การพิมพ์ข้อมูลที่ เวิร์กสเตชันโดยไม่กดปุ่ม enter ไม่ถือว่าเป็น transaction. ถ้างานที่เวิร์กสเตชัน ไม่ตรงกับเกณฑ์การ inactive, งานจะถูกพิจารณาว่า active.

- สถานะของงานคือรอการแสดงผล.
- งานไม่ถูกตัดการเชื่อมต่อ.
- สถานะของงานไม่เปลี่ยนแปลง.

- ระบบย่อยที่งานกำลังรันอยู่ไม่ได้อยู่ในสภาพที่ถูกจำกัด.

การจัดการงาน inactive

เพื่อจัดการงาน inactive ที่พบในระบบ, ให้ใช้ค่ากำหนดของระบบ When a job reaches time-out (QINACTMSGQ) . เพื่อกำหนดอ็อปชันการประมวลผลให้เลือกจากอ็อปชันดังต่อไปนี้:

- ตั้งค่ากำหนดของระบบ QINACTMSGQ เป็นชื่อ message queue.
ถ้าคุณระบุชื่อ message queue สำหรับค่ากำหนดของระบบ QINACTMSGQ, ผู้ใช้หรือ โปรแกรมสามารถ จะมอนิเตอร์ message queue และดำเนินการ action ที่จำเป็น, เช่นการจบงาน ได้.
ถ้าเวิร์กสเตชันที่มี secondary job pair มีสถานะ inactive, ระบบจะส่งข้อความสองข้อความ (หนึ่งข้อความสำหรับแต่ละงานของ secondary job pairs) ไปที่ message queue. จากนั้น ผู้ใช้หรือโปรแกรมสามารถใช้คำสั่ง ENDJOB กับหนึ่งหรือทั้งสองงานรอง, หรือคำสั่ง DSCJOB กับงานแอ็คทีฟที่จอแสดงผล อย่างไม่อย่างหนึ่ง.
- ตั้งค่ากำหนดของระบบ QINACTMSGQ เป็น *DSCJOB.
ถ้าคุณระบุ *DSCJOB สำหรับค่ากำหนดของระบบ QINACTMSGQ, ระบบจะตัดการเชื่อมต่องานทั้งหมดที่เวิร์กสเตชัน. ระบบส่งข้อความที่ระบุว่างานทั้งหมดที่เวิร์กสเตชันได้ถูกตัดการเชื่อมต่อ จาก QSYSOPR หรือ message queue ที่ตั้งค่า. (message queue ที่ตั้งค่า คือ message queue ที่ระบุในพารามิเตอร์ MSGQ ของ display device description. โดยค่าดีฟอลต์ค่านี้คือ QSYS หรือ QSYSOPR.) ถ้างานแบบโต้ตอบไม่สนับสนุนการตัดการเชื่อมต่องาน (เช่น, TELNET sessions ที่ใช้ QPADEVxxxx device descriptions), งานจะจบแทน.
ข้อความ จะถูกส่งสำหรับแต่ละช่วงเวลาที่งาน inactive ต่อไป.
- ตั้งค่ากำหนดของระบบ QINACTMSGQ เป็น *ENDJOB.
ถ้าคุณระบุ *ENDJOB สำหรับ ค่ากำหนดของระบบ QINACTMSGQ, ระบบจะจบงานทั้งหมดที่เวิร์กสเตชัน. ระบบส่งข้อความที่ระบุว่างานทั้งหมดที่เวิร์กสเตชันได้จบลงที่ QSYSOPR หรือ message queue ที่ตั้งค่า.

หมายเหตุ: งาน source pass-through , งาน client VTM (virtual terminal manager), และงาน 3270 device emulation ถูกแยกออกจากไทม์เอาต์ เนื่องจากเป็นงานที่ inactive เสมอ. นอกจากนี้ งาน MRT ในสภาวะแวดล้อม System/36 ยังถูกแยกออก ด้วยเนื่องจากเป็นงานแบ็คซ์.

การจบงานแบบโต้ตอบ:

คุณสามารถจบงานแบบโต้ตอบได้หลายวิธี .

1. คุณสามารถใช้ iSeries Navigator เพื่อจบงาน. จากหน้าต่าง Confirm Delete/End คุณสามารถระบุได้ว่า คุณต้องการให้ งานแบบโต้ตอบนี้จบในลักษณะที่ควบคุมหรือจบแบบทันที.
2. คุณสามารถใช้คำสั่งอินเตอร์เฟสแบบอักขระ End Job (ENDJOB) .
3. ถ้าต้องการจบงานแบบโต้ตอบโดยทันทีโดยใช้อินเตอร์เฟสแบบอักขระ, ให้ใช้คำสั่ง Sign Off (SIGNOFF) ที่เวิร์กสเตชัน. ถ้าต้องการจบการเชื่อมต่อผ่านทางเน็ตเวิร์ก, ให้ใช้พารามิเตอร์ end connection (ENDCNN) บนคำสั่ง SIGNOFF .
4. ถ้าต้องการตัดการเชื่อมต่องานทั้งหมดจากอุปกรณ์, ให้ใช้คำสั่ง Disconnect Job (DSCJOB) .

ถ้าต้องการใช้ iSeries Navigator และหน้าต่าง Confirm Delete/End, ให้ทำตามดังต่อไปนี้:

1. ขยาย My Connections → End point system → Work Management → Active Jobs.
2. คลิกขบวนการที่คุณต้องการจบ และคลิก Delete/End. หน้าต่าง Confirm Delete/End จะปรากฏขึ้น โดยคุณสามารถระบุวิธี และเวลาที่คุณต้องการให้งานแบบโต้ตอบจบได้ในหน้าต่างนี้.

หมายเหตุ: ถ้าต้องการจบงานแบบโต้ตอบทั้งหมดที่เกี่ยวข้องกับเวิร์กสเตชัน, หรืองานทั้งหมดที่เกี่ยวข้องกับกลุ่ม (ถ้างานเป็นงานกลุ่ม), ให้ตั้งค่าของฟิลด์ **Action for related interactive jobs** เป็น **End for group jobs** หรือ **End all** อย่างไม่อย่างหนึ่ง (สิ่งนี้เท่าเทียมกับพารามิเตอร์ **ADLINTJOBS** บนคำสั่ง **ENDJOB**).

และคุณยังสามารถร้องขอให้ระบบย่อย ส่งข้อความไปที่ message queue เมื่องานแบบโต้ตอบ inactive ตามช่วงเวลาที่จะระบุ. จากนั้นคุณ, หรือโปรแกรมที่มอนิเตอร์ message queue นั้น, สามารถจบหรือตัดการเชื่อมต่องานได้.

การตัดการเชื่อมต่องานทั้งหมดจากอุปกรณ์:

คำสั่ง **Disconnect Job (DSCJOB)** อนุญาตให้ผู้ใช้ แบบโต้ตอบตัดการเชื่อมต่อกับงานแบบโต้ตอบทั้งหมดที่เวิร์กสเตชัน และกลับไปที่จอแสดงผล signon. โลन्ที่สลับจะถูกยกเลิกเฉพาะถ้ามีการระบุใน workstation device description ของเวิร์กสเตชันนี้ และถ้า ไม่มีเวิร์กสเตชันอื่นบนโลन्นี้แอ็คทีฟ. ถ้างานถูกตัดการเชื่อมต่อเมื่อใช้งานถึง ช่วงตัดการเชื่อมต่อในค่ากำหนดของระบบ **Time-out interval for disconnected jobs (QDSCJOBITV)**, งานจะสิ้นสุดลงและไม่มีกรรมบันทึกการใช้งาน กับสพูลเอาต์พุตของงาน.

ข้อจำกัด:

1. งานที่จะตัดการเชื่อมต่อต้องเป็นงานแบบโต้ตอบ.
2. งานที่ถูกพักไว้ไม่สามารถตัดการเชื่อมต่อได้.
3. งานแบบ pass-through ไม่สามารถตัดการเชื่อมต่อ ยกเว้นว่าผู้ใช้ได้ใช้ฟังก์ชัน การร้องขอระบบเพื่อกลับไปที่ระบบซอร์สจากระบบเป้าหมายแบบ pass-through .
4. ต้องออกคำสั่งจากภายในงานที่จะตัดการเชื่อมต่อ, หรือ ผู้ออกคำสั่งต้องกำลังรันอยู่ภายใต้โปรไฟล์ผู้ใช้ซึ่งเหมือนกับ job user identity ของงานที่จะตัดการเชื่อมต่อ, หรือผู้ออกคำสั่ง ต้องกำลังรันอยู่ภายใต้โปรไฟล์ผู้ใช้ซึ่งมีสิทธิ์พิเศษ job control (*JOBCTL).
5. job user identity คือชื่อของโปรไฟล์ผู้ใช้ซึ่งมีงาน ที่เป็นที่รู้จักของงานอื่น.
6. งานจะไม่สามารถถูกตัดการเชื่อมต่อถ้า PC organizer แอ็คทีฟ.

คำสั่ง: **Disconnect Job (DSCJOB)**

ข้อควรพิจารณาการตัดการเชื่อมต่องาน:

มีหลายปัจจัยที่คุณต้องพิจารณาในทุกครั้งที่คุณ ตัดการเชื่อมต่องาน.

- อีอพนันบนเมนู System Request อนุญาตให้คุณตัดการเชื่อมต่องานแบบโต้ตอบ, ส่งผลให้จอแสดงผล signon ปรากฏขึ้น. อีอพนันจะเรียกคำสั่ง **Disconnect Job DSCJOB** .
- เมื่อเชื่อมต่อกับงานอีกครั้ง, ค่าที่ระบุบนจอแสดงผล signon สำหรับโปรแกรม, เมนู, และไลบรารีปัจจุบันจะถูกข้ามไป.
- งานซึ่งมีฟังก์ชัน PC organizer หรือ PC text assist แอ็คทีฟอยู่ไม่สามารถตัด การเชื่อมต่อได้.
- งาน TCP/IP TELNET สามารถตัดการเชื่อมต่อได้เฉพาะถ้าเซสชันกำลังใช้ device description ที่ผู้ใช้ระบุ ชื่อ เท่านั้น. คุณ สามารถสร้าง device description ที่ผู้ใช้ ระบุชื่อโดยใช้วิธีใดวิธีหนึ่งดังนี้:
 - การใช้ Network Stations พร้อมกับพารามิเตอร์ **DISPLAY NAME**
 - การใช้ iSeries Client Access support พร้อมกับฟังก์ชัน workstation ID
 - การใช้ TCP/IP TELNET Device Initialization exit point เพื่อระบุ ชื่อเวิร์กสเตชัน.
- ถ้าไม่สามารถตัดการเชื่อมต่องานเนื่องจากเหตุผลใดๆ, งานจะถูกจบแทน .

- งานที่ตัดการเชื่อมต่อทั้งหมดในระบบย่อยจะจบลง เมื่อระบบย่อยสิ้นสุดลง. ถ้า ระบบย่อยกำลังสิ้นสุด, จะไม่สามารถใช้คำสั่ง DSCJOB ในงานใดๆ ในระบบย่อย.
- สามารถใช้ค่ากำหนดของระบบ Disconnect Job Interval (QDSCJOBITV) เพื่อระบุช่วงเวลาที่สามารถตัดการเชื่อมต่อองานได้. ถ้าถึงช่วงเวลานั้นแล้ว, งานที่ตัดการเชื่อมต่อจะจบลง.
- งานที่ตัดการเชื่อมต่อที่ไม่เกินกว่าค่ากำหนดของระบบ QDSCJOBITV จะจบลง เมื่อระบบย่อยจบลงหรือเมื่อเกิดการ IPL.

การหลีกเลี่ยงฟังก์ชันที่ใช้เวลารันนานจากเวิร์กสเตชัน:

เพื่อหลีกเลี่ยงฟังก์ชันที่ใช้เวลารันนาน (เช่น บันทึก/คืนสภาพ) จาก เวิร์กสเตชันโดยไม่ทำให้ระบบชะงัก, ผู้ควบคุมระบบสามารถส่งงานไปที่ คิวงาน.

subsystem description QSYS/QBATCH หรือ QSYS/QBASE, ซึ่งจัดให้โดย IBM, มีคิวงาน QSYS/QBATCH ซึ่งสามารถใช้สำหรับวัตถุประสงค์นี้ได้. หากคุณสร้าง ระบบย่อยของคุณเอง, คุณควรอ้างอิงที่คิวงานสำหรับระบบย่อยนั้น. ผู้ควบคุมระบบสามารถส่งคำสั่งจากเมนูผู้ควบคุมระบบ.

ข้างล่างนี้เป็นตัวอย่างของการส่งคำสั่งที่ใช้เวลารันนาน:

```
SBMJOB JOB(SAVELIBX) JOB(QBATCH) JOBQ(QSYS/QBATCH)
CMD(SAVLIB LIBX DEV(DKT01))
```

การจัดการงานแบบ prestart

ใช้งานแบบ prestart เพื่อลดจำนวนเวลาที่ต้องใช้ในการจัดการ คำขอการเริ่มโปรแกรม. หัวข้อนี้อธิบายภารกิจทั่วไปที่เกี่ยวข้องกับงานแบบ prestart.

การเริ่มงานแบบ prestart:

โดยปกติงานแบบ prestart จะเริ่มในเวลาเดียวกับที่ระบบย่อยเริ่มขึ้น. คุณเริ่มงานแบบ prestart ด้วยตนเอง เมื่องานแบบ prestart ทั้งหมดได้ถูกจบ โดยระบบเนื่องจากข้อผิดพลาดหรืองานไม่ถูกเริ่มขึ้นในระหว่างที่ระบบย่อย เริ่มขึ้นเนื่องจาก STRJOBS (*NO) บน prestart job entry. ถ้าต้องการเริ่ม งานแบบ prestart, ให้ใช้อินเตอร์เฟสแบบอักษร.

คำสั่ง: Start Prestart Jobs (STRPJ)

คำสั่ง STRPJ ไม่ควรถูกใช้จนกว่าการเริ่มต้นของระบบย่อยที่เกี่ยวข้องจะเสร็จสมบูรณ์. เพื่อให้มั่นใจว่างานแบบ prestart ที่จำเป็นได้เริ่มเรียบร้อยแล้ว, ให้ใช้ไคต์ delay loop แล้วลองใหม่ถ้าคำสั่ง STRPJ ล้มเหลว.

จำนวนของ งานแบบ prestart ที่สามารถเ้าคที่พีได้ในเวลาเดียวกันถูกจำกัดโดยแอ็ตทริบิวต์ MAXJOBS บน prestart job entry และโดยแอ็ตทริบิวต์ MAXJOBS สำหรับระบบย่อย. แอ็ตทริบิวต์ MAXACT บน communications entry ควบคุม จำนวนของ คำขอการเริ่มโปรแกรมที่สามารถให้บริการได้ผ่านทาง communications entry ในเวลาเดียวกัน.

หมายเหตุ: ถ้าคุณระบุ *NO บนแอ็ตทริบิวต์ STRJOBS, จะไม่มีงานแบบ prestart ใดเริ่มขึ้นสำหรับ prestart job entry เมื่อระบบย่อยเริ่มต้น. การรันคำสั่ง STRPJ ไม่ได้ทำให้ค่าของพารามิเตอร์ STRJOBS เปลี่ยนแปลง.

ตัวอย่าง: ตัวอย่างนี้จะเริ่มงานแบบ prestart สำหรับ prestart job entry PJPGM ในระบบย่อย SBS1. ระบบย่อย SBS1 ต้องแอ็คทีฟในขณะที่ออก คำสั่งนี้. จำนวนของงานที่เริ่มต้นคือจำนวนที่ระบุในค่า INLJOBS ของ prestart job entry PJPGM. ระบบย่อยจะเริ่มโปรแกรม PJPGM ในไลบรารี PJLIB.

```
STRPJ SBS(SBS1) PGM(PJLIB/PJPGM)
```

การจัดคิวหรือการปฏิเสธคำขอการเริ่มโปรแกรม:

ถ้าคำขอการเริ่มโปรแกรมมาถึงเมื่อจำนวนปัจจุบันของงานแบบ prestart น้อยกว่าจำนวนที่ระบุในแอตทริบิวต์ MAXJOBS บน prestart job entry, และไม่มีงานแบบ prestart พร้อมใช้ในการจัดการคำขอการเริ่ม โปรแกรม, คุณมีทางเลือกที่จะปฏิเสธหรือจัดคิวคำขอใหม่นี้.

เพื่อปฏิเสธหรือจัดคิวคำขอการเริ่มโปรแกรม, ให้ใช้แอตทริบิวต์ WAIT บน prestart job entry.

WAIT(*NO) หมายความว่าถ้าไม่มีงานแบบ prestart พร้อมใช้งานในทันที, คำขอการเริ่มโปรแกรมจะถูกปฏิเสธ.

WAIT(*YES) หมายความว่าถ้าไม่มีงานแบบ prestart พร้อมใช้งานในทันทีและไม่มีงานแบบ prestart สามารถเริ่มได้เนื่องจาก MAXJOBS ในการให้บริการคำขอการเริ่มโปรแกรม, คำขอการเริ่มโปรแกรมจะถูกปฏิเสธ. ถ้าไม่มีงานแบบ prestart พร้อมใช้งานในทันที, แต่งานแบบ prestart เพิ่มเติม สามารถหรือได้ถูกเริ่ม, คำขอการเริ่มโปรแกรมจะถูกจัดคิว.

คำสั่งนี้จะใส่เพิ่ม prestart job entry สำหรับโปรแกรม PGM1 ในไลบรารี QGPL ที่ PJSBS subsystem description ที่มีอยู่ในไลบรารี QGPL. entry ระบุงานแบบ prestart 15 งาน (โปรแกรม PGM1 ในไลบรารี QGPL) จะถูกเริ่มขึ้น เมื่อระบบย่อย PJSBS ในไลบรารี QGPL เริ่มขึ้น. เมื่อพูลของ งานแบบ prestart ที่พร้อมใช้งานลดลงเป็นสี่ (เนื่องจากงานแบบ prestart กำลังให้บริการ คำขอที่ระบุสำหรับโปรแกรม PGM1 ในไลบรารี QGPL), งาน เพิ่มเติมสิบงานจะถูกเริ่มขึ้น. ถ้าไม่มีงานแบบ prestart พร้อมใช้งานสำหรับ entry นี้ เมื่อได้รับคำขอ, คำขอจะถูกปฏิเสธ.

```
ADDPJE SBS(QGPL/PJSBS) PGM(QGPL/PGM1) INLJOBS(15)
      THRESHOLD(5) ADLJOBS(10) WAIT(*NO)
```

การจบงานแบบ prestart:

คุณสามารถใช้ iSeries Navigator หรืออินเตอร์เฟซแบบอักษร เพื่อจบงานแบบ prestart และไฟล์ข้อมูลออนไลน์ที่เกี่ยวข้องสำหรับ prestart job entry ในระบบย่อยที่แอ็คทีฟ. งานสามารถกำลังรอสำหรับคำขอหรืออาจ เชื่อมโยงกับคำขอแล้ว. Spooled output files ที่เกี่ยวข้องกับงานที่กำลังจะ จบยังสามารถถูกจบหรืออนุญาตให้คงอยู่ต่อไปบนเอาต์พุตคิวได้. นอกจากนี้ยังสามารถเปลี่ยนข้อจำกัดจำนวนของข้อความที่จะบันทึกที่แต่ละบันทึกการใช้งานได้ด้วย.

iSeries Navigator:

ถ้าต้องการใช้ iSeries Navigator, ให้ทำตามต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **Work Management** → **Active Jobs**.
2. ระบุตำแหน่งงานแบบ prestart ที่คุณต้องการจบ.
3. คลิกขวาที่งานแบบ prestart และคลิก **Delete/End**.
4. กรอกข้อมูลในหน้าต่าง Confirm Delete/End ให้สมบูรณ์และคลิก **Delete**.

อินเตอร์เฟซแบบอักษร:

คำสั่ง : End Prestart Job (ENDPJ)

ตัวอย่าง: คำสั่งนี้จะจบงานทั้งหมดที่เชื่อมโยงกับ prestart job entry PJPGM ในระบบย่อย SBS1 โดยทันที. สพูลเอาต์พุตที่สร้างขึ้นโดย งานแบบ prestart เหล่านี้จะถูกลบออกและมีการบันทึกบันทึกการใช้งาน.

```
ENDPJ SBS(SBS1) PGM(PJLIB/PJPGM) OPTION(*IMMED)
      SPLFILE(*YES)
```

ตัวอย่าง: คำสั่งนี้จะจบงานทั้งหมดที่เชื่อมโยงกับ prestart job entry PJPGM2 ในระบบย่อย SBS2. สพูลเอาต์สำหรับงานแบบ prestart เหล่านี้จะถูกบันทึกสำหรับการประมวลผลปกติโดย spooling writer. งานต่างๆ มีเวลา 50 วินาทีในการปฏิบัติรูทีนการล้างข้อมูล, หลังจากนั้นงานจะถูกจบลงในทันที.

```
ENDPJ  SBS(SBS2)  PGM(PJPGM2)  OPTION(*CNTRLD)
        DELAY(50)  SPLFILE(NO)
```

การจัดการคลาสอ็อบเจกต์งาน

คลาสอ็อบเจกต์มีแอตทริบิวต์การรันที่ควบคุมสถานะแวดล้อมรันใหม่ ของงาน. คลาสอ็อบเจกต์ที่ได้มาจาก IBM, หรือคลาส, ใช้ได้กับทั้ง แอ็พพลิเคชันแบบโต้ตอบและแอ็พพลิเคชันแบบแบ็ชตามปกติ. คลาสที่งานใช้ ถูกระบุอยู่ใน subsystem description routing entry ที่ใช้ในการเริ่มงาน. ถ้างานมีขั้นตอนการเรอต์หลายขั้นตอน, คลาสที่ใช้โดย ขั้นตอนการเรอต์แต่ละขั้นตอนจะถูกระบุอยู่ใน routing entry ที่ใช้ใน การเริ่มต้นขั้นตอนการเรอต์.

การสร้างคลาสอ็อบเจกต์:

คุณสามารถสร้างคลาสอ็อบเจกต์โดยใช้อินเทอร์เฟซแบบอักขระ. คลาสกำหนดแอตทริบิวต์การประมวลผลสำหรับงานที่ใช้คลาส. คลาสที่งานใช้ ถูกระบุอยู่ใน subsystem description routing entry ที่ใช้ในการเริ่มงาน. ถ้างานมีขั้นตอนการเรอต์หลายขั้นตอน, คลาสที่ใช้โดย ขั้นตอนการเรอต์แต่ละขั้นตอนจะถูกระบุอยู่ใน routing entry ที่ใช้ใน การเริ่มต้นขั้นตอนการเรอต์.

คำสั่ง: Create Class (CRTCLS)

ตัวอย่าง: ตัวอย่างนี้จะสร้างคลาสที่มีชื่อว่า CLASS1. คลาสถูกจัดเก็บ อยู่ในไลบรารีปัจจุบันที่ระบุสำหรับงาน. ข้อความผู้ใช้ 'This class for all batch jobs from Dept 4836' เป็นคำอธิบายคลาส. แอ็ตทริบิวต์ของคลาสนี้แสดงระดับความสำคัญการรันเป็น 60 และการแบ่งเวลาเป็น 900 มิลลิวินาที. ถ้างานยังรันไม่เสร็จสิ้นหลังจากสิ้นสุดช่วงการแบ่งเวลา, งานจะไม่สามารถถูกย้ายออกจากแหล่งเก็บข้อมูลหลักได้ จนกว่างานจะได้รับการจัดสรรช่วงการแบ่งเวลาอื่น. สมมุติว่าใช้ค่าดีฟอลต์สำหรับพารามิเตอร์อื่น.

```
CRTCLS  CLS(CLASS1)  RUNPTY(60)  TIMESLICE(900)
        TEXT('This class for all batch jobs from Dept 4836')
```

การเปลี่ยนคลาสอ็อบเจกต์:

คุณสามารถเปลี่ยนแอตทริบิวต์ของคลาสอ็อบเจกต์ได้โดยใช้ อินเทอร์เฟซแบบอักขระ. สามารถเปลี่ยนแอตทริบิวต์ต่างๆ, ยกเว้นแอตทริบิวต์สิทธิ์พบลิก. อ้างอิงคำสั่ง Revoke Object Authority (RVKOBJAUT) และคำสั่ง Grant Object Authority (GRTOBJAUT) สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการเปลี่ยนการให้สิทธิ์อ็อบเจกต์.

คำสั่ง: Change Class (CHGCLS)

ตัวอย่าง: ตัวอย่างนี้จะเปลี่ยนคลาสที่มีชื่อว่า CLASS1 ในไลบรารี บนรายชื่อไลบรารีของงาน. ระดับความสำคัญการรันสำหรับคลาสจะถูกเปลี่ยน เป็น 60 และการแบ่งเวลาเป็น 900 มิลลิวินาที.

```
CHGCLS  CLS(CLASS1)  RUNPTY(60)  TIMESLICE(900)
```

การจัดการ threads

มีหลายภารกิจที่คุณสามารถทำได้เมื่อจัดการ threads.

การดู threads ที่กำลังรันภายใต้งานเฉพาะ:

งานแอ็คทีฟทุกงานที่กำลังรันบนระบบ iSeries มี thread ที่กำลังรันอยู่ภายใต้งานอย่างน้อยหนึ่ง thread. thread คือหน่วยงานอิสระที่รันอยู่ภายในงานที่ใช้ซอร์สเดียวกันกับงาน. เนื่องจากงานขึ้นอยู่กับงานที่ทำ โดย thread, จึงมีความสำคัญที่ต้องทราบวิธีการค้นหา threads ที่กำลังรัน ภายในงานเฉพาะ.

iSeries Navigator:

ถ้าต้องการดู thread ที่รันภายใต้งานเฉพาะ, ให้ทำดังต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → **server** → **Work Management** → **Active Jobs**.
2. คลิกขบวนการซึ่งคุณต้องการทำงาน, และคลิก **Details** → **Threads**.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work With Job (WRKJOB)

ตัวอย่าง: ตัวอย่างต่อไปนี้จะแสดงจอภาพ Work With Threads สำหรับงาน Crtpfrdta.

```
WRKJOB JOB(Crtpfrdta) OPTION(*THREAD)
```

คุณสามารถทำอะไรกับ threads:

เนื่องจาก threads ช่วยงานประมวผลมากกว่าหนึ่งการดำเนินการในแต่ละ ครั้งในขณะที่รัน, ดังนั้นการมอนิเตอร์ threads ที่กำลังรันอยู่ภายในงานหนึ่งอาจเป็น สิ่งจำเป็น. การทำเช่นนี้ช่วยให้คุณสามารถรันงานได้อย่างมีประสิทธิภาพ. คุณสามารถใช้ iSeries Navigator เพื่อค้นหา thread ที่คุณต้องการจัดการ.

หลังจากที่คุณระบุตำแหน่ง thread แล้ว, คุณสามารถคลิกขวาที่ thread และ เลือก action อย่างใดอย่างหนึ่งต่อไปนี้:

Reset Statistics

อนุญาตให้คุณรีเซ็ตข้อมูลรายการที่คุณกำลังดูอยู่, และเซตเวลาที่ผ่านไปให้เป็น 00:00:00.

Details เนื่องจากฟังก์ชันของ thread คล้ายกับฟังก์ชันของงาน, ดังนั้นฟังก์ชันทั้งสองจึงมีการดำเนินการบางอย่างเหมือนกัน. Details มีข้อมูลรายละเอียดเกี่ยวกับ thread actions ต่อไปนี้:

- Call stack
- รายชื่อไลบรารี
- Locked Objects
- Transaction
- Elapsed Performance Statistics

พักไว้ อนุญาตให้คุณพัก thread. สามารถพัก thread ได้หลายครั้ง. ระบบปฏิบัติการจะคอยตรวจสอบจำนวนครั้งที่ thread ถูกพัก.

รีลีส รีลีส thread ที่ถูกพักไว้. ต้องรีลีส thread ในแต่ละครั้งที่ถูกพักไว้เพื่อให้ thread รัน.

ลบออก/จบ

อนุญาตให้คุณจบหนึ่งหรือหลาย thread ที่เลือก.

Thread Properties

แสดงแอ็ททริบิวต์ต่างๆ ของ thread.

สำหรับข้อมูลรายละเอียดเพิ่มเติมเกี่ยวกับ actions ที่คุณสามารถทำบน threads, ให้ดูคำอธิบายทางออนไลน์ iSeries Navigator

การดูคุณสมบัติของ thread:

thread จะช่วยให้งานสามารถทำงานหลายๆ อย่างได้ในเวลาเดียวกัน. หาก thread หยุดการประมวลผล, thread จะหยุดงานไม่ให้รัน. หน้าต่าง Thread Properties ใน iSeries Navigator ช่วยให้คุณสามารถดู thread และแอ็ททริบิวต์ประสิทธิภาพ thread ต่างๆ ที่สามารถช่วยให้เข้าใจว่าเพราะเหตุใด thread จึงไม่ได้อันอยู่.

iSeries Navigator:

ถ้าต้องการดูแอ็ททริบิวต์ของ thread, ให้ทำตามต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → **server** → **Work Management** → **Active Jobs** หรือ **Server Jobs**.
2. คลิกชื่องานซึ่งคุณต้องการทำงาน, และคลิก **Details** → **Threads**.
3. คลิกขวา thread ที่คุณต้องการทำงาน, และคลิก **Properties**.

ข้อมูลภายใต้แท็บ General ช่วยคุณในการดูแอ็ททริบิวต์ของ thread. แอ็ททริบิวต์เหล่านี้รวมถึง thread identifier, สถานะโดยละเอียดของ thread, ผู้ใช้ปัจจุบัน, ชนิดของการรัน thread, งานที่ thread กำลังรันอยู่ข้างใต้, และกลุ่มดิสก์พูลที่ thread กำลังรันอยู่ข้างใน.

ข้อมูลภายใต้แท็บ Performance ช่วยคุณในการดูองค์ประกอบประสิทธิภาพ พื้นฐานและอนุญาตให้คุณเปลี่ยนระดับความสำคัญการรันของ thread ได้. **ระดับความสำคัญในการรัน** จะระบุความสำคัญของ thread ที่สัมพันธ์กับ thread อื่นๆ ที่รันในระบบ. ช่วงค่าจะเริ่มตั้งแต่ระดับความสำคัญของงานไปจนถึง 99 (หมายความว่าระดับความสำคัญสูงสุดจะต่างกันไป). ระดับความสำคัญการรัน thread ไม่มีทางสูงกว่าระดับความสำคัญในการรันงานที่ thread กำลังรัน.

คุณยังสามารถดูค่าประสิทธิภาพที่ถูกคำนวณไว้ นับตั้งแต่ที่ thread เริ่มทำงาน, ซึ่งรวมถึง CPU และ disk I/O รวม. นอกจากนี้คุณยังสามารถดู, รีเฟรช, เซ็ตออฟการรีเฟรชอัตโนมัติ, หรือรีเซ็ต **Elapsed performance statistics** ที่ถูกคำนวณไว้สำหรับ thread.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work With Job (WRKJOB)

ตัวอย่าง: ตัวอย่างต่อไปนี้จะแสดงจอภาพ Work With Threads สำหรับงาน Crtpfrdta.

```
WRKJOB JOB(Crtpfrdta) OPTION(*THREAD)
```

การจบหรือการลบ threads:

initial thread, ซึ่งถูกสร้างเมื่องานเริ่มทำงาน, ไม่สามารถถูกลบทิ้งหรือจบการทำงานได้. อย่างไรก็ตาม, บางครั้งก็เป็นเรื่องจำเป็นที่ต้องจบการทำงานของ thread รองเพื่อให้งานสามารถรันต่อไปได้. ให้ระวัง thread ที่คุณตั้งใจจะจบ เนื่องจากงานที่รันอยู่ภายใน thread อาจไม่สมบูรณ์ถ้าไม่มีงานของ thread นั้น.

สำคัญ: การจบการทำงานของ thread ไม่ควรเป็นส่วนหนึ่งของการจัดการงานประจำวัน. การจบ thread เป็นสิ่งที่มีความสำคัญมากกว่าการจบงาน เนื่องจากงานใน threads อื่นอาจจะหรืออาจจะไม่หยุด. เมื่อคุณจบงาน, งานทั้งหมดจะหยุดทำงาน. อย่างไรก็ตาม, เมื่อคุณจบ thread, จะมีเพียงบางส่วนของงานเท่านั้นที่หยุดทำงาน. threads อื่นอาจจะหรืออาจจะไม่รันต่อไป. ถ้า thread อื่น รันต่อไปโดยไม่มี thread ที่คุณจบ, thread เหล่านี้ อาจให้ผลลัพธ์ที่ไม่พึงปรารถนา.

ในการลบหรือจบ thread รอง, คุณต้องมีสิทธิ์พิเศษ service (*SERVICE) หรือสิทธิ์ Thread Control.

iSeries Navigator:

ถ้าต้องการลบหรือจบการทำงานของ thread, ให้ทำตามต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → **server** → **Work Management** → **Active Jobs** หรือ **Server Jobs**.
2. คลิกขบวนการที่คุณต้องการทำงานด้วย, และคลิก **Details**, แล้ว **Threads**.
3. คลิกขวา thread ที่คุณต้องการจบ, และคลิก **Delete/End**.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: **Work With Job (WRKJOB) Option 20: Work with threads**, ถ้าแอ็คทีฟ

ตัวอย่าง: ตัวอย่างต่อไปนี้จะแสดงจอภาพ Work With Threads สำหรับงาน Crtpfrdta.

```
WRKJOB JOB(Crtpfrdta) OPTION(*THREAD)
```

ที่จอภาพ Work With Threads, ให้เลือก อีพชัณ: 4=จบ.

การจัดการการจ้ดตารางเวลางาน

คุณสามารถจ้ดตารางเวลางานเพื่อรันโดยใช้ Advanced Job Scheduler, โดยการใช้น้้าต่าง iSeries Navigator Job Properties, หรือโดยการเปลี่ยน job schedule entry ผ่านทางอินเตอร์เฟซ แบบอักษร.

การจ้ดตารางเวลางานแบ้ตซ์โดยใช้ iSeries Navigator

น้้าต่าง Job Properties – Job Queue นำเสนอวิธีการจ้ดตารางเวลางานแบ้ตซ์เพื่อรันทันที, รันครั้งเดียวที่วันที่และเวลาที่ระบุ หรือรัน ในช่วงสม่่าเสมอ (เช่นวันที่หนึ่งของแต่ละเดือน).

ถ้าต้องการจ้ดตารางเวลางานโดยใช้ iSeries Navigator, ให้ทำตามต่อไปนี้:

1. ขยาย **My Connections** → **server** → **Work Management** → **Job Queues** → **Active Job Queues** หรือ **All Job Queues** → **คิวงานที่มีงานของคุณ**.
2. คลิกขวาที่งานและคลิก **Properties**.
3. บนน้้าต่าง Job Properties, ให้คลิกแท็บ Job Queues.
4. เพื่อจ้ดตารางเวลางาน, ให้ใช้อีพชัณที่อยู่ภายใต้ **When to make job available to run**.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับวิธีการใช้น้้าต่างนี้, ให้ดูคำอธิบาย iSeries Navigator .

การจ้ดตารางเวลางานโดยใช้ Management Central Scheduler

ถ้าคุณไม่ได้ติดตั้ง add-in Advanced Job Scheduler, คุณสามารถใช้ Management Central Scheduler เพื่อจ้ดตารางเวลางานได้.

คุณสามารถเรียกใช้ Management Central scheduler โดยการคลิกบนปุ่ม **Schedule** ที่จะปรากฏขึ้นบนหลายน้้าต่างของ iSeries Navigator. ตัวอย่างเช่น สมมุติว่าคุณต้องการใช้น้้าต่าง Run Command iSeries Navigator เพื่อส่งงานการล้างข้อมูล แต่คุณไม่ต้องการให้งานรันจนกว่าจะผ่านพ้น ชั่วโมงที่มีการใช้งานสูงสุด.

1. ใน iSeries Navigator คลิกขวาเซิร์ฟเวอร์ที่คุณต้องการจะรันงานการล้างข้อมูล และคลิก **Run Command**.

2. จากหน้าต่าง Run Command, ให้พิมพ์ไวยากรณ์แบบอักขระสำหรับการรันงานของคุณ. ถ้าคุณต้องการความช่วยเหลือ ให้พิมพ์คำสั่งแรกและคลิก Prompt.
3. เมื่อคุณทำคำสั่งเสร็จสมบูรณ์แล้ว, ให้คลิก Schedule. หน้าต่าง Management Central Scheduler จะแสดงขึ้น โดยในหน้าต่างนี้ คุณสามารถจัดตารางเวลานงานนี้ให้รันครั้งเดียว, หรือให้รันเป็นงานที่ต่อเนื่อง.

การจัดตารางเวลางานด้วย Advanced Job Scheduler

ข้อมูลต่อไปนี้จะช่วยคุณในการบริหาร Advanced Job Scheduler. คุณต้องติดตั้งไลเซนส์โปรแกรมก่อน, แล้วจึงอ่านเกี่ยวกับวิธีการที่จะช่วยให้คุณปรับแต่ง Advanced Job Scheduler ได้ตามความต้องการ. ท้ายที่สุด, ภารกิจส่วนที่เหลือจะช่วยให้คุณทำงานและบริหารตัวจัดตารางเวลานี้ได้.

มีอะไรใหม่ใน Advanced Job Scheduler for V5R4:

มีการพัฒนาหลายอย่างใน Advanced Job Scheduler.

การใส่เพิ่มหลายคำสั่งที่งานที่จัดตารางเวลา

- รายการคำสั่งคือชุดของคำสั่งที่จัดเก็บไว้ซึ่งเป็นคำสั่งที่ Advanced Job Scheduler ใช้ในการประมวลผลงาน Management Central. ขณะนี้คุณสามารถใส่เพิ่มชุดของคำสั่ง ที่งาน Management Central ที่จัดตารางเวลาไว้ และควบคุมลำดับการรันของ คำสั่งเหล่านั้นได้. ในอดีต คุณสามารถจัดตารางเวลางาน Management Central ได้ เพียงหนึ่งงานเท่านั้น (คุณสามารถจัดตารางเวลางานหนึ่งงานเพื่อรวบรวมสินค้าคงคลัง, และจัดตารางเวลาอีกงานหนึ่งเพื่อติดตั้ง โปรแกรมฟิกซ์, และอีกงานหนึ่งเพื่อรันคำสั่งต่าง ๆ). แต่ขณะนี้ คุณสามารถสร้างงาน Advanced Job Scheduler หนึ่งงานที่จะทำกิจกรรมเหล่านั้นทั้งหมดได้.

เมื่อคุณจัดตารางเวลางาน, คุณสามารถเลือกที่จะสร้างงานที่จัดตารางเวลาใหม่, หรือสร้างงานที่จัด ตารางเวลาใหม่โดยใช้ฐานจากงานที่จัดตารางเวลาที่มีอยู่, หรือใส่เพิ่มงานที่งานที่มีอยู่. นอกจากนี้ยังสามารถใช้คำสั่ง CL ที่งานที่จัดตารางเวลาของ Management Central ได้อีกด้วย. ตัวอย่างเช่น, ถ้าต้องการหน่วงเวลาระหว่างงานต่างๆ คุณสามารถใช้คำสั่ง Delay Job (DLYJOB).

งานจะถูกประมวลผลบนระบบจุดปลายที่เลือก เมื่องานถูกสร้างขึ้น. อย่างไรก็ตาม, คำสั่ง CL ทั้งหมดจะถูกประมวลผลบนระบบ Central. แต่ละงานต้องเสร็จสมบูรณ์ก่อนที่จะประมวลผลงานหรือคำสั่ง CL ถัดไปใน รายการได้.

หลังจากที่คุณคลิกปุ่ม Schedule สำหรับ งาน Management Central, หน้าต่างถัดไปจะพร้อมดีให้คุณระบุว่าคุณต้องการจะสร้างงานใหม่, สร้างงานใหม่โดยใช้ฐานจากงานอื่น, หรือใส่เพิ่มงานที่งานที่มีอยู่แล้ว.

ฟิลด์ Command อยู่บน หน้าต่าง Scheduled Job Properties – General. (My Connections → server → Work Management → Advanced Job Scheduler → Scheduled Jobs → คลิกขวาที่งาน → Properties)

การควบคุมแบนเนอร์การแจ้งเตือน

- เมื่อคุณแจกจ่ายสพูลไฟล์โดยใช้ Report Distribution, ขณะนี้คุณสามารถ เลือกจากรายการไอเท็ม และพิมพ์ไอเท็มในตัวอักษรขนาดใหญ่บนหน้า แบนเนอร์ของสพูลไฟล์ใหม่. ไอเท็มดีฟอลต์คือ Job name และ Spooled file name. คุณสามารถเลือกแบนเนอร์ที่จะพิมพ์ในตัวอักษรขนาดใหญ่ได้ 2 แบนเนอร์.

ฟิลด์ Available banner items อยู่บนหน้าต่าง Notification Properties. (My Connections → server → Work Management → Advanced Job Scheduler → คลิกขวา Notification → Properties)

การใส่เพิ่มเมนูอีพซันการส่งอีเมลบนระบบ

- เมนูข้อพจนใหม่ช่วยให้คุณสามารถใช้ Advanced Job Scheduler ในการส่ง อีเมล. เมื่อคุณเลือกไอเท็มเมนูนี้, หน้าต่าง New E-Mail Message จะปรากฏขึ้น. นี่เป็นหน้าต่างเดียวกันกับที่จะปรากฏขึ้นเมื่อคุณคลิก My Connections → server → Work Management → Advanced Job Scheduler → Notifications → คลิกขวา E-mail → New E-mail.

My Connections → คลิกขวา เซิร์ฟเวอร์ → Send e-mail via AJS

แจกจ่ายรายงานผ่านทาง Basic Operations container

- หน้าต่าง **Distribute Reports** แสดงให้คุณเห็นถึง ที่ที่จะแจกจ่ายสพูลไฟล์ด้วยตัวเอง โดยสพูลไฟล์เหล่านี้ถูกสร้างขึ้น โดยงานของผู้ใช้ที่กำลังใช้รายการการแจกจ่ายรายงานอยู่. งานอาจเป็นงานที่เริ่มโดย Advanced Job Scheduler หรือเริ่มโดยผู้ใช้งานเอง. จะมีการพร้อมท์ Report Distribution List ให้คุณ. Report Distribution List คือรายการของสพูลไฟล์ และผู้รับที่จะจัดส่งสพูลไฟล์ไปให้.

My Connections → server → Basic Operations → Jobs → คลิกขวาที่ งาน → Distribute Reports

ตารางเวลาที่พร้อมใช้งานสำหรับผู้รับอีเมล

- ตารางเวลาที่พร้อมใช้งานคือ ตารางเวลาซึ่งมีข้อมูลผู้รับที่จะรับ ข้อความการแจ้งเตือนพร้อมใช้งานอยู่แล้ว. คุณสามารถเลือก Always available, blank (ไม่พร้อมใช้งาน) หรือข้อพจนตารางเวลาซึ่งได้กำหนดไว้ก่อนหน้านี้ในหน้าต่างคุณสมบัติ **Advanced Job Scheduler – Schedules**.

ฟิลด์ **Availability schedule** อยู่บนหน้าต่าง **Recipient Properties – Email**. (My Connections → server → Work Management → Advanced Job Scheduler → Notifications → Recipients → คลิกขวาที่ชื่อผู้รับ → Properties)

Work Flow Manager

- Work Flow Manager คือเครื่องมือใหม่ที่อนุญาตให้คุณกำหนดหน่วยของงานที่สามารถประกอบด้วยชุดของขั้นตอนแบบอัตโนมัติและขั้นตอนแบบทำเอง. จากนั้นสามารถ จัดตารางเวลาหรือเริ่มต้นหน่วยงานด้วยตนเองได้. ด้วยจุดตรวจสอบการแจ้งเตือนหลายจุด, ผู้ใช้จึงสามารถได้รับแจ้งถ้าขั้นตอนที่เริ่มต้น, เสร็จสิ้น, ไม่ได้ รันตามเวลาที่ระบุ, หรือใช้เวลานานเกินไปขัดจำกัดเวลาการรัน. แต่ละขั้นตอนอาจมีงาน predecessor และงาน successor ได้. Predecessor jobs สำหรับขั้นตอนหนึ่งต้องเสร็จสมบูรณ์ก่อน ขั้นตอนจึงจะสามารถเสร็จสมบูรณ์โดยอัตโนมัติหรือด้วยตนเองได้. หลังจากขั้นตอนเสร็จสมบูรณ์แล้ว, successor jobs จะถูกเช็คให้รัน. ถือเป็นเรื่องปกติที่จะระบุ predecessor jobs ที่เหมือนกับ successor jobs ของขั้นตอนก่อนหน้านี้. การทำเช่นนี้ทำให้ขั้นตอนต้องรอ จนกว่างานจะเสร็จสมบูรณ์ ก่อนที่จะแจ้งถึงความสมบูรณ์ของขั้นตอน.

สิ่งที่เหมาะสมสำหรับการใช้ Advanced Job Scheduler Work Flow Manager คือ การประมวลผลค่าจ้าง. กระบวนการค่าจ้างประกอบด้วยขั้นตอนที่ต้องทำด้วยตนเองหลายอย่าง เช่น การอินพุตบัตรลงเวลา, การตรวจสอบรายงาน, และการพิมพ์ และการจ่ายเช็ค. ขั้นตอนแบบอัตโนมัติ สามารถจัดการกับไฟล์งานแบบแบ็ตช์, ประมวลผลการอินพุตบัตรลงเวลา, รันการอัปเดตค่าจ้าง, และสร้างรายงานและเช็ค.

My Connections → server → Work Management → Advanced Job Scheduler → Work Flow Manager

Object Resource Dependency ของระบบไฟล์รวม

- หน้าต่าง **Resource Dependencies** แสดงข้อมูลเกี่ยวกับ resource dependencies ของงานเฉพาะ, รวมถึงรายการของ dependencies, ความต้องการที่จำเป็นก่อนที่จะรันงานต่อไป, และเวลาที่จะรอก่อนการรีเซตงาน, และการอนุญาตให้คุณใส่เพิ่ม, ลบออก, หรือดูคุณสมบัติ ของ resource dependency. สิ่งใหม่ใน V5R4 คือคุณสามารถระบุ ว่า dependency object นี้เป็นระบบไฟล์รวมหรือไม่และระบุพาธ.

My Connections → server → Work Management → Advanced Job Scheduler → Scheduled Jobs → คลิกขวาที่งาน → Resource Dependencies → Create a new dependency type object

Page Selection สำหรับส่วนแบบสพลไฟล์การแจ้งเตือน

- Page Selection ช่วยให้คุณสามารถระบุข้อมูลการเลือกที่เป็นข้อความ และตำแหน่งของข้อมูลภายในหน้าแต่ละหน้าของสพลไฟล์. คุณสามารถระบุข้อความที่มีอยู่ที่ตำแหน่งเฉพาะบนแต่ละหน้า หรือที่ตำแหน่งใดๆ บน หน้า. และคุณยังสามารถแบ่งเซตย่อยสพลไฟล์ได้โดยการเลือกช่วงหน้า.

ฟังก์ชัน Page Selection สามารถพบได้ที่ My Connections → server → Work Management → Advanced Job Scheduler → Notification → Report Distribution List → คลิกขวาที่รายการ → Properties → คลิกสพลไฟล์ → คลิก Properties

การใส่เพิ่มอ็อพชันการไม่รีเซ็ตงานที่พักไว้

- ในปัจจุบันประสิทธิภาพการทำงานอาจได้รับผลกระทบ ถ้างานที่จัดตารางเวลาไว้ ถูกพักไว้เป็นครั้งคราว. ในแต่ละครั้งที่ดำเนินการถึงวันที่และเวลาที่จัดตารางเวลาไว้ สำหรับงานที่ถูกพัก, งานเซิร์ฟเวอร์ Advanced Job Scheduler จะกำหนดว่างาน ยังคงถูกพักอยู่หรือไม่, และถ้าใช่, ระบบจะคำนวณวันที่และเวลาที่ควรจะรัน งานที่พักไว้. สิ่งใหม่สำหรับ V5R4 คือ, คุณสามารถหยุดการคำนวณนี้โดยการตรวจให้แน่ใจว่าได้ยกเลิกการเลือกฟิลด์ **Reset held jobs**. เมื่อยกเลิกการเลือกฟิลด์ **Reset held jobs** แล้ว, เมื่อถึงวันที่และเวลาที่จัดตารางเวลาไว้สำหรับงานที่พัก, ฟิลด์วันที่และเวลาที่จัดตารางเวลาไว้จะถูกลบออก, และไม่มีการทริกเกอร์ activity การประมวลผลเพิ่มเติมบนงานที่พัก. จากนั้นเมื่อคุณ รีเซ็ตงาน, เซิร์ฟเวอร์จะคำนวณวันที่และเวลาที่จัดตารางเวลาไว้ที่งานควรจะรัน. การใช้ฟิลด์ **Reset held jobs** ประยุกต์ใช้กับ งานทั้งหมดที่กำหนดโดยใช้ Advanced Job Scheduler.

ฟิลด์ **Reset held jobs** อยู่บนหน้าต่าง **Advanced Job Scheduler Properties – General**. (My Connections → server → Work Management → คลิกขวาที่ **Advanced Job Scheduler** → Properties)

การติดตั้ง Advanced Job Scheduler:

ในครั้งแรกที่คุณเชื่อมต่อกับเซิร์ฟเวอร์ Management Central, iSeries Navigator จะถามว่าคุณต้องการติดตั้ง Advanced Job Scheduler หรือไม่. ถ้าคุณเลือกไม่ และต้องการติดตั้งในขณะนี้, คุณสามารถทำได้โดยใช้คุณลักษณะพิเศษ Install Plug-Ins ของ iSeries Navigator.

1. จากหน้าต่าง iSeries Navigator ของคุณ, ให้คลิก **File** จากเมนูบาร์.
2. คลิก **Install Options** → **Install Plug-Ins**.
3. คลิกระบบซอร์สซึ่งมีการติดตั้ง Advanced Job Scheduler และคลิก **OK**. ตรวจสอบกับผู้ดูแลระบบ ถ้าคุณไม่แน่ใจว่าจะใช้ระบบซอร์สอะไร.
4. ป้อน **iSeries User ID** และ **Password**, และคลิก **OK**.
5. คลิก **Advanced Job Scheduler** จากรายการเลือก Plug-in .
6. คลิก **Next** แล้วคลิก **Next** อีกครั้ง.
7. คลิก **Finish** เพื่อให้สมบูรณ์และออกจากหน้าต่าง.

ขณะนี้คุณได้ติดตั้ง Advanced Job Scheduler แล้ว.

การระบุตำแหน่งตารางเวลา:

ถ้าต้องการระบุตำแหน่งตารางเวลา, ให้ทำตามขั้นตอนต่อไปนี้:

1. ให้ขยาย **Management Central**.
2. คลิก **Scan Now** เมื่อเห็นข้อความว่า iSeries Navigator ได้ตรวจพบส่วนประกอบใหม่. คุณอาจเห็นข้อความนี้อีกครั้งเมื่อคุณเข้าใช้งาน ระบบจาก **My Connections** container.
3. ให้ขยาย **My Connections** → ของเซิร์ฟเวอร์ iSeries ที่มีการติดตั้งโปรแกรมไลเซนส์ Advanced Job Scheduler → **Work Management** → **Advanced Job Scheduler**.

หลังจากที่คุณทำงานเบื้องต้นเกี่ยวกับ Advanced Job Scheduler เสร็จสิ้นแล้ว, คุณก็พร้อมที่จะตั้งค่า Advanced Job Scheduler.

การตั้งค่า Advanced Job Scheduler:

หลังจากที่คุณได้ติดตั้ง Advanced Job Scheduler แล้ว, คุณต้อง ตั้งค่า. เมื่อคุณทำงานเบื้องต้นนี้เสร็จสิ้นแล้ว, คุณก็พร้อมที่จะเริ่มต้นงานการจัดตารางเวลา.

การกำหนดคุณสมบัติทั่วไป:

กำหนดคุณสมบัติทั่วไปที่ใช้โดย Advanced Job Scheduler. คุณสามารถระบุระยะเวลาการเก็บรักษา activity และ log entry สำหรับ Advanced Job Scheduler, รวมทั้งช่วงเวลาที่งานจะไม่ได้รับอนุญาตให้รัน.

คุณสามารถระบุวันทำงานที่จะให้งานทำการประมวลผล, และมีแอ็พพลิเคชันใดบ้างที่ต้องการสำหรับงานที่จัดตารางเวลาแต่ละงาน. ถ้าคุณได้ติดตั้งผลิตภัณฑ์ที่ได้ประกาศแจ้งไว้, คุณยังสามารถตั้งค่าคำสั่งที่จะใช้เพื่อส่งการแจ้งเตือนเมื่องานเสร็จสิ้นหรือล้มเหลว หรือคุณสามารถใช้คำสั่ง Send Distribution using Job Scheduler (SNDDSTJS) เพื่อแจ้งเตือนผู้รับ.

คุณสามารถ ระบุระยะเวลาการเก็บรักษาเร็กคอร์ด activity สำหรับงาน, รวมทั้งช่วงเวลาที่งานจะไม่ได้รับอนุญาตให้รัน. คุณสามารถระบุวันทำงานที่จะให้งานทำการประมวลผล, และมีแอ็พพลิเคชันใดบ้างที่ต้องการสำหรับงานที่จะส่งแต่ละงาน.

คุณสามารถมีผลิตภัณฑ์ที่ได้แจ้งไว้ล่วงหน้าติดตั้งอยู่ซึ่งจะทำให้คุณได้รับการเตือน (ข้อความ) เมื่องานจบลง. คุณสามารถกำหนดคำสั่งการแจ้งเตือนซึ่งจะส่งข้อความเตือนเมื่องานเสร็จสมบูรณ์หรือล้มเหลว หรือคุณสามารถใช้คำสั่ง Send Distribution using Job Scheduler (SNDDSTJS) เพื่อแจ้งเตือนผู้รับ.

ถ้าต้องการตั้งค่าคุณสมบัติทั่วไปสำหรับ Advanced Job Scheduler, ให้ทำตามขั้นตอนต่อไปนี้:

1. ขยาย **Work Management** จากหน้าต่าง iSeries Navigator ของคุณ.
2. คลิกขวา **Advanced Job Scheduler** และคลิก **Properties**.
3. ระบุ **Activity Retention**. activity retention เป็นระยะเวลาที่คุณต้องการให้เก็บรักษาเร็กคอร์ด activity สำหรับงาน. ค่าที่เป็นไปได้คือ 1 ถึง 999 วันหรือจำนวนครั้ง. คลิก **Days** เพื่อระบุจำนวนวัน ถ้าคุณต้องการเก็บรักษา activity สำหรับจำนวนวันที่แน่นอน, หรือคลิก **Occurrences per job** ถ้าคุณต้องการเก็บรักษา activity สำหรับจำนวนที่แน่นอนของการเกิดขึ้นต่องาน.
4. ให้ระบุ **Log retention**. log retention ระบุ, จำนวนวัน, ของระยะเวลาที่คุณต้องการเก็บรักษา Advanced Job Scheduler log entries.
5. คุณสามารถระบุ **Reserved period**. งานจะไม่รันในระหว่างช่วงเวลานี้.
6. ให้ระบุวันทำงานจากในรายการ. ถ้ามีการเลือกวัน, วันจะถูกกำหนดให้เป็นวันทำงานและสามารถใช้อ้างอิงเมื่อจัดตารางเวลางาน.

7. คลิก **Application required for scheduled job** เพื่อกำหนดว่า จำเป็นต้องใช้แอปพลิเคชันสำหรับแต่ละงานที่จัดตารางเวลาหรือไม่. แอปพลิเคชัน คืองานที่ถูกจัดกลุ่มเข้าด้วยกันสำหรับการประมวลผล. จะไม่สามารถเลือกได้ถ้างานที่มีอยู่ไม่ประกอบด้วยแอปพลิเคชัน. ถ้าคุณเลือกที่จะมี แอปพลิเคชันที่ต้องการสำหรับงาน, ให้ไปที่การทำงานกับแอปพลิเคชัน.
8. คลิก **Calendars** เพื่อตั้งค่าการจัดตารางเวลา, วันหยุด, และปีบัญชีที่จะใช้, ตั้งค่าปฏิทินวันหยุด, และตั้งค่าปีบัญชี.
9. คลิก **Base periodic frequency on start time** เพื่อกำหนดพื้นฐานของรันใหม่บนเวลาเริ่มต้นสำหรับงานที่จะถูกจัดตารางเวลาให้รันเป็นช่วงเวลา. ตัวอย่างเช่น, งานจะรันทุกๆ 30 นาที, เริ่มทำงานที่เวลา 8:00 am. (สำหรับงานที่รันตามรอบของนาฬิกา, ให้ระบุ 7:59 am เป็นเวลาสิ้นสุด งาน.) งานที่รันเป็นเวลาทั้งหมด 20 นาที. ด้วยการเลือกฟิลต์นี้, งาน จะรันที่เวลา 8:00 am, 8:30 am, 9:00 am, และต่อไป. ถ้าไม่ได้เลือกฟิลต์นี้, งานจะรันที่เวลา 8:00 am, 8:50 am, 9:40 am, 10:30 am, และต่อไป.
10. คลิก **Reset held jobs** เพื่อคำนวณใหม่ และแสดงวันที่และเวลาถัดไปที่งานที่พักไว้จะรัน.
11. ระบุ **Start time of day**. นี่เป็นเวลาของวันที่คุณพิจารณาว่าเป็นการเริ่มต้นวันใหม่. งานทั้งหมดที่ถูกระบุให้ใช้เวลานี้ของวันจะมีวันที่ของงานเปลี่ยนเป็นวันก่อนหน้าถ้าเวลาที่งานเริ่มต้นเป็นเวลาก่อนหน้าเวลาในฟิลต์ **Start time of day**.
12. ระบุ **Job monitor user**. ฟิลต์นี้ระบุชื่อของโปรไฟล์ผู้ใช้ที่ใช้เป็นเจ้าของของงานการมอนิเตอร์. งานทั้งหมดที่มีการระบุ **Current user** จะใช้โปรไฟล์ผู้ใช้ของงานการมอนิเตอร์. โปรไฟล์ผู้ใช้ดีฟอลต์ของงานการมอนิเตอร์คือ QIJS.
13. ในฟิลต์ **Notification command**, คุณสามารถ ระบุคำสั่งได้. ใช้คำสั่ง Send Distribution using Job Scheduler Notification (SNDDSTJS) ที่ให้มาพร้อมกับระบบ หรือคำสั่งที่ระบุโดยซอฟต์แวร์ การแจ้งเตือนของคุณ. คำสั่ง SNDDSTJS จะใช้ฟังก์ชันการแจ้งเตือนของ Advanced Job Scheduler. ผู้รับที่กำหนดสามารถได้รับข้อความ job schedule entry ที่จับแบบปกติและแบบไม่ปกติ.

การระบุค่าระดับการอนุญาต:

ระบุระดับการอนุญาตสำหรับงาน, ฟังก์ชันของผลิตภัณฑ์, และ แสดงการอนุญาตดีฟอลต์สำหรับงานใหม่.

คุณสามารถระบุระดับการอนุญาตสำหรับงาน, ฟังก์ชันของผลิตภัณฑ์, และแสดงการอนุญาตดีฟอลต์สำหรับงานใหม่ที่จะใช้กับแต่ละ Job Control/Application. การอนุญาตสำหรับงานช่วยให้คุณจะสามารถที่จะให้หรือปฏิเสธการเข้าถึงการปฏิบัติการต่อไปนี้: การส่ง, การจัดการ, การอนุญาต, การแสดงผล, การทำสำเนา, การอัปเดต, หรือการลบ. คุณยังสามารถให้หรือปฏิเสธการเข้าใช้งานแต่ละฟังก์ชันของผลิตภัณฑ์นี้ด้วย เช่น Work with Schedule Calendars, Send Reports, และ Add Job.

ค่าดีฟอลต์ระดับการอนุญาตถูกถ่ายโอนไปยังงานใหม่เมื่อมีการเพิ่มงานใหม่เข้าไป. ในแต่ละกรณี, ระบบจะถ่ายโอนการอนุญาตสำหรับงานใหม่ตามที่ระบุในแอปพลิเคชันภายใน job definition. ถ้าไม่มีการใช้แอปพลิเคชัน, ระบบจะถ่ายโอนค่าการอนุญาตงานใหม่เป็น *SYSTEM.

การระบุระดับการอนุญาตสำหรับฟังก์ชันของผลิตภัณฑ์:

ถ้าต้องการระบุระดับการอนุญาตสำหรับฟังก์ชันของผลิตภัณฑ์, ให้ทำตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จากหน้าต่าง iSeries Navigator.
2. คลิกขวา **Advanced Job Scheduler** และคลิก **Properties**.
3. คลิก **Permissions**.
4. เลือกฟังก์ชันและคลิก **Properties**.
5. บนหน้าต่าง Function Permissions Properties, ให้แก้ไขระดับ การอนุญาตตามความจำเป็น. คุณสามารถให้หรือปฏิเสธการเข้าใช้งานของผู้ใช้ปกติหรือผู้ใช้เฉพาะได้.

การระบุระดับการอนุญาตให้แก่งาน:

ถ้าต้องการระบุระดับการอนุญาตให้แก่งาน, ให้ทำตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จากหน้าต่าง iSeries Navigator.
2. คลิกขวา **Advanced Job Scheduler** และคลิก **Scheduled Jobs** เพื่อแสดงรายการงาน.
3. คลิกขวางานที่จัดตารางเวลา และคลิก **Permissions**.
4. บนหน้าต่าง **Permissions Properties**, ให้แก้ไขระดับการอนุญาต ตามความจำเป็น. คุณสามารถให้หรือปฏิเสธการเข้าถึงงานของผู้ใช้พบลิกหรือผู้ใช้เฉพาะได้. ยิ่งไปกว่านั้น, คุณสามารถระบุการส่ง, การจัดการ, การอนุญาต, การแสดงผล, การทำสำเนา, การอัปเดต, หรือการลบการอนุญาตต่างๆ ได้.

การระบุระดับการอนุญาตดีฟอลต์:

ถ้าต้องการระบุระดับการอนุญาตดีฟอลต์สำหรับงานใหม่ที่เกี่ยวข้องกับ Job Control/Application, ให้ทำตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จากหน้าต่าง iSeries Navigator.
2. คลิกขวา **Advanced Job Scheduler** และคลิก **Properties**.
3. คลิก **Job Controls/Applications**.
4. เลือกการควบคุมงานหรือแอ็พพลิเคชันจากรายการ และคลิก **New Job Permissions**.
5. บนหน้าต่าง **Function Permissions Properties**, ให้แก้ไขระดับ การอนุญาตตามความจำเป็น. คุณสามารถให้หรือปฏิเสธการเข้าถึงงานของผู้ใช้พบลิกหรือผู้ใช้เฉพาะได้. ยิ่งไปกว่านั้น, คุณสามารถระบุการส่ง, การจัดการ, การอนุญาต, การแสดงผล, การทำสำเนา, การอัปเดต, หรือการลบการอนุญาตต่างๆ ได้.

การตั้งค่าปฏิทินตารางเวลา:

การตั้งค่าปฏิทินของวันที่เลือกสำหรับการจัดตารางเวลาหรือกลุ่มของงาน. ปฏิทินนี้สามารถระบุวันที่ที่จะนำมาใช้สำหรับการจัดตารางเวลาหนึ่ง, หรือสามารถนำไปใช้ร่วมกับตารางเวลาอื่นๆ.

ปฏิทินตารางเวลา คือปฏิทินของวันที่คุณเลือกซึ่งอาจใช้สำหรับการจัดตารางเวลาหรือกลุ่มงาน. คุณอาจแสดงผลปฏิทินตารางเวลา, ใส่เพิ่มปฏิทินตารางเวลาใหม่, ใส่เพิ่มปฏิทินตารางเวลาใหม่โดยใช้ฐานเดิมจากปฏิทินตารางเวลาที่มีอยู่แล้ว, หรือลบปฏิทินที่มีอยู่แล้วออก, ถ้าปฏิทินไม่ได้ถูกใช้โดยงานที่จัดตารางเวลาในปัจจุบัน.

คุณสามารถเลือกปฏิทิน และแสดงผลคุณสมบัติของมันเพื่อทำการเปลี่ยนแปลงได้. เมื่อคุณเลือกปฏิทิน, รายละเอียดของปฏิทินจะถูกแสดงขึ้นภายใต้ **Details**.

ถ้าต้องการตั้งค่าปฏิทินการจัดตารางเวลา, ให้ทำตามขั้นตอนเหล่านี้:

1. เปิด **Work Management** จากหน้าต่าง iSeries Navigator .
2. คลิกขวา **Advanced Job Scheduler** และคลิก **Properties**.
3. บนหน้า General, คลิก **Calendars**.
4. บนหน้า **Scheduling Calendars**, คลิก **New**.
5. ระบุ **Name**.
6. ในฟิลด์ **Description** , ระบุข้อความที่ อธิบายปฏิทิน.

7. ให้เลือก **Reference calendar** ถ้าสามารถเลือกได้. นี่คือนโยบายที่ได้ตั้งค่าไว้ก่อนหน้านี้แล้ว, และ คุณสมบัติของมันจะถูกนำมาประยุกต์ใช้กับปฏิทินใหม่ราวกับว่าคุณได้ผสมรวมสองปฏิทินเข้าด้วยกัน. คุณจะไม่มีปฏิทินอ้างอิงถ้าครั้งนี้เป็นการเรียกใช้ Advanced Job Scheduler ครั้งแรกของคุณ.
8. เลือกวันที่ที่คุณต้องการเพิ่มลงในปฏิทินของคุณ. คุณต้องระบุว่าแต่ละวันที่คุณได้เลือกจะใช้สำหรับปีปัจจุบัน หรือทุกๆ ปีในฟิลด์ **Selected date** , ก่อนที่คุณจะสามารถใส่วันที่อื่นถัดไปเพิ่มในปฏิทิน. มิฉะนั้น, วันที่ใดๆ ที่คุณเลือกจะถูกยกเลิกเมื่อคุณคลิกวันที่อื่น.
9. ให้ระบุ ถ้าคุณต้องการวันแน่นอนของสัปดาห์ เพิ่มสอดแทรกบนปฏิทิน.

การตั้งค่าปฏิทินวันหยุด:

การตั้งค่าปฏิทินสำหรับวันที่คุณไม่ต้องการให้มีการประมวลผลตามตารางเวลาทำงาน. สามารถระบุวันที่เลือกสำหรับวันหยุดเว้นแต่ละวัน, หรือสามารถข้ามการประมวลผลสำหรับวันนั้น.

ปฏิทินวันหยุด เป็นปฏิทิน exception สำหรับวันในปฏิทินที่คุณไม่ต้องการประมวลผลงาน Advanced Job Scheduler. สามารถระบุวันที่เลือกสำหรับวันหยุดเว้นแต่ละวัน ซึ่งคุณระบุในปฏิทิน วันหยุด. คุณสามารถแสดงผลปฏิทินวันหยุด, ใส่เพิ่มปฏิทินวันหยุดใหม่, ใส่เพิ่มปฏิทินวันหยุดใหม่โดยใช้ฐานเดิมจากปฏิทินที่มีอยู่แล้ว, หรือลบปฏิทินที่มีอยู่แล้วออก, ถ้าปฏิทินไม่ได้ถูกใช้โดยงานที่จัดตารางเวลาในปัจจุบัน.

เราสามารถใส่ตารางเวลาที่กำหนดไว้ล่วงหน้าในปฏิทินวันหยุด. คุณสามารถสร้างตารางเวลา THIRDFRI ที่มีความถี่ในทุกๆ วันศุกร์ที่สามของแต่ละเดือน. เมื่อคุณใช้ THIRDFRI ใน ปฏิทินวันหยุด, นั้นหมายความว่ากำหนดให้งานทั้งหมดที่ใช้ปฏิทินวันหยุดนี้ไม่รันในวันศุกร์ที่สามของทุกเดือน. เราสามารถใช้ตารางเวลามากกว่าหนึ่งตารางในปฏิทินวันหยุดหนึ่ง. วันที่ที่ถูกสร้างขึ้นมาจากตารางเวลาจะถูกแสดงอยู่ในปฏิทินโดยมีกรอบสีดำล้อมรอบ.

คุณสามารถเลือกปฏิทิน และแสดงผลคุณสมบัติของมันเพื่อทำการเปลี่ยนแปลงได้. เมื่อคุณเลือกปฏิทิน, รายละเอียดของปฏิทินจะถูกแสดงขึ้นภายใต้ Details.

การตั้งค่าปฏิทินวันหยุด:

ถ้าต้องการตั้งค่าปฏิทินวันหยุด, ให้ทำตามขั้นตอนต่อไปนี้:

1. ขยาย **Work Management** จากหน้าต่าง iSeries Navigator.
2. คลิกขวาที่ **Advanced Job Scheduler** และเลือก **Properties**.
3. บนหน้า **General**, คลิก **Calendars**.
4. คลิกแท็บ **Holiday Calendars** .
5. คลิก **New** และป้อนชื่อปฏิทิน.
6. ในฟิลด์ **Description** , ระบุข้อความเพื่ออธิบายปฏิทิน.
7. เลือก **Reference calendar** ถ้าสามารถเลือกได้. นี่คือนโยบายที่ได้ตั้งค่าไว้ก่อนหน้านี้แล้ว, และ คุณสมบัติของมันจะถูกนำมาประยุกต์ใช้กับปฏิทินใหม่ราวกับว่าคุณได้ผสมรวมสองปฏิทินเข้าด้วยกัน. คุณจะไม่มีปฏิทินอ้างอิงถ้าครั้งนี้เป็นการเรียกใช้ Advanced Job Scheduler ครั้งแรกของคุณ.
8. เลือกวันที่ที่คุณต้องการเพิ่มลงในปฏิทินของคุณ. คุณต้องระบุว่าแต่ละวันที่คุณเลือกจะใช้สำหรับปีปัจจุบัน หรือทุกๆ ปีในฟิลด์ **Selected date** , ก่อนที่คุณจะสามารถใส่วันที่อื่นถัดไปเพิ่มในปฏิทิน. มิฉะนั้น, วันที่ใดๆ ที่คุณเลือกจะถูกยกเลิกเมื่อคุณคลิกวันที่อื่น.

9. ให้เลือกวันทางเลือกสำหรับวันงาน. คุณสามารถเลือกวันทำงานก่อนหน้า, วันทำงานถัดไป, วันที่เฉพาะ หรือไม่เลือกเลย. ถ้าต้องการเลือกวันที่เฉพาะ, ให้คลิก **Specific alternate date**, และพิมพ์วันที่ลงไป.
10. เลือกเจาะจงวันในสัปดาห์ที่จะแทรกเพิ่มเติมลงไปใน ปฏิทิน.

การใส่เพิ่มตารางเวลาลงในปฏิทินวันหยุด:

ถ้าต้องการใส่เพิ่มปฏิทินวันหยุดลงในงานที่จัดตารางเวลา, ให้ทำตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จากหน้าต่าง iSeries Navigator.
2. คลิกขวา **Advanced Job Scheduler** และคลิก **Properties**.
3. บนหน้า General, คลิก **Calendars**.
4. บนหน้า **Holiday calendar**, เลือกปฏิทินวันหยุดและคลิก **Properties**.
5. จากมุมมองล่างของแท็บ, คลิก **Schedules**.
6. เลือกตารางเวลาที่เหมาะสม และคลิก **Add**.
7. เพื่อเปลี่ยน **Alternate day**, ให้คลิกขวาที่ ตารางเวลาจากรายการ **Selected Schedules** และคลิก **Alternate Day** ที่ถูกต้อง.

การตั้งค่าปฏิทินปีบัญชี:

การตั้งค่าปฏิทินปีบัญชีของวันที่เลือกสำหรับการจัดตารางเวลาหรือกลุ่มงาน. ใช้ปฏิทิน ชนิดนี้ ถ้าคุณต้องการแบ่งปีบัญชี เป็น รอบระยะเวลาแทนที่จะแบ่งเป็นเดือน.

ปฏิทินปีบัญชี คือปฏิทินของวันที่คุณเลือก ซึ่งคุณสามารถใช้สำหรับการจัดตารางเวลาหรือกลุ่มงาน. ใช้ปฏิทินปีบัญชีเพื่อกำหนดปฏิทินปีบัญชีเฉพาะสำหรับธุรกิจของคุณ. คุณสามารถระบุวันที่เริ่มต้นและวันที่สิ้นสุดสำหรับแต่ละรอบระยะเวลาในปีบัญชีได้.

ถ้าต้องการตั้งค่า ปฏิทินปีบัญชี, ให้ทำตามขั้นตอนเหล่านี้:

1. เปิด **Work Management** จากหน้าต่าง iSeries Navigator .
2. คลิกขวา **Advanced Job Scheduler** และคลิก **Properties**.
3. บนหน้าต่าง General, คลิก **Calendars**.
4. บนหน้า **Fiscal Calendars**, คลิก **New**.
5. ระบุ **Name**.
6. ในฟิลด์ **Description** , ให้พิมพ์ข้อความเพื่อ อธิบายปฏิทิน.
7. คลิก **New** บนหน้าต่าง **Fiscal Calendar Properties** เพื่อสร้าง entry ใหม่.
8. เลือกช่วงระยะเวลา และระบุวันที่เริ่มต้นและวันที่สิ้นสุด. คุณสามารถระบุช่วงระยะเวลาได้สูงถึง 13 ช่วง.
9. คลิก **OK** เพื่อบันทึก fiscal calendar entry.
10. ทำซ้ำขั้นตอนที่ 7 ถึง 9 ตามความจำเป็น.

การระบุเมลเซิร์ฟเวอร์เพื่อใช้สำหรับการแจ้งเตือน:

ตั้งค่าเมลเซิร์ฟเวอร์ที่ใช้สำหรับข้อความแจ้งเตือนอีเมล. ต้องใช้เมล เซิร์ฟเวอร์ ถ้าคุณต้องการส่งการแจ้งเตือนทางอีเมล.

ถ้าต้องการตั้งค่าคุณสมบัติการแจ้งเตือน, ให้ทำตามขั้นตอนเหล่านี้:

1. เปิด **Work Management** จากหน้าต่าง iSeries Navigator .
2. ขยาย **Advanced Job Scheduler**.
3. คลิกขวา **Notification** และคลิก **Properties**.
4. ระบุจำนวนวันที่ต้องการจัดเก็บข้อความ. ระบุตัวเลขในฟิลด์ **Message retention**.
5. ระบุชื่อ **Outgoing mail server (SMTP)**. ตัวอย่าง, SMTP.yourserver.com.
6. ระบุ **Port**. หมายเลขพอร์ตดีฟอลต์คือ 25.
7. ระบุอีเมลแอดเดรสในฟิลด์ **Reply address**. ข้อความที่ตอบกลับทั้งหมดจะถูกส่งไปที่แอดเดรสนี้.
8. เลือก **Yes** หรือ **No** ในฟิลด์ **Log send activity**. activity การส่งถูกใช้เพื่อการพิจารณาปัญหา.
9. ระบุ **Number of banner pages** ที่ใช้ได้. สิ่งนี้จะถูกใช้ใน Report Distribution.
10. คลิก **OK** เพื่อบันทึกคุณสมบัติการแจ้งเตือน.

การตั้งค่าสถานะแวดล้อมการจัดตารางเวลาหลายสถานะ:

คุณสามารถตั้งค่าสถานะแวดล้อมการจัดตารางเวลาบนระบบเดียวกันได้. โดยการทำเช่นนี้, โลบารี่ข้อมูลต้นฉบับสามารถทำหน้าที่เป็นโลบารี่ข้อมูลที่แอ็คทีฟ และโลบารี่ข้อมูลที่ทำสำเนาสามารถใช้สำหรับการทดสอบ. ดังนั้นคุณจึงมีสถานะแวดล้อมการจัดตารางเวลา สองสถานะ, สถานะหนึ่งสำหรับการทดสอบและอีกสถานะหนึ่งคือสถานะจริง. นอกจากนี้, โลบารี่ข้อมูลที่ใช้ทดสอบสามารถทำหน้าที่เป็นตัวสำรองข้อมูลได้ ถ้าระบบดั้งเดิมล้มเหลว. คุณลักษณะนี้ช่วยให้คุณมีความปลอดภัยมากขึ้นถ้าคุณสร้างข้อผิดพลาดในโลบารี่ข้อมูลต้นฉบับ เพราะว่าคุณมีสำเนาสำรองของโลบารี่ข้อมูล.

มีหลายเหตุผลว่าเพราะเหตุใดคุณอาจต้องการตั้งค่าสถานะแวดล้อม การจัดตารางเวลาหลายสถานะ. คุณอาจต้องการมีผลิตภัณฑ์เวอร์ชันการผลิตและ เวอร์ชันการทดสอบที่รันอยู่พร้อมกัน. สถานะแวดล้อมแบบนี้ช่วยให้คุณทดสอบตารางเวลางานต่างๆ ก่อนที่จะใช้ตารางจริงในโลบารี่ข้อมูลบนระบบการผลิต. หรือคุณอาจมีระบบหนึ่งที่เป็นตัวสำรองสำหรับระบบหนึ่งหรือหลายระบบซึ่งคุณสามารถใช้ผลิตภัณฑ์พวก data-mirroring เพื่อจำลองโลบารี่ข้อมูล Advanced Job Scheduler (QUSRJJS) จากระบบต้นทางไปยังโลบารี่ที่มีชื่อแตกต่างกัน. ในกรณีนี้, โลบารี่ข้อมูลยังคงแอ็คทีฟอยู่จนกว่าจะมีปัญหาเกิดขึ้นกับระบบต้นทาง.

สถานะแวดล้อมการจัดตารางเวลาคือภาพจำลองของโลบารี่ QUSRJJS เพียงแต่มีข้อมูลที่แตกต่างกัน. ตัวอย่างเช่น คุณสามารถมีโลบารี่ข้อมูลอื่นที่ชื่อว่า QUSRJSTST โดยมีอ็อบเจกต์ทั้งหมดเหมือนกับที่มีใน QUSRJJS. แต่ละโลบารี่ถือว่าเป็นโลบารี่ข้อมูลเช่นกัน.

ถ้าต้องการ ตั้งค่าสถานะแวดล้อมการจัดตารางเวลาหลายสถานะ, ให้ทำตามขั้นตอนเหล่านี้:

1. จัดหาโลบารี่ข้อมูลจากระบบ

เพื่อสร้างโลบารี่ข้อมูล, คุณต้องจัดหาโลบารี่ข้อมูลหนึ่งโลบารี่จากระบบ. คุณสามารถจัดหาโลบารี่ข้อมูลจากระบบได้สามวิธีต่อไปนี้:

- บันทึกโลบารี่ข้อมูลจากระบบและจัดเก็บโลบารี่บนระบบ การผลิต.
- ทำสำเนาโลบารี่ข้อมูลบนระบบปัจจุบันโดยใช้คำสั่ง Copy Library (CPYLIB).
- มิเรอร์โลบารี่ข้อมูลบนระบบทดสอบ. ระบบเหล่านี้ควรจะรันที่ระดับรีลีส์เวอร์ชันเดียวกัน.

หมายเหตุ: โลบารี่ข้อมูลที่ทำสำเนา, คัดลอก, หรือมิเรอร์จะใช้ชื่อที่แตกต่างจาก ระบบดั้งเดิม.

2. กำหนดโลบารี่ข้อมูลให้ผู้ใช้

หลังจากที่คุณจัดทาลอเบรารีข้อมูลสำหรับทดสอบแล้ว, ให้ใส่เพิ่มลอเบรารีข้อมูลไปยังคุณสมบัติของ Advanced Job Scheduler และกำหนดผู้ใช้งานให้กับลอเบรารีข้อมูล. ดังนั้น, เมื่อมีผู้ใช้งานคนหนึ่งใช้งาน Advanced Job Scheduler, การเปลี่ยนแปลงที่เกิดขึ้นจากผู้ใช้งานจะถูกเก็บอยู่ในลอเบรารีข้อมูลที่กำหนดให้กับผู้ใช้งานคนนั้น.

3. ทำสำเนาจากลอเบรารีข้อมูลทดสอบไปที่ลอเบรารีข้อมูลจริง (เป็นอ็อพชั่น)

ถ้าคุณใช้ลอเบรารีข้อมูลเพื่อการทดสอบ, คุณอาจต้องการทำสำเนาจากลอเบรารีข้อมูลทดสอบไปยังลอเบรารีข้อมูลจริงที่จะใช้. คุณต้องทำอย่างนี้ เฉพาะถ้าคุณได้คืนสภาพหรือทำสำเนาลอเบรารีข้อมูลในขั้นตอนที่ 1 และมีงานที่คุณต้องการย้ายไปไว้ที่ลอเบรารีข้อมูลจริงที่จะใช้. คุณไม่จำเป็นต้องทำเช่นนี้ ถ้าคุณมีเรอร์ลอเบรารีข้อมูลจากระบบจริงไปยังระบบทดสอบ. เพื่อทำสำเนาจากลอเบรารีข้อมูลของระบบหนึ่งไปยังระบบอื่น, ให้ใช้คำสั่ง Copy Job using Job Scheduler (CPYJOBJS). สำหรับข้อมูลเพิ่มเติมเกี่ยวกับพารามิเตอร์เฉพาะสำหรับคำสั่งนี้, ให้ดูคำอธิบาย ทางออนไลน์.

การกำหนดลอเบรารีข้อมูลให้ผู้ใช้:

ระบุว่าลอเบรารีข้อมูลอะไรที่เชื่อมโยงกับผู้ใช้แต่ละคน. ลอเบรารีข้อมูลหนึ่งจะประกอบด้วยอ็อบเจกต์ทั้งหมดที่พบอยู่ในลอเบรารี QUSRIJS. คุณสามารถมีลอเบรารีข้อมูลได้ไม่จำกัดจำนวน.

ลอเบรารีข้อมูลจะจัดเก็บการเปลี่ยนแปลงใดๆ ที่ผู้ใช้ทำโดยใช้ Advanced Job Scheduler. ลอเบรารีข้อมูลประกอบด้วยอ็อบเจกต์ทั้งหมดที่พบอยู่ในลอเบรารี QUSRIJS. คุณสามารถมีลอเบรารีข้อมูลได้ไม่จำกัดจำนวน.

ถ้าต้องการกำหนดลอเบรารีข้อมูลให้แก่ผู้ใช้, ให้ทำตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จากหน้าต่าง iSeries Navigator.
2. คลิกขวา **Advanced Job Scheduler** และคลิก **Properties**.
3. บนหน้าต่าง Data Libraries, คลิก **Add** เพื่อ ระบุลอเบรารีข้อมูล. ลอเบรารีข้อมูลที่อยู่ในรายการเป็นลอเบรารีที่พร้อมใช้งานสำหรับผู้ใช้ทั้งหมดบนระบบ.
4. บนหน้าต่าง Users, คลิก **Add** เพื่อใส่เพิ่มผู้ใช้ใหม่.
5. ระบุชื่อ.
6. เลือกลอเบรารีข้อมูล.
7. คลิก **OK** เพื่อใส่เพิ่มผู้ใช้.
8. คลิก **Properties** เพื่อเปลี่ยน ลอเบรารีข้อมูลที่ได้กำหนดให้แก่ผู้ใช้.

ด้วยการใช้ลอเบรารีข้อมูล, คุณสามารถตั้งค่าสภาวะแวดล้อมการจัดตารางเวลาได้หลายสภาวะ.

การบริหาร Advanced Job Scheduler:

ข้อมูลต่อไปนี้จะช่วยคุณในการบริหาร Advanced Job Scheduler. อันดับแรกคุณต้องจัดตารางเวลายานโดยใช้ Advanced Job Scheduler. จากนั้น, ใช้ภารกิจส่วนที่เหลือในการบริหารงาน.

การสร้างและจัดตารางเวลายาน:

จัดตารางเวลายานและระบุคำสั่งที่เกี่ยวข้องกับงาน. คุณยังสามารถระบุคำสั่งการเริ่มต้นและสิ้นสุดเพื่อรันเวอร์ชันพิเศษของงานที่จัดตารางเวลาได้ด้วย.

ถ้าต้องการสร้างและจัดตารางเวลายานที่จัดตารางเวลาใหม่, ให้ทำตามขั้นตอนเหล่านี้:

1. เปิด **Work Management** จากหน้าต่าง iSeries Navigator .

2. คลิกขวาที่ **Advanced Job Scheduler**.
3. คลิกขวา **Scheduled Jobs** และคลิก **New Scheduled Job**.

การสร้างและจัดตารางเวลากลุ่มงาน:

ตั้งค่าและจัดตารางเวลาลำดับของงานที่รันอย่างต่อเนื่องตามลำดับที่ระบุ. งานที่อยู่ภายในกลุ่มงานหนึ่งจำเป็นต้องเสร็จสิ้นลงก่อนที่จะมีการส่งงานถัดไปเข้ามาประมวลผล.

กลุ่มงานคือ งานต่างๆ ที่ถูกจัดกลุ่มเข้าด้วยกันเพื่อรันอย่างต่อเนื่องตามลำดับที่ระบุ. การจบงานลงอย่างปกติโดยสมบูรณ์เป็นสิ่งจำเป็นสำหรับงานแต่ละงานในกลุ่ม ก่อนที่งานถัดไปในกลุ่มจะถูกส่งเข้ามาประมวลผล. ถ้างานใดๆ ในกลุ่มไม่จบลงแบบสมบูรณ์ตามปกติ, การประมวลผลจะหยุดลงสำหรับกลุ่มนั้น.

ถ้าต้องการสร้างและจัดตารางเวลากลุ่มงานใหม่, ให้ทำตามขั้นตอนเหล่านี้:

1. เปิด **Work Management** จากหน้าต่าง iSeries Navigator .
2. คลิก **Advanced Job Scheduler**.
3. คลิกขวา **Job Groups** และคลิก **New Job Group**.

อ้างอิงคำอธิบายทางออนไลน์สำหรับข้อมูลเพิ่มเติม เมื่อคุณกรอกรายละเอียด สำหรับกลุ่มงานใหม่.

ตารางเวลางานที่ได้ถูกกำหนดไว้ก่อน:

สร้างตารางเวลางานที่ประกอบด้วยข้อมูลที่เป็นต่อการจัดตารางเวลางาน หรือการคำนวณวันที่ยกเว้นภายในปฏิทินวันหยุด.

คุณสามารถสร้างตารางเวลาที่มีข้อมูลที่จำเป็นสำหรับการจัดตารางเวลางานหรือการคำนวณวันที่ยกเว้นในปฏิทินวันหยุด.

ตัวอย่างเช่น, คุณสามารถสร้างตารางเวลา ENDOFWEEK ซึ่งมีข้อมูลของวันในสัปดาห์ที่จะทำงาน, พร้อมด้วยข้อมูลของปฏิทินเพิ่มเติม. ตารางเวลา ENDOFWEEK สามารถใช้ได้โดยงานทั้งหมดที่สอดคล้องกับความถี่ของการจัดตารางเวลานั้น. คุณสามารถเข้าใช้งาน คุณลักษณะพิเศษนี้ผ่านทาง iSeries Navigator เท่านั้น.

คุณสามารถใช้ตารางเวลาที่ได้ถูกกำหนดไว้ก่อนอันเดียวกันกับที่ใช้ในงานเกี่ยวกับปฏิทินวันหยุดของคุณ. คุณสามารถสร้างตารางเวลา THIRDFRI ที่มีความถี่ในทุกๆ วันศุกร์ที่สามของแต่ละเดือน. เมื่อคุณใช้ THIRDFRI ใน ปฏิทินวันหยุด, นั้นหมายความว่ากำหนดให้งานทั้งหมดที่ใช้ปฏิทินวันหยุดนี้ไม่รันในวันศุกร์ที่สามของแต่ละเดือน. คุณสามารถใช้หนึ่งหรือหลายตารางเวลาในหนึ่งปฏิทินวันหยุด. วันที่ที่ถูกสร้างขึ้นมาจากตารางเวลาจะถูกแสดงอยู่ในปฏิทินโดยมีกรอบสีดำล้อมรอบ.

การตั้งค่าตารางเวลาที่ได้ถูกกำหนดไว้ก่อน:

ถ้าต้องการตั้งค่าตารางเวลาที่ได้ถูกกำหนดไว้ก่อน, ให้ทำตามขั้นตอนเหล่านี้.

1. ขยาย **Work Management** จากหน้าต่าง iSeries Navigator.
2. คลิกขวา **Advanced Job Scheduler** และคลิก **Properties**.
3. คลิกแท็บ **Schedules**.
4. คลิก **New** และพิมพ์ชื่อสำหรับตารางเวลา.
5. พิมพ์คำอธิบายสำหรับตารางเวลา.
6. เลือกความถี่และวันที่ที่คุณต้องการรวมในตารางเวลาของคุณ, รวมไปถึงปฏิทินเพิ่มเติมใดๆ .

อ้างอิงคำอธิบายทางออนไลน์สำหรับข้อมูลเพิ่มเติมในขณะที่คุณกรอรายละเอียดสำหรับตารางเวลาใหม่.

การใส่เพิ่มตารางเวลาที่งานที่จัดตารางเวลา:

ถ้าต้องการใส่เพิ่มตารางเวลาที่งานที่จัดตารางเวลา, ให้ทำตามขั้นตอนเหล่านี้:

1. เปิด **Work Management** จากหน้าต่าง iSeries Navigator .
2. คลิกขวาที่ **Advanced Job Scheduler**.
3. คลิก **Scheduled Jobs** เพื่อแสดงรายการของงาน.
4. คลิกขวางานที่จัดตารางเวลาและคลิก **Properties**.
5. คลิกแท็บ **Schedule**.
6. จากมุมมองของแท็บ, เลือกอ็อปชัน **Schedule** ที่เหมาะสม.

การใส่เพิ่มตารางเวลาลงในปฏิทินวันหยุด:

ปฏิทินวันหยุดเป็นปฏิทินยกเว้นสำหรับวันที่คุณไม่ต้องการ ประมวลผลงาน Advanced Job Scheduler. สามารถระบุวันทางเลือกสำหรับวันยกเว้นแต่ละวัน ซึ่งคุณระบุในปฏิทิน วันหยุด.

ถ้าต้องการใส่เพิ่มตารางเวลาในปฏิทินวันหยุด, ให้ทำตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จากหน้าต่าง iSeries Navigator.
2. คลิกขวา **Advanced Job Scheduler** และคลิก **Properties**.
3. บนหน้า **General** , คลิก **Holiday Calendars**.
4. บนหน้า **Holiday Calendars**, เลือกปฏิทินวันหยุดและคลิก **Properties**.
5. จากมุมมองซ้ายล่างของแท็บ, คลิก **Schedules**.
6. เลือกตารางเวลาที่เหมาะสม และคลิก **Add**.
7. เพื่อเปลี่ยน **Alternate day**, ให้คลิกขวาที่ ตารางเวลาจากรายการ **Selected Schedules** และคลิก **Alternate Day** ที่ถูกต้อง.

อ้างอิงคำอธิบายทางออนไลน์สำหรับข้อมูลเพิ่มเติม.

การสร้างงานที่จัดตารางเวลาชั่วคราว:

มีหลายครั้งที่อาจจะจำเป็นต้องรันงานที่ถูกจัดตารางเวลาในขณะนี้หรือในอนาคตเพิ่มเติมจากตารางเวลาปกติ. ใช้คำสั่ง **Submit Job using Job Scheduler (SBMJOBJS)** , อ็อปชัน 7 จากจอแสดงผล **Work with Jobs** , หรืออ็อปชัน **Run** จาก iSeries Navigator. และอาจจำเป็นต้องประมวลผลเพียง บางส่วนของคำสั่งที่อยู่ในรายการคำสั่ง เมื่อตั้งค่า การรันพิเศษนี้.

คำสั่ง **SBMJOBJS** ช่วยให้คุณสามารถระบุลำดับของ คำสั่งการเริ่มต้นและการสิ้นสุดได้. ตัวอย่างเช่น, **JOBA** มี 5 คำสั่ง, ลำดับที่ 10 ถึง 50. คุณสามารถระบุในคำสั่ง **SBMJOBJS** เพื่อให้ เริ่มต้นที่ลำดับ 20 และสิ้นสุดที่ลำดับ 40. การระบุเช่นนี้จะทำให้ข้ามลำดับ 10 และ 50.

iSeries Navigator อนุญาตให้คุณเลือกคำสั่งเริ่มต้นและคำสั่งสิ้นสุดจากรายการ คำสั่ง.

ถ้าต้องการรันเวอร์ชันพิเศษของงานที่จัดตารางเวลาไว้ด้วย iSeries Navigator, ให้ทำตามขั้นตอนเหล่านี้:

1. เปิด **Work Management** จากหน้าต่าง iSeries Navigator .

2. คลิกขวาที่ **Advanced Job Scheduler**.
3. คลิก **Scheduled Jobs** เพื่อแสดงรายการของงาน.
4. คลิกขวาบนงานที่จัดตารางเวลา และคลิก **Run**.
5. ระบุว่าจะรันงานในขณะนี้หรือในอนาคต.
6. เลือกคำสั่งเริ่มต้นและคำสั่งสิ้นสุด.

อ้างอิงคำอธิบายทางออนไลน์สำหรับข้อมูลเพิ่มเติม เมื่อคุณกรอกรายละเอียด สำหรับงานใหม่.

การจัดตารางเวลา job dependencies:

ตั้งค่างานหรือกลุ่มของงานที่ dependent (ขึ้น) ต่อกัน. คุณสามารถเลือกชนิดของ dependency ซึ่งจะส่งผลไปถึงวิธีการที่งานจะถูกประมวลผลในสถานะแวดล้อมของคุณ.

Advanced Job Scheduler ช่วยให้คุณสามารถตั้งค่า dependencies ที่จะส่งผลกระทบต่อวิธีที่งานจะถูกประมวลผลในสถานะแวดล้อมของคุณ. Dependencies เป็นตัวกำหนดว่างานหรือกลุ่มของงานสามารถรันได้เมื่อไร. คุณสามารถเลือกที่จะให้มี dependencies ทั้งหมดก่อนที่งานจะสามารถรันได้, หรือคุณสามารถให้อย่างน้อยหนึ่ง dependency ก่อนที่งานจะรันได้. Dependencies รวมถึงต่อไปนี้:

- **Job dependencies**

Job dependencies อ้างอิงถึงความสัมพันธ์ของ predecessor และ successor สำหรับงาน. งานชนิด Predecessor คืองานที่ต้องรันก่อนที่งานชนิด successor จะรัน. งานชนิด successor คืองานที่รันหลังงานชนิด predecessor ได้ประมวลผลเรียบร้อยแล้ว. สามารถมีงานชนิด successor ได้หลายงานสำหรับงานชนิด predecessor เพียงงานเดียว และมีงานชนิด predecessor หลายงานสำหรับงานชนิด successor งานเดียว. นอกจากนี้, คุณสามารถระบุให้ข้าม dependent job ถ้า predecessors และ successors ของงานรันบนวันซึ่ง dependent job ไม่ได้ถูกจัดตารางเวลาให้รัน.

- **Active dependencies**

Active dependencies คือ รายชื่อของงานที่ไม่สามารถแอ็คทีฟเมื่องานที่เลือกถูกส่ง. ถ้างานใดๆ แอ็คทีฟอยู่, Advanced Job Scheduler จะไม่อนุญาตให้งานที่ระบุ. งานที่ถูกเลือกจะถูกหน่วง จนกว่างานทั้งหมดในรายชื่อจะไม่แอ็คทีฟ.

- **Resource dependencies**

Resource dependencies มีพื้นฐานอยู่บนสองสามสิ่ง. แต่ละชนิดที่ตามมาจะอธิบายถึงพื้นที่ที่ถูกตรวจสอบ. ต่อไปนี้เป็นชนิดของ resource dependencies:

ไฟล์ งานขึ้นอยู่กับการมีอยู่หรือการไม่มีอยู่ของไฟล์ และขึ้นอยู่กับการมีระดับการจัดสรรตรงกับระดับการจัดสรรที่ระบุในการประมวลผลหรือไม่. นอกจากนี้ยังสามารถตรวจสอบด้วยว่า มีการแสดงเรีกคอร์ดก่อนที่งานจะถูกประมวลผลหรือไม่. ตัวอย่างเช่น, สามารถตั้งค่า JOBA เพื่อให้รันเฉพาะเมื่อมีไฟล์ ABC อยู่เท่านั้น, และสามารถจัดสรรไฟล์ เฉพาะถ้ามีเรีกคอร์ดอยู่ในไฟล์.

อ็อบเจกต์

งานขึ้นอยู่กับการมีอยู่หรือการไม่มีอยู่ของอ็อบเจกต์ชนิด QSYS และขึ้นอยู่กับการมีระดับการจัดสรรตรงกับระดับการจัดสรรที่ระบุในการประมวลผลหรือไม่. ตัวอย่างเช่น, เราสามารถตั้งค่า JOBA เพื่อให้รันได้เฉพาะเมื่อมีพื้นที่ข้อมูล XYZ อยู่เท่านั้น. นอกจากนี้ งานยังอาจขึ้นอยู่กับการมีอยู่หรือการไม่มีอยู่ของอ็อบเจกต์ที่พบใน ระบบไฟล์รวม. ถ้างานขึ้นอยู่กับการมีอยู่ของอ็อบเจกต์ใดๆ ในพาร, ให้จบพารระบบไฟล์รวมด้วย forward slash '/'.

Hardware configuration

งานขึ้นอยู่กับการมีอยู่หรือการไม่มีอยู่ของ hardware configuration และสถานะของมันที่จะประมวลผล. ตัวอย่างเช่น, สามารถตั้งค่า JOBA เพื่อให้ รันเฉพาะเมื่อมีอุปกรณ์ TAP01 อยู่และมีสถานะเป็น Available.

ไฟล์เน็ตเวิร์ก

งานขึ้นอยู่กับสถานะของไฟล์เน็ตเวิร์กที่จะประมวลผล.

ระบบย่อย

งานขึ้นอยู่กับสถานะของระบบย่อยที่จะประมวลผล.

ถ้าต้องการทำงานกับ job dependencies, ให้ทำตามขั้นตอนเหล่านี้:

1. เปิด **Work Management** จากหน้าต่าง iSeries Navigator .
2. ขยาย **Advanced Job Scheduler**.
3. คลิก **Scheduled Jobs**.
4. คลิกขวา **Job Name** ซึ่งมี dependencies ที่คุณต้องการทำงานด้วย.
5. ให้เลือกอย่างใดอย่างหนึ่งต่อไปนี้: **Job Dependencies, Active Dependencies หรือ Resource Dependencies**. ให้อ้างอิงคำอธิบายทางออนไลน์สำหรับข้อมูลเพิ่มเติม.

Work Flow Manager:

สำหรับ V5R4, Work Flow Manager อนุญาตให้คุณกำหนด หน่วยของงานที่ประกอบด้วยขั้นตอนแบบอัตโนมัติและขั้นตอนแบบทำเอง. จากนั้นสามารถจัดตาราง เวลาและรันหน่วยของงานเหล่านี้แบบโต้ตอบได้. Work Flow Manager อยู่ใน Advanced Job Scheduler container ในอินเตอร์เฟซ iSeries Navigator.

แต่ละขั้นตอนภายในเวิร์กโฟลว์สามารถมี predecessor Advanced Job Scheduler jobs หนึ่งหรือหลายงาน และ successor Advanced Job Scheduler jobs หนึ่งหรือหลายงาน. เมื่อเวิร์กโฟลว์เริ่มขึ้น, ขั้นตอนแรกจะถูกแฟล็กเพื่อรัน. เมื่อขั้นตอนแรกเสร็จสมบูรณ์แล้ว, ขั้นตอนถัดไปจะถูกแฟล็กเพื่อรัน, และต่อไป.

ข้างล่างนี้เป็นข้อควรพิจารณาเพิ่มเติมบางอย่างเมื่อใช้ Work Flow Manager:

- คุณสามารถเริ่มเวิร์กโฟลว์ด้วยตนเองที่ขั้นตอนใดๆ. เมื่อทำเช่นนั้น, หมายความว่า คุณจะข้าม ขั้นตอนก่อนหน้านี้ทั้งหมดในเวิร์กโฟลว์.
- ขั้นตอนแบบอัตโนมัติจะเสร็จสมบูรณ์หลังจากที่ขั้นตอนก่อนหน้านี้ทั้งหมดได้เสร็จสมบูรณ์แล้ว. นี้รวม ถึง predecessor Advanced Job Scheduler jobs ทั้งหมดด้วย.
- หลังจากที่ขั้นตอนเสร็จสมบูรณ์, successor Advanced Job Scheduler jobs จะ ถูกแฟล็กเพื่อรัน.
- ขั้นตอนแบบทำเองสามารถเสร็จสมบูรณ์ในลำดับใดๆ ก็ได้ ตราบเท่าที่ predecessor jobs ของขั้นตอนเสร็จสิ้นแล้ว.
- คุณสามารถทำเครื่องหมายขั้นตอนแบบทำเองที่สมบูรณ์แล้วให้เป็นไม่สมบูรณ์ และรันขั้นตอนนั้นอีกครั้งได้ ตราบเท่าที่ไม่มีขั้นตอนแบบอัตโนมัติที่ไม่สมบูรณ์อยู่ต่อจากนั้น.
- คุณสามารถกำหนดให้ขั้นตอนรองกว่างานจะเสร็จสมบูรณ์ ก่อนที่จะแจ้ง ความสมบูรณ์ของงานได้ โดยการระบุ predecessor jobs ที่เหมือนกับ successor jobs ของขั้นตอนก่อนหน้านี้.
- คุณสามารถแจ้งให้ผู้ใช้อื่นทราบเมื่อขั้นตอนเริ่มต้น, หยุด, หรือเมื่อขั้นตอน ไม่ได้เริ่มตามเวลาที่ระบุ, หรือเมื่อขั้นตอนให้เวลานานเกินไป. ตัวอย่างเช่น คุณสามารถแจ้งให้ ผู้ใช้ที่รับผิดชอบขั้นตอนแบบทำเองหนึ่งทราบ ว่า ขั้นตอนแบบอัตโนมัติก่อนหน้านี้ เสร็จสมบูรณ์แล้ว.

เมื่อคุณใช้เวิร์กโฟลว์, ไฟล์บันทึก activity จะแสดงเวลาที่เวิร์กโฟลว์เริ่มต้น, ขั้นตอนที่รัน, สถานะของขั้นตอนแบบอัตโนมัติ (สำเร็จหรือล้มเหลว), เวลาที่เวิร์กโฟลว์สิ้นสุด, และสถานะสุดท้ายของเวิร์กโฟลว์.

ตารางที่ 7. Work Flow Example

Work Flow	PAYROLL
จัดตารางเวลา	ทุกวันศุกร์ เวลา 1:00 p.m.
การแจ้งเตือน	เวิร์กโฟลว์ Clerk - Payroll ได้เริ่มขึ้นแล้ว
ขั้นตอน 1	อัตโนมัติ - ระบุ successor job ที่จะ initialize ไฟล์ค่าจ้าง
ขั้นตอน 2	อัตโนมัติ: <ul style="list-style-type: none"> ระบุ successor job จากขั้นตอน 1 เป็น predecessor job สำหรับขั้นตอนนี้ แจ้งให้ Clerk ทราบว่าสามารถป้อนบัตรลงเวลาได้
ขั้นตอน 3	ด้วยตนเอง: <ul style="list-style-type: none"> Clerk จะเสร็จสมบูรณ์หลังจากป้อนบัตรลงเวลา ระบุ successor job ที่จะประมวลผลไฟล์บัตรลงเวลาและพิมพ์รายงานบัตรลงเวลา แจ้งให้ Supervisor ทราบถ้าขั้นตอนไม่เสร็จสิ้นภายใน 120 นาที
ขั้นตอน 4	อัตโนมัติ: <ul style="list-style-type: none"> ระบุ successor job จากขั้นตอนก่อนหน้านี้เป็น predecessor job ไม่มี Successor jobs แจ้ง Clerk ให้ตรวจสอบรายงานบัตรลงเวลา
ขั้นตอน 5	ด้วยตนเอง: <ul style="list-style-type: none"> Clerk จะสมบูรณ์หลังจากการตรวจสอบรายงาน ระบุ successor job เพื่อประมวลผลค่าจ้าง
ขั้นตอน 6	อัตโนมัติ: <ul style="list-style-type: none"> ระบุ successor job จากขั้นตอนก่อนหน้านี้เป็น predecessor job ไม่มี Successor jobs แจ้งให้ Clerk และ Supervisor ทราบว่าค่าจ้างเสร็จสมบูรณ์แล้ว

ในตัวอย่างนี้ เวิร์กโฟลว์ PAYROLL จะเริ่มต้นทุกวันศุกร์เวลา 1:00 p.m. มีการส่งการแจ้งเตือนให้ Clerk ทราบว่าเวิร์กโฟลว์ได้เริ่มต้นแล้ว.

เนื่องจากขั้นตอน 1 เป็นขั้นตอนแบบอัตโนมัติและไม่มี predecessor jobs ใดๆ, ขั้นตอนจะแฟล็ก successor job ที่ initializes ไฟล์ค่าจ้างเพื่อรัน แล้วทำให้เสร็จสมบูรณ์. ขั้นตอน 2 มี successor job ของขั้นตอน 1 เป็น predecessor. ดังนั้นขั้นตอน 2 จะคอยให้งานที่ initializes ไฟล์ค่าจ้างเสร็จสมบูรณ์. หลังจากที่เสร็จสมบูรณ์แล้ว, ขั้นตอน 2 จะแจ้งให้ Clerk ทราบว่าเขาสามารถป้อนบัตรลงเวลาได้. ไม่มี successor jobs ที่จะแฟล็กเพื่อรัน.

Clerk จะทำขั้นตอน 3 ให้เสร็จสมบูรณ์ด้วยตนเองหลังจากที่ป้อนบัตรลงเวลาทั้งหมดแล้ว. successor job ที่ประมวลผลไฟล์บัตรลงเวลาและพิมพ์รายงานบัตรลงเวลาจะถูกแฟล็กเพื่อรัน. ข้อควรระวังคือ, Supervisor จะได้รับการแจ้งเตือน ถ้าขั้นตอน

ไม่เสร็จสิ้นภายใน 120 นาที. เนื่องจาก predecessor job ของขั้นตอน 4 คือ successor ของขั้นตอน 3, ดังนั้นขั้นตอน 4 จะรอจนกว่างานที่ประมวลผลไฟล์บัตรลงเวลาและพิมพ์รายงานบัตรลงเวลาจะเสร็จสมบูรณ์.

หลังจากที่งานเสร็จสมบูรณ์ Clerk จะได้รับแจ้งว่าสามารถตรวจสอบรายงานบัตรลงเวลาได้แล้ว. ไม่มี successor jobs ที่จะแฟล็กเพื่อรัน. หลังจากการตรวจสอบรายงานบัตรลงเวลา, Clerk จะทำขั้นตอน 5 ให้เสร็จสิ้นด้วยตนเอง. successor job ที่ประมวลผลค่าจ้างและจัดทำเช็คจะถูกแฟล็กเพื่อรัน.

เนื่องจาก predecessor job ของขั้นตอน 6 คือ successor ของขั้นตอน 5, ดังนั้นขั้นตอน 6 จะรอจนกว่างานที่ประมวลผลค่าจ้างและจัดทำเช็คจะเสร็จสมบูรณ์. หลังจากที่งานเสร็จสมบูรณ์, จะมีการแจ้งให้ Clerk และ Supervisor ทราบว่า Payroll ได้เสร็จสมบูรณ์แล้ว. ขณะนี้สามารถพิมพ์และแจกจ่ายเช็คได้.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ Work Flow Manager ให้ดูคำอธิบายทางออนไลน์.

การสร้างเวิร์กโฟลว์ใหม่:

เมื่อคุณสร้างเวิร์กโฟลว์ใหม่ นั้นหมายความว่า คุณจะระบุวิธีการเริ่ม เวิร์กโฟลว์, เวลาประมวลผลสูงสุดของเวิร์กโฟลว์, ขั้นตอนภารกิจและลำดับการรัน, การจัดตารางเวลา, การแจ้งเตือนและรายละเอียดของเอกสาร.

ถ้าต้องการสร้างเวิร์กโฟลว์ใหม่, ให้ทำตามต่อไปนี้:

- ใน iSeries Navigator ให้ขยาย My Connections → server → Work Management → Advanced Job Scheduler → **คลิกขวา Work Flow Manager → New Work Flow.**
หน้าต่าง New Work Flow จะปรากฏขึ้น.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับวิธีการทำหน้าต่าง New Work Flow ให้เสร็จสมบูรณ์ ให้ดูคำอธิบายทางออนไลน์.

หลังจากที่คุณตั้งค่าเวิร์กโฟลว์ของคุณแล้ว, คุณสามารถจัดการ เวิร์กโฟลว์ได้โดยการคลิกขวาบนชื่อเวิร์กโฟลว์ และคลิก **Work Flow Status.**

การเริ่มเวิร์กโฟลว์:

เมื่อคุณเริ่มเวิร์กโฟลว์, คุณสามารถเลือกได้ว่าคุณต้องการให้เวิร์กโฟลว์ เริ่มต้นบนลำดับแรกหรือบนลำดับที่ระบุ.

ถ้าต้องการเริ่มเวิร์กโฟลว์, ให้ทำตามขั้นตอนต่อไปนี้:

1. จาก iSeries Navigator, ให้ขยาย **Work Management** จาก → **Advanced Job Scheduler** → **Work Flow Manager** → **คลิกขวาบน เวิร์กโฟลว์** → **Start.** หน้าต่าง Start Work Flow จะปรากฏขึ้น.
2. เลือกว่าคุณต้องการให้เวิร์กโฟลว์ เริ่มต้นบนลำดับแรกหรือบนลำดับที่ระบุ. ถ้าคุณเลือกที่จะเริ่มที่ลำดับอื่นที่ไม่ใช่ลำดับแรก, ขั้นตอนก่อนหน้านี้ทั้งหมดจะถูกทำเครื่องหมายว่าเสร็จสิ้นแล้ว.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับหน้าต่าง Start Work Flow, ให้ดูคำอธิบายทางออนไลน์.

การทำงานกับเวิร์กโฟลว์:

คุณสามารถควบคุมและมอนิเตอร์เวิร์กโฟลว์เมื่อเวิร์กโฟลว์รันได้โดยใช้ หน้าต่าง Work Flow Status.

คุณสามารถเข้าใช้งานหน้าต่าง Work Flow Status โดยการขยาย My Connections → server → Work Management → Advanced Job Scheduler → Work Flow Manager → คลิกขวานเวิร์กโฟลว์ → Status.

- หน้าต่าง General แสดงสถานะปัจจุบันของเวิร์กโฟลว์.
- หน้าต่าง Steps แสดงรายการของขั้นตอนทั้งหมดที่ได้กำหนดที่เวิร์กโฟลว์ในปัจจุบัน.

คุณสามารถดูได้ว่าขั้นตอนใดถูกกำหนดเป็นแบบอัตโนมัติหรือแบบทำเอง อย่างไม่อย่างหนึ่ง และดูเวลาเริ่มต้นและสิ้นสุดขั้นตอน.

- เพื่อทำเครื่องหมายว่าขั้นตอนแบบทำเองได้เสร็จสมบูรณ์แล้ว, ให้เลือกขั้นตอนที่ถูกต้องและทำเครื่องหมายเลือกกล่อง Complete .
- ขั้นตอนแบบทำเองสามารถทำเครื่องหมายว่าเสร็จสมบูรณ์แล้วในลำดับใดๆ ก็ได้ ถ้า predecessor Advanced Job Scheduler jobs ทั้งหมดสำหรับขั้นตอนได้เสร็จสมบูรณ์แล้ว.
- ขั้นตอนแบบทำเองสามารถถูกทำเครื่องหมายว่า ไม่เสร็จสมบูรณ์ ถ้าไม่มีขั้นตอน แบบอัตโนมัติที่เสร็จสมบูรณ์อยู่ถัดไปในรายการ.
- สามารถเริ่มเวิร์กโฟลว์ด้วยตนเองได้ในทุกขั้นตอน. การทำเช่นนี้จะข้ามขั้นตอนก่อนหน้าทั้งหมด.

ถ้าต้องการรีเฟรชรายการ, ให้คลิก Refresh.

- หน้าต่าง Documentation แสดงข้อความเอกสารคู่มือ สำหรับเวิร์กโฟลว์.

การมอนิเตอร์ activity งานด้วย Advanced Job Scheduler:

ใช้ Advanced Job Scheduler เพื่อดูประวัติหรือสถานะของงานหรือ กลุ่มงาน. คุณยังสามารถตั้งค่าการเก็บรักษา activity, ซึ่งคุณกำหนดได้ว่าจะเก็บรักษาเรกคอร์ด activity ของงานไว้นานเท่าไร.

activity ของงานที่จัดตารางเวลาไว้:

activity ของงานที่ถูกจัดตารางเวลาไว้ช่วยให้คุณทราบว่า จะเก็บรักษาเรกคอร์ด activity ของ Advanced Job Scheduler ไว้นานเท่าไร. ค่าที่เป็นไปได้คือ 1 ถึง 999 วันหรือจำนวนครั้ง. คุณสามารถระบุให้เก็บรักษา activity ตามจำนวนวันที่ต้องการ, หรือตามจำนวนครั้งที่เกิดขึ้นต่องานแต่ละงาน.

รายละเอียดเกี่ยวกับงานที่จัดตารางเวลาไว้แสดงอยู่ข้างล่างนี้:

- ชื่อ ชื่อของงานที่จัดตารางเวลา.
- กลุ่ม ชื่อของกลุ่มงานสำหรับงาน.
- ลำดับ หมายเลขลำดับของงานภายในกลุ่ม, ถ้างานอยู่ใน กลุ่มงาน.
- สถานะความคืบหน้า สถานะของงาน.
- เวลาเริ่มต้น เวลาที่งานเริ่มต้นการรัน.
- เวลาสิ้นสุด เวลาที่งานสิ้นสุด.
- เวลาที่ใช้ไป จำนวนเวลาในหน่วยชั่วโมงและนาทีที่งานใช้ในการประมวลผล.

การระบุการเก็บรักษา activity:

ถ้าต้องการระบุการเก็บรักษา activity, ให้ทำตามขั้นตอนเหล่านี้:

1. เปิด Work Management จากหน้าต่าง iSeries Navigator .
2. ขยาย Advanced Job Scheduler.

3. คลิกขวาบน **Scheduled Job Activity** และคลิก **Properties**.

การดูรายละเอียดของ activity งานที่จัดตารางเวลา:

ถ้าต้องการดูรายละเอียดของ activity งานที่จัดตารางเวลาไว้, ให้ทำตามขั้นตอนเหล่านี้:

1. เปิด **Work Management** จากหน้าต่าง iSeries Navigator .
2. ขยาย **Advanced Job Scheduler**.
3. ดับเบิลคลิกที่ **Scheduled Job Activity**.

การดู activity งานที่จัดตารางเวลาสำหรับงานเฉพาะ:

ถ้าต้องการดู activity ของงานที่จัดตารางเวลาไว้สำหรับงานเฉพาะ, ให้ทำตามขั้นตอนเหล่านี้:

1. เปิด **Work Management** จากหน้าต่าง iSeries Navigator .
2. ขยาย **Advanced Job Scheduler**.
3. คลิก **Scheduled jobs**.
4. คลิกขวา Job Name ที่มี activity ซึ่งคุณต้องการจะแสดง และ คลิก **Activity**.

การดูรายละเอียดของไฟล์บันทึก activity:

ไฟล์บันทึก activity แสดง activity ที่เกิดขึ้นภายในตัวจัดตารางเวลา เช่น การเพิ่มงานในตาราง, การเปลี่ยนงานในตาราง, หรือ การส่ง. การละเมิดความปลอดภัย, ลำดับการประมวลผลโดยงานที่ถูกจัดตารางเวลาไว้, และความผิดพลาดต่างๆ ที่เกิดขึ้นจะถูกแสดงให้เห็น. วันที่และเวลาสำหรับ activity ก่อนหน้านั้นจะถูกแสดงไว้ด้วย.

เพื่อดูรายละเอียดของข้อความ, ให้ดับเบิลคลิกวันที่และเวลา. ถ้าต้องการดูรายละเอียดไฟล์บันทึก activity, ให้ทำตามขั้นตอนเหล่านี้:

1. เปิด **Work Management** จากหน้าต่าง iSeries Navigator .
2. ขยาย **Advanced Job Scheduler**.
3. คลิก **Activity Log**. entry ของวันที่ปัจจุบันจะแสดงขึ้น. เพื่อเปลี่ยนเกณฑ์การเลือก, ให้เลือก **Include** จากเมนู **Options**.

การดูไฟล์บันทึก activity สำหรับงานเฉพาะ:

ถ้าต้องการดูไฟล์บันทึก activity ของงานที่เจาะจง, ให้ทำตามขั้นตอนเหล่านี้:

1. เปิด **Work Management** จากหน้าต่าง iSeries Navigator .
2. ขยาย **Advanced Job Scheduler**.
3. คลิก **Scheduled jobs**.
4. คลิกขวา **Job Name** ที่มีไฟล์บันทึก activity ที่คุณต้องการแสดง และคลิก **Activity log**.

คุณยังสามารถใช้หน้า **Last Run** ของ คุณสมบัติของงานเพื่อดูความก้าวหน้าของงานได้ด้วย. ระบุคำสั่ง Set Step using Job Scheduler (SETSTPJS) ก่อนหรือหลังจากขั้นตอน ในโปรแกรม CL พร้อมกับ description ที่ระบุความก้าวหน้าของงาน. เมื่องานดำเนินไปถึงคำสั่ง SETSTPJS ในโปรแกรม, description จะถูกแสดงขึ้นในหน้า Last Run และบนอุปกรณ์ไร้สายของคุณ.

การมอนิเตอร์ข้อความด้วย Advanced Job Scheduler:

ใส่เพิ่ม identifier ข้อความลงในคำสั่งใดๆ ที่อยู่ในรายการคำสั่งของงานเพื่อมอนิเตอร์ข้อความ.

แต่ละคำสั่งในรายการคำสั่งของงานงานหนึ่งสามารถมี identifier ข้อความเพื่อการมอนิเตอร์ได้หลายตัว. เมื่องานรันและมีการออกข้อความแสดงความผิดพลาด ที่ตรงกับข้อความใดข้อความหนึ่งที่ย้อนสำหรับคำสั่งที่เลือก, งานจะล็อกข้อผิดพลาด แต่ยังคงประมวลผลต่อไปด้วยคำสั่งถัดไปใน รายการ.

ถ้าเลขศูนย์ถูกระบุไว้ในสองตำแหน่งขวาสุดหรือทั้งสี่ตำแหน่งขวาสุด อย่างใดอย่างหนึ่ง, เช่น ppmm00, นั่นคือการระบุ identifier ข้อความแบบทั่วไป. ตัวอย่าง, ถ้า CPF0000 ถูกระบุไว้, หมายความว่าข้อความที่ขึ้นต้นด้วย CPF ทั้งหมดจะถูกมอนิเตอร์.

ถ้าต้องการใส่เพิ่ม identifier ข้อความลงในคำสั่ง, ให้ทำตามขั้นตอนเหล่านี้:

1. เปิด **Work Management** จากหน้าต่าง iSeries Navigator .
2. คลิกขวาที่ **Advanced Job Scheduler**.
3. คลิก **Scheduled Jobs** เพื่อแสดงรายการของงาน.
4. คลิกขวานงานที่จัดตารางเวลาและคลิก **Properties**.
5. เลือกคำสั่งจากรายการ และคลิก **Properties**.
6. คลิก **Messages**.
7. ใส่ identifier ข้อความที่จะมอนิเตอร์และคลิก **Add**.

การสร้างและทำงานกับพื้นที่ข้อมูลแบบโลคัล:

พื้นที่ข้อมูลแบบโลคัลคือส่วนของพื้นที่ว่างที่ถูกจัดสรรสำหรับ งาน. ไม่ทุกงานที่ใช้พื้นที่ข้อมูลแบบโลคัลของตน แต่มีเพียงบางงานเท่านั้น. แต่ละคำสั่งภายใน งานมีสิทธิเข้าใช้งานพื้นที่ข้อมูลแบบโลคัลของงาน. คุณอาจต้องการใช้พื้นที่ข้อมูลแบบโลคัล ถ้าคุณกำลังจัดตารางเวลางาน ซึ่งก่อนหน้านี้กำหนดให้คุณต้องระบุ พารามิเตอร์เพิ่มเติมด้วยตนเอง. ใช้พื้นที่ข้อมูลแบบโลคัลเพื่อระบุพารามิเตอร์เพิ่มเติม เพื่อให้คุณไม่ต้องระบุพารามิเตอร์ด้วยตนเองในทุกครั้งที่งานเริ่มขึ้น.

ถ้าต้องการระบุข้อมูลพื้นที่ข้อมูลแบบโลคัลสำหรับงานที่จัดตารางเวลา, ให้ทำ ตามขั้นตอนเหล่านี้:

1. เปิด **Work Management** จากหน้าต่าง iSeries Navigator .
2. ขยาย **Advanced Job Scheduler** → **Scheduled Jobs**.
3. คลิกขวานงานและคลิก **Properties**.
4. แก้ไขหน้าต่าง Local Data Area ตามความจำเป็น.

อ้างอิงคำอธิบายทางออนไลน์สำหรับข้อมูลเพิ่มเติม เมื่อคุณกรอกรายละเอียด สำหรับพื้นที่ข้อมูลแบบโลคัล.

การสร้างและทำงานกับแอ็พพลิเคชัน/ตัวควบคุมงาน:

แอ็พพลิเคชันคืองานหลายๆ งานที่ถูกรวมเป็นกลุ่มไว้เพื่อการประมวลผล. มีความหมายกว้างกว่ากลุ่มของงานที่ไม่มีความจำเป็นต้องประมวลผลตามลำดับ. งานหลายงานในแอ็พพลิเคชันต่างๆ สามารถประมวลผลไปพร้อมกันได้และงานแต่ละงานไม่ต้องรอให้งานอื่นประมวลผลเสร็จก่อน. เราสามารถทำงานกับทุกๆ งานภายในแอ็พพลิเคชันและแต่ละงานสามารถมีชุดของค่าตีฟอลต์ของตัวเอง. ตัวควบคุมงานทั้งหลายคือค่าตีฟอลต์ที่ถูกกำหนดลงไปในงานในขณะที่คุณเพิ่มงานลงในตารางเวลางาน เช่นเดียวกับค่าตีฟอลต์ที่ถูกใช้เมื่อมีการส่งงาน.

แอ็พพลิเคชั่น คืองานที่ถูกจัดกลุ่มเข้าด้วยกันสำหรับการประมวลผล. ตัวอย่างเช่น, คุณอาจมีชุดของงานที่คุณใช้สำหรับระบบเงินเดือนที่คุณต้องการจัดกลุ่มเข้าด้วยกันสำหรับประมวลผลทางการบัญชี.

ตัวควบคุมงาน คือค่าดีฟอลต์ที่ถูกกำหนดให้กับงานหนึ่งในขณะที่คุณเพิ่มมันลงไปในตารางเวลางาน เช่นเดียวกับการใช้ค่าดีฟอลต์เมื่อมีการส่งงานให้เข้าไปประมวลผล. ตัวควบคุมงาน ดีฟอลต์ ได้แก่ ปฏิทิน, ปฏิทินวันหยุด, คิวงาน, รายละเอียดของงาน และอื่นๆ.

คุณสามารถแสดงผลแอ็พพลิเคชั่น/ตัวควบคุมงานที่มีอยู่ทั้งหมดบนระบบของคุณ. คุณสามารถใส่เพิ่มแอ็พพลิเคชั่น/ตัวควบคุมงานใหม่, ใส่เพิ่มแอ็พพลิเคชั่น/ตัวควบคุมงานใหม่โดยใช้ฐานเดิมที่มีอยู่, หรือลบแอ็พพลิเคชั่น/ตัวควบคุมงานออก. คุณยังสามารถเลือกแอ็พพลิเคชั่น/ตัวควบคุมงาน และแสดงผลคุณสมบัติของมันเพื่อทำการเปลี่ยนแปลง.

ถ้าต้องการสร้างแอ็พพลิเคชั่น/ตัวควบคุมงานใหม่, ให้ทำตามขั้นตอนดังต่อไปนี้:

1. ขยาย **Work Management** จากหน้าต่าง iSeries Navigator.
2. คลิกขวา **Advanced Job Scheduler** และคลิก **Properties**.
3. คลิกแท็บ **Applications/Job Controls**.
4. คลิก **New** และป้อนชื่อแอ็พพลิเคชั่น.
5. พิมพ์คำอธิบายสำหรับแอ็พพลิเคชั่น.
6. ให้เลือกคนติดต่อสำหรับแอ็พพลิเคชั่น. คนติดต่อคือ ชื่อของผู้ใช้ที่คุณติดต่อด้วยถ้าคุณมีปัญหาเกี่ยวกับงานที่อยู่ในแอ็พพลิเคชั่น. คุณสามารถระบุคนติดต่อได้สูงสุดถึง 5 คนต่อแอ็พพลิเคชั่น. คุณยังอาจเลือกที่จะใส่เพิ่มหรือลบคนติดต่อออกจากรายชื่อคนติดต่อ.
7. คุณอาจพิมพ์ข้อมูลเพิ่มเติมเพื่อช่วยให้คุณระบุถึงแอ็พพลิเคชั่นได้. ข้อมูลนั้นจะเชื่อมโยงกับ แอ็พพลิเคชั่นใหม่. ข้อมูลนี้อาจเป็นประโยชน์หากเกิดปัญหาใดๆ ขึ้น.

การทำงานกับการแจ้งเตือน:

ข้อมูลนี้อธิบายวิธีการใช้ฟังก์ชันการแจ้งเตือนของ Advanced Job Scheduler.

ภายในการแจ้งเตือน, คุณสามารถทำภารกิจต่างๆ ได้. การแจ้งเตือนอนุญาตให้คุณระบุคุณสมบัติของผู้รับและคุณสมบัติรายการการแจกจ่ายรายงาน. ยิ่งกว่านั้น, คุณสามารถส่งอีเมล และจัดลำดับการส่งต่อข้อความ ในกรณีที่ผู้รับไม่มีการตอบกลับภายในเวลาที่กำหนด.

ก่อนที่คุณจะ สามารถส่งข้อความอีเมลได้, คุณต้องระบุเมลเซิร์ฟเวอร์ที่จะใช้สำหรับการแจ้งเตือน .

ต่อไปนี้เป็นฟังก์ชันการแจ้งเตือนที่สำคัญของ Advanced Job Scheduler:

ผู้รับ เมื่อจัดตารางเวลางาน, คุณสามารถระบุว่าการให้ส่งข้อความการแจ้งเตือน ให้แก่ผู้รับที่ระบุหรือไม่. คุณสามารถส่งข้อความแจ้งเตือนถ้าหากงานล้มเหลว, งานเสร็จสมบูรณ์, หรืองานนั้นไม่เริ่มทำงานภายในเวลาที่กำหนด. สำหรับผู้รับแต่ละคน, คุณต้องกำหนดคุณสมบัติของผู้รับ. คุณ สามารถเข้าถึงคุณสมบัติของผู้รับได้โดยการเลือก **Advanced Job Scheduler** → **Notification** → **Recipients**, แล้วเลือกผู้รับจากรายชื่อของผู้รับ.

รายการการแจกจ่ายรายงาน

ให้ใช้รายการการแจกจ่ายรายงานในการระบุรายการของสพูลไฟล์ที่อนุญาตให้แจกจ่ายได้. แต่ละสพูลไฟล์ที่เกิดจากงานจะถูกตรวจว่ามีรายชื่อที่ตรงกันอยู่ในรายการสพูลไฟล์หรือไม่. ถ้าตรงกัน, ผู้รับที่เกี่ยวข้องกับสพูลไฟล์นั้นจะได้

รับสำเนาของสพูลไฟล์ผ่านทาง อีเมล, สำเนาของสพูลไฟล์ในเอาต์พุตคิว, หรือทั้งสองอย่าง. คุณสามารถ เข้าใช้งาน รายการการแจกจ่ายรายงานโดยการเลือก **Advanced Job Scheduler** → **Notification** → **Report distribution list**.

อีเมล คุณสามารถส่งข้อความอีเมลไปยังผู้รับที่ถูกระบุอยู่ในรายการผู้รับหรืออาจส่งไปยังอีเมลแอดเดรสที่เฉพาะเจาะจงก็ได้. ในคุณสมบัติของผู้รับ ต้องมีอีเมลแอดเดรสที่ต้องการส่งข้อความไป. ตอนที่ส่งข้อความอีเมล, คุณสามารถแนบสพูลไฟล์ไปด้วย. คุณสามารถส่งสพูลไฟล์ที่มีรูปแบบเป็น PDF. ยิ่งไปกว่านั้น, คุณสามารถระบุรายการการส่งต่อที่จะใช้ ถ้าผู้รับ ไม่ตอบสนองภายในช่วงเวลาที่จะระบุ.

การระบุสพูลไฟล์ที่จะแนบกับอีเมล:

ถ้าต้องการระบุสพูลไฟล์ที่จะแนบไปกับอีเมล, ให้ทำดังต่อไปนี้:

1. ขยาย **Basic Operations** จากหน้าต่าง iSeries Navigator.
2. คลิก **Printer Output**.
3. คลิกขวาที่สพูลไฟล์และคลิก **Send via AJS**.
4. การระบุผู้รับ, เรื่อง, และข้อความ.

หมายเหตุ: ยังสามารถทำได้จาก **Output Queues**.

รายการการส่งต่อ

รายการการส่งต่อระบุรายชื่อของผู้รับตามลำดับpriorityชั้นตัวหลังสุดไปแรกสุด. จะมีการแจ้งเตือนผู้รับตามลำดับในรายการ. ถ้าผู้รับคนแรกไม่ตอบข้อความ, ข้อความจะถูกส่งไปยังผู้รับคนถัดไป. กระบวนการนี้จะดำเนินไปจนกว่าจะมีการตอบข้อความนั้น. เพื่อกำหนดรายการการส่งต่อ, ให้ไปที่ **Advanced Job Scheduler** → **Notification** → **Escalation Lists**.

การหยุดข้อความไม่ให้ส่งต่อ:

ถ้าต้องการหยุดข้อความไม่ให้ส่งต่อ, ให้ทำดังนี้:

1. ขยาย **Work Management** จากหน้าต่าง iSeries Navigator.
2. คลิก **Advanced Job Scheduler** → **Notification** → **E-mail** → **Sent**.
3. คลิกขวาที่ข้อความการส่งต่อ และคลิก **Stop**.

หมายเหตุ: ถ้าต้องการดูเฉพาะข้อความการส่งต่อ, ให้เลือก **View** → **Customize this view** → **Include** จากหน้าต่าง iSeries Navigator. จากนั้น, ในฟิลด์ **Type**, เลือก **Escalating**.

การทำงานกับรายชื่อไลบรารี:

รายชื่อไลบรารีคือรายชื่อของไลบรารีที่ผู้ใช้เป็นผู้กำหนดเพื่อให้ **Advanced Job Scheduler** ใช้เมื่อมีการประมวลผลงาน.

รายชื่อไลบรารี เป็นรายชื่อของไลบรารีที่ผู้ใช้กำหนดเองที่ถูกใช้งานโดย **Advanced Job Scheduler** เพื่อค้นหาข้อมูลที่ต้องการ ในขณะที่กำลังประมวลผล. คุณสามารถแสดงรายชื่อไลบรารี, ใส่เพิ่มรายชื่อไลบรารีใหม่, ใส่เพิ่มรายชื่อไลบรารีใหม่โดยใช้ฐานเดิมจากไลบรารีที่มีอยู่แล้ว, หรือลบรายชื่อไลบรารีออก, ถ้ารายชื่อไลบรารีไม่ได้ถูกใช้อยู่โดยงานที่จัดตารางเวลาในปัจจุบัน.

คุณสามารถเลือกรายชื่อ และแสดงผลคุณสมบัติของมันเพื่อทำการเปลี่ยนแปลง. คุณสามารถใส่ไลบรารีได้สูงสุดถึง 250 ไลบรารีบนรายชื่อไลบรารี.

ถ้าต้องการใส่เพิ่มรายชื่อไลบรารีใหม่, ให้ทำตามขั้นตอนเหล่านี้:

1. เปิด **Work Management** จากหน้าต่าง iSeries Navigator .
2. คลิกขวา **Advanced Job Scheduler** และคลิก **Properties**.
3. คลิกแท็บ **Library Lists**.
4. คลิก **New** และพิมพ์ชื่อสำหรับรายชื่อไลบรารี.
5. พิมพ์คำอธิบายสำหรับรายชื่อไลบรารี.
6. คลิก **Browse** เพื่อดูรายชื่อของไลบรารีที่มีอยู่, และคลิกไลบรารี.
7. คลิก **Add** เพื่อเพิ่มรายชื่อของไลบรารีที่เลือก.

การทำงานกับตัวแปรคำสั่ง:

ตัวแปรคำสั่ง (เดิมเรียกว่าพารามิเตอร์) คือตัวแปรที่คุณอาจเก็บและใช้ในหลายๆ งานที่ถูกส่งผ่าน Advanced Job Scheduler. ตัวอย่างของตัวแปรคำสั่งได้แก่ การเริ่มต้นแต่ละเดือน, หมายเลขฝ่าย, รหัสบริษัท, และอื่นๆ .

ตัวแปรคำสั่ง (ที่เดิมเรียกกันว่า พารามิเตอร์) คือตัวแปรที่คุณเก็บอยู่ใน Advanced Job Scheduler และใช้ในงานที่ถูกส่งผ่านไปยัง Advanced Job Scheduler. ตัวแปรคำสั่งประกอบด้วยข้อมูลที่จะถูกแทนที่ภายในสตริงคำสั่งของตารางเวลาทำงาน. ตัวอย่างของตัวแปรคำสั่งได้แก่ การเริ่มต้นของแต่ละเดือน, หมายเลขแผนกของบริษัท, หมายเลขบริษัท และอื่นๆ . คุณสามารถแสดงผลตัวแปรคำสั่ง, ใส่เพิ่มตัวแปรคำสั่งใหม่, ใส่เพิ่มตัวแปรคำสั่งใหม่โดยใช้ฐานเดิมจากตัวแปรคำสั่งที่มีอยู่แล้ว, หรือลบตัวแปรคำสั่งออก, ถ้าตัวแปรไม่ได้ถูกใช้งานในปัจจุบันโดยงานที่จัดตารางเวลา.

คุณสามารถเลือกตัวแปรคำสั่งที่มีอยู่แล้วและแสดงผลคุณสมบัติของมันเพื่อทำการเปลี่ยนแปลง.

ถ้าต้องการใส่เพิ่มตัวแปรคำสั่ง, ให้ทำตามขั้นตอนเหล่านี้:

1. เปิด **Work Management** จากหน้าต่าง iSeries Navigator .
2. คลิกขวา **Advanced Job Scheduler** และคลิก **Properties**.
3. คลิกแท็บ **Command Variables**.
4. คลิก **New** และพิมพ์ชื่อสำหรับตัวแปรคำสั่ง.
5. พิมพ์คำอธิบายสำหรับตัวแปรคำสั่ง.
6. พิมพ์ความยาวของตัวแปรคำสั่ง. ความยาวสามารถกำหนดได้ตั้งแต่ 1 ถึง 90.
7. ให้เลือกวิธีที่คุณต้องการกำหนดค่าทดแทน:
 - a. ให้ระบุข้อมูลที่ใช้สำหรับตัวแปรคำสั่ง. คุณอาจใช้อักขระใดๆ ก็ได้ในฟิลด์นี้. จำนวนของอักขระในข้อมูลต้องไม่มากกว่าความยาวที่ระบุในฟิลด์ **Length**.
 - b. พิมพ์สูตรที่ใช้คำนวณวันที่. (สำหรับตัวอย่าง, ให้ดูคำอธิบายทางออนไลน์.)
 - c. พิมพ์ชื่อโปรแกรมที่คุณใช้เพื่อเรียกค่าทดแทนออกมา.
 - d. พิมพ์ไลบรารีที่คุณใช้เพื่อเรียกค่าทดแทนออกมา.
 - e. ให้เลือกว่าคุณต้องการดึงข้อมูลค่าทดแทนออกมาจากผู้ควบคุมระบบตอนรันใหม่หรือไม่.

การทำงานกับ **Advanced Job Scheduler for Wireless**:

Advanced Job Scheduler for Wireless ทำงานบนอุปกรณ์สองประเภท. อุปกรณ์ Wireless Markup Language (WML) คือ โทรศัพท์มือถือที่พร้อมเชื่อมต่ออินเทอร์เน็ต. อุปกรณ์ Hypertext Markup Language (HTML) คือ PDA หรือ PC เว็บเบราว์เซอร์. ตลอดหัวข้อนี้, อุปกรณ์ที่แตกต่างอื่นจะถูกอ้างอิงถึงเป็น WML และ HTML.

ข้อกำหนดฮาร์ดแวร์และซอฟต์แวร์:

กำหนดว่าคุณมีซอฟต์แวร์และฮาร์ดแวร์ทั้งหมดที่จำเป็นสำหรับการรัน Advanced Job Scheduler for Wireless หรือไม่.

องค์ประกอบต่อไปนี้เป็นสิ่งจำเป็นสำหรับการรัน Advanced Job Scheduler for Wireless:

- ไสเซนส์โปรแกรม 5722-JS1 V5R3: ผลิตภัณฑ์ Advanced Job Scheduler ที่มี Advanced Job Scheduler for Wireless.
- อุปกรณ์ที่ใช้รันฟังก์ชัน
 - โทรศัพท์ที่พร้อมเชื่อมต่ออินเทอร์เน็ตที่มีการบริการอินเทอร์เน็ตไร้สาย
 - PDA พร้อมเว็บเบราว์เซอร์, โมเด็มไร้สาย, และบริการอินเทอร์เน็ตไร้สาย
 - เวอร์กสเตชันที่มีเว็บเบราว์เซอร์แบบดั้งเดิม
- เซิร์ฟเวอร์ที่รัน i5/OS^(R) V5R3 หรือหลังจากนั้นในเน็ตเวิร์ก TCP/IP.
- เว็บแอปพลิเคชันเซิร์ฟเวอร์ที่รันบนระบบศูนย์กลาง, ซึ่งได้แก่สิ่งต่อไปนี้:
 - ASF Jakarta Tomcat Application server
 - แอปพลิเคชันเซิร์ฟเวอร์อื่นที่รันบนระบบศูนย์กลาง, ที่มีความสามารถ เป็น host servlets
- เซิร์ฟเวอร์ HTTP ที่ติดตั้งบนเซิร์ฟเวอร์ iSeries
- ระบุเซิร์ฟเวอร์ HTTP ของคุณพร้อมด้วยคุณลักษณะพิเศษไร้สาย Advanced Job Scheduler. ในการทำเช่นนั้น, ให้เชื่อมต่อระบบ iSeries ของคุณที่มี Advanced Job Scheduler ติดตั้งอยู่ โดยการใช้อินเทอร์เน็ตเฟสแบบอักษร. จากนั้น, ระบุคำสั่งต่อไปนี้:

CALL QIJS/QIJSINT

การเลือกอุปกรณ์:

เลือกอุปกรณ์ที่เข้ากันได้กับ Advanced Job Scheduler for Wireless.

โทรศัพท์ที่พร้อมเชื่อมต่ออินเทอร์เน็ตและ PDAs ไร้สายเป็นเทคโนโลยีที่เปลี่ยนแปลงอย่างรวดเร็วมาก. เทคโนโลยีเหล่านี้แตกต่างทั้งในด้านขนาดจอภาพ, ในด้านการมองและความรู้สึก, และในลักษณะที่สำคัญอื่นๆ อีกมากมาย. ส่วนต่อไปนี้ช่วยคุณในการเลือก อุปกรณ์ที่เข้ากันได้กับ Advanced Job Scheduler for Wireless. อุปกรณ์ไร้สาย อื่นๆ อาจจะทำงานเข้ากันได้ด้วย ถ้าอุปกรณ์นั้นสนับสนุนการ บราวซิ่งอินเทอร์เน็ตแบบไร้สาย, แต่การโต้ตอบอาจแตกต่างกัน.

โทรศัพท์ที่พร้อมเชื่อมต่ออินเทอร์เน็ต เลือกโทรศัพท์ที่พร้อมเชื่อมต่ออินเทอร์เน็ตเพื่อใช้กับ Advanced Job Scheduler for Wireless.

PDAs เลือก PDA เพื่อใช้กับ Advanced Job Scheduler for Wireless.

PCs คุณยังสามารถใช้เว็บเบราว์เซอร์แบบดั้งเดิมกับ Advanced Job Scheduler for Wireless ได้ด้วย.

การตั้งค่าสถานะแวดล้อมไร้สายของคุณ:

แก้ไขเว็บแอปพลิเคชันเซิร์ฟเวอร์ของคุณและ firewall configuration เพื่อให้ Advanced Job Scheduler for Wireless รันได้อย่างเหมาะสม.

ก่อนที่คุณจะเริ่มใช้ Advanced Job Scheduler for Wireless, ตรวจสอบให้แน่ใจว่าคุณได้ตั้งค่าอย่างถูกต้องหรือตั้งค่าไอเท็มดังต่อไปนี้แล้ว:

1. ตั้งค่าเว็บแอปพลิเคชันเซิร์ฟเวอร์ของคุณ ตั้งค่า Advanced Job Scheduler for Wireless เพื่อรันโดยใช้ ASF Jakarta Tomcat servlet engine. คำสั่งเครื่องเหล่านี้จะระบุวิธีการสร้าง และเริ่มเว็บแอปพลิเคชันเซิร์ฟเวอร์ของคุณ. นอกจากนี้, คำสั่งยังระบุโปรแกรมที่คุณต้องรันก่อนการทำงานกับฟังก์ชันไร้สายของ Advanced Job Server ด้วย.
2. การตั้งค่าไฟร์วอลล์ของคุณ หัวข้อนี้อธิบายวิธีการตั้งค่าไฟร์วอลล์ของคุณสำหรับ iSeries Navigator for Wireless. ขั้นตอน configuration เหล่านี้ยังใช้กับ Advanced Job Scheduler for Wireless ด้วย. ให้เรียกดูหัวข้อนี้เพื่อพิจารณาว่า คุณต้องดัดแปลงไฟร์วอลล์ของคุณเพื่อให้สามารถเข้าถึงระบบจากอุปกรณ์ไร้สายหรือไม่.
3. การเลือกภาษา ภาษาดีฟอลต์ถูกเซตเป็น English, แต่คุณสามารถตั้งค่าอุปกรณ์ของคุณให้แสดงภาษาที่คุณเลือกได้.

หลังจากที่คุณทำขั้นตอนเหล่านี้เสร็จสิ้นแล้ว, คุณก็พร้อมที่จะเชื่อมต่อกับ เซิร์ฟเวอร์ของคุณ และเริ่มต้นโดยใช้ Advanced Job Scheduler for Wireless.

การตั้งค่าเว็บแอปพลิเคชันเซิร์ฟเวอร์ของคุณ:

ก่อนการทำงานกับ Advanced Job Scheduler for Wireless, คุณต้องเริ่มต้นและตั้งค่าเว็บแอปพลิเคชันเซิร์ฟเวอร์. ขั้นตอนต่อไปนี้เป็น การตั้งค่า ASF Tomcat servlet engine สำหรับเซิร์ฟเวอร์ HTTP (สนับสนุนโดย Apache) เพื่อรัน Advanced Job Scheduler for Wireless.

ข้อกำหนด

ก่อนคุณจะเริ่มต้น, คุณต้องมีสิทธิการใช้งานระดับ QSECOFR และมีสิ่งต่อไปนี้ติดตั้งไว้แล้ว:

- IBM^(R) HTTP Server (5722-DG1)

หมายเหตุ: วิธีการต่อไปนี้จะสร้าง instance ของเซิร์ฟเวอร์ HTTP อันใหม่ขึ้นมา; คุณไม่สามารถใช้วิธีการต่อไปนี้เป็นเพื่อตั้งค่า Advanced Job Scheduler บนเซิร์ฟเวอร์ HTTP ที่มีอยู่แล้ว.

Initialize Advanced Job Scheduler for Wireless บนเซิร์ฟเวอร์ HTTP

การรันคำสั่งต่อไปนี้จะใส่เพิ่ม Advanced Job Scheduler for Wireless servlet ที่ Apache Software Foundation Jakarta Tomcat servlet engine. และยังคงตั้งค่า IBM เซิร์ฟเวอร์ HTTP (สนับสนุนโดย Apache) ที่ชื่อว่า Advanced Job SchedulerP ซึ่งคอยรอรับคำขอบนพอร์ต 8210.

ก่อนการทำงานกับ Advanced Job Scheduler for Wireless, คุณต้อง initialize Advanced Job Scheduler for Wireless บน HTTP server instance บนระบบ iSeries ของคุณ. เพื่อทำเช่นนี้, ให้ระบุคำสั่งต่อไปนี้จากอินเตอร์เฟซแบบอักษร. คำสั่งนี้จะรันโปรแกรมที่ให้มาพร้อมกับระบบ iSeries .

CALL QIJS/QIJSINT

หลังจากที่คุณตั้งค่าเว็บแอปพลิเคชันเซิร์ฟเวอร์และ initialize Advanced Job Scheduler instance บนเว็บแอปพลิเคชันเซิร์ฟเวอร์แล้ว, คุณสามารถตั้งค่า สภาวะแวดล้อม Advanced Job Scheduler wireless ต่อไปได้.

การตั้งค่าไฟร์วอลล์ของคุณ:

กำหนดว่าคุณต้องแก้ไขไฟร์วอลล์ของคุณเพื่อให้สามารถเข้าถึงระบบ จากอุปกรณ์ไร้สายหรือไม่.

เมื่อคุณใช้ iSeries Navigator for Wireless, คุณจะเข้าถึงระบบของคุณอย่างน้อยหนึ่งระบบจากอินเทอร์เน็ต. ถ้าวันนี้คุณเข้าถึงระบบใดๆ ของคุณจากอินเทอร์เน็ต, คุณอาจจะมี การตั้งค่าไฟร์วอลล์ที่ป้องกันการเข้าใช้งานที่ไม่ได้รับอนุญาต. ขึ้นอยู่กับ configuration ไฟร์วอลล์ของคุณ, คุณอาจต้องแก้ไขการตั้งค่าไฟร์วอลล์ของคุณเพื่อรัน iSeries Navigator for Wireless.

ถ้าคุณไม่เคยเข้าใช้งานระบบของคุณจากอินเทอร์เน็ตและไม่มี การตั้งค่าไฟร์วอลล์, IBM Redbook ต่อไปนี้แสดงกลยุทธ์สำหรับการทำสิ่งดังกล่าวในบทที่แสดงข้อมูลเกี่ยวกับสถาปัตยกรรมโฮสต์ที่กั้นกรองและสถาปัตยกรรม subnet ที่กั้นกรอง. ดู AS/400® Internet Security Scenarios: A Practical Approach®.

การเลือกภาษา:

เมื่อคุณเชื่อมต่อกับ Advanced Job Scheduler for wireless, คุณสามารถ ระบุภาษาที่จะใช้ได้. หากคุณไม่ต้องการระบุภาษาเฉพาะ, คุณสามารถเชื่อมต่อกับ iSeries ของคุณต่อไป.

ถ้าต้องการระบุภาษา, ให้ใช้ URL ดังต่อไปนี้:

โฮสต์. โดเมน: พอร์ต/servlet/AJSPervasive?lng=lang

- โฮสต์: ชื่อโฮสต์ของระบบที่มีผลิตภัณฑ์.
- โดเมน: โดเมนที่โฮสต์ตั้งอยู่.
- พอร์ต: พอร์ตที่ instance ของเว็บเซิร์ฟเวอร์กำลังเฝ้าฟังอยู่
- lang: identifier 2 อักขระสำหรับภาษา. ข้อมูลต่อไปนี้ คือรายการของภาษาที่พร้อมใช้งานและ identifier 2 อักขระของภาษา. (ar: Arabic de: German en: English es: Spanish fr: French it: Italian ja: Japanese)

ขณะนี้คุณสามารถเริ่มทำงานกับ Advanced Job Scheduler for Wireless.

การเชื่อมต่อกับ iSeries:

เชื่อมต่อกับ iSeries ของคุณที่มีผลิตภัณฑ์ Advanced Job Scheduler โดยใช้อุปกรณ์ไร้สายของคุณ.

เพื่อเริ่มโดยใช้ Advanced Job Scheduler for Wireless, ให้ระบุ URL ของ iSeries ของคุณลงใน อุปกรณ์ไร้สายของคุณ. เมื่อใช้อุปกรณ์ของคุณไปที่ URL บน iSeries, ให้ใช้ รูปแบบดังต่อไปนี้. ตรวจสอบให้แน่ใจว่ามีการพิมพ์จุดสิ้นสุดของ URL (/servlet/AdvancedJobSchedulerPervasive) อย่างแม่นยำดังที่แสดงต่อไปนี้:

โฮสต์. โดเมน: พอร์ต/servlet/AdvancedJobSchedulerPervasive

โฮสต์: ชื่อโฮสต์ของ iSeries. โดเมน: โดเมนที่ iSeries ตั้งอยู่. พอร์ต: พอร์ตที่ instance ของเว็บเซิร์ฟเวอร์กำลังเฝ้าฟังอยู่. ค่าดีฟอลต์คือ 8210.

ถ้าต้องการระบุภาษาเฉพาะที่จะใช้, ให้ดูหัวข้อ การเลือกภาษา.

โทรศัพท์ที่พร้อมเชื่อมต่ออินเทอร์เน็ต และโครงร่างบราวเซอร์ของ PDA

ถ้าคุณเชื่อมต่อกับคุณลักษณะพิเศษ Advanced Job Scheduler for Wireless สำเร็จแล้วบน iSeries ของ คุณ, จอแสดงผลเริ่มต้นจะมีข้อมูลสรุปเกี่ยวกับโทรศัพท์ที่พร้อมเชื่อมต่ออินเทอร์เน็ต หรือ PDA. ข้อมูลสรุประบุถึงข้อมูลที่เป็นอยู่ในปัจจุบัน, มีงานที่จัดตารางเวลาอยู่จำนวนเท่าไร, มี activity entry อยู่จำนวนเท่าไร, และอ็อปชันการตรวจสอบสถานะของการมอนิเตอร์งานหรือการส่งข้อความไปยังผู้รับ. นอกจากนี้, ข้อสรุปยังมีสถานะโดยรวมทั้งหมดของ OK หรือ Attention ปรากฏบนส่วนบนสุดของจอแสดงผล. ถ้ามีการระบุ Attention, งานจะมีข้อความที่ต้องการความสนใจมากขึ้น. งานที่ต้องการความสนใจจะประกอบด้วยเครื่องหมายอุทธาน.


โครงร่างบราวเซอร์แบบดั้งเดิม

โครงร่างบราวเซอร์แบบดั้งเดิมจะเหมือนกันทุกประการกับจอแสดงผลของโทรศัพท์ที่พร้อมเชื่อมต่ออินเทอร์เน็ต และ PDA. อย่างไรก็ตาม, จำนวนของเนื้อหาจะน้อยกว่าขนาดของจอแสดงผล. ดังนั้น, คุณสามารถลดขนาดของเว็บบราวเซอร์ลงเพื่อจะเพิ่มเนื้อที่มากขึ้นในการทำงานกับแอ็พพลิเคชันอื่นๆ ในขณะที่ยังคงเปิด Advanced Job Scheduler for Wireless Web browser ได้อยู่. นอกจากนี้, ถ้าคุณกำลังใช้งานอินเทอร์เน็ตบราวเซอร์แบบเดิมบนเครื่องพีซีของคุณ, คุณสามารถเลือกใช้ Show all จากเมนูหลักของ Advanced Job Scheduler. แล้ว, คุณก็สามารถเรียกดูเนื้อหาที่มากขึ้นในหน้าเว็บเพจหน้าเดียว.

หลังจากที่คุณได้เชื่อมต่อกับระบบของคุณเสร็จเรียบร้อยแล้ว, คุณอาจต้องการปรับการเชื่อมต่อของคุณตามความต้องการ.

การปรับการเชื่อมต่อตามความต้องการของคุณ:

โดยใช้อุปกรณ์ไร้สายของคุณ, คุณสามารถปรับอินเทอร์เน็ตเฟสได้ตามความต้องการที่เฉพาะเจาะจงของคุณ. ตัวอย่างเช่น, คุณอาจต้องการดูงานเพียงบางงาน และระบุว่าไม่ต้องการดูชื่อกลุ่มของงาน. และคุณอาจไม่ต้องการเข้าถึงรายการของ activity ที่จัดตารางเวลา. เพจการปรับตามความต้องการบนอุปกรณ์ไร้สายของคุณอนุญาตให้คุณกรองงานต่างๆรวมทั้งการเปลี่ยนจอแสดงผลตามที่ชื่นชอบ.

คุณสามารถปรับการเชื่อมต่อของคุณตามความต้องการได้หลายวิธี ไม่ว่าคุณกำลัง ใช้ PC, PDA, หรือโทรศัพท์ที่พร้อมเชื่อมต่ออินเทอร์เน็ต. เพื่อใช้ประโยชน์จากคุณลักษณะพิเศษเหล่านี้, ให้อ่านเว็บไซต์ของผลิตภัณฑ์ Advanced Job Scheduler . 

การจัดการ Advanced Job Scheduler for Wireless:

ใช้อุปกรณ์ไร้สายของคุณในการทำงานกับ Advanced Job Scheduler. ในการใช้อุปกรณ์ไร้สาย คุณลักษณะพิเศษที่พร้อมใช้งานมีดังต่อไปนี้:

การดูงานที่แอ็คทีฟ, พักไว้, และงานที่คงค้าง

คุณสามารถดูรายการของงานปกติ (งาน Advanced Job Scheduler) หรืองาน Management Central ที่มีสถานะเป็นแอ็คทีฟ, พักไว้, หรือคงค้าง. คุณยังสามารถปรับการแสดงรายการของงานโดยให้เรียงลำดับตามชนิดของงาน, ชื่อ, หรือเวลา. ยิ่งไปกว่านั้น, คุณสามารถระบุได้ว่าจะให้โลบารีข้อมูลใดเก็บข้อมูลสำหรับ งานและ activities.

การดู job dependencies

คุณสามารถดู predecessor และ successor jobs สำหรับงานเฉพาะหนึ่งได้. Successor คืองานที่ต้องอาศัยงานหนึ่งหรือหลายงานอื่น (predecessors) ในการรัน. ในทางกลับกัน, successor job อาจเป็น predecessor job สำหรับ successor jobs อื่น.

การแสดงความข้อความ

ถ้างานมีข้อความที่รอกงานอยู่, คุณสามารถดูข้อความ และตอบกลับข้อความได้โดยใช้อุปกรณ์ไร้สายของคุณ.

การเริ่มงาน

คุณสามารถใช้อุปกรณ์ไร้สายของคุณเพื่อส่งงาน. อีพซันที่คุณสามารถใช้เมื่อส่งงานขึ้นอยู่กับอุปกรณ์ไร้สายที่คุณใช้.

การทำงานกับ Advanced Job Scheduler activity

คุณสามารถโต้ตอบกับ Advanced Job Scheduler activity จากอุปกรณ์ไร้สายได้. แต่ละ activity มีอีพซันที่แตกต่างกันขึ้นอยู่กับสถานะของ activity entry.


Internationalization

Advanced Job Scheduler for Wireless ใช้โค้ดประเทศและภาษาที่เชื่อมโยงกับ iSeries^(TM) Java^(TM) Virtual Machine ของคุณ เพื่อกำหนดภาษาและรูปแบบของวันที่/เวลาที่จะใช้บนอุปกรณ์ไร้สายของคุณ. ถ้าค่าดีฟอลต์ Java Virtual Machine ไม่ใช่โค้ดที่คุณต้องการใช้, คุณสามารถเปลี่ยนโค้ดได้โดยง่าย. ให้ดูคำอธิบายทางออนไลน์สำหรับรายละเอียดเพิ่มเติม.

ให้ดูคำอธิบายทางออนไลน์สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับการทำภารกิจที่ต้องการ.

การแก้ไขปัญหาเกี่ยวกับ Advanced Job Scheduler:

เมื่องานไม่รันตามเวลาที่จัดตารางเวลาไว้, ให้ค้นหาสิ่งที่คุณสามารถทำได้.

เพื่อแก้ไขปัญหาเกี่ยวกับ Advanced Job Scheduler, ขั้นแรกให้อ่านเว็บเพจ คำถามที่ถามบ่อยใน Job Scheduler . อ่านเกี่ยวกับคำถามที่ถามเป็นประจำเกี่ยวกับวิธีการใช้งานฟังก์ชันที่ต้องการด้วย Advanced Job Scheduler.

นอกจากนั้น, ต่อไปนี้คือรายการของไอเท็มที่คุณสามารถตรวจสอบได้ เมื่อมีงานที่ไม่รันตามเวลาที่ได้ถูกจัดตารางเวลาไว้:

โปรแกรมฟิซซ์ ปัจจุบัน

สิ่งแรกที่คุณควรตรวจสอบคือโปรแกรมฟิซซ์ของคุณควรจะเป็นตัวล่าสุด. เมื่อคุณทำการร้องขอโปรแกรมฟิซซ์, ขอให้มั่นใจว่าคุณได้ร้องขอรายชื่อของโปรแกรมฟิซซ์ทั้งหมด. โปรแกรมฟิซซ์ทั้งหมดอาจไม่ได้อยู่ใน cumulative packages.

ตรวจสอบการมอไนเตอร์งาน

- งาน QIJSSCD ควรจะแอนด์ที่พอย์ในระบบย่อย QSYSWRK. ถ้าไม่, ให้ใช้คำสั่ง Start Job Scheduler (STRJS).
- การมอไนเตอร์งานอาจวิ่งวนซ้ำถ้าสถานะของงานเป็น RUN นานเกินสิบนาที. ถ้าการมอไนเตอร์วนซ้ำ, ให้สั่งจบงานด้วย *IMMED, และเริ่มการมอไนเตอร์งานอีกครั้ง (STRJS).
- ถ้ามีข้อความที่จะต้องตอบ, ให้ตอบกลับด้วย C (Cancel). การมอไนเตอร์งานจะหน่วง 90 วินาทีแล้วจึงเริ่มการมอไนเตอร์อีกครั้ง. พิมพ์บันทึกการใช้งานของงานการมอไนเตอร์. บันทึกการใช้งานนี้จะมีข้อความแสดงความผิดพลาด.

การตรวจสอบไฟล์บันทึก Advanced Job Scheduler

ใช้คำสั่ง Display Log for Job Scheduler (DSPLOGJS) สำหรับงาน. กด F18 เพื่อไปที่จุดสิ้นสุดของรายการ. จะมี entry ให้เห็นเพื่ออธิบายว่าเพราะเหตุใดงานจึงไม่รัน. ตัวอย่างของ entry รวมไปถึงความล้มเหลวของรีซอร์ส, การแอนด์ที่พหรือสถานะการรันขึ้นอยู่กับงานอื่น, หรือข้อผิดพลาดในการส่ง.

การขึ้นอยู่กับงานอื่น

ถ้างานเป็นงานที่ขึ้นอยู่กับงานอื่น, ให้ใช้อีพซัน 10 จากจอแสดงผล Work with Jobs เพื่อที่จะแสดงการขึ้นอยู่กับงาน

อื่น. กต F8 เพื่อแสดงรายชื่อของงาน predecessor ทั้งหมด. งานที่ขึ้นอยู่กับงานอื่นไม่สามารถรันได้ ยกเว้นว่างาน predecessor ทั้งหมดจะแสดง * YES ในคอลัมน์ Complete .

การติดตามความก้าวหน้าของงาน

ถ้างานทำงานได้ไม่ถูกต้อง, คุณสามารถใช้คำสั่ง Set Step using Job Scheduler (SETSTPJS) ก่อนหน้าหรือหลังจากขั้นตอนใน โปรแกรม CL ของคุณ เพื่อกำหนดว่าปัญหาคืออะไร. ระบุคำสั่งพร้อมด้วยคำอธิบายในโปรแกรม CL ของคุณ. ใช้คำสั่งนี้ได้มาครั้งตามความจำเป็น. คำอธิบายที่เกี่ยวข้องกับคำสั่งปัจจุบัน แสดงอยู่ในฟิลด์ Command step บนหน้า Last Run ของคุณสมบัติงานที่ จัดตารางเวลา. นอกจากนี้, คุณสามารถดูฟิลด์ Command step ได้บนหน้าต่าง Status ของงานแก็คทีฟ. ฟิลด์ Command step จะถูกอัปเดตโดยอัตโนมัติในทุกครั้งที่ งานพบคำสั่ง SETSTPJS. ใช้คำสั่งนี้ เพื่อช่วยพิจารณาความก้าวหน้าของงาน.

การเก็บตัวอย่างข้อมูลเหล่านี้จะช่วยคุณในการวิเคราะห์ปัญหา:

เงื่อนไขของข้อความแสดงความผิดพลาด

พิมพ์บันทึกการใช้งานสำหรับเซสชันแบบโต้ตอบ, งานที่มอนิเตอร์หรืองานที่ถูกจัดตารางเวลาไว้, ขึ้นอยู่กับว่าข้อผิดพลาดเกิดขึ้นที่ใด.

วันที่ของตารางเวลางานไม่ถูกต้อง

ใช้คำสั่ง DSPJOBJS สำหรับงานพร้อมด้วย OUTPUT(*PRINT). พิมพ์รายงานปฏิทินถ้ามีการใช้ปฏิทินในงานนั้น. พิมพ์รายงานปฏิทินวันหยุดถ้ามีการใช้ปฏิทินวันหยุดในงานนั้น. กดปุ่ม Print เพื่อพิมพ์จอแสดงผลของแต่ละ fiscal calendar entry สำหรับปฏิทินปีบัญชีที่ใช้ภายในงาน.

ไฟล์บันทึก Advanced Job Scheduler

พิมพ์ไฟล์บันทึก Advanced Job Scheduler สำหรับช่วงเวลาที่ มีข้อสงสัยเสมอ.

ไฟล์ QALJSMST และ QALJSHST

ไฟล์ QALJSMST และไฟล์ QALJSHST ในไลบรารี QUSRIJS อาจจำเป็นต้องทำเจอร์นลไว้ก่อนที่จะพยายามจำลองปัญหาให้เกิดขึ้นอีก. และ, ไลบรารี QUSRIJS อาจเป็นที่ต้องการของฝ่ายสนับสนุนของ IBM .

การทำงานกับ job schedule entries

นอกเหนือจากหน้าต่าง Job Properties - Job Queue ของ iSeries Navigator, คุณยังสามารถเปลี่ยน job schedule entry ได้โดยตรงโดยใช้อินเตอร์เฟซแบบอักษร. ข้างล่างนี้เป็นรายการของภารกิจอินเตอร์เฟซ แบบอักษรทั่วไป ที่คุณสามารถใช้เมื่อทำงานกับ job schedule entries.

สำคัญ: ห้ามใช้ Work with Job Schedule Entries (WRKJOBSCDE) เพื่อเปลี่ยนหรือลบงานที่จัดตารางเวลาซึ่งถูกจัดตารางเวลาโดยใช้ Management Central Scheduler หรือ Advanced Job Scheduler. ถ้างานถูกเปลี่ยนหรือลบออก โดยใช้ WRKJOBSCDE, Management Central จะไม่ได้รับการแจ้งถึงการเปลี่ยนแปลง. ภารกิจไม่สามารถรันได้ตามที่คาด, และข้อความแสดงความผิดพลาดอาจปรากฏขึ้นในบันทึกการใช้งานเซิร์ฟเวอร์ Management Central.

การใส่เพิ่ม job schedule entry:

คำสั่ง Add Job Schedule Entry (ADDJOBSCDE) ช่วยคุณจัดตารางเวลางานแบ็ตช์โดยการใส่เพิ่ม entry ที่ตารางเวลางาน. คุณสามารถใช้คำสั่งนี้เพื่อจัดตารางเวลางานแบ็ตช์ให้ส่งพร้อมกันในคราวเดียว, หรือ เพื่อจัดตารางเวลางานแบ็ตช์ให้ส่งในช่วงเวลาที่สม่ำเสมอ.

คำสั่ง: Add Job Schedule Entry (ADDJOBSCDE)

ตัวอย่าง: คำสั่งนี้จะส่งงานที่มีชื่อว่า CLEANUP ทุกวันศุกร์ เวลา 11 p.m. งานจะใช้รายละเอียดของงาน CLNUPJOB ในไลบรารี CLNUPLIB. ถ้าระบบถูกปิดหรืออยู่ในสภาพที่ถูกจำกัดเมื่อเวลา 11 p.m. ของวันศุกร์, งานจะไม่ถูกส่งที่ IPL หรือเมื่อระบบฟื้นจากสภาพที่ถูกจำกัด.

```
ADDJOBSCDE JOB(CLEANUP) SCDDATE(*NONE)
            CMD(CALL PGM(CLNUPLIB/CLNUPPGM))
            SCDDAY(*FRI) SCDTIME('23:00:00')
            FRQ(*WEEKLY) RCYACN(*NOSBM)
            JOBD(CLNUPLIB/CLNUPJOB)
```

การเปลี่ยน job schedule entry:

คำสั่งนี้จะเปลี่ยน entry ในตารางเวลางาน, แต่ไม่ส่งผลต่อ งานใดๆ ที่ได้ส่งเรียบร้อยแล้วโดยใช้ entry นี้. ถ้าต้องการเปลี่ยน job entry, ให้ใช้อินเตอร์เฟซแบบอักษร.

เพื่อเปลี่ยน job schedule entry, คุณต้องมีสิทธิเหมือนกันกับสิทธิที่ต้องใช้ในการใส่เพิ่ม entry. อย่างไรก็ตาม, สิทธิในการใช้งานแต่ละอ็อบเจกต์จะถูกตรวจสอบเฉพาะเมื่อคุณกำลังเปลี่ยนพารามิเตอร์นั้นสำหรับ entry เท่านั้น. ยิ่งกว่านั้น, ถ้าคุณไม่มีสิทธิพิเศษ *JOBCTL, คุณสามารถเปลี่ยน ได้เฉพาะ entries ที่โปรไฟล์ผู้ใช้ของคุณใส่เพิ่มที่อ็อบเจกต์ตารางเวลางาน.

คำสั่ง: Change Job Schedule Entry (CHGJOBSCDE)

ตัวอย่าง: คำสั่งนี้จะเปลี่ยน job schedule entry BACKUP หมายเลข 001584 เพื่อให้งานของ entry ถูกส่งไปที่คิวงาน QBATCH ในไลบรารี QGPL.

```
CHGJOBSCDE JOB(BACKUP) ENRYNBR(001584) JOBQ(QGPL/QBATCH)
```

ตัวอย่าง: คำสั่งนี้จะเปลี่ยนตารางเวลา ของงานแบ็คอัพให้รันโปรแกรม A เมื่อเวลา 11 a.m. ของวันที่ 12/15/03 และในวันเดียวกันของทุกสัปดาห์.

```
CHGJOBSCDE JOB(EXAMPLE) ENRYNBR(*ONLY) CMD(CALL PGM(A))
            FRQ(*WEEKLY) SCDDATE(121503) SCDTIME(110000)
```

การพัก job schedule entry:

คำสั่ง Hold Job Schedule Entry (HLDJOBSCDE) ช่วยคุณพัก entry, entries ทั้งหมด, หรือเซตของ entries ในตารางเวลางาน. ถ้า entry ถูกพัก, จะไม่มีการส่งงานใดๆ ที่เวลาที่จัดตารางไว้. ถ้าต้องการพัก job schedule entry, ให้ใช้อินเตอร์เฟซแบบอักษร.

เพื่อพัก entries, คุณต้องมีสิทธิพิเศษ job control (*JOBCTL); ถ้าไม่มี คุณจะสามารถพักได้เฉพาะ entries ที่คุณใส่เพิ่มเท่านั้น. ถ้าคุณพัก job schedule entry:

- entry จะถูกพักจนกว่า entry จะถูกรีลีสโดยใช้คำสั่ง Release Job Schedule Entry (RLSJOBSCDE) หรือ Work with Job Schedule Entries (WRKJOBSCDE).
- งานจะไม่ถูกส่งเมื่อถูกรีลีส, แม้ว่าวันที่และเวลาที่งานถูก จัดตารางเวลาให้ส่งจะผ่านมาแล้วก็ตามในขณะที่ entry ถูกพักอยู่. แต่งานจะถูกส่งในวันที่ในอนาคตซึ่งจัดตารางเวลาไว้ให้ส่ง.

คำสั่ง: Hold Job Schedule Entry (HLDJOBSCDE)

ตัวอย่าง: ตัวอย่างต่อไปนี้จะพัก job schedule entry CLEANUP.

```
HLDJOBSCDE JOB(CLEANUP)
```

การพิมพ์รายการของ job scheduled entries:

ถ้าต้องการพิมพ์รายการของ job schedule entries, ให้ใช้อินเทอร์เฟซแบบ อักษร.

คำสั่ง: Work with Job Schedule Entries (WRKJOBSCDE)

ตัวอย่าง: ตัวอย่างต่อไปนี้จะพิมพ์รายการของ job schedule entries.

```
WRKJOBSCDE OUTPUT(*PRINT)
```

ตัวอย่าง: ตัวอย่างต่อไปนี้จะพิมพ์ข้อมูลรายละเอียดเกี่ยวกับแต่ละ job schedule entry.

```
WRKJOBSCDE OUTPUT(*PRINT) PRTFMT(*FULL)
```

การรีลีส job schedule entry:

คำสั่ง Release Job Schedule Entry (RLSJOBSCDE) ช่วยให้คุณสามารถรีลีส entry, entries ทั้งหมด, หรือชุดของ entries ในตารางเวลางาน. ถ้าคุณรีลีส job schedule entry, งานจะไม่ถูกส่งในทันที, แม้ว่าวันที่และเวลาซึ่งถูกจัดตารางเวลาให้ส่งได้ผ่านไปแล้วในขณะที่ entry ถูกพักไว้. ถ้าเวลาที่จัดตารางเวลาผ่านไปแล้วในขณะที่ entry ถูกพัก จะมีการส่งข้อความเตือนเพื่อระบุว่างานหนึ่งหรือหลายงาน ขาดไป. จากนั้นงานจะถูกส่งในวันที่ในอนาคตซึ่ง จัดตารางเวลาให้ส่ง. ถ้าต้องการรีลีส job schedule entries, ให้ใช้อินเทอร์เฟซแบบอักษร.

เพื่อรีลีส entries, คุณต้องมีสิทธิพิเศษ job control (*JOBCTL); ถ้าไม่มี คุณจะสามารกรีลีสได้เฉพาะ entries ที่คุณใส่เพิ่มเท่านั้น.

คำสั่ง: Release Job Schedule Entry (RLSJOBSCDE)

ตัวอย่าง: ตัวอย่างนี้จะรีลีส job schedule entries ทั้งหมดที่มีสถานะถูกพัก.

```
RLSJOBSCDE JOB(*ALL) ENRYNBR(*ALL)
```

การลบ job schedule entry:

คำสั่ง Remove Job Schedule Entry (RMVJOBSCDE) ช่วยให้คุณสามารถลบ entry, entries ทั้งหมด, หรือชุดของ entries ในตารางเวลางาน. แต่ละ job schedule entry เท่ากับงานแบ็คชหนึ่งงาน, และมีข้อมูล ที่จำเป็นสำหรับการรันงานอย่างอัตโนมัติหนึ่งครั้งหรือวันที่ช่วงเวลาม่าเสมอที่จัด ตารางเวลาไว้. จะมีการส่งข้อความให้คุณและ message queue ถูกระบุใน job schedule entry เมื่อลบ entry เสร็จเรียบร้อยแล้ว. ถ้าต้องการลบ job schedule entry, ให้ใช้อินเทอร์เฟซแบบอักษร.

เพื่อลบ entries, คุณต้องกำลังรันอยู่ภายใต้โปรไฟล์ผู้ใช้ซึ่งมีสิทธิพิเศษ job control (*JOBCTL); ถ้าไม่มี คุณจะสามารกลบได้เฉพาะ entries ที่คุณใส่เพิ่มเท่านั้น.

คำสั่ง: Remove Job Schedule Entry (RMVJOBSCDE)

ตัวอย่าง: ตัวอย่างต่อไปนี้จะลบงาน PAYROLL จาก ตารางเวลางาน.

```
RMVJOBSCDE JOB(PAYROLL) ENRYNBR(*ONLY)
```

เมื่องานระบบลบ single-submission entry หรือเมื่อ entry ถูกลบโดยคำสั่ง Remove Job Schedule Entry (RMVJOBSCDE), ข้อความระบบ CPC1239 จะถูกส่งไปที่ message queue ที่ระบุใน entry. ถ้า single-submission entry ถูกพักไว้เมื่อถึงเวลาที่จัด

ตารางเวลาของ entry และ entry ระบุ *NO สำหรับแอ็ททริบิวต์การบันทึก, entry จะถูกลบออก เมื่อ entry ถูกรีเซ็ตด้วยคำสั่ง Release Job Schedule Entry. ในกรณีนี้, ข้อความ CPC1245 จะถูกส่งไปที่ message queue ที่ระบุใน entry.

การจัดการระบบย่อย

เนื่องจากงานรันในระบบย่อย, คุณอาจต้องมอนิเตอร์ activity ระบบย่อยสำหรับปัญหาที่อาจเกิดขึ้นได้ซึ่งอาจส่งผลกระทบต่อความสามารถในการรันของงาน.

ระบบย่อยคือสถานที่ทำงานสำหรับงานบนเซิร์ฟเวอร์ iSeries. ชุดงานทั้งหมดของผู้ใช้ได้ทำผ่านทางชิ้นงานต่างๆ ที่รันอยู่ในระบบย่อย จึงเป็นสิ่งสำคัญที่จะต้อง มอนิเตอร์พื้นที่เหล่านี้หากมีการทำงานที่ช้าเกิดขึ้น. ใน iSeries Navigator, คุณสามารถดูงาน และคิวงานที่เชื่อมโยงกับระบบย่อย. นอกจากนี้, คุณยังมีฟังก์ชันเกี่ยวกับงานและคิวงานที่เหมือนกันกับพื้นที่อื่นที่แสดงงานและคิวงานด้วย.

ภารกิจระบบย่อยทั่วไป

ข้อมูลนี้อธิบายภารกิจทั่วไปที่คุณสามารถ ปฏิบัติบนระบบย่อย.

การดูแอ็ททริบิวต์ระบบย่อย:

ระบบย่อยมีแอ็ททริบิวต์ต่างๆ. แอ็ททริบิวต์เหล่านี้ให้ข้อมูลเกี่ยวกับ สถานะปัจจุบันของระบบย่อย, หรือเกี่ยวกับค่าที่ระบุใน subsystem description.

เมื่อคุณใช้ iSeries Navigator, สามารถดูแอ็ททริบิวต์ ต่อไปนี้ของระบบย่อยที่แอ็คทีฟ:

- **ระบบย่อย:** ชื่อของระบบย่อย, และไลบรารีที่มี subsystem description.
- **Description:** description ของระบบย่อย.
- **สถานะ:** สถานะปัจจุบันของระบบย่อย. คำอธิบายจะมีรายละเอียดของสถานะต่างๆ ที่ใช้ได้.
- **งานแอ็คทีฟ:** จำนวนของงานที่แอ็คทีฟในปัจจุบัน, กำลังรันหรือกำลังรอที่จะรัน ใดๆ หนึ่ง, ในระบบย่อย. จำนวนงานดังกล่าวไม่รวมงานระบบย่อย.
- **งานแอ็คทีฟสูงสุด:** จำนวนสูงสุดของงานที่สามารถแอ็คทีฟ, กำลังรันหรือกำลังรอที่จะรัน ใดๆ หนึ่ง, ในระบบย่อย.
- **งานระบบย่อย:** ชื่อของงานระบบย่อย, รวมถึงผู้ใช้และ หมายเลข

iSeries Navigator:

ถ้าต้องการดูแอ็ททริบิวต์ของระบบย่อย, ให้ทำตามขั้นตอนเหล่านี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → **server** → **Work Management** → **Subsystems** → **Active Subsystems**.
2. คลิกขวารระบบย่อยที่คุณต้องการดู, แล้วคลิก **Properties**.

อินเตอร์เฟซแบบอักษร:

ถ้าต้องการใช้อินเตอร์เฟซแบบอักษร, ให้พิมพ์คำสั่งต่อไปนี้:

คำสั่ง: Display Subsystem Description (DSPSBSD)

ตัวอย่าง: คำสั่งนี้จะแสดง subsystem description menu สำหรับระบบย่อย QBATCH.

การหยุดระบบย่อย:

คุณสามารถใช้ iSeries Navigator หรืออินเทอร์เฟซแบบอักษร เพื่อหยุดหนึ่งหรือหลายระบบย่อยที่แ็คทีฟ และระบุสิ่งที่จะเกิดขึ้นกับงานแ็คทีฟที่กำลังจะประมวลผล. ไม่มีงานหรือขั้นตอนการเราต์ใหม่ถูกเริ่มในระบบย่อย หลังจากทีระบบย่อยถูกหยุด.

เมื่อระบบย่อยถูกหยุด, คุณสามารถระบุสิ่งที่จะเกิดขึ้นกับงานแ็คทีฟที่กำลัง ฎระบบประมวลผลอยู่ได้. เช่น, คุณสามารถระบุให้งานทุกงานในระบบย่อยหยุดการทำงานทันที (Immediate), หรือระบุให้งานได้รับอนุญาตให้ทำต่อไปจนเสร็จสิ้นการประมวลผลก่อนทีระบบย่อยจะจบลง (Controlled).

สำคัญ: ขอแนะนำให้จบการทำงานของระบบย่อยด้วยอ็อปชัน Controlled ในทุกเมื่อที่เป็นไปได้. การจบลักษณะนี้จะช่วยให้งานแ็คทีฟจบการทำงานด้วยตัวของมันเอง. ใช้อ็อปชันนี้เพื่อให้มั่นใจว่างานเสร็จสิ้นการทำงานก่อนทีระบบย่อยจะจบการทำงาน. สิ่งนี้จะช่วยให้โปรแกรมที่กำลังรันอยู่ทำการ cleanup (กระบวนการสิ้นสุดงาน) ให้เรียบร้อย. การระบุค่า Immediate อาจก่อให้เกิดผลที่ไม่พึงปรารถนา, เช่น ข้อมูลถูกอัปเดตเพียงบางส่วน.

การหยุดมีอยู่สองชนิด.

Controlled (แนะนำ)

จบระบบย่อยในลักษณะที่ควบคุม. งานจะถูกจบในลักษณะที่ควบคุม ด้วย. สิ่งนี้ช่วยให้โปรแกรมที่กำลังรันอยู่ทำการ cleanup (กระบวนการสิ้นสุดงาน) ให้เรียบร้อย. เมื่องานที่กำลังจะจบมีโพรซีเจอร์การจัดการสัญญาณ สำหรับสัญญาณอะซิงโครนัส SIGTERM, จะมีการสร้างสัญญาณ SIGTERM สำหรับงานนั้น. แ็พพลิเคชันมีจำนวนเวลาตามที่ระบุสำหรับพารามิเตอร์ DELAY ในการทำ cleanup ให้เสร็จสมบูรณ์ก่อนทีงานจะถูกจบ.

Immediate

จบระบบย่อยในทันที. งานจะจบในทันทีด้วย. เมื่องานที่กำลังจะจบ มีโพรซีเจอร์การจัดการสัญญาณสำหรับสัญญาณอะซิงโครนัส SIGTERM, จะมีการสร้างสัญญาณ SIGTERM สำหรับงานนั้น และค่ากำหนดของระบบ QENDJOBLMT ระบุขีดจำกัดเวลา. นอกเหนือจากการจัดการสัญญาณ SIGTERM, โปรแกรมที่กำลังรันอยู่ไม่ได้รับอนุญาตให้ปฏิบัติการ cleanup ใดๆ.

iSeries Navigator:

ถ้าต้องการใช้ iSeries Navigator, ให้ทำดังต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย My Connections → server → Work Management → Subsystems → Active Subsystems.
2. คลิกขวาทีระบบย่อยหนึ่งหรือหลายระบบทีคุณต้องการหยุด, แล้วคลิก Stop.
3. ระบุอ็อปชันทีต้องการใช้เมื่อระบบย่อยถูกหยุด.
4. คลิก Stop.

อินเทอร์เฟซแบบอักษร:

ถ้าต้องการใช้อินเทอร์เฟซแบบอักษร, ให้พิมพ์คำสั่งต่อไปนี้:

คำสั่ง: End Subsystem (ENDSBS)

ตัวอย่าง: คำสั่งนี้จะจบงานแฉีกที่ทั้งหมดในระบบย่อย QBATCH และจบระบบย่อย. งานแฉีกที่มีเวลา 60 วินาทีในการปฏิบัติขั้นตอน การจบงานที่แฉีกพลีเคชันจัดให้.

```
ENDSBS SBS(QBATCH) OPTION(*CNTRLD) DELAY(60)
```

ใช้พารามิเตอร์ End Subsystem Option (ENDSBSOPT) เพื่อพัฒนาประสิทธิภาพสำหรับการจบระบบย่อย. ถ้าคุณระบุ ENDSBSOPT(*NOJOBLOG), ระบบย่อยจะจบ, แต่จะไม่มีการสร้างบันทึกการใช้งานสำหรับงานทุกงานที่อยู่ใน ระบบย่อย.

ถ้ามีปัญหาเกิดขึ้นในงาน, แต่คุณได้ระบุ *NOJOBLOG, การวินิจฉัยปัญหาอาจเป็นเรื่องยากหรือไม่สามารถทำได้ เนื่องจากปัญหาไม่ได้ ถูกบันทึกในบันทึกการใช้งาน. ถ้าคุณใช้แฉีกทริบิวต์งาน LOGOUTPUT(*PND) บันทึกการใช้งานจะถูกวางในสถานะคงค้าง, แต่ไม่ถูกบันทึก. อย่างไรก็ตาม, บันทึกการใช้งานยังคงพร้อมใช้งานถ้ามีความต้องการ. ดูหัวข้อที่เกี่ยวข้องกับ บันทึกการใช้งาน สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการคงค้างบันทึกการใช้งาน.

ถ้าคุณระบุ ENDSBSOPT(*CHGPTY *CHGTSL), ระดับความสำคัญการรันและการแบ่งเวลาจะเปลี่ยนสำหรับงานทั้งหมดที่จบในระบบย่อยนี้. งานจะเสร็จสมบูรณ์ช้าลงสำหรับ processor cycles และจะจบโดยมีผลกระทบต่องานที่กำลังรันอยู่ในระบบย่อยอื่นน้อยลง.

คุณสามารถระบุอ็อปชันทั้งหมดสามอ็อปชัน (*NOJOBLOG, *CHGPTY, และ *CHGTSL) บนพารามิเตอร์ ENDSBSOPT, เช่น:

```
ENDSBSOPT(*NOJOBLOG *CHGPTY *CHGTSL)
```

หมายเหตุ: ถ้า คุณระบุ *ALL สำหรับชื่อระบบย่อยและมีงานใดๆ ที่กำลังรันอยู่ภายใต้ QSYSWRK, คุณควรจะใช้ *CNTRLD เพื่อป้องกันไม่ให้ระบบย่อยจบอย่างผิดปกติ.

การเริ่มระบบย่อย:

คำสั่ง Start Subsystem (STRSBS) จะเริ่ม ระบบย่อยโดยใช้ subsystem description ที่ระบุในคำสั่ง. เมื่อระบบย่อย เริ่มต้น, ระบบจะจัดสรรรีซอร์สที่จำเป็นและพร้อมใช้งาน (หน่วยเก็บ, เวิร์กสแตชัน, และคิวงาน) ที่ถูกระบุใน subsystem description. คุณสามารถเริ่มระบบย่อยโดยใช้อินเตอร์เฟซ iSeries Navigator หรืออินเตอร์เฟซแบบอักขระ.

iSeries Navigator:

ถ้าต้องการเริ่มระบบย่อยโดยใช้ iSeries Navigator, ให้ทำตามต่อไปนี้:

1. ขยาย My Connections → server → Work Management.
2. คลิกขวา Subsystems และคลิก Start Subsystem.
3. ระบุ Name และ Library ของ ระบบย่อยที่จะเริ่มและคลิก OK.

อินเตอร์เฟซแบบอักขระ:

คำสั่ง: Start Subsystem (STRSBS)

ตัวอย่าง: คำสั่งนี้จะเริ่มระบบย่อยของผู้ใช้ที่เชื่อมโยงกับ TELLER subsystem description ในไลบรารี QGPL. ชื่อระบบย่อยคือ TELLER.

```
STRSBS SBS(QGPL/TELLER)
```

การสร้าง subsystem description

คุณสามารถสร้าง subsystem description ได้สองวิธี. คุณสามารถก๊อปปี้ subsystem description ที่มีอยู่และเปลี่ยนแปลง, หรือคุณสามารถสร้าง description ใหม่ทั้งหมด.

สองแนวทางที่คุณสามารถใช้มีดังต่อไปนี้:

1. ถ้าต้องการก๊อปปี้ subsystem description ที่มีอยู่, โดยใช้อินเตอร์เฟซแบบ อักขระ, ให้ทำดังต่อไปนี้:
 - a. Create a Duplicate Object (CRTDUPOBJ) ของ subsystem description ที่มีอยู่. (คุณยังสามารถใช้คำสั่ง Work with Objects (WRKOBJ) หรือ Work with Objects using Programming Development Manager (WRKOBJPDM) ได้ด้วย.)
 - b. เปลี่ยนก๊อปปี้ของ subsystem description เพื่อให้ก๊อปปี้ทำงาน ในลักษณะที่คุณต้องการ. ตัวอย่างเช่น, คุณจะต้องลบ job queue entry ออกเนื่องจาก entry ระบุคิวงานที่ระบบย่อยดั้งเดิม ใช้อยู่. จากนั้นคุณจะต้องสร้าง job queue entry ใหม่ที่ระบบพารามิเตอร์ซึ่งระบบย่อยใหม่จะใช้.

ให้จำว่าต้อง ตรวจสอบ autostart job entries, workstation entries, prestart job entries, และ communication entries, และตรวจสอบว่าไม่มีความขัดแย้ง ระหว่างระบบย่อยทั้งสองระบบ. ตัวอย่างเช่น, ต้องตรวจสอบว่า workstation entries ไม่ทำให้ระบบย่อยทั้งสองระบบจัดสรรอุปกรณ์แสดงผลเดียวกัน.
2. ถ้าต้องการสร้าง subsystem description ใหม่ทั้งหมด, ให้ใช้อินเตอร์เฟซ แบบอักขระและทำดังต่อไปนี้:
 - a. Create a Subsystem Description (CRTSBSD).
 - b. Create a Job Description (CRTJOB).
 - c. Create a Class (CRTCLS) สำหรับ Add Prestart Job Entry (ADDPJE) และ Add Routing Entry (ADDRTGE).
 - d. Add Work Entries ที่ subsystem description.
 - Add Workstation Entry (ADDWSE)
 - Add Job Queue Entry (ADDJOBQE)
 - Add Communications Entry (ADDCMNE)
 - Add Autostart Job Entry (ADDAJE)
 - Add Prestart Job Entry (ADDPJE)
 - e. Add Routing Entries (ADDRTGE) ที่ subsystem description.

การใส่เพิ่ม autostart job entries:

คุณใช้อินเตอร์เฟซแบบอักขระเพื่อใส่เพิ่ม autostart job entry. งานแบบ autostart จะเริ่มโดยอัตโนมัติเมื่อระบบย่อยที่เกี่ยวข้อง เริ่มต้นขึ้น. โดยทั่วไปงานเหล่านี้จะทำงานการกำหนดค่าเริ่มต้นที่เชื่อมโยงกับระบบย่อย. งานแบบ autostart ยังสามารถทำงานที่ซ้ำๆ หรือนำเสนอฟังก์ชัน การให้บริการศูนย์กลางสำหรับงานอื่นที่อยู่ในระบบย่อยเดียวกัน.

คำสั่ง: Add Autostart Job Entry (ADDAJE)

ตัวอย่าง: ตัวอย่างนี้จะใส่เพิ่ม autostart job entry ที่ description ของ ระบบย่อย ABC.

```
ADDAJE SBSD(USERLIB/ABC) JOB(START)
      JOBD(USERLIB/STARTJD)
```

หมายเหตุ: เพื่อให้การเปลี่ยนแปลงมีผลบังคับใช้, ต้องจบระบบย่อยที่แฉีกที่ฟ แล้วรีสทาร์ท.

การใส่เพิ่ม communications entries:

แต่ละ communication entry อธิบายอุปกรณ์สื่อสารหนึ่งหรือหลายอุปกรณ์, ชนิดของอุปกรณ์, หรือตำแหน่งรีโมตซึ่งระบบย่อยจะเริ่มงาน เมื่อได้รับ คำขอการเริ่มโปรแกรม. ระบบย่อยสามารถจัดสรรอุปกรณ์สื่อสาร ถ้าอุปกรณ์ไม่ได้ถูกจัดสรรให้กับระบบย่อยหรืองานอื่นอยู่ในปัจจุบัน. อุปกรณ์สื่อสารที่ถูกจัดสรรแล้วในปัจจุบันสามารถถูกจัดสรรคืนในภายหลังได้, เพื่อให้อุปกรณ์นั้นพร้อมใช้งานสำหรับระบบย่อยอื่นได้. ถ้าต้องการใส่เพิ่ม communications entry ที่ subsystem description, ให้ใช้อินเตอร์เฟซแบบอักษร.

คำสั่ง: Add Communications Entry (ADDCMNE)

ตัวอย่าง: ตัวอย่างนี้จะใส่เพิ่ม communications entry สำหรับอุปกรณ์ APPC ที่มีชื่อว่า COMDEV และโหมด *ANY ไปที่ subsystem description SBS1, ซึ่งตั้งอยู่ในไลบรารี ALIB. พารามิเตอร์ DFTUSR มีค่าดีฟอลต์เป็น *NONE, ซึ่งหมายความว่าไม่สามารถป้อนงานใดเข้าในระบบผ่านทาง entry นี้ ยกเว้นว่าจะมีการให้ข้อมูลการรักษาความปลอดภัยที่ถูกต้องบนคำขอการเริ่มโปรแกรม.

```
ADDCMNE  SBS1(ALIB/SBS1)  DEV(COMDEV)
```

หมายเหตุ: คุณต้องระบุพารามิเตอร์ DEV หรือพารามิเตอร์ RMTLOCNAME อย่างใดอย่างหนึ่ง, แต่ไม่ใช่ทั้งสองตัว.

การใส่เพิ่ม job queue entries:

job queue entry ระบุคิวงานซึ่งงานจะถูกเลือกสำหรับการรันในระบบย่อย. งานที่เริ่มต้นจากคิวงานคืองานแบ็ตช์. คุณสามารถใส่เพิ่ม job queue entry โดยใช้อินเตอร์เฟซแบบอักษร.

คุณสามารถระบุไอเท็มดังต่อไปนี้ใน job queue entry.

- ชื่อคิวงาน (JOBQ)
- จำนวนสูงสุดของงานที่สามารถแอ็คทีฟในเวลาเดียวกันจากคิวงาน (MAXACT)
- ลำดับของระบบย่อยในการเลือกคิวงานซึ่งมีงานที่สามารถถูกเริ่มต้นได้ (SEQNBR)
- จำนวนสูงสุดของงานที่สามารถแอ็คทีฟในเวลาเดียวกันสำหรับระดับความสำคัญ ของคิวงานที่ระบุ (MAXPTYn)

คำสั่ง: Add Job Queue Entry (ADDJOBQE)

ตัวอย่าง: คำสั่งนี้จะใส่เพิ่ม job queue entry สำหรับคิวงาน NIGHT (ในไลบรารี QGPL) ที่ NIGHTSBS subsystem description ซึ่งตั้งอยู่ในไลบรารี QGPL. entry ระบุว่าสามารถมีงานแบ็ตช์จากคิวงาน NIGHT ที่แอ็คทีฟในเวลาเดียวกันได้มากถึงสามงานในระบบย่อย. สมมุติว่า หมายเลขลำดับดีฟอลต์คือ 10.

```
ADDJOBQE  SBS1(QGPL/NIGHTSBS)  JOBQ(QGPL/NIGHT)  MAXACT(3)
```

การใส่เพิ่ม prestart job entries:

Prestart job entries ระบุงานแบบ prestart ที่สามารถเริ่มต้น เมื่อระบบย่อยเริ่มต้นหรือเมื่อป้อนคำสั่ง Start Prestart Jobs (STRPJ). คุณสามารถใส่เพิ่ม prestart job entries ที่ subsystem description โดยใช้อินเตอร์เฟซแบบอักษร.

คำสั่ง: Add Prestart Job Entry (ADDPJE)

ตัวอย่าง: ตัวอย่างต่อไปนี้จะใส่เพิ่ม prestart job entry ที่ subsystem description ABC.

ADDPJE SBSDB(USERLIB/ABC) PGM(START)
JOBDB(USERLIB/STARTPJ)

การใส่เพิ่ม routing entries:

แต่ละ routing entry ระบุพารามิเตอร์ที่ใช้ในการเริ่มขั้นตอนการเรอต์สำหรับงาน. Routing entries ระบุพูลระบบย่อยของแหล่งเก็บข้อมูลหลักที่จะใช้, โปรแกรมการควบคุมที่จะรัน (โดยปกติคือ โปรแกรมที่ระบบจัดให้ QCMD), และข้อมูลรันไทม์เพิ่มเติม (จัดเก็บอยู่ในคลาสอ็อบเจกต์). ถ้าต้องการใส่เพิ่ม routing entry ที่ subsystem description, ให้ใช้อินเตอร์เฟสแบบอักษร.

คำสั่ง: Add Routing Entry (ADDRTGE)

ตัวอย่าง: คำสั่งนี้จะใส่เพิ่ม routing entry 46 ที่ subsystem description PERT ในไลบรารี ORDLIB. เพื่อใช้ routing entry 46, ข้อมูลการเรอต์ ต้องเริ่มด้วยสตริงอักษร WRKSTN2 ที่เริ่มต้นในตำแหน่ง 1. ขั้นตอนการเรอต์สามารถแก้ไขที่ใดก็ได้ไม่จำกัดจำนวนผ่านทาง entry นี้ได้ตลอดเวลา. โปรแกรม GRAPHIT ในไลบรารี ORDLIB จะรันในพูลของหน่วยความจำ 2 โดยใช้คลาส AZERO ในไลบรารี MYLIB.

```
ADDRTGE SBSDB(ORDLIB/PERT) SEQNBR(46) CMPVAL(WRKSTN2)  
PGM(ORDLIB/GRAPHIT) CLS(MYLIB/AZERO) MAXACT(*NOMAX)  
POOLID(2)
```

การใส่เพิ่ม workstation entries:

workstation entry จะถูกใช้เมื่องานเริ่มต้น เมื่อผู้ใช้ signs on หรือถ่ายโอนงานแบบโต้ตอบจากระบบย่อยอื่น. คุณสามารถระบุไอเท็มดังต่อไปนี้ใน workstation entry. ชื่อพารามิเตอร์แสดงอยู่ในวงเล็บ. ใช้อินเตอร์เฟสแบบอักษรเพื่อใส่เพิ่ม workstation entries.

- ชื่อหรือชนิดของเวิร์กสเตชัน (WRKSTN หรือ WRKSTNTYPE)
- ชื่อรายละเอียดของงาน (JOBDB) หรือชื่อรายละเอียดของงาน ในโปรไฟล์ผู้ใช้
- จำนวนสูงสุดของงานที่สามารถแก้ไขที่ใดก็ได้ในเวลาเดียวกันผ่านทาง entry (MAXACT)
- เวลาที่เวิร์กสเตชันจะถูกจัดสรร, นั่นคือเมื่อระบบย่อย ถูกเริ่มหรือเมื่องานแบบโต้ตอบเข้าสู่ระบบย่อยผ่านทาง คำสั่ง Transfer Job (TFRJOB) และพารามิเตอร์ AT อย่างใดอย่างหนึ่ง.

ถ้าต้องการใส่เพิ่ม workstation entry ที่ subsystem description, ให้ใช้อินเตอร์เฟสแบบอักษร.

คำสั่ง: Add Workstation Entry (ADDWSE)

ตัวอย่าง: ตัวอย่างต่อไปนี้จะใส่เพิ่ม workstation entry DSP12 ที่ระบบย่อย ABC.

```
ADDWSE SBSDB(USERLIB/ABC) WRKSTN(DSP12)  
JOBDB(USERLIB/WSE)
```

การสร้างไฟล์แสดงผล signon.:

ไฟล์แสดงผล signon ใช้ในการแสดงจอแสดงผล signon ที่เวิร์กสเตชัน ซึ่งถูกจัดสรรให้กับระบบย่อย. สามารถเปลี่ยนไฟล์แสดงผล signon เมื่อระบบย่อยแก้ไขที่ใดก็ได้. อย่างไรก็ตาม, จะไม่ใช่ไฟล์แสดงผล signon ใหม่ จนกว่าระบบย่อยจะเริ่มต้นในครั้งถัดไป. ถ้าต้องการสร้างไฟล์แสดงผล signon, ให้ใช้อินเตอร์เฟสแบบอักษร.

สามารถสร้างไฟล์แสดงผล signon ใหม่โดยใช้ไฟล์แสดงผล signon ที่ส่งมาจาก IBM เป็นจุดเริ่มต้น. ซอร์สสำหรับไฟล์แสดงผลนี้ตั้งอยู่ใน โลบรารี QGPL ใน source physical file QDDSSRC. ขอแนะนำให้นำไปให้คุณสร้าง source physical file ใหม่และก๊อปปี้ไฟล์แสดงผลที่ได้มาจาก IBM ไปที่ source physical file ใหม่ก่อนที่จะทำการเปลี่ยนแปลงใดๆ. ด้วยวิธีนี้, ซอร์สที่ได้มาจาก IBM ยังคงพร้อมใช้งานได้.

ข้อควรพิจารณา:

- ลำดับในการประกาศฟิลด์ในไฟล์แสดงผล signon ต้องไม่มีการเปลี่ยนแปลง. แต่ตำแหน่งในการแสดงผลฟิลด์บนจอแสดงผลสามารถเปลี่ยนได้.
- ห้ามเปลี่ยนขนาดทั้งหมดของอินพุตบัพเฟอร์หรือเอาต์พุตบัพเฟอร์. อาจเกิดปัญหาที่ร้ายแรงได้ ถ้ามีการเปลี่ยนลำดับหรือขนาดของบัพเฟอร์.
- ห้ามใช้ฟังก์ชันคำอธิบาย data descriptions specifications (DDS) ใน ไฟล์แสดงผล signon.
- ระบุ 256 เสมอบนพารามิเตอร์ MAXDEV .
- ไม่สามารถระบุคีย์เวิร์ด MENUBAR และ PULLDOWN ใน signon display file description.
- ความยาวบัพเฟอร์สำหรับไฟล์แสดงผลต้องเป็น 318. ถ้าความยาวน้อยกว่า 318, ระบบย่อยจะใช้จอแสดงผล signon ดีฟอลต์, QDSIGNON ในโลบรารี QSYS.
- ไม่สามารถลบบรรทัดลิสทิง.
- เมมเบอร์ QDSIGNON คือไฟล์แสดงผล signon ที่ได้มาจาก IBM ซึ่งใช้รหัสผ่านที่เป็นอักขระ 10 ตัว.
- เมมเบอร์ QDSIGNON2 คือไฟล์แสดงผล signon ที่ได้มาจาก IBM ซึ่งใช้รหัสผ่านที่เป็นอักขระ 128 ตัว.

คำสั่ง: Create Display File (CRTDSPF)

ฟิลด์แบบ hidden ในไฟล์แสดงผลที่มีชื่อว่า UBUFFER สามารถเปลี่ยนเป็น ฟิลด์จัดการที่เล็กลงได้. UBUFFER มีความยาว 128 ไบต์และถูกกำหนดเป็นฟิลด์สุดท้าย ในไฟล์แสดงผล. สามารถเปลี่ยนฟังก์ชันของฟิลด์นี้เป็นอินพุต/เอาต์พุตบัพเฟอร์ เพื่อให้ข้อมูลที่ระบุในฟิลด์นี้จะพร้อมใช้งานที่แอสพลีเคชันโปรแกรม เมื่อเริ่มงานแบบโต้ตอบ. คุณสามารถเปลี่ยนฟิลด์ UBUFFER ให้มี ฟิลด์ขนาดเล็กได้มากตามที่คุณต้องการ ถ้าอยู่ภายใต้ข้อกำหนดดังต่อไปนี้:

- ฟิลด์ใหม่ต้องตามหลังฟิลด์อื่นทั้งหมดในไฟล์แสดงผล. ตำแหน่งของฟิลด์ บนจอแสดงผลไม่ใช่สิ่งสำคัญ トラบเท่าที่ลำดับที่ฟิลด์ถูกวาง ใน data description specifications (DDS) ยังคงเป็นไปตามข้อกำหนดนี้.
- ความยาวทั้งหมดต้องเป็น 128. ถ้าความยาวของฟิลด์ต่างๆ มากกว่า 128, บางข้อมูลจะไม่ถูกส่งผ่าน.
- ฟิลด์ทั้งหมดต้องเป็นฟิลด์อินพุต/เอาต์พุต (ชนิด B ในซอร์ส DDS) หรือฟิลด์แบบ hidden (ชนิด H ในซอร์ส DDS).

การระบุจอแสดงผล signon ใหม่:

ระบบย่อยใช้ไฟล์แสดงผล signon ที่ถูกระบุในพารามิเตอร์ SGNDSPF ของ subsystem description ในการสร้างจอแสดงผล signon ที่เวิร์กสเตชันผู้ใช้. ถ้าต้องการเปลี่ยนไฟล์แสดงผล signon จากค่าดีฟอลต์ (QDSIGNON) เป็นค่าที่คุณสร้างขึ้น, ให้ใช้อินเตอร์เฟซแบบอักขระ.

หมายเหตุ: ใช้เวอร์ชันทดสอบของระบบย่อยในการตรวจสอบว่า จอแสดงผล ถูกต้อง ก่อนที่จะพยายามเปลี่ยนระบบย่อยการควบคุม.

คำสั่ง: Change Subsystem Description (CHGSBSD)

ระบุไฟล์แสดงผลใหม่บนพารามิเตอร์ SGNDSPF .

ตัวอย่าง: คำสั่งข้างล่างนี้จะเปลี่ยนไฟล์แสดงผล signon สำหรับ ระบบย่อย QBATCH จากค่าดีฟอลต์เป็นไฟล์ใหม่ที่มีชื่อว่า MYSIGNON.

```
CHGSBSD SBSD(QSYS/QBATCH) SGNDSPF(MYSIGNON)
```

การเปลี่ยน subsystem description

คำสั่ง Change Subsystem Description (CHGSBSD) จะเปลี่ยนแอตทริบิวต์การดำเนินการของ subsystem description ที่ระบุ. คุณสามารถเปลี่ยน subsystem description ในขณะที่ระบบย่อยแอคทีฟ. ถ้าต้องการเปลี่ยน subsystem description, ให้ใช้อินเตอร์เฟซแบบอักษร.

หมายเหตุ: คุณไม่สามารถระบุค่า *RMV บนพารามิเตอร์ POOLS ในขณะที่ระบบย่อยแอคทีฟ, เนื่องจากงานอาจถูกหยุดชั่วคราว.

คำสั่ง: Change Subsystem Description (CHGSBSD)

ตัวอย่าง: คำสั่งนี้จะเปลี่ยน definition ของพูลของหน่วยความจำ 2 ที่ระบบย่อย PAYCTL ใช้เป็นหน่วยความจำขนาด 1500K และ activity level เป็น 3. ไฟล์แสดงผล signon จะถูกเปลี่ยนเป็นไฟล์แสดงผล COMPANYYA และตั้งอยู่ในไลบรารี QGPL. ถ้าระบบย่อยแอคทีฟเมื่อใช้คำสั่งนี้, จะไม่มีการใช้ COMPANYYA จนกว่าระบบย่อยจะเริ่มในครั้งถัดไป.

```
CHGSBSD SBSD(QGPL/PAYCTL) POOLS((2 1500 3))
SGNDSPF(QGPL/COMPANYA)
```

การเปลี่ยน autostart job entries:

คุณสามารถระบุรายละเอียดของงานอื่นสำหรับ autostart job entry ที่ได้กำหนดไว้ก่อนหน้านี้. ถ้าต้องการเปลี่ยน autostart job entry, ให้ใช้อินเตอร์เฟซแบบอักษร

คำสั่ง: Change Autostart Job Entry (CHGAJE)

ตัวอย่าง: ตัวอย่างต่อไปนี้จะเปลี่ยนรายละเอียดของงานที่ใช้โดย autostart job entry ที่มีชื่อว่า START, ในระบบย่อย ABC, ในไลบรารี USERLIB.

```
CHGAJE SBSD(USERLIB/ABC) JOB(START)
JOBDB(USERLIB/NEWJD)
```

หมายเหตุ: เพื่อให้การเปลี่ยนแปลงมีผลบังคับใช้, ต้องจบระบบย่อยที่แอคทีฟแล้วรีสตาร์ท.

การเปลี่ยน communication entries:

คุณสามารถเปลี่ยนแอตทริบิวต์ของ communications entry ที่มีอยู่ใน subsystem description ที่มีอยู่ได้โดยใช้อินเตอร์เฟซแบบอักษร.

- เมื่อเปลี่ยนพารามิเตอร์ Job description (JOBDB) หรือ Default user profile (DFTUSR), communications entry จะเปลี่ยนไปด้วย; อย่างไรก็ตาม, ค่าของพารามิเตอร์เหล่านี้จะไม่เปลี่ยนสำหรับ งานใดๆ ที่แอคทีฟในขณะนั้น.
- ถ้าค่าของพารามิเตอร์ Maximum active jobs (MAXACT) ลดลงเป็นจำนวนที่น้อยกว่าจำนวนทั้งหมดของงานที่แอคทีฟผ่านทาง communications entry, จะไม่มีการประมวลผลคำขอการเริ่มโปรแกรมใหม่. งานแอคทีฟจะรันต่อไป; แต่ไม่มีการประมวลผลคำขอการเริ่มโปรแกรมเพิ่มเติม จนกว่าจำนวนของงานแอคทีฟจะน้อยกว่าค่าที่ระบุสำหรับ พารามิเตอร์ MAXACT.

คำสั่ง: Change Communications Entry (CHGCMNE)

ตัวอย่าง: ตัวอย่างนี้จะเปลี่ยน communications entry (ใน subsystem description QGPL/BAKER) สำหรับอุปกรณ์ A12 และ โหมด *ANY. activity level สูงสุดจะถูกเปลี่ยนเป็น *NOMAX ซึ่งหมายความว่า communications entry ไม่มีข้อจำกัดเกี่ยวกับจำนวนของคำขอการเริ่มโปรแกรมที่สามารถแอ็คทีฟได้พร้อมกัน. อย่างไรก็ตาม, ค่า MAXJOBS ใน subsystem description BAKER จะจำกัดจำนวนทั้งหมดของงานที่สามารถแอ็คทีฟได้ในระบบย่อย. จำนวนนี้รวมถึงงานที่สร้างโดยคำขอการเริ่มโปรแกรม. และยังมีขีดจำกัด ที่ผู้ใช้สามารถระบุจำนวนของงานแอ็คทีฟ ซึ่งสามารถเรอต์ผ่านทาง routing entry (MAXACT). ขีดจำกัดที่ระบุใน routing entry สามารถควบคุมจำนวนของงานที่ไหล หรือ ระดับการเรียกซ้ำของโปรแกรม. ในทุกกรณี, งานที่เป็นผลมาจากการประมวลผลคำขอการ เริ่มโปรแกรมไม่สามารถมีมากเกินไปจำนวนขีดจำกัดเหล่านี้.

```
CHGCMNE SBSD(QGPL/BAKER) DEV(A12) MAXACT(*NOMAX)
```

การเปลี่ยน job queue entries:

คุณสามารถเปลี่ยน job queue entry ที่มีอยู่ใน subsystem description ที่ระบุได้. สามารถใช้คำสั่งนี้ในขณะที่ระบบย่อยแอ็คทีฟ หรือ inactive. ถ้าต้องการเปลี่ยน job queue entry ในระบบย่อย, ให้ใช้อินเตอร์เฟซแบบอักษร.

คำสั่ง: Change Job Queue Entry (CHGJOBQE)

ตัวอย่าง: คำสั่งนี้จะเปลี่ยนจำนวนสูงสุดของงานที่สามารถ แอ็คทีฟได้พร้อมกันจากคิวงาน QBATCH ในไลบรารี QGPL. หมายเลขลำดับของ job queue entry จะไม่เปลี่ยนแปลง. สามารถมีงานจากคิวงาน QBATCH ที่แอ็คทีฟพร้อมกันได้มากถึงสี่งาน. งานจากระดับความสำคัญ 1 สามารถ แอ็คทีฟได้หนึ่งงาน. ไม่มีข้อจำกัดจำนวนสูงสุดของงานที่สามารถแอ็คทีฟ พร้อมกันจากระดับความสำคัญ 2. ระดับความสำคัญ 3 ถึง 9 ไม่เปลี่ยนแปลง.

```
CHGJOBQE SBSD(QGPL/QBATCH) JOBQ(QGPL/QBATCH) MAXACT(4)
MAXPTY1(1) MAXPTY2(*NOMAX)
```

การเปลี่ยน prestart entries:

คุณสามารถเปลี่ยน prestart job entry ใน subsystem description ที่ระบุได้. ระบบย่อยอาจแอ็คทีฟอยู่เมื่อ prestart job entry ถูกเปลี่ยน. การทำการเปลี่ยนแปลงที่ entry เมื่อระบบย่อยแอ็คทีฟจะมีผลต่อการดำเนิน การในอนาคต. งานแบบ prestart ใหม่ใดๆ ที่เริ่มขึ้นหลังจากใช้คำสั่ง จะใช้ ค่าที่เกี่ยวข้องกับงานใหม่. คำสั่งนี้ระบุงานแบบ prestart ที่เริ่มขึ้นเมื่อ ระบบย่อยเริ่มต้น หรือเมื่อมีการใช้คำสั่ง Start Prestart Jobs (STRPJ).

ถ้าต้องการเปลี่ยน prestart job entry ของ subsystem description, ให้ใช้อินเตอร์เฟซแบบอักษร.

คำสั่ง: Change Prestart Job Entry (CHGPJE)

ตัวอย่าง: ตัวอย่างนี้จะเปลี่ยน prestart job entry สำหรับโปรแกรม PGM1 ในไลบรารี QGPL ใน PJSBS subsystem description ที่มีอยู่ใน ไลบรารี QGPL. งานแบบ prestart ที่เกี่ยวข้องกับ entry นี้จะไม่เริ่มขึ้น ในครั้งถัดไปที่เริ่มขึ้นเมื่อ PJSBS subsystem description ในไลบรารี QGPL ถูกเริ่มขึ้น. แต่ต้องใช้คำสั่ง STRPJ ในการเริ่มงานแบบ prestart . เมื่อต้องเริ่มงานมากขึ้น, จะต้องเริ่มงานเพิ่มเติมอีกหนึ่งงาน.

```
CHGPJE SBSD(QGPL/PJSBS) PGM(QGPL/PGM1) STRJOBS(*NO)
THRESHOLD(1) ADLJOBS(1)
```

การเปลี่ยน routing entries:

คุณสามารถเปลี่ยน routing entry ใน subsystem description ที่ระบุได้โดยใช้อินเตอร์เฟซแบบอักษร. routing entry ระบุพารามิเตอร์ที่จะใช้ในการเริ่มขั้นตอนการเรตสำหรับงานหนึ่ง. ระบบย่อยที่เกี่ยวข้องสามารถแอ็คทีฟ เมื่อทำการเปลี่ยนแปลง.

คำสั่ง: Change Routing Entry (CHGRTGE)

ตัวอย่าง: คำสั่งนี้จะเปลี่ยน routing entry 1478 ใน subsystem description ORDER ที่พบในไลบรารี LIB5. ใช้โปรแกรมเดียวกัน, แต่ขณะนี้ จะรันในพูลของหน่วยความจำ 3 โดยใช้คลาส SOFAST ในไลบรารี LIB6.

```
CHGRTGE SBSD(LIB5/ORDER) SEQNBR(1478) CLS(LIB6/SOFAST) POOLID(3)
```

การเปลี่ยน workstation entries:

คุณสามารถระบุรายละเอียดของงานอื่นสำหรับ workstation entry ที่ได้กำหนดไว้ก่อนหน้านี้ได้โดยใช้อินเตอร์เฟซแบบอักษร.

- เมื่อมีการระบุพารามิเตอร์ Job description (JOBDD), workstation entry จะถูกเปลี่ยนแปลง; อย่างไรก็ตาม, ค่าของพารามิเตอร์นี้จะไม่เปลี่ยนสำหรับงานใดๆ ที่เริ่มต้นผ่านทาง entry นี้ซึ่งแอ็คทีฟอยู่ในขณะนั้น.
- ถ้าค่าของพารามิเตอร์ Maximum active jobs (MAXACT) ลดลงเป็นจำนวนที่น้อยกว่าจำนวนทั้งหมดของเวิร์กสเตชันที่แอ็คทีฟผ่านทาง workstation entry, จะไม่มีเวิร์กสเตชันเพิ่มเติมใดได้รับอนุญาตให้ sign on. เวิร์กสเตชันที่แอ็คทีฟจะไม่ถูก signed off. สามารถสร้างงานเพิ่มเติมสำหรับเวิร์กสเตชันที่แอ็คทีฟได้โดยใช้คำสั่ง Transfer Secondary Job (TFRSECJOB) หรือคำสั่ง Transfer to Group Job (TFRGRPJOB). เวิร์กสเตชันอื่นจะไม่ได้รับอนุญาตให้ sign on จนกว่าจำนวนของ เวิร์กสเตชันที่แอ็คทีฟจะน้อยกว่าค่าที่ระบุสำหรับพารามิเตอร์ MAXACT.

คำสั่ง: Change Workstation Entry (CHGWSE)

ตัวอย่าง: คำสั่งนี้จะเปลี่ยน workstation entry สำหรับเวิร์กสเตชัน A12 ในระบบย่อย BAKER ที่พบในไลบรารีเอนกประสงค์. งานจะถูกสร้างขึ้นสำหรับ เวิร์กสเตชัน A12 เมื่อมีการป้อนรหัสผ่านของผู้ใช้บนพร้อมท์ signon และกดปุ่ม Enter.

```
CHGWSE SBSD(QGPL/BAKER) WRKSTN(A12) AT(*SIGNON)
```

การเปลี่ยนจอแสดงผล signon:

ระบบของคุณถูกจัดส่งมาพร้อมกับไฟล์แสดงผล signon ดีฟอลต์ QDSIGNON, ซึ่งตั้งอยู่ในไลบรารี QSYS. ในสภาพที่คุณมีสถานะแวดล้อม หลายภาษา, คุณอาจต้องการเปลี่ยนสิ่งที่แสดงบนจอภาพ signon. หรือ, คุณอาจต้องการใส่เพิ่มข้อมูลบริษัทของคุณลงบนจอภาพ signon. ในสถานการณ์ดังกล่าว สิ่งแรกที่ต้องทำคือคุณต้องสร้างไฟล์แสดงผลใหม่. ถ้าต้องการทำเช่นนั้น, ให้ใช้อินเตอร์เฟซแบบอักษร.

แอ็ททริบิวต์ SGNDSPF ใน subsystem description ซ้ำไปที่ไฟล์แสดงผล signon ที่ผู้ใช้จะเห็นเมื่อ signing on ที่ระบบย่อย.

ขั้นตอนที่ใช้ในการเปลี่ยนจอแสดงผล signon สรุปได้ดังต่อไปนี้:

1. สร้างไฟล์แสดงผล signon ใหม่.
2. เปลี่ยน subsystem description เพื่อใช้ไฟล์แสดงผลที่เปลี่ยนแปลง แทนค่าดีฟอลต์ของระบบ.
3. ทดสอบการเปลี่ยนแปลง.

การลบ subsystem description

คำสั่ง Delete Subsystem Description (DLTSBSD) จะลบ subsystem descriptions (รวมถึง work entries หรือ routing entries ใดๆ ที่ใส่เพิ่ม) ที่ระบุออกจากระบบ. คิวงานที่ได้กำหนดที่ระบบย่อยนี้โดยคำสั่ง Add Job Queue Entry (ADDJOBQE) จะไม่ถูกลบออก. ในข้อเท็จจริง, เมื่อคุณลบ subsystem description (SBSD), จะไม่มีการลบอ็อบเจ็กต์ใดๆ ที่ SBSD อ้างอิง.

ระบบย่อยที่เกี่ยวข้องต้อง inactive ก่อนจึงจะสามารถลบออกได้. ใช้อินเตอร์เฟซแบบอักษรในการลบระบบย่อย.

คำสั่ง: Delete Subsystem Description (DLTSBSD)

คำสั่งนี้จะลบ inactive subsystem description ที่ชื่อว่า BAKER ออกจากไลบรารี LIB1.

```
DLTSBSD  SBSDB(LIB1/BAKER)
```

การลบ autostart job entries:

คุณสามารถลบ autostart job entry ออกจาก subsystem description โดยใช้อินเตอร์เฟซแบบอักษร.

คำสั่ง: Remove Autostart Job Entry (RMVAJE)

ตัวอย่าง: ตัวอย่างต่อไปนี้จะลบ autostart entry สำหรับงาน START ออกจาก subsystem description ABC.

```
RMVAJE  SBSDB(USERLIB/ABC)  JOB(START)
```

หมายเหตุ: เพื่อให้การเปลี่ยนแปลงมีผลบังคับใช้ ต้องจบบระบบย่อยที่แ็คทีฟ แล้วรีสตาร์ท.

การลบ communication entries:

คุณสามารถลบ communication entries ออกจาก subsystem description โดยใช้อินเตอร์เฟซแบบอักษร. งานทั้งหมดที่แ็คทีฟผ่านทาง communications entry ที่จะลบออกต้องถูกจบลงก่อน คำสั่งนี้จึงจะสามารถ รันได้.

คำสั่ง: Remove Communications Entry (RMVCMNE)

ตัวอย่าง: คำสั่งนี้จะลบ communications device entry สำหรับอุปกรณ์ COMDEV จาก subsystem description SBS1 ในไลบรารี LIB2.

```
RMVCMNE  SBSDB(LIB2/SBS1)  DEV(COMDEV)
```

การลบ job queue entries:

คุณสามารถลบ job queue entries ออกจาก subsystem description โดยใช้อินเตอร์เฟซแบบอักษร. งานบนคิวงานยังคงอยู่บนคิว เมื่อ job queue entry ถูกลบออกจาก subsystem description. จะไม่สามารถลบ job queue entry ถ้างานแ็คทีฟในปัจจุบันใดๆ ถูกเริ่มจาก คิวงาน.

คำสั่ง: Remove Job Queue Entry (RMVJOBQE)

ตัวอย่าง: คำสั่งนี้จะลบ job queue entry ที่อ้างอิงคิวงาน BATCH2 ใน MYLIB ออกจาก NIGHTRUN subsystem description ที่จัดเก็บ ในไลบรารี MYLIB.

```
RMVJOBQE  SBSDB(MYLIB/NIGHTRUN)  JOBQ(MYLIB/BATCH2)
```

การลบ prestart job entries:

คุณสามารถลบ prestart job entries ออกจาก subsystem description โดยใช้อินเตอร์เฟซแบบอักษร. จะไม่สามารถลบ prestart job entry ถ้างานแ็คทีฟในปัจจุบันใดๆ ถูกเริ่มโดยใช้ entry.

เมื่อลบ entry ที่มีการระบุ *LIBL เป็นชื่อไลบรารี, โปรแกรมที่มีชื่อที่ระบุจะถูกค้นหาในรายชื่อไลบรารี. ถ้าพบโปรแกรมในรายชื่อไลบรารี แต่ entry มีอยู่พร้อมกับชื่อไลบรารีอื่น (ซึ่งพบในภายหลังในรายชื่อไลบรารี), จะไม่มี entry ถูกลบออก. ถ้าไม่พบ โปรแกรมในรายชื่อไลบรารี แต่มี entry อยู่, จะไม่มี entry ถูก ลบออก.

คำสั่ง: Remove Prestart Job Entry (RMVPJE)

ตัวอย่าง: คำสั่งนี้จะลบ prestart job entry สำหรับโปรแกรม PGM1 (ในไลบรารี QGPL) ออกจาก PJE subsystem description ที่มีอยู่ในไลบรารี QGPL.

```
RMVPJE  SBSDB(QGPL/PJE)  PGM(QGPL/PGM1)
```

การลบ routing entries:

คุณสามารถลบ routing entry ออกจาก subsystem description ที่ระบุได้โดยใช้อินเตอร์เฟซแบบอักษร. ระบบย่อยสามารถแก้ไขที่ในเวลาที่ยัง คำสั่งได้. อย่างไรก็ตาม, จะไม่สามารถลบ routing entry ถ้ามีงานแก้ไขที่ในปัจจุบันใดๆ ที่ถูกเริ่มต้นโดยใช้ entry.

คำสั่ง: Remove Routing Entry (RMVRTGE)

ตัวอย่าง: คำสั่งนี้จะลบ routing entry 9912 ออกจาก subsystem description PERT ในไลบรารี OR.

```
RMVRTGE  SBSDB(OR/PERT)  SEQNBR(9912)
```

การลบ workstation entries:

คุณสามารถลบ workstation entry ออกจาก subsystem description โดยใช้อินเตอร์เฟซแบบอักษร. ระบบย่อยสามารถแก้ไขที่ในเวลาที่ยัง คำสั่งได้. อย่างไรก็ตาม, งานทั้งหมดที่แก้ไขที่ผ่านทาง workstation entry ต้องถูกจบก่อน entry จึงจะสามารถลบออกได้.

คำสั่ง: Remove Work Station Entry (RMVWSE)

ตัวอย่าง: ตัวอย่างนี้จะลบ workstation entry สำหรับเวิร์กสแตชัน B53 ออกจาก subsystem description ที่มีชื่อว่า CHARLES ในไลบรารี LIB2.

```
RMVWSE  SBSDB(LIB2/CHARLES)  WRKSTN(B53)
```

การตั้งค่าระบบย่อยแบบโต้ตอบ

ข้อมูลในส่วนนี้อธิบายวิธีการตั้งค่าระบบย่อยแบบโต้ตอบใหม่.

ขั้นตอนเหล่านี้ใช้กับวิธีการป้อนคำสั่งด้วยตนเอง. อย่างไรก็ตาม, คุณควรใช้โปรแกรม CL ในการสร้างระบบย่อยของคุณ เพื่อให้คุณสามารถสร้าง configurations ของคุณใหม่สำหรับการกู้คืนได้.

เมื่อคุณตั้งค่า ระบบย่อยแบบโต้ตอบใหม่ คุณควรพิจารณาถึงจำนวนอุปกรณ์ที่จะถูกจัดสรรให้ที่ระบบย่อยนั้น. เนื่องจากระบบย่อยมีฟังก์ชันการจัดการอุปกรณ์, เช่นการแสดงผล signon และการจัดการการแก้ไขข้อผิดพลาดอุปกรณ์, คุณอาจต้องจำกัดจำนวนของอุปกรณ์ ที่จัดสรรที่ระบบย่อยเดียว. ให้ดูหัวข้อ Communication limits สำหรับข้อมูลเพิ่มเติม.

หมายเหตุ: ข้อมูลนี้นำเสนอสาระสำคัญของสิ่งที่เกี่ยวข้องกับการตั้งค่าระบบย่อยแบบโต้ตอบ. รายงานจากประสบการณ์เกี่ยวกับระบบย่อยมีคำอธิบายอย่างละเอียดของแต่ละ ขั้นตอนและอ็อปชันเพิ่มเติมที่พร้อมใช้งานสำหรับแต่ละ ขั้นตอน.

การตั้งค่าระบบย่อยแบบโต้ตอบ: สร้างไลบรารี:

สร้างไลบรารีซึ่งคุณใช้จัดเก็บอ็อบเจกต์ configuration ระบบย่อย ของคุณ.

ตัวอย่างนี้ใช้ SBSLIB เป็นไลบรารี.

```
CRTLIB SBSLIB TEXT('LIBRARY TO HOLD SUBSYSTEM CONFIGURATION OBJECTS')
```

การตั้งค่าระบบย่อยแบบโต้ตอบ: สร้างคลาส:

สร้างคลาส. คลาสกำหนดลักษณะประสิทธิภาพสำหรับ ระบบย่อยแบบโต้ตอบของคุณ.

เพื่อสร้างคลาสที่เหมือนกับคลาส QINTER, ให้ป้อน คำสั่งต่อไปนี้:

```
CRTCLS SBSLIB/INTER1 RUNPTY(20) TIMESLICE(2000) PURGE(*YES) DFTWAIT(30)  
TEXT('Custom Interactive Subsystem Class')
```

คุณสามารถใช้คลาส QINTER ใน QGPL สำหรับระบบย่อยแบบโต้ตอบที่ ปรับตามความต้องการของคุณ, หรือคุณสามารถสร้างคลาสเดียวเพื่อใช้สำหรับระบบย่อยแบบโต้ตอบทั้งหมดของคุณ, หรือคุณสามารถสร้างหนึ่งคลาสสำหรับแต่ละระบบย่อยแบบโต้ตอบ.

ตัวเลือกของคุณ ควรขึ้นอยู่กับว่า คุณต้องการจะปรับการตั้งค่าประสิทธิภาพสำหรับระบบย่อย ตามความต้องการของคุณหรือไม่. ระบบย่อยที่ได้มาจาก IBM ถูกจัดส่งมาพร้อมกับคลาสที่สร้างขึ้นสำหรับแต่ละระบบย่อย, โดยมีชื่อของคลาสเหมือนกับชื่อของระบบย่อย.

ถ้าคุณไม่ต้องการสร้างคลาสสำหรับแต่ละระบบย่อย โดยมีชื่อที่เหมือนกับระบบย่อย, คุณจะต้องระบุชื่อคลาส บนคำสั่ง Add Routing Entry (ADDRTGE). ที่เป็นเช่นนี้เพราะดีฟอลต์สำหรับพารามิเตอร์ CLS คือ *SBSD, ซึ่งหมายความว่า ชื่อคลาสมีชื่อเหมือนกับ subsystem description.

การตั้งค่าระบบย่อยแบบโต้ตอบ: สร้าง subsystem descriptions:

สร้าง subsystem description. ทำซ้ำขั้นตอนนี้สำหรับแต่ละระบบย่อย ที่คุณต้องการกำหนด.

ตัวอย่างต่อไปนี้จะสร้าง subsystem description ด้วยแอตทริบิวต์เฉพาะของ QINTER.

```
CRTSBSD SBSDB(SBSLIB/INTER1) POOLS((1 *BASE) (2 *INTERACT)) SGNDSPF(*QDSIGNON)
```

การตั้งค่าระบบย่อยแบบโต้ตอบ: สร้างคิวงาน:

สร้างคิวงานสำหรับระบบย่อยโดยใช้ชื่อที่เหมือนกับ ชื่อระบบย่อย และใส่เพิ่ม job queue entry ที่ subsystem description.

ขั้นตอนนี้เป็นขั้นตอนบังคับ ถ้าคุณต้องการจะใช้คำสั่ง Transfer Job (TFRJOB) เพื่อถ่ายโอนงานเข้าในระบบย่อยที่ปรับตามความต้องการของคุณ.

```
CRTJOBQ JOBQ(SBSLIB/INTER1)  
ADDJOBQE SBSDB(SBSLIB/INTER1) JOBQ(SBSLIB/INTER1) MAXACT(*NOMAX)
```

การตั้งค่าระบบย่อยแบบโต้ตอบ: ใส่เพิ่ม routing entry:

ใส่เพิ่ม routing entry ที่ระบบย่อย.

routing entries ที่จัดให้มาพร้อมกับระบบสำหรับ QINTER มีฟังก์ชันเพิ่มเติมบางอย่าง. ถ้าคุณต้องการฟังก์ชันเหล่านั้น, ให้ใส่เพิ่ม routing entries เหล่านี้ที่ subsystem descriptions ที่ปรับตามความต้องการของคุณ.

```
ADDRTGE SBSDB(SBSLIB/INTER1) SEQNBR(9999) CMPVAL(*ANY) PGM(QSYS/QCMD) POOLID(2)
```

การตั้งค่าระบบย่อยแบบโต้ตอบ: ใส่เพิ่ม workstation entries:

ใส่เพิ่ม workstation entries ที่ subsystem description. นี่คือขั้นตอนหลัก สำหรับการกำหนดว่าอุปกรณ์ใดจะถูกจัดสรรที่ระบบย่อยใด.

คุณต้องกำหนดว่าระบบย่อยใดควรจัดสรรอุปกรณ์ใด (AT(*SIGNON)). นอกจากนี้, ต้องกำหนดว่าคุณต้องการอนุญาตให้ใช้ TFRJOB จากระบบย่อยหนึ่งไปยังระบบย่อยอื่นหรือไม่ (AT(*ENTER)).

```
ADDWSE SBSDB(SBSLIB/PGRM) WRKSTN(PGMR*) AT(*SIGNON)
ADDWSE SBSDB(SBSLIB/ORDERENT) WRKSTN(ORDERENT*) AT(*SIGNON)
ADDWSE SBSDB(QGPL/QINTER) WRKSTN(QPADEV*) AT(*SIGNON)
```

ในตัวอย่างนี้, หลักการตั้งชื่อระบบย่อยและอุปกรณ์เป็นไปตาม ชนิดของงานที่ผู้ใช้ทำ. โปรแกรมเมอร์ทั้งหมดมีอุปกรณ์ที่ตั้งชื่อว่า PGMR และรันในระบบย่อย PGRM. Order entry personnel ทั้งหมดมี อุปกรณ์ที่ตั้งชื่อว่า ORDERENT และรันในระบบย่อย ORDERENT. ผู้ใช้อื่น ทั้งหมดจะใช้หลักการตั้งชื่อดีฟอลต์ระบบของ QPADEVxxxx และรันในระบบย่อย QINTER ที่ได้มาจาก IBM.

การตั้งค่าระบบย่อยแบบโต้ตอบ: ปรับ QINTER ตามความต้องการ:

เมื่อคุณเริ่มใช้ชุดของระบบย่อยของคุณเอง, คุณอาจไม่ต้องการ ใช้ QINTER. อย่างไรก็ตาม, ถ้าคุณมีเหตุผลที่จะใช้ QINTER ต่อไป, คุณต้องตรวจสอบให้แน่ใจว่า QINTER ถูกตั้งค่าให้ไม่จัดสรรเวิร์กสเตชัน ที่คุณต้องการจะรันภายใต้ระบบย่อยอื่นของคุณ. สามารถทำเช่นนี้ได้สองวิธี.

การลบ *ALL workstation entry ออกจาก QINTER:

1. ลบ *ALL workstation entry ออกจาก QINTER, แล้วใส่เพิ่ม workstation entries เฉพาะที่บ่งชี้ถึงอุปกรณ์ที่คุณต้องการให้ QINTER จัดสรร. การลบ workstation type entry ของ *ALL คือการป้องกัน QINTER ไม่ให้พยายามจัดสรรเวิร์กสเตชัน ทั้งหมด.
2. ใส่เพิ่ม workstation entry สำหรับอุปกรณ์ที่มีชื่อว่า DSP* เพื่ออนุญาตให้อุปกรณ์แสดงผล twinax-attached ทั้งหมดถูกจัดสรรที่ QINTER ต่อไป.

ในตัวอย่างนี้, อุปกรณ์แสดงผล twinax-attached จะรันใน QINTER ต่อไป; QINTER จะไม่พยายามจัดสรรอุปกรณ์อื่น.

```
RMVWSE SBSDB(QGPL/QINTER) WRKSTNTYPE(*ALL)
ADDWSE SBSDB(QGPL/QINTER) WRKSTN(DSP*)
```

วิธีที่สอง:

ใส่เพิ่ม workstation entry เพื่อบอกให้ QINTER ไม่จัดสรรอุปกรณ์ ที่ถูกจัดสรรไปที่ระบบย่อยอื่น. อย่างไรก็ตาม, อนุญาตให้ QINTER จัดสรรอุปกรณ์อื่น ที่ไม่ได้ถูกจัดสรรไปที่ระบบย่อยได้ต่อไป. สิ่งนี้จะเก็บ workstation type entry ของ *ALL ในระบบย่อย QINTER และใส่เพิ่ม workstation name entries ที่มีพารามิเตอร์ AT ของอุปกรณ์ ที่จะถูกจัดสรรไปที่ระบบย่อยอื่น.

```
ADDWSE SBS(DQGPL/QINTER) WRKSTN(PGMR*) AT(*ENTER)
ADDWSE SBS(DQGPL/QINTER) WRKSTN(ORDERENT*) AT(*ENTER)
```

การตั้งค่าระบบย่อยแบบโต้ตอบ: ตั้งค่าคอนโซล:

ข้อควรพิจารณาสุดท้ายเกี่ยวกับ QINTER, ซึ่งมีความสำคัญมากคือ workstation type entry ของ *CONS สำหรับคอนโซล. ตรวจสอบให้แน่ใจว่าคุณไม่ได้ป้องกันไม่ให้บุคคลใด signing on ที่คอนโซลโดยไม่ได้ตั้งใจ. คุณอาจป้องกันสิ่งนี้ โดยการไม่ได้ใส่ workstation entries ใดๆ สำหรับคอนโซลที่ระบบย่อย แบบโต้ตอบที่ปรับตามความต้องการของคุณ.

ระบบที่จัดส่งมาพร้อมกับระบบย่อยการควบคุมมี workstation entry เป็น AT(*SIGNON) สำหรับคอนโซล (*CONS workstation type entry). QINTER มี AT(*ENTER) workstation type entry สำหรับคอนโซล.

วิธีปฏิบัติที่ดีคือ ควรรันคอนโซลในระบบย่อยการควบคุมเสมอ และไม่ถ่ายโอนงานคอนโซลเข้าในระบบย่อยแบบโต้ตอบอื่น. การทำเช่นนี้ ช่วยป้องกันผู้ใช้ที่คอนโซลไม่ให้จบงานของตนโดยไม่ได้ตั้งใจ.

ตัวอย่าง เช่น, ถ้าผู้ใช้ที่คอนโซลถ่ายโอนงานของตนไปที่ INTER1 และลิม, และในภายหลังเตรียมการสำรองข้อมูลโดยใช้คำสั่ง by End System (ENDSYS), งานคอนโซลจะถูก จบลงด้วย. นี่ไม่ใช่สิ่งที่ผู้ควบคุมเครื่องตั้งใจ.

การกำหนดผู้ใช้ที่ระบบย่อยเฉพาะ:

หัวข้อนี้อธิบายเทคนิคต่างๆ ที่สามารถใช้สำหรับการกำหนดชื่ออุปกรณ์ และการเชื่อมโยงชื่ออุปกรณ์เหล่านั้น กับผู้ใช้. หลังจากเสร็จสมบูรณ์แล้ว, คุณสามารถใช้ workstation entries เพื่อนำผู้ใช้ไปที่ระบบย่อยที่ต้องการ.

ระบบมีหลักการตั้งชื่อดีฟอลต์ที่จะใช้สำหรับ เซสชันการแสดงผล. ในบางครั้งหลักการตั้งชื่อนี้อาจไม่เพียงพอสำหรับการเรอต์ workstation entries ในหลายระบบย่อยโดยใช้โปรไฟล์ผู้ใช้.

คุณสามารถทำการเปลี่ยนแปลงบนระบบของคุณเพื่อเพิ่มประสิทธิภาพ ดีฟอลต์ของระบบ โดยการกำหนดและจัดการหลักการตั้งชื่ออุปกรณ์ของคุณเอง. โดยสามารถ ทำได้หลายวิธี. แต่ละวิธีมีข้อดีและข้อเสียแตกต่างกันไป.

การกำหนดค่าเริ่มต้นอุปกรณ์เทอร์มินัลและ terminal exit points:

Telnet Device Initialization and Terminal Exit Points. exit points ช่วยให้สามารถกำหนดชื่ออุปกรณ์ในทันทีที่โคลเอ็นต์ signing on ที่ระบบ.

exit point แสดง client IP address และชื่อโปรไฟล์ผู้ใช้ (พร้อมกับข้อมูลเพิ่มเติม). จากนั้นคุณสามารถทำการแม็ป โคลเอ็นต์ของคุณเองไปที่ device description ที่ควรจะใช้ สำหรับโคลเอ็นต์.

exit point การกำหนดค่าเริ่มต้นอุปกรณ์ยังนำเสนอ วิธีการข้าม signon panel ด้วย.

ข้อดีของการใช้ exit points เหล่านี้ ในการจัดการหลักการตั้งชื่ออุปกรณ์ของคุณคือ คุณมีการควบคุมศูนย์กลาง บนเซิร์ฟเวอร์ iSeries สำหรับโคลเอ็นต์ทั้งหมดของคุณ.

ข้อเสียคือต้องใช้ทักษะการทำ โปรแกรมมิง.

exit point การเลือกอุปกรณ์:

exit point ช่วยให้คุณสามารถระบุหลักการตั้งชื่อที่ใช้สำหรับอุปกรณ์เสมือน และคอนโทรลเลอร์เสมือนที่สร้างขึ้นโดยอัตโนมัติ และสามารถระบุข้อจำกัดในการสร้างอย่างอัตโนมัติที่ใช้สำหรับค่าขอพิเศษ.

ด้วย exit point นี้ คุณสามารถระบุหลักการตั้งชื่อต่างๆ สำหรับอุปกรณ์ที่สร้างขึ้นโดยอัตโนมัติซึ่งใช้โดย Telnet, 5250 Display Station Pass-through, และ virtual terminal APIs.

นอกจากนั้น, คุณสามารถจัดการอุปกรณ์ Pass-through และค่ากำหนดของระบบ Telnet (QAUTOVRT) ได้อย่างถูกต้องมากขึ้น. ตัวอย่างเช่น, คุณอาจใช้ค่าหนึ่งสำหรับอุปกรณ์ที่สร้างขึ้นโดยอัตโนมัติสำหรับ Telnet และใช้อีกค่าหนึ่งสำหรับอุปกรณ์ 5250 Display Station Pass-through.

exit point นี้ช่วยให้คุณสามารถควบคุมหลักการตั้งชื่อดีฟอลต์ที่ใช้สำหรับอุปกรณ์ (เช่น QPADEV*) แต่ exit point เพียงอย่างเดียวไม่ทำให้คุณสามารถระบุอุปกรณ์สำหรับผู้ใช้. exit point นี้มีประโยชน์มากที่สุด ถ้าคุณกำลังใช้วิธีการหลายอย่างในการเชื่อมต่อกับระบบ (Telnet, 5250 Display Station Pass-through, WebFacing, และอื่นๆ) เนื่องจาก exit point นี้อนุญาตให้คุณใช้หลักการตั้งชื่อ อุปกรณ์ที่แตกต่างกันได้และมีการจัดการ QAUTOVRT ที่แม่นยำสำหรับ วิธีการเข้าใช้งานต่างๆ.

PC5250 (iSeries Access) workstation ID support:

คุณสามารถตั้งค่า iSeries Access เพื่อเชื่อมต่อกับชื่อเวิร์กสเตชันเฉพาะได้. ถ้าคุณคลิกปุ่มคำอธิบายจากหน้าต่างนี้, อีอ็อปชันหลากหลาย สำหรับการระบุ workstation ID, เช่นการสร้างชื่อใหม่ถ้า ชื่อที่ระบุถูกใช้ไปแล้ว, จะแสดงขึ้น.

ข้อเสียของวิธีนี้คือ คุณต้องจัดการการตั้งค่า PC5250 configuration บนแต่ละและทุกโคลเอ็นต์ที่เชื่อมต่อกับ เซิร์ฟเวอร์ของคุณ.

OS/400 Telnet Client:

ด้วยการใช้คำสั่ง OS/400® Telnet Client (STRTCPTELN หรือ TELNET), คุณสามารถระบุชื่ออุปกรณ์ที่ใช้ในการ sign on ที่ระบบเซิร์ฟเวอร์.

ข้อเสียของวิธีดีฟอลต์คือ คุณต้องตรวจสอบให้แน่ใจว่าการใช้คำสั่ง STRTCPTELN (TELNET) ทั้งหมดระบุ remote virtual display value อย่างถูกต้อง. เพื่อลดความกังวลนี้, คุณสามารถสร้างเวอร์ชันที่เลือก กำหนดของคำสั่ง STRTCPTELN เพื่อให้แน่ใจถึง remote virtual terminal display value และเรียกใช้คำสั่งที่ IBM จัดให้.

การสร้างคอนโทรลเลอร์และอุปกรณ์เสมือนด้วยตนเอง:

คุณสามารถสร้างคอนโทรลเลอร์และอุปกรณ์เสมือนด้วยตนเองได้.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการสร้างอุปกรณ์เสมือนสำหรับ Telnet, ให้ดูหัวข้อ Configure the Telnet Server ใน iSeries Information Center.

ในวิธีนี้ คุณสามารถควบคุมชื่อของคอนโทรลเลอร์และอุปกรณ์ของคุณได้, แต่คุณไม่สามารถแม้อุปกรณ์เฉพาะกับ ผู้ใช้เฉพาะ.

การสร้างระบบย่อยการควบคุม

IBM จัดเตรียม controlling subsystem configurations ที่สมบูรณ์สองรายการ: QBASE (ระบบย่อย การควบคุมดีฟอลต์), และ QCTL. ในแต่ละครั้ง สามารถมีระบบย่อยการควบคุมที่แอ็คทีฟ บนระบบได้เพียงหนึ่งระบบย่อยเท่านั้น. โดยปกติ, subsystem configurations ที่ IBM จัดให้ควรจะเพียงพอ สำหรับความต้องการทางธุรกิจส่วนใหญ่. อย่างไรก็ตาม, คุณสามารถสร้างระบบย่อยการควบคุม เวอร์ชันของคุณเอง และตั้งค่าให้ตรงกับความต้องการเฉพาะของบริษัทของคุณ ได้มากขึ้น.

ให้ใช้ระบบย่อยการควบคุมที่ได้มาจาก IBM ซึ่งได้แก่ QBASE หรือ QCTL เป็นต้นแบบสำหรับการสร้างระบบย่อยการควบคุมของคุณเอง.

หมายเหตุ: หากคุณสร้างระบบย่อย การควบคุมของคุณเอง, คุณควรใช้ชื่ออื่นที่ไม่ใช่ QBASE หรือ QCTL.

subsystem description สำหรับระบบย่อยการควบคุมควรมีข้อมูล ดังต่อไปนี้:

- routing entry ที่ประกอบด้วย:
 - *ANY หรือ QCMDI อย่างใดอย่างหนึ่ง เป็นข้อมูลการเรอต์
 - QSYS/QCMD เป็นโปรแกรมที่จะเรียกใช้
 - คลาส QSYS/QCTL หรือคลาสที่ผู้ใช้กำหนด. (ที่เป็นเช่นนี้เนื่องจากผู้ใช้, โดยทั่วไปคือ ผู้ควบคุมระบบ, ต้องสามารถป้อนคำสั่งเพื่อทำบางสิ่ง เช่น ให้มีหน่วยความจำมากขึ้น ถ้ามีการใช้งานถึง threshold ของหน่วยความจำรอง.)
- workstation entry สำหรับคอนโซลที่มีชนิด *SIGNON (*SIGNON คือ คำสำหรับพารามิเตอร์ AT®, ที่ระบุบนคำสั่ง Add Work Station Entry (ADDWSE).)

คำ *SIGNON บ่งชี้ว่าจอแสดงผล signon จะถูกแสดงขึ้นที่เวิร์กสเตชัน เมื่อระบบย่อยเริ่มต้นขึ้น. ข้อกำหนดนี้ทำให้มั่นใจได้ว่า ระบบย่อยมี อุปกรณ์แบบโต้ตอบสำหรับ entry ของคำสั่งระดับระบบและระบบย่อย. คำสั่ง End System (ENDSYS) สิ้นสุด iSeries โปรแกรม ที่ได้ไลเซนส์เป็น single session (หรือจอแสดงผล signon) ที่คอนโซลในระบบย่อย การควบคุม. subsystem description ที่ไม่มี workstation entry สำหรับคอนโซลไม่สามารถถูกเริ่มเป็นระบบย่อยการควบคุม.

- entry สำหรับเวิร์กสเตชันอื่น:

สิ่งนี้นำเสนอซอร์สทางเลือก ของอินพุตการควบคุม. หากตรวจพบปัญหาเกี่ยวกับคอนโซลในระหว่าง attended IPL และถ้าคำกำหนดของระบบ console problem occurs (QSCPFCONS) ถูกเซตเป็น '1', IPL จะทำงานต่อไปในโหมด unattended. จากนั้น, ถ้า subsystem description สำหรับระบบย่อยการควบคุมมี workstation entry สำหรับเวิร์กสเตชันอื่น, สามารถจะใช้เวิร์กสเตชันทางเลือกได้.

- routing entry ประกอบไปด้วย:
 - QSYS/QARDRIVE เป็นโปรแกรมที่จะเรียกใช้,
 - และ QSYS/QCTL เป็นคลาส

หลังจากที่คุณสร้างระบบย่อยการควบคุมแล้ว, ให้เปลี่ยนคำกำหนดของระบบ Controlling subsystem/library (QCTLSBSD) เป็นดังนี้ (สมมุติให้ description มีชื่อว่า QGPL/QCTLA):

```
CHGSYSVAL SYSVAL(QCTLSBSD) VALUE('QCTLA QGPL')
```

การเปลี่ยนแปลง จะมีผลบังคับใช้ที่ IPL ถัดไป.

การวางระบบในสภาพที่ถูกจำกัด

ถ้าระบบย่อยทั้งหมด, รวมถึงระบบย่อยการควบคุม จบลง, ระบบจะอยู่ในสภาพถูกจำกัด. คุณสามารถวางระบบในสภาพที่ถูกจำกัดได้โดยการใช้หนึ่งหรือสองคำสั่งจากเวิร์กสเตชัน แบบโต้ตอบ.

คำสั่ง: End Subsystem พร้อมกับพารามิเตอร์ *ALL (ENDSBS SBS(*ALL))

คำสั่ง: End System (ENDSYS)

สำคัญ: คำสั่ง ENDSBS หรือ ENDSYS ควรออกใช้จากงานแบบโต้ตอบในระบบย่อยการควบคุม, และ เฉพาะจากเวิร์กสเตชันที่มี entry ใน controlling subsystem description ระบบ AT(*SIGNON). งานแบบโต้ตอบซึ่งคำสั่งถูกออกใช้จากที่นั้น ยังคงแอคทีฟ เมื่อระบบย่อยการควบคุมเข้าสู่สภาพถูกจำกัด. ถ้างานที่ออกคำสั่งเป็นหนึ่งในสองงานที่แอคทีฟที่เวิร์กสเตชัน (การใช้ปุ่ม System Request หรือคำสั่ง TFRSECJOB), งานทั้งสองจะไม่ถูก forced ให้จบ. อย่างไรก็ตาม, ระบบย่อยการควบคุมจะไม่จบ สภาพที่ถูกจำกัด จนกว่าคุณจะทำงานใดงานหนึ่ง. งานกลุ่มการหยุดชั่วคราว ยังป้องกันไม่ให้ระบบย่อยการควบคุมจบ (จนกว่างานกลุ่ม จะจบลง).

เมื่อระบบอยู่ในสภาพที่ถูกจำกัด, activity ส่วนใหญ่บนระบบจะจบลง, และมีเวิร์กสเตชันที่แอคทีฟเพียงเวิร์กสเตชันเดียวเท่านั้น. ระบบต้องอยู่ใน สภาพนี้สำหรับการรันคำสั่ง เช่น Save System (SAVSYS) หรือ Reclaim Storage (RCLSTG).

โปรแกรมการวินิจฉัยปัญหา เกี่ยวกับอุปกรณ์บางโปรแกรมอาจต้องการให้ระบบอยู่ในสภาพที่ถูกจำกัดด้วย. เพื่อจบสภาพที่ถูกจำกัด คุณต้องเริ่มระบบย่อยการควบคุม อีกครั้ง.

การจัดการพูลหน่วยความจำ

การตรวจสอบให้แน่ใจว่างานมีหน่วยความจำเพียงพอที่จะดำเนินงานอย่างมีประสิทธิภาพถือเป็นสิ่งที่สำคัญ. ถ้ามีการให้หน่วยความจำกับระบบย่อย A มากเกินไปและไม่เพียงพอที่ระบบย่อย B, งานในระบบย่อย B อาจเริ่มรันได้ไม่ดี. ข้อมูลต่อไปนี้อธิบายงานหลากหลายที่เกี่ยวข้องกับการจัดการ พูลหน่วยความจำ.

การดูข้อมูลพูลหน่วยความจำ

คุณสามารถดูข้อมูลเกี่ยวกับพูลหน่วยความจำที่มีอยู่บนระบบของคุณ ได้โดยใช้ iSeries Navigator หรืออินเตอร์เฟซแบบอักษร.

iSeries Navigator:

ใน iSeries Navigator, ให้ขยาย My Connections → server → Work Management → Memory Pools → Active Pools หรือ Shared Pools.

Active Pools container แสดงทั้ง shared และ private pools トラバเท่าที่พูลยังแอคทีฟอยู่. Shared Pools container แสดง shared pools ทั้งหมดไม่ว่าสถานะปัจจุบันของพูลจะเป็นอะไรก็ตาม. Inactive private pools ไม่มีอยู่นอก pool definition จนกว่าพูลจะถูกเรียกทำงานโดยระบบย่อย. ดังนั้นจึงไม่สามารถดูพูลโดยใช้ iSeries Navigator.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Display Subsystem Description (DSPSBSD)

ใช้อ็พชัน 2 – Pool Definitions เพื่อดู private และ shared pool definitions ทั้งหมดที่มีอยู่ใน subsystem definition นี้.

คำสั่ง: Work with Shared Pools (WRKSHRPOOL)

การกำหนดจำนวนของระบบย่อยที่ใช้พูลหน่วยความจำ

ระบบย่อยจะได้รับการจัดสรรหน่วยความจำจำนวนหนึ่งเพื่อใช้รันงาน. เป็นสิ่งสำคัญที่ต้องทราบจำนวนของระบบย่อยที่กำลังดึงหน่วยความจำออกมาจาก พูลหน่วยความจำเดียวกัน. หลังจากที่คุณทราบจำนวนของระบบย่อยที่กำลังส่งงานไปที่พูลและจำนวนของงานที่กำลังรันในพูล, คุณอาจต้องการลด contention รีซอร์สโดยการปรับเปลี่ยนขนาดและ activity level ของพูล.

iSeries Navigator:

ถ้าต้องการใช้ iSeries Navigator เพื่อมอนิเตอร์จำนวนของระบบย่อยที่กำลังใช้พูลหน่วยความจำ, ให้ทำดังต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → **server** → **Work Management** → **Memory Pools** → **Active Pools** หรือ **Shared Pools**.
2. คลิกขวาพูลหน่วยความจำที่คุณต้องการทำงานด้วยและคลิก **Subsystems**.
จากหน้าต่างนี้ คุณสามารถกำหนดจำนวนของระบบย่อยที่กำลังใช้แต่ละหน่วยความจำเพื่อรันงานของตน.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work with Subsystems (WRKSBS)

คำสั่งนี้แสดงรายการของระบบย่อยทั้งหมดและพูลของระบบย่อย.

การกำหนดจำนวนของงานในพูลหน่วยความจำ

iSeries Navigator นำเสนอวิธีการแสดงรายการของงานที่กำลังรันอยู่ในพูลหน่วยความจำอย่างรวดเร็ว.

ถ้าต้องการกำหนดจำนวนของงานในพูลหน่วยความจำ, ให้ทำดังต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → **server** → **Work Management** → **Memory Pools** → **Active Pools** หรือ **Shared Pools**.
2. คลิกขวาพูลหน่วยความจำที่คุณต้องการใช้และคลิก **Jobs**. หน้าต่างจะปรากฏขึ้นเพื่อแสดงรายการของงานภายในพูลหน่วยความจำ.

คุณยังสามารถดูจำนวนของ thread ในพูลหน่วยความจำโดยการดูคอลัมน์ Thread Count. การนับ thread นำเสนอ ข้อมูลเพิ่มเติมเกี่ยวกับจำนวนของ activity ในพูลหน่วยความจำ.

จากจุดนี้ไป, คุณจะสามารถใช้ฟังก์ชันเกี่ยวกับงานเดียวกันได้เสมือนคุณได้อยู่ในพื้นที่ Active jobs หรือ Server jobs.

การกำหนดพูลที่งานเดี่ยวกำลังรันอยู่

ถ้าคุณมีงานที่ไม่ได้ปฏิบัติเนื่องจากคุณคาดว่าอาจต้อง ตรวจสอบพูลหน่วยความจำซึ่งงานกำลังรันอยู่. ถ้าต้องการกำหนดพูลที่งานเดี่ยว กำลังรันอยู่, ให้ใช้ iSeries Navigator หรือใช้อินเตอร์เฟซแบบ อักษร.

หลังจากที่คุณได้ระบุพูลซึ่งงานกำลังรันอยู่แล้ว, คุณสามารถดูข้อมูลพูลหน่วยความจำและกำหนดว่าต้องทำการเปลี่ยนแปลงหรือไม่. ตัวอย่างเช่น, ถ้าเกิดการเพจมากเกินไป, พูลหน่วยความจำอาจต้องใหญ่ขึ้น. อีกสาเหตุหนึ่งที่ทำให้ประสิทธิภาพลดลงคือ อาจมีงานอื่นอยู่ใน พูลมากเกินไปและคุณต้องเราต์งานนี้ไปที่พูลอื่น.

iSeries Navigator:

ถ้าต้องการใช้ iSeries Navigator, ให้ทำตามต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → **server** → **Work Management** → **Active Jobs** หรือ **Server Jobs**, ขึ้นอยู่กับชนิดของงานที่คุณต้องการทำงานด้วย.
2. ค้นหางานที่มีพูลหน่วยความจำที่คุณต้องการดู.
3. คลิกขวาที่ **Job Name** และคลิก **Properties**.
4. คลิกแท็บ **Resources**. หน้าต่าง Job Properties – Resources แสดงข้อมูลเฉพาะเกี่ยวกับพูลหน่วยความจำของงาน.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work with job (WRKJOB)

อ็อปชัน 1: แสดงแอ็ด ทริบิวต์สถานะงาน

ฟิลด์ Subsystem pool ID มีชื่อของพูลที่กำหนดสำหรับระบบย่อยซึ่งงานกำลังรันอยู่. ฟิลด์นี้จะเป็นพื้นที่เปล่าสำหรับงานที่ไม่แอ็คทีฟในเวลาที่มีการร้องขอแสดงผล. และยังเป็นพื้นที่เปล่าสำหรับงานระบบ (ชนิด SYS), งานการมอนิเตอร์ระบบย่อย (ชนิด SBS) ที่ไม่ได้รับภายในระบบย่อย และ batch immediate jobs (BCI) ที่กำลังรันอยู่ในพูลหน่วยความจำหลัก.

คำสั่ง: Work with active job (WRKACTJOB)

คุณสามารถใช้คำสั่ง WRKACTJOB เพื่อดู system pool ID สำหรับงานแอ็คทีฟ.

การจัดการพารามิเตอร์การปรับสำหรับแชร์พูล

ถ้าต้องการจัดการพารามิเตอร์การปรับสำหรับแชร์พูล, ให้ใช้ iSeries Navigator หรือคำสั่งอินเตอร์เฟซแบบอักษร.

iSeries Navigator:

ถ้าต้องการเข้าใช้งานพารามิเตอร์การปรับโดยใช้ iSeries Navigator, ให้ทำตามต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → **server** → **Work Management** → **Memory Pools** → **Active Pools** หรือ **Shared Pools**.
2. คลิกขวาพูลซึ่งคุณต้องการปรับและคลิก **Properties**.
3. คลิกแท็บ **Tuning**.

จากหน้าต่าง Shared Properties – Tuning คุณสามารถปรับเปลี่ยนค่าเฉพาะ เช่น เปอร์เซ็นต์การจัดสรรพูล, page faults ต่อวินาที, และระดับความสำคัญด้วยตนเองได้.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work with Shared Pool (WRKSHRPOOL)

เลือก อ็อปชัน 11 – แสดงข้อมูลการปรับ.

การจัดการ configuration ของพูล

ถ้าต้องการเปลี่ยนขนาดของพูล, activity level หรืออ็อปชันการเพจ, ให้ใช้ iSeries Navigator หรือคำสั่งอินเตอร์เฟซแบบอักษร.

iSeries Navigator:

ถ้าต้องการเข้าใช้งานค่า configuration ของแชร์พูลโดยการใช้อะไร iSeries Navigator, ให้ทำตามต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย My Connections → server → Work Management → Memory Pools → Active Pools หรือ Shared Pools.
2. คลิกขวาพูลซึ่งคุณต้องการปรับและคลิก Properties.
3. คลิกแท็บ Configuration .

จากหน้าต่าง Shared Properties – Configuration คุณสามารถปรับเปลี่ยน ค่าเฉพาะ เช่น ขนาดของพูล, activity level หรืออ็อปชันการเพจด้วยตนเองได้.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work with Shared Pool (WRKSHRPOOL)

การเปลี่ยนขนาดของพูลหน่วยความจำ

ขนาดของพูลหน่วยความจำมีผลโดยตรงกับปริมาณของงานที่ระบบย่อยสามารถประมวลผลได้. ระบบย่อยยังมีหน่วยความจำมาก, ยิ่งสามารถทำงานให้เสร็จ สมบูรณ์ได้มากขึ้น. นับเป็นเรื่องสำคัญที่คุณควรมอนิเตอร์ระบบของคุณอย่างรอบคอบ ก่อน จะเริ่มการเปลี่ยนพารามิเตอร์ของพูลหน่วยความจำของคุณ. และคุณยังต้อง ตรวจสอบซ้ำระดับเหล่านี้เป็นครั้งคราว, เนื่องจากอาจต้องมีการปรับบางอย่างใหม่.

ตรวจสอบให้แน่ใจว่าคุณได้ปิดตัวปรับระบบ ก่อนที่คุณจะเริ่มเปลี่ยนขนาดของ พูลหน่วยความจำด้วยตนเอง. ตัวปรับระบบจะปรับขนาดของพูลหน่วยความจำที่แบ่ง ใช้โดยอัตโนมัติให้เหมาะสมกับปริมาณงานที่ระบบกำลังทำงานอยู่. ถ้าไม่ได้ปิดตัวปรับระบบ, การเปลี่ยนแปลงที่คุณทำด้วยตนเองอาจถูกเปลี่ยน โดยอัตโนมัติโดยตัวปรับ.

ปิดตัวปรับระบบโดยการเปลี่ยนค่ากำหนดของระบบ Automatically adjust memory pools และ activity levels (QPFRADJ) เป็น 0. (0 = ไม่มีการปรับเปลี่ยน)

iSeries Navigator:

1. ใน iSeries Navigator, ให้ขยาย My Connections → server → Work Management → Memory Pools → Active Pools หรือ Shared Pools.
2. คลิกขวาพูลหน่วยความจำที่คุณต้องการทำงาน (เช่น, Interactive) และคลิก Properties. หน้าต่าง Memory Pool Properties จะปรากฏขึ้น.
3. จากแท็บ Configuration ของหน้าต่าง Properties, คุณสามารถเปลี่ยนจำนวนของหน่วยความจำที่กำหนดได้. หน่วยความจำที่กำหนดคือจำนวนสูงสุด ของหน่วยความจำที่พูลสามารถใช้ได้. ตัวเลขที่คุณใส่ให้ในที่นี้ควรจะเป็นขนาดหน่วยความจำที่คุณคิดว่าพูลจะต้องการใช้เพื่อสนับสนุนระบบย่อยที่พูลให้บริการอยู่.

หมายเหตุ: Base pool คือพูลหน่วยความจำที่ไม่มีจำนวนของหน่วยความจำที่กำหนดไว้. แต่มีขนาดหน่วยความจำต่ำสุดที่จำเป็นในการใช้รัน. Base pool ประกอบด้วยทุกสิ่งที่ไม่ได้ถูกจัดสรรไว้ในที่อื่น. ตัวอย่างเช่น, คุณอาจจะมีหน่วย

ความจำอยู่ 1000 MB บนระบบ ซึ่ง 250 MB ถูกจัดสรรให้กับ Machine pool และ 250 MB ถูกจัดสรรให้กับ Interactive pool. อีก 500 MB ไม่ได้ถูกจัดสรรให้กับสิ่งใด. หน่วยความจำที่ไม่ได้ถูกจัดสรรนี้จะถูกจัดเก็บใน Base pool จนกว่าจะมีความจำเป็นต้องใช้.

ให้ใช้ความระมัดระวังในการย้ายหน่วยความจำ. เพราะการย้ายหน่วยความจำจากพูลหนึ่งไปยังอีกพูลหนึ่งอาจช่วยแก้ปัญหาให้กับระบบย่อยหนึ่งได้, แต่ก็อาจสร้างปัญหาให้กับระบบย่อยอื่นๆ, ซึ่ง, อาจทำให้ประสิทธิภาพโดยรวมของระบบลดลงด้วย.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Change System Value (CHGSYSVAL)

ตัวอย่าง: คำสั่งข้างล่างนี้จะเปลี่ยนขนาดของ Machine pool.

```
CHGSYSVAL QMCHPOOL 'new-size-in-KB'
```

นี่สอดคล้อง กับพูล 1 บนจอแสดงผล WRKSYSSTS.

ตัวอย่าง: คำสั่งข้างล่างนี้จะเปลี่ยนขนาดต่ำสุดของ base pool.

```
CHGSYSVAL QBASPOOL 'new-minimum-size-in-KB'
```

นี่สอดคล้อง กับพูล 2 บนจอแสดงผล WRKSYSSTS.

หมายเหตุ: คำกำหนดของระบบ QBASPOOL ควบคุมขนาดต่ำสุดของ base pool เท่านั้น. Base pool มีหน่วยความจำทั้งหมดที่ไม่ได้ถูกจัดสรรให้กับพูลอื่น.

การเปลี่ยนขนาดของ shared pool:

คำสั่ง: Change Shared Storage Pool (CHGSHRPOOL)

การ เปลี่ยนขนาดของ shared pools มีผลบังคับใช้ในทันที ถ้า shared pool แอ็คทีฟ และมีหน่วยความจำที่พร้อมใช้งานมากพอ.

คำสั่ง: Work with Shared Storage Pools (WRKSHRPOOL)

คำสั่งนี้ช่วยให้คุณเข้าถึง ข้อมูลชื่อและสถานะของ shared pools. โดยการใช้เมนูอ็อปชัน คุณสามารถเปลี่ยนค่าของขนาดพูล และ activity level สูงสุดได้.

การสร้างพูลหน่วยความจำส่วนตัว

พูลหน่วยความจำส่วนตัว (เรียกอีกอย่างว่าพูลหน่วยความจำที่ผู้ใช้กำหนด) สามารถใช้โดยระบบย่อยที่ได้มาจาก IBM หรือ โดยระบบย่อยที่ผู้ใช้กำหนด. คุณสามารถกำหนด definition พูลหน่วยความจำได้สูงสุดถึง 10 definition สำหรับระบบย่อยหนึ่ง. คุณสร้างพูลหน่วยความจำส่วนตัวใน subsystem description.

ถ้าต้องการสร้างพูลหน่วยความจำส่วนตัว, ให้ใช้อินเตอร์เฟซแบบอักษร.

คำสั่ง: พารามิเตอร์ Create Subsystem Description (CRTSBSD) POOLS .

คำสั่ง: พารามิเตอร์ Change Subsystem Description (CHGSBSD) POOLS .

หมายเหตุ: แม้ว่าแต่ละ subsystem description สามารถมีพูลหน่วยความจำที่ผู้ใช้ กำหนดได้มากถึง 10 พูล, แต่มีข้อจำกัดในการดำเนินการที่ว่า พูลหน่วยความจำที่สามารถแอดคิฟในเวลาใดๆ ไม่สามารถมีมากกว่า 64 พูล. (ตัวเลขนี้รวมถึง Base memory pool และ Machine memory pool.) ถ้าใช้งานครบตามขีดจำกัดการจัดสรรสูงสุด ก่อนจะมีการจัดสรรพูลหน่วยความจำทั้งหมดสำหรับระบบย่อย, Base pool จะถูกใช้สำหรับขั้นตอนการเราต์ใดๆ ที่ยังคงต้องการพูลหน่วยความจำอยู่.

การจัดการคิวงาน

เมื่อคุณจัดการงานบนระบบ, คุณอาจพบว่ามีความจำเป็น ต้องดำเนินการงานที่กำลังรออยู่ในคิวงาน. บางที อาจมีบางคน ต้องการ ำเนินงานในทันที แต่งานรออยู่ในคิวโดยมีระดับความสำคัญต่ำ. หรือบางทีคุณต้องการปฏิบัติการบำรุงรักษาบางอย่างบนระบบย่อย และต้องการย้ายงานทั้งหมดไปที่คิวซึ่งไม่ได้เชื่อมโยงกับระบบย่อยนั้น.

ข้อมูลต่อไปนี้อธิบายวิธีการดำเนินการงานการจัดการเหล่านี้.

การกำหนดคิวงานที่ระบบย่อย

ถ้าต้องการกำหนด job queue entry ที่ subsystem description, ให้ใช้ อินเตอร์เฟซแบบอักขระ.

คำสั่ง: Add Job Queue Entry (ADDJOBQE)

พารามิเตอร์บนคำสั่งนี้ระบุ:

- จำนวนของงานที่สามารถแอดคิฟในเวลาเดียวกันบนคิวงานนี้ (MAXACT)
- ลำดับที่ระบบย่อยจะจัดการกับงานจากคิวงานนี้ (SEQNBR)
- จำนวนงานที่สามารถแอดคิฟได้พร้อมกันสำหรับแต่ละระดับความสำคัญซึ่งมีเก้า ระดับ (MAXPTYn) (n=1 ถึง 9)

ตัวอย่าง: ตัวอย่างต่อไปนี้จะใส่เพิ่ม job queue entry สำหรับ คิวงาน JOBQA ใน TEST subsystem description. ไม่มีจำนวนสูงสุดของงานที่สามารถ แอดคิฟบนคิวงานนี้ และงานจะถูกประมวลผลโดยมี หมายเลขลำดับเป็นห้า.

```
ADDJOBQE SBSDB(TEST) JOBQ(LIBA/JOBQA) MAXACT(*NOMAX) SEQNBR(5)
```

ระบบย่อยจัดการกับหลายคิวงานได้อย่างไร:

เพื่อสาธิตวิธีการที่ระบบย่อยใช้ในการจัดการกับหลายคิวงาน, ให้พิจารณา สถานการณ์จำลองนี้.

Job Queue A (SEQNBR=10)

งาน 1

งาน 2

งาน 3

Job Queue B (SEQNBR=20)

งาน 4

งาน 5

งาน 6

Job Queue C (SEQNBR=30)

งาน 7

งาน 8

แต่ละ job queue entry ในสถานการณ์จำลองนี้ถูกระบุเป็น MAXACT(*NOMAX). ในอันดับแรก ระบบย่อยจะเลือกงานจากคิวงาน A เนื่องจาก job queue entry มีหมายเลขลำดับต่ำสุด. ถ้าจำนวนสูงสุดของงานในระบบย่อยคือ 3 (พารามิเตอร์ MAXJOBS(3) บนคำสั่ง Create Subsystem Description (CRTSBSD)), ระบบย่อยสามารถเลือกงานทั้งหมดจากคิวงาน A ให้แอสคิพีได้พร้อมกัน.

เมื่องานใดงานหนึ่งในสามงานนี้เสร็จสมบูรณ์แล้ว, activity level จะไม่สูงสุด อีกต่อไป; ดังนั้นงานใหม่หนึ่งงานจะถูกเลือกจากคิวงาน B เนื่องจาก คิวงานนี้มีหมายเลขลำดับต่ำสุดถัดไป (สมมุติว่าไม่มีการใส่เพิ่มงานใหม่ที่ คิวงาน A). เนื่องจากแต่ละ job queue entry ระบุ MAXACT(*NOMAX), ค่า MAXACT จึงไม่ได้ป้องกันการเริ่มต้นงาน. ถ้า แต่ละ job queue entry ระบุ MAXACT(1), ผลคืองาน 1, 4, และ 7 อาจจะถูก เริ่มต้น. ถ้า job queue entry A ถูกระบุเป็น MAXACT(2), ผลคือ งาน 1, 2, และ 4 อาจจะถูกเริ่มต้น.

การเปลี่ยนจำนวนของงานที่รันได้พร้อมกันในคิวงาน

ระบบย่อย QBASE ถูกจัดส่งมาพร้อมกับ job queue entry สำหรับคิวงาน QBATCH . entry นี้อนุญาตให้มีงานแบ็คชั้ที่จะรันได้เพียงหนึ่งงานเท่านั้นในแต่ละครั้ง. ถ้าคุณต้องการให้ สามารถรันงานแบ็คชั้ได้พร้อมกันหลายงานจากคิวงานนั้น คุณต้องเปลี่ยน job queue entry.

ถ้าต้องการเปลี่ยนจำนวนของงานที่รันได้พร้อมกันจากคิวงาน, ให้ใช้อินเตอร์เฟซแบบอักษร:

คำสั่ง: Change Job Queue Entry (CHGJOBQE)

ตัวอย่าง: คำสั่งต่อไปนี้อนุญาตให้รันงานแบ็คชั้ได้พร้อมกันสองงานจาก คิวงาน QBATCH ในระบบย่อย QBASE. (สามารถใช้คำสั่งนี้ได้ตลอดเวลาและคำสั่งมีผลบังคับใช้ทันที.)

```
CHGJOBQE SBSD(QBASE) JOBQ(QBATCH) MAXACT(2)
```

การล้างคิวงาน

เมื่อคุณล้างข้อมูลคิวงาน, งานทุกงานบนคิวจะถูกลบออก. รวมถึงงานใดๆ ที่อยู่ในสถานะถูกพักด้วย. คุณสามารถใช้ iSeries Navigator หรืออินเตอร์เฟซแบบอักษรในการล้างคิวงาน. งานที่กำลังรันอยู่จะไม่ได้รับผลกระทบ เนื่องจากถือว่างานเหล่านั้นเป็นงานแอสคิพีและไม่ได้อยู่บนคิว อีกต่อไป.

iSeries Navigator:

ถ้าต้องการใช้ iSeries Navigator ในการล้างคิวงาน, ให้ทำตามขั้นตอน เหล่านี้:

1. ขยาย My Connections → server → Job Queues → Active Job Queues หรือ All Job Queues.
2. คลิกขวาที่คิวงานและคลิก Clear. หน้าต่าง Confirm Clear จะปรากฏขึ้น ในหน้าต่างนี้คุณสามารถระบุได้ว่า ต้องการให้มีการจัดทำบันทึกการใช้งานเมื่อล้างคิวหรือไม่.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Clear Job Queue (CLRJOBQ)

ตัวอย่าง: คำสั่งนี้จะลบงานทั้งหมดที่มีอยู่ในปัจจุบันในคิวงานที่ได้มาจาก IBM, QBATCH. งานใดๆ ที่ถูกอ่านอยู่ในปัจจุบันจะไม่ได้รับผลกระทบ.

การสร้างคิวงาน

ถ้าต้องการสร้างคิวงาน, ให้ใช้อินเตอร์เฟซแบบอักษร.

คำสั่ง: Create Job Queue (CRTJOBQ)

ตัวอย่าง: ตัวอย่างต่อไปนี้จะสร้างคิวงานที่มีชื่อว่า JOBQA ในไลบรารี LIBA:

```
CRTJOBQ JOBQ(LIBA/JOBQA) TEXT('test job queue')
```

หลังจากที่คุณสร้างคิวงานแล้ว, ต้องกำหนดคิวงานที่ระบบย่อยก่อน จึงจะสามารถรันงานใดๆ ได้. เพื่อกำหนดคิวงานที่ระบบย่อย, ให้ใส่เพิ่ม job queue entry ที่ subsystem description.

การลบคิวงาน

ถ้าต้องการลบคิวงาน, ให้ใช้อินเตอร์เฟซแบบอักษร.

ข้อจำกัด:

- คิวงานที่กำลังจะลบไม่สามารถมี entries ใดๆ . งานทั้งหมดบนคิว ต้องเสร็จสมบูรณ์, ลบออก, หรือย้ายไปที่คิวงานอื่น.
- ระบบย่อยไม่สามารถแอนด์ที่ฟที่คิวงาน.

วิธีการลบคิวงานมีมากกว่าหนึ่งวิธี. แม้ว่าในที่นี้จะแสดงวิธีการลบ สองวิธี, แต่แนะนำให้ใช้วิธีการใช้คำสั่ง WRKJOBQ เนื่องจากวิธีนี้จะแสดงจำนวนและสถานะของงาน.

คำสั่ง: Work with Job Queue (WRKJOBQ)

ถ้าจำนวนของงานเป็น 0, คุณสามารถใช้อ็อปชัน 4=ลบออก เพื่อลบคิวงานออกจากไลบรารีได้.

ใช้ DLTJOBQ พร้อมกับ automated scripts และสภาวะแวดล้อมการล้างข้อมูล. ระวังเมื่อใช้วิธีการนี้เนื่องจากการดำเนินการรีโพลต์ของคำสั่งนี้คือ การค้นหารายชื่อไลบรารี และลบคิวงานแรกตรงกับชื่อที่ระบุ. ถ้าคุณมีคิวงานสองคิวที่มีชื่อเหมือนกันในไลบรารีต่างกัน, คุณอาจลบคิวงานที่ผิดออก. คุณสามารถแก้ไขการดำเนินการนี้โดยการ ระบุไลบรารีเฉพาะ.

คำสั่ง: Delete Job Queue (DLTJOBQ)

ตัวอย่าง: คำสั่งนี้จะลบคิวงาน SPECIALJQ ในไลบรารี SPECIALLIB.

```
DLTJOBQ JOBQ(SPECIALLIB/SPECIALJQ)
```

กำหนดว่าระบบย่อยใดมีคิวงานที่ถูกจัดสรร

คุณสามารถกำหนดว่าระบบย่อยใดมีคิวงานที่ถูกจัดสรรได้โดยใช้อินเตอร์เฟซ iSeries Navigator หรืออินเตอร์เฟซแบบอักษร. สิ่งนี้มีประโยชน์เมื่อคุณพบว่าจำเป็นต้อง ลบคิวงานออก เนื่องจากคุณไม่สามารถลบคิวงานซึ่งมี ระบบย่อยแอนด์ที่ฟ.

iSeries Navigator:

ถ้าต้องการดูว่าระบบย่อยใดมีคิวงานที่ถูกจัดสรร, ให้ทำตามนี้:

1. ใน iSeries Navigator ให้ขยาย Work Management → Job Queues → All Job Queues.

2. ระบุตำแหน่งคิวงานในระนาบด้านขวาของ iSeries Navigator. คุณจะเห็นระบบย่อยที่มีคิวงานที่ถูกจัดสรรในคอลัมน์ Subsystem .

(ถ้าคุณไม่เห็นคอลัมน์ Subsystem, ให้ใส่เพิ่มคอลัมน์ที่จอแสดงผล. คลิกขวา All Job Queues → Customize this view → Columns.)

3. หรือคุณสามารถคลิกขวาที่คิวงาน และคลิก **Properties**. ระบบย่อยจะถูกแสดงรายการบนหน้า General ของหน้าต่าง Job Queue Properties.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: WRKJOBQ JOBQ(LIBA/JOBQA) โดยที่ JOBQA คือชื่อของคิวงาน

1. พิมพ์คำสั่ง WRKJOBQ JOBQ(LIBA/JOBQA). จอแสดงผล Work with Job Queue จะปรากฏขึ้น. ฟังก์ชันคีย์ subsystem description จะปรากฏขึ้นในพื้นที่ฟังก์ชันคีย์ของจอแสดงผล เมื่อคิวงานถูกจัดสรรที่ระบบ.
2. กดฟังก์ชันคีย์ subsystem description. จอแสดงผล Work with Subsystem Descriptions จะปรากฏขึ้นเพื่อแสดงระบบย่อยซึ่งคิวงานถูกจัดสรร.

การพักคิวงาน

เมื่อคุณพักคิวงาน นั้นหมายถึงคุณป้องกันการประมวลผลงานทั้งหมดที่กำลังรออยู่บนคิวงานในปัจจุบัน. การพักคิวงาน ไม่มีผลต่องานที่กำลังรันอยู่. สามารถวางงานเพิ่มเติม บนคิวงานในขณะที่คิวงานถูกพักได้, แต่งานจะไม่ถูกประมวลผล.

เพื่อพักคิวงาน, คุณสามารถใช้ iSeries Navigator หรือใช้อินเตอร์เฟซแบบอักษร .

iSeries Navigator:

ใน iSeries Navigator, ให้ขยาย My Connections → server → Work Management → Job Queues → Active Job Queues → **คลิกขวาที่คิว** → Hold.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Hold Job Queue (HLDJOBQ)

ในตัวอย่างนี้, คิวงาน QBATCH จะถูกพัก. งานทั้งหมดที่ไม่ได้รันอยู่ในขณะที่ออกคำสั่งนี้จะถูกพักไว้ จนกว่า คิวจะถูกรีลีสหรือลบ.

```
HLDJOBQ JOBQ(QBATCH)
```

การรีลีสคิวงาน

เมื่อคุณรีลีสคิวงาน, งานทั้งหมดที่ถูกพักไว้ อันเป็นผลมาจาก การพักคิวงาน จะถูกรีลีสด้วย. ถ้า งานแต่ละงานถูกพักก่อนที่คิวงานจะถูกพัก, งานจะไม่ถูก รีลีส.

ถ้าต้องการรีลีสคิวงาน, ให้ใช้ iSeries Navigator หรืออินเตอร์เฟซแบบอักษร .

iSeries Navigator:

ใน iSeries Navigator, ให้ขยาย My Connections → server → Work Management → Job Queues → All Job Queues → **คลิกขวาที่คิว** → Release.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Release Job Queue (RLSJOBQ)

ตัวอย่างนี้จะรีลีสคิวงาน QBATCH.

RLSJOBQ JOBQ(QBATCH)

การย้ายงานไปที่คิวงานอื่น

มีหลายเหตุผลที่คุณอาจต้องการย้ายงานไปที่คิวอื่น. ตัวอย่างเช่น, ในบางครั้งงานอาจถูก backlogged ในคิวเนื่องจากเป็นงานที่ใช้เวลารันนาน. บางครั้งรันใหม่ที่จัดตารางเวลาของงานขัดแย้งกับงานใหม่ที่มีระดับความสำคัญสูงกว่า. วิธีหนึ่งในการจัดการสถานการณ์นี้คือ การย้ายงานที่รอไปยังคิวอื่นที่ไม่ยุ่ง.

คุณสามารถใช้ iSeries Navigator หรืออินเตอร์เฟซแบบอักษรอย่างใดอย่างหนึ่ง เพื่อย้ายงานจากคิวหนึ่งไปยังคิวอื่น.

iSeries Navigator:

ถ้าต้องการใช้ iSeries Navigator, ให้ทำตามต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **Work Management** → **Job Queues** → **All Job Queues**.
2. ระบุตำแหน่งและเปิดคิวที่มีงานอยู่ในปัจจุบัน.
3. คลิกขวานที่คิวที่ต้องการย้าย. หน้าต่าง Move ที่คุณสามารถระบุคิวเป้าหมายจะเปิดขึ้น.

หมายเหตุ: ถ้าคุณต้องการย้าย งานมากกว่าหนึ่งงานจากคิวนี้, ให้กดปุ่ม CTRL ค้างไว้ในขณะที่คุณคลิก แต่ละงาน. จากนั้นคลิกขวา แล้วคลิก Move.

- งานที่กำลังรอเพื่อรันจะถูกย้ายไปที่ตำแหน่งสัมพันธ์เดียวกันบนคิวเป้าหมาย (เช่น, งานที่มีระดับความสำคัญคิวงาน 3 จะถูกย้ายไปอยู่ หลังจากงานระดับความสำคัญ 3 อื่นที่กำลังรอเพื่อรันบนคิวเป้าหมาย).
- งานที่ถูกพักยังคงถูกพักอยู่และถูกวางในตำแหน่งสัมพันธ์เดียวกัน บนคิวเป้าหมาย (เช่น, งานที่พักที่มีระดับความสำคัญคิวงาน 3 จะถูกย้ายไปอยู่ หลังจากงานที่พักที่มีระดับความสำคัญ 3 อื่นบนคิวเป้าหมาย).
- งานที่ถูกจัดตารางเวลาเพื่อรันจะถูกย้ายไปที่คิวเป้าหมาย และ เวลาที่จัดตารางเวลาของงานยังคงไม่เปลี่ยนแปลง.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Change Job (CHGJOB)

ตัวอย่าง: ตัวอย่างต่อไปนี้จะย้ายงาน JOBA ไปที่คิวงาน JOBQB.

CHGJOB JOB(JOBA) JOBQ(LIBA/JOBQB)

การวางงานบนคิวงาน

งานถูกวางบนคิวงานโดยการย้ายงานที่มีอยู่จากคิวงานหนึ่งไปยังคิวงานอื่น, หรือโดยการส่งงานใหม่ อย่างใดอย่างหนึ่ง. ใช้ iSeries Navigator เพื่อย้ายงานระหว่างคิวต่างๆ. ใช้อินเตอร์เฟซแบบอักษรเพื่อส่งงานใหม่.

iSeries Navigator:

ในการใช้อินเตอร์เฟซ iSeries Navigator, งานต้องมีอยู่แล้วใน คิวงานอื่น. จากนั้นคุณสามารถย้ายงานจากคิวหนึ่งไปยังคิวงานอื่นได้. (ถ้าต้องการวางงานใหม่บนคิวงาน, ให้ใช้อินเตอร์เฟซบรรทัดรับคำสั่ง.)

1. ใน iSeries Navigator, ให้ขยาย **Work Management** → **Job Queues** → **All Job Queues**.

2. คลิกขวางานที่คุณต้องการย้าย. หน้าต่าง Move ที่คุณสามารถระบุคิวปลายทางจะเปิดขึ้น.

อินเตอร์เฟซแบบอักษร:

ข้อมูลต่อไปนี้เป็นรายการของวิธีการใช้อินเตอร์เฟซแบบอักษร สำหรับการวางงานใหม่บนคิวงานใหม่.

- Submit Job (SBMJOB): อนุญาตให้งานที่กำลังรันอยู่ ส่งงานอื่นไปที่คิวงานเพื่อรันในภายหลังเป็นงานแบ็คซ์. สามารถวางได้เพียงหนึ่งองค์ประกอบของข้อมูลค่าขอเท่านั้นบน message queue ของงานใหม่. ข้อมูลค่าขอสามารถเป็นคำสั่ง CL ถ้า routing entry ที่ใช้สำหรับงานระบุ โปรแกรมการประมวลผลคำสั่ง CL (เช่น โปรแกรม QCMD ที่ได้มาจาก IBM).
- Add Job Schedule Entry (ADDJOBSCDE): ระบบจะ ส่งงานโดยอัตโนมัติไปที่คิวงานในเวลาและวันที่ที่ระบุใน job schedule entry.
- Submit Database Jobs (SBMDBJOB): ส่งงาน ไปที่คิวงานเพื่อให้งานสามารถรันเป็นงานแบ็คซ์ได้. input stream ถูกอ่านจาก ไฟล์ฐานข้อมูลฟิสิกส์หรือจากไฟล์ฐานข้อมูลโลจิคัล ใดอย่างหนึ่ง ที่มีรูปแบบเร็คคอร์ดเดียว. คำสั่งนี้อนุญาตให้คุณระบุชื่อของไฟล์ฐานข้อมูลนี้และ เมมเบอร์ของไฟล์, ชื่อของคิวงานที่จะใช้, และเลือกตัดสินใจว่างานที่จะส่งสามารถถูกแสดงโดยคำสั่ง Work with Submitted Jobs (WRKSBMJOB) หรือไม่.
- Start Database Reader (STRDBRDR): อ่าน batch input stream จากฐานข้อมูลและวางหนึ่งหรือหลายงานบนคิวงาน.
- Transfer Job (TFRJOB): ย้ายงานปัจจุบันไปยัง คิวงานอื่นในระบบย่อยแอ็คทีฟ.
- Transfer Batch Job (TFRBCHJOB): ย้ายงานปัจจุบัน ไปยังคิวงานอื่น.

การค้นหาคิวงานทั้งหมดสำหรับงานเฉพาะ

คุณสามารถใช้ iSeries Navigator หรืออินเตอร์เฟซแบบอักษร ใดอย่างหนึ่ง เพื่อค้นหาคิวงานสำหรับงานเฉพาะ.

iSeries Navigator:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → **server** → **Basic Operations** → **คลิกขวา Jobs** → **Customize this View** → **Include**.
2. ใช้หน้าต่าง **Jobs-Include** เพื่อลดจำนวนของงานที่จะแสดง . ตรวจสอบให้แน่ใจว่าฟิลด์ **Job queue** ถูกเซ็ท เป็น **All**.
3. เมื่อคุณคลิก **OK** งานทั้งหมดที่ตรงกับเกณฑ์ของคุณ จะแสดงขึ้น.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work with Job Queues (WRKJOBQ)

ตัวอย่าง: ตัวอย่างต่อไปนี้จะสร้างรายการของงานทั้งหมดบนคิวงาน JOBQA.

```
WRKJOBQ JOBQ(LIBA/JOBQA)
```

การค้นหางานถ้าคุณไม่ทราบชื่อของคิวงาน:

ถ้าคุณไม่ทราบชื่อของคิวงาน, ให้ทำดังต่อไปนี้:

1. ป้อนคำสั่งโดยไม่ป้อนพารามิเตอร์ **JOBQ**. หน้าต่าง Work with All Job Queues จะปรากฏขึ้นพร้อมกับรายการของคิวงานทั้งหมด ซึ่งคุณได้รับสิทธิให้ใช้งานได้.
2. สแกนรายการนี้จนกว่าคุณจะเห็นชื่อของคิวงานที่อาจมีงาน ซึ่งคุณกำลังพยายามหาอยู่.

หลังจากที่คุณพบงานในคิวงานแล้ว, คุณสามารถดูที่งานนั้นได้ โดยการป้อนงานพร้อมกับอ็อปชันสำหรับงานที่คุณต้องการดู. จอแสดงผล Work with Job จะปรากฏขึ้น. จอแสดงผลนี้แสดงอ็อปชันต่างๆ สำหรับการดู ข้อมูลทั้งหมดที่พร้อมใช้งานสำหรับงานที่คุณได้เลือก.

ถ้าคุณทราบงานที่คุณจะค้นหา, คำสั่งต่อไปนี้จะนำคุณไปยังจอแสดงผลงาน โดยตรง.

WRKJOB JOB(หมายเลข/ผู้ใช้/ชื่อ) OPTION(*DFNA)

ถ้าคุณไม่แน่ใจเกี่ยวกับงานที่จะค้นหา, ให้ใช้คำสั่ง Work with Submitted Jobs (WRKSBJJOB) หรือ Work with User Jobs (WRKUSRJOB).

การระบุระดับความสำคัญสำหรับคิวงาน

ถ้าต้องการระบุลำดับการประมวลผลคิวงานโดยระบบย่อย, ให้ใช้อินเตอร์เฟสแบบอักษร.

คำสั่ง: Add Job Queue Entry (ADDJOBQE)

พารามิเตอร์บนคำสั่งนี้ระบุ:

- จำนวนของงานที่สามารถแอดคิพีในเวลาเดียวกันบนคิวงานนี้ (MAXACT)
- ลำดับที่ระบบย่อยจะจัดการกับงานจากคิวงานนี้ (SEQNBR)
- จำนวนงานที่สามารถแอดคิพีได้พร้อมกันสำหรับแต่ละระดับความสำคัญซึ่งมีเก้า ระดับ (MAXPTYn) (n=1 ถึง 9)

การจัดการเอาต์พุตคิว

เอาต์พุตคิวจะช่วยจัดการพรินเตอร์เอาต์พุตที่สร้างขึ้นเมื่อสิ้นสุดการทำงาน. เป็นสิ่งสำคัญที่ต้องทำความเข้าใจวิธีการรักษาเอาต์พุตคิวของคุณอย่างมีประสิทธิภาพ เพื่อให้เอาต์พุตที่พิมพ์ของคุณดำเนินไปอย่างราบรื่น.

พรินเตอร์เอาต์พุตอยู่บนเอาต์พุตคิว. เอาต์พุตคิวกำหนดลำดับที่พรินเตอร์เอาต์พุตจะถูกประมวลผลโดยอุปกรณ์การพิมพ์. โดยการจัดการเอาต์พุตคิวของคุณ, คุณสามารถมั่นใจถึงการประมวลผลที่ราบรื่นของ พรินเตอร์เอาต์พุตของคุณ.

การสร้างเอาต์พุตคิว

คำสั่ง Create Output Queue (CRTOUTQ) จะสร้าง เอาต์พุตคิวใหม่สำหรับสพูลไฟล์. entry จะถูกวางบนเอาต์พุตคิวสำหรับแต่ละสพูลไฟล์. ลำดับในการบันทึกไฟล์ไปที่อุปกรณ์เอาต์พุต ถูกกำหนดโดยระดับความสำคัญเอาต์พุตของสพูลไฟล์และค่าที่ระบุบน Order of files on queue prompt (พารามิเตอร์SEQ). ใช้อินเตอร์เฟสแบบอักษรในการสร้างเอาต์พุตคิว.

คำสั่ง: CRTOUTQ (Create Output Queue)

ตัวอย่าง: คำสั่งนี้จะสร้างเอาต์พุตคิวที่มีชื่อว่า DEPTAPRT และวางเอาต์พุตคิวในไลบรารีปัจจุบัน. เนื่องจากมีการระบุ AUT (*EXCLUDE) และสมมุติค่า OPRCTL(*YES), เอาต์พุตคิวจึงสามารถถูกใช้งานและควบคุมโดยผู้ใช้ ที่เป็นคนสร้างคิวและผู้ใช้ที่มีสิทธิควบคุมงาน หรือสิทธิควบคุมสพูลเท่านั้น. เนื่องจากมีการระบุ SEQ(*FIFO), สพูลไฟล์ จึงถูกวางในลำดับ first-in first-out (เข้าก่อนออกก่อน) บนคิว. ถ้าผู้ใช้ใน Department A ได้รับอนุญาตให้ใช้เอาต์พุตคิวนี้, ต้องใช้คำสั่ง Grant Object Authority (GRTOBJAUT) เพื่อมอบสิทธิในการใช้งานที่จำเป็นให้แก่ผู้ใช้. ข้อมูลที่มีอยู่ในไฟล์ต่างๆ บนคิวนี้อาจสามารถถูกแสดงโดยผู้ใช้ที่เป็นเจ้าของไฟล์, โดยเจ้าของ คิว, โดยผู้ใช้ที่มีสิทธิควบคุมงาน, หรือโดยผู้ใช้ซึ่งมีสิทธิ ควบคุมสพูลเท่านั้น. โดยค่าดีฟอลต์, จะไม่มีการพิมพ์ตัวแบ่งงานที่ตอนต้น ของเอาต์พุตสำหรับแต่ละงาน.

CRTOUTQ OUTQ(DEPTAPRT) AUT(*EXCLUDE) SEQ(*FIFO)
TEXT('SPECIAL PRINTER FILES FOR DEPTA')

ตัวอย่าง: ข้างล่างนี้เป็นอีกตัวอย่างหนึ่งของวิธีการสร้างเอาต์พุตคิว.

```
CRTOUTQ  OUTQ(QGPL/JONES) +  
          TEXT('Output queue for Mike Jones')
```

การกำหนดเอาต์พุตคิวที่งานหรือรายละเอียดของงาน

ก่อนที่คุณจะสามารถใช้เอาต์พุตคิวที่สร้างขึ้นใหม่ได้, คุณต้องกำหนด เอาต์พุตคิวที่งานหรือที่รายละเอียดของงาน. คุณสามารถกำหนดเอาต์พุตคิวโดยใช้ iSeries Navigator หรืออินเตอร์เฟซแบบอักษร.

iSeries Navigator:

ถ้าต้องการใช้ iSeries Navigator เพื่อกำหนดเอาต์พุตคิวที่งาน, ให้ทำตามต่อไปนี้.

ใน iSeries Navigator, ให้ขยาย **Work Management** → **Active Jobs** → **คลิกขวาที่งานและคลิก Properties** → **คลิกแท็บ Printer Output**.

อินเตอร์เฟซแบบอักษร:

คุณยังสามารถเปลี่ยนรายละเอียดของงานเพื่อใช้เอาต์พุตคิวใหม่ได้ด้วย. จากนั้น, งานทั้งหมดที่ใช้รายละเอียดของงานจะใช้เอาต์พุตคิวใหม่. ใช้อินเตอร์เฟซแบบอักษรเพื่อกำหนดเอาต์พุตคิวที่ รายละเอียดของงาน.

คำสั่ง: Change Job Description (CHGJOB)

ตัวอย่างต่อไปนี้จะเปลี่ยนรายละเอียดของงาน AMJOBS เพื่อใช้เอาต์พุตคิว QPRINT.

```
CHGJOB JOB(AMJOBS/AMJOBS) OUTQ(*LIBL/QPRINT)
```

การเข้าใช้งานพรินเตอร์เอาต์พุต

เนื่องจากคุณมีตัวเลือกในการดึงพรินเตอร์เอาต์พุตออกมาจากงาน เมื่องานรันเสร็จสิ้นแล้ว (การแยกพรินเตอร์เอาต์พุตออกจากงานอย่างสมบูรณ์), คุณสามารถเข้าใช้งานพรินเตอร์เอาต์พุตของคุณได้ใน iSeries Navigator ผ่านทาง Basic Operations หรือผ่านทางจัดการระบบงาน.

iSeries Navigator:

ถ้าต้องการเข้าใช้งานพรินเตอร์เอาต์พุตของงานผ่าน Basic Operations, ให้ทำตามต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → **server** → **Basic Operations** → **Job**.
2. **คลิกขวา**งานที่คุณต้องการแสดงพรินเตอร์เอาต์พุต และคลิก **Printer Output**. หน้าต่าง Printer Output จะ ปรากฏขึ้น.

ถ้าต้องการเข้าใช้งานพรินเตอร์เอาต์พุตผ่านทางโพลเดอร์ Output Queues, ให้ทำ ดังต่อไปนี้:

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

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work with Output Queue (WRKOUTQ <ชื่อเอาต์พุตคิว>)

คำสั่ง: Work with Spooled Files (WRKSPLF JOB(ชื่องานที่ถูกต้อง)

การล้างเอาต์พุตคิว

เมื่องานสร้างพรินเตอร์เอาต์พุต เอาต์พุตจะถูกส่งไปยังเอาต์พุตคิวเพื่อพิมพ์. ส่วนใหญ่แล้วคุณจะไม่พิมพ์พรินเตอร์เอาต์พุตทั้งหมด. iSeries Navigator ช่วยให้คุณสามารถล้างข้อมูลเอาต์พุตคิวของคุณโดยใช้ไอพชั่น Clear . การล้างเอาต์พุตคิวจะลบเอาต์พุตทั้งหมดออกจากคิว.

iSeries Navigator:

ถ้าต้องการล้างเอาต์พุตคิว, ให้ทำตามขั้นตอนเหล่านี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → **server** → **Work Management** → **Output Queues**.
2. คลิกขวาเอาต์พุตคิวที่คุณต้องการล้างข้อมูล, และคลิก **Clear**.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Clear Output Queue (CLROUTQ)

คำสั่งนี้จะลบ entries สำหรับสพูลไฟล์ทั้งหมดออกจากเอาต์พุตคิว, QPRINT, ที่กำลังรอที่จะพิมพ์หรือกำลังจะพัก. entries สำหรับไฟล์ที่กำลังพิมพ์ในปัจจุบันและไฟล์ที่ยังคงได้รับข้อมูลจากโปรแกรมที่กำลังรันอยู่ในปัจจุบันจะไม่ได้รับผลกระทบ.

```
CLROUTQ  OUTQ(QPRINT)
```

การลบเอาต์พุตคิว

คุณใช้อินเตอร์เฟซแบบอักษรในการลบเอาต์พุตคิว.

ก่อนที่จะสามารถลบเอาต์พุตคิวได้, เอาต์พุตคิวต้องเป็นไปตามข้อกำหนด ดังต่อไปนี้.

เอาต์พุตคิวที่กำลังจะลบไม่สามารถมี entries ใดๆ. เอาต์พุตสำหรับแต่ละไฟล์ต้องถูก พิมพ์, ลบออก, หรือย้ายไปที่เอาต์พุตคิวอื่น. ระบบย่อยไม่สามารถแอ็คทีฟ. คิวไม่สามารถถูกใช้โดย spooling writer. จะไม่สามารถลบคิวออก ถ้าคิวถูกสร้างโดยระบบสำหรับพรินเตอร์เฉพาะ.

คำสั่ง: Delete Output Queue (DLTOUTQ)

คำสั่งนี้จะลบเอาต์พุตคิว PUNCH2 ออกจากระบบ.

```
DLTOUTQ  OUTQ(PUNCH2)
```

การดูเอาต์พุตคิวบนระบบ

เอาต์พุตคิวกำหนดลำดับที่พรินเตอร์เอาต์พุตจะถูกส่งไปยังอุปกรณ์การพิมพ์. คุณสามารถดู เอาต์พุตคิวโดยใช้ iSeries Navigator.

ถ้าต้องการดูเอาต์พุตคิวบนระบบ, ให้ทำดังต่อไปนี้:

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

ใน iSeries Navigator, คุณสามารถปรับรายการของเอาต์พุตคิวที่คุณกำลังดูตามความต้องการของคุณได้โดยใช้หน้าต่าง Include. หน้าต่าง Include อนุญาตให้คุณใส่ข้อจำกัดของสิ่งที่จะแสดงใน iSeries Navigator. ตัวอย่างเช่น, คุณสามารถรัน Include เพื่อแสดงเพียงบางเอาต์พุตคิวได้.

ถ้าต้องการใช้ ฟังก์ชัน include, ให้คลิกเมนู View, แล้ว **Customize this View**.

การจัดการบันทึกการใช้งาน

งานส่วนใหญ่บน iSeries ของคุณมีบันทึกการใช้งานที่เชื่อมโยง. บันทึกการใช้งานบอกให้คุณ ทราบถึงสิ่งต่างๆ มากมาย เช่น งานจะเริ่มเมื่อไร, งานจะจบเมื่อไร, คำสั่งอะไรกำลังรันอยู่, หมายเหตุความล้มเหลวและข้อความแสดงความผิดพลาด. ข้อมูลนี้ช่วยให้คุณเห็นภาพว่าวัฏจักรการรันงานเป็นอย่างไร.

ข้อมูลต่อไปนี้อธิบายภารกิจหลายอย่างที่คุณสามารถปฏิบัติ เมื่อทำงานกับบันทึกการใช้งาน.

การจัดการเซิร์ฟเวอร์บันทึกการใช้งาน

ระบบย่อย QSYSWRK ควบคุมเซิร์ฟเวอร์บันทึกการใช้งาน. อย่างไรก็ตาม, มีบางภารกิจ ที่คุณสามารถปฏิบัติเพื่อปรับหรือจัดการเซิร์ฟเวอร์บันทึกการใช้งาน.

Reconfigure เซิร์ฟเวอร์บันทึกการใช้งาน:

เมื่อจัดส่งมา, เซิร์ฟเวอร์บันทึกการใช้งานจะรันใน QSYSWRK. QSYSWRK จะแอ็คทีฟอย่างต่อเนื่อง. เพื่อเพิ่มประสิทธิภาพ, คุณอาจต้อง reconfigure เซิร์ฟเวอร์ บันทึกการใช้งานของคุณเพื่อรันในระบบย่อยอื่น.

ถ้าต้องการ reconfigure เซิร์ฟเวอร์บันทึกการใช้งานเพื่อรันในระบบย่อยอื่น, ให้ใช้อินเตอร์เฟซแบบอักษรและทำตามขั้นตอนเหล่านี้.

1. ใส่เพิ่ม routing entry ที่เหมือนกับ entry จาก QSYSWRK ลงใน subsystem description ของคุณ. นี่คือ routing entry Seq Nbr 500, Program QWCJLSVR, Library QSYS, Compare Value 'QJOBLOGSVR', Start Pos 1.
2. เปลี่ยนคิวงานที่ระบุในรายละเอียดของงาน QJOBLOGSVR เป็นคิวงานที่มีอยู่ในระบบย่อยของคุณ.
3. ใส่เพิ่ม QJOBLOGAJ autostart job entry (พร้อมกับ routing entry, ถ้าจำเป็น) ลงในระบบย่อยของคุณ. การเพิ่มนี้จะทำให้เซิร์ฟเวอร์บันทึกการใช้งานเริ่มขึ้น โดยอัตโนมัติเมื่อระบบย่อยเริ่มขึ้น.
 - หรือถ้าคุณต้องการ, คุณสามารถแทนที่ autostart job entry ด้วยการเรียกคำสั่ง STRLOGSVR ในโปรแกรมเริ่มทำงาน.
4. การลบ QJOBLOGAJ autostart job entry ออกจาก QSYSWRK.

ดังเช่นในอีกตัวอย่างหนึ่งของ reconfiguring เซิร์ฟเวอร์บันทึกการใช้งาน, คุณสามารถใช้คำสั่ง Change Class (CHGCLS) เพื่อเปลี่ยน Run priority (RUNPTY) ที่ระบุในคลาส QJOBLOGSVR (ในไลบรารี QSYS).

```
CHGCLS CLS(QSYS/QJOBLOGSVR) RUNPTY(50)
```

การจบเซิร์ฟเวอร์บันทึกการใช้งาน:

ใช้คำสั่ง End Job Log Server (ENDLOGSVR) เพื่อจบเซิร์ฟเวอร์บันทึกการใช้งาน. เซิร์ฟเวอร์บันทึกการใช้งานจะบันทึกบันทึกการใช้งาน สำหรับงานที่อยู่ในสถานะคงค้างบันทึกการใช้งาน. ถ้ามีงานเซิร์ฟเวอร์บันทึกการใช้งาน ที่แอ็คทีฟอยู่มากกว่าหนึ่งงานในขณะนี้ออกคำสั่งนี้, งานเซิร์ฟเวอร์ บันทึกการใช้งานทั้งหมดจะถูกจบลง.

คุณต้องมีสิทธิ์พิเศษ job control (*JOBCTL) ในการใช้คำสั่งนี้.

สำคัญ: ถ้าคุณต้องการเพียงแต่หยุดการจัดทำบันทึกการใช้งาน เนื่องจาก, เช่น, บันทึกการใช้งานยาวมากหรือใช้รีซอร์สมากเกินไป, ให้ดูหัวข้อที่เกี่ยวข้อง *การหยุดการจัดทำบันทึกการใช้งาน*.

เมื่อใช้คำสั่ง ENDLOGSVR, คุณสามารถระบุได้ว่า คุณต้องการให้เซิร์ฟเวอร์จบแบบทันที (ไม่แนะนำ) หรือจบในลักษณะที่ควบคุม.

iSeries Navigator:

1. จาก iSeries Navigator, ให้คลิกขวาที่ระบบจุดปลายซึ่งเซิร์ฟเวอร์บันทึกการใช้งานกำลังรันอยู่ และคลิก **Run Command**.
2. ใน Command to run: ให้พิมพ์ ENDLOGSVR.
3. หน้าต่าง End Job Log Server แสดงขึ้นเพื่อช่วยคุณระบุ พารามิเตอร์ของคำสั่งนี้. กรอกข้อมูลในหน้าต่างให้สมบูรณ์และคลิก **OK**. หน้าต่างปิดและคุณจะถูกส่งกลับไปหน้าต่าง Run Command .
4. ขณะนี้คุณสามารถคลิก **OK** เพื่อรันคำสั่งในทันที, หรือคุณสามารถคลิก **Schedule** เพื่อจัดตารางเวลา ที่คุณต้องการให้คำสั่งรัน อย่างไม่อย่างหนึ่ง.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: End Job Log Server (ENDLOGSVR)

การเริ่มต้นเซิร์ฟเวอร์บันทึกการใช้งาน

โดยค่าดีฟอลต์, เซิร์ฟเวอร์ บันทึกการใช้งานจะเริ่มโดยอัตโนมัติเมื่อระบบย่อย QSYSWRK เริ่มต้น. คุณสามารถเริ่มเซิร์ฟเวอร์บันทึกการใช้งานด้วยตนเองได้โดยใช้คำสั่ง Start Job Log Server (STRLOGSVR) .

เมื่อคุณใช้คำสั่ง STRLOGSVR, คุณสามารถระบุ จำนวนของเซิร์ฟเวอร์บันทึกการใช้งานเพิ่มเติมที่คุณต้องการเริ่ม, หรือคุณสามารถ ปล่อยให้ระบบคำนวณจำนวนที่คุณต้องการได้. ถ้าจำนวนของเซิร์ฟเวอร์ที่ร้องขอ เกินกว่าจำนวนแอสต์ที่สูงสุดที่ใช้ได้, เฉพาะผลต่างระหว่างจำนวนสูงสุด และจำนวนปัจจุบันของเซิร์ฟเวอร์แอสต์ที่พ่านั้นจะถูกเริ่มต้น. จำนวนสูงสุดของ เซิร์ฟเวอร์บันทึกการใช้งานที่สามารถแอสต์หรืออยู่บนคิวงานในเวลาเดียวกันคือ 30.

iSeries Navigator:

ถ้าต้องการใช้ iSeries Navigator, ให้ทำตามต่อไปนี้:

1. จาก iSeries Navigator, คลิกขวา end point system ที่เซิร์ฟเวอร์บันทึกการใช้งานตั้งอยู่ และเลือก **Run Command**.
2. ในฟิลด์ **Command to run:** ให้พิมพ์ STRLOGSVR.
3. คลิก **Prompt**.
4. หน้าต่าง Start Job Log Server จะแสดงขึ้นเพื่อช่วยคุณระบุ พารามิเตอร์ของคำสั่งนี้. กรอกข้อมูลในหน้าต่างให้สมบูรณ์และคลิก **OK**. หน้าต่างปิดและคุณจะถูกส่งกลับไปหน้าต่าง Run Command .
5. ขณะนี้คุณสามารถคลิก **OK** เพื่อรันคำสั่งในทันที, หรือคุณสามารถคลิก **Schedule** เพื่อจัดตารางเวลา ที่คุณต้องการให้คำสั่งรัน อย่างไม่อย่างหนึ่ง.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Start Log Server (STRLOGSVR)

วิธีการแสดงบันทึกการใช้งาน

คุณสามารถดูบันทึกการใช้งานจากที่ใดก็ได้ภายในการจัดการระบบงานที่คุณเข้าไปใช้งาน, เช่น ผ่านทางพื้นที่ระบบย่อยหรือพื้นที่พูลหน่วย ความจำ. คุณสามารถใช้ iSeries Navigator หรืออินเทอร์เฟซแบบอักษรเพื่อแสดงบันทึกการใช้งานได้.

iSeries Navigator:

ถ้าต้องการเข้าใช้บันทึกการใช้งานสำหรับงานแอคทีฟหรืองานเซิร์ฟเวอร์, ให้ทำต่อไปนี้:

1. ใน iSeries Navigator, ให้ขยาย **My Connections** → **server** → **Work Management** → **Active Jobs** หรือ **Server Jobs**.
2. คลิกขวาบนงาน (เช่น, Qbatch) และคลิก **Job Log**. หากต้องการข้อมูลเพิ่มเติม, ให้อ้างอิงคำอธิบายในหน้าต่าง Job Log. ถ้าต้องการดูรายละเอียดเพิ่มเติมของข้อความ, ให้คลิกขวาบนข้อความและคลิก **Properties**. หน้าต่าง Message Properties แสดงข้อมูลข้อความอย่างละเอียด. หน้าต่างนี้ แสดงรายละเอียดของข้อความและคำอธิบายข้อความ. ข้อความอย่างละเอียดจะช่วยบอกข้อมูลเพื่อให้คุณสามารถแก้ไขปัญหาได้.

รายการต่อไปนี้อธิบายวิธีการเพิ่มเติมในการเข้าใช้งานบันทึกการใช้งาน:

- **Basic Operations** → **Printer**
- **Basic Operations** → **Jobs** → **คลิกขวาบนงาน** → **Printer Output**
- **Work Management** → **Active Jobs** → **คลิกขวาบนงาน** → **Printer Output**
- **Work Management** → **Output Queues**
- **Users and Groups** → **All Users** → **คลิกขวาบนผู้ใช้** → **User Objects** → **Printer Output**

อินเทอร์เฟซแบบอักษร:

- คำสั่ง: Work with Job Logs (WRKJOBLOG)
- คำสั่ง: Work with Job WRKJOB OPTION(*JOBLOG).
- คำสั่ง: Work with Job (WRKJOB) JOB (<ชื่องานที่ ถูกต้อง>) OPTION(*SPLF)
- คำสั่ง: Work with Output Queue (WRKOUTQ)
- คำสั่ง: Work with Spool File (WRKSPLF)

ถ้าคุณมีข้อมูลไม่เพียงพอที่จะใช้คำสั่งข้างต้น, คำสั่ง Work with User Jobs (WRKUSRJOB) หรือคำสั่ง Work with Submitted Jobs (WRKSBMJOB) อาจ มีประโยชน์.

จะทำอะไรเมื่อบันทึกการใช้งานไม่แสดงขึ้น

ใน iSeries Navigator, ถ้าต้องการค้นหาและแสดงบันทึกการใช้งาน, ไม่ว่าจะเป็งานแบ็ตช์หรืองานแบบโต้ตอบ, ให้คลิกขวาที่งาน แล้วคลิก Job log จาก เมนู. อย่างไรก็ตาม, ขึ้นอยู่กับสถานะของงานของคุณหรือวิธีการตั้งค่าบันทึกการใช้งานในรายละเอียดของงาน, บันทึกการใช้งานของคุณอาจอยู่ในเอาต์พุตคิว, หรืออาจอยู่ในสถานะการคงค้างบันทึกการใช้งาน หรืออาจไม่พร้อมใช้งาน.

ข้างล่างนี้คือขั้นตอนปฏิบัติถ้าเมนูออพชัน Job log ไม่พร้อมใช้งาน สำหรับงานของคุณ.

คำแนะนำ: เซ็ตการแสดงผลคอลลัมน์สำหรับ Active jobs (หรือ Server jobs) ให้รวมคอลลัมน์ Status. สิ่งนี้จะช่วยให้ทราบตำแหน่งที่จะค้นหา บันทึกการใช้งานของคุณได้รวดเร็วและง่ายขึ้น.

เพื่อเข้าถึงบันทึกการใช้งาน: My Connections → server → Work Management → Active Jobs หรือ Server Jobs → คลิกขวา ที่งานและเลือก Job log.

ถ้าเมนูอื่นพจน Job log ไม่พร้อมใช้งาน หรือถ้าคุณได้รับข้อความแสดงความ ผิดพลาดที่ระบุว่า โปรแกรมไม่สามารถดึงข้อมูลบันทึกการใช้งาน, ให้พิจารณาทำดัง ต่อไปนี้:

1. ตรวจสอบสถานะของงาน.

ทางเลือก	คำอธิบาย
กำลังรันอยู่	ตรวจสอบหน้าต่าง Job Properties – Job Log และตรวจให้แน่ใจว่ามีการเลือกกล่อง Produce a job log . ถ้าไม่ได้เลือกกล่องดังกล่าว, จะไม่มีการจัดทำบันทึกการใช้งาน.
จบแล้ว	งานนี้ไม่ได้จบในลักษณะ ปกติ. อาจเนื่องจากข้อผิดพลาดหรือการยุติโดยผู้ใช้. คลิกขวาบนงาน แล้วคลิกPrinter Output. ถ้าคุณไม่เห็นบันทึกการใช้งาน ของคุณที่นั่น, ให้ตรวจสอบหน้าต่าง Job Properties – Job Log และตรวจให้แน่ใจว่ามีการเลือกเช็คบ็อกซ์, Produce printer output for job log.
สมบูรณ์ – พรินเตอร์เอาต์พุตพร้อมใช้งาน	งานนี้จบลงตามปกติ. คลิกขวาบนงาน แล้วคลิกPrinter Output. ถ้าคุณไม่เห็นบันทึกการใช้งานของคุณที่นั่น, ให้ตรวจสอบหน้าต่าง Job Properties – Job Log และตรวจให้แน่ใจว่ามีการเลือกฟิลด์ Create printer output for job log if job ends normally .
สมบูรณ์ – คงค้างบันทึกการใช้งาน	จะไม่มีการจัดทำ บันทึกการใช้งาน. บันทึกการใช้งานยังคงค้างอยู่จนกว่าจะถูกลบออก. คุณจะต้องใช้ คำสั่ง Display Job Log (DSPJOBLOG) เพื่อดูบันทึกการใช้งาน คงค้าง.

2. บันทึกการใช้งานอาจถูกสฟูลไปที่เอาต์พุตคิวและพิมพ์แล้ว, ซึ่งทำให้บันทึกถูกลบออกจากระบบ.

3. อีกสาเหตุหนึ่งที่น่าจะเป็นไปได้คือบันทึกการใช้งานได้ถูกลบออกโดย ผู้ใช้อื่น.

การระบุเอาต์พุตคิวสำหรับบันทึกการใช้งาน

โดยค่าดีฟอลต์ ไฟล์พรินเตอร์ที่จะใช้ในการสฟูลบันทึกการใช้งานคือ QPJOBLOG. คุณสามารถมีไฟล์พรินเตอร์ QPJOBLOG ได้หลายไฟล์บนระบบของคุณ. ใน QSYS เอาต์พุตคิวที่แอดทริบิวต์ OUTQ จะใช้คือ QEZJOBLOG, ในไลบรารี QUSRSYS. เมื่อระบบสร้างบันทึกการใช้งาน, ระบบจะค้นหาไฟล์พรินเตอร์ QPJOBLOG ในรายชื่อไลบรารีของงาน. ไฟล์แรกทีพบคือไฟล์ที่ระบบจะใช้. คุณใช้อินเตอร์เฟสแบบอักษรในการปรับเปลี่ยนการตั้งค่าเหล่านี้.

1. การเปลี่ยนแอดทริบิวต์ไฟล์พรินเตอร์ QPJOBLOG OUTQ เป็น *JOB.
 - a. คำสั่ง: Change Printer File CHGPRTF FILE(QPJOBLOG) OUTQ(*JOB)
2. การเปลี่ยนแอดทริบิวต์ OUTQ ของงานเป็นเอาต์พุตคิวที่คุณต้องการ. คุณสามารถทำสิ่งนี้ได้โดยการใช้อินเตอร์เฟสแบบอักษรหรือ iSeries Navigator.
 - a. คำสั่ง: Change Job CHGJOB OUTQ(MYLIB/MYOUTQ)
 - b. iSeries Navigator: Work Management → Active Jobs → คลิกขวาที่ งานและเลือก Properties → Printer Tab

การหยุดการจัดทำบันทึกการใช้งานเฉพาะ

ถ้าคุณต้องการหยุดการจัดทำบันทึกการใช้งานเฉพาะหนึ่ง, ห้ามใช้คำสั่ง End Job Log Server (ENDLOGSVR) . เนื่องจากคำสั่ง ENDLOGSVR จะจบเซิร์ฟเวอร์บันทึกการใช้งาน ทั้งหมดซึ่งส่งผลให้หยุดการจัดทำบันทึกการใช้งานทั้งหมด.

แต่, ให้ใช้ขั้นตอนต่อไปนี้ในการหยุดการจัดทำบันทึกการใช้งาน เฉพาะแทน.

1. จาก iSeries Navigator, คลิกขวางานที่คุณต้องการหยุดการจัดทำบันทึกการใช้งานและคลิก **Properties**. (My Connections → server → Work Management → Active Jobs หรือ Server Jobs)
2. คลิกแท็บ **Job Log**.
3. ยกเลิกการเลือกกล่อง **Produce a job log** และคลิก **OK**.

การจัดทำบันทึกการใช้งานจะยุติลงและบันทึกการใช้งานจะอยู่ในสถานะ บันทึกการใช้งานคงค้าง.

การป้องกันการจัดทำบันทึกการใช้งาน

การป้องกันการจัดทำบันทึกการใช้งานเป็นสิ่งที่มีความสำคัญ หากคุณทราบอยู่แล้วว่า คุณจะไม่ต้องการบันทึกการใช้งานและคุณต้องการสแกนรีเซ็ตของระบบ. เมื่อคุณระบุว่าคุณไม่ต้องการให้จัดทำบันทึกการใช้งาน, บันทึกการใช้งานจะไม่ถูกจัดทำและยังคงคงค้างอยู่จนกว่าจะถูกลบออกโดยคำสั่ง **Remove Pending Job Log (QWTRMVJL)** หรือคำสั่ง **End Job (ENDJOB)** อย่างเป็นทางการหนึ่ง.

ถ้าต้องการป้องกันการจัดทำบันทึกการใช้งาน, ให้ทำตามต่อไปนี้:

1. ใน iSeries Navigator ให้เปิดหน้าต่าง **Job Properties – Job Log**. (My Connections → server → Work Management → Active Jobs (หรือ System Jobs) → คลิกขวางาน → Properties → แท็บ Job Log)
2. ยกเลิกการเลือกกล่อง **Produce a job log** และคลิก **OK**.

การควบคุมข้อมูลในบันทึกการใช้งาน

เมื่อจัดการกับปัญหา, คุณอาจต้องการบันทึกจำนวนสูงสุดของ ข้อมูลสำหรับงานที่เกิดปัญหาขึ้นบ่อย. อีกทางหนึ่งคือ, คุณอาจไม่ต้องการสร้างบันทึกการใช้งานสำหรับงานที่จบลงตามปกติ. หรือ คุณอาจไม่ต้องการรวมข้อมูล.

คุณสามารถควบคุมข้อมูลที่จะใส่เพิ่มลงในบันทึกการใช้งานได้โดยการตั้งค่า ระดับข้อความ, ความรุนแรงของข้อความ หรือ ค่าระดับเนื้อความใน รายละเอียดของงาน. อย่างไรก็ตาม, ถ้าคุณต้องการควบคุมข้อมูลที่จะบันทึกลงใน บันทึกการใช้งานของงานเฉพาะ, ให้ใช้หน้าต่าง **Job Properties – Job Log** ใน iSeries Navigator.

หน้าต่างนี้อำนวยความสะดวกให้คุณ ควบคุมสิ่งต่างๆ ดังนี้:

- จะจัดทำบันทึกการใช้งานหรือไม่ และจัดทำบันทึกการใช้งานด้วยวิธีการใด
- จะทำอะไรเมื่อใช้งานถึงขนาดสูงสุด
- จะลือคำสั่งจากโปรแกรม CL หรือไม่
- จะเก็บรักษาข้อความในบันทึกการใช้งานหรือไม่ และข้อความอะไรที่ควร เก็บรักษาไว้ (ระดับการลือและความรุนแรงของข้อความ)
- จะสร้างพรินเตอร์เอาต์พุตสำหรับบันทึกการใช้งานหรือไม่ถ้างานจบลงอย่างปกติ และจะพิมพ์อะไร

ถ้าต้องการเข้าใช้งานหน้าต่าง Job Properties – Job Log ให้ทำตามต่อไปนี้:

1. จาก iSeries Navigator ให้เปิดหน้าต่าง Job Properties ของงาน และคลิกแท็บ **Job Log**. My Connections → server → Work Management → Active Job → คลิกขวางาน → Properties.
2. สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับอ็อปชันต่างๆ ที่พร้อมใช้บนหน้าต่างนี้, ให้อ้างอิงคำอธิบายทางออนไลน์.

การเปลี่ยนระดับไฟล์บันทึกของงาน:

ระดับไฟล์บันทึกของงานคือระดับตัวเลขที่กำหนดให้กับ ชนิดข้อความที่จะบันทึก. คุณสามารถเปลี่ยนระดับไฟล์บันทึก ในรายละเอียดของงานได้โดยการใช้อินเตอร์เฟซแบบอักษร. อย่างไรก็ตาม, ถ้า คุณต้องการเปลี่ยนระดับไฟล์บันทึกของงานเฉพาะ, ให้ใช้หน้าต่าง **Job Properties – Job Log** ใน iSeries Navigator.

ถ้าต้องการเข้าใช้งานหน้าต่าง **Job Properties – Job Log** ให้ทำดังต่อไปนี้:

1. จาก iSeries Navigator ให้เปิดหน้าต่าง **Job Properties** ของงานและคลิกแท็บ **Job Log**. (**My Connections** → **server** → **Work Management** → **Active Job** → **คลิกขวาที่งาน** → **Properties**)
2. สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับอ็อปชันที่พร้อมใช้บนหน้าต่างนี้, ให้อ้างอิงคำอธิบายทางออนไลน์.

การควบคุมข้อมูลบันทึกการใช้งานแบ็ตซ์:

สำหรับแบ็ตซ์แอ็พพลิเคชัน, คุณอาจต้องการเปลี่ยนจำนวนของข้อมูลที่บันทึก. ระดับไฟล์บันทึก (LOG(40 *NOLIST)) ที่ระบุในรายละเอียดของงาน สำหรับระบบย่อยที่ได้มาจาก IBM QBATCH นับเป็นไฟล์บันทึกที่สมบูรณ์ ถ้างานจบลงอย่างผิดปกติ. ถ้างานจบลงอย่างปกติ, จะไม่มีการจัดทำบันทึกการใช้งาน.

สามารถควบคุมบันทึกการใช้งานที่ระดับคิวงาน (QBATCH) โดยการปรับ การตั้งค่าบันทึกการใช้งานสำหรับงานระบบย่อย QBATCH. อ็อปชันในการควบคุมวิธีการจัดทำ บันทึกการใช้งานที่ระดับงานระบบย่อยจะเหมือนกับอ็อปชันในการควบคุมบันทึกการใช้งานที่ระดับงานปกติ.

ถ้าต้องการปรับเปลี่ยนการตั้งค่าบันทึกการใช้งานสำหรับระบบย่อยคิวงาน, ให้ทำดังต่อไปนี้:

ใน iSeries Navigator ให้เปิดหน้าต่าง **Subsystem Properties – Job Log** สำหรับ ระบบย่อยคิวงาน. (**Work Management** → **Subsystems** → **Active Subsystems** → **QBATCH** → **คลิกขวาที่งาน QBATCH** → **Properties** → **แท็บ Job Log**)

หมายเหตุ: หากคุณยกเลิกการเลือกฟิลด์ **Produce a job log field (*PND)** สำหรับระบบย่อย, บันทึกการใช้งานที่ระบุที่ระบบย่อย จะไม่ถูกแสดงรายการพร้อมกับพริ้นเตอร์เอาต์พุตอื่น. จากนั้น คุณจะต้องใช้คำสั่ง **Display Job Log (DSPJOBLOG)** เพื่อดูบันทึกการใช้งาน ที่คงค้าง.

ถ้างานแบ็ตซ์กำลังรันโปรแกรม CL, คำสั่งโปรแกรม CL จะถูกบันทึกเฉพาะเมื่อมีการระบุ LOGCLPGM(*YES) บนคำสั่ง **Create Control Language Program (CRTCLPGM)** หรือคำสั่ง **Change Program (CHGPGM)**.

การลบไฟล์เอาต์พุตบันทึกการใช้งาน

บันทึกการใช้งานจะถูกลบออกจากระบบเมื่องานเสร็จสมบูรณ์ตามปกติ, หรือเมื่อใช้คำสั่ง **Remove Pending Job Log (QWTRMVJL) API** หรือ **End Job (ENDJOB)**. นอกจากนี้ ถ้ามีการระบุ "clear incomplete job logs" บน IPL, งานทั้งหมดในบันทึกการใช้งานที่คงค้างจะถูกลบออกจากระบบ ในระหว่างการ IPL. สามารถดูไฟล์เอาต์พุตบันทึกการใช้งานที่เหลือได้ข้างใต้ **Basic Operations** → **Printer Output**.

เพื่อลบบันทึกการใช้งานที่พบใน Printer Output, ให้คลิกขวาที่ ชื่อไฟล์ของบันทึกการใช้งานที่คุณต้องการลบออก และคลิก **Delete**.

วิธีการพิจารณาว่าการลบบันทึกการใช้งานมีความปลอดภัยหรือไม่

การตัดสินใจว่าจะเก็บรักษบันทึกการใช้งานหรือจะลบออกถือเป็นสิ่งที่ท้าทาย. บันทึกการใช้งาน คือสิ่งที่คุณจำเป็นต้องเก็บไว้ เพื่อให้คุณสามารถแก้ไขปัญหาได้. บันทึกการใช้งาน คือสิ่งที่คุณไม่ต้องการเก็บเนื่องจากทำให้ระบบของคุณยุ่งเหยิง. เมื่อตัดสินใจว่าจะลบบันทึกการใช้งานใด, หรือจะป้องกันการลบบันทึกการใช้งานใด, ให้พิจารณาแนวทางดังต่อไปนี้:

- นี่เป็นงานที่คุณสามารถแก้ไขได้ง่ายโดยไม่ต้องดูที่บันทึกการใช้งานหรือไม่?
- นี่เป็นงานที่คล้ายกับงานอื่นในระบบหรือไม่? ถ้างานล้มเหลว, งานที่คล้ายกันอื่นจะล้มเหลวด้วยหรือไม่? ถ้าใช่, คุณอาจต้องเก็บงานที่สร้างบันทึกการใช้งาน เพียงงานเดียวเท่านั้น.

การจัดทำพรินเตอร์เอาต์พุตจากบันทึกการใช้งานที่ค้าง

งานที่ไม่มีการตั้งค่า iSeries Navigator **Job Properties – Job Log**, ฟิลด์ **Produce a job log** ที่เลือก จะไม่จัดทำบันทึกการใช้งาน. แต่บันทึกการใช้งานจะอยู่ในบันทึกการใช้งานที่ค้าง. ถ้าต้องการจัดทำพรินเตอร์เอาต์พุตจากบันทึกการใช้งานที่อยู่ในบันทึกการใช้งานที่ค้าง, ให้ใช้อินเตอร์เฟซแบบอักษร.

คำสั่ง: Display Job Log (DSPJOBLOG)

การล้างบันทึกการใช้งานที่ค้าง

การล้างข้อมูล, หรือการลบงานออกจากบันทึกการใช้งานที่ค้างสามารถทำได้หลายวิธี. คุณสามารถจบงานด้วยค่าของ 0 สำหรับพารามิเตอร์ **Maximum log entries (LOGMLT)**. ถ้างานจบลงแล้ว, คุณสามารถรัน **Remove Pending Job Log (QWTRMVJL)** API. และคุณยังสามารถใช้คำสั่ง **Work with Job Logs (WRKJOBLOG)** ได้อีกด้วย.

ถ้าต้องการจบงานด้วยการเซต **LOGMLT** เป็น 0, ให้ใช้ iSeries Navigator หรืออินเตอร์เฟซแบบอักษร.

iSeries Navigator:

1. ใน iSeries Navigator, ให้ขยาย **Work Management** → **Active Jobs**.
2. ระบุตำแหน่งงานที่คุณต้องการจบ.
3. คลิกขวาที่งานและคลิก **Delete/End**.
4. บนหน้าต่าง **Confirm Delete/End**, ให้เซต **Delete printer output** เป็น No.
5. จัดการหน้าต่าง **Confirm Delete/End** ให้สมบูรณ์ และคลิก **Delete**.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: End Job (ENDJOB LOGMLT(0))

การจัดการ job accounting

ฟังก์ชัน job accounting ไม่แอคทีฟโดยค่าดีฟอลต์. ต้องใช้ขั้นตอนเริ่มต้น บางอย่างเพื่อตั้งค่าฟังก์ชันนี้. ข้อมูลต่อไปนี้อธิบายวิธีการตั้งค่า job accounting และปฏิบัติการกิจทั่วไปส่วนใหญ่ที่เกี่ยวข้องกับ job accounting.

การตั้งค่า job accounting

ถ้าต้องการตั้งค่า job accounting, ให้ใช้อินเตอร์เฟซแบบอักษร.

1. การสร้าง journal receiver. สามารถสร้าง journal receiver ด้วยชื่อ และไลบรารีใดๆ ที่คุณเลือก. ขอแนะนำให้ใช้ชื่อตามหลักการตั้งชื่อ เช่น **ACGJRN1** เพื่อให้สามารถสร้าง receivers เพิ่มเติม (เช่น **ACGJRN2**, **ACGJRN3**) ด้วยคำสั่ง **Change Journal CHGJRN JRNRCV(*GEN)**.

- a. คำสั่ง: Create Journal Receiver (CRTJRNRCV)

```
CRTJRNRCV JRNRCV(USERLIB/ACGJRN1)
```

2. การสร้าง job accounting journal. ชื่อเจอร์นัลต้องเป็น QSYS/QACGJRN, และคุณต้องมีสิทธิ์ในการใส่เพิ่มอ็อบเจกต์ที่ไลบรารี QSYS.

a. คำสั่ง: Create Journal (CRTJRN)

```
CRTJRN JRN(QSYS/QACGJRN) JRNRCV(USERLIB/ACGJRN1) AUT(*EXCLUDE)
```

journal receiver ควรเหมือนกับ receiver ที่สร้างในขั้นตอน 1. สามารถเซตสิทธิ์เป็นสิ่งที่ใด ๆ ที่คุณเลือก, แต่แนะนำให้ใช้ *EXCLUDE เนื่องจากสามารถใช้ข้อมูลที่รวบรวมเพื่อคิดค่าธรรมเนียมสำหรับการใช้รีซอร์สจากผู้ใช้ได้.

3. การเปลี่ยนค่ากำหนดของระบบ journal accounting information (QACGLVL). สามารถเซตค่ากำหนดของระบบเป็นข้อมูล journal job accounting, หรือข้อมูล พรินเตอร์, หรือทั้งสองอย่าง. *JOB จะจัดทำ job (JB) journal entries, ในขณะที่ *PRINT จะจัดทำ direct print (DP) หรือ spooled print (SP) journal entries. ค่า *NONE หมายความว่า ไม่มีการทำเจอร์นัลสำหรับเจอร์นัล QACGJRN. ข้อมูล Job accounting จะถูกเจอร์นัล เฉพาะสำหรับงานที่เริ่มหลังจากที่ได้เซตค่ากำหนดของระบบเป็นค่าอื่นที่ไม่ใช่ *NONE แล้วเท่านั้น.

a. คำสั่ง: Work with System Values (WRKSYSVAL) หรือ Change System Value (CHGSYSVAL)

```
CHGSYSVAL SYSVAL(QACGLVL) VALUE('*JOB *PRINT')
```

4. เซ็ตพารามิเตอร์ accounting code ACGCDE สำหรับ แต่ละโปรไฟล์ผู้ใช้. สามารถเซต accounting code เป็นสตริงตัวอักษรผสมตัวเลข ที่ยาวได้ถึง 15 อักขระ. หากการกำหนดผู้ใช้ปัจจุบันเป็นสิ่งที่สำคัญสำหรับ การวิเคราะห์ job accounting journal entry ของคุณ, ขอแนะนำให้เซต พารามิเตอร์ ACGCDE เป็นชื่อของโปรไฟล์ผู้ใช้.

a. คำสั่ง: Change User Profile (CHGUSRPRF) หรือ Create User Profile (CRTUSRPRF)

```
CHGUSRPRF USRPRF(USERID1) ACGCDE(USERID1)
```

และยังสามารถระบุ accounting code สำหรับกลุ่มของผู้ใช้ได้โดยใช้คำสั่ง Change Job Description (CHGJOBDD) หรือ Create Job Description (CRTJOBDD).

accounting code ดีฟอลต์สำหรับ รายละเอียดของงานคือ *USRPRF, ซึ่งหมายความว่าระบบจะใช้ accounting code จากโปรไฟล์ผู้ใช้ของงาน. ถ้ามีการระบุค่าอื่นที่ไม่ใช่ *USRPRF ในรายละเอียดของงาน, ค่านั้นจะนำหน้า accounting code ที่ระบุ ในโปรไฟล์ผู้ใช้

สามารถเขียนโปรแกรมพื้นฐานเพื่อเปลี่ยนโปรไฟล์ผู้ใช้ทั้งหมด บนระบบ เพื่อให้ accounting code ของโปรไฟล์ตรงกับชื่อโปรไฟล์ผู้ใช้.

การควบคุมการกำหนดโค้ดบัญชีผู้ใช้

ส่วนที่สำคัญของแอปพลิเคชันการประมวลผลข้อมูลใดๆ คือ การตรวจสอบให้มั่นใจว่ามีการระบุฟิลด์การควบคุมที่ถูกต้อง. สำหรับ job accounting codes, อาจต้องใช้ ฟังก์ชันการตรวจสอบความถูกต้องที่ซับซ้อน ซึ่งไม่เพียงแต่ตรวจสอบเฉพาะ การมีอยู่จริงของโค้ดการพิสูจน์ตัวตนเท่านั้น, แต่ยังตรวจสอบผู้ใช้ที่ได้รับอนุญาต ให้ใช้โค้ดที่ระบุด้วย.

สามารถกำหนดโค้ดบัญชีผู้ใช้ได้ในพื้นที่ดังต่อไปนี้:

- โปรไฟล์ผู้ใช้
- รายละเอียดของงาน
- ในงาน (คำสั่ง Change Accounting Code (CHGACGCDE))

ถ้าจำเป็นต้องควบคุมการกำหนดโค้ดบัญชีผู้ใช้, ให้พิจารณาข้อมูลดังต่อไปนี้:

1. ก่อนการป้อนโค้ดบัญชีผู้ใช้ในโปรไฟล์ผู้ใช้, ให้ตรวจให้แน่ใจว่า โค้ดถูกต้องสำหรับผู้ใช้.

2. ควบคุมการเปลี่ยนโค้ดบัญชีผู้ใช้บนคำสั่ง Change Job Description (CHGJOB) โดยป้อนเฉพาะสิทธิเจ้าหน้าที่ดูแลความปลอดภัย ที่คำสั่ง CHGACGCDE .
 - หรือ, ใช้คำสั่ง CHGACGCDE เพื่ออนุญาตให้ผู้ใช้เปลี่ยน job accounting code ของตนเองหรืองานอื่น. ในการเปลี่ยนงานอื่น, ผู้ใช้ต้องมีสิทธิพิเศษ *JOBCTL ด้วย.
3. ใช้โปรแกรม CL และคำสั่งเพื่อป้องกันการเปลี่ยนโค้ดบัญชีผู้ใช้สำหรับงานบนคิวงาน หรือป้องกันไม่ให้งานหนึ่งเปลี่ยนโค้ดบัญชีผู้ใช้ของงานอื่น. ตัวอย่างเช่น, ควรให้สิทธิในคำสั่ง CHGACGCDE เป็นการส่วนตัว และรวมอยู่ในโปรแกรม CL ที่มีการเปลี่ยนเฉพาะงาน ปัจจุบันเท่านั้น (เช่น เมื่อมีการระบุ JOB(*)). คำสั่งจะถูกอนุญาตให้ใช้อย่างเหมาะสม.

การแสดงผลข้อมูลที่รวบรวม

หลังจากการรวบรวมข้อมูลในเจอร์นัล job accounting, คุณต้องบันทึก journal entries ไปที่ไฟล์และแสดงผล.

ถ้าต้องการทำเช่นนี้, ให้ปฏิบัติตามขั้นตอนเหล่านี้:

หมายเหตุ: ในตัวอย่างต่อไปนี้, ชื่อเจอร์นัล job accounting คือ QACGJRN.

1. สร้างก๊อปปีของ model outfile ที่ได้มาจากระบบสำหรับ accounting journal. QAJBACG4 คือ model outfile สำหรับรูปแบบ *TYPE4 outfile.
 - a. คำสั่ง: Create Duplicate Object (CRTDUPOBJ)


```
CRTDUPOBJ OBJ(QAJBACG4) FROMLIB(QSYS) OBJTYPE(*FILE) TOLIB(QTEMP)
NEWOBJ(MYJBACG4)
```
2. ดัมพ์ journal entries ไปที่ outfile ที่คุณเพิ่งสร้างขึ้น. ในตัวอย่างต่อไปนี้ เฉพาะ 'JB' หรือ job type journal entries เท่านั้นที่จะถูก ดัมพ์.
 - a. คำสั่ง: Display Journal (DSPJRN)


```
DSPJRN JRN(QACGJRN) ENTYP(JB) OUTPUT(*OUTFILE) OUTFILFMT(*TYPE4)
OUTFILE(QTEMP/MYJBACG4)
```
3. เริ่ม SQL session. จากนั้นใช้คำสั่ง SELECT จากภายใน SQL session เพื่อเลือกฟิลด์ที่คุณต้องการแสดง.
 - a. คำสั่ง: Start Structured Query Language (STRSQL)


```
STRSQL
SELECT JAJOB, JAUSER, JAUSPF, JACDE, JACPU FROM QTEMP/MYJBACG4
```

คุณสามารถแสดงรายการของชื่อฟิลด์แบบโต้ตอบ หรือแสดงไปที่ไฟล์ โดยการสร้างและรันเคียวรีโดยใช้คำสั่ง Work with Queries (WRKQRY).

การแปลง job accounting journal entries

คุณสามารถใช้พารามิเตอร์ **OUTFILE** บนคำสั่ง Display Journal (DSPJRN) เพื่อบันทึก job accounting journal entries เข้าในไฟล์ฐานข้อมูลที่คุณสามารถประมวลผลได้.

พารามิเตอร์ **OUTFILE** ช่วยให้คุณสามารถตั้งชื่อ ไฟล์หรือเมมเบอร์ได้. ถ้ามีเมมเบอร์อยู่, เมมเบอร์จะถูกลบออกก่อนที่จะบันทึก เร็กคอร์ด. ถ้าไม่มีเมมเบอร์อยู่, เมมเบอร์จะถูกใส่เพิ่ม. ถ้าไฟล์ไม่มีอยู่, ไฟล์จะถูกสร้างโดยใช้รูปแบบเร็กคอร์ด QJORDJE. รูปแบบนี้กำหนดฟิลด์หัวข้อ มาตรฐานสำหรับแต่ละ journal entry, แต่ข้อมูล job accounting จะถูกกำหนดเป็นฟิลด์ขนาดใหญ่หนึ่งฟิลด์.

เพื่อหลีกเลี่ยงการที่ต้องประมวลผลข้อมูล accounting เป็นฟิลด์ขนาดใหญ่หนึ่งฟิลด์, จึงได้มีการจัดไฟล์อ้างอิงฟิลด์สองไฟล์ เพื่อช่วยคุณในการประมวลผล job accounting journal entries. ไฟล์ QSYS/QAJBACG4 มีรูปแบบเร็กคอร์ด QWTJAJBE และใช้สำหรับ JB entries. ไฟล์ QSYS/QAPTACG5 มีรูปแบบเร็กคอร์ด QSPJAPTE และใช้สำหรับ DP หรือ SP entries. ใช้รูปแบบเดียวกันสำหรับ printer file entries ทั้งหมดโดยไม่คำนึงว่าเอาต์พุตจะเป็น SP (spooled) หรือ DP (nonspooled). DP entry สำหรับไฟล์ที่พิมพ์โดยตรงมีฟิลด์บางฟิลด์ที่ไม่ได้ใช้; ฟิลด์เหล่านี้ มีพื้นที่เปล่า.

แนวทางบางอย่างที่คุณอาจจะใช้มีดังนี้:

- JB entries และ DP หรือ SP entries พื้นฐานสามารถประมวลผลได้โดยการสร้าง เอาต์พุตไฟล์สองไฟล์โดยใช้รูปแบบไฟล์อ้างอิงของฟิลด์ที่ให้มา และการรันคำสั่ง DSPJRN หนึ่งครั้ง สำหรับ JB และหนึ่งครั้งสำหรับ DP หรือ SP. การทำเช่นนี้ช่วยให้คุณสามารถกำหนดโลจิคัลไฟล์ บนฟิลิคัลไฟล์สองไฟล์ และใช้โปรแกรมภาษาชั้นสูงในการประมวลผล described file ภายนอก.
- คุณสามารถประมวลผลเฉพาะ JB entries โดยการสร้างไฟล์โดยใช้ไฟล์อ้างอิงของ ฟิลด์ที่ให้มา (QSYS/QAJBACG4) ไฟล์ใดไฟล์หนึ่งเพื่อสร้าง described file ภายนอกได้. จากนั้นไฟล์นี้สามารถถูกประมวลผลโดยเคียวรียูทิลิตี้หรือ โปรแกรมภาษาชั้นสูงได้.
- คุณสามารถแปลง journal entries ทั้งสองชนิดโดยใช้รูปแบบดีฟอลต์ DSPJRN ของ QJORDJE. จากนั้นคุณสามารถใช้ program-described file เพื่อประมวลผล journal entries ในโปรแกรมภาษาชั้นสูง.

DDS ต่อไปนี้จะกำหนดฟิลิคัลไฟล์สำหรับ JB journal entries โดยใช้ไฟล์อ้างอิงของฟิลด์ QAJBACG4 ใน QSYS. คุณสามารถสร้างไฟล์ (โดยใช้คำสั่ง Create Physical File (CRTPF)) ที่มีชื่อเหมือนกันกับ (QAJBACG4) ชื่อไฟล์ต้นแบบได้.

```
R QWTJAJBE FORMAT(QSYS/QAJBACG4)
```

DDS ต่อไปนี้จะกำหนดฟิลิคัลไฟล์สำหรับ DP หรือ SP journal entries โดยใช้ไฟล์อ้างอิงของฟิลด์ QAPTACG5 ใน QSYS. คุณสามารถสร้างไฟล์ (โดยใช้คำสั่ง CRTPF) ที่มีชื่อเหมือนกันกับ (QAPTACG5) ชื่อไฟล์ต้นแบบได้.

```
R QSPJAPTE FORMAT(QSYS/QAPTACG5)
```

คุณสามารถ ระบุคีย์ฟิลด์ในฟิลิคัลไฟล์ ไฟล์ใดไฟล์หนึ่ง; อย่างไรก็ตาม, ในตัวอย่างนี้, จะใช้โลจิคัลไฟล์สำหรับการจัดลำดับ. ถ้าคุณสร้างฟิลิคัลไฟล์สองไฟล์ (หนึ่งไฟล์ สำหรับ JB และหนึ่งไฟล์สำหรับ DP หรือ SP) ด้วยเมมเบอร์ที่มีชื่อเหมือนกัน, คุณสามารถใช้คำสั่ง DSPJRN ต่อไปนี้เพื่อแปลง entries ได้. สมมติว่าคุณได้สร้างฟิลิคัลไฟล์ ที่มีชื่อเหมือนกันกับชื่อของไฟล์ต้นแบบในไลบรารี YYYY ของคุณ.

```
DSPJRN JRN(QACGJRN) JRNCDE(A) ENTTYP(JB)
OUTPUT(*OUTFILE) OUTFILE(YYYY/QAJBACG4)
DSPJRN JRN(QACGJRN) JRNCDE(A) ENTTYP(SP DP)
OUTPUT(*OUTFILE) OUTFILE(YYYY/QAPTACG5)
```

คุณสามารถควบคุมการใช้ และเกณฑ์การเลือกของคำสั่ง DSPJRN เพื่อให้คุณจะได้ไม่ต้อง แปลง entries เดียวกันหลายครั้ง. ตัวอย่างเช่น, คุณสามารถเลือก entries ทั้งหมดในช่วงของวันที่ที่ระบุ. คุณสามารถแปลง ทั้งหมดที่ cutoff point สำหรับการวิเคราะห์ job accounting ของคุณ, เช่น, รายเดือน. อาจมีการใช้หนึ่งหรือหลาย journal receivers ในระหว่างเดือน. หมายถึง การใช้คำสั่ง DSPJRN ที่เมมเบอร์เดียวกันในแต่ละครั้ง ทำให้เมมเบอร์ถูกลบออกก่อนที่จะใส่เพิ่ม entries ใหม่ใดๆ. ห้ามใช้พารามิเตอร์ JOB ของคำสั่ง DSPJRN เนื่องจาก บาง entries จะถูกสร้างขึ้นสำหรับงานโดยงานระบบ ดังนั้นจึงไม่ปรากฏตามที่คุณคาดไว้.

Allowing the Processing of Both Physical Files:

ป้อน DDS ดังต่อไปนี้เพื่อสร้างโลจิคัลไฟล์เพื่อให้สามารถประมวลผลทั้งสองฟิลส์ได้. การทำเช่นนี้ทำให้คุณสามารถอ่านไฟล์เดี่ยวในลำดับ accounting code และพิมพ์รายงานโดยใช้โปรแกรมภาษาชั้นสูง:

```
R QWTJAJBE PFILE(YYYY/QAJBACG4)
K JACDE
R QSPJAPTE PFILE(YYYY/QAPTACG5)
K JACDE
```

Processing Basic Job Accounting Record:

ถ้าคุณ ต้องการใช้โลจิคัลไฟล์เพื่อประมวลผลเฉพาะเรกคอร์ด job accounting พื้นฐาน ในลำดับ accounting code ตามชื่อผู้ใช้, คุณสามารถป้อน DDS ต่อไปนี้สำหรับโลจิคัลไฟล์:

```
R QWTJAJBE PFILE(YYYY/QAJBACG4)
K JACDE
K JAUSER
```

โลจิคัลไฟล์นี้สามารถถูกประมวลผลโดยเคียวรียูทิลิตี้ หรือโดยโปรแกรมภาษาชั้นสูง. ถ้าเกิดการจบบ่อยอย่างผิดปกติ, ชื่องานที่ถูกต้องใน 30 ไบต์แรกของฟิลด์ JARES ใน journal entry จะอธิบายถึงงานระบบที่บันทึก entry ที่ IPL ถัดไป ไม่ใช่งานที่ใช้รีซอร์ส. ด้วยเหตุผลนี้, การวิเคราะห์ใดๆ ที่ทำบน JB entries จึงควรใช้ฟิลด์ JAJOB, JAUSER, และ JANBR.

การกักคืนและ job accounting

ถ้างานจบบ่อยอย่างผิดปกติ, accounting entry สุดท้ายจะถูกบันทึก และ accounting entries ที่บันทึกก่อนหน้านี้ทั้งหมดจะปรากฏขึ้นในเจอร์นัล. ถ้าเกิดการจบบ่อยอย่างผิดปกติ, accounting data ต่อไปนี้จะหายไปที่ ขั้นตอนการเรวต์ครั้งสุดท้ายหรือ end-of-accounting segment ภายหลังสุด, โดยขึ้นอยู่กับว่าสิ่งใดเกิดขึ้นภายหลังสุด .

- ข้อมูลจำนวนของบรรทัดและหน้าที่พิมพ์
- จำนวนของไฟล์ที่สร้าง
- การดำเนินการ put, get, และอัปเดตฐานข้อมูล
- การดำเนินการอ่านและบันทึกการสื่อสาร
- การดำเนินการ Auxiliary I/O
- เวลา transaction
- จำนวนของฟิลด์ transaction
- เวลาแอนด์ที่ฟ
- เวลาที่หยุดชั่วคราว

หลังจากการจบบ่อยอย่างผิดปกติ, เวลาในการทำงานให้เสร็จสมบูรณ์ในเจอร์นัล จะไม่เหมือนกับเวลาในข้อความ CPF1164. ข้อความจะใช้เวลาที่ใกล้เคียง กับเวลาของการจบบ่อยมากที่สุด, แต่ job accounting journal entries จะถูกส่งไปที่เจอร์นัลในระหว่าง IPL, และเวลาในการทำงานให้สมบูรณ์คือ เวลาในระบบปัจจุบัน, ซึ่งช้ากว่าเวลาที่เกิดการจบบ่อยอย่างผิดปกติ.

ถ้าระบบจบบ่อยอย่างผิดปกติ, journal entries บางอย่างอาจสูญหายได้. entry เหล่านี้คือ entries ที่ถูกบันทึกไปที่เจอร์นัลแต่ไม่ได้บันทึกที่ดิสก์ (สิ่งนี้ เท่ากับ FORCE(*NO) บนคำสั่ง Send Journal Entry (SNDJRNE)). รวมถึงสิ่งดังต่อไปนี้:

- JB entries ที่เกิดขึ้นโดยคำสั่ง Change Accounting Code (CHGACGCDE)
- DP และ SP entries

เมื่อใดก็ตามที่งานเสร็จสมบูรณ์, accounting code entry หลังสุดจะถูก force ที่ ดีส์ (ถ้ามีการระบุ FORCE(*YES) บนคำสั่ง SNDJRNE). เมื่อใดก็ตามที่ accounting entry ถูก forced ที่ ดีส์, entries ก่อนหน้านั้นทั้งหมดใน เจอร์นัล, โดยไม่คำนึงถึงงานที่ทำให้เกิด, จะถูก forced ที่ ดีส์.

Exception

ถ้ามีการระบุเฉพาะ *PRINT accounting บนระบบ, จะไม่มีการทำ FORCE(*YES) journal entries การจบงานใดๆ. ดังนั้น, ถ้า accounting entry ที่สำคัญยิ่งถูกบันทึกโดยคำสั่ง CHGACGCDE คุณต้องตรวจสอบให้แน่ใจว่า entry จะไม่สูญหายไปในกรณีของการจบระบบอย่าง ผิดปกติ, คุณสามารถใช้คำสั่ง SNDJRNE และระบุอ็อปชัน FORCE(*YES). ถ้าไฟล์จะถูกเจอร์นัลที่ accounting journal ด้วย, การเปลี่ยนแปลงฐานข้อมูล ใดๆ จะถูก forced ที่เจอร์นัลเสมอ, และสิ่งนี้ทำให้ accounting entries ก่อนหน้านั้นทั้งหมดถูก forced ด้วย.

ถ้าเกิดการจบระบบอย่างผิดปกติหรือคุณ เปลี่ยน accounting code ของงานที่ไม่ใช่ของคุณเอง, ชื่องานที่ถูกต้องใน 30 ไบต์แรกของฟิลด์ JARES ใน journal entry อธิบายถึง งานระบบที่บันทึก JB entry ที่ IPL ถัดไป ไม่ใช่งานที่ใช้รีซอร์ส. ควรใช้ฟิลด์ JAJOB, JAUSER, และฟิลด์ JANBR สำหรับ การวิเคราะห์.

job accounting journal หรือ journal receiver ที่เสียหาย:

ถ้าเกิดความเสียหายที่เจอร์นัลหรือที่ receiver ปัจจุบันของเจอร์นัลและทำให้ไม่สามารถเจอร์นัล accounting entries, ข้อความ CPF1302 จะถูกส่งไปที่ QSYSOPR message queue, และ accounting data จะถูกบันทึกไปที่ไฟล์บันทึก QHST ในข้อความ CPF1303. งานที่พยายามจะส่ง journal entry จะทำงานต่อไป ตามปกติ. การกู้คืนจากเจอร์นัลหรือ journal receiver ที่เสียหาย เหมือนกับการกู้คืน สำหรับเจอร์นัลอื่น.

เจอร์นัล QACGJRN ไม่ควรถูกจัดสรรโดยงานอื่น. ถ้าเจอร์นัลถูก จัดสรรโดยงานอื่น, journal entry จะถูกเปลี่ยนเป็นเนื้อความ และส่งไปที่ไฟล์บันทึก QHST เป็นข้อความ CPF1303.

คุณสามารถใช้พารามิเตอร์ OUTFILE บนคำสั่ง Display Journal (DSPNJRN) เพื่อบันทึก accounting journal entries ที่ไฟล์ฐานข้อมูลซึ่งคุณสามารถ ประมวลผลได้.

คุณยังสามารถใช้คำสั่ง Receive Journal Entry (RCVJRNE) บนเจอร์นัล QACGJRN เพื่อรับ entries เมื่อ entry ถูกบันทึกไปที่เจอร์นัล QACGJRN. ถ้า job accounting journal หรือ journal receivers เสียหาย, ระบบจะดำเนินการต่อไปและบันทึก accounting data ในบันทึกประวัติ (ที่ผ่านมา). เพื่อกู้คืนจากความเสียหายของเจอร์นัลหรือ journal receiver, ให้ใช้คำสั่ง Work with Journal (WRKJRN). หลังจากการกู้คืนเจอร์นัลหรือ journal receiver ที่เสียหาย, ให้เปลี่ยนค่ากำหนดของระบบ Journal accounting information (QACGLVL) เป็นค่าที่เหมาะสมสำหรับการติดตั้งของคุณ. (ยกเว้นว่าคุณเปลี่ยนค่ากำหนดของระบบ QACGLVL, ระบบจะไม่บันทึกข้อมูล accounting ใน journal receiver ใหม่.)

การเข้าใช้งานข้อความ CPF1303:

เพื่อเข้าใช้งานข้อมูลจากข้อความ CPF1303, ให้สร้างโปรแกรมภาษาชั้นสูง.

ในการกำหนดเรกคอร์ดที่ตรงกับข้อความ CPF1303, ให้รวมฟิลด์ดังต่อไปนี้:

System Time Char (8)

Message Record Number Bin (4)

Qualified Job Name Char (26)
Entry Type (JB, DP, or SP) Char (2)
Length of Data Bin (2)

ตามด้วยฟิลด์:

JAJOB ถึง JASPN สำหรับ JB entries
JAJOB ถึง JABYTE สำหรับ SP และ DP entries

สำหรับโปรแกรมตัวอย่าง, ให้อ้างอิงส่วนในหนังสือคู่มือ CL Programming ที่อธิบายการประมวลผล ไฟล์ QHST สำหรับข้อความความสมบูรณ์ของงาน.

ข้อความ CPF1164 ประกอบไปด้วย เร็กคอร์ดสามเร็กคอร์ดเสมอ และข้อความ CPF1303 ประกอบด้วยเร็กคอร์ดสี่เร็กคอร์ดเสมอ. ข้อมูลที่มีอยู่ในฟิลด์ journal prefix มาตรฐานไม่รวม อยู่ในข้อความนี้. ข้อมูลทั้งหมดที่ต้องการคือข้อมูลเกี่ยวกับการจบงาน, วันที่, และเวลา. สามารถดูข้อมูลนี้ได้ ใน เร็กคอร์ด 1 ของข้อความ CPF1303.

การอ้างอิง

คุณอาจพบว่าบทความ Information Center เหล่านี้มีประโยชน์ในขณะที่ใช้การจัดการระบบงาน.

(IBM iSeries Information Center, Version 5 Release 4 (V5R4) → Systems management → Work management → Reference)

ตารางงานเซิร์ฟเวอร์

ใช้ตารางเซิร์ฟเวอร์นี้เป็นส่วนอ้างอิงในการหาวิธีการแก้ปัญหาเซิร์ฟเวอร์, งานเซิร์ฟเวอร์, รายละเอียดของงาน, และระบบย่อยกับรายการอื่น.

ตัวค้นหาค่ากำหนดของระบบ

ใช้ตัวค้นหาค่ากำหนดของระบบ OS/400 ในการค้นหาข้อมูลเกี่ยวกับค่ากำหนดของระบบอย่างรวดเร็ว. คุณสามารถค้นหา ประเภทของค่ากำหนดของระบบเมื่อค่าปรากฏใน iSeries Navigator หรือค้นหาชื่อค่ากำหนดของระบบ ที่คุณใช้ในอินเตอร์เฟซแบบอักษร.

Work Management Application Programming Interface (API)

หน้า Work Management APIs แสดงรายการของ APIs ที่ดึงข้อมูล และจัดการ; งาน, พูลของหน่วยความจำระบบย่อย, คิวงานระบบย่อย, พื้นที่ ข้อมูล, เน็ตเวิร์กแอ็ดทริบิวต์, สถานะของระบบ, ค่ากำหนดของระบบ, และ flight recorders. และรวมถึงรายการของโปรแกรมทางออกการจัดการระบบงานด้วย.

ตัวค้นหา IPL system reference code (SRC)

ใช้ตัวค้นหา IPL SRC ในการค้นหาข้อมูลเกี่ยวกับข้อความ SRC ที่แสดงบนเซิร์ฟเวอร์ของคุณเมื่อคุณทำ IPL. SRCs บ่งชี้สถานะ ของ IPL และมักมีประโยชน์ในการวิเคราะห์ปัญหา. คุณสามารถค้นหา SRC โดยใช้ชื่อ, หรือแสดงรายการของ SRCs ที่ใช้ทั่วไปได้.

การแก้ไขปัญหา

ใช้ข้อมูลในหัวข้อนี้เพื่อช่วยคุณแก้ไขปัญหาทั่วไปบางอย่าง ที่เกิดขึ้นในการจัดการระบบงาน.

งานของฉันทหยุดทำงาน

หัวข้อนี้อธิบายสาเหตุที่อาจเป็นไปได้ที่ทำให้งานหยุดทำงาน.

งานกำลังรอที่จะล็อกบนอ็อบเจ็กต์

- วิธีการวินิจฉัย: ให้ดูสถานะของงานใน iSeries Navigator; ดู การกำหนด สถานะของงาน. งานที่กำลังรอการล็อกจะมีสถานะเป็น *กำลังรอการล็อก*.
- วิธีแก้ไข: ดูรายการของอ็อบเจ็กต์ที่ล็อกสำหรับงาน เพื่อกำหนดอ็อบเจ็กต์ที่ งานกำลังรอที่จะล็อก. แล้วใช้การดำเนินการแบบ Lock Holders กับอ็อบเจ็กต์เพื่อระบุว่างานใดที่กำลังล็อกอ็อบเจ็กต์นั้นอยู่. จากนั้นคุณต้องกำหนดว่าทำไมงานดังกล่าวจึงต้องถูกล็อกเอาไว้, และจะทำเช่นไรเพื่อปลดล็อกนั้นออกให้ได้.

งานถูกพัก

- วิธีการวินิจฉัย: ให้ดูสถานะของงานใน iSeries Navigator; ดู การกำหนด สถานะของงาน.
- วิธีแก้ไข: คลิกขวาที่งานและคลิก **Release**.

ต่อไปนี้เป็นสาเหตุที่พอจะเป็นไปได้ที่งานในคิวงานอาจหยุดการทำงาน :

คิวงานถูกพัก

- วิธีการวินิจฉัย: ดูสถานะของคิวงานใน iSeries Navigator;
- วิธีแก้ไข:
1. ย้ายงานไปไว้ในคิวงานที่ไม่ได้ถูกพักไว้; ให้ดูการย้ายงานไปยังคิวงานอื่น.
 2. รีเซ็ตคิวงาน. หากต้องการทำเช่นนั้น, ให้คลิกขวาที่งานและคลิก**Release**.

คิวงานไม่ได้ถูกจัดสรรไว้ให้โดยระบบย่อยแอ็คทีฟ

- วิธีการวินิจฉัย: ดูสถานะของคิวงานใน iSeries Navigator.
- วิธีแก้ไข:
1. ย้ายงานไปยังคิวงานที่ได้รับการจัดสรรไว้โดยระบบย่อยแอ็คทีฟ; ให้ดู การย้ายงานไปยังคิวงานอื่น.
 2. สตาร์ทระบบย่อยที่มี job queue entry สำหรับคิวงานดังกล่าว; ให้ดูที่ การเริ่มระบบย่อย.
 3. ใส่ job queue entry ของคิวงานดังกล่าวลงไปในระบบย่อยแอ็คทีฟ โดยใช้คำสั่ง Add Job Queue Entry (ADDJOBQE).

ครบจำนวนสูงสุดของระบบย่อย

- วิธีการวินิจฉัย: ดูค่างานแอ็คทีฟสูงสุดสำหรับระบบย่อยใน iSeries Navigator. หากต้องการทำเช่นนั้น, ให้คลิกขวาที่ระบบย่อยและคลิก **Properties**.
- วิธีแก้ไข:
1. ย้ายงานไปไว้ในคิวงานอื่น; ให้ดู การย้ายงานไปยังคิวงานอื่น.
 2. เพิ่มค่าสูงสุด. หากต้องการทำเช่นนั้น, ให้ใช้คำสั่ง Change Subsystem Description (CHGSBSD).

ครบจำนวนสูงสุดของคิวงาน

- วิธีการวินิจฉัย: ดูค่างานแอ็คทีฟสูงสุดสำหรับคิวงานใน iSeries Navigator. หากต้องการทำเช่นนั้น, ให้คลิกขวาที่คิวงานและคลิก **Properties**. แล้วจึงเลือกแท็บ **Activity**.
- วิธีแก้ไข:
1. ย้ายงานไปไว้ในคิวงานอื่น; ให้ดู การย้ายงานไปยังคิวงานอื่น.
 2. เพิ่มค่าสูงสุด. หากต้องการทำเช่นนั้น, ให้ใช้คำสั่ง Change Job Queue Entry (CHGJOBQE).

ครบค่าสูงสุดสำหรับระดับความสำคัญ

- วิธีการวินิจฉัย: ระบุระดับความสำคัญของคิวงาน ของงานโดยการดูที่คุณสมบัติของงานนั้น. จากนั้นดูงานแอคทีฟสูงสุดด้วยค่าระดับความสำคัญงานสำหรับคิวงาน ใน iSeries Navigator. หากต้องการทำเช่นนั้น, ให้คลิกขวาที่คิวงานและคลิก **Properties**. จากนั้นเลือกแท็บ **Activity** และคลิกปุ่ม **Advanced**.
- วิธีแก้ไข:
1. ย้ายงานไปไว้ในคิวงานอื่น; ให้ดู การย้ายงานไปยังคิวงานอื่น.
 2. เปลี่ยนระดับความสำคัญคิวงานของงาน; ดู การระบุ ระดับความสำคัญของงานภายในคิวงาน.
 3. เพิ่มค่าสูงสุด. หากต้องการทำเช่นนั้น, ให้ใช้คำสั่ง **Change Job Queue Entry (CHGJOBQE)**.

งานของฉันประสบปัญหาทำงานช้า

ข้อมูลในบทความนี้อธิบายเหตุผลที่เป็นไปได้ว่า เพราะเหตุใด งานจึงมีประสิทธิภาพไม่ดี.

หน่วยความจำไม่เพียงพอ

- วิธีการวินิจฉัย: ให้อูที่คุณสมบัติของงานเพื่อระบุว่างานกำลังรันอยู่ในพูลหน่วยความจำ อันใด. จากนั้นดูคุณสมบัติของพูลหน่วยความจำใน iSeries Navigator, โดยดูที่ **Check memory pool use**. อัตราความผิดพลาดสูงในพูลบอกได้ว่ามีหน่วยความจำในพูลที่ไม่ เพียงพอ, หรือในพูลมีงานเป็นจำนวนมากเกินไปที่แย่งใช้หน่วยความจำ กันอยู่.
- วิธีแก้ไข:
1. ให้เปิดตัวปรับแต่งระบบ ในกรณีที่คุณไม่ได้ใช้งานอยู่. ค่ากำหนดของระบบ QPFRADJ จะปรับพูลหน่วยความจำและ activity level โดยอัตโนมัติ.
 2. ถ้าเป็นไปได้, ให้ปรับแต่งพูลด้วยตัวคุณเองโดยให้เพิ่มจำนวนหน่วย ความจำในพูล หรือลด activity level ของพูลหน่วยความจำลง. คุณอาจต้องการตรวจสอบพูลเครื่องเพื่อตรวจสอบขนาดของหน่วยความจำที่ใช้งานว่าไม่มีผลต่องานทั้งหมดบนระบบ.

Activity level ต่ำเกินไป

- วิธีการวินิจฉัย: ดูที่คุณสมบัติของงานเพื่อระบุ สถานะและดูว่างานกำลังรันอยู่ในพูลหน่วยความจำอันใด. ถ้างานแสดงสถานะเป็น *Waiting for activity level*, ให้อูที่คุณสมบัติของพูลหน่วยความจำใน iSeries Navigator, โดยดูที่ **Check memory pool use**. อัตราการเปลี่ยนไปสู่สถานะการไม่มีสิทธิในพูลที่มีสูงนั้นเป็นการชี้ให้เห็นว่าในพูลมีจำนวนงานมากเกินไปซึ่งแย่งกันใช้หน่วยความจำกันอยู่.
- วิธีแก้ไข:
1. ให้เปิดตัวปรับแต่งระบบ ในกรณีที่คุณไม่ได้ใช้งานอยู่. ค่ากำหนดของระบบ QPFRADJ จะปรับพูลหน่วยความจำและ activity level โดยอัตโนมัติ.
 2. ให้ปรับพูลด้วยตนเองโดยการเพิ่ม activity level ของพูลหน่วยความจำ.

รีซอร์ส CPU ไม่เพียงพอ

- วิธีการวินิจฉัย: ดูคอลัมน์ CPU % สำหรับงานและงานอื่นในรายการ **Active Jobs** ของ iSeries Navigator. หากระบบยุ่งมาก, งานของคุณอาจได้รับรีซอร์ส CPU ไม่เพียงพอที่จะทำงานให้ เสร็จสิ้นได้.
- วิธีแก้ไข:
1. ถ้าเป็นไปได้, ให้จบหรือพักงานที่ไม่จำเป็นบนระบบ.
 2. หากงานจำนวนหนึ่งคือการทำงานที่เน้นใช้ CPU มาก, ให้เปลี่ยนระดับความสำคัญในการรันงานเหล่านี้ (ค่าระดับความสำคัญการรันที่สูงขึ้นเท่ากับระดับความสำคัญการรันที่ต่ำลงของงาน).

อ็อปชันการเพจพูลหน่วยความจำ

- วิธีการวินิจฉัย: หากแอสพลเคชันคือการทำงานที่เน้นใช้ดิสก์, หาก CPU ทำงานอย่างไม่เต็มประสิทธิภาพและมีหน่วยความจำไม่เพียงพอ, การใช้ expert cache จะมีประโยชน์มาก.
- วิธีแก้ไข: expert cache สามารถเปิดใน iSeries Navigator โดยการเปลี่ยนอ็อปชัน Paging สำหรับ shared memory pool เป็น Calculated. อ็อปชัน Paging อยู่ที่แท็บ Configuration ของหน้า Properties ของพูลหน่วยความจำและจะมีอยู่บนพูลแบบแบ่งใช้เท่านั้น (ไม่ใช่พูลส่วนตัว).

ระดับความสำคัญในการรันของงานต่ำ

- วิธีการวินิจฉัย: ดูคุณสมบัติ ของงานเพื่อระบุ ระดับความสำคัญของงานที่สัมพันธ์กับงานอื่นๆ ในระบบ.
- วิธีแก้ไข: ถ้าหากงานมีระดับความสำคัญการรันที่ต่ำ (ค่าสูง) โดยเปรียบเทียบกับงานอื่นๆ และไม่ได้ใช้ CPU มากนัก เพราะ งานที่มีระดับความสำคัญที่สูง (ค่าต่ำ) จะใช้ CPU หรือสล็อตส่วนใหญ่อยู่แล้ว, คุณอาจจำเป็นต้องเพิ่มระดับความสำคัญของงานให้มากขึ้น; ให้ดูหัวข้อ Job properties . และ, บนระบบที่มีการใช้งาน CPU สูงและงานมีระดับความสำคัญการรัน ต่ำ, การตั้งค่ากำหนดของระบบ Dynamically adjust job priorities within priority bands (QDYNPTYSCD) และ Dynamically adjust job priorities of interactive jobs (QDYNPTYADJ) อาจจะมีประโยชน์ .

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับประสิทธิภาพ, ให้ดูที่ Performance. หากคุณต้องการข้อมูลเพิ่มเติม เกี่ยวกับวิธีการปรับประสิทธิภาพบนระบบของคุณ, ให้ดู Tune performance.

การสืบสวนงานแบบ prestart

หัวข้อนี้แนะนำเสนอขั้นตอนที่ช่วยให้คุณในการตอบคำถาม, "ฉันจะ ค้นหาผู้ใช้จริงของงานแบบ prestart และกำหนดรีซอร์สที่ใช้โดยงานแบบ prestart นั้นได้อย่างไร?"

iSeries Navigator

คุณสามารถใช้ iSeries Navigator ส่วนมุมมองการจัดการระบบงานและการมอนิเตอร์ Management Central เพื่อ การวิเคราะห์ เวลาจริงของสิ่งที่เกิดขึ้นบนระบบของคุณได้.

1. ใช้มุมมอง Server Jobs เพื่อดูงานเซิร์ฟเวอร์ที่แอคทีฟและผู้ใช้ ปัจจุบัน. (My Connections → server → Work Management → Server Jobs)
 - คลิกขวาที่ Server Jobs container และเลือก **Customize this view** → **Columns** และตรวจให้แน่ใจว่า Current User, Total CPU Time, และ Total CPU DB Time อยู่ในรายการ **Columns to be displayed** .
 - ถ้ารายการงานเซิร์ฟเวอร์ที่แอคทีฟของคุณมีขนาดใหญ่, คุณสามารถจำกัดสิ่งที่ จะแสดงโดยใช้ชื่องาน, หมายเลขงาน, ผู้ใช้ปัจจุบัน, หรือสถานะ. คลิกขวาที่ Server Jobs container และคลิก **Customize this view** → **Include** .
 - คุณสามารถเรียงลำดับการแสดงผลของรายการงานเซิร์ฟเวอร์ที่แอคทีฟได้โดยการ คลิกบนหัวข้อคอลัมน์.
 - คุณสามารถเซตจอสถิติให้รีเฟรชโดยอัตโนมัติในช่วงเวลาที่สม่ำเสมอได้. (My Connections → คลิกขวาที่เซิร์ฟเวอร์ → **Customize this view** → **Auto Refresh**) สิ่งนี้ช่วยให้คุณทราบถึงการเปลี่ยนแปลงแบบทันที.

หลังจากที่คุณได้กำหนดตำแหน่งงานที่สนใจแล้ว, คุณสามารถคลิกขวาที่งาน และเข้าใช้งาน call stack ของงาน, บันทึกการใช้งาน, สถิติประสิทธิภาพที่ลดลง, SQL statement ล่าสุด, และคุณสมบัติของงาน.

2. ใช้ Management Central เพื่อตั้งค่าการมอนิเตอร์ระบบที่จะมอนิเตอร์ CPU Utilization โดยรวม. (Management Central → Monitors → คลิกขวาที่ระบบ → New Monitor)
 - ในขณะที่การมอนิเตอร์รันอยู่, คุณสามารถคลิกจุดใดๆจุดหนึ่งเพื่อดูรายละเอียด ระดับถัดไปได้. ตัวอย่างเช่น, เมื่อมอนิเตอร์ CPU Utilization คุณสามารถแสดงรายการของงานที่มี CPU utilization สูงสุดได้. จากนั้นคุณสามารถ คลิกขวาที่

งานที่กำลังใช้ CPU เป็นจำนวนมาก และคลิก **Properties** เพื่อแสดงคุณสมบัติของงาน. (ให้ดูคำอธิบายทางออนไลน์ สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับวิธีการใช้การมอนิเตอร์ระบบ.)

3. ใช้ Management Central เพื่อตั้งค่าการมอนิเตอร์งานเพื่อมอนิเตอร์งาน เซิร์ฟเวอร์ที่เหมาะสม และแจ้งให้คุณทราบเมื่องานเหล่านั้นเริ่มใช้รีซอร์สมากเกินไป. (Management Central → Monitors → คลิกขวาที่งาน → New Monitor)

I อินเทอร์เน็ตแบบอักษระ

I คำสั่ง: Work with Active Job (WRKACTJOB)

I คำสั่งนี้จะแสดงผู้ใช้ปัจจุบันของ initial thread (ซึ่งคืองานเมื่องานถูก single threaded). นี่เป็นข้อมูลเดียวกับที่แสดงอยู่ใน GUI.

ข้อมูลที่เกี่ยวข้องกับการจัดการระบบงาน

การจัดการระบบงานเป็นสิ่งที่ซับซ้อน. เนื่องจากมีส่วนประกอบเป็นจำนวนมาก. ข้อมูลต่อไปนี้เป็นรายการของหัวข้อต่างๆ ที่พบได้ใน IBM iSeries Information Center, Version 5 Release 4 (V5R4). หัวข้อเหล่านี้สามารถช่วยให้คุณจัดการกับส่วนต่างๆ ที่มีความเกี่ยวข้องกับระบบของคุณ, แต่อยู่นอกพื้นที่การจัดการระบบงาน.

(IBM iSeries Information Center, Version 5 Release 4 (V5R4) → Systems management → Work management → Related information for work management)

การทำเน็ตเวิร์ก

ความเข้าใจถึงเทคโนโลยีการทำเน็ตเวิร์กของคุณนับเป็นส่วนที่สำคัญยิ่ง ของโซลูชันอ็อบิซิเนสทั้งหมดของบริษัทของคุณ. เรียนรู้วิธีการเชื่อมต่อธุรกิจของคุณกับ อินเทอร์เน็ต, ตั้งค่าอีเมล, และให้บริการอ็อบเจกต์มัลติมีเดียแก่ Web browser clients. คุณสามารถรวมไฟล์และการให้บริการพิมพ์, การจัดการโปรไฟล์ผู้ใช้, และการดำเนินการเน็ตเวิร์ก. ค้นหาข้อมูลเกี่ยวกับเซิร์ฟเวอร์ Windows(R) ที่สามารถรวม เข้าในเซิร์ฟเวอร์, และอ่านเกี่ยวกับการรักษาความปลอดภัยที่น่าเสนอซึ่ง สามารถช่วยปกป้องรีซอร์สของคุณได้.

Retrieve network attributes (API)

Retrieve Network Attributes (QWCRNETA) API ช่วยคุณดึงข้อมูล เน็ตเวิร์กแอตทริบิวต์.

การปรับประสิทธิภาพ

การทำความเข้าใจกระบวนการต่างๆ ทั้งหมดที่มีผลต่อประสิทธิภาพของระบบ นับเป็นสิ่งที่ท้าทายสำหรับผู้ใช้ที่ไม่มีประสบการณ์. การแก้ไขปัญหาเกี่ยวกับประสิทธิภาพ ต้องใช้ชุดของเครื่องหมายจำนวนมากอย่างมีประสิทธิภาพ, แต่ละเครื่องมือมีเซต ของข้อกำหนดและฟังก์ชันที่ได้รับการสนับสนุนไม่ซ้ำกัน. แม้ว่า你能รวบรวมและวิเคราะห์ ข้อมูลประสิทธิภาพแล้ว, แต่อาจไม่ทราบว่าจะจัดการกับข้อมูลเหล่านั้น อย่างไร. หัวข้อนี้จะช่วยให้คุณทราบถึงภารกิจและเครื่องมือที่เกี่ยวข้องกับ การจัดการประสิทธิภาพ.

Performance explorer

Performance explorer เป็นเครื่องมือการรวบรวมข้อมูลที่ช่วยผู้ใช้ในการระบุสาเหตุของปัญหาประสิทธิภาพ ซึ่งไม่สามารถระบุโดยการรวบรวม ข้อมูลโดยใช้ Collection Services หรือโดยการวิเคราะห์แนวโน้มทั่วไป.

การบริหารเวลา

การบริหารเวลาช่วยคุณในการบริหารเวลาบนเซิร์ฟเวอร์ IBM eServer™ i5 ของคุณ. ภายในส่วนประกอบการบริหารเวลาของ iSeries™ Navigator, คุณ สามารถทำงานกับเขตเวลาและฟังก์ชันการปรับเปลี่ยนเวลาได้.

คำกำหนดของระบบ i5/OS

คำกำหนดของระบบคือชั้นของข้อมูลที่มีผลกระทบต่อสภาวะแวดล้อม การปฏิบัติการในระบบ iSeries™ ทั้งระบบ. คำกำหนดของระบบไม่ใช่ข้อจำกัดบนระบบ. แต่, คำกำหนดของระบบมีข้อมูลการควบคุมสำหรับการดำเนินการของส่วนบางส่วนจากระบบ.

ภาคผนวก. หมายเหตุ

ข้อมูลนี้ถูกพัฒนาขึ้นสำหรับผลิตภัณฑ์และบริการที่เสนอขายในประเทศสหรัฐอเมริกา.

IBM อาจไม่เสนอขายผลิตภัณฑ์, บริการ, หรือคุณสมบัติต่างๆ ที่กล่าวถึงในเอกสาร นี้ในประเทศอื่น. ปรีक्षाตัวแทน IBM ในท้องถิ่นของคุณสำหรับข้อมูลเกี่ยวกับผลิตภัณฑ์ และบริการที่เสนอขายในพื้นที่ของคุณในปัจจุบัน. การอ้างถึงผลิตภัณฑ์, โปรแกรม, หรือบริการของ IBM ไม่ได้หมายความว่าต้องใช้ เฉพาะผลิตภัณฑ์, โปรแกรม, หรือบริการที่เป็นของ IBM เท่านั้น. ผลิตภัณฑ์, โปรแกรม, หรือบริการอื่นที่ทำงานได้เทียบเท่า และไม่ละเมิดลิขสิทธิ์ของ IBM ก็สามารถใช้ได้. อย่างไรก็ตาม, ถือเป็นความรับผิดชอบของผู้ใช้ที่จะประเมินและตรวจสอบการดำเนินการของผลิตภัณฑ์, โปรแกรม, หรือบริการที่ไม่ใช่ของ IBM.

IBM อาจมีสิทธิบัตรหรือเอกสารการขอสิทธิบัตร ที่ครอบคลุมสิ่งที่อธิบายในเอกสารนี้. การตกแต่งเอกสารใหม่ไม่ได้ทำให้คุณได้สิทธิของสิทธิบัตรเหล่านั้น. คุณสามารถสอบถามเกี่ยวกับไลเซนส์, โดยเขียน, และส่งไปที่:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

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

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

ย่อหน้าต่อไปนี้ไม่ใช่กับสหราชอาณาจักร 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. บางรัฐไม่อนุญาตการปฏิเสธการรับประกันอย่างชัดแจ้ง หรือโดยนัยในการทำการซื้อขายบางอย่าง, ดังนั้น, ข้อความข้างต้นนี้อาจไม่ได้ใช้กับคุณ.

ข้อมูลนี้อาจมีความไม่ถูกต้องทางเทคนิคหรือความผิดพลาดทางการพิมพ์. มีการเปลี่ยนแปลงข้อมูลในเอกสารนี้เป็นระยะๆ และการเปลี่ยนแปลงเหล่านี้จะถูกรวมในเอกสารฉบับแก้ไขใหม่. IBM อาจทำการปรับปรุง และ/หรือเปลี่ยนแปลงในผลิตภัณฑ์ และ/หรือโปรแกรมที่อธิบายในเอกสารนี้ได้ตลอดเวลาโดยไม่ต้องแจ้งให้ทราบ.

การอ้างถึงเว็บไซต์ที่ไม่ใช่ของ IBM นั้นถูกใช้เพื่อความสะดวกเท่านั้น และไม่ได้มีการรับรองเว็บไซต์เหล่านั้น. เนื้อหาในเว็บไซต์เหล่านั้น ไม่ได้เป็นส่วนหนึ่งของเนื้อหาสำหรับผลิตภัณฑ์นี้ของ IBM และการใช้เว็บไซต์เหล่านั้นเป็นความเสี่ยงของตัวเอง.

IBM อาจใช้หรือเผยแพร่ข้อมูลใดๆ ที่คุณให้ไว้ในทางที่ IBM เชื่อว่าเหมาะสม โดยไม่ก่อให้เกิดข้อผูกมัดใดๆ กับคุณ.

สำหรับผู้ที่มิใช่เจ้าของโปรแกรมนี้ที่ต้องการมีข้อมูลเกี่ยวกับโปรแกรมสำหรับ จุดประสงค์ให้ทำงานได้: (i) การแลกเปลี่ยนข้อมูลระหว่างโปรแกรมที่ถูกสร้างขึ้นอย่าง เป็นอิสระและโปรแกรมอื่น (รวมทั้งโปรแกรมนี้) และ (ii) การใช้ข้อมูลร่วมกันซึ่งมีการแลกเปลี่ยนข้อมูล, ควรติดต่อ:

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

ข้อมูลเหล่านี้อาจพร้อมใช้งาน, ภายใต้ข้อตกลงและเงื่อนไขที่เหมาะสม, ซึ่งรวมถึงบางกรณี, เช่น การจ่ายค่าธรรมเนียม.

- | โปรแกรมไลเซนส์ที่อธิบายในข้อมูลนี้ และเนื้อหาของไลเซนส์ทั้งหมด ถูกจัดหามาโดยบริษัท IBM ภายใต้เงื่อนไขของ IBM
- | Customer Agreement, IBM International Program Licensed Agreement, IBM License Agreement for Machine Code, หรือ
- | ข้อตกลงอื่นๆ ที่เทียบเท่าระหว่าง IBM และคุณ.

ข้อมูลประสิทธิภาพใดๆ ที่มีอยู่ในนี้ถูกกำหนดอยู่ในสภาพแวดล้อมที่ถูกควบคุม. ดังนั้น, ผลที่ได้จากสภาพแวดล้อมของการปฏิบัติการอื่นอาจแตกต่างกันเป็นอย่างมาก. บางระบบการวัดอาจทำบนระบบที่พัฒนาแล้วและไม่มีการรับรองว่าระบบการวัดเหล่านี้ จะเหมือนกันกับบนระบบที่มีทั่วไป. นอกเหนือจากนี้, การวัดบางอย่างอาจเป็นการประมาณ ผ่านการคาดการณ์. ซึ่งผลที่แท้จริงอาจแตกต่างกัน. ผู้ใช้เอกสารนี้ควรตรวจสอบข้อมูลที่ใช้ได้สำหรับสภาวะแวดล้อมเฉพาะสำหรับผู้ใช้.

ข้อมูลเกี่ยวกับผลิตภัณฑ์ที่ไม่ใช่ของ IBM ได้รับมาจากซัพพลายเออร์ของผลิตภัณฑ์เหล่านั้น, การประกาศทางสาธารณะชนหรือแหล่งที่พร้อมใช้งานสำหรับสาธารณะอื่นๆ. IBM ไม่ได้ทดสอบผลิตภัณฑ์เหล่านั้น และไม่สามารถยืนยันความถูกต้องของประสิทธิภาพการทำงาน, ความเข้ากันได้ หรือข้อร้องเรียนอื่นๆ ที่เกี่ยวข้อง กับผลิตภัณฑ์ที่ไม่ใช่ของ IBM. คำถามเกี่ยวกับผลิตภัณฑ์ที่ไม่ใช่ของ IBM ควรถามไปที่ซัพพลายเออร์ของ ผลิตภัณฑ์เหล่านั้น.

ข้อความทั้งหมดที่เกี่ยวข้องกับทิศทางในอนาคตหรือความตั้งใจของ IBM สามารถ เปลี่ยนแปลงหรือยกเลิกได้โดยไม่ต้องแจ้งให้ทราบ, และแสดงถึงเป้าหมายและวัตถุประสงค์เท่านั้น.

ข้อมูลนี้มีตัวอย่างของข้อมูลและรายงานที่ใช้ในการดำเนินธุรกิจประจำวัน. เพื่อแสดงให้เห็นอย่างสมบูรณ์ที่สุดที่เป็นไปได้, ตัวอย่างเหล่านี้ประกอบด้วย ชื่อของแต่ละราย, ชื่อของบริษัท, ตราสินค้าและผลิตภัณฑ์. ชื่อทั้งหมดเหล่านี้ถูกทำขึ้น และคล้ายคลึงกับชื่อและที่อยู่ของหน่วยธุรกิจจริง.

COPYRIGHT LICENSE:

ข้อมูลนี้ประกอบด้วยโปรแกรมแอปพลิเคชันตัวอย่างในภาษาต้นฉบับ, ซึ่งแสดงเทคนิคของโปรแกรมบนระบบปฏิบัติการที่หลากหลาย. คุณอาจคัดลอก, ดัดแปลง และ แจกจ่ายโปรแกรมตัวอย่างเหล่านี้ในรูปแบบใดๆ โดยไม่ต้องจ่ายค่าตอบแทนแก่ IBM, สำหรับ จุดประสงค์ของการพัฒนา, การใช้งาน, การตลาด หรือการแจกจ่ายแอปพลิเคชันโปรแกรม ตาม แอปพลิเคชันโปรแกรมมิงอินเตอร์เฟซ สำหรับแพลตฟอร์มปฏิบัติการที่โปรแกรมตัวอย่างถูกเขียน. ตัวอย่างเหล่านี้ ไม่ได้ผ่านการทดสอบภายใต้ทุกสถานการณ์. ดังนั้น IBM ไม่สามารถรับประกัน หรือกล่าวถึงความเชื่อถือได้, ความสามารถในการบริการ, หรือการทำงานของโปรแกรมเหล่านี้.

แต่ละสำเนาหรือบางส่วนของโปรแกรมตัวอย่าง หรืองานใดๆ ที่มาจากโปรแกรมเหล่านี้ ต้องมีข้อความแสดงลิขสิทธิ์ ดังนี้:

© (ชื่อบริษัทของคุณ) (ปี). บางส่วนของโค้ดนี้ได้มาจาก IBM Corp. ตัวอย่างโปรแกรม. © ลิขสิทธิ์ของ IBM Corp. ใส่ข้อมูลปี. สงวนลิขสิทธิ์ทั้งหมด.

คู่มือการจัดการระบบงานสำหรับ Programming Interfaces อนุญาตให้ผู้ใช้บันทึกโปรแกรมเพื่อรับการให้บริการของ IBM i5/OS.

ถ้าคุณกำลังดูสำเนาชั่วคราวของข้อมูล, ภาพและสีที่แสดงอาจไม่ปรากฏ.

เครื่องหมายการค้า

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

- | eServer
- | i5/OS
- | IBM
- | iSeriesSystem i5

- | Intel, Intel Inside (โลโก้), MMX, และ Pentium เป็นเครื่องหมายการค้าของ Intel Corporation ในสหรัฐอเมริกา, ประเทศอื่น, หรือทั้งสองกรณี.

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

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

- | Linux เป็นเครื่องหมายการค้าของ Linus Torvalds ในสหรัฐอเมริกา, ประเทศอื่น, หรือทั้งสองกรณี.

UNIX เป็นเครื่องหมายการค้าจดทะเบียนของ The Open Group ในประเทศสหรัฐอเมริกาและประเทศอื่นๆ.

ชื่ออื่นๆ ของบริษัท, ผลิตภัณฑ์, และการบริการ อาจเป็นเครื่องหมายการค้า หรือเครื่องหมายการบริการของผู้อื่น.

ข้อกำหนดและเงื่อนไข

คำอนุญาตในการใช้สิ่งพิมพ์เผยแพร่เหล่านี้เป็นไปตามข้อกำหนดและเงื่อนไขต่อไปนี้.

การใช้งานเป็นส่วนตัว: คุณสามารถทำซ้ำสิ่งพิมพ์เผยแพร่เหล่านี้สำหรับการส่วนตัว, มิใช่เพื่อการพาณิชย์โดยมีข้อแม้ว่าจะจะมีการคงค่าประกาศเกี่ยวกับกรรมสิทธิ์ทั้งหมด. คุณไม่สามารถแจกจ่าย, แสดง หรือสร้างงานที่สืบเนื่องจาก, หรือจากบางส่วนของสิ่งพิมพ์เผยแพร่เหล่านี้, โดยมิได้รับอนุญาตจาก IBM.

การใช้งานในเชิงพาณิชย์: คุณอาจทำซ้ำ แจกจ่าย และแสดงสิ่งพิมพ์เผยแพร่นี้ได้เฉพาะภายในองค์กรของคุณ โดยมีเงื่อนไขว่าจะต้องคงค่าประกาศเกี่ยวกับกรรมสิทธิ์. ไม่อนุญาตให้คุณผลิตงานที่มาจากสิ่งพิมพ์เผยแพร่, หรือทำซ้ำ, แจกจ่าย หรือแสดงสิ่งพิมพ์เผยแพร่หรือส่วนใดส่วนหนึ่งภายนอกองค์กรของคุณ, โดยมิได้รับความยินยอมอย่างชัดแจ้งจาก IBM.

นอกเหนือจากคำอนุญาตที่ได้แสดงไว้ในที่นี้, IBM ไม่ได้ให้อำนาจดำเนินการ, ไลเซนส์หรือสิทธิ์อื่นใด ทั้งโดยชัดแจ้งและโดยนัย กับสิ่งพิมพ์เผยแพร่หรือสารสนเทศ, ข้อมูล, ซอฟต์แวร์ หรือทรัพย์สินทางปัญญาอื่นๆ ที่อยู่ภายในที่นี้.

IBM ขอสงวนสิทธิ์ในการเพิกถอนคำอนุญาตที่ให้ไว้ในที่นี้, เมื่อใดก็ตามที่พิจารณาแล้วว่า, การใช้สิ่งพิมพ์เผยแพร่เหล่านี้ก่อให้เกิดความเสียหายต่อผลประโยชน์ของบริษัท, หรือเมื่อ IBM ได้พิจารณาแล้วว่า ไม่มีการปฏิบัติตามข้อกำหนดข้างต้นอย่างเหมาะสม.

คุณไม่สามารถดาวน์โหลด, ส่งออกหรือส่งออกข้อมูลนี้เข้าได้ ยกเว้นจะได้ปฏิบัติตามกฎหมายและข้อบังคับที่ผลบังคับใช้, รวมไปถึงกฎหมายและข้อบังคับในการส่งออกของสหรัฐอเมริกา.

IBM ไม่รับประกันเกี่ยวกับเนื้อหาของสิ่งพิมพ์เผยแพร่เหล่านี้. สิ่งพิมพ์เผยแพร่เหล่านี้ถูกจัดทำขึ้น "ตามสภาพที่เป็น" โดยไม่มีการรับประกันใดๆ ทั้งโดยชัดแจ้งหรือโดยนัย ตลอดจนไม่มีการรับประกันโดยนัยต่อการนำไปจัดจำหน่าย, การไม่ละเมิดสิทธิ, และความเหมาะสมสำหรับวัตถุประสงค์เฉพาะอย่างใดอย่างหนึ่ง.



พิมพีในสหรัฐอเมริกา